Fault Detection by Signal Reconstruction in Nuclear Power Plants

Ibrahim Ahmed, Enrico Zio and Gyunyoung Heo

Abstract

In this work, the recently developed auto associative bilateral kernel regression (AABKR) method for on-line condition monitoring of systems, structures, and components (SSCs) during transient process operation of a nuclear power plant (NPP) is improved. The advancement enhances the capability of reconstructing abnormal signals to the values expected in normal conditions during both transient and steady-state process operations. The modification introduced to the method is based on the adoption of two new approaches using dynamic time warping (DTW) for the identification of the time position index (the position of the nearest vector within the historical data vectors to the current on-line query measurement) used by the weighted-distance algorithm that captures temporal dependences in the data. Applications are provided to a steady-state numerical process and a case study concerning sensor signals collected from a reactor coolant system (RCS) during start-up operation of a NPP. The results demonstrate the effectiveness of the proposed method for fault detection during steady-state and transient operations.

1. Introduction

In a nuclear power plant (NPP), accurate situation awareness of key systems, structures, and components (SSCs) is important for safety, reliability, and economics which are key drivers for operation. However, faults and failures can occur in sensors and equipment, which can lead to unexpected shutdown of the power reactors. Such situations may compromise the safety and reliability of the SSCs and result in risk and economic losses that may amount to hundreds of thousands of Euros [1]. For example, in United State of America, the economic loss as a result of shutting down a NPP is approximately $1.25 million per day [2]. Thus, if unnecessary shutdown of the system as a result of faults and failures can be prevented, economic loss due to shutdown can be minimized. Therefore, it is of paramount importance to improve the situation awareness of SSCs in NPPs in order to ensure that their faults and failures are detected early, which can be achieved through on-line signal analysis techniques [3–6], and fault detection and diagnosis (FDD) methods [7–13]. There are several techniques of signal analysis, fault detection, and fault diagnostics, which can be classified into two main categories: model-based

[14–17] and data-driven [18–22] methods. The model-based approaches require understanding of the target system's physical structure in order to develop a mathematical model of system response for the purpose of FDD. Data-driven techniques, instead, use the historical data measured by the installed sensors and collected overtime during system' operation to develop an empirical model. In both cases, the developed model can, then, be applied to the target SSC for on-line signal analysis, monitoring, and FDD during operation, from which the condition of the SSC can be retrieved and sent to the human operator/maintenance engineer as alert or alarm, in case of any fault or failure has occurred in the SSC. Based on the status of the SSC, necessary operator action or maintenance intervention can be performed on the SSC to avoid undesired conditions during operation. Adopting these methods in NPP come with several benefits, including [23]:

- Provide system engineers and maintenance staff with necessary information to make informed, cost-effective operations, and maintenance decisions based on the actual condition of the system/equipment.

- Allow early mitigation or corrective actions.

- Reduce the likelihood of unplanned plant trips or power reductions.

- Reduce challenges to safety systems.

- Reduce equipment damage.

- Facilitate the implementation of condition-based predictive maintenance (PdM and CBM) practices.

- Provide significant financial savings, especially if outage duration is reduced.

The recent advancement in data analysis and computational efficiency are motivating the nuclear and other industries to apply CBM for allowing early mitigation, minimizing unplanned shutdown, increasing safety, and reducing maintenance costs. A simple CBM strategy is a scheme that monitors the target component via a fault detection system that continuously collects data from sensors installed on the target component [24], makes a detection decision based on the collected information, provides to operators the condition of the component (normal or abnormal), and triggers an alarm in case of abnormal conditions, which alert the decision makers, for example, stakeholders, operators, and maintenance engineers, for deciding whether or not an intervention on a maintenance action is required on the component.

Figure 1 illustrates an architecture of the fault detection system considered in this work, which is based on an empirical model for signal reconstruction. Typically, as shown in **Figure 1**, a fault detection system is a decision tool based on (i) the model that reconstructs the values of on-line signals expected in normal conditions; and (ii) the residual calculator that analyses the differences between the measured on-line signal values and the reconstructed values, whereby an alarm is triggered if the residuals are statistically deviated from the allowable range representative of normal conditions.

Several empirical models have been developed and used for signal reconstruction. Such techniques include kernel regression (KR)—a special and simple form of Gaussian process regression (GPR) (which has been adapted for signal reconstruction as an auto-associative kernel regression (AAKR) [20] in nuclear industry),
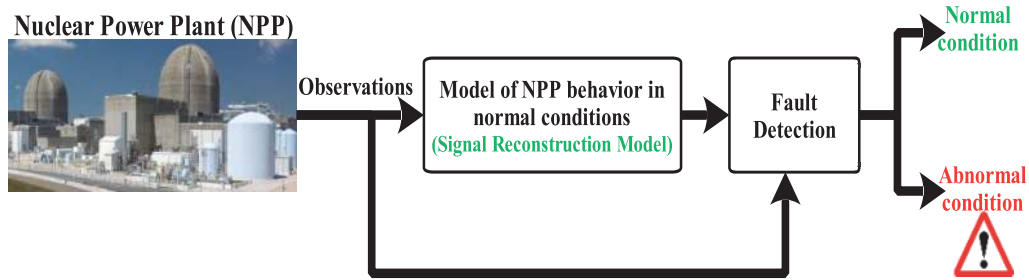
Figure 1.
A typical fault detection system.

auto-associative artificial neural networks (AANNs) [25–27], principal component analysis (PCA) [28–33], multivariate state estimation technique (MSET) [34, 35], Parzen estimation [36], support vector machines [37, 38], evolving clustering method [39], partial least squares (PLS) [40], and fuzzy logic systems [41–45]. However, most data-driven models are developed under steady state plant operation, whereas it is fundamental to have signal validation and monitoring during transient operation as well, considering the fact that most industrial systems' operations are time-varying. Transient operations are any non-steady state, time-varying conditions, which includes start-up, shutdown, and load-following modes of the system, whose time-series data are characterized by an explicit order dependency between observations—a time dimension.

In general, model-based approaches provide valid FDD techniques and are a powerful way to investigate FDD issues in highly dynamic and time-varying systems. However, the high performance of model-based FDD is often achieved at the cost of highly complex process modeling that requires sophisticated system design procedures [46]. Consequently, there is the need for low-complexity data-driven algorithms that could be used for time-varying analysis of the transient operation of the process system. In this respect, AAKR has proven superior to PCA [47] and is less computationally demanding than AANN. AAKR is typically trained to reconstruct the output of its own input under normal conditions. It has been successfully used in actual NPP steady-state operations for instrument channel calibration and condition monitoring [48]. It is a nonparametric technique for estimating a regression function. Unlike parametric models, AAKR relies on the data to determine the model structure.

However, some drawbacks of AAKR, such as spillover effects and robustness issues, can lead to missed alarms or delays in fault detection, and to a difficulty in correctly identifying the sensor variable responsible for a fault that is detected [49]. In order to address these drawbacks, a robust distance measure has been proposed, based on removing the largest elemental difference that contributes to the Euclidean distance metric so as to enable the model to correctly predict sensor values [50]. In [51], a modified AAKR has been proposed, based on a similarity measure between the observational data and historical data, with a pre-processing step that projects both the observed and historical data into a new space defined by a penalty vector.

Although those modifications have improved the AAKR performance, the underlining structure of the AAKR is still based on the traditional unilateral kernel regression and lacks temporal information, which makes its application inappropriate for signal analysis during transient operation because only the current query vector affects the model. Any previous information leading to the current query signals vector is completely ignored. Although this procedure is acceptable and even preferable for many applications, it is not acceptable for transient operations, in which the previous information directly affects the next data point [52, 53].

Recently, a weighted-distance based auto associative bilateral kernel regression (AABKR) for on-line monitoring during process transient operations has been proposed [54] and successfully applied to start-up transient data from an NPP [54, 55]. The AABKR captures both the spatial and temporal information in the data. The time dimension of these kinds of time-series data is, in fact, a structure that provides additional information. The AABKR systematically distributes the weights along the time dimension, using a weighted-distance algorithm that captures temporal dependences in the data [54, 55]. The weighted-distance algorithm uses a derivative-based comparator for the identification of a 'time position index' (the position of the nearest vector, within the historical data vectors, to the current on-line query observation) [54], which directly eliminates the use of on-line time input to the model.

However, when applied to data from steady-state, the performance of the AABKR in terms of correct fault diagnosis (i.e., the identification of the sensor variable responsible for the fault) is not satisfactory, as the fault, in most cases, is detected in both faulty and fault-free sensor signals [54, 55]. After thorough examination, it has been observed that [54]: (1) the AABKR suffers significantly from the spillover effect; (2) the effect is the result of the wrong identification of the 'time position index' by derivatives; and (3) the values of derivatives approximated from a typical steady-state process are, obviously, constants (and nearly zeroes) for most of the data points, particularly, when the process change in time is almost negligible, resulting in wrong identification of the 'time position index'.

It is worth noting that, a correct identification of time position index is crucial for the temporal weighted-distance algorithm that captures the temporal correlation in the data. The consequence of this effect is that, if a fault occurs, it might indeed be detected but, with an incorrect fault diagnosis of the variable responsible for that fault.

Motivated by these observations, we here propose a modified AABKR for efficient on-line monitoring, applicable not only in transient process operations but also in steady-state operations. We develop new algorithms, based on dynamic time warping (DTW), for the identification of temporal dependencies in the data [55]. To evaluate the performance of the proposed methods, we use both synthetic data from a numerical process and real-time data collected from a pressurized water reactor power plant.

2. Problem formulation

We consider a fault detection system designed to monitor the condition of a plant, as shown in **Figure 1**, and the sequence of time-varying observations ordered in time (a time series data). Time is the independent variable, and we assume it to be discrete; thus, time-varying data are a sequence of pairs $[(x_1, t_1); (x_2, t_2); \cdots; (x_M, t_M)]$ with $(t_1 < t_2 < \cdots < t_M)$, where each x_i is a data point in the feature space and t_i is the time at which x_i is observed. The data for more than one signal are sequences of time-varying data points, so long as their sampling rates $(t_i - t_{i-1} = \Delta t = \eta)$ are the same.

With this definition, we assume that:

    i. The sequences of data within an historical time-varying dataset, taken by the sensor in healthy condition, are measured and are available as a memory data matrix, $\mathbf{X} \in \mathbb{R}^{M \times p}$, whose elements, x_{ij}, are functions of the scalar parameter time, t, where $\mathbf{X}$ is a p-dimensional matrix of signals with M observation sequence vectors, and x_{ij} represents the ith observation of the jth signal.

ii. The sequences of data in **X** are large enough to be representative of the plant's normal operating condition.

iii. With r defined as a window length, the sequences of the real-time on-line measurement vectors $\mathbf{X}_q^* \in \mathbb{R}^{r \times p}$ can be collected, with $\mathbf{x}_{q_r}^* = \begin{bmatrix} x_{r,1}^* & x_{r,2}^* & \cdots & x_{r,p}^* \end{bmatrix}$ being the last vector, i.e., the current measurement vector at present time, t.

iv. The sequence of real-time observational matrix, $\mathbf{X}_q^*$, can be updated from current measurement, $\mathbf{x}_{q_r}^*$ backward to the size of the moving window, r, whenever another on-line query vector is available.

On the basis of above descriptions and by using **X**, the objective of the present work is to develop a signal reconstruction model reproducing the plant behavior in normal conditions. Such model receives in input the observed sequence of real-time measurement matrix, $\mathbf{X}_q^*$, whose rth vector is the present measurement, $\mathbf{x}_{q_r}^*$, containing the actual observations of p signals monitored at the present time, t, and produces in output, $\hat{\mathbf{x}}_{q_r}$, the reconstruction values of the signals expected in normal condition. Based on this, the actual plant condition at the present time, t can, then, be determined by the analysis of residuals: a fault is detected if the variations between the observations and the reconstructions are large enough in, at least, one of the signals in comparison to predefined thresholds.

3. Auto associative bilateral kernel regression

3.1 Mathematical framework of AABKR

In AABKR, the validation of the signals at present time, t is based on the analysis of the on-line query pattern, $\mathbf{X}_q^*$, to reconstruct the current vector, $\mathbf{x}_{q_r}^*$, at time, t. The basic idea behind AABKR is to capture both the spatial and temporal correlations in the time-series data (see **Figure 2**), for effective signal reconstruction in the transient operation of industrial systems. The historical memory matrix **X** is reorganized into $\mathcal{A} \in \mathbb{R}^{N \times r \times p}$ sequences of array of N matrices of length r, containing the measurement vectors having $(r-1)$ overlapping between the two
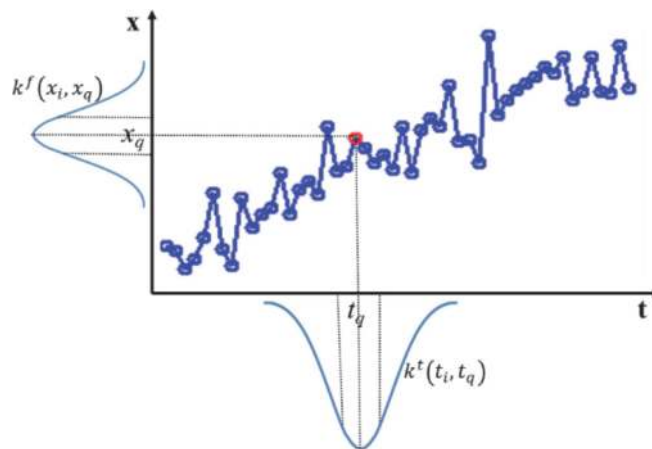
Figure 2.
Graphical representation of the bilateral directions for a time-series.

consecutive time windows, with the r sequence vectors in each matrix of $\mathcal{A}$ array represented as $\mathcal{A}_r^k \in \mathbb{R}^{r \times p}$, where $k = 1, 2, \ldots, N$, and $(N = M - r + 1)$.

The AABKR is expressed in such a way that each neighboring value is weighted on its proximity in space and time. Hence, the mathematical framework of the AABKR is summarized as follows [54, 55]:

1) Feature distance calculation

The feature distance between $\mathbf{x}_{q_r}^*$ and each of the historical memory vectors in matrix $\mathbf{X}$, is computed using the Manhattan distance (L^1-Norm) as

$$d_i\left(\mathbf{X}_i, \mathbf{x}_{q_r}^*\right) = \left\|\mathbf{X}_i - \mathbf{x}_{q_r}^*\right\|_1 = \sum_{j=1}^{p}\left|x_{i,j} - x_{r,j}^*\right| \tag{1}$$

and produces the distance vector, $\mathbf{d} \in \mathbb{R}^{M \times 1}$.

2) Feature kernel quantification

The feature distances calculated above are used to determine the feature weights by evaluating the Gaussian kernel

$$k_i^f = exp\left(\frac{-d_i^2}{2h_f^2}\right) \tag{2}$$

where h_f is a kernel bandwidth for feature preservation, which controls how much the nearby memory feature vectors are weighted. This leads to the $\mathbf{k}^f \in \mathbb{R}^{M \times 1}$ vector. The superscript/subscript f indicates the feature components.

3) Time position index identification

Here, the time position index, that is, the temporal location of the nearest vector, within the memory vector, to the query vector observation, is determined using a derivative-based comparator. This provides the input to the weighted-distance algorithm in *Step 4*. Instead of directly using the derivative in the prediction to capture the temporal correlation of the data, which might not be a good choice because of process measurement noise, the derivatives are used as a *comparator* to determine the time position index within the memory vectors to which the query data vector is nearest. The derivative-based comparator is described as follows.

The backward-difference first-order derivative approximation of the current historical measurement vector in each matrix $\mathcal{A}_r^k$ based on r data points accuracy with respect to t is the element-by-element derivative:

$$\frac{\partial \mathcal{A}_r^k}{\partial t} = \left[\begin{array}{cccc} \frac{\partial x_{r,1}^k}{\partial t} & \frac{\partial x_{r,2}^k}{\partial t} & \cdots & \frac{\partial x_{r,p}^k}{\partial t} \end{array} \right] \tag{3}$$

whereas, that of the current query vector, $\mathbf{x}_{q_r}^*$ in matrix $\mathbf{X}_q^*$ is the element-by-element derivative:

$$\frac{\partial \mathbf{x}_{q_r}^*}{\partial t} = \left[\begin{array}{cccc} \frac{\partial x_{r,1}^*}{\partial t} & \frac{\partial x_{r,2}^*}{\partial t} & \cdots & \frac{\partial x_{r,p}^*}{\partial t} \end{array} \right]. \tag{4}$$

The first-order derivatives in Eqs. (3) and (4) have been approximated from the data using finite-difference derivative approximation. The backward finite difference derivative approximation is chosen to implement real-time on-line monitoring. The model needs r successive data points to evaluate the derivative of the current data point from the current measurement backward to the size of the moving window r at every sampling time, using backward finite-difference derivative approximation.

From Eqs. (3) and (4), the distance between the derivative of a query vector, $\mathbf{x}_{q_r}^*$ and each kth derivative vector of $\mathcal{A}_r^k$ can be calculated by Eq. (5) using the Manhattan distance (L^1 norm):

$$\Delta_k \left(\frac{\partial \mathcal{A}_r^k}{\partial t}, \frac{\partial \mathbf{x}_{q_r}^*}{\partial t} \right) = \left\| \frac{\partial \mathcal{A}_r^k}{\partial t} - \frac{\partial \mathbf{x}_{q_r}^*}{\partial t} \right\|_1 = \sum_{j=1}^{p} \left| \frac{\partial x_{r,j}^k}{\partial t} - \frac{\partial x_{r,j}^*}{\partial t} \right| \tag{5}$$

This gives the derivative distance vector, $\Delta \in \mathbb{R}^{N \times 1}$.

Then, using the minimum value in Δ, the index $i = \varepsilon$, which indicates the location of the vector in the memory data, $\mathbf{X}$, to which the current query vector $\mathbf{x}_{q_r}$ is closest, can be obtained. The time position index is, therefore, the index at which the Manhattan distance between the derivative of the current query vector and those of the rth vectors in each of the $\mathcal{A}_r^k$ is minimized plus the overlapping length between the two consecutive time windows, which is determined as:

$$\varepsilon = \left(\arg \min_{k \in [1:N]} (\Delta_k) \right) + r - 1 \tag{6}$$

4) Temporal weighted-distance algorithm

The temporal weighted-distance algorithm captures the temporal correlations in the data. It calculates the measures that capture the temporal variations in the data. The distance, δ, accounts for the time at which the query vector is observed. This algorithm calculates the temporal correlation of a query input with the memory data, without using the query time input (t_q), and eliminates the direct use of t_q, which becomes indefinite when applied to on-line monitoring. In this way, the effect of the query time input is confined within the time duration of the historical memory data. The distance is calculated based on the assumption that the time-varying historical data collected in building the model were sampled at a constant time interval, η.

Based on the time position index determined in *Step 3*, the temporal weighted-distance algorithm that captures the temporal correlation is formulated as

$$\delta_i = \begin{cases} \delta_\varepsilon, & i = \varepsilon \\ \delta_\varepsilon + (i - \varepsilon).\eta, & i > \varepsilon \& \varepsilon \neq M; \quad i \in [1, M] \\ \delta_\varepsilon + (\varepsilon - i).\eta, & i < \varepsilon \& \varepsilon \neq 1 \end{cases} \tag{7}$$

giving the weighted-distance vector $\delta \in \mathbb{R}^{M \times 1}$:

$$\delta = \begin{bmatrix} \delta_1 & \cdots & \delta_{\varepsilon-2} & \delta_{\varepsilon-1} & \delta_\varepsilon & \delta_{\varepsilon+1} & \delta_{\varepsilon+2} & \cdots & \delta_M \end{bmatrix}^T \tag{8}$$

Once the values of δ_ε and η are known, the other values in Eq. (8) can be determined progressively using Eq. (7). The second and third equations in Eq. (7)

follow arithmetic progression (AP): the first term and the common difference of the two progressions are δ_ε and η, respectively. A zero value for the first term of the two progressions, $\delta_\varepsilon = 0$, has been recommended [55] because the distance of the nearest vector in the memory data to the query vector is close to zero. Whereas the value of the common difference can be arbitrarily selected or taken to be the time interval, η. The other distance values to the right and left of δ_ε in Eq. (8) can be progressively calculated using the second and third equations in Eq. (7), respectively. See Appendix C of [55] for the proof of this algorithm (Eq. (7)).

5) Temporal kernel quantification

Having determined the weighted-distance, the kernel weight can be calculated using the Gaussian kernel function:

$$k_i^t = exp\left(\frac{-\delta_i^2}{2h_t^2}\right) \tag{9}$$

where the k_i^t is the ith kernel weight calculated from the temporal weighted-distance, δ_i; the superscript/subscript t indicates the temporal components; h_t is the bandwidth for the time-domain preservation, which can also serve as noise rejection and controls how much the nearby times in the memory vectors are weighted. This gives the vector $\mathbf{k}^t \in \mathbb{R}^{(M \times 1)}$.

6) Adaptive bilateral kernel evaluation

Depending on the magnitude of a fault that occurs in a process, the result of the direct multiplication of the two kernels at $i = \varepsilon$ could be zero, which would result in an inaccurate model prediction because the model prediction tends to follow the fault occurrence, so the fault would not be detected. To resolve such issue and achieve robust model signal reconstruction, and to reduce the impact of spillover onto other signals when one or more signals is in fault condition, Eq. (10) is formulated [54] adaptively for the combined kernels of Eqs. (2) and (9) as:

$$k_i^{ab} = \begin{cases} k_i^f * k_i^t, & 1 \leq i \leq M \& i \neq \varepsilon \\ \dfrac{\left(k_i^f + k_i^t\right)}{2}, & i = \varepsilon \end{cases} \quad ; \quad i\epsilon[1, M] \tag{10}$$

resulting in the adaptive bilateral kernel vector $\mathbf{k}^b \in \mathbb{R}^{M \times 1}$.

This reduces the effect of the dominance of one feature distance value over another when a fault occurs. The adaptive nature of Eq. (10) is to dynamically compensate for faulty sensor inputs to the bilateral kernel evaluation and always ensure that a larger weight is assigned to the closest vector within the memory data to the query vector, so as to guarantee an approximate signal reconstruction. This reduces the effect of the degeneration of the feature kernel when a fault of high magnitude has occurred.

7) Output estimation

Finally, the adaptive bilateral kernel weights are combined with the memory data vectors to give the predictions as:

$$\hat{x}_{r,j}^* = \frac{\sum_{i=r}^M k_i^{ab} \cdot x_{i,j}}{\sum_{i=r}^M k_i^{ab}} \tag{11}$$

If a normalized adaptive bilateral kernel vector, $\mathbf{w} \in \mathbb{R}^{(M-r+1)\times 1}$ is defined:

$$w_i = \frac{k_i^{ab}}{\sum_{i=r}^M k_i^{ab}}, \tag{12}$$

then, Eq. (11) can be rewritten in matrix form to predict all the signals of the query vector simultaneously as:

$$\hat{\mathbf{x}}_{q_r}^* = \mathbf{w}^T \mathbf{X} \tag{13}$$

where $\mathbf{X} \in \mathbb{R}^{(M-r+1)\times p}$.

8) Fault detection

After training of the model, the root mean square error (RMSE) on the predictions of the fault-free validation dataset can be calculated using residuals $\left(\mathbf{e}_{q_r} = \mathbf{x}_{q_r} - \hat{\mathbf{x}}_{q_r}^* \right)$ between the actual values and the predicted values of the validation dataset, and can be used to set the threshold limit for fault detection in each signal as follows:

$$T_j^D = 3 * RMSE_j. \tag{14}$$

Because the residuals can be assumed to be Gaussian and randomly distributed with a mean of zero and variance of $RMSE_j{}^2$, a constant value equal to 3 has been selected in [54] to minimize the false alarm rate and ensure that a fault is detected when the residuals exceed the threshold.

3.2 Analysis of the limitation of the AABKR

In this section, we analyze the limitation of the AABKR described in Section 3.1, in terms of signal reconstruction from faulty sensor signals. The major limitation can be understood from the description presented as follows.

We observed that, in an extreme, limit or worst case scenario, where the fault deviation intensity in a faulty sensor signal is significant, the feature distance vector degenerates and tends to zero (i.e., $\mathbf{k}^f \approx \mathbf{0}$), so that the signal reconstructed by the AABKR model is bound to be:

$$\hat{x}_{q,j}^* = x_{\varepsilon,j} \tag{15}$$

This observation can be understood better by the following analysis.

Recall the reconstructed output from a weighted average of Eq. (11), re-written as:

$$\hat{x}_{q,j}^* = \frac{\sum_{i=1}^M \left(k_i^f \circledast k_i^t \right) x_{i,j}}{\sum_{i=1}^M \left(k_i^f \circledast k_i^t \right)} \tag{16}$$

where, $k_i^f \circledast k_i^t = k_i^{ab}$ is the adaptive bilateral kernel evaluated at $\mathbf{x}_i$. The symbol $\circledast$ represents the bilateral kernel combination operator that combines the feature and temporal kernels, given by Eq. (10).

Applying the properties of limit to Eq. (16), we have:

$$\lim_{\mathbf{k}^f \to \mathbf{0}} \left(\hat{x}_{q,j}^* \right) = \lim_{\mathbf{k}^f \to \mathbf{0}} \left(\frac{\sum_{i=1}^M \left(k_i^f \circledast k_i^t \right) x_{i,j}}{\sum_{i=1}^M \left(k_i^f \circledast k_i^t \right)} \right) \tag{17}$$

Equation (17) can be re-written as:

$$\lim_{\mathbf{k}^f \to \mathbf{0}} \left(\hat{x}_{q,j}^* \right) = \frac{\lim_{\mathbf{k}^f \to \mathbf{0}} \left(\sum_{i=1}^M \left(k_i^f \circledast k_i^t \right) x_{i,j} \right)}{\lim_{\mathbf{k}^f \to \mathbf{0}} \left(\sum_{i=1}^M \left(k_i^f \circledast k_i^t \right) \right)} \tag{18}$$

provided that:

$$\lim_{\mathbf{k}^f \to \mathbf{0}} \left(\sum_{i=1}^M \left(k_i^f \circledast k_i^t \right) \right) \neq 0 \tag{19}$$

Note that, judging from the Eq. (10), Eqs. (18) and (19) hold valid. Hence, the limit in Eq. (18) can be simplified as:

$$\lim_{\mathbf{k}^f \to \mathbf{0}} \left(\hat{x}_{q,j}^* \right) = \frac{\lim_{k_1^f \to 0} \left(\left(k_1^f \circledast k_1^t \right) x_{1,j} \right) + \lim_{k_2^f \to 0} \left(\left(k_2^f \circledast k_2^t \right) x_{2,j} \right) + \cdots + \lim_{k_M^f \to 0} \left(\left(k_M^f \circledast k_M^t \right) x_{M,j} \right)}{\lim_{k_1^f \to 0} \left(k_1^f \circledast k_1^t \right) + \lim_{k_2^f \to 0} \left(k_2^f \circledast k_2^t \right) + \cdots + \lim_{k_M^f \to 0} \left(k_M^f \circledast k_M^t \right)} \tag{20}$$

But, from the adaptive bilateral combination of Eq. (10):

$$\lim_{k_i^f \to 0; \, i \neq \varepsilon \, \& \, 1 \leq i \leq M;} \left(k_i^f \circledast k_i^t \right) = 0 \tag{21}$$

and

$$\lim_{k_i^f \to 0, \, i = \varepsilon} \left(k_i^f \circledast k_i^t \right) = 0.5 \tag{22}$$

Hence Eq. (20) becomes

$$\lim_{\mathbf{k}^f \to \mathbf{0}} \left(\hat{x}_{q,j}^* \right) = \frac{\lim_{k_i^f \to 0, \, i = \varepsilon} \left(\left(k_i^f \circledast k_i^t \right) x_{i,j} \right)}{\lim_{k_i^f \to 0, \, i = \varepsilon} \left(k_i^f \circledast k_i^t \right)} = x_{\varepsilon,j} \left(\frac{0.5}{0.5} \right) \tag{23}$$

Thus,

$$\lim_{\mathbf{k}^f \to \mathbf{0}} \left(\hat{x}_{q,j}^* \right) = x_{\varepsilon,j} \tag{24}$$

This implies that, in a limit case of faulty sensor query signal, $x_{q,j}^*$, the reconstructed query signal, $\hat{x}_{q,j}^*$ is equal to the historical (memory) data point, $x_{\varepsilon,j}$ located at the identified time position index, ε.

From the above analysis, it is clear that the fault detection capability of the AABKR largely depends on the accuracy of the time position index identification algorithm. If the time position index has been identified correctly, the fault can be detected even though the fault deviation intensity is large and tends to infinity. However, if the time position index has been identified wrongly, the fault might or might not be detected depending on the identified index and the intensity of the fault. The consequence of this situation, even if the fault has been detected, is that the sensor signal responsible for the fault might not be diagnosed correctly which might lead to wrong diagnosis of the system under consideration. In this regards, a more robust approach for time position index identification is proposed and discussed in the next section.

4. Modified AABKR

In this section, the modified AABKR is presented. The framework of the proposed method is depicted in **Figure 3**. It comprises the steps of calculating the feature distance that captures the spatial variation in the data; calculating the feature kernel weights based on the calculated feature distance; identifying the time position index using DTW technique; computing the temporal weighted-distance that captures the temporal variation and dependencies in the data, based on the time position index; calculating the temporal kernel weights, based on the calculated weighted-distance; and evaluating the adaptive bilateral kernel that computes the combined kernels and dynamically compensates for faulty sensor inputs to the bilateral kernel evaluation, and then, makes the prediction using a weighted average for the purpose of fault detection. Only the modifications to the original AABKR are discussed in this section. The basics of the DTW technique is first presented in Section 4.1. Then, the developed methods based on the DTW are discussed in Section 4.2. Finally, a demonstration of the developed identification methods is showcased in Section 4.3.

4.1 Dynamic time warping

Dynamic time warping (DTW) is a technique for finding an optimal alignment between two time-dependent sequences. This technique uses a dynamic programming approach to align the time-series data [56]. Suppose we have two time-series sequences of values taken from the feature space, $\mathbf{x} = [\begin{array}{cccc} x_1 & x_2 & \cdots & x_L \end{array}]$ and $\mathbf{y} = [\begin{array}{cccc} y_1 & y_2 & \cdots & y_M \end{array}]$, of length L and M, respectively. To align these sequences using DTW, an $L \times M$ matrix, $\mathbf{D}$, is first established, where the element $d_{l,m}$ of $\mathbf{D}$ is a local distance measured between the points x_l and x_m, usually called *cost function* since the DTW technique is based on the dynamic programming algorithm. Thus, the task of optimal alignment of these sequences is the arrangement of all sequence
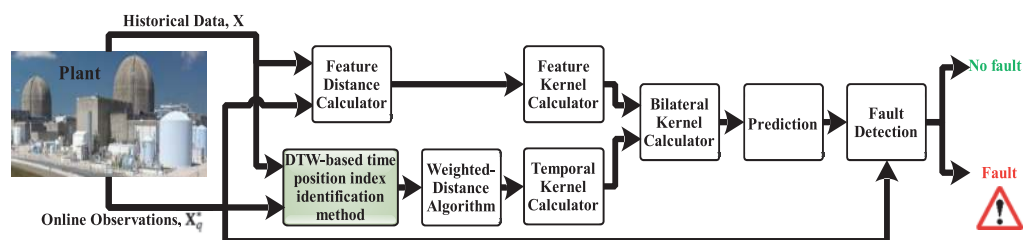
Figure 3.
Framework of the modified AABKR for fault detection system.

points for minimizing the *cost function*. Once the local cost matrix has been built, the algorithm finds the alignment path which runs through the low-cost area—valley on the matrix. The alignment path, usually called the warping path, is a sequence of points, $\mathbf{w} = [\,w_1 \quad w_2 \quad \cdots \quad w_{\mathcal{P}}\,]$ and defines the correspondence of an element x_l to y_m with $w_p = (l_p, m_p) \in [1:L] \times [1:M]$ for $p \in [1:\mathcal{P}]$ satisfying the following three criteria [57, 58]:

1. Boundary condition: $w_1 = (1,1)$ and $w_{\mathcal{P}} = (L, M)$.

2. Monotonicity condition: $l_1 \leq l_2 \leq \cdots \leq l_{\mathcal{P}}$ and $m_1 \leq m_2 \leq \cdots \leq m_{\mathcal{P}}$.

3. Step size condition: $w_{p+1} - w_p \in \{(1,0), (0,1), (1,1)\}$ for $p \in [1:L-1]$.

The total cost of a warping path $\mathbf{w}$ between $\mathbf{x}$ and $\mathbf{y}$ with respect to the local cost matrix (which represents all pairwise distances) is:

$$d_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = \sum_{p=1}^{\mathcal{P}} d(w_p) = \sum_{p=1}^{\mathcal{P}} d\left(x_{l_p}, y_{m_p}\right) \tag{25}$$

Moreover, an optimal warping path between $\mathbf{x}$ and $\mathbf{y}$ is a warping path, $\mathbf{w}^*$, that has minimal total cost among all possible warping paths. The DTW distance, DTW $(\mathbf{x}, \mathbf{y})$ between $\mathbf{x}$ and $\mathbf{y}$ is, then, defined as the total cost of $\mathbf{w}^*$:

$$\mathrm{DTW}(\boldsymbol{x}, \boldsymbol{y}) = d_{\boldsymbol{w}^*}(\boldsymbol{x}, \boldsymbol{y}) = \min\left\{d_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{w} \in \boldsymbol{W}^{L \times M}\right\} \tag{26}$$

where $\mathbf{W}^{L \times M}$ is the set of all possible warping paths.

To determine the optimal warping path, one could test every possible warping path between $\mathbf{x}$ and $\mathbf{y}$, which would however very be computationally intensive. Therefore, the optimal warping path can be found using dynamic programming by building an accumulated cost matrix called global cost matrix, $\mathbf{G}$, which is defined by the following recursion [57]:

First row:

$$\mathbf{G}(1, m) = \sum_{k=1}^{m} d(x_1, y_k), \quad m \in [1, M] \tag{27}$$

First column:

$$\mathbf{G}(l, 1) = \sum_{k=1}^{m} d(x_k, y_1), \quad l \in [1, L] \tag{28}$$

All other elements:

$$\mathbf{G}(l, m) = d(x_l, y_m) + \min \left\{ \begin{array}{l} \mathbf{G}(l-1, m-1), \\ \mathbf{G}(l-1, m), \\ \mathbf{G}(l, m-1) \end{array} \right\}, l \in [1, L] \,\&\, m \in [1, M] \tag{29}$$

This means that, the accumulated global distance is the sum of the distance between the current elements and the minimum of the accumulated global

distances of the neighboring elements. The required time for building matrix **G** is $O(LM)$.

Having determined the accumulated cost matrix **G**, obviously, the DTW distance between **x** and **y** is simply given by Eq. (30):

$$DTW(\mathbf{x}, \mathbf{y}) = \mathbf{G}(L, M). \qquad (30)$$

Once the accumulated cost matrix has been built, the optimal warping path could be found by backtracking from the point $w_{\mathcal{P}} = (L, M)$ to the $w_1 = (1, 1)$, using a greedy strategy.

4.2 DTW-based time position index identification approaches

One of the significant steps that requires modification in AABR is the identification of the time position index. The goal here is that the identification of the time position index should be solely based on the feature data and must be freed from the use of the derivatives, in order to improve the monitoring performance of the AABKR during steady-state operations. To achieve this goal, we developed two approaches based on the DTW algorithm described in Section 4.1 for the identification of the time position index. The two approaches are described in the following subsections.

4.2.1 First approach: based on the generated subsequences of the memory data

For simplicity of description, we assume $p = 1$ (single signal) and present the description of the algorithm by referring to **Figure 4**, where $\mathbf{X} \in \mathbb{R}^{M \times p}$ and $\mathbf{X}_q^* \in \mathbb{R}^{r \times p}$ are memory data and query data, respectively. We also assume that the memory data, **X** has already been reorganized into an array containing N matrices each of length r with $(r - 1)$ overlapping between them, as described in Section 3.1 where $(N = M - r + 1)$. By using DTW, the goal is to find a subsequence $\mathcal{A}_r^k[a^* : \varepsilon] \in \mathbb{R}^{r \times p} = [x_{a^*}, x_{a^*+1}, \cdots x_{\varepsilon}]$, with $1 \leq a^* \leq \varepsilon \leq M$, that minimizes the DTW distance to $\mathbf{X}_q^*$ over all N matrices generated from **X**. Note that, $\mathcal{A}_r^k$ is a kth subsequence in the generated array, where $k = 1, 2, \ldots, N$. Thus, the DTW distance between $\mathbf{X}_q^*$ and each of the kth matrix, $\mathcal{A}_r^k$ in $\mathcal{A}$ can be determined by:

$$DTW\left(\mathbf{X}_q^*, \mathcal{A}_r^k\right) = \mathbf{G}(r, r) \qquad (31)$$

which can be calculated from the local cost function matrix, **D**, using Eqs. (27)–(29) of the DTW algorithm described in Section 4.1. Each element of **D**
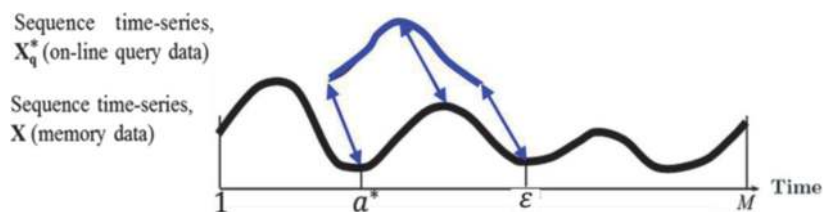
Figure 4.
Alignment between two time-dependent data: Sequences $\mathbf{X}_q$ and $\mathbf{X}$.

is a local distance measured between the points in $\mathbf{X}_q^*$ and $\mathcal{A}_r^k$, which can be calculated by:

$$d[l,m] = d\left(\mathbf{x}_l^*, \mathbf{x}_m\right) = \left\|\mathbf{x}_l^* - \mathbf{x}_m\right\|_1 \tag{32}$$

where $l = 1, 2, \ldots, r$, is the row index of the query data, $\mathbf{X}_q^*$, and $m = 1, 2, \ldots, r$, is the row index of the kth subsequence, $\mathcal{A}_r^k$, of the memory data.

Then, the time position index can be determined as:

$$\varepsilon = \left(\arg\min_{k \in [1:N]}\left\{\mathrm{DTW}\left(\mathbf{X}_q^*, \mathcal{A}_r^k\right)\right\}\right) + r - 1. \tag{33}$$

This index can, then, be used in the weighted-distance algorithm for the calculation of the temporal weighted-distance. This approach is summarized in Algorithm A.2.1 of Appendix A.2.

4.2.2 Second approach: based on the entire memory data

In this approach, instead of calculating the DTW distances between the query input data and each of the subsequence data generated from the memory data, the mapping between the query input data and the memory data can be determined directly, from which the time position index can be obtained. Thus, the generation of the array from the memory data is not required in this case. This approach is described as follows.

First, calculate the local cost function matrix, $\mathbf{D}$ between the query data, $\mathbf{X}_q^*$, and the memory data, $\mathbf{X}$. Having calculated $\mathbf{D}$, the calculation of $\mathbf{G}$ from $\mathbf{D}$ using dynamic programming [57] is a bit modified through the following recursion:

First row:

$$\mathbf{G}(1,i) = d\left(\mathbf{x}_1^*, \mathbf{x}_i\right), \quad i \in [1,M] \tag{34}$$

First column:

$$\mathbf{G}(l,1) = \sum_{s=1}^{i} d\left(\mathbf{x}_s^*, \mathbf{x}_1\right), \quad l \in [1,r] \tag{35}$$

All other elements:

$$\mathbf{G}(l,i) = d\left(\mathbf{x}_l^*, \mathbf{x}_i\right) + \min \left\{\begin{array}{l} \mathbf{G}(l-1, i-1), \\ \mathbf{G}(l-1, i), \\ \mathbf{G}(l, i-1) \end{array}\right\}, l \in [1,r] \& i \in [1,M] \tag{36}$$

leading to an accumulated matrix, $\mathbf{G} \in \mathbb{R}^{r \times M}$.

Finally, the time position index is obtained as follows, using the last row of $\mathbf{G}$:

$$\varepsilon = \arg\min_{i \in [r:M]} \left(\mathbf{G}[r,i]\right) \tag{37}$$

Note that the calculation of $\mathbf{G}$ is the same as earlier discussed, except that, its first row is taken equal to the first row of $\mathbf{D}$ without accumulating [57], as shown in Eq. (34). This is because, our goal is to determine the time position index using the

last row of **G** and to minimize the impact of the fault (if it occurs) during the determination of the index. The obtained time position index in Eq. (37) can, then, be used in the weighted-distance algorithm for the calculation of the temporal weighted distance. This approach is summarized in Algorithm A.2.2 of Appendix A.2.

4.3 Demonstration case

To demonstrate the two approaches of time position index identification, we consider a typical univariate time-dependent process:

$$x(t) = -\frac{1}{200,000}t^2 + 8 + g(t). \tag{38}$$

where $g(t)$ is assumed to be an independent and normally distributed Gaussian noise at present time, t with mean equal to zero and standard deviation, 0.08. For simplicity of demonstration, a memory dataset, **X**, consisting of 50 samples (with t = 1–50 s at constant time intervals of $\eta = 1s$), has been generated from the above process. By setting the window size, r = 10, a query input of length r is, then, generated. The actual location of the query input within the memory data is from t = 36–45 s, where the time position index of the current data point is at time t = 45 s. The goal is to automatically locate this index using the proposed methods. The generated memory data is plotted as shown in **Figure 5**, where the location of the query input is indicated in blue color.

With respect to the first approach, the global matrices, **G**s, between the query observations and each of the subsequences of the memory data, are first computed and visualized in **Figure 6** for the case of fault-free data. Next, a fault is added to the data point at present time t (t = 45 s) within the query data pattern and, then, the global matrices are recomputed as depicted in **Figure 7**. The DTW distances from global matrices are presented in **Figure 8** for both cases of fault and no fault. It can be seen that, the index at which the present data point is closest to has been identified in both fault-free and faulty cases. From **Figure 8** and by using Eq. (33), the time position index is $\varepsilon = 36 + (10 - 1) = 45$.

With respect to the second approach, the global matrix between the query observation and memory data is first computed and visualized in **Figure 9** for the fault-free case. Next, a fault is added to the data point at present time t (t = 45 s) within the query data pattern and, then, the global matrix is recomputed as shown in **Figure 10**. Finally, the last row of the global matrix is used to determine the index
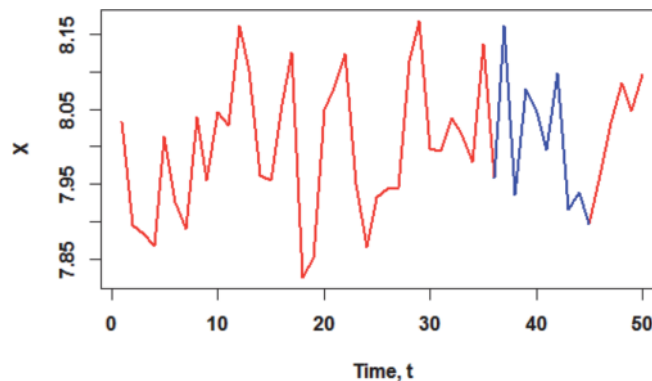
Figure 5.
Memory data (in red) and query input (in blue, located at t = 36–45 s).
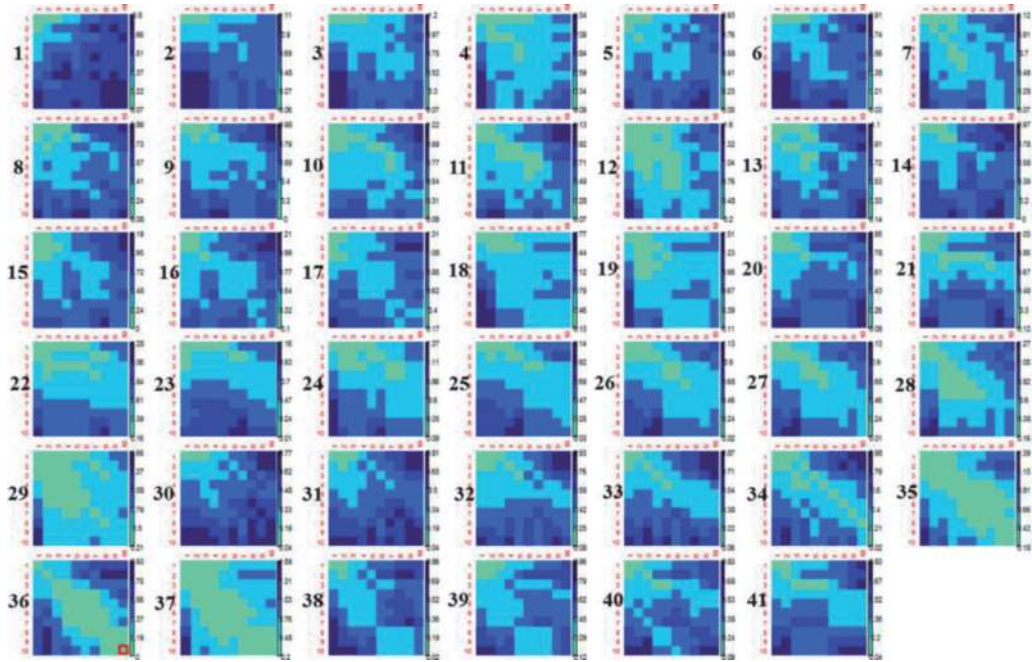
Figure 6.
Global matrices, $\mathbf{G}$, between $\mathbf{X}^*_q$ and subsequences of $\mathbf{X}$ (fault-free case).
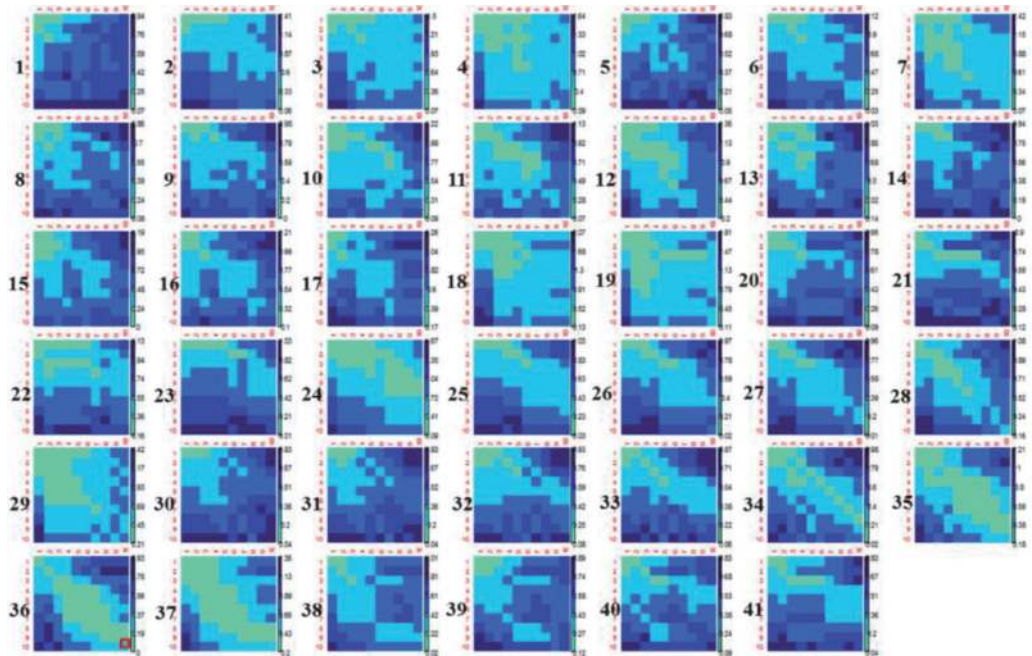
Figure 7.
*Global matrices, $\mathbf{G}$, between $\mathbf{X}^*_q$ and subsequences of $\mathbf{X}$ (fault case).*

in both cases of fault and no fault, using Eq. (37). The locations identified in both cases ($\varepsilon = 45$) are marked in red square box as shown in **Figures 9** and **10**.

We observe that if the fault deviation intensity increases, the identification accuracy of the second approach decreases and the time position index would not be identified correctly, whereas, the first approach would still identify the time position index correctly but with high computational demand. That is, the second approach is less computationally demanding than the first approach but it is less accurate.
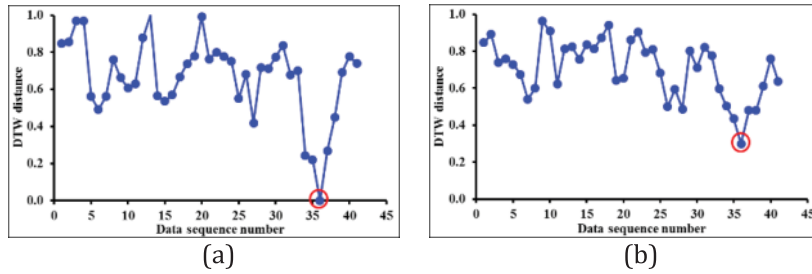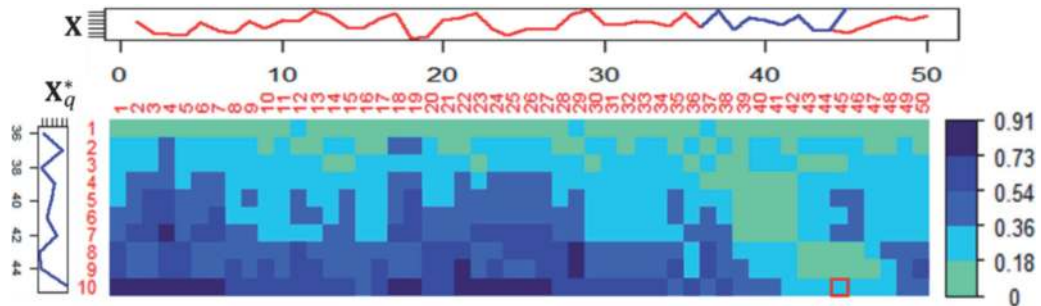
Figure 8.
DTW distances between X^*_q and subsequences of X. (a) Fault-free case and (b) faulty case.
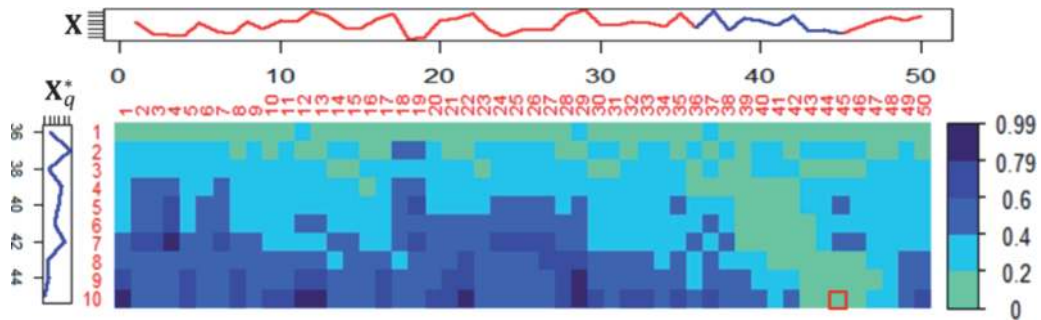
Figure 9.
Global matrix, G, between X^*_q and memory data, X (fault-free case).

Figure 10.
Global matrix, G, between X^*_q and memory data, X (fault case).

5. Applications

In this section, a typical steady-state numerical example taken from a literature is first used to evaluate the fault detection capability of the proposed signal reconstruction methods in steady-state operation and, then, applied to the transient start-up operation of a nuclear power plant.

5.1 Validation on the steady-state process

A typical steady-state numerical example [54], mimicking a typical industrial system, is considered to evaluate the performance of the proposed method in steady-state operation. The model of the process is:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} -0.2310 & -0.0816 & -0.2662 \\ -0.3241 & 0.7055 & -0.2158 \\ -0.2170 & -0.3056 & -0.5207 \\ -0.4089 & -0.3442 & -0.4501 \\ -0.6408 & 0.3102 & 0.2372 \\ -0.4655 & -0.4330 & 0.5938 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} + noise \qquad (39)$$

where t_1, t_2, and t_3 are zero-mean random variables with standard deviations of 1, 0.8, and 0.6, respectively. The noise included in the process is zero-mean with a standard deviation of 0.2, and is normally distributed. To build the model, 1000 samples are generated using such process. The number of simulated faults is 2000, with the data samples generated according to the model above and the fault magnitude being a random number uniformly distributed between 0 and 5. The signal variable in fault is also random uniformly sampled among the six possible variables, as simulated in [59].

Although this application represents a typical steady-state operation, we assume a set of sequential time-series data, assigning a constant time interval of $\eta = 1s$ with $r = 3$. To measure the performance of the propose methods, we employ different measures of the performance metrics (e.g., missed alarm rate (MAR), missed and false alarms rate (M&FAR), true and false alarms rate (T&FAR), true alarm rate (TAR), and fault detection rate (FDR)) as proposed in [54] and briefly defined in Appendix A.3. The purpose of this application is to verify the performance of the proposed method in monitoring during steady-state operation and to compare the results with those of AAKR (see Appendix A.1) and AABKR.

Table 1 and **Figure 11** show the alarm rates of AAKR, AABKR, and the modified AABKR computed from the prediction of the simulated faults. It is interesting to note that although the MAR of AAKR is a bit higher than that of AABKR, the performance of the two models does not differ significantly and both models have suffered from the spillover effects (i.e., the effect that a faulty signal has on the predictions of the fault-free signals) as evident from the values of T&FAR (i.e., the detection of faults in both faulty signal and at least one fault-free signal). Conversely, the performance of the modified AABKR is better than those of the other two methods in terms of TAR (i.e., the detection of fault only in a signal that actually has the fault, without false alarm in other fault-free signals) and T&FAR; hence, the modified AABKR is more resistant to spillover and more robust than both AAKR and AABKR. It can be observed that even though the TAR value of the modified AABKR is larger than those of the other two models, FDR values of the three methods did not differ significantly because of the larger values of T&FAR for AAKR and AABKR (53.4 and 61.8%, respectively). Therefore, it is important to further examine the rate of correct fault diagnosis of the three methods using absolute residual values of the faults successfully detected, which produced the FDR values. **Figure 12** shows the rate of correct fault diagnosis of the three methods. We

Model	MAR	M&FAR	T&FAR	TAR	FDR
AAKR	19.20	1.60	53.40	25.80	80.80
AABKR	5.65	8.26	61.82	24.27	94.35
Modified AABKR	12.46	6.81	28.18	**52.55**	87.54

Table 1.
Alarm rates (%) of validation on the numerical steady-state process.
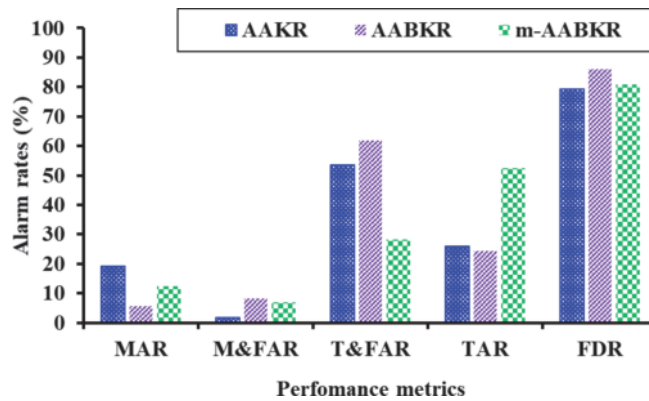
Figure 11.
*Alarm rates in a steady-state numerical process.*
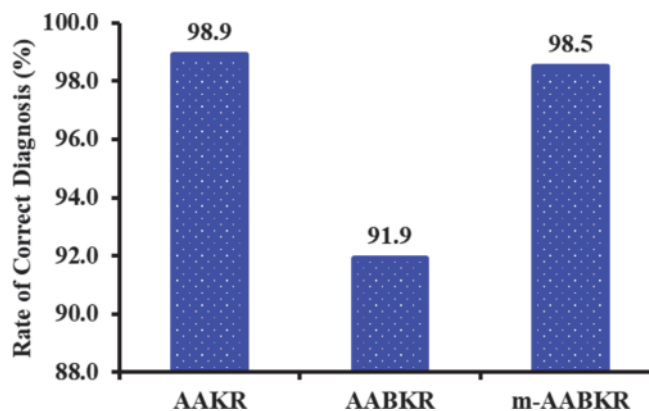
Figure 12.
Rate of correct fault diagnosis in a steady-state numerical process.

observe that the performance of the modified AABKR is comparable to that of AAKR, and the method can, thus, also be used effectively for signal validation during steady-state process operation.

5.2 Start-up transient operations in nuclear power plants

The real-time nuclear simulator data used in [54] is taken here to test the applicability of the proposed methods in transient operations. The data is collected from the simulator without any faults during heating from the cool-down mode (start-up operation). The simulator was designed to reproduce the behavior of a three loop pressurized water reactor (PWR) and to carry out various operational modes, such as start-up, preoperational tests, preheating, hot start-up, cold shut-down, power control, and the operational conditions in steady and accident states. **Figure 13** shows a basic three-loop PWR which is just an illustration of a real process. Six sensors' process signals from the reactor coolant system (RCS) were selected for monitoring during the start-up operation: S1 (cold leg temperature), S2 (core exit temperature), S3 (hot leg temperature), S4 (safety injection flow), S5 (residual heat removal flow), and S6 (sub-cooling margin temperature). The data consist of 1000 observations sequentially collected at constant time intervals of 1 s.

Because these data are fault-free, we first used the entire 1000 observations to train the method and to determine the optimal model parameters using 10-fold cross-validation. Then, we simulated an abnormal condition within these data for use as the testing data set for model evaluation. To effectively examine the fault
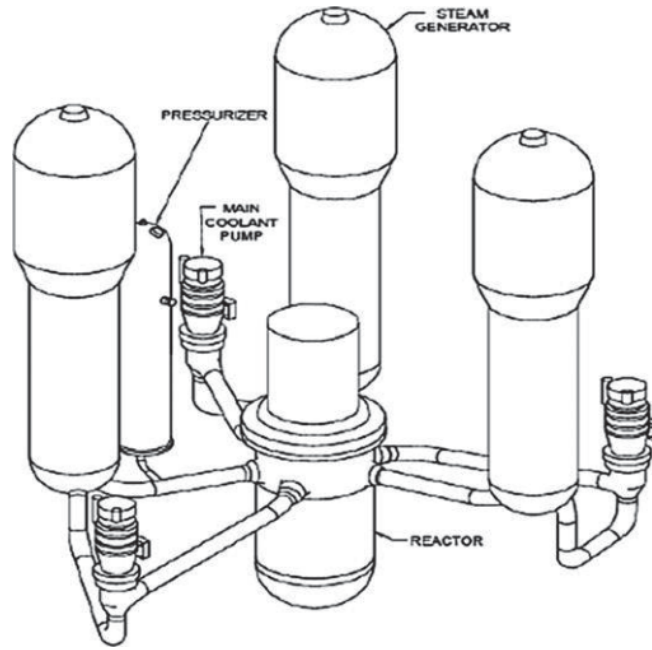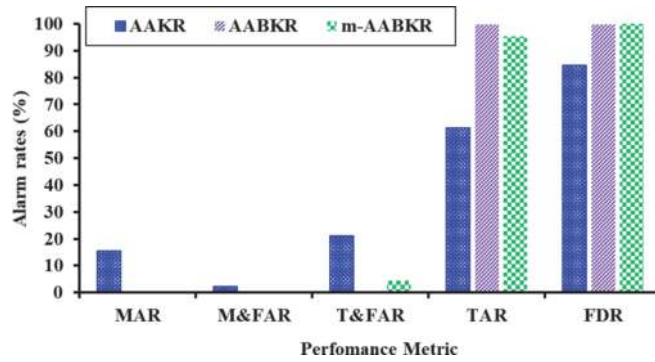
Figure 13.
Schematic diagram of three loops PWR reactor coolant system [60].

detection capability of the proposed method, we conducted a thousand-runs Monte Carlo simulation experiment, in which the fault magnitude was a random number sampled from a bimodal uniform distribution, $U([-10, -2] \cup [2, 10])$, and added to a signal. In this case, the sensor signal in fault at any time step of the Monte Carlo simulation was random uniformly sampled among the six possible sensor signals.

Summary statistics for the alarm rates from AAKR, AABKR, and the modified AABKR from the thousand-runs Monte Carlo simulation experiment are presented in **Table 2**. The mean values of the distributions of the alarm rates are depicted in

Model	Summary statistics	Alarm rates				
		MAR	M&FAR	T&FAR	TAR	FDR
AAKR	Mean	15.43	2.18	21.02	61.37	84.57
	Median	15.41	2.15	21.02	61.35	84.59
	Max	19.54	3.73	25.78	66.13	89.11
	Min	10.89	0.81	16.86	55.82	80.46
AABKR	Mean	0.14	0.01	0.09	99.76	99.86
	Median	0.12	0.00	0.12	99.77	99.88
	Max	0.59	0.23	0.70	100	100
	Min	0.00	0.00	0.00	98.95	99.41
Modified AABKR	Mean	0.01	0.32	4.36	95.30	99.99
	Median	0.00	0.35	4.32	95.33	100
	Max	0.23	0.95	7.50	97.69	100
	Min	0.00	0.00	2.20	91.97	99.76

Table 2.
Alarm rates (%) of a thousand-runs Monte Carlo simulation in start-up data.

Figure 14.
Means of the alarm rates in start-up process operating condition.

Figure 14. The mean values of the distributions of the TARs are 61.4, 99.8, and 95.3% for AAKR, AABKR, and the modified AABKR, respectively. While the mean values of the distributions of the FDRs are 84.57, 99.86, and 99.99% for AAKR, AABKR, and the modified AABKR, respectively. From the results, the performance of both AABKR and the modified AABKR is better than that of AAKR. On average, the TAR from AABKR is slightly higher than that of the modified AABKR. However, there is no significant difference between the FDR of the AABKR and that of the modified AABKR. Thus, for single-sensor faults, the modified AABKR, on average, has performance similar to that of the AABKR, and can be used to validate the sensors' states during transient operations, with the benefit of eliminating the use of derivatives entirely, thereby extending the applicability to steady-state process operation.

6. Conclusions

In this work, we have considered signal reconstruction models for fault detection in nuclear power plants. In order to improve the performance of the AABKR and extend its applicability to steady-state operating conditions, we have proposed a modification, based on a different procedure for the determination of the time position index, the position of the nearest vector within the memory vector to the query vector observation, which provides the input to the weighted-distance algorithm that captures temporal dependencies in the data. Two different approaches based on DTW for time position index identification, have been developed. The basic idea is that, the use of derivative in AABKR, which becomes constant and nearly zero during steady-state operation when the process change in time is negligible and makes it impossible to identify the time position index correctly, can be completely eliminated while maintaining an acceptable performance in monitoring during both steady-state and transient operations.

The modified AABKR method has been applied, first, to a typical steady-state process and, then, to a case study concerning the monitoring of a reactor coolant system of a PWR NPP during start-up transient operation. We have conducted Monte Carlo simulation experiments to critically examine the fault detection capability of the proposed method and the results have been compared to those of AAKR and AABKR using several performance metrics. The obtained results have shown that the reconstructions provided by the modified AABKR are more robust than those of AAKR and AABKR, in particular, during steady-state operations. The method can, then, be used for signal reconstruction during both steady-state and

transient operations, with the benefit of eliminating the use of derivatives entirely while maintaining an acceptable performance. If these approaches are adopted, the cause of abnormalities can be identified, proper maintenance intervention can be planned and earlier mitigation can be allowed to avoid the risk of catastrophic failure.

The future works will focus on (1) the development of an ensemble model in order to benefit from the exploitations of different capabilities of the three methods for signal reconstructions; and (2) the development of a method for on-line updating of the memory data, allowing the model to automatically adapt to the changes in different operating conditions.

A. Appendices

A.1 Auto associative kernel regression

The framework of the AAKR technique comprises three steps, briefly presented below [61, 62].

1) Distance calculation

The distance between the query vector $\mathbf{x}_q$ and each of the memory data vectors is computed. There are many different distance metrics that can be used, but the most commonly used one is the Euclidean distance (L^2-Norm):

$$d_i\left(\mathbf{X}_i, \mathbf{x}_q\right) = \left\|\mathbf{X}_i - \mathbf{x}_q\right\|_2 = \sqrt{\sum_{j=1}^{p} \left(x_{ij} - x_{qj}\right)^2} \tag{A1}$$

For a single query vector, this calculation is repeated for each of M memory vectors, resulting into $\mathbf{d} \in \mathbb{R}^{M \times 1}$.

2) Similarity weight quantification

The distance d_i in vector $\mathbf{d}$ is used to determine the weights for the AAKR, for example, by evaluating the Gaussian kernel:

$$k_i\left(\mathbf{X}_i, \mathbf{x}_q\right) = exp\left(\frac{-d_i^2}{2h^2}\right) \tag{A2}$$

where h is the bandwidth.

3) Output estimation

Finally, the quantified weights (Eq. (A2)) are combined with the memory data vectors to make estimations by using a weighted average:

$$\hat{x}_{qj} = \frac{\sum_{i=1}^{M} k_i x_{ij}}{\sum_{i=1}^{M} k_i} \tag{A3}$$

A.2 Algorithms

A.2.1 Time position index identification—First approach

Algorithm B.1: Time position index identification – First approach

---

Inputs:$X[M, p]$ and $X_q^*[r, p]$ /* **X** is a matrix of memory time-series data with M observational sequences of p-dimensional signals and X_q^* is a matrix of query input time-series data of window length, r */

Output:ε /* time position index */

   1   **begin**
   2   Generate N matrices from **X** with$(r - 1)$ overlapping
   3   **for**$k = 1$ to N **do**
   4   Get $\mathcal{A}_r^k[r, p]$
   5   Compute local cost function matrix, **D** using Eq. (32)
   6   Compute global cost matrix,**G** using Eqs.(27),(28) and (29)
   7   Obtain DTW distance using Eq. (31)
   8   **end**
   9   Determine the time position index, ε using Eq. (33)
  10   **return**ε
          end
 11

A.2.2 Time position index identification—Second approach

Algorithm B.2: Time position index identification – Second approach

---

Inputs:$X[M, p]$ and $X_q^*[r, p]$ /* **X** is a matrix of memory time-series data with M observational sequences of p-dimensional variables, and X_q^* is a matrix of query input time-series data of window length, r */

Output:ε /* time position index */

  1   **begin**
  2   Compute local cost function matrix, **D** using Eq. (32)
  3   Compute global cost matrix **G** using Eqs.(34),(35) and (36)
  4   Get the last row of **G** as **G**$[r,]$
  5   Determine the time position index, ε using Eq. (37)
  6   **return**ε
        end
  7

A.3 Performance metrics

We evaluated and compared the proposed methods using a set of performance metrics proposed in [54]: missed alarm rate (MAR), missed and false alarms rate (M&FAR), true and false alarms rate (T&FAR), true alarm rate (TAR), and fault detection rate (FDR). These metrics are briefly summarized as follows:

A.3.1 Missed alarm rate

A missed alarm occurs when at least one process variable is erroneously not detected as faulty, $\left(\left| e_j \right| \leq T_j^D \right)$, when a fault is actually present. In this case, at least

one missed alarm occurs, and no false alarm occurs in any of the other variables. The MAR is calculated as:

$$\text{MAR} = \frac{\sum missed\ alarms}{total\ number\ of\ samples\ in\ fault\ condition} * 100\% \qquad (A4)$$

A.3.2 Missed and false alarms rate

It is possible that a fault will be missed in a faulty signal but detected in at least one fault-free signal. This gives both missed and false alarms: a fault is detected $\left(|e_j| > T_j^D\right)$ in at least one process signal, when no fault is actually present (false alarm), and at least one process signal has a fault that is not detected $\left(|e_j| \leq T_j^D\right)$ (missed alarm). The M&FAR is calculated as:

$$\text{M\&FAR} = \frac{\sum simultaneous\ missed\&false\ alarms}{total\ number\ of\ samples\ in\ fault\ condition} * 100\% \qquad (A5)$$

A.3.3 True and false alarms rate

It is possible that a fault will be detected in a faulty signal and also detected in at least one fault-free signal. This gives both true and false alarms: a fault is detected in at least one process signal, $\left(|e_j| > T_j^D\right)$, when no fault is actually present (false alarm), and a fault is correctly detected in one process signal, $\left(|e_j| > T_j^D\right)$ (true alarm). The T&FAR is calculated as:

$$\text{T\&FAR} = \frac{\sum simultaneous\ true\&false\ alarms}{total\ number\ of\ samples\ in\ fault\ condition} * 100\% \qquad (A6)$$

A.3.4 True alarm rate

This represents the presence of only true alarms. A fault is detected in at least one process signal, $\left(|e_j| > T_j^D\right)$, when a fault is actually present, and no false alarm exists in other fault-free signals (true alarm). The TAR is calculated as:

$$\text{TAR} = \frac{\sum true\ alarms|no\ false\ alarms}{total\ number\ of\ samples\ in\ fault\ condition} * 100\% \qquad (A7)$$

A.3.5 Fault detection rate

The FDR is expressed as the ratio of the number of faulty data points detected as faulty to the total number of samples specific to a fault. In this case, a fault is detected in at least one process signal, $\left(|e_j| > T_j^D\right)$, when a fault is actually present regardless of false alarms in other signals. The FDR is calculated as:

$$\text{FDR} = \frac{\sum correctly\ decteted\ faults\ in\ the\ system}{total\ number\ of\ samples\ in\ fault\ condition} * 100\% \qquad (A8)$$

FDR measures the ability of a model to detect the presence of the fault in a system when a fault is actually present. Thus, FDR is a summation of the M&FAR, T&FAR, and TAR, which implies that:

$$FDR = M\&FAR + T\&FAR + TAR \qquad (A9)$$

Nomenclature

Abbreviations

AABKR	auto-associative bilateral KR
AAKR	auto-associative KR
AANN	auto-associative artificial neural network
DTW	dynamic time warping
ECM	evolving clustering method
FDD	fault detection and diagnosis
FDR	fault detection rate
GPR	Gaussian process regression
KR	kernel regression
M&FAR	missed and false alarms rate
MAR	missed alarm rate
MSET	multivariate state estimation technique
NPPs	nuclear power plants
PCA	principal component analysis
PLS	partial least squares
PWR	pressurized water reactor
RCS	reactor coolant system
RMSE	root mean square error
SSCs	systems, structures, and components
SVM	support vector machine
T&FAR	true and false alarms rate
TAR	true alarm rate

Symbols and notations

$\mathbf{X}$	matrix of training/memory data set
$\mathbf{X}_q^*$	query test pattern $r \times p$ matrix
$\mathbf{x}_{q_r}^*$	the last rth vector in $\mathbf{X}_q^*$
$\mathcal{A}$	array containing N matrices generated from $\mathbf{X}$
$\mathcal{A}_r^k$	kth $r \times p$ matrix in the array $\mathcal{A}$
M	number of observations in the memory data
N	number of matrices in the array $\mathcal{A}$ generated from $\mathbf{X}$
p	number of process variables
r	sliding window length
i	observation time index of memory data
q	a subscript, indicating query input
j	process variable index
x_{ij}	ith observation of the jth variable
t	time, independent variable
$\hat{\boldsymbol{x}}_{q_r}$	estimated value of $\mathbf{x}_{q_r}^*$
d	distance metric
k	kernel weight of AAKR
h	kernel bandwidth of AAKR
t_q	query time input
f	feature component of AABKR

$\mathbf{k}^f$	feature kernel vector
$\mathbf{k}^t$	temporal kernel vector
h_f	feature kernel bandwidth
h_t	temporal kernel bandwidth
η	constant time interval
ε	time position index
$\partial \mathcal{A}_r^k / \partial t$	derivative of the last row in $\mathcal{A}_r^k$ with respect to time, t
$\partial \mathbf{x}_{q_r}^* / \partial t$	derivative of current measurement vector in $\mathbf{X}_q^*$ with respect to t
$\boldsymbol{\Delta}$	derivative distance vector
$\boldsymbol{\delta}$	temporal weighted distance vector
$\mathbf{k}^{ab}$	adaptive bilateral kernel vector
$\mathbf{e}_{q_r}$	residual vector
T_j^D	threshold for fault detection in the jth variable
$\mathbf{D}$	local cost (distance) matrix of the DTW
$\mathbf{G}$	global cost (accumulated) matrix of the DTW

Author details

Ibrahim Ahmed[1], Enrico Zio[1,2*] and Gyunyoung Heo[3]

1 Department of Energy, Politecnico di Milano, Milan, Italy

2 MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France

3 Department of Nuclear Engineering, Kyung Hee University, Yongin-si, Republic of Korea

*Address all correspondence to: enrico.zio@polimi.it

IntechOpen

References

[1] Rouhiainen V. Safety and Reliability: Technology Theme—Final Report. Espoo: VTT Publications 592; 2006

[2] Coble J, Ramuhalli P, Bond L, Hines JW, Upadhyaya B. A review of prognostics and health management applications in nuclear power plants. International Journal of Prognostics and Health Management. 2015;**6**:1-22

[3] IAEA. On-Line Monitoring for Improving Performance of Nuclear Power Plants Part 2: Process and Component Condition Monitoring and Diagnostics. IAEA Nuclear Energy Series: No. NP-T-1.2. Vienna, Austria: IAEA; 2008

[4] EPRI. On-Line Monitoring of Instrument Channel Performance, Volume 1: Guidelines for Model Development and Implementation. Vol. Vol. 1. Palo Alto, CA: EPRI; 2004. p. 1003361

[5] IAEA. On-Line Monitoring for Improving Performance of Nuclear Power Plants Part 1: Instrument Channel Monitoring. IAEA Nuclear Energy Series: No. NP-T-1.1. Vienna, Austria: IAEA; 2008

[6] EPRI. Equipment Condition Assessment Volume 1: Application of On-Line Monitoring Technology. Vol. Vol. 1. Palo Alto, CA: EPRI; 2004. p. 1003695

[7] Heo G, Lee SK. Internal leakage detection for feedwater heaters in power plants using neural networks. Expert Systems with Applications. 2012; **39**:5078-5086

[8] An SH, Heo G, Chang SH. Detection of process anomalies using an improved statistical learning framework. Expert Systems with Applications. 2011;**38**: 1356-1363

[9] Li F, Upadhyaya BR, Coffey LA. Model-based monitoring and fault diagnosis of fossil power plant process units using group method of data handling. ISA Transactions. 2009;**48**: 213-219

[10] Isermann R. Fault-Diagnosis Applications: Model-Based Condition Monitoring—Actuators, Drives, Machinery, Plants, Sensors, and Fault-Tolerant Systems. Berlin Heidelberg, Springer; 2011

[11] Nabeshima K, Suzudo T, Suzuki K. Real-time nuclear power plant monitoring with neural network. Journal of Nuclear Science and Technology. 1998;**35**:93-100

[12] Baraldi P, Gola G, Zio E, Roverso D, Hoffmann M. A randomized model ensemble approach for reconstructing signals from faulty sensors. Expert Systems with Applications. 2011;**38**: 9211-9224

[13] Jiang Q, Yan X. Plant-wide process monitoring based on mutual information-multiblock principal component analysis. ISA Transactions. 2014;**53**:1516-1527

[14] Ding SX. Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools. Berlin Heidelberg, Springer; 2008

[15] Chen J. Robust residual generation for model-based fault diagnosis of dynamic systems [Ph.D. thesis]. University of York; 1995

[16] Patton RJ, Chen J. Observer-based fault detection and isolation: Robustness and applications. Control Engineering Practice. 1997;**5**:671-682

[17] Simani S, Fantuzzi C, Patton RJ. Model-Based Fault Diagnosis in

Dynamic Systems Using Identification Techniques. London, Springer; 2002

[18] Chiang LH, Russell EL, Braatz RD. Fault Detection and Diagnosis in Industrial Systems. London, Springer; 2000

[19] Ma J, Jiang J. Applications of fault detection and diagnosis methods in nuclear power plants: A review. Progress in Nuclear Energy. 2011;**53**: 255-266

[20] Garvey J, Garvey D, Seibert R, Hines JW. Validation of on-line monitoring techniques to nuclear plant data. Nuclear Engineering and Technology. 2007;**39**:149-158

[21] Zio E. Prognostics and health management of industrial equipment. In: Kadry S, editor. Diagnostics and Prognostics of Engineering Systems: Methods and Techniques. USA: IGI Global; 2012. pp. 333-356

[22] Zio E, Di Maio F, Stasi M. A data-driven approach for predicting failure scenarios in nuclear systems. Annals of Nuclear Energy. 2010;**37**:482-491. DOI: 10.1016/j.anucene.2010.01.017

[23] EPRI. On-Line Monitoring for Equipment Condition Assessment: Application at Progress Energy. Palo Alto, CA: EPRI; 2004. p. 1008416

[24] Ahmad R, Kamaruddin S. An overview of time-based and condition-based maintenance in industrial application. Computers and Industrial Engineering. 2012;**63**:135-149

[25] Zio E, Broggi M, Pedroni N. Nuclear reactor dynamics on-line estimation by locally recurrent neural networks. Progress in Nuclear Energy. 2009;**51**: 573-581. DOI: 10.1016/j. pnucene.2008.11.006

[26] Simani S, Marangon F, Fantuzzi C. Fault diagnosis in a power plant using

artificial neural networks: Analysis and comparison. In: European Control Conference. Karlsrushe, Germany: IEEE; 1999. pp. 2270-2275

[27] Na MG, Shin SH, Jung DW, Kim SP, Jeong JH, Lee BC. Estimation of break location and size for loss of coolant accidents using neural networks. Nuclear Engineering and Design. 2004; **232**:289-300

[28] Yao M, Wang H. On-line monitoring of batch processes using generalized additive kernel principal component analysis. Journal of Process Control. 2015;**28**:56-72

[29] Dunia R, Qin SJ, Edgar TF, McAvoy TJ. Identification of faulty sensors using principal component analysis. AICHE Journal. 1996;**42**: 2797-2812

[30] Harkat M-F, Djelel S, Doghmane N, Benouaret M. Sensor fault detection, isolation and reconstruction using nonlinear principal component analysis. International Journal of Automation and Computing. 2007;**4**:149-155

[31] Heo G, Choi SS, Chang SH. Thermal power estimation by fouling phenomena compensation using wavelet and principal component analysis. Nuclear Engineering and Design. 2000;**199**: 31-40

[32] Cui P, Li J, Wang G. Improved kernel principal component analysis for fault detection. Expert Systems with Applications. 2008;**34**:1210-1219

[33] Zhao C, Gao F. Fault-relevant principal component analysis (FPCA) method for multivariate statistical modeling and process monitoring. Chemometrics and Intelligent Laboratory Systems. 2014;**133**:1-16

[34] Gross KC, Singer RM, Wegerich SW, Herzog JP, VanAlstine R, Bockhorst F. Application of a model-

based fault detection system to nuclear plant signals. In: International Conference on Intelligent Systems Applications to Power Systems (ISAP'97). Seoul, Korea, Argonne National Laboratory; 1997

[35] Singer RM, Gross KC, Herzog JP, King RW, Wegerich S. Model-based nuclear power plant monitoring and fault detection: Theoretical foundations. In: International Conference on Intelligent Systems Applications to Power Systems (ISAP'97). Seoul, Korea, Argonne National Laboratory; 1997

[36] Chen X, Xu G. A self-adaptive alarm method for tool condition monitoring based on Parzen window estimation. Journal of Vibroengineering. 2013;**15**: 1537-1545

[37] Mandal SK, Chan FTS, Tiwari MK. Leak detection of pipeline: An integrated approach of rough set theory and artificial bee colony trained SVM. Expert Systems with Applications. 2012; **39**:3071-3080

[38] Rocco CM, Zio E. A support vector machine integrated system for the classification of operation anomalies in nuclear components and systems. Reliability Engineering and System Safety. 2007;**92**:593-600

[39] Zio E, Baraldi P, Zhao W. Confidence in signal reconstruction by the evolving clustering method. In: Proceedings of the Annual Conference of the Prognostics and System Health Management (PHM 2011). Shenzhen: IEEE; 2011

[40] Muradore R, Fiorini P. A PLS-based statistical approach for fault detection and isolation of robotic manipulators. IEEE Transactions on Industrial Electronics. 2012;**59**:3167-3175

[41] Zio E, Di Maio F. A fuzzy similarity-based method for failure detection and recovery time estimation. International

Journal of Performability Engineering. 2010;**6**:407-424

[42] Zio E, Di Maio F. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. Reliability Engineering and System Safety. 2010;**95**:49-57. DOI: 10.1016/j.ress.2009.08.001

[43] Zio E, Gola G. A neuro-fuzzy technique for fault diagnosis and its application to rotating machinery. Reliability Engineering and System Safety. 2009;**94**:78-88. DOI: 10.1016/j. ress.2007.03.040

[44] Zio E, Baraldi P, Popescu IC. A fuzzy decision tree method for fault classification in the steam generator of a pressurized water reactor. Annals of Nuclear Energy. 2009;**36**:1159-1169. DOI: 10.1016/j.anucene.2009.04.011

[45] Marseguerra M, Zio E, Baraldi P, Oldrini A. Fuzzy logic for signal prediction in nuclear systems. Progress in Nuclear Energy. 2003;**43**:373-380

[46] Ding SX. Data-driven design of monitoring and diagnosis systems for dynamic processes: A review of subspace technique based schemes and some recent results. Journal of Process Control. 2014;**24**:431-449

[47] Hu Y, Palmé T, Fink O. Fault detection based on signal reconstruction with auto-associative extreme learning machines. Engineering Applications of Artificial Intelligence. 2017;**57**:105-117

[48] Hines JW, Garvey J, Garvey DR, Seibert R. Technical Review of On-Line Monitoring Techniques for Performance Assessment Volume 3: Limiting Case Studies. U.S. Nuclear Regulatory Commission, NUREG/CR-6895. Vol. 32008. Washington DC, USA, US Nuclear Regulatory Commission

[49] Baraldi P, Di Maio F, Genini D, Zio E. Comparison of data-driven

reconstruction methods for fault detection. IEEE Transactions on Reliability. 2015;**64**:852-860

[50] Hines JW. Robust distance measures for on-line monitoring. US Patent. Patent No.: US008311774 B2; 2012

[51] Baraldi P, Di Maio F, Turati P, Zio E. Robust signal reconstruction for condition monitoring of industrial components via a modified auto associative kernel regression method. Mechanical Systems and Signal Processing. 2015;**60**:29-44

[52] Ahmed I, Heo G. Development of a modified kernel regression model for a robust signal reconstruction. In: Transactions of the Korean Nuclear Society Virtual Autumn Meeting. Gyeongju, Korea, Korean Nuclear Society; 2016

[53] Ahmed I, Heo G. Development of a transient signal validation technique via a modified kernel regression model. In: 10th International Embedded Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPIC&HMIT 2017). San Francisco, CA, USA, American Nuclear Society; 2017. pp. 1943-1951

[54] Ahmed I, Heo G, Zio E. On-line process monitoring during transient operations using weighted distance auto associative bilateral kernel regression. ISA Transactions. 2019;**92**:191-212. DOI: 10.1016/j.isatra.2019.02.010

[55] Ahmed I. Bilateral kernel methods for time-series states validation in process systems [Ph.D. thesis]. Republic of Korea: Department of Nuclear Engineering, Kyung Hee University; 2020

[56] Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1978;**26**:43-49

[57] Meinard M. Dynamic Time Warping. Information Retrieval for Music and Motion. Berlin, Heidelberg: Springer; 2007. DOI: 10.1007/978-3-540-74048-3_4

[58] Berndt DJ, Clifford J. Using dynamic time warping to find patterns in time series. In: Work Knowledge Discovery in Databases. Palo Alto, California, USA, AAAI; 1994. pp. 359-370

[59] Alcala CF, Qin SJ. Reconstruction-based contribution for process monitoring. Automatica. 2009;**45**: 1593-1600

[60] Xiao B-B, Wang T-C. An investigation of FLEX implementation in Maanshan NPP by using MAAP 5. In: Jiang H, editor. Proceedings of the 20th Pacific Basin Nuclear Conference (PBNC 2016). Singapore: Springer; 2017. pp. 81-96. DOI: 10.1007/978-981-10-2311-8_8

[61] Heo G. Condition monitoring using empirical models: Technical review and prospects for nuclear applications. Nuclear Engineering and Technology. 2008;**40**:49-68

[62] Hines JW, Garvey D, Seibert R, Usynin A. Technical Review of On-Line Monitoring Techniques for Performance Assessment, Volume 2: Theoretical Issues. U.S. Nuclear Regulatory Commission, NUREG/CR-6895. Vol. Vol. 22008. Washington DC, USA, US Nuclear Regulatory Commission