An on-line SAX and HMM-based anomaly detection and visualization tool for early disturbance discovery in a dynamic industrial process

Christos Yiakopoulos[a], Konstantinos Gryllias[b], Moncef Chioua[c], Martin Hollender[c], Ioannis Antoniadis[a]

[a] Dynamics and Structures Laboratory, Machine Design and Control Systems Section, School of Mechanical Engineering, National Technical University of Athens, Athens, Greece
[b] KU Leuven, Department of Mechanical Engineering, Division PMA, Celestijnenlaan 300, BOX 2420, Leuven, 3001, Belgium
[c] ABB AG, Forschungszentrum Deutschland, Ladenburg, Germany

Abstract

In order to achieve an optimum and successful operation of an industrial process, it is important firstly to detect upsets, equipment malfunctions or other abnormal events as early as possible and secondly to identify and remove the cause of those events. Univariate and multivariate statistical process control methods have been widely applied in process industries for early fault detection and localization.
The primary objective of the proposed research is the design of an anomaly detection and visualization tool that is able to present to the shift operator -and to the various levels of plant operation and company management - an early, global, accurate and consolidated presentation of the operation of major subgroups or of the whole plant, aided by a graphical form.
Piecewise Aggregate Approximation *(PAA)* and Symbolic Aggregate Approximation *(SAX)* are considered as two of the most popular representations for time series data mining, including clustering, classification, pattern discovery and visualization in time series datasets. However *SAX* is preferred since it is able to transform a time series into a set of discrete symbols, e.g. into alphabet letters, being thus far more appropriate for a graphical representation of the corresponding information, especially for the shift operator. The methods are applied on individual time records of each process variable, as well as on entire groups of time records of process variables in combination with Hidden Markov Models. In this way, the proposed visualization tool is not only associated with a process defect, but it allows also identifying which specific abnormal situation occurred and if this has also occurred in the past. Case studies based on the benchmark Tennessee Eastman process demonstrate the effectiveness of the proposed approach. The results indicate that the proposed visualization tool captures meaningful information hidden in the observations and shows superior monitoring performance.

Keywords

Process monitoring, novelty detection, Symbolic Aggregate Approximation, Hidden Markov Models, Tennessee Eastman process.

1. Introduction

Proper industrial process monitoring and diagnosis are important not only for quality improvement but also to guarantee process, environment and operator safety. Moreover, unplanned outages and equipment failures can have a significant impact on plant economics. Thus, modern plant processes are usually equipped with distributed control systems (DCSs) to

ensure safe operation and high product quality. Typically within a DCS, an alarm system is installed and alarm thresholds are configured for the important process variables so that the operators can maintain variables within their defined operating limits [1]. These alarm thresholds are carefully determined during the design and commissioning of plants using various process control techniques on online measurement data of variables. This is a strategy of traditional alarm management systems [2], where alarm thresholds are configured for each single process variable, assuming a steady operation state. Also, in addition to statistical process control techniques, in recent years, more intelligent approaches [3-11] have being developed to extract information on dynamic process variability and dynamic control limits from historical data. Using dynamic alarm limits, the unnecessary alarms can be mitigated.

The advent of modern process measurement and information systems has resulted in a significant increase in the amount of process data available to plant operators and engineers. Thus, data-driven techniques for control and monitoring purposes have received considerable attention in the last decades as it is often very difficult to monitor and analyze such large amounts of data and additionally assess the condition of a processing facility. However, most of the available datasets are captured during normal operating conditions. Additionally, an inevitable consequence of the high degree of system complexity is the large number of possible failure modes, whose the effects on observable data are often poorly defined. Thus, there are often insufficient examples of failures to construct accurate fault-detection systems. As a result, conventional fault-specific failure-detection schemes are usually limited to identifying a small subset of known, well-understood modes of failure.

An alternative to identifying rare and unexpected modes of failure is the novelty detection approaches [12, 13], where a model of normality is constructed from normal system data. Departures from normal behavior are classified as novel events. Novelty detection is alternatively known as one-class classification or outlier detection or anomaly detection. As novelty detection methods are able to work when only normal data are available, such methods are of considerable promise for health monitoring in the case of lacking fault samples and prior knowledge.

It has been realized in practice by several studies that the novelty detection is an extremely challenging task. For this reason several models of novelty detection have been propoced performing well on different types of data [14-20]. On the other hand it is clearly evident that there is no single best model for novelty detection and that the success depends not only on the type of the method used but also on the statistical properties of the data handled.

Additionally, the impact of each specific alarm to the overall criticality of the plant can be only indirectly inferred by the operator, based on his overall experience, taking all the potential risks in case of alarm floods. Alarms indicate to the operators that a hazardous situation is gradually under development and give them the opportunity to perform some actions trying to avoid a critical plant situation, such as an emergency shutdown or a catastrophic effect.

Taking into account the above notes, the design of a graphic tool that would present to the shift operator an accurate and consolidated view of the plant critical situation is very important. Visualization is an important mechanism and a fundamental task to present the processed time series for further analysis by users and to discover patterns by a user interface.

Data visualization techniques are very important for data analysis, since the human eye has been frequently advocated as the ultimate data-mining tool. The increasing interest in time series data mining in the last decade has resulted in the introduction of a variety of similarity measures, representations and algorithms [21-23]. However, only a limited number of studies have been

published on the visualization of massive time series datasets while practitioners, who work with time series on a daily basis rarely take advantage of the wealth of tools that the data mining community has made available.

Keogh et al. [24] and Hochheiser and Shneiderman [25] developed a tool called *TimeSearcher* which is a time series exploratory and visualization tool, so that a user can retrieve time series by querying. Based on their previously developed *TimeBoxes*, which are rectangular direct-manipulation time series queries, they proposed an extension by introducing variable time *timeboxes (VTT)*, permitting the specification of queries to allow uncertainty in the time axis. Four methods (sample events, aggregated sample events, event index and interleaved event index) representing the unevenly space time series data are studied in Aris et al. [26]. Furthermore, *TimeSearcher2* has been developed by Buono et al. [27] combining both filter and pattern search capability in a new search interface. This work illustrates how traditional features such as overview and details combined with an interactive search interface can help users to perform exploratory analysis on time series.

Lin et al. [21] introduced a visualization technique called *VizTree* to discover patterns using a tree structure. Initially, the time series data are discretized to symbols through *SAX* [28]. Afterwards, the symbolic representation is encoded into a modified suffix tree wherein each branch represents one subsequence pattern. The frequencies of the various patterns are encoded as the line thickness of the suffix tree; so frequently occurring patterns show up as dense regions and rarely occurring patterns corresponding to possibly anomalies show up as sparse lines. Being the *VizTree* scalable to large data sets, it can perform well only for certain data mining tasks like anomaly detection and motif discovery. Additionally, it is demanding in terms of user training and computer resources.

The same type of representation but using different visualization tools have also been proposed by using dot plots [29]. An efficient dot plotting tool has been developed to exploit the random projection algorithm used to solve the approximate time series motif finding. The scalability and the updatability of the algorithm allow its application for live monitoring of time series data. A dynamic sliding window approach based on time series segmentation is implemented to create more descriptive dot plots for recurrent time series data when the size of the recurrent states varies significantly.

Moreover, Fu et al. [30] extended this work to the discovery of interesting patterns across different resolutions by adopting the symbolic representation of perceptually important points *(PIP)* instead of applying the *SAX* as in the *VizTree*. The proposed method is based on a modified version of *VizTree*. By converting the time series to a symbol string based on data point importance, the potential patterns with different lengths can be encoded in the *VizTree* for visual pattern discovery while the important points and the overall shape of the time series patterns can be preserved even under a high compression ratio.

Recently, Li H. et al [31] proposed a new visualization tool based on shape features called sector visualization. A time series is transformed to a string with several words by Shapes based Symbol Aggregate approXimation *(SSAX)*. Then, in accordance to the frequency of a substring, a Sector Visualization *(SV)* is constructed by setting unequal radius and different colors of fan sector areas and arcs.

Furthermore, Li H. et al. [32] proposed an alternative visualization approach based on binary patterns. The frequency of the occurrences of each binary pattern is counted by forming a matrix that is used to measure the similarity of long time series by the Euclidean distance function. Additionally, the bitmaps for the visualization of the long time series are constructed.

Two alarm data visualization tools, known as the High Density Alarm Plot *(HDAP)* and the Alarm Similarity Color Map *(ASCM)*, which respectively charts top alarms over a given time period and highlights related and redundant alarms in a convenient manner, were introduced by Kondaveeti et al. [33] to assess the performance of alarm systems. The proposed graphical tools are instrumental in performance assessment of industrial alarm systems in terms of effectively identifying nuisance alarms, such as chattering.
Kumar N. et al. [34] employed a simple parameter-light tool that allows users to efficiently navigate through large collections of time series. The proposed approach extracts features from a time series of arbitrary length and uses information about the relative frequency of its features to color a bitmap in a principled way. By visualizing the similarities and differences within a collection of bitmaps, a user can quickly discover clusters, anomalies, and other regularities within their data collection.
In this paper, a novel approach for the visualization of long time series for process control is proposed. The novel statistical scheme is based on the symbolic approach Symbolic Aggregate approXimation (*SAX)* and the stochastic process called Hidden Markov Models *(HMM)*. The proposed *SAX*-based Markov model is implemented as a 2-stage procedure that includes: (a) an off-line training and (b) an on-line testing.

2. Basic theory

High dimensionality of time series hampers the visualization, therefore it has to be reduced. The *S*ymbolic *A*ggregate approXimation (SAX) is used in order to transform data in a symbolic dataset/string which represents the low-dimensional trends of the data. Furthermore, the reduced state-based representation allows for the statistical analysis of anomalous trends by using Hidden Markov Models and Transition Matrix Analysis.

2.1. Theoretical background of Symbolic Aggregate approXimation

Data mining methods, like the classification algorithms, are based on the attribute-value representation of the data, where the explicit temporal relationship between the attributes is not possible. Therefore a fundamental problem in time series data mining is the time series representation. The symbolic representations have demonstrated to be a very effective tool to reduce the dimensionality of the time series and to produce interpretable symbols within the domain.
The Symbolic Aggregate approXimation *(SAX)* is one of the most powerful methods of time series symbolic representation [28]. *SAX* is based on the fact that normalized time series present a highly Gaussian distribution, so by determining the breakpoints that correspond to the alphabet size, one can obtain equal sized areas under the Gaussian curve.
The suitable choice of representation greatly affects the ease and efficiency of time series data mining. For this reason, a number of different versions of *SAX* representations have been introduced.
In [35], Koegh et al. introduced the *HOT SAX* algorithm to find time series discords which are subsequences of a longer time series being maximally different compared to all the rest of the time series subsequences. Discords require only one intuitive parameter (the length of the subsequence) unlike to most anomaly detection algorithms that typically require many parameters.

Shieh J. et al. [36] developed a novel multi-resolution symbolic representation called indexable Symbolic Aggregate approXimation *(iSAX)* that supports indexing of massive datasets. The *iSAX* approach allows for both fast exact search and ultra-fast approximate search. This approach is based on a modification of the *SAX* representation to allow extensible hashing.
The Extension of *SAX (ESAX)* [37] provides a more meaningful representation for different datasets, especially for high frequency dataset such as a financial dataset. The minimum and the maximum values of the subsequence is considered and more important points are added in equal sized segments without losing the symbolic nature of the original approach.
In 2012, Niennattrakul V. et al. [38] developed a novel template matching framework with the use of Dynamic Time Warping *(DTW)* distance, where a shape-based averaging algorithm is utilized to construct meaningful templates. The proposed framework reduces storage and computation requirements as only a few time series sequences are retrieved and compared with an incoming query data.
The classical *SAX* symbolic representation is performed in two steps. First the Piecewise Aggregate Approximation *(PAA)* algorithm is applied to the raw time series.
In *PAA*, each sequence c of time series data of length *n* is divided into *w* segments with equal length and the average value of each segment is used as a coordinate of a *w*-dimensional feature vector. The representation can be visualized as an attempt to approximate the original time series with a linear combination of box basis functions. The *ith* element of *c* is defined as:

$$\overline{c_i} = \frac{w}{n}\sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_i \quad (1)$$

SAX is based on *PAA* and assumes normality of the resulting aggregated values. Thus, before dimensionality reduction, the time series is normalized to have zero mean and standard deviation equal to one, as it is meaningless to compare time series with different offsets and amplitudes. Afterwards the transformed *PAA* representation is symbolized into a sequence of discrete strings. This is achieved by determining the number and the location of the break-points [31]. The number of break-points is related to the desired alphabet size and divide the distribution space into equiprobable regions, where *a* is the alphabet size specified by the user. Once the breakpoints are determined, each region is assigned a symbol. A *SAX* approximation is presented in Fig. 3.
Finally, the main advantages of *SAX* performance are:

- compatibility with low resolution signals
- robustness against high frequency noise
- improved computational efficiency in high dimensionality data

2.2. A brief description of Hidden Markov Models

Hidden Markov Models *(HMM)* [39] are powerful finite state machines that are widely used to model temporal and sequence data. *HMM* are an extension of Markov chains in which the states are not directly visible but only observations depended on the state are visible. Each state has a probability distribution over the possible output observations. Therefore, the sequence of observations generated by *HMM* gives information about the corresponding sequence of states. The terminology 'hidden' means that the observer does not know in which state the system may be in, but has only a probabilistic insight on where it should be. Thus, for a given *HMM*, (a) the

hidden state sequence that is most likely to have generated a given test sequence can be calculated and (b) the probability of a given test sequence can be estimated for given state transition and observation matrices. Both *HMM* capabilities (a) and (b) can be used to solve the anomaly detection problem for discrete sequences.
A Markov chain is a mathematical model that consists of a finite number of hidden states:

$$S = [S_1, S_2, \dots, S_n] , \tag{2}$$

where n is the number of states.
Then, a sequence X_1, X_2, ..., X_n of random elements of a set is a Markov chain if the conditional distribution of X_{n+1} given X_1, ..., X_n depends on X_n only. The set in which the X_i takes values is called the state space of the Markov chain.
A Markov chain has stationary transition probabilities if the conditional distribution X_{n+1} given X_n does not depend on n.
The joint distribution of a Markov chain is determined by

- the marginal distribution of X_1, called the initial distribution
- the conditional distribution of X_{n+1} given X_n, called the transition probability distribution

If the state space is finite, written $\{x_1, ..., x_n\}$, then the initial distribution can be associated with a vector
$\lambda=(\lambda_1, ..., \lambda_n)$ defined by

$$P_r(X_1 = x_i) = \lambda_i, \;\; i = 1, \dots, n \tag{3}$$

and the transition probabilities can be associated with a matrix P_r having elements p_{ij} defined by

$$P_r\left(X_{n+1} = x_j \middle| X_n = x_i\right) = p_{ij}, \;\; 1 \le i, j \le n \tag{4}$$

where p_{ij} is the probability of moving from state S_i to S_j.
Thus, for a Markov chain, the next state depends only on the current state and not on the sequence of the states that preceded it.
As a consequence, according to the above statement, the proposed methodology can presume the current state of a process allowing a Markov model to predict the next state of the process condition. The outcome of *HMM* process is (a) a transition matrix *(TM)* that gives the probability of transition of one state to another and (b) an equilibrium (fixed) probability vector *(PV)* that introduces the distribution of each state.

3. SAX-HMM-based process monitoring framework

In this section, the proposed process monitoring scheme integrating *SAX* and *HMM* is briefly presented. The method consist of two steps, an off-line modeling/training stage and an on-line monitoring/testing stage, described in sections 3.1 and 3.2, respectively. The visualization tool based on *SAX* and *HMM* is described in section 3.3. Moreover, the proposed training and testing procedures are presented in Figures 1 and 2, respectively.

3.1. Off-line modeling/training

(1) The time series of a process variable should be normalized before being transformed to symbols by *SAX*. Without normalization many time series data mining tasks have little meaning. In this case, where offset and amplitude changes are important, normalization parameters (mean and standard deviation) of training procedure are used for the normalization of the testing data.

(2) Then, the symbolic approach *SAX* performs the discretization in two steps. First, the normalized time series is divided into equal-sized segments. The values of each segment can be approximated and replaced by their mean (‘mean representation’) or/and their maximum value (‘max representation’) or/and their minimum (‘min representation’), etc. In this study, each segment is replaced by its 'max' or 'min' (‘max-min representation’) that corresponds to the maximum absolute value of the segment.
Next, a series of breakpoints is determined in order to divide the distribution space into N equiprobable regions, where N is the alphabet size specified by the user. The value of the upper and lower limit of the confidence region depends on the number of the break points [28]. The greater the number of break points is, the greater the threshold values of the confidence region are and vice versa. As a consequence, the amount of the statistically processed SAX-based samples that are located outside of the 95% confidence region decreases as the number of break points increases. Thus, in this case, the alphabet size is considered equal to 20, ensuring higher resolution and minimum number of SAX-based samples outside the 95% confidence region. Once the breakpoints are determined, each region is assigned a symbol.
Figure 3 illustrates a time series with length equal to 200 samples that is divided into 25 equal-sized segments.
As it is observed in Figure 3(a), the time series is converted to an alphabet string *'bdcc**a**c**a**fcqossdrs**t**qs**t**r**t**ss**t**'* using the ‘max-min representation’, while Figure 3(b) shows the same time series that is converted to the alphabet string *'fihhfdkhijklmljopojpnnqpq*' using the ‘mean representation’. It is obvious that the ‘max-min representation’ option in contrast to the ‘mean representation’ is able to preserve the beneficial peak information of the signal outside of the 95% confidence field that is necessary for the proper definition of the state regions. From this point on, in order to visualize process behaviors, it is considered that the symbols inside the 95% confidence region correspond to a normal process condition (called *state 1/ cyan area in Fig. 4* at the online paper version), while the symbols (*a & t*) outside this region correspond to an 'uncertain' condition (called *uncertain state 2/ pink area in fig. 4*).
As it is known, the high dimensionality of time series hampers the visualization, therefore it has to be reduced. Thereafter, as it presented in Figure 4, the alphabet string for a process variable is converted to a numerical string of numbers 1 and 2 *'1111**2**1**2**11111111**2**11**2**1**2**11**2**'*. In that way, the complicated SAX-samples string consisting of multiple alphabet symbols is transformed to a simple numerical string restricting the computational confusion and increasing the efficiency of the HMM algorithm. Hence, the measured signal is transformed from the 'time domain' to the 'state domain' through a simple redundancy reduction procedure.

(3) The *uncertain state 2* consists of the 5% of the normal samples of a training signal. Moreover, the *uncertain state 2* extends into infinity, meaning that in a testing procedure, this field could contain both normal and abnormal data. For this reason, an extra state is considered, called *state 3*, which corresponds to the pure abnormal process condition. As a consequence, a

threshold, called *normal limit*, should be estimated in order to divide the *uncertain state 2* into two fields: (a) *state 2* (still normal variable behavior) and (b) *state 3* (abnormal variable behavior). This estimated limit value will be actually useful in the testing procedure.
Therefore, an appropriate procedure should be performed to (a) estimate a *normal limit* and (b) replace the symbol of the *uncertain state 2* at a numerical string with the symbol of the *state 3* wherever it is necessary.

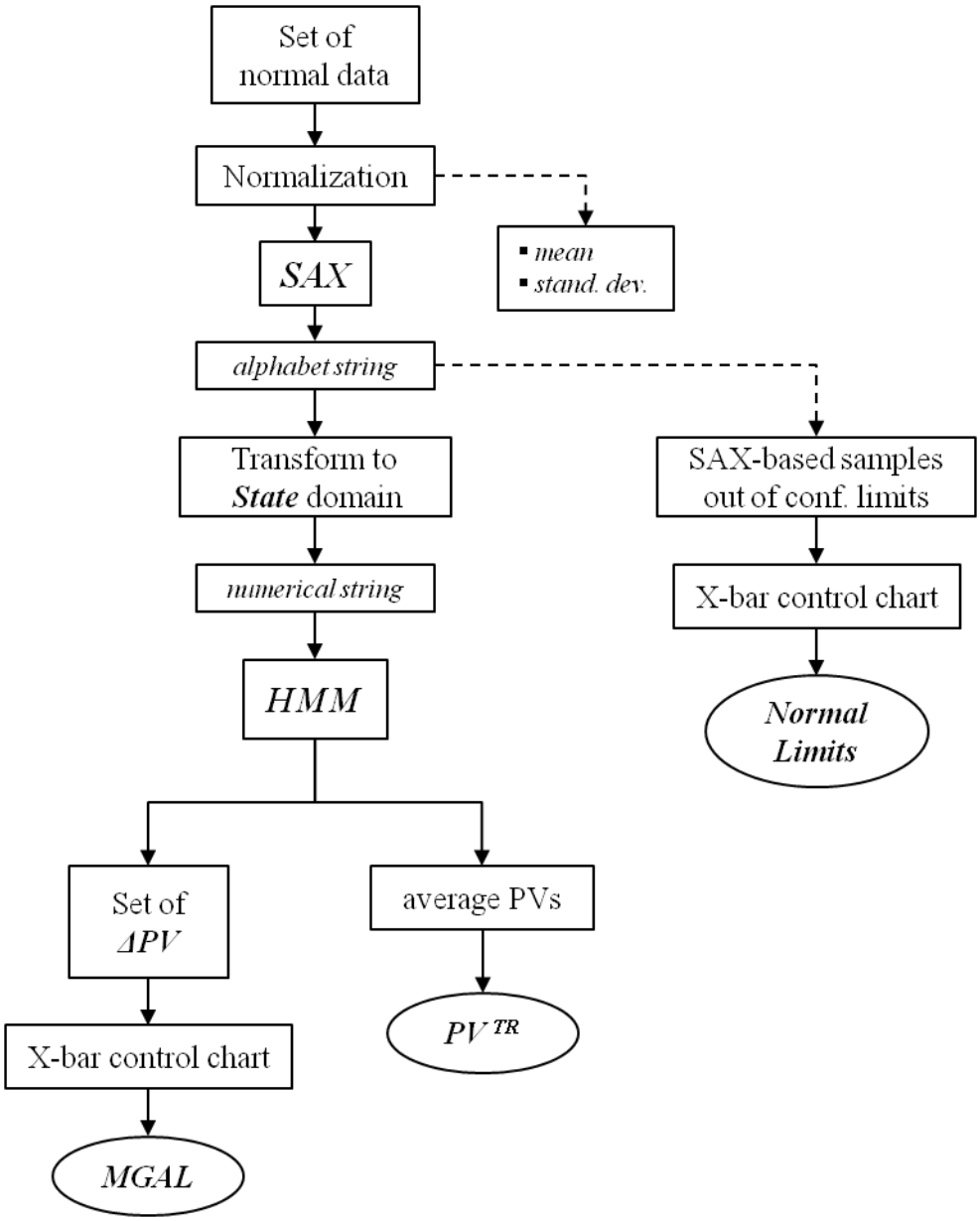


Figure 1: Training process for a single variable.

During this process, the *SAX*-based samples, which are outside of the 95% confidence region, are statistically processed. A control chart procedure (x-bar chart) is used to produce: (a) an upper normal limit that is above the upper limit of the confidence region (green dashed line of 1.64

corresponding to $N = 20$ at Fig. 7) and inside the region of the *uncertain state 2* and (b) a lower normal limit that is under the lower limit of the confidence region (green dashed line of -1.64 at Fig. 7) and inside the region of the *uncertain state 2*. Hence, the *uncertain state 2* is separated by these two *normal limits* to *state 2* (still normal) and *state 3* (abnormal).
Each *SAX*-based sample that is outside the 95% confidence field (higher than 1.64 or lower than -1.64) is compared with the *normal limits*. As it is presented in Fig. 8, if the *SAX* value is higher than the *upper normal limit* ($d_1 > 1.64$ & $d_1 >$ *upper normal limit*) or lower than the *lower normal limit*, the *uncertain state 2* is replaced by *state 3*, otherwise ($d_2 > 1.64$ & $d_2 <$ *upper normal limit*) the *uncertain state 2* retains its characteristics and its name changes to *state 2*. Moreover, the symbol of the *uncertain state 2* of a numerical string is replaced with the numerical symbol of *state 3* wherever it is necessary. In the training procedure, the updated numerical string consists only of symbols 1 and 2.

(4) The updated numerical string for a single variable is further processed using Hidden Markov Models. The aim of this process is to detect the current state of the process predicting its next possible condition state through the implementation of the Markov model. The outcomes of this process are (a) a transition matrix (*TM*) and (b) a probability vector (*PV*), called the fixed vector of *HMM*.
Figure 5 presents the transition diagram of the Markov model for the training stage including 2 process states and the probabilities A_{ij} of moving from one state to another. In the produced *TM*, the states are indicated by the side and the top. All entries of *TM* are between 0 and 1 while the sum of the entries in any row must be 1.
The second outcome, the *PV*, gives the probabilities of a transition from one state to another in k repetitions of an experiment. Thus, the produced *PV* of training stage has 2 elements. The first element x_1, which is the distribution of *state 1*, corresponds to the pure normal process condition and the second element x_2, which is the distribution of the *uncertain state 2* or the *state 2* corresponds to the 'still normal' process condition. In accordance to (3), an extra element x_3 is added to the *PV* that is produced in the training process, which is the distribution of *state 3*, corresponding to the pure abnormal condition. In the training procedure, the sum of distributions x_1 and x_2 of the *PV* should be 1 (Fig. 6(a)), while the element x_3 is considered equal to 0.
Figure 6(a) visualizes a *PV* produced by a training process. The blue colored section of the chart pie corresponds to the element x_1, while the green colored section is relative to the distribution of the *state 2*. Therefore, the blue part dominates on a pie that corresponds to a pure normal time series, while there is no red sector. A green chart pie is also relative to a normal behavior of the testing variable.

(5) Given a series of *PVs* of a variable for a set of training/normal time series, a global probability vector PV^{TR} for a variable in normal state is estimated as the average value of these *PVs*.

(6) Moreover, the *PVs* difference between 2 time series of the same variable is used to describe how dissimilar or not these time series are. This distance measure is called State Indicator *STI*.
Therefore, for a variable in normal state, the distances between the *PVs* of all the training time series are calculated. These *STIs* describe the *'permissible dissimilarity'* between normal time series of the same variable.

Finally, for each variable, a statistical procedure through a x-bar control chart is performed using these *PVs* differences (Fig. 9). The resulted threshold for each variable is called *Markov Generated Alarm Limit MGAL* and corresponds to a local limit that detects normal and abnormal behavior.

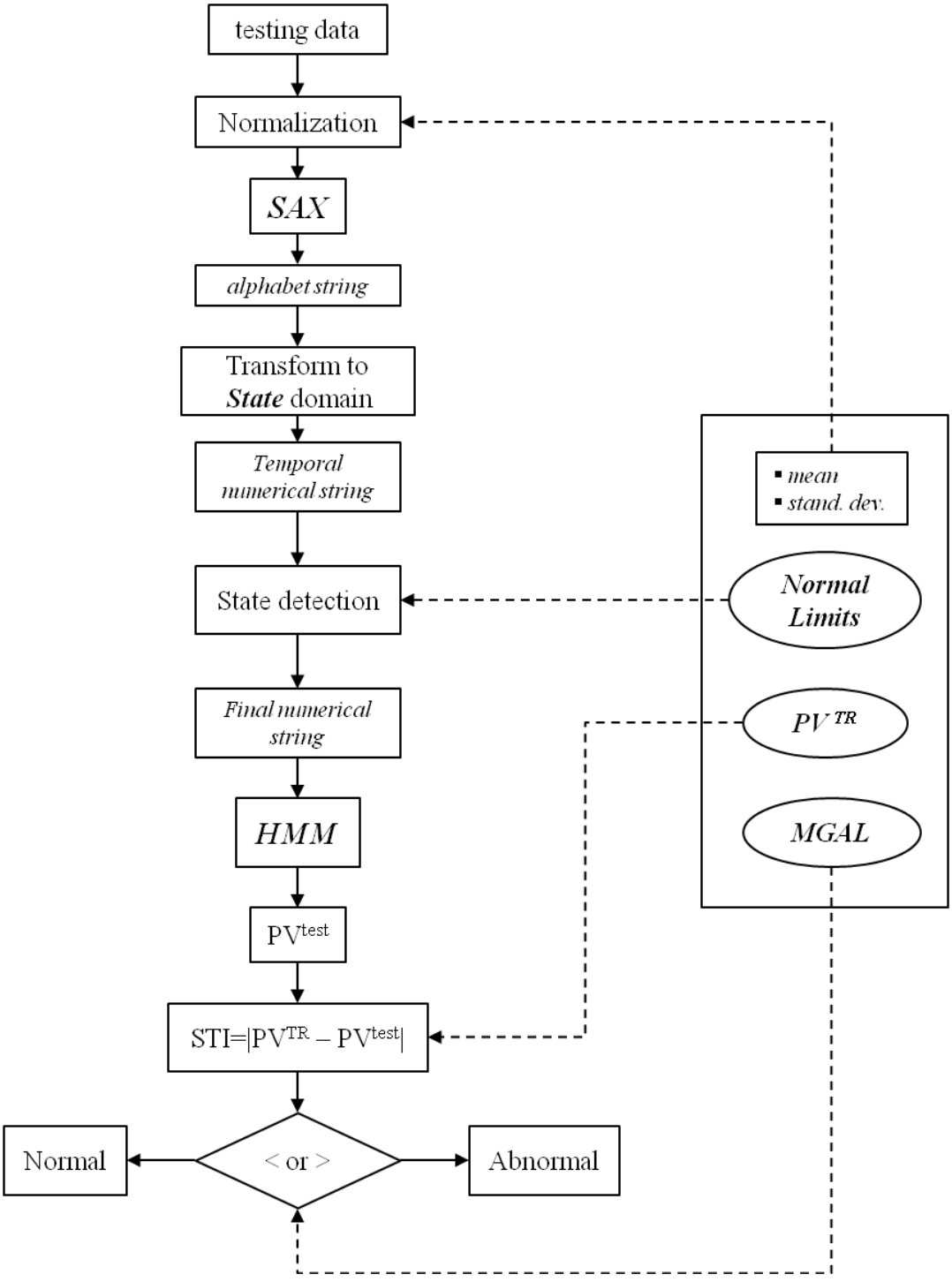
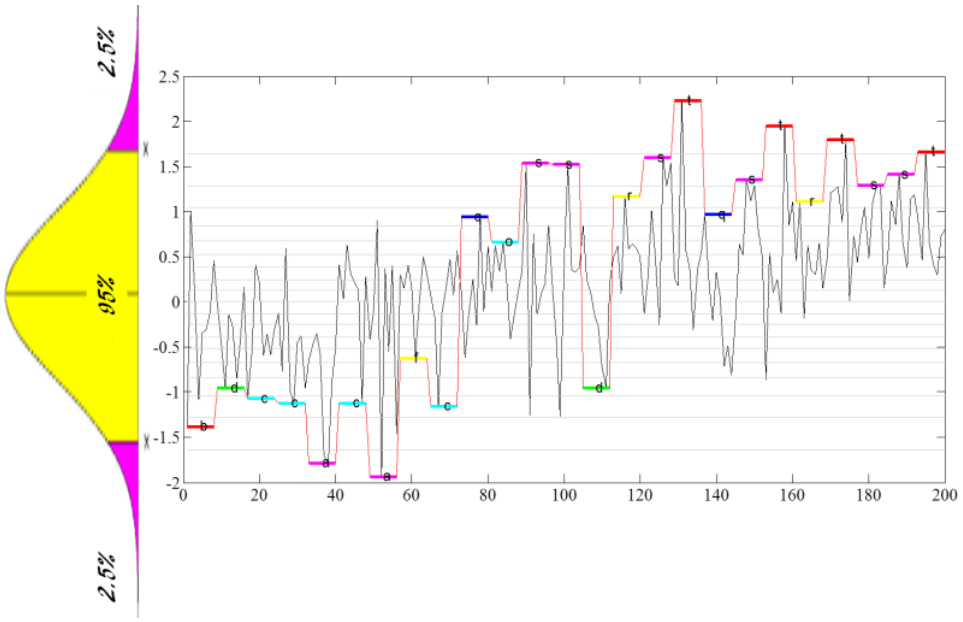
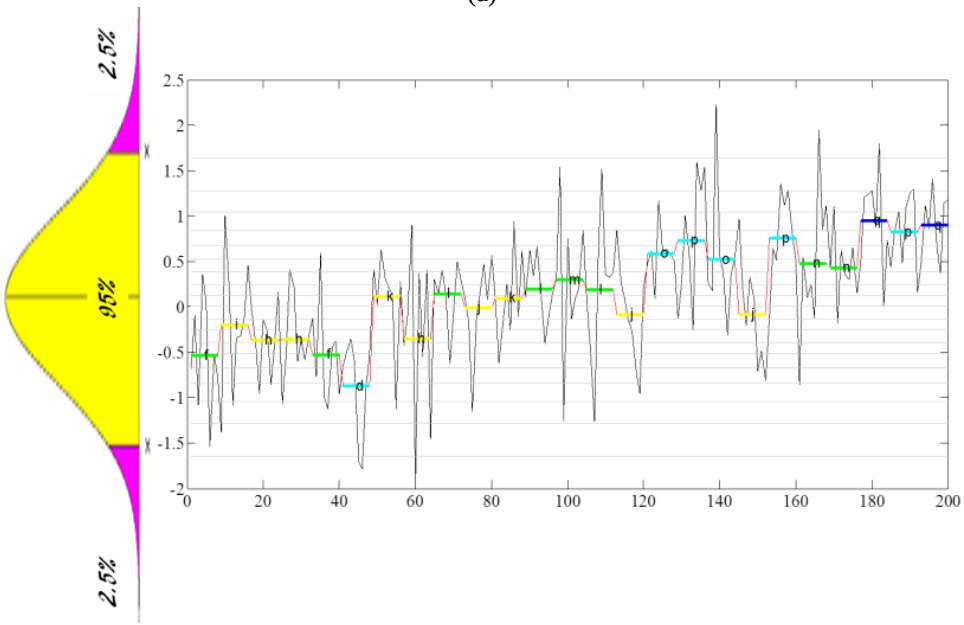


Figure 2: Testing process for a single variable.



(a)



(b)

Figure 3: SAX-based samples corresponding to (a) abs(max) and (b) mean of a time series segment.

As a conclusion of the training process, the outcomes are:

- a mean and standard deviation for each variable in normal state that will be used for the normalization of a relative testing signal
- a *MGAL* for each variable in normal state
- a probability vector PV^{TR} for each variable in normal state

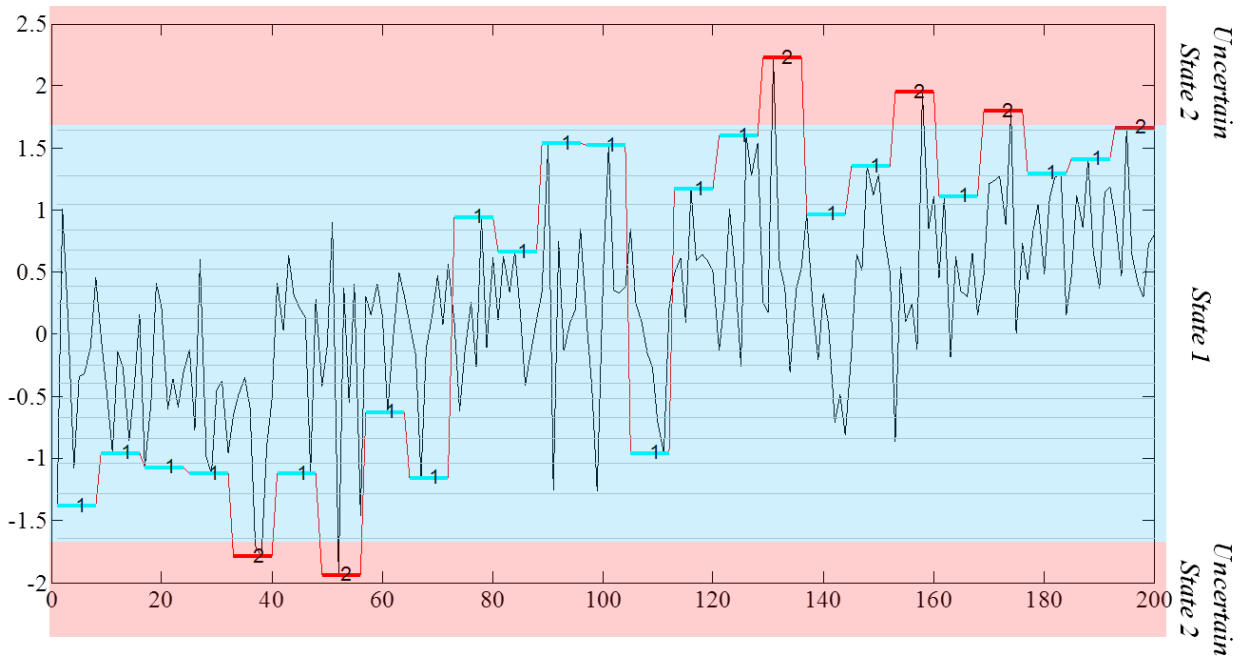


Figure 4: A time series has been transformed to the SAX-based *'state domain'*.

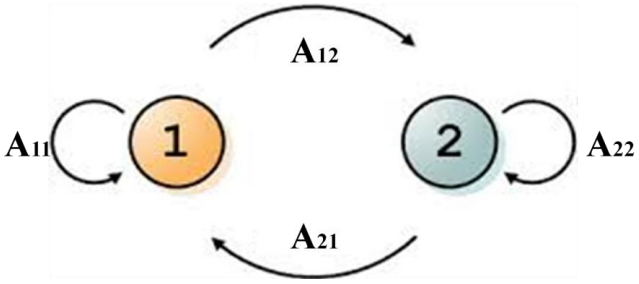
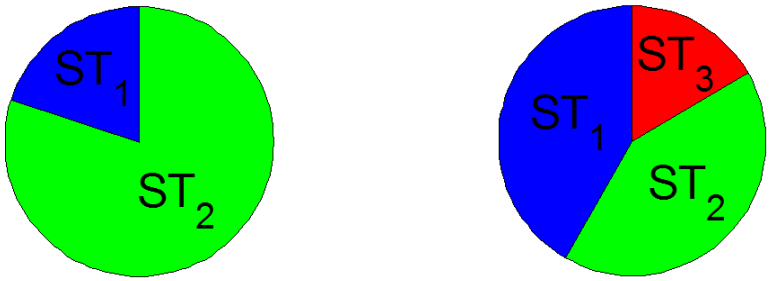


Figure 5: The transition diagram of *state 1* and *temporary state 2*.

3.2. On-line monitoring

The steps of a testing process are similar to those of the training process. Initially, the testing time series of each variable are normalized using the mean and standard deviation of the relative variable in normal state of the training process.
Then, *SAX* is performed to these normalized time series. An alphabet string and a temporary/initial numerical string are produced for each process variable. The statistical procedure mentioned in the previous paragraph is performed in order to redefine the states replacing *uncertain state 2* with *state 2* and/or *state 3*. Next, *HMM* is implemented using the final/updated numerical string.



(a) (b)

Figure 6: Visualization of a *PV* that is calculated during (a) a training and (b)a testing process.

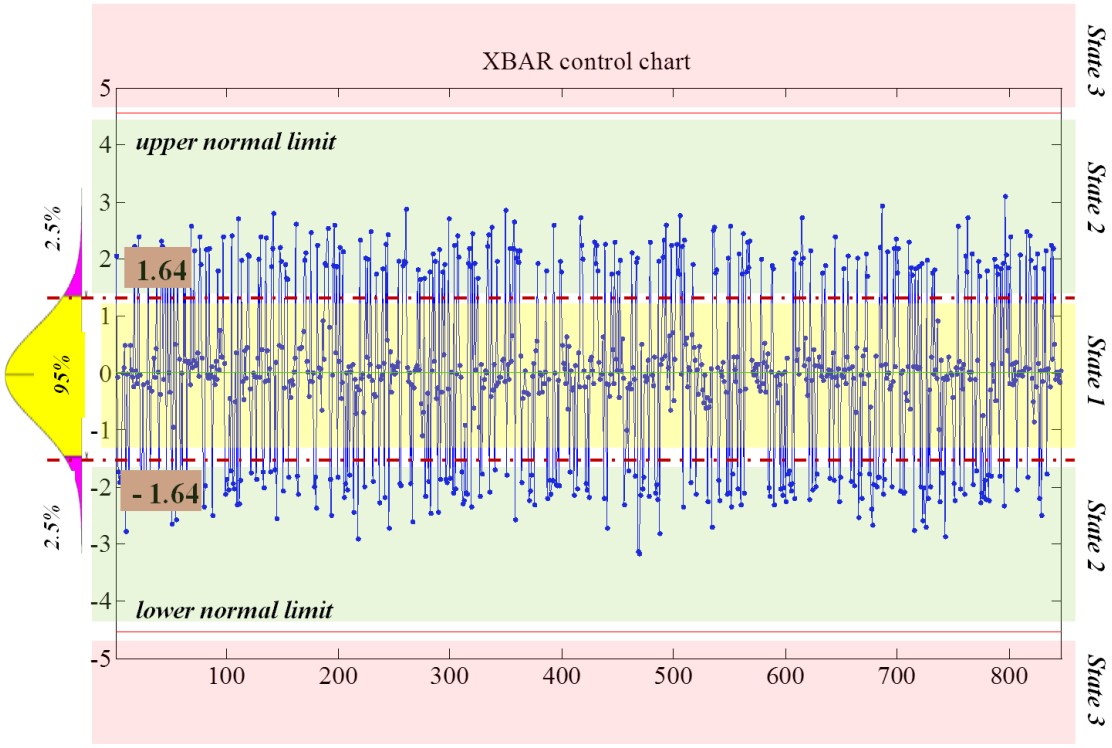


Figure 7: Limits of *states*.

Hence, as it is shown in Figure 2, a probability vector *PV* is estimated for each single testing variable and its difference ΔPV with the relative probability vector PV^{TR} of the training process is used to calculate the *STI* distance. Thus, the produced numerical string for a testing variable that deviates from its normal operating behavior includes one, two and three numbers. As a consequence, the produced *PV* performing *HMM* has 3 elements. Figure 6(b) visualizes a *PV* produced by an abnormal testing process. Please, note that a *PV* can be presented as a pie chart of 3 parts maximum (Fig. 6(b)). The blue part is relative to *state 1*, the green part corresponds to *state 2* and the red part is the *state 3*.

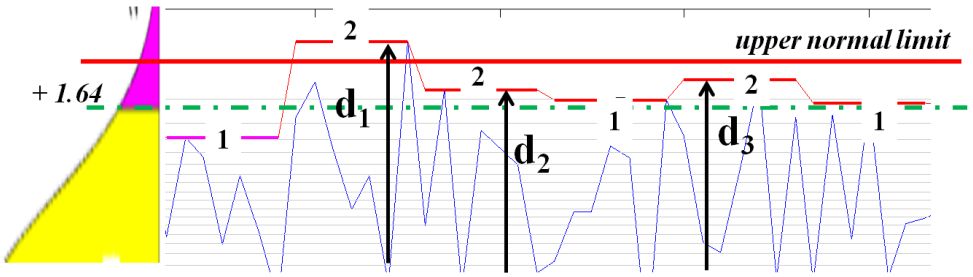


Figure 8: Data separation procedure of *uncertain state*.

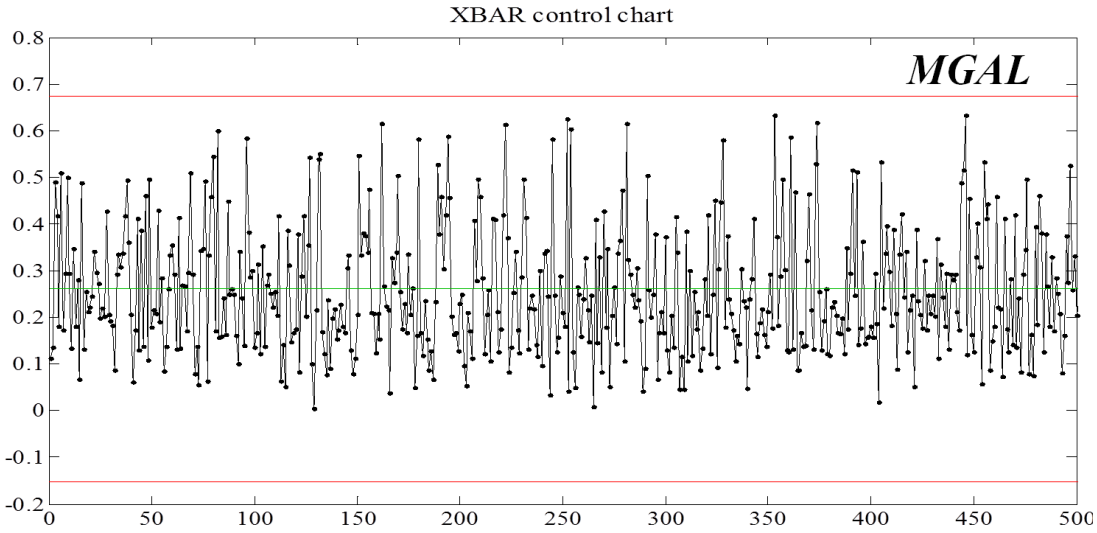


Figure 9: *MGAL* estimation using a x-bar control chart.

As a consequence, a testing variable is considered to deviate from its normal behavior if the calculated *STI* difference is higher than the relative threshold *MGAL* that is estimated in the training process, otherwise this variable demonstrates a normal state.

3.3. Visualization tool

Time series visualization is one of the most fundamental tasks to present the essential phenomena hiding in large time series databases. The visualization tool based on *SAX* and *HMM* only reflects the changeable *STI* values over time.

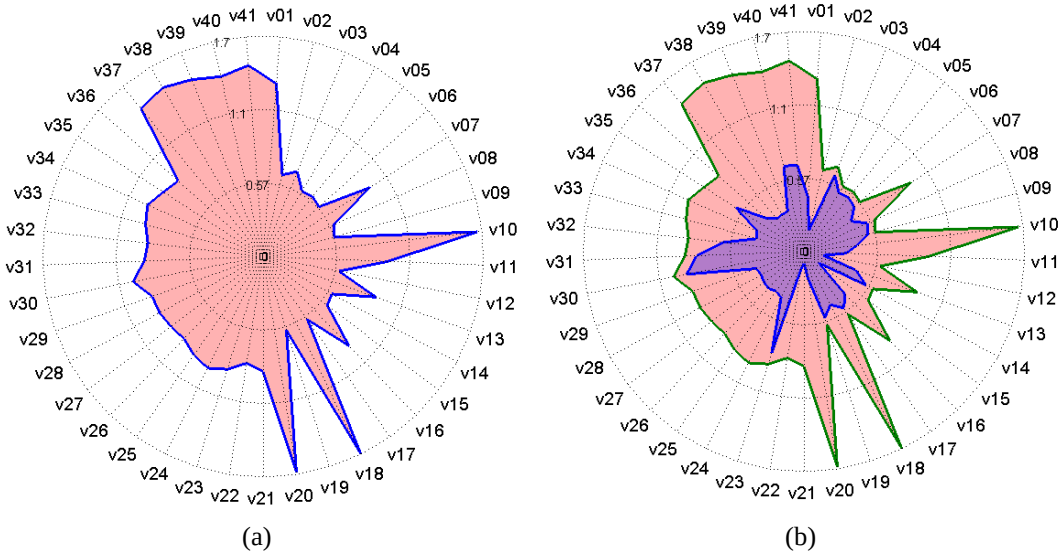
Once the *MGAL* values of all the process variables have been calculated, then their radial representation is created. Radar plot is a graphical method displaying multivariate data in the form of a two-dimensional chart of three or more quantitative variables represented on axes starting from the same point. The relative position and angle of the axes is typically uninformative.

As shown in Fig. 10(a), a pink area is shaped from the *MGAL* values in a radar plot. The enclosed central shape is highlighted as necessary for ease in viewing. This formed section provides an overall realistic and useful picture of the entire normal process. The resulting radar

plot graphically shows areas/process variables of relative strength and relative weakness, as well as depicting the general overall performance.

Some knowledge and information for the overall trending of the process can be demonstrated and revealed by monitoring the overall *STI* shape of all the process variables. The relationship between *MGALs* region and *STIs* region, as shown in Fig. 10(b), can generate useful alarms in fault situations.

As it can be observed in the radar plots of Fig. 10(b), the blue region formed by the instant *STIs* is enclosed in the pink area that corresponds to the normal state. In other words, the local *STI* value for each single process variable is lower than its relative *MGAL*, representing the normal behavior. For human eye, the radar plots seem to be an easily understandable visualization tool that provides an instant picture of the entire process condition.



(a) (b)

Figure 10: (a) A normal region formed by *MGAL* values and (b) instantaneous *STI* values of testing process variables in a radar plot.

4. Application to the TE benchmark process and analysis of results

4.1. Process description

In this subsection the well-known Tennessee Eastman Process (*TEP)* is used in order to evaluate the effectiveness of the *SAX-HMM*-based monitoring scheme in disturbance detection. The *TEP* was developed by Down and Vogel [40] to provide a realistic industrial process for control and monitoring studies. The base control scheme of *TEP* and the process variables are described in [40]. The plant as shown in [40, 41] consists of five major unit operations: a reactor, a product condenser, a vapor-liquid separator, a recycle compressor and a product stripper. *TE* process contains two blocks of process variables: 41 measured variables and 11 manipulated variables. All process measurements are contaminated by Gaussian noise. In this study, for process monitoring purpose, 22 continuous and 19 composition process variables are selected as input variables. The sampling rate of input variables is set to 0.01 h and their length *L* is equal to 7,000 samples simulating 70 operating hours.

10 sets of training data with 7,000 normal samples in each set are generated under the operating mode 1 and then 20 types of disturbances that are described in [40] were monitored. The normal datasets are used to develop the off-line model producing the normal region presented in Fig. 10(a) that is formed by the *MGAL* values, while the process data corresponding to the set of the simulated faults are collected for testing. Initially, the normal data that are used for the experiments were normalized to avoid inaccuracy problems due to the scale difference between variables. The testing data are normalized and scaled using the mean value and the standard deviation of the corresponding training data. Each testing data set starts with no fault present and then, the fault in each testing case is introduced at a monitoring instant from No. 450 up to No. 500.

A monitoring instant corresponds to a sample of the State Indicator STI^{var} trajectory. The number of the monitoring instants t_{sti} is defined as follows:

$$t_{sti} = \frac{L - N_{str}}{n_{shift}} + 1 \quad (5)$$

The performance of the proposed methodology depends on the variation of three key parameters:

i) The length of the test time series N_{str}.
ii) The number of the samples n_{shift} that the current test time series is shifted over time resulting to the updated time series that is going to be studied.
iii) The number of segments n_{seqt} in which the test time series is divided by the *SAX*.

Moreover, the length of the test time series N_{str} corresponds to the minimum number of samples of the measured signal which should be used in order to calculate the first *PV* and the *STI*, and start the online monitoring procedure.

In this study, the number of samples n_{shift} is considered fixed and equal to 8 samples. This means that the test time series is updated every 4.8 min. Moreover, the length of both the training and the testing signals is considered equal to 48 samples. Additionally, each time series is divided into 3 and 6 equal-sized segments. The option A (N_{str} = 48, n_{shift} = 8, n_{seqt} = 6) is used for the training stage and the option B (N_{str} = 48, n_{shift} = 8, n_{seqt} = 3) is implemented in the testing stage increasing the effect of the last measured samples on the final result.

4.2. Disturbance detection

The Tennessee Eastman process simulation contains 20 preprogrammed faults [40]. Faults 1-7 are associated with step changes in different process variables. Faults 8-12 are associated with random variations in certain variables. Fault 13 presents a slow drift in the reaction kinetics. For Faults 14 and 15, two cooling water valves are stuck. Faults 16-20 are unknown disturbances in the *TE* process. In this study, Faults 1 and 19 are employed and the monitoring results are presented and discussed.

4.2.1. Case study A on Fault 1

Fault 1 involves a step change in the *A/C* feed ratio in stream 4 and the knock-on effects are compensated by the control loops. Since more than half of the monitored variables deviate significantly from their normal behavior, the fault is expected to be easily detected. The fault is introduced at the sample No. 3900 (Fig. 12(b)) in the process.

The results of the proposed visualization tool based on *SAX* and *HMM* at different monitoring instants of the process are illustrated in Fig. 11. As it can be observed in Fig. 11, the continuous process variable 21 is the first variable which is captured (Fig. 11(b)) to deviate from its normal behavior. The rest 40 measurement and manipulated variables remain steady after the fault occurs keeping their normal operating behavior based on the possible activation of the in-control operations.

The panels of Fig. 11 present some updates of the produced radar plot, as the disturbance deteriorates. In the radar plots of the monitoring instants Fig. 11(b) – Fig. 11(g), it can be observed that the normal field is overlaid by the blue region shaped by the instant *STIs*. This temporal relationship of these regions for multiple variables reveals a process progression that deteriorates over time. Moreover, as it can be observed in the last panels Fig. 11(h) and Fig, 11(i), the controllers' reaction assist some process variables to return to their normal operating behavior.

The trajectory of the first affected process variable 21 when fault 1 occurs is shown in Fig. 12(b). It is worth noting that the $MGAL^{21}$ of the process variable 21, which has been estimated in the training phase, is equal to 0.8848 (purple dash line in Fig. 12(a)), while the calculated probability vector using *HMM* is $PV^{21}{}_{norm}$ = [0.5454 0.4547 0.0]. Following the trajectory of the State Indicator STI^{21} of the variable 21 in Fig. 12(a), it is obvious that its values are higher than the $MGAL^{21}$ after the monitoring instant No. 483. The *SAX* representation of the testing time series in the monitoring instant No. 483 using the option B is presented in Fig. 14. The monitoring time instant No. 483 includes the measured samples 3,857 - 3,904 of the variable 21. The last one *SAX*-based value lies in an *abnormal state 3*, while the other values move inside the estimated normal limits. As a consequence, the corresponding transition matrix $TM^{21/483}$

$$TM^{21/483} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.5 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (6)$$

results to the estimated probability vector $PV^{21/483}{}_{test}$ = [0.0 0.5 0.5] meaning that the state behavior of the variable in monitoring instant No. 483 corresponds to the *abnormal state 3*. Furthermore, in agreement with $PV^{21/483}$, the half of the corresponding pie chart is painted red (Fig. 14). An important remark is that the elements $TM^{var/instant}{}_{(i,j)}$ of the transition matrix $TM^{var/instant}$ are used to quantify the state information and estimate the probability that the process variable will switch or remain to a state. In this case, the elements $TM^{21/483}{}_{(2,2)}$ and $TM^{21/483}{}_{(2,3)}$ indicate that the process variable 21 remains temporally to *still normal* state 2 and transits to *abnormal state 3*, but there is no clear information if it remains to the current *state.* Thus, the proposed methodology based on the concept of *HMM* presumes that the current state of the process variable gathers high probability to be abnormal. Then, the ΔPV distance is calculated giving the State Indicator STI^{21} of variable 21 at the monitoring instant No. 483 a value equal to 1.091. As it can be observed in Fig. 12(a), the calculated State Indicator STI^{21} for the monitoring instant No. 483 is higher than the $MGAL^{21}$. Thus, the proposed method achieves to detect without delay that this variable deviates from its normal behavior.

Continuing the analysis, the previous *SAX*-based approach of the testing signal in monitoring instant No. 482 is presented in Fig. 13. The monitoring time instant No. 482 includes the measured samples 3,849 - 3,896 of the variable 21. All the *SAX*-based values lay in *normal states 1 and 2* resulting through *HMM* to the transition matrix $TM^{21/482}$

$$TM^{21/482} = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (7)$$

and the estimated probability vector $PV^{21/482}_{test}$ = [0.5 0.5 0.0] (pie chart presented in Fig. 13). As it can be observed in Fig. 12(a), the State Indicator STI^{21} (= 0.0907) of the variable 21 takes a value lower than the $MGAL^{21}$ in the monitoring instant 482. The elements $TM^{21/482}_{(2,1)}$ and $TM^{21/482}_{(1,2)}$ indicate that the process variable 21 remains to *normal state*. Thus, in this case study, the proposed method using option *B* results in decreased false and missing alarm rates.

Moreover, another practical problem is the possibility of incorrect continuation of the abnormality indication that may result to possible repeated false alarms that would not notify the operators in time whenever the variable returns to its normal operating behavior.

One approach to address this challenge is to study and compare the outcomes of the proposed method at the monitoring instants No. 682 and No. 683 when the variable 21, as presented in Fig. 12(a), enters and moves again in the normal zone as a result of the controllers' reaction.

The *SAX* representations of the test time series in the monitoring instants No. 682 and No. 683, presented in Fig. 15 and 16, can justify a minimum delay of the estimated STI^{21} to return in time to normal values.

Once, the *SAX*-based samples have been produced using option *B* (Fig. 15), the probability vector $PV^{21/682}_{test}$ = [0.0 0.5 0.5] (pie chart presented in Fig. 15) for the monitoring instant No. 682 is estimated through:

$$TM^{21/682} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.5 \end{bmatrix} \quad (8)$$

The elements $TM^{21/682}_{(3,2)}$ and $TM^{21/682}_{(3,3)}$ indicate that initially the process variable 21 transits from *state 3* to *state 2* but it is not clear if it will remain permanently to *still normal state 2*. The ΔPV distance at the monitoring instant No. 682 is calculated giving a decreasing State Indicator of variable 21 (1.091) but which is still higher than $MGAL^{21}$, as the variable is moving into its normal operating behavior. The monitoring time instant No. 682 includes the measured samples 5.449 - 5.496 of the variable 21.

At the next monitoring instant No. 683 (Fig. 16), the corresponding transition matrix $TM^{21/683}$

$$TM^{21/683} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix} \quad (9)$$

results to the estimated probability vector $PV^{21/683}_{test}$ = [0.0 1.0 0.0] demonstrating that the process variable 21switches and remains to *still normal state 2*.

Thus, the proposed approach using option *B* doesn't notify the operators immediately that the variable 21 returns to its normal operating behavior and delays the normality detection (missing alarm) for one *SAX*-based sample or 4.8 minutes (Fig. 12(a)).

Nevertheless, in case the variable temporally crosses the estimated normal limits because of, for example, an inaccuracy of the training stage and then it returns to its abnormal field, the proposed method using option *B* provides a right in-control signal.

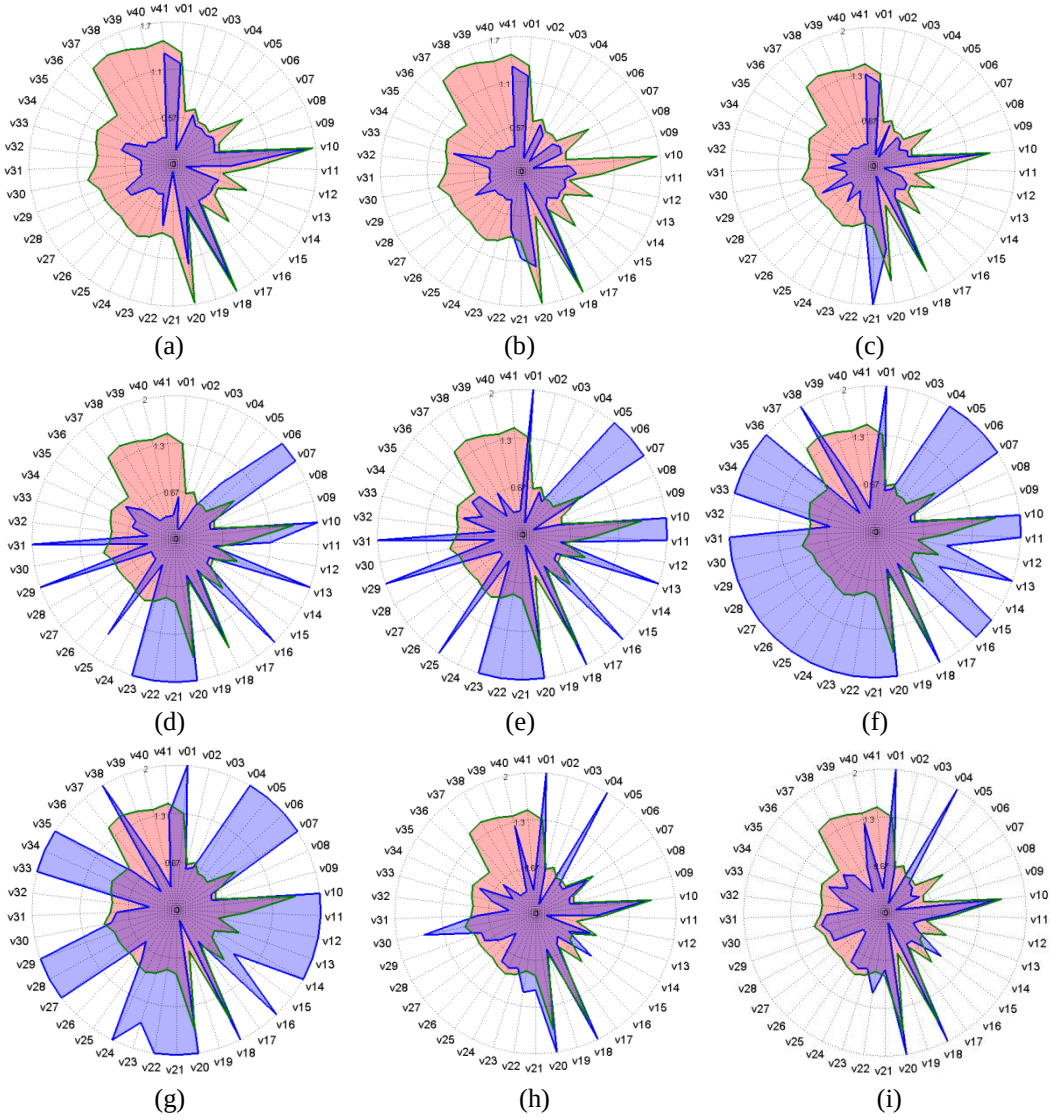


Figure 11: Visualization of the process corresponding to the case A for the monitoring instants: (a) No. 482, (b) No. 483, (c) No. 484, (d) No. 488, (e) No. 489, (f) No. 520, (g) No. 550, (h) No. 682 and (i) No. 683.

In order to assess the quality of the proposed monitoring scheme and evaluate the monitoring performance of the process, more fault detection results are presented in Fig. 17.

The measured variable 1 responds to the disturbance with a time lag of 28.8 min and then fluctuates as a result of the controllers' reaction. It can be clearly seen that the variable still deviates from its normal operating behavior when the controllers attempt to handle the disturbance.

As demonstrated in the previous example, the state indicator STI^1 is detected below the $MGAL^1$ (=1.334, purple dash line in Fig. 17(a)) before the fault occurs. Moreover, the State Indicator of the variable 1 is higher than the $MGAL^1$ after the monitoring instant No. 489 corresponding to 3,905 - 3,952 samples of the measured variable $x_{meas}01$. As it can be observed in Fig. 17(a), once the variable 1 is affected by the disturbance, the state indicator STI^1 takes its maximum value in an instant and remains constant and above the $MGAL^1$ until the end of the process.

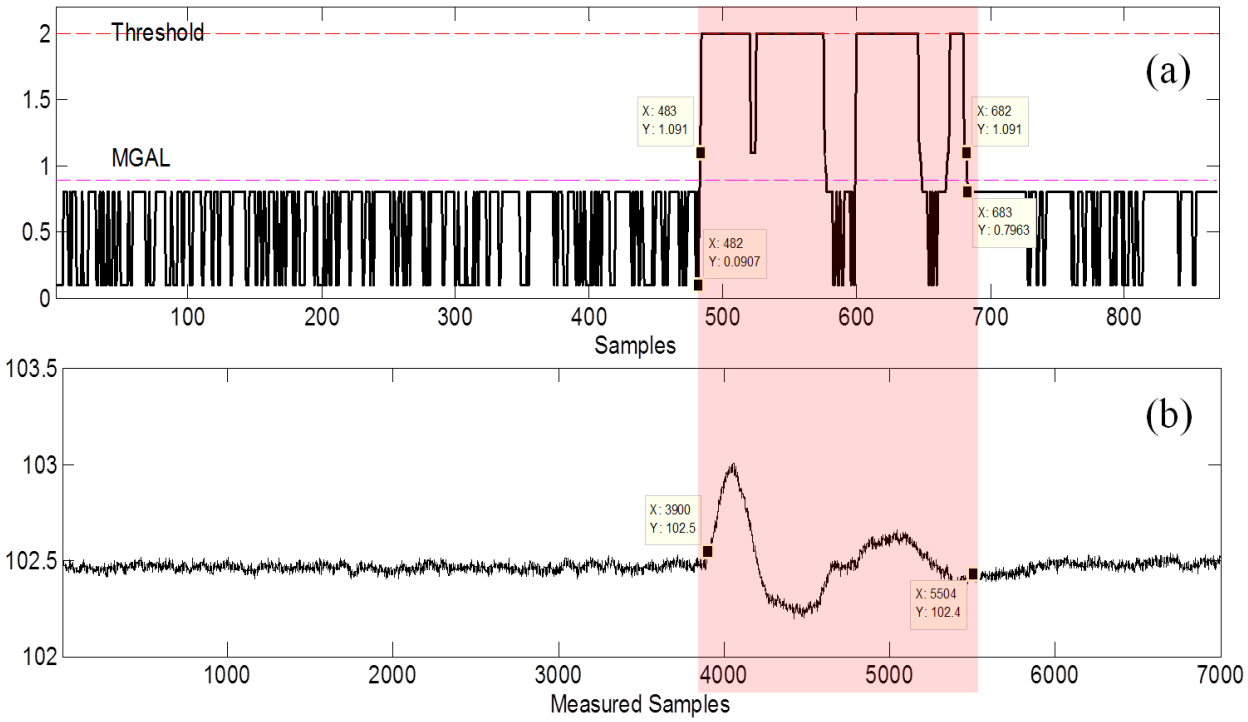


Figure 12: (a) STI trajectory and (b) measurements of the process variable 21 for the entire process of case A.

The *SAX* representation of the test time series in monitoring instant No. 489 is illustrated in Fig. 19. The first two *SAX*-based values lie in *normal state*, while the last one moves over the estimated upper normal limit. Then, the corresponding transition matrix

$$TM^{1/489} = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (10)$$

results to the estimated probability vector $PV^{1/489}{}_{test}$ = [0.0 0.0 1.0] (red pie chart presented at Fig. 19) demonstrating that the state behavior of the variable in the monitoring instant No. 489 corresponds to the *abnormal state 3*.
Then, the ΔPV distance (=$|PV^{1/489}{}_{test}$ - $PV^{1}{}_{norm}$ |, *where* $PV^{1}{}_{norm}$ = [0.7921 0.2079 0.0]) is calculated giving the State Indicator STI^{1} equal to 2.0 (≡ threshold = $\Delta PVmax$).
Next, for comparison purposes, the monitoring instant No. 488 is tested. This monitoring instant includes the measured samples 3,897 - 3,944 of the variable 1. As shown in Fig. 18, all the *SAX*-based samples are inside the normal limits. This behavior can also be observed through the transition matrix $TM^{1/488}$

$$TM^{1/488} = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (11)$$

and the corresponding probability vector $PV^{1/488}{}_{test}$ = [0.5 0.5 0.0] (blue-green pie chart presented at Fig. 18).

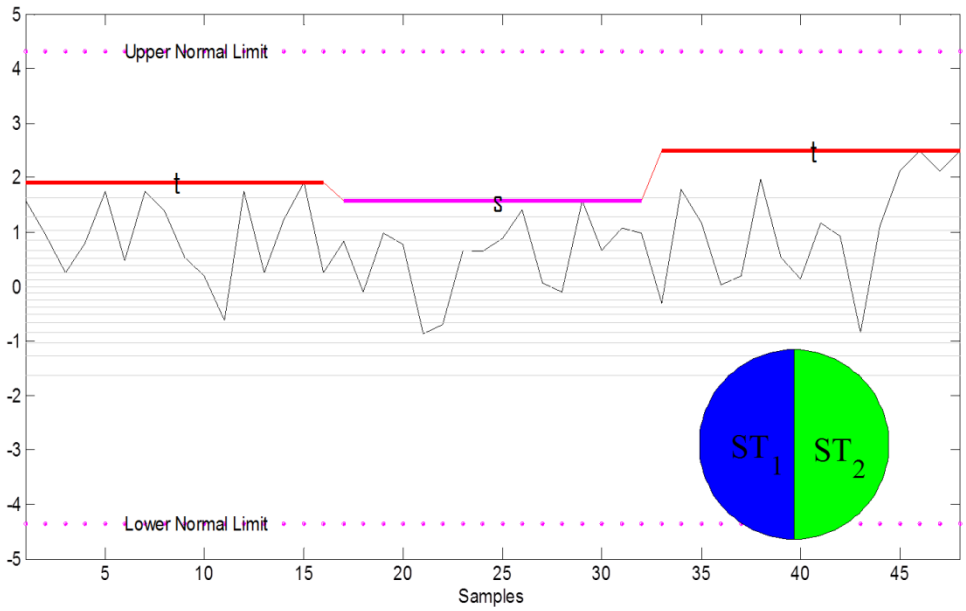


Figure 13: Time waveform (black solid line) of process variable $x_{meas}21$ and its SAX representation (red- purple solid line) for the monitoring instant No. 482.

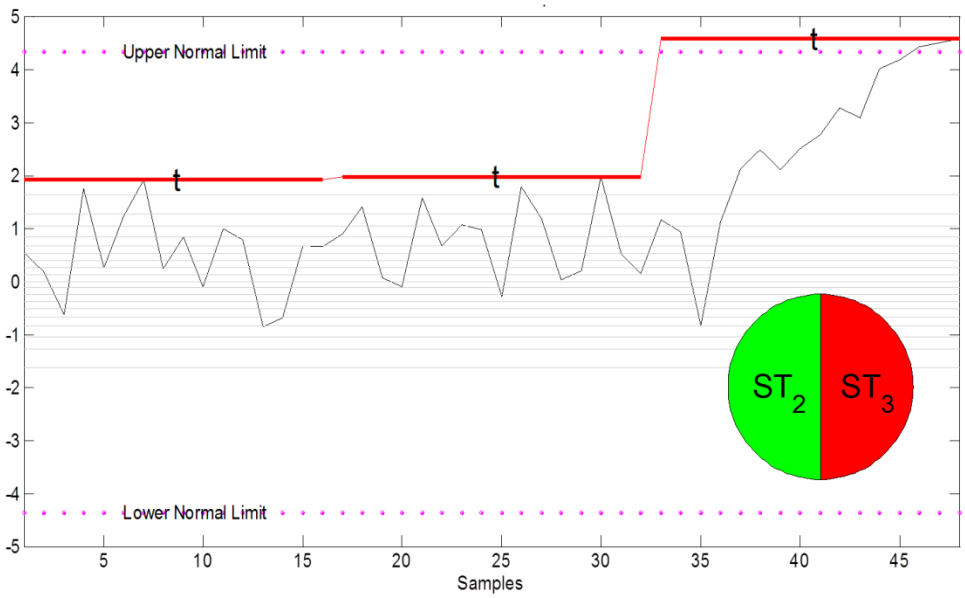


Figure 14: Time waveform (black solid line) of process variable $x_{meas}21$ and its SAX representation (red solid line) for the monitoring instant No. 483.

Moreover, the State Indicator of the variable 1 is lower (= 0.5842) than the value of $MGAL^1$ in the monitoring instant No. 488. As a consequence, a normal behavior of the process variable 1 is indicated at this monitoring instant.

Additionally, in order to further evaluate the effectiveness of the proposed method, the process variable 9 is studied. Once the abnormal event starts (No. 3,900 measurement sample), a random variation is introduced in the variable 9 (Fig. 20(b)). As illustrated in the Fig. 20(b), this fault does not produce significant deviations in the measured variable 9. The changes are compensated by the control loops keeping the normal operating behavior of the process variable 9. As shown

in Fig. 20(a), the State Indicator STI^9 of the variable 9 is lower than the $MGAL^9$ for the entire process providing excellent diagnosis performance in real time.
Furthermore, in this work, the results presented at Fig. 12 -20 are used to shape the pattern of the radar plot corresponding to the overall process operating condition at each monitoring instant.

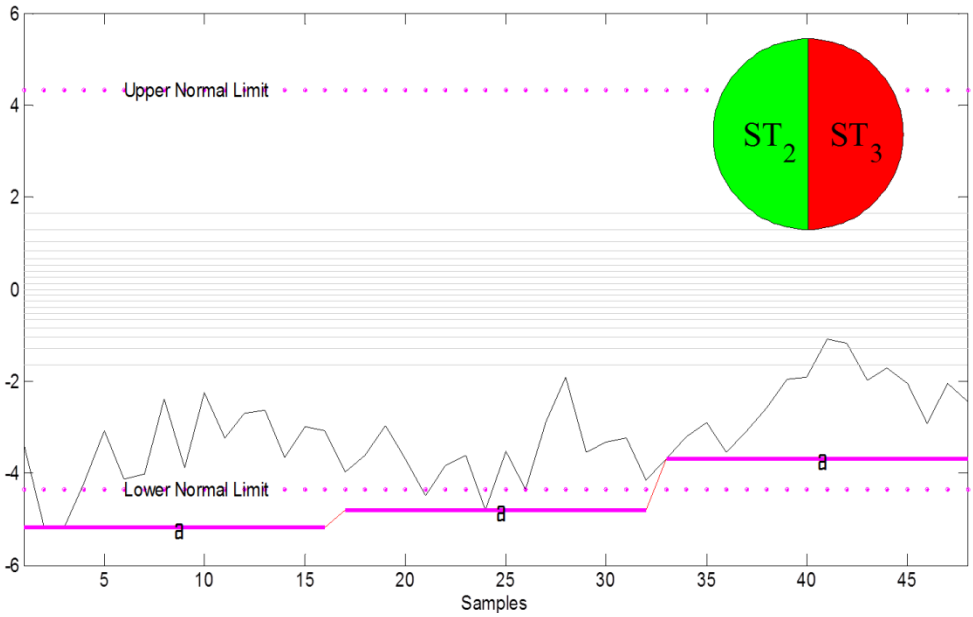


Figure 15: Time waveform (black solid line) of process variable 21 and its SAX representation (purple solid line) for monitoring instant No. 682.

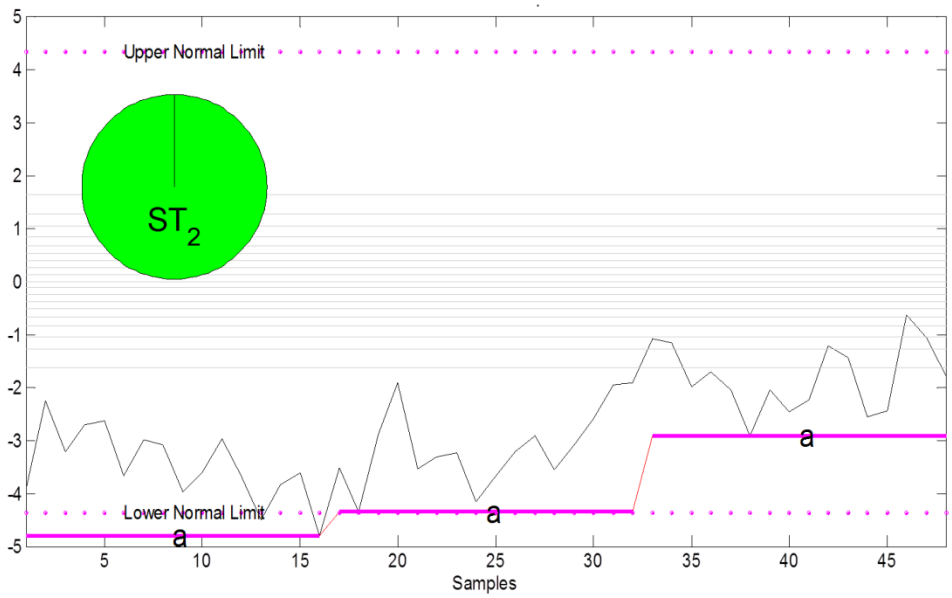


Figure 16: Time waveform (black solid line) of process variable 21 and its SAX representation (purple solid line) for monitoring instant No. 683.

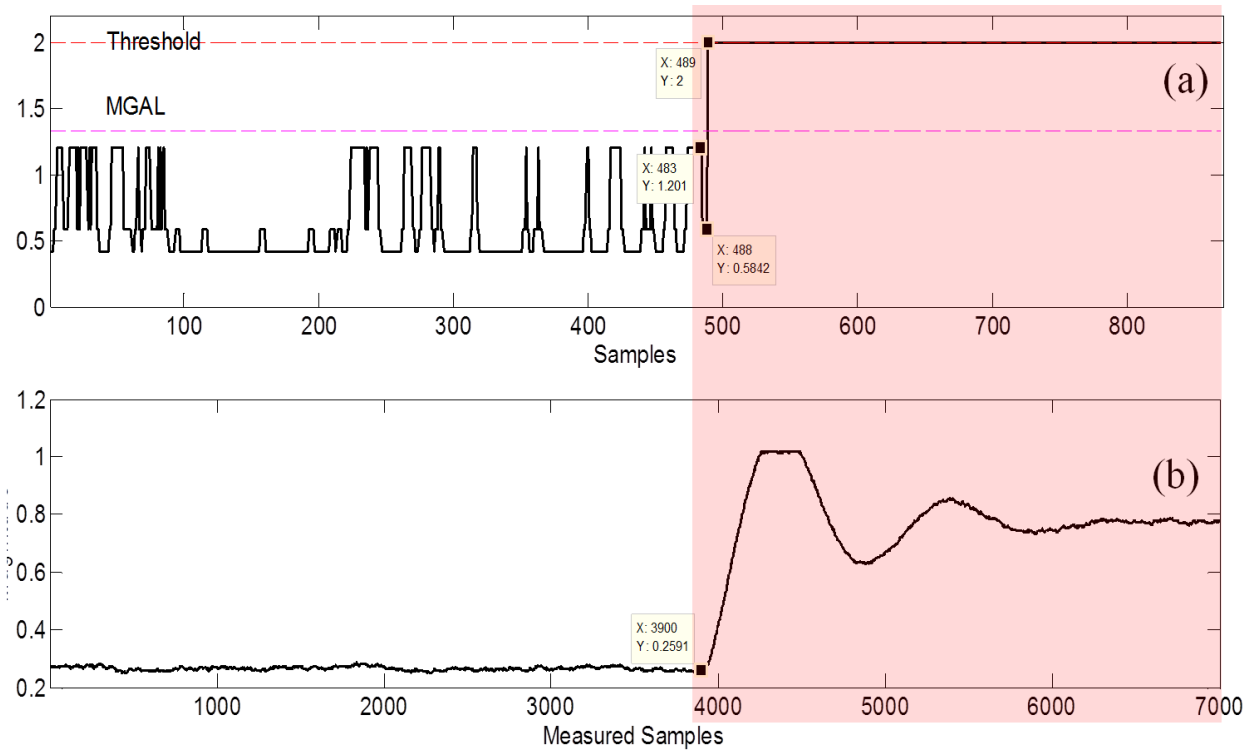


Figure 17: (a) STI trajectory and (b) measurements of the process variable 1 for the entire process of case A.

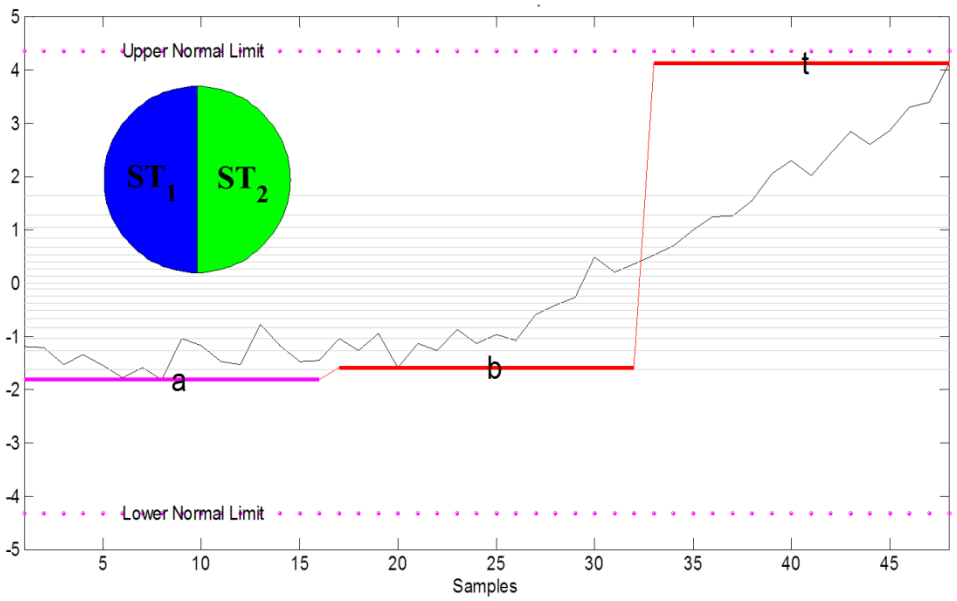


Figure 18: Time waveform (black solid line) of process variable 1 and its SAX representation (red-purple solid line) for monitoring instant No. 488.

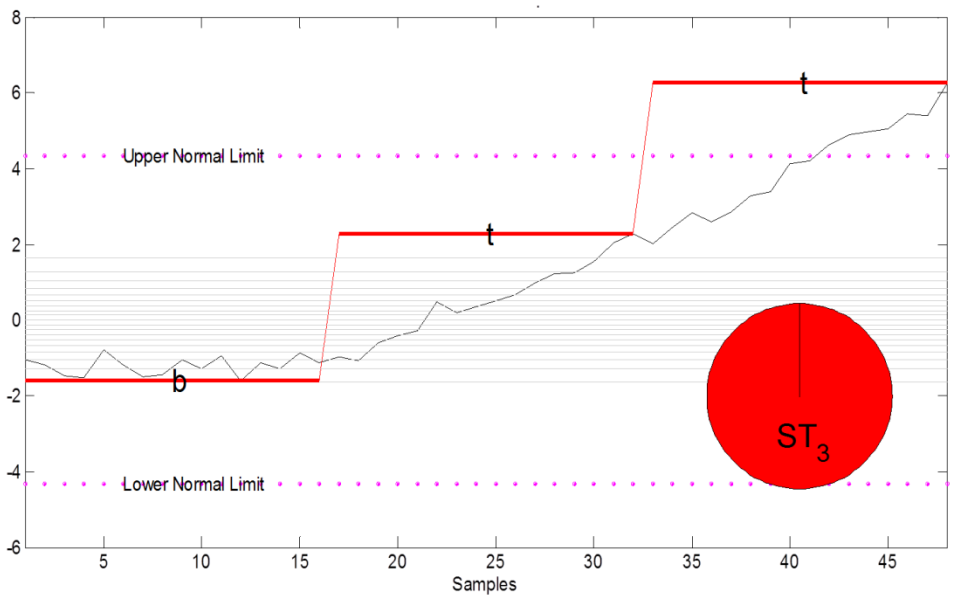


Figure 19: Time waveform (black solid line) of process variable 1 and its SAX representation (red solid line) for monitoring instant No. 489.

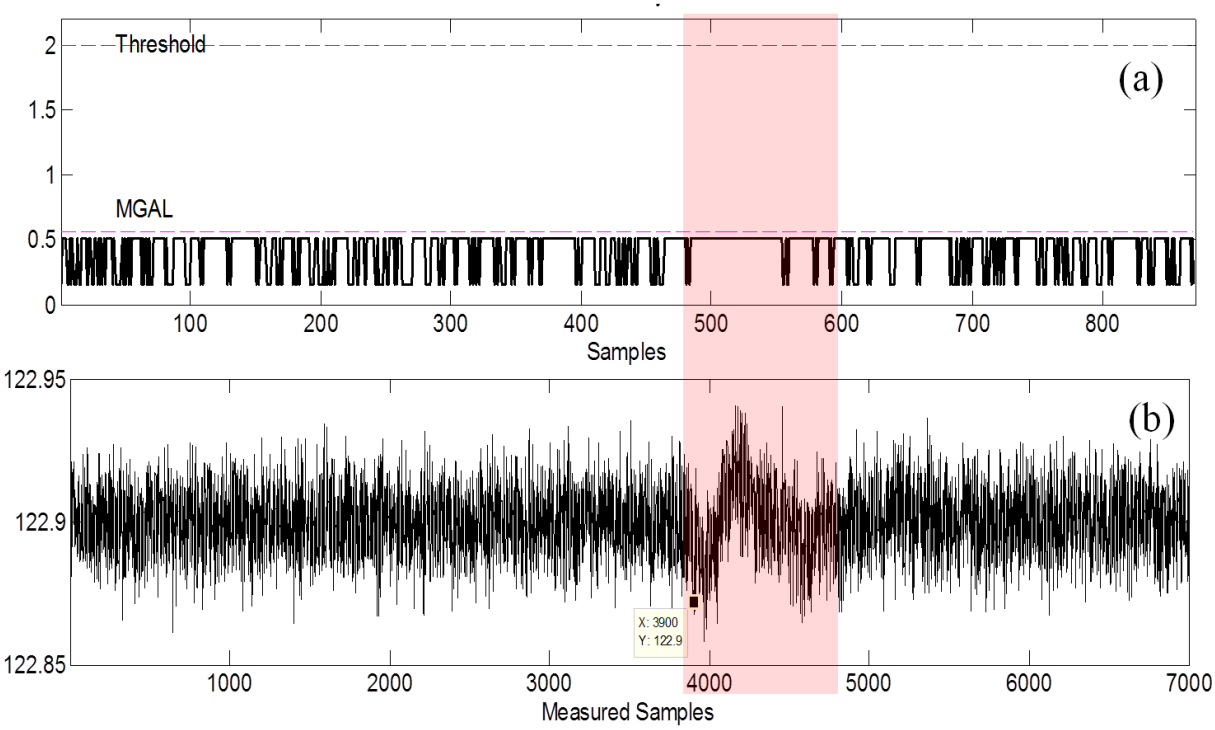


Figure 20: (a) STI trajectory and (b) measurements of the process variable 9 for the entire process of case A.

5.2.2. Case study B on Fault 19

The proposed monitoring approach is further applied to detect different scenarios of process faults. Therefore, fault 19, which is an unknown disturbance in the *TE* process, is employed and the monitoring results are presented and discussed. The fault in testing case B has been

introduced in the sample No. 3,725 and it is marked in the Figs. 22(b), 25(b), 26(b), 27(b) and 28(b).

In Fig. 21, a series of results of the proposed visualization tool at different monitoring instants of the process are illustrated. As it can be observed, the continuous process variable 14 is the first variable that responds to the fault occurrence (Fig. 21(c)). Most of the measurement and manipulated variables keep their normal operating behavior based on the reaction or not of the controllers for the entire process. Similarly to the testing case A, the panels of Fig. 21 present some updates of the produced radar plot due to the process progression that deteriorates over time.

Fig. 22 presents the entire trajectory of the measured variable 14 according to the *TEP* model. As shown in Fig. 22(b), the trajectory of the process variable 14 presents distinct peaks dealing with the existence of the disturbance 19. Meanwhile, it can be observed from Fig. 22(a) that the State Indicator STI^{14} of variable 14 indicates an obvious difference between normal and faulty conditions after the monitoring instant No. 461. However, due to the mechanism of the fault 19, the STI^{14} remains higher than the $MGAL^{14}$ (= 0.6018 (purple dash line in Fig. 22(a)) for a few too short time periods.

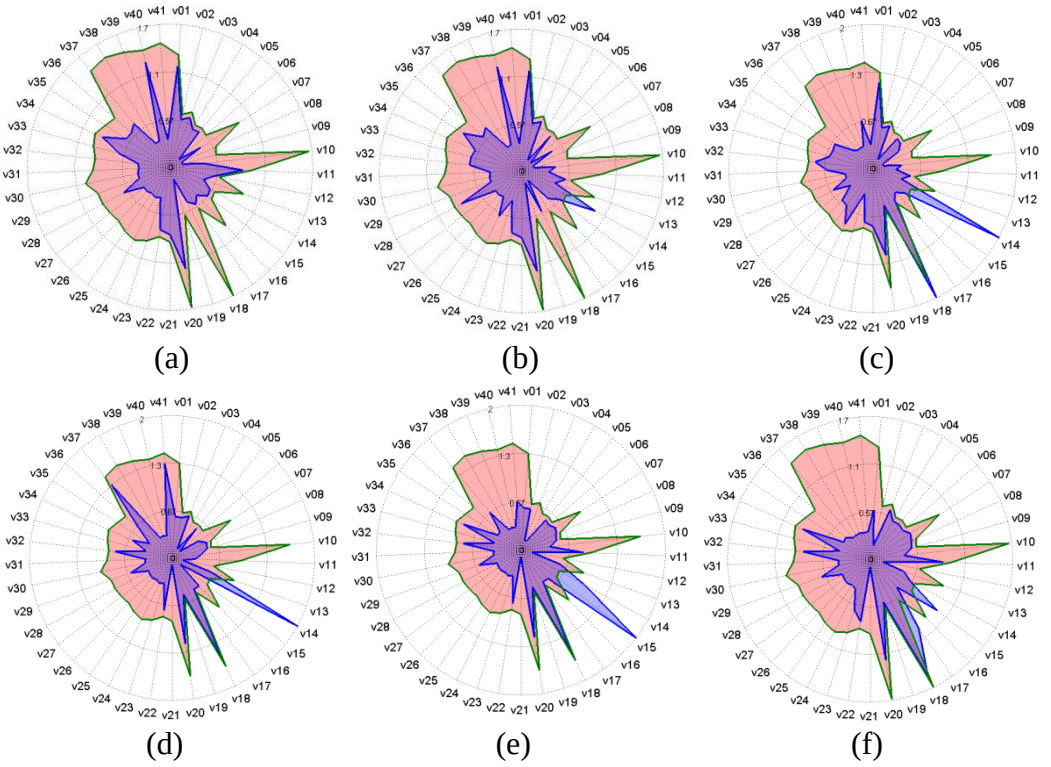


Figure 21: Visualization of the process corresponding to the case B for the monitoring instants: (a) No. 460, (b) No. 461, (c) No. 463, (d) No. 584, (e) No. 586 and (f) No. 587.

In Fig. 24, the *SAX* representation of the testing time series in monitoring instant No. 461 is presented. The measured samples 3,681 - 3,728 of the variable 14 correspond to the monitoring time instant No. 461. The last one *SAX*-based value lies below the lower normal limit, while the other *SAX*-based values move inside the estimated normal limits.

Then, according to the corresponding transition matrix

$$TM^{14/461} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.5 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (12)$$

the elements $TM^{14/461}{}_{(2,2)}$ and $TM^{14/461}{}_{(2,3)}$ indicate that during the monitoring instant No. 461 the process variable 14 remains temporally to the *still normal state* 2 and afterwards switches to the *abnormal state 3*.

Then, the State Indicator STI^{14} is calculated (= 1.0) through the ΔPV distance ($=|PV^{14/461}{}_{test} - PV^{14}{}_{norm}|$, *where* $PV^{14}{}_{norm}$=[0.4295 0.5705 0.0] and $PV^{14/461}{}_{test}$ =[0.0 1.0 0.0] (pie chart in Fig. 23)). As it can be observed in Fig. 22(b), the fault 19 has a direct influence on the variable 14. At Fig. 22 (a) it can be noted that the calculated STI^{14} obviously achieves to raise an alarm as soon as a fault occurs since it can figure out the changes of the variable. In other words, the fault is rapidly detected so there is no missed alarm and the proposed monitoring tool achieves to notify the operators exactly at the time instant when the variable 14 deviates from its normal behavior.

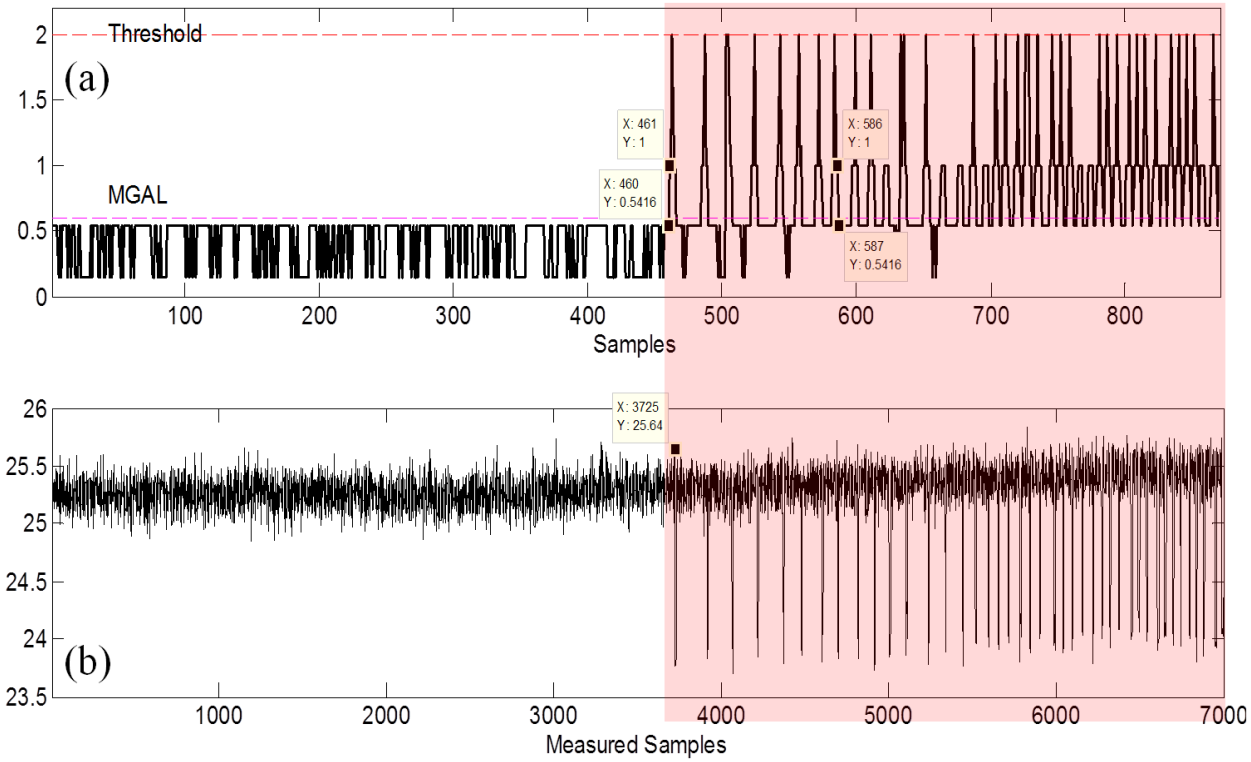


Figure 22: (a) STI trajectory and (b) measurements of the process variable 14 for the entire process of case B.

Next, the monitoring approach is tested on the previous monitoring instant No. 460. In the Fig. 23, the *SAX* result for the measured samples 3,673 - 3,720 of the variable 14 is presented. All the *SAX*-based values lie in *normal state 2* resulting through *HMM* to the transition matrix

$$TM^{14/460} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (13)$$

and the estimated probability vector $PV^{14/460}_{test}$ = [0.0 1.0 0.0] (pie chart in Fig. 23). The non-zero element $TM^{14/460}_{(2,2)}$ indicates that the process variable 14 remains to *normal state*. Additionally, as shown in Fig. 22(a), the State Indicator STI^{14} (= 0.5416) of variable 14 in the monitoring instant No. 460 is lower than the $MGAL^{14}$. Thus, in this particular monitoring instant, the STI^{14} doesn't exceed the normal limits of the monitoring chart in line with the normal operating behavior of the variable 14.

In order to illustrate the on-line overall performance of the visualization tool through the simultaneous control of several variables, the method is applied indicatively to the process variables 1, 5, 10 and 17.

As shown in Fig. 21(b), the panel of the radar plot indicates that this process has no abnormal behavior in monitoring instant No. 460. This monitoring result is established and confirmed by the Figs. 22(a), 25(a), 26(a), 27(a) and 28(a).

General speaking, a conclusion that can be drawn from the above figures is that the fault 19 has a direct influence on the variable14 (Fig. 22(a)), a delayed influence on the variable 17 (Fig. 28(a)), no influence on variables 1 (Fig. 25(a)) and 5 (Fig. 26(a)), and a delayed influence without impact on variable 10 (Fig. 27(a)) as a result of the controllers' reaction. Figs. 22, 25, 26, 27 and 28 confirm the ability of the State Indicator *STI* of each variable to adapt immediately to the current operating condition. Further, these individual results are aggregated on a radar plot offering a monitoring of the entire industrial process.

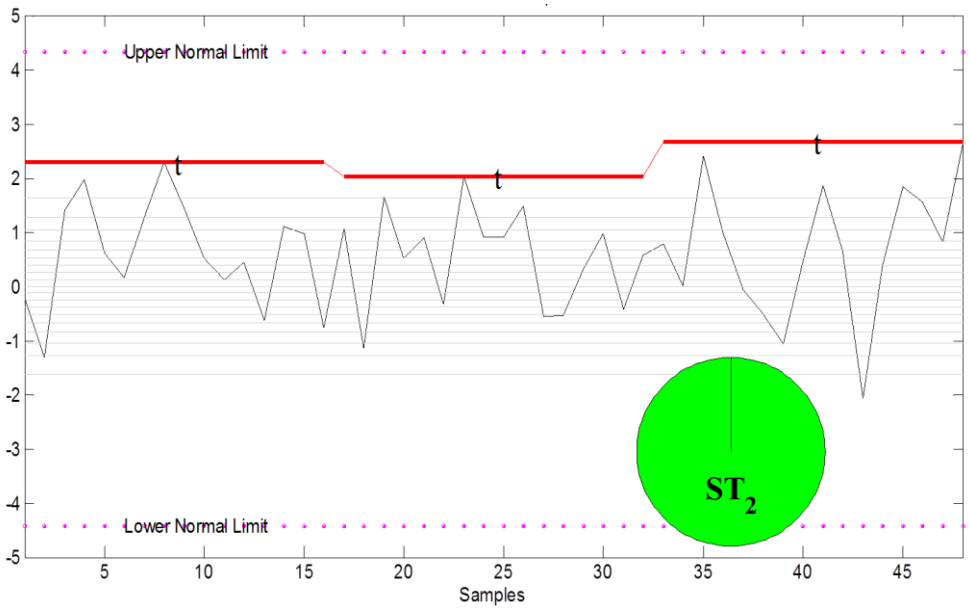


Figure 23: Time waveform (black solid line) of process variable 14 and its SAX representation (red solid line) for monitoring instant No. 460.

In Fig. 28, it can be observed that the disturbance 19 has a long delayed influence on the variable 17 with a long delay of over 2 hrs and 9 min. Moreover, this fault seems to have a non-continuous impact on variable 19. After performing an analysis similar to the previous examples, the state indicator STI^{17} is detected below the $MGAL^{17}$ (=0.5918, purple dash line in Fig. 28(a)) before the fault occurs.

After the delayed impact on the variable 17, the State Indicator STI^{17} responds immediately to the fault and yields an out-of-control signal. As it is observed in Fig. 28(a), the STI^{17} produced by the ΔPV distance is higher (= 1.0) than the $MGAL^{17}$ after the monitoring instant No. 587.

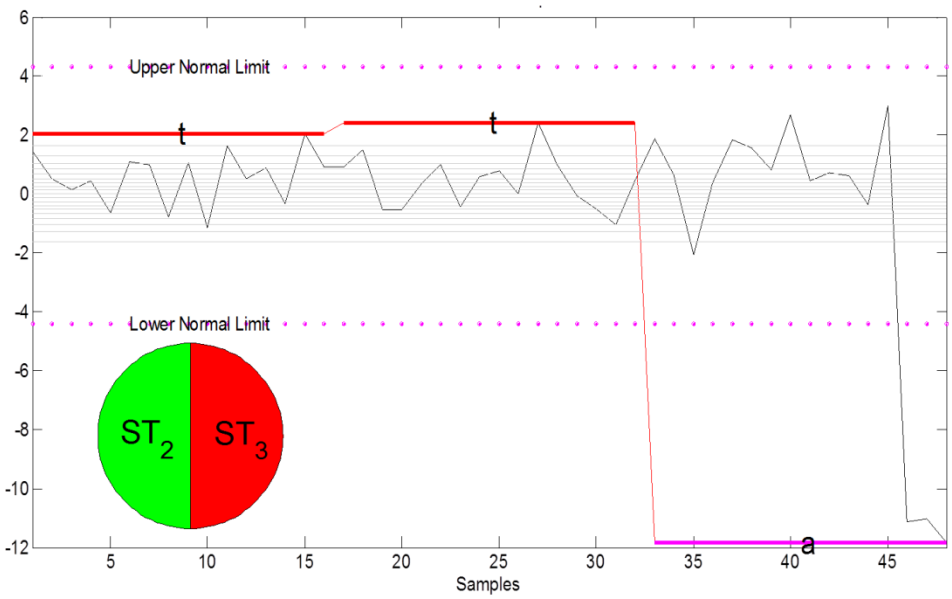


Figure 24: Time waveform (black solid line) of process variable 14 and its SAX representation (red-purple solid line) for monitoring instant No. 461.

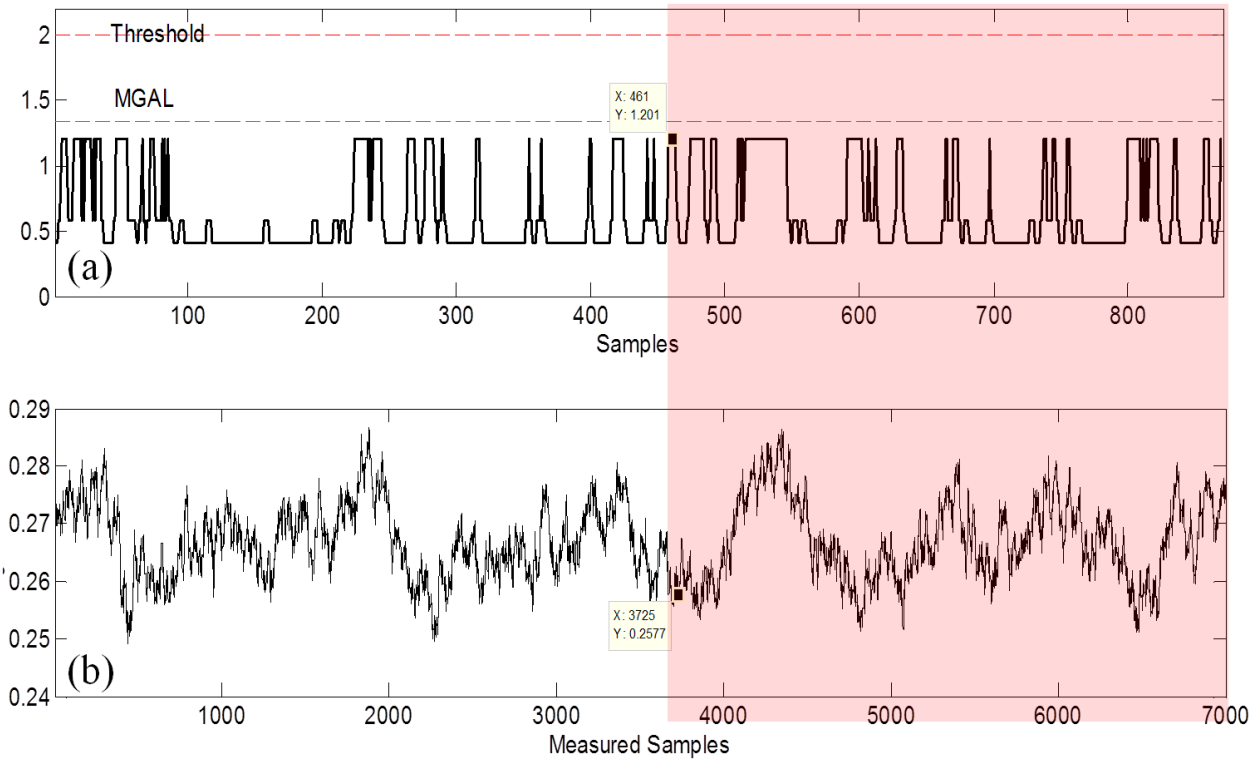


Figure 25: (a) STI trajectory and (b) measurements of the process variable 1 for the entire process of case B.

The *SAX*-based sample of the test time series that consists of the samples 4,689 - 4,736 corresponding to the monitoring instant No. 587 is illustrated in Fig. 30. The first two *SAX*-based values lie in *normal state*, while the last one is observed below the lower normal limit. Then, the corresponding transition matrix

$$TM^{17/587} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.5 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (14)$$

results to the estimated probability vector $PV^{17/587}{}_{test}$ = [0.0 0.5 0.5] (red-green pie chart (Fig. 30)) meaning that the state behavior of the variable in the monitoring instant No. 587 corresponds to the *abnormal state 3*.

The *SAX* representation of the monitoring instant No. 586 is analyzed and presented in Fig. 29. All the samples are located inside the normal limits signifying that till this monitoring instant the variable 17 doesn't respond to the fault occurrence. The corresponding transition matrix represents this behavior.

$$TM^{17/586} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (15)$$

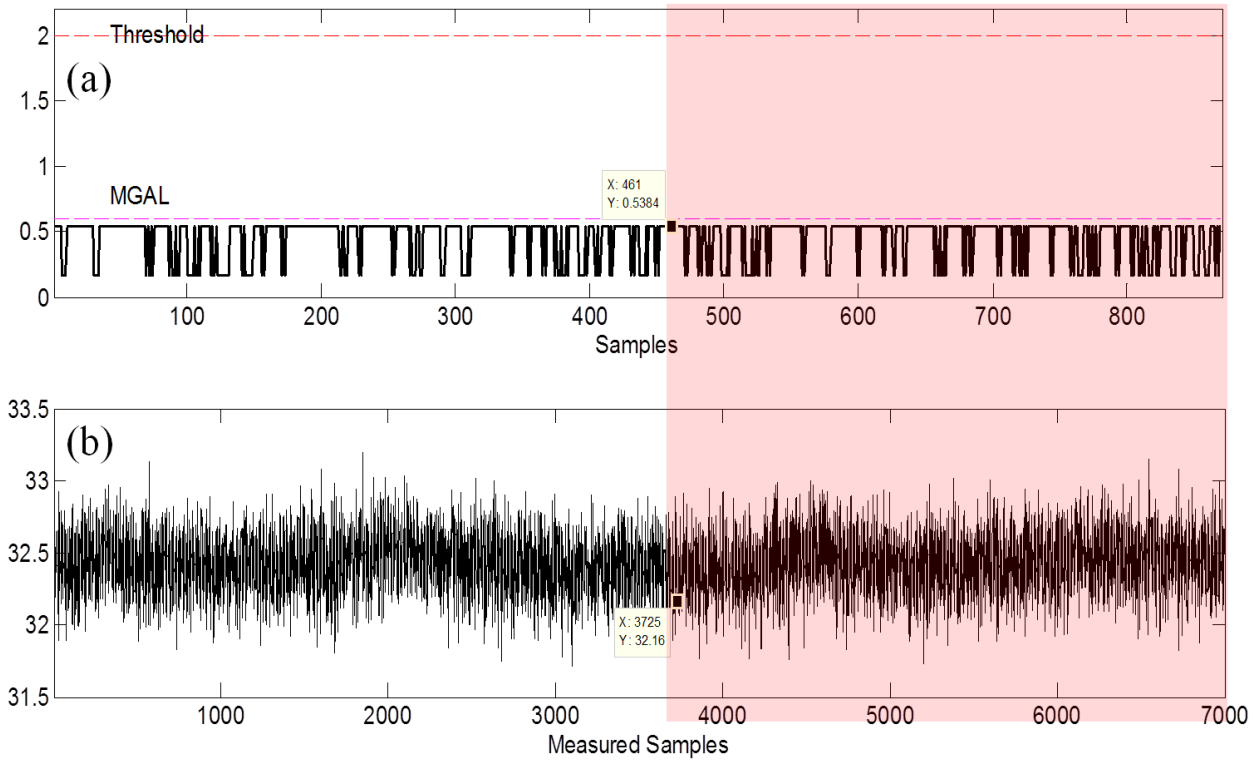


Figure 26: (a) STI trajectory and (b) measurements of the process variable 5 for the entire process of case B.

The proposed monitoring strategy initiates out-of-control signals immediately when the process fault occurs and affects the variable. These superior monitoring results demonstrate that *SAX* and *HMM*-based monitoring approach can effectively capture the dynamic changes of variables. Additionally, as shown in Fig. 11 and Fig. 21, the proposed visualization tool forms different patterns throughout the process dealing with a specific process disturbance. This could possibly be used to identify which abnormal situation has occurred and if it has also occurred in the past.

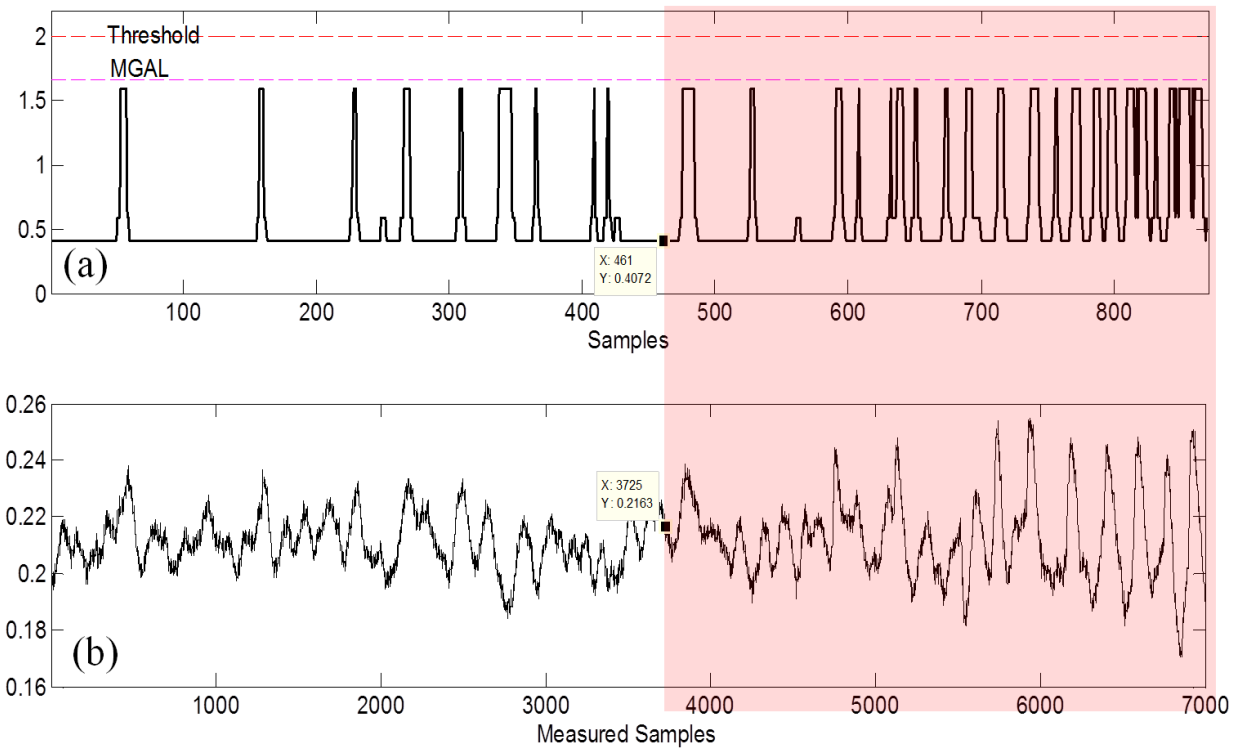


Figure 27: (a) STI trajectory and (b) measurements of the process variable 10 for the entire process of case B.

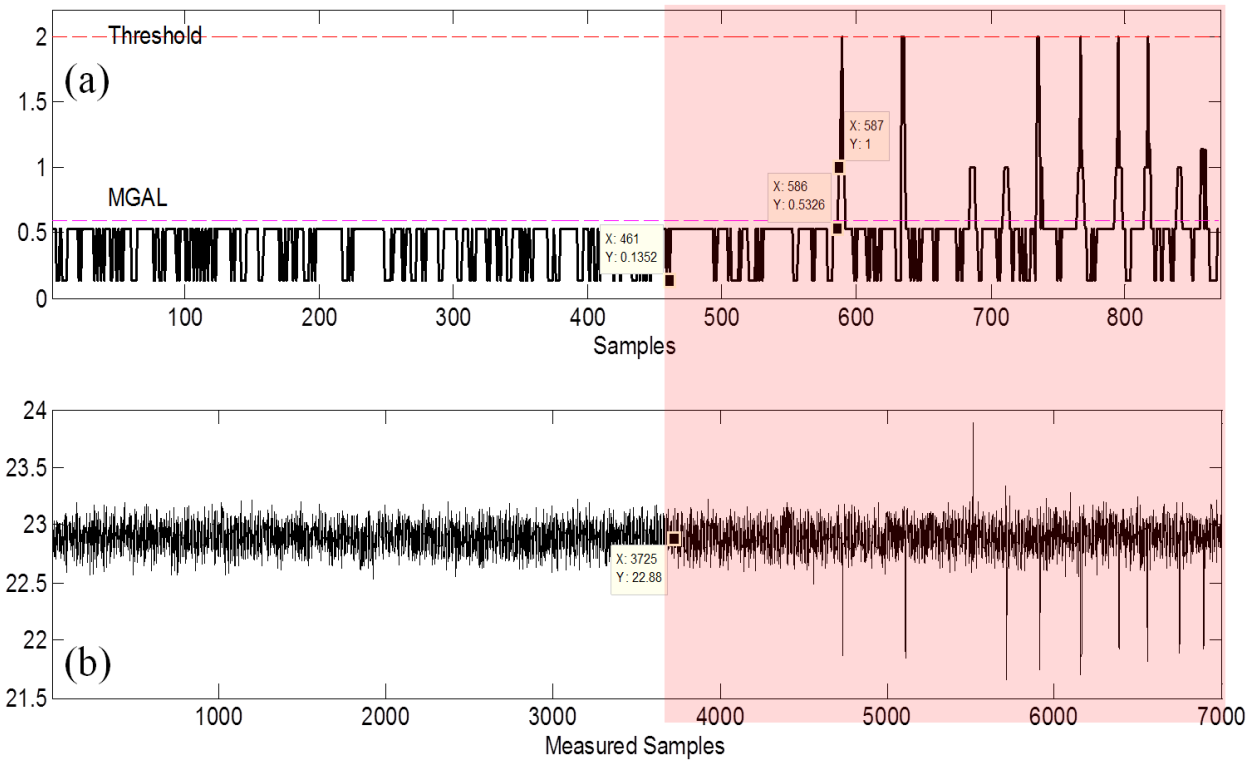


Figure 28: (a) STI trajectory and (b) measurements of the process variable 17 for the entire process of case B.

However, there are still some open questions that should be considered in a future work. Foremost is the selection of the number of the relevant *SAX*-based samples. Thus, the appropriate values of the key parameters should be determined by the operators in accordance with the requirements of the application as none of them have a priori optimal value. The developed strategy can offer better performance of the monitoring scheme, especially for maximization of production rate.

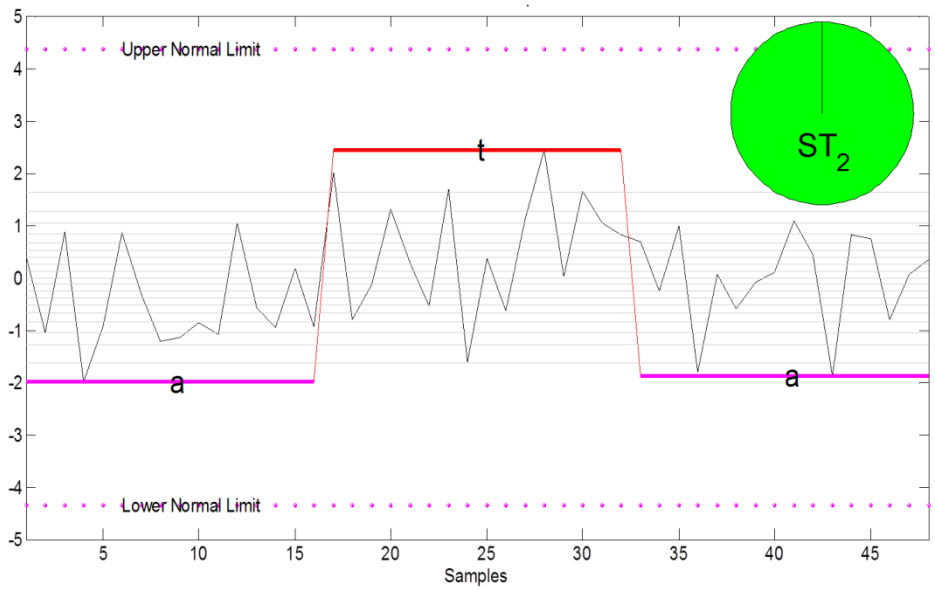


Figure 29: Time waveform (black solid line) of process variable 17 and its SAX representation (red-purple solid line) for monitoring instant No. 586.

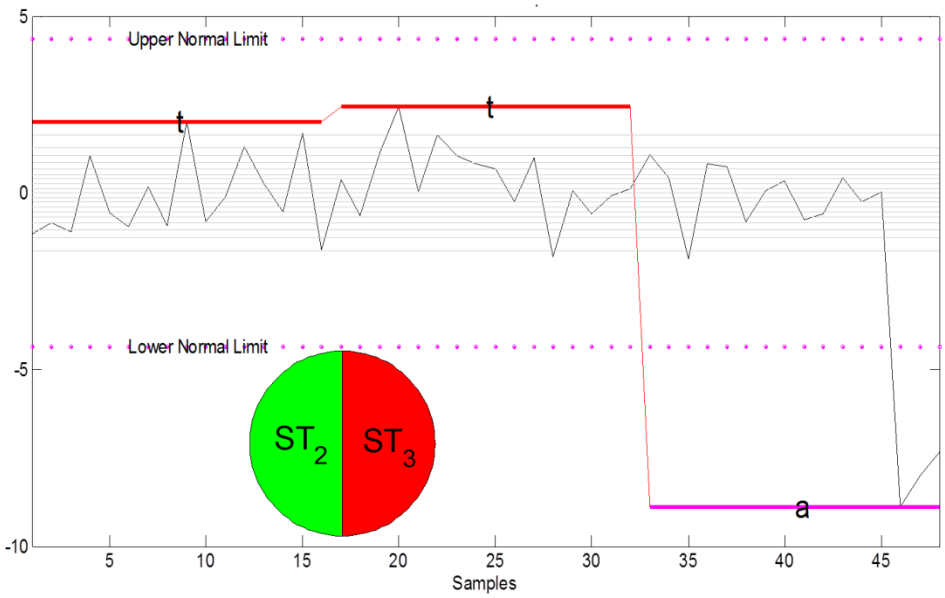


Figure 30: Time waveform (black solid line) of process variable 17 and its SAX representation (red-purple solid line) for monitoring instant No. 587.

6. Comparison

In this section, the proposed process monitoring method is compared and contrasted through application to the TEP with the visualization technique VizTree [21] and the PCA [44] and CVA [44] statistic models.

Firstly, the visualization proficiency of SAX-HMM based method, which allows users to visually explore the time series and perform real-time hypotheses testing in order to perform fault detection, is evaluated using as case studies the variables No 1, 4, 9, 10, 14, 21 and 38 from the case A and the variables No 1, 5, 10, 14 and 17 from the case B. The monitored variables deviate from their normal operating behavior with delays of varying length when the fault occurs. In order to estimate the detection delay, a reference fault start point is defined as the first waveform peak after the initialization of the abnormal operating condition that is higher than the highest peak of the normal data. For instance, as presented in Fig. 31, the fault IDV01 starts developing at the sample No 3,900 , but affects the variable No 1 and results in an increase of its magnitude with a delay of 44 samples. The performance of the proposed approach is evaluated and compared with the VizTree method. Applying the VizTree technique the distributional differences of patterns between the normal training data and the testing data can be visualized. The comparison of the methods is based on the time delay detection of the abnormal operating condition and on the number of the false alarms which are noted during the normal operating condition. The performance sensitivity of the VizTree method is analysed with respect to the variation of three key parameters: (a) the sliding window length (in samples), (b) the number of symbols per window and (c) the alphabet size. The executable code of VizTree has been downloaded from [43].

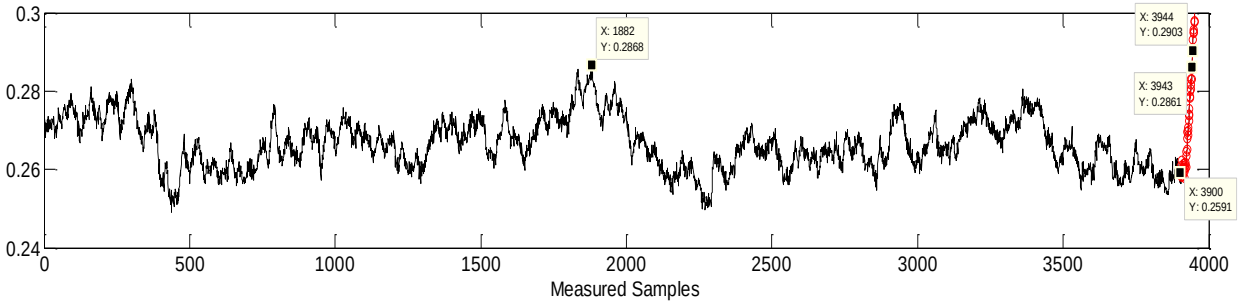


Figure 31: Time waveform of the variable No 1 (case A: fault IDV1). For the references to colors in the figure, the reader can be referred to the web version of this article.

Three typical results obtained by applying the VizTree method are summarized in Figs. 32, 33 and 34 respectively: a) a typical successful application, b) a failed application and c) an application with false alarms.

In Fig. 32, the blue time waveform corresponds to the reference/normal data while the green time waveform is the testing data corresponding to the variable No 14 of the IDV01 abnormal process. The length of the time series is 7,000 samples. Also, the length of the sliding window is set to 96; the number of segments and the alphabet size is 3. The resulting diff-tree illustrates the differences of the pattern distributions in the two datasets. Clicking on the branch C that is ranked #2, the abnormal process start is detected and highlighted brown in the time series

window. Similarly, the branch A that is ranked #1 corresponds to the abnormal process. Moreover, the normal operating condition is shown in the branch B that is ranked #3.

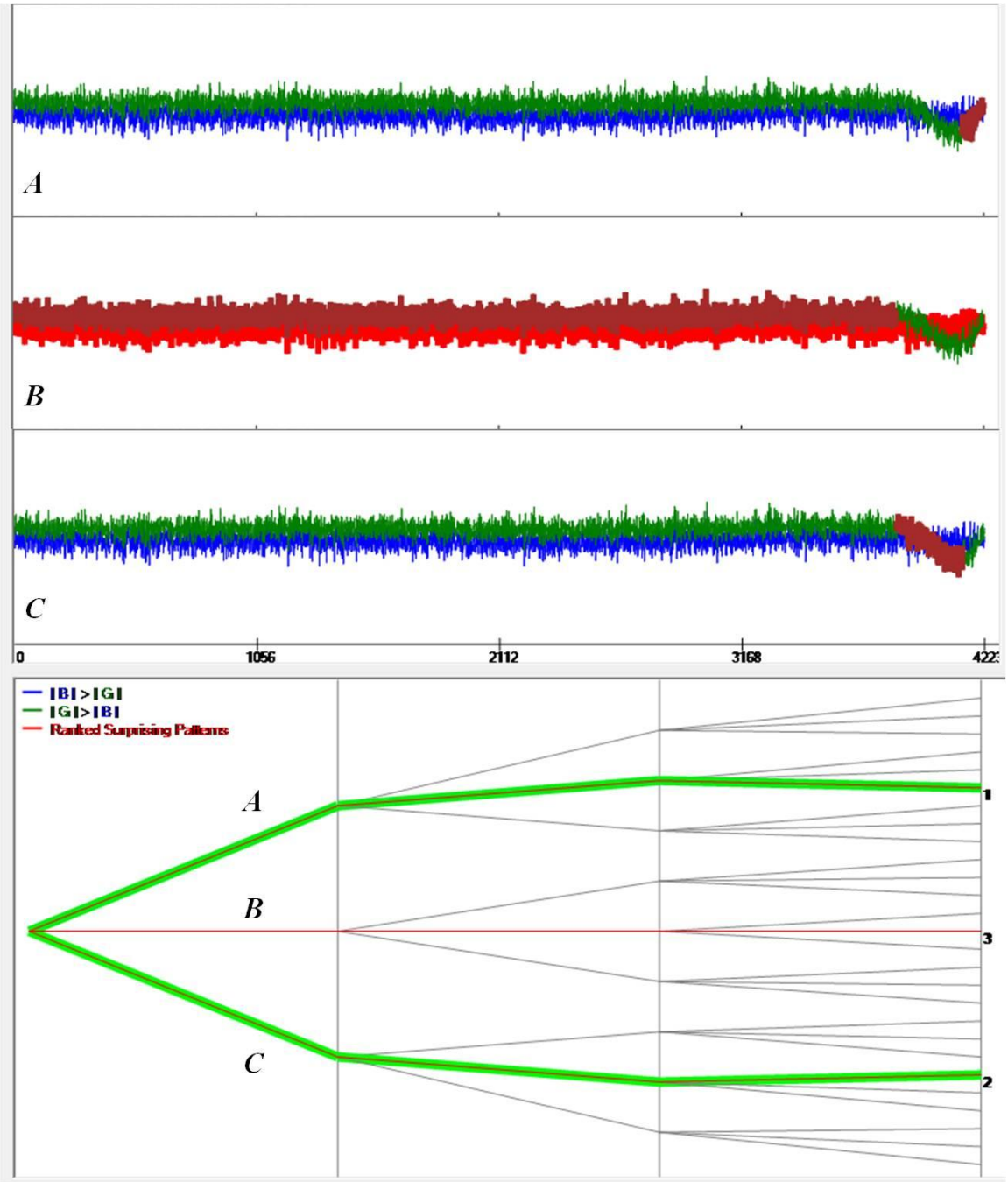


Figure 32: A successful performance of the VizTree technique for the case A (fault IDV01 and process variable No 14). For the references to colors in the figure, the reader can be referred to the web version of this article.

Next, as shown in Fig. 33, the VizTree approach completely fails to detect the abnormal start in the process variable No 14 under the fault IDV19 (case B). In this test, the length of the sliding window is set to 48; the number of segments and the alphabet size is 3.

In Fig. 34, the time series are simulations of the variable No 1 in the normal (blue) and the abnormal (green) operation IDV1 (case A). According to the Tables 5 and 11, the abnormal process IDV1 affects the variable 1 with a delay of 44 samples. As hereinbefore, the length of the sliding window is set to 96; the number of segments is 6 and the alphabet size is 3. As shown in Fig. 34, the two thick paths A and B, which are ranked #2 and #1, respectively, demonstrate three possible abnormal states. However, as it is observed in both of the top graphs in Fig. 34, the diff-tree exhibits two false alarms.

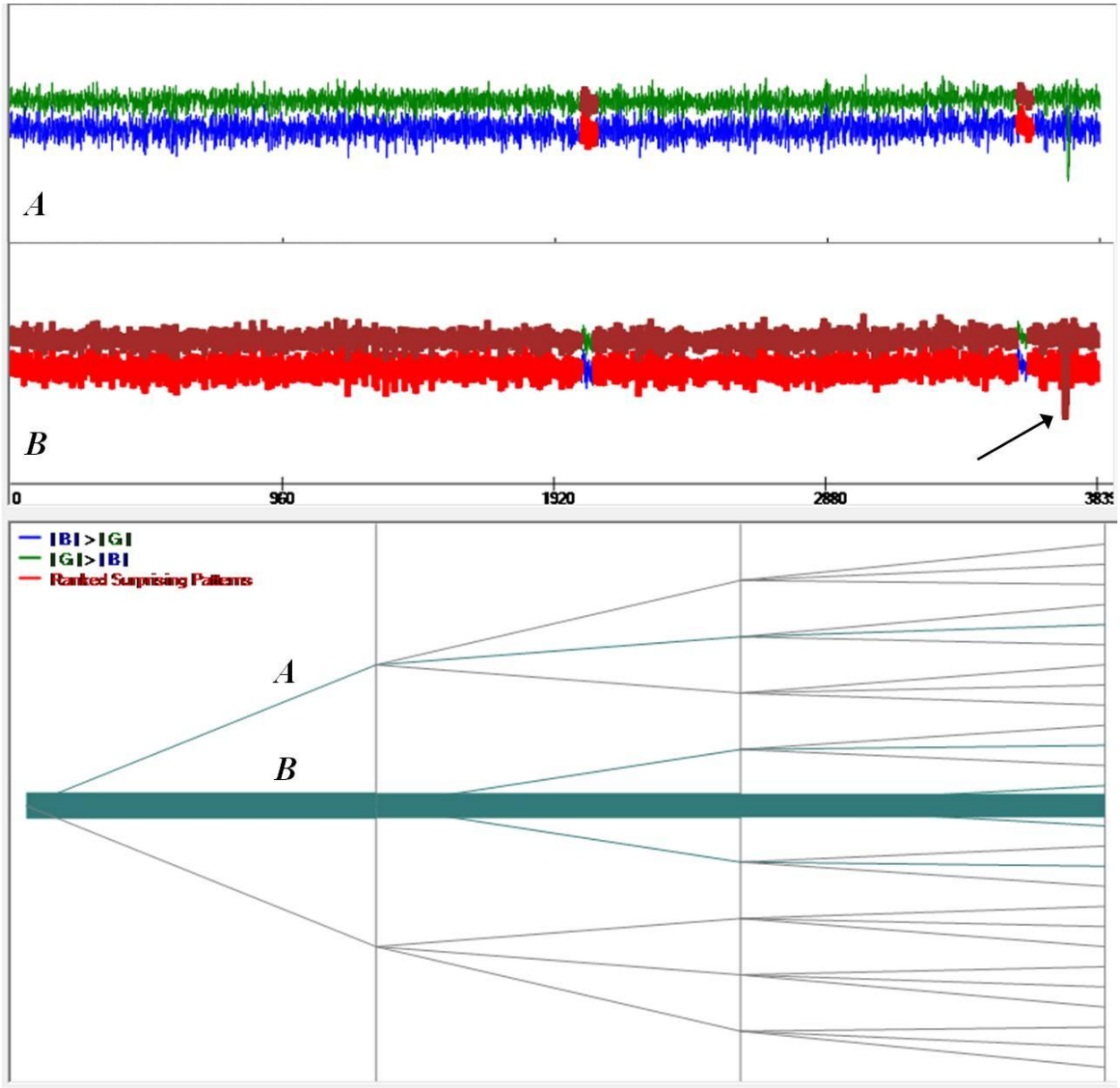


Figure 33: A failed performance of the VizTree technique for the case B (fault IDV19 and process variable No 14). For the references to colors in the figure, the reader can be referred to the web version of this article.

The outcome, the detection delay in samples and the number of false alarms of the VizTree technique are computed and presented in Tables 1 - 7 for the case A and in Tables 9 - 13 for the

case B assuming that the length of the sliding window is set to 24, 48, 96 and 192 samples; the number of segments is equal to 3, 6 and 9 and the alphabet size is 3, 6 and 9. The results indicate a timely or delayed fault detection with a 'V', while an 'X' indicates a failed fault detection. Although the VizTree is able to capture the fault, it achieves a short detection delay applying different combinations of the key parameters for each single variable or it results to a long detection delay using common values of the parameters. Moreover, the inevitable false alarms make difficult to determine whether the 'out-of-control branch' of the VizTree is a result of a fault or a false alarm. In contrast to the VizTree approach, the proposed method does not present problems with false alarms.

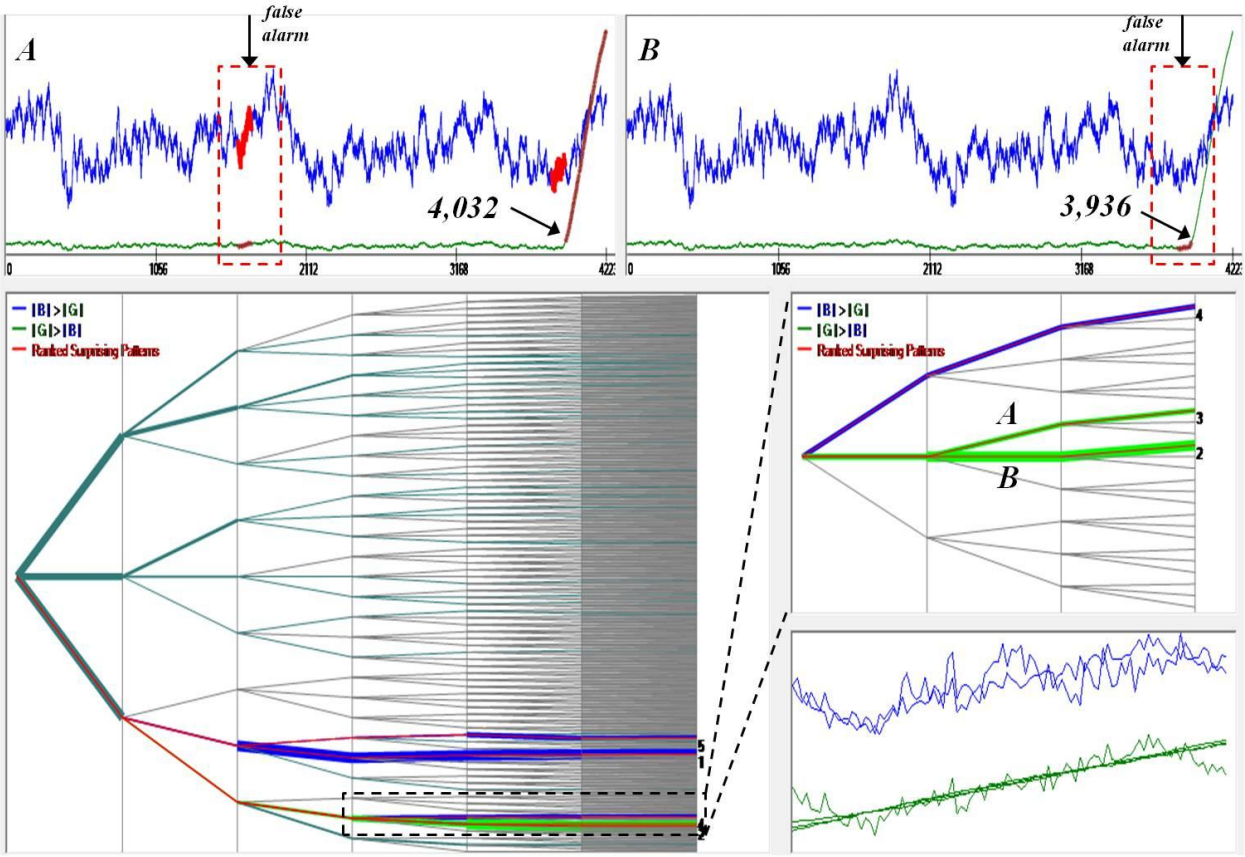


Figure 34: A performance of the VizTree technique resulting to a false alarm for the case A (fault IDV1 and process variable No 1). For the references to colors in the figure, the reader can be referred to the web version of this article.

Also, Tables 8 and 14 quantify the SAX-HMM based method performance applied on case A and B, respectively. The key parameters of the proposed technique are N_{str} = 24, 48, 96, 192, n_{shift} = 8 and n_{segt} = 3, 6, 9. The second column refers to the time instance (in samples) that the fault affects a single process variable in accordance to the 'reference fault-start point' rule. The sample number where an abnormal behavior of a single variable is detected is reported in the third column. The fault detection delays in samples is further presented in the fourth column. Finally, in the last column, the fault detection delay of the proposed method is compared against the best

performance of the VizTree, corresponding to the case with the earlier detection with the minimum number of false alarms (bold values in Tables 1 - 7 and 9 -13).
As a conclusion of the analysis and the interpretation of the Tables, the SAX-HMM based method is not prone to false alarms and is more sensitive to fault detection compared to the VizTree visualization tool in the case of a single process variable.

Table 1
Parameters and performance of VizTree for variable No 1 of the case A

Parameters	**Variable 1**																			
Sliding window length	24	24	24	24	24	48	48	48	48	48	96	96	96	96	96	**192**	192	192	192	192
# symbols per window	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9	**3**	3	3	6	9
Alphabet size	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3	**3**	6	9	3	3
Result	X	V	V	V	V	V	V	V	V	V	V	V	V	V	V	**V**	V	V	V	V
# false alarms	-	15	16	9	3	17	8	12	2	1	7	4	2	2	3	**0**	0	0	0	3
Delay S_v	-	16	16	16	16	40	40	40	40	40	88	88	88	88	88	**88**	88	88	88	88

Table 2
Parameters and performance of VizTree for variable No 4 of the case A

Parameters	**Variable 4**																			
Sliding window length	24	24	24	24	24	48	48	**48**	**48**	48	96	96	96	96	96	192	192	192	192	192
# symbols per window	3	3	3	6	9	3	3	**3**	**6**	9	3	3	3	6	9	3	3	3	6	9
Alphabet size	3	6	9	3	3	3	6	**9**	**3**	3	3	6	9	3	3	3	6	9	3	3
Result	V	V	V	V	V	V	V	**V**	**V**	V	V	V	V	V	V	V	V	V	V	V
# false alarms	10	9	4	7	5	3	1	**0**	**0**	0	0	0	0	0	2	0	0	0	0	0
Delay S_v	2	2	2	2	2	26	26	**26**	**26**	26	74	74	74	74	74	74	74	74	74	74

Table 3
Parameters and performance of VizTree for variable No 9 of the case A

Parameters	**Variable 9**																			
Sliding window length	24	24	24	24	24	48	48	48	48	48	96	96	96	96	96	192	192	192	192	192
# symbols per window	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9
Alphabet size	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3
Result	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
# false alarms	-	3	4	4	4	4	3	4	1	2	2	3	2	2	2	1	1	4	2	2
Delay S_v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Furthermore, the SAX-HMM-based approach is compared with the commonly used PCA, DPCA and CVA methods in order to detect as soon as possible an abnormal operation condition in industrial processes. Since the proposed method is not a dimensionality reduction technique such as the mentioned statistical methods, it would be instructive to compare the effectiveness of them

only in terms of the fault detection delay. Thus, the detection delays for the multivariate statistical methods PCA, DPCA and CVA are estimated through application to the TEP using the Table 5 from [34] and are listed in samples and not in minutes in Table 15 of this paper. Furthermore the detection delays for the proposed method are estimated. The fault is indicated only when 6 consecutive STI values of a single variable exceed the relative MGAL threshold, and the fault detection delay is recorded as the first time instant in which the MGAL region for the very first time is exceeded from any process variable. Also, all the data have been processed by using the following key parameters of the proposed technique: N_{str} = 24, n_{shift} = 1 and n_{segt} = 3, in order the results to be comparable with the results of the study of [34]. The detection delays for 10 different faults of TEP are listed in Table 15. In order to calculate the sample detection delay (12th column) the SAX-HMM result is compared with the best results of the statistical models (bold values). Based on the abovementioned comparison, the proposed SAX-HMM based approach provide significantly earlier fault detection in comparison to the PCA, the dynamic version of PCA and the CVA models in most of the cases.

Table 4
Parameters and performance of VizTree for variable No 10 of the case A

Parameters	**Variable 10**																			
Sliding window length	24	24	24	24	24	48	48	48	48	48	96	96	96	96	**96**	192	192	192	192	192
# symbols per window	3	3	3	6	9	3	3	3	6	9	3	3	3	6	**9**	3	3	3	6	9
Alphabet size	3	6	9	3	3	3	6	9	3	3	3	6	9	3	**3**	3	6	9	3	3
Result	V	V	V	V	V	V	X	V	X	V	V	V	V	V	**V**	V	V	V	V	V
# false alarms	1	8	9	28	16	2	4	2	6	0	2	4	1	1	**0**	2	1	1	0	0
Delay S_v	138	18	18	18	18	162	-	18	-	138	18	18	18	114	**18**	114	114	114	114	114

Table 5
Parameters and performance of VizTree for variable No 14 of the case A

Parameters	**Variable 14**																			
Sliding window length	24	24	24	24	24	48	48	48	48	48	**96**	96	96	96	96	192	192	192	192	192
# symbols per window	3	3	3	6	9	3	3	3	6	9	**3**	3	3	6	9	3	3	3	6	9
Alphabet size	3	6	9	3	3	3	6	9	3	3	**3**	6	9	3	3	3	6	9	3	3
Result	X	X	V	V	V	V	V	V	V	V	**V**	V	V	V	V	V	V	V	V	V
# false alarms	7	-	2	1	0	5	5	4	3	3	**0**	0	0	1	1	0	0	0	0	0
Delay S_v	-	-	28	28	28	4	4	4	4	4	**4**	4	4	4	100	100	100	100	100	100

Table 6
Parameters and performance of VizTree for variable No 21 of the case A

Parameters	**Variable 21**																			
Sliding window length	24	24	24	24	24	48	48	48	48	48	**96**	96	96	96	96	192	192	192	192	192
# symbols per window	3	3	3	6	9	3	3	3	6	9	**3**	3	3	6	9	3	3	3	6	9
Alphabet size	3	6	9	3	3	3	6	9	3	3	**3**	6	9	3	3	3	6	9	3	3
Result	X	V	V	V	V	V	V	V	V	V	**V**	V	V	V	V	V	V	V	V	V
# false alarms	13	3	2	3	4	6	2	1	1	1	**0**	0	0	0	0	1	0	0	0	0
Delay S_v	-	12	12	12	12	36	36	36	36	36	**36**	36	36	36	36	132	132	132	132	132

Table 7
Parameters and performance of VizTree for variable No 38 of the case A

Parameters	**Variable 38**																			
Sliding window length	24	24	24	24	24	48	48	48	48	**48**	96	96	96	96	96	192	192	192	192	192
# symbols per window	3	3	3	6	9	3	3	3	6	**9**	3	3	3	6	9	3	3	3	6	9
Alphabet size	3	6	9	3	3	3	6	9	3	**3**	3	6	9	3	3	3	6	9	3	3
Result	X	X	X	X	X	X	V	V	V	**V**	V	V	V	V	V	V	V	V	V	V
# false alarms	-	4	4	-	-	-	3	1	3	**0**	6	6	2	5	2	1	1	1	1	1
Delay S_v	-	-	-	-	-	-	127	31	127	**127**	31	31	31	31	31	31	31	31	31	31

Table 8
SAX-HMM based method performance for case A

Variable	**1st possible faulty peak (S_1 samples)**	**Fault detection (S_2 samples)**	**Delay ($S=S_2-S_1$ samples)**	**Difference with VizTree ($\Delta S=S-S_v$ samples)**
1	3,944	3,952	8	-80
4	3,958	3,984	26	0
9	-	-	-	-
10	3,918	3,936	18	0
14	3,932	4,024	92	+88
21	**3,900**	**3,904**	**4**	**-32**
38	4,001	4,056	55	-72

Table 9
Parameters and performance of VizTree for variable No 1 of the case B

Parameters	**Variable 1**																			
Sliding window length	**24**	24	24	24	24	48	48	48	48	48	96	96	96	96	96	192	192	192	192	192
# symbols per window	**3**	3	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9
Alphabet size	**3**	6	9	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3
Result	**V**	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
# false alarms	**0**	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	1	1
Delay S_v	**-**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10
Parameters and performance of VizTree for variable No 5 of the case B

Parameters	**Variable 5**																			
Sliding window length	**24**	24	24	24	24	48	48	48	48	48	96	96	96	96	96	192	192	192	192	192
# symbols per window	**3**	3	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9
Alphabet size	**3**	6	9	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3
Result	**V**	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
# false alarms	**0**	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Delay S_v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 11
Parameters and performance of VizTree for variable No 10 of the case B

Parameters	**Variable 10**																			
Sliding window length	24	24	24	24	24	48	48	48	**48**	48	96	96	96	96	96	192	192	192	192	192
# symbols per window	3	3	3	6	9	3	3	3	**6**	9	3	3	3	6	9	3	3	3	6	9
Alphabet size	3	6	9	3	3	3	6	9	**3**	3	3	6	9	3	3	3	6	9	3	3
Result	V	V	V	V	V	V	V	V	**V**	V	V	V	V	V	V	V	V	V	V	V
# false alarms	3	6	12	6	6	5	5	4	**0**	1	9	2	1	1	1	0	0	0	1	2
Delay S_v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 12
Parameters and performance of VizTree for variable No 14 of the case B

Parameters	**Variable 14**																			
Sliding window length		24	**24**	24	24	48	48	48	48	48	96	96	96	96	96	192	192	192	192	192
# symbols per window	3	3	**3**	6	9	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9
Alphabet size	3	6	**9**	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3
Result	V	V	**V**	V	V	X	X	X	V	X	X	V	V	V	V	X	X	V	V	V
# false alarms	8	7	**0**	1	6	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Delay S_v	18	18	**18**	18	18	-	-	-	18	-	-	18	18	18	18	-	-	114	114	114

7. Conclusion

Symbolic Aggregate Approximation *(SAX)* is a well-established technique, transforming a continuous time series to a discrete one. The main reason that renders this technique attractive is the fact that the symbolic representation can be set in the form of letters of an alphabet, as well as a Web available tool for the visualization and data mining of time series. However, direct usage of this technique in anomaly detection and data mining of time series can only have a limited success, since most of the relevant faults are both very rare and unique.
In this work, the theoretical basis for a novel approach is set. According to this approach, *SAX* is used firstly in order to typify the status of the plant in a limited number of two or three states,

representing a normal or abnormal situation. Then, based on the theory of Hidden Markov Models, two quantitative indices are derived for each process variable:

- A State Indicator (*STI*), indicating the quantitative distance of the process variable from an abnormal state.
- A Markov Generated Alarm Limit *(MGAL)*, in order to detect and separate normal and abnormal behavior of a process variable.

Application of this approach has demonstrated that *STI* and *MGAL* can in general accurately track the normal/abnormal behavior of a single process variable, based on training using only normal data.

Table 13
Parameters and performance of VizTree for variable No 17 of the case B

Parameters	**Variable 17**																			
Sliding window length	24	24	24	24	24	48	48	48	48	48	96	96	96	96	96	192	192	192	192	**192**
# symbols per window	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3	3	6	**9**
Alphabet size	3	6	9	3	3	3	6	9	3	3	3	6	9	3	3	3	6	9	3	**3**
Result	V	V	V	V	V	X	X	X	V	X	X	X	X	V	V	X	X	V	X	**V**
# false alarms	14	7	2	3	9	2	2	2	9	7	0	0	0	3	4	0	0	1	1	**0**
Delay S_v	18	18	18	18	18	-	-	-	18	-	-	-	-	18	18	-	-	66	-	**66**

Table 14
SAX-HMM based method performance for case B

Variable	**1st possible faulty peak (S_1 samples)**	**Fault detection (S_2 samples)**	**Delay ($S=S_2-S_1$ samples)**	**Difference with VizTree ($\Delta S=S-S_v$ samples)**
1	-	-	-	-
5	-	-	-	-
10	-	-	-	-
14	**3,726**	**3,728**	**2**	**-16**
17	4,734	4,736	2	-64

Table 15
Detection delays for SAX-HMM based method, PCA, DPCA and CVA

Fault	**Fault start (S_1 samples)**	**1st detected variable**	**Detection delay (samples)**								**ΔD (samples)**
			SAX-HMM	**PCA (T^2)**	**PCA (Q)**	**DPCA (T^2)**	**DPCA (Q)**	**CVA (T_S^2)**	**CVA (T_r^2)**	**CVA (Q)**	
1	3,900	21	4	7	3	6	5	**2**	3	2	+2
2	3,415	16	**5**	17	12	16	13	13	15	25	-7
4	3380	21	**1**	-	3	151	**1**	462	**1**	-	0
7	4,380	20	1	1	1	1	1	1	1	**0**	+1
8	3,490	20	**16**	23	20	23	21	20	20	21	-4
13	3,600	21	**23**	49	37	45	40	42	39	43	-14
14	3,350	9	4	4	**1**	6	**1**	2	**1**	**1**	+3
17	3,345	21	**13**	29	25	28	24	27	20	63	-7
19	3,725	14	**2**	-	-	-	82	-	11	-	-9
20	3,920	20	**27**	87	87	89	84	82	66	72	-39

Moreover, valuable information can be provided for an entire set of process variables, either in the form of a cumulative *STI* index for the ensemble of *PVs*, or in the form of *MGAL* and instant *STIs* regions.

To conclude, the developed approach provides a far more effective approach than traditional *SAX*. The approach can be further developed and exploited towards various directions, such as:

- Independent audit of individual alarms
- Group alarm formulation
- Multivariate data mining
- Near miss estimation
- Plant Condition monitoring and automated quality control

Acknowledgments

This work was supported by Research Area Control of ABB Corporate Research.

References

[1] Pariyani A., Seider W. D., Oktem U. G., Soroush M., Incidents investigation and dynamic analysis of large alarm databases in chemical plants: a fluidized-catalytic-cracking unit case study, Industrial & Engineering Chemistry Research 49 (2010), 8062-8079.

[2] IEC 62682 Alarm management for the process industries (2014)

[3] Zhu J., Ge Z., Song Z., HMM-driven Robust Probabilistic Principal Component Analyzer for Dynamic Process Fault Classification, Industrial Electronics, IEEE Transactions on, (2015), in press.

[4] Li G., Qin S. J., Zhou D., A New Method of Dynamic Latent-Variable Modeling for Process Monitoring, Industrial Electronics, IEEE Transactions on, 61 (11) (2014), 6438-6445.

[5] Mori J. Yu J., Quality relevant nonlinear batch process performance monitoring using a kernel based multiway non-Gaussian latent subspace projection approach, Journal of Process Control, 24 (2014), 57-71.

[6] Jiang B., Huang D., Zhu X., Yang F., Braatz R. D., Canonical variate analysis-based contributions for fault identification, Journal of Process Control, 26 (2015), 17-25.

[7] Liu Y. J., Chen T., Yao Y., Nonlinear process monitoring and fault isolation using extended maximum variance unfolding, Journal of Process Control, 24 (2014), 880-891.

[8] Tong C., Yan X., Statistical process monitoring based on multi-manifold, Chemometrics and Intelligent Laboratory Systems, 130 (2014), 20-28.

[9] Escobar M. S., Kaneko H., Funatsu K., Combined generative topographic mapping and graph theory unsupervised approach for nonlinear fault identification, AIChE Journal, 61 (2015), 1559-1571.

[10] Robertson G., Thomas M. C., Romagnoli J. A., Topological preservation techniques for nonlinear process monitoring, Computers and Chemical Engineering 76 (2015), 1-16

[11] .Jiang Q., Yan X., Probabilistic monitoring of chemical process using adaptively weighted factor analysis and its application, Chemical Engineering Research and Design, 92 (2014), 127-138.

[12] Clifton L., Clifton D. A., Zhang Y., Watkinson P., Tarassenko L., Yin H., Probabilistic Novelty Detection With Support Vector Machines, Reliability, IEEE Transactions on, 63 (2) (2014), 455-467.

[13] Oster J., Behar J., Sayadi O., Nemati S., Johnson A., Clifford G., Semi-supervised ECG Ventricular Beat Classification with Novelty Detection Based on Switching Kalman Filters, Biomedical Engineering, IEEE Transactions on, 62 (9) (2015), 2125-2134.

[14] Wang S., Yu J., Lapira E., Lee J., A modified support vector description based novelty detection approach for machinery components, Applied Soft Computing, 13 (2013), 1193-1205.

[15] Liu Y. H., Liu Y. C., Chen Y. J., Fast support vector data description for novelty detection, Neural Networks, IEEE Transaction on, 21 (8) (2010), 1296-1313.

[16] Heyns T., Heyns P. S., de Villiers J. P., Combining synchronous averaging with Gaussian mixture model novelty detection scheme for vibration-based condition monitoring of a gear box, Mechanical Systems and Signal Processing, 32 (2012), 200-215.

[17] Sotiris A. V., Tse P. W., Pecht M. G., Anomaly Detection through a Bayesian Support Vector Machine, Reliability, IEEE Transansactions on, 59 (2010), 277–286.

[18] Guo W., Tse P. W., Djordjevich A., Faulty bearing signal recovery from large noise using a hybrid method based on spectral kurtosis and ensemble empirical mode decomposition, Measurement, 45 (2012), 1308–1322.

[19] Yu J., Multiway discrete hidden Markov model-based approach for dynamic batch process monitoring and fault classification, AIChE Journal, 58 (9) (2012), 2714-2725.
[20] Meng L. J., Xiang J. W., Wang Y. X., Jiang Y. Y., Gao H. F., A hybrid fault diagnosis method using morphological filter-translation invariant wavelet and improved ensemble empirical mode decomposition, Mechanical Systems and Signal Processing, 50-51 (2015), 101-115.
[21] Lin J., Keogh E., Lonardi S., Visualizing and discovering non-trivial patterns in large time series databases, Information Visualization 4 (2005), 61 – 82.
[22] Weber M., Alexa M., Mueller W., Visualizing Time-Series on Spirals, IEEE Symposium on Information Visualization (2001), 7-14.
[23] Jugel U., Jerzak Z., Hackenbroich G., M4: a visualization-oriented time series data aggregation, Proceedings of the VLDB Endowment ,7 (10) (2014), 797-808.
[24] Keogh E., Hochheiser H., Shneiderman B., An augmented visual query mechanism for finding patterns in time series dat,.Proceedings of the 5th International Conference on Flexible Query Answering Systems (2002), 240–250.
[25] Hochheiser H., Shneiderman B, Dynamic query tools for time series data sets, timebox widgets for interactive exploration, Information Visualization 3 (1) (2004), 1–18.
[26] Aris A., Shneiderman B., Plaisant C., Shmueli G., Jank W., Representing unevenly-spaced time series data for visualization and interactive exploration, Proceedings of the International Conference on Human–Computer Inter-action (2005), 835–846.
[27] Buono P., Aris A., Plaisant C., Khella A., Shneiderman B., Interactive pattern search in time series. Proceedings of the Conference on Visu Data Analysis (2005), 175–186.
[28] Lin J., Koegh E., Lonardi S., Visualizing and discovering non-trivial patterns in large time series databases, Information Visualization 4 (2) (2005), 61-82.
[29] Lin J., Keogh E., Wei L., Lonardi S., Experiencing SAX: a novel symbolic representation of time series, Data Mining and Knowledge Discovery 15 (2007), 107-144.
[30] Yankov D., Keogh E., Lonardi S., Fu A., Dot plots for time series analysis, Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (2005), 159–168.
[31] Fu T. C., Chung F. L., Kwok K. Y., Ng C.M., Stock time series visualization based on data point importance, Engineering Applications of Artificial Intelligence 21 (8) (2008), 1217–1232.
[32] Li H., Yang L., Time visualization based on shape features, Knowledge-Based Systems, 41 (2013), 43-53.
[33] Kondaveeti S. R., Izadi I., Shah S. L., Black T., Chen T., Graphical tools for routine assessment of industrial alarm systems, Computer & Chemical Engineering 46 (2012), 39-47.
[34] Kumar N., Lolla N., Keogh E., Lonardi S., Ratanamahatana C. A., Time-series Bitmaps: A Practical Visualization Tool for working with Large Time Series Databases, SIAM Data Mining Conference (2005).
[35] Li H., Guo C., Yang L., A method of similarity measure and visualization for long time series using binary patterns, 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) (2011).
[36] Rabiner L. R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, 77 (2) (1989), 257-286.
[37] Keogh E., Lin J., Fu A., HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence, Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05), (2005).
[38] Shieh J., Keogh E., iSAX: disk-aware mining and indexing of massive time series datasets, Data Mining and Knowledge Discovery, 19 (2009), 24-57.
[39] Lkhagva B., Suzuki Y., Kawagoe K., Extended SAX: extension of symbolic aggregate approximation for financial time series data representation, Proceedings of Data Engineering Workshop (2006).
[40] Niennattrakul V., Srisai D., Ratanamahatana C. A., Shape-based template matching for time series data, Knowledge-Based Systems, 26 (2012), 1-8.
[41] Downs J. J., Vogel E. F., A plant-wide industrial process control problem, Computers and Chemical Engineering, 17 (3) (1993), 245-255.
[42] Ricker N. L., Decentralized control of the Tennessee Eastman challenge process, Journal of Process Control, 6 (4) (1996), 205-221.
[43] http://cs.gmu.edu/~jessica/viztree.htm
[44] Russell E. L., Chiang L. H., Braatz R. D., Fault detection in industrial process using canonical variate analysis and dynamic principal component analysis, Chemometrics and Intelligent Laboratory Systems, 51 (2000), 81-93.