Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 4724507, 12 pages
https://doi.org/10.1155/2019/4724507

Research Article

An Enhanced Evaluation Method of Sequential Probability Ratio Test

Gabor Gardonyi,[1] **Gabor Por**,[2] **and Krisztian Samu**[1]

[1]*Department of Mechatronics, Optics and Mechanical Engineering Informatics, Budapest University of Technology and Economics, Muegyetem rkp. 3., 1111 Budapest, Hungary*
[2]*Institute of Nuclear Techniques, Budapest University of Technology and Economics, Muegyetem rkp. 3., 1111 Budapest, Hungary*

Correspondence should be addressed to Gabor Gardonyi; gardonyi@mogi.bme.hu

Received 21 October 2018; Revised 3 February 2019; Accepted 25 February 2019; Published 18 March 2019

Academic Editor: Akhil Garg

Accurate event detection has high priority in many technical applications. Events in acquired data series, their duration, and statistical parameters provide useful information about the observed system and about its current state. This information can be used for condition monitoring, state identification, and many kinds of forecasting as well. In some cases background noise covers the events and simple threshold or power monitoring methods cannot be used effectively. A novel method called Scaled Sequential Probability Ratio Test (SSPRT) produces 2D array of data via special cumulative sum calculation. A peak determination algorithm has also been developed to find significant peaks and to store the corresponding data for further evaluation. The method provides straight information about the endpoints and possible duration of the detected events as well as shows their significance level. The new method also gives representative visual information about the structure of detected events. Application example for thermomechanical fatigue test monitoring and another for vibration based rotational speed estimation of a four-cylinder internal combustion engine is discussed in this paper.

1. Introduction

Since the 18th century there has been a growing interest in statistical hypothesis testing. In the literature, several theories have been already proposed to extend the applicability of the mathematical background or to optimize calculation. However, new methodologies [1–4] and applications [5–10] of statistical hypothesis tests are published every year. The so-called Bayes' theorem in probability theory has been established by Thomas Bayes in the early 1700s. Jerzy Neyman and Egon Pearson analysed the efficiency of hypothesis tests and have published their work in 1933 [11]. The Neyman-Pearson lemma offers a rule of thumb when all the data have been already collected. The lemma states that when performing a hypothesis test between two simple hypotheses $H_0 : \theta = \theta_0$ and $H_1 : \theta = \theta_1$, the likelihood ratio test which rejects H_0 in favour of H_1 when $\Lambda(x) = L(x \mid \theta_0)/L(x \mid \theta_1) \leq \eta$ where $P(\Lambda(x) \leq \eta \mid H_0) = \alpha$ is the most powerful test at significance level α for a threshold η. The results of Neyman and Pearson inspired Abraham Wald in the mid-1940s to reformulate it as a sequential analysis problem [12]. A test can solely be called sequential, if the number of observations is not predetermined, but it is dependent on the outcome of the observations. A great summary about the evolution of sequential hypotheses tests and the early stages of the development can be found in Walds paper [12] in Chapter B.

In the last decades, there have been a surge of practical applications of the SPRT methodology in many areas including low frequency sonar detection, passive acoustic detection of marine mammals, tracking of signals, target tracking, early detection of changes in signals, computer simulations, data mining, clinical trials, gene ordering, agricultural sciences, horticulture, pest management, educational testing, economics and finance [13]. The latest published applications of SPRT focus for example on human core temperature prediction to prevent hyperthermia [5], diagnostics of the COMPASS tokamak [6], and W7X stellarator [10]; furthermore, many vibration measurement based applications of the methodology can be found in the literature [7, 8].

Some practical technical applications of SPRT have been also published by the authors. In these cases SPRT has been used for malfunction detection [14], place identification [15], or AE (acoustic emission) event detection [16]. However in the case of the referenced fusion diagnostic [6, 10] and AE applications [16] the duration of events seems to be a critically important information. Based on this information burst events derived from the test material could be separated from other types of events. Also in the case of vibration signal analysis of an internal combustion engine, setting limits for the length of detected events can help to make determination of cylinder activities more accurate. This paper discusses an AE application where ultrasonic burst events can be localised in noisy environment using the new methodology. The rate of hits shows clear correlation to the applied pulling force to the test specimen. Tachometerless speed estimation possibility of an internal combustion engine is also demonstrated.

In the following we introduce a novel extension to the classic SPRT method which produces a probability space and can help to get straight information about the most probable duration of detected events. In this paper a suggested peak detection method is also introduced which has been developed especially to evaluate the data resulted by the presented test method.

2. Scaled Sequential Probability Ratio Test

Introducing the Scaled Sequential Probability Ratio Test (SSPRT), it is an extended version of the classic SPRT method, which provides straight information about the endpoint and the duration of subsections of a signal with changed statistical distributions. In the following we refer to these changed subsections as events. The discussed methodology can be applied on discrete data series and the output data also stays discrete in time.

The methodology of the SSPRT can be separated into three main steps. First the likelihood ratio is calculated point by point via SPRT. Second a series of fixed size cumulative sums (cumulative sum array, CSA) are calculated sequentially with different predefined sizes. The range and the desired resolution of the observed event duration must be defined before the calculation starts. This way the method produces a 1D numeric array as output for every single numeric input value. After concatenating these 1D arrays the result changes into a 2D numeric array. This partial result of the SSPRT is quiet representative when displayed in a colormap, waterfall diagram, contour map, or as a 3D surface. The resulting surface shows a probability space, where the peaks determines the main properties (endpoint location and event duration) of the most possible event types. The third step of the SSPRT is the localisation of these local maximum points and selection of the most relevant ones.

2.1. Calculation of the Sequential Probability Ratio Test. This paper just gives a general overview about the background of SPRT. For definitions and detailed discussion of the theory please see [12]. We consider a simple hypothesis $H_0 : \theta = \theta_0$ against a simple alternative $H_1 : \theta = \theta_1$. The SPRT for testing H_0 against H_1 is given as

$$z_i = \log \frac{f(x_i, \theta_1)}{f(x_i, \theta_0)} \tag{1}$$

where x_i denotes the i^{th} observation on x. According to the original method two constants are chosen where $B < A$ and both are dependent on the size of the acceptable decision errors. The errors are specified by $P(H_1 \mid H_0) \leq \alpha$ and $P(H_0 \mid H_1) \leq \beta$ where α and β are small constants. The Cumulative Sum (CUSUM) of z_i is calculated based on

$$Z_m = z_1 + \cdots + z_m \tag{2}$$

Then the rule of decision is made as follows:

(i) if $\log B < Z_i < \log A \longrightarrow$ take another observation,

(ii) if $Z_i \geq \log A \longrightarrow$ reject H_0,

(iii) if $Z_i \leq \log B \longrightarrow$ accept H_0.

The threshold values A and B and the error probabilities α and β are connected in the following way:

$$A = \frac{(1-\beta)}{\alpha} \tag{3}$$

$$B = \frac{\beta}{(1-\alpha)} \tag{4}$$

Although Wald has shown that the SPRT procedure will stop with a decision with a finite number of observations, in some practical cases it is necessary to define a maximum test length. The procedure is then called the Truncated SPRT (TSPRT) [4, 7]. By truncating if the sequential process does not lead to a final decision after a given number of observation, an additional rule is given for the acceptance of hypothesis:

(i) If $Z_n > 0$, accept H_1.

(ii) If $Z_n \leq 0$, accept H_0.

However in this manner we change the probabilities of α and β. In the literature applications of fixed sample size tests can be also found. Other interesting extensions of the classical method are for multihypothesis testing [3, 17].

The first step of the SSPRT data evaluation is a special SPRT evaluation method, where parallel calculation of several SPRTs has to be done. The main idea of the algorithm is that if the z_i value has already been calculated in the i^{th} iteration, it can be reused for multiple hypothesis testing. The only criteria are that the different SPRT processes must be based on same hypothesis and the only difference between them is in the fixed run length. In our application at this first stage of calculation the lower and upper decision levels have been left out of consideration. In this manner the evaluation process always continues and never stops or restarts. This is because the SSPRT was developed for real-time applications and must be insensitive for initial value problems. For evaluation of the SPRT with fixed size log likelihood ratio (LLR) summation, a suitable method is the later discussed

CUSUM algorithm. To make the final decision, one can easily use the rules of TSPRT already discussed before or define alternative decision methods and apply them in a later step.

During the tests of the implemented methods we used test signals mostly with normal distribution and zero mean. These terms makes the SPRT equations much simpler. The point-by-point formula is shown by (5):

$$z_i = \log \frac{f(x_i, \theta_1)}{f(x_i, \theta_0)} = \frac{\sigma_1^2 - \sigma_0^2}{2\sigma_1^2\sigma_0^2} x_i^2 - \log \frac{\sigma_1}{\sigma_0} \tag{5}$$

We can take some more simplifications if we introduce a new variable c which marks the prior knowledge about the ratio of the deviations after and before the change point.

$$c = \frac{\sigma_1}{\sigma_0} \tag{6}$$

where σ_0 is the standard deviation of the whole signal and σ_1 is the standard deviation of the changed subsections. If there are rare and relatively short events in the signal, we can say that σ_0 is the standard deviation of the background noise, because of the negligible effect of the rare events. If we reorder the equation for c, we get $\sigma_1 = c\sigma_0$. Equation (7) shows the reordered form of (5).

$$z_i = \frac{c^2\sigma_0^2 - \sigma_0^2}{2c^2\sigma_0^2\sigma_0^2} x_i^2 - \log(c) = \frac{c^2 - 1}{2c^2} \frac{x_i^2}{\sigma_0^2} - \log(c) \tag{7}$$

where $(c^2 - 1)/(2c^2)$ and $(-\log(c))$ are constants and can be signed as c' and c'' respectively. These constants can be computed and predefined. Equation (8) is the final equation for z_i.

$$z_i = c' \frac{x_i^2}{\sigma_0^2} + c'' \tag{8}$$

where x_i is the i^{th} sample in the source data series. The σ_0 parameter can be a predefined value or a sequentially computed parameter which can be easily refreshed using the x_i observation. This way the SPRT will be an adaptive hypothesis test, which can change its sensitivity according to the background noise level. This implementation needs a few learning samples at the beginning of any evaluation to get the output with acceptable reliability.

2.2. Calculation of the Cumulative Sum Array. Many kind of sequential change point detection methods are available in the literature. These are usually called control charts or process-behaviour charts, but they are also known as Shewhart charts. Such elementary algorithms are for example the Combined Adaptive (CA) chart, Fixed Sampling Distance (FSD) chart, Fixed Sample Size (FSS) chart, Variable Sampling Distance (VSD) chart, Variable Sample Size (VSS) chart, Cumulative Sum (CUSUM) chart, Finite Moving Average (FMA) chart, and the Exponentially Weighted Moving Average (EWMA) chart. Most of the algorithms are used in practice work on samples of data with fixed size window or moving fixed size window.

The idea of the LLR-based CUSUM algorithm is that the prechange mean of the LLR is negative and the postchange mean is positive. If the analysed signal contains a changed subsection with a given length, the local result of the CUSUM [16] will be the highest at the end of the changed subsection if the cumulative calculation has been started just before the first change. In the case of our application a special implementation of the CUSUM chart has been used. The SSPRT algorithm realizes parallel sequential hypothesis tests. In this case the same hypothesis must be supposed for parallel threads. Only the probabilities of α and β of parallel TSPRTs varies because of the difference between the fixed numbers of summed LLR values. This way the LLR value must be calculated only once for a given sample in the source data series. The same value can be used to perform the actual iteration of the fixed size rolling CUSUM calculations. Henceforth the evaluation of a given data series using parallel CUSUM algorithms with different parameters is noted as CSA (Cumulated Sum Array). The CSA can be calculated in several ways and it results in a numeric array for every single numeric input. The calculation methodology will be discussed in later sections.

The CUSUM step of the SSPRT slightly differs from the conventional way of calculation. Many improved CUSUM methods are available in the literature. An interesting sequential solution has been called one-sided process inspection scheme or the extended two-sided version of it introduced by Page [18] has already much less sensitivity to the initial score because of the applied adaptive threshold. The one-sided version computes the cumulative sum as $S_n = \sum_{k=1}^{n} x_k$ and takes action if $S_n - \min_{0 \le i < n} S_i \ge h$. The rule can be reformulated as a sequential problem as follows in (9). The method takes action after the n^{th} observation if $S'_n \ge h$.

$$\begin{aligned} S'_n &= \max\left(S'_{n-1} + x_n, 0\right) \quad (n \ge 1) \\ S'_0 &= 0 \end{aligned} \tag{9}$$

This method is sometimes called *CUSUM Algorithm as a Repeated SPRT* [17]. However in the SSPRT we need the "repeated" property of such algorithms but do not need the adaptive threshold because of using the decision rules of the TSPRT methodology discussed before. The SSPRT is a sequential algorithm which analyses the input series and makes decisions point by point. It must be totally insensitive to initial value settings and should never reset the CUSUM procedure because of special rules.

During SSPRT we take parallel CUSUM procedures simultaneously. One of these parallel processes can also be introduced as a special fixed sample size elementary change point detection algorithm which works on samples of data with fixed size sliding window.

Many publications show novel methods for optimized change detection using CUSUM as can be seen in references [2, 19] or analyse work efficient parallel scan solutions to get CUSUM values [20]. As in the case of these papers, the focus of our work has been to develop a general framework that offers scalable performance and operates in limited memory and in real-time. We supposed in the previous section that

the events buried in the background noise are short and rare enough to keep the rolling σ_0 parameter unbiased. We also suppose that the minimum and maximum length of the significant events are known and signed as L_{min} and L_{max} respectively. The SSPRT use a sliding window over the data stream. Its length is the amount of historical data that an algorithm stores and considers during the computation. The SSPRT method contains several parallel CUSUM processes, where only the length of the sliding window is different. The numeric results of these CUSUMs are the rows of the SSPRT output data. They operate with $N_j : \{N_0, N_1, \ldots, N_{m-1}\}$ window sizes, where m is a user defined parameter and notes the number of rows. The N_j parameters must be chosen and $L_{min} \leq N_0 \leq N_1 \leq \cdots \leq N_{m-1} \leq L_{max}$. If a single number from the source data stream at time instant i is x_i and N_j is the length of the sliding window, then the algorithm considers only the data points $\{x_{i-N_j+1}, \ldots, x_i\}$. As new data become available, the sliding window slides by one element, discarding the oldest and incorporating the latest data element. Now we show how to compute one row of the SSPRT if the N_j parameter has been already chosen. If we follow a naive way to implement this algorithm, the equation will be as

$$CSA\ I.: S_i^j = x_{i-N_j+1} + \cdots + x_i = \sum_{k=i-N_j+1}^{i} x_k \tag{10}$$

This algorithm needs N_j addition operation for every single S_i^j, i.e., to get one column of the SSPRT at time instant i $\sum_{j=0}^{m-1} N_j$; addition operations have to be done. This method needs to store the most recent N_{m-1} pieces of input data in a buffer. The advantages of this algorithm are the simplicity and that using this it is not necessary to calculate all of the S_i^j values in predefined order. One can easily skip unnecessary points or can also calculate the output values as independent processes via multicore processor system. The disadvantage is that this algorithm needs in normal continuous operation a very large number of addition operations and the number of operation depends on the chosen N_j parameters. Another bad property is that the same data samples must be read from the buffer more times during the same iteration.

It must be noted that for example the corresponding subsections of the source data series of S_{i-1}^j and S_i^j have huge ($N_j - 1$ samples) overlap. If it is wanted to fulfil much less operation steps, one can easily reuse the outputs of the previous iteration. Now we show the recursive equation for S_i^j calculation in (11).

$$CSA\ II.: S_i^j = S_{i-1}^j - x_{i-N_j} + x_i \tag{11}$$

This implementation contains only one subtraction and one addition, i.e., to get one column of the SSPRT only $2m$ numeric operations have to be done contrary to the previous algorithm. However it requires almost the same buffer size to store input data temporarily, and it also needs another buffer for the outputs of the previous iteration. It means one buffer of size $N_{m-1} + 1$ and another of size m.

We can take similar simplification if we take notice of the overlap of the corresponding subsection of S_i^{j-1} and S_i^j. Since $N_{j-1} \leq N_j$, S_i^j can also be computed as follows:

$$\begin{aligned} CSA\ III.: S_i^j &= \left(x_{i-N_{j-1}+1} + \cdots + x_i\right) \\ &\quad + \left(x_{i-N_j+1} + \cdots + x_{i-N_{j-1}}\right) \\ &= S_i^{j-1} + \sum_{k=i-N_j+1}^{i-N_{j-1}} x_k \end{aligned} \tag{12}$$

In this case the overall process needs the same buffer size for the input elements. The advantage of this algorithm is that it does not need to store an array of previous output data. It uses only one output data from the previous CUSUM row instead.

In the next session two suggested scale types and the corresponding equations are shown. First of all we have to fix some basic parameters which define the properties of the chosen scale. These parameters are the minimum and maximum length of the sliding window expressed in number of samples (L_{min} and L_{max} respectively) and the number of scales (m). All of these parameters are positive integers and must be greater than zero: $1 \leq L_{min} \leq L_{max}$ and $1 \leq m$.

2.2.1. Linear Scaled CSA. First we show a basic implementation where the L_{min} parameter has to be ignored and $m = L_{max}$. Now the only available setting is L_{max} and the resolution of the output will be this way the highest because $N_j = j + 1$ for $j : 0, \ldots, m - 1$.

Using one of the above described CSA equations we get an output array for every single input data, and the sequence evaluates the presence of subsections that have been changed in the input data series and have got the given N_j length. The resolution of the observed length is this way the best, equaling the sampling rate of the input data series. When using the default $L_{min} = 1$, both of the CSA II and the CSA III algorithms can be used easily. If an alternative L_{min} is required, the CSA II seems to be practical, because in the case of the CSA III the rows that correspond to window sizes below L_{min} must be also computed to get the final result. These additional output values can be released after the actual column has been completely calculated.

The duration of the detected events are often less important as the accurate localisation. To decrease the buffer sizes and to get an output via less overlapped calculations, a possible way is choosing m less than the observed range of length.

A modified parameter m' can be defined as $m' = \min(m, R)$, where $R = L_{max} - L_{min} + 1$. This is the feasible resolution according to the chosen parameters. The slight modification of the m parameter also modifies the feasible range (R') of the analysis as $R' = (m' - 1)d$. First, we have to compute (13) for the optimal distance between two windows that are neighbours.

$$d = \left\lfloor \frac{R}{m'} \right\rfloor \tag{13}$$

N_7=8	-8	-6	-4	-4	-2	0	2	4	4	2	0	0	-2
N_6=7	-7	-5	-3	-3	-1	1	3	**5**	3	1	1	-1	-3
N_5=6	-6	-4	-2	-2	0	2	4	4	2	2	0	-2	-4
N_4=5	-5	-3	-1	-1	1	3	3	3	3	1	-1	-3	-5
N_3=4	-4	-2	0	0	2	2	2	**4**	2	0	-2	-4	-4
N_2=3	-3	-1	1	1	1	1	3	3	1	-1	-3	-3	-3
N_1=2	-2	0	**2**	0	0	2	2	2	0	-2	-2	-2	-2
N_0=1	-1	1	1	-1	1	1	1	1	-1	-1	-1	-1	-1

Figure 1: Example for linear scaled CSA calculation.

The array of the length of all used windows can be now expressed as

$$N_j = L_{min} + jd \quad \text{for } j : 0, \ldots, \left(m' - 1\right) \tag{14}$$

Let see an example to show how this scaling method works. Let the basic parameters be $L_{min} = 1$, $L_{max} = 8$, m=8. The input data is generated as a series of negative and positive units. The negative units illustrate the LLR which corresponds to the background noise and the positive values show where the statistic has been changed. Two events are simulated in this example. The first one starts at the second sample and has only two samples. The other one starts at the fifth sample of the series and has four samples. It can happen that the classification of some samples, which has been acquired during an event, is incorrect. If two or more detected short events are near enough to each other, that they also can be parts of a more dominant long change must be analysed.

Figure 1 shows the output data array (S_i^j), where the bottom row (S_i^0) equals to the input data series because $N_0 = 1$. In the figure the white cells indicate negative data points. According to the decision rules of the TSPRT method, in the case of grey or black cells the alternative hypothesis is accepted; i.e., the signal has been changed. The SSPRT can be a special decision procedure, which results in point-by-point m decisions corresponding to the most recent N_j input samples. On the other hand the output can be seen as a special probability space, where local peaks show the most probable parameter combinations (location and duration) of a possible event.

As it can be seen in the presented example, two changed subsections are in the input data series. Two of the three highlighted local peaks are at the same location. It means that the last four samples are parts of a changed subsection, but it is also possible that these points are parts of a longer event which contains seven data points before the time instant. In a later chapter it will be discussed how is it possible to locate the most relevant peaks in this data array in a sequential way.

2.2.2. Logarithmic Scaled CSA. In the case of many applications logarithmic scaling makes the result much informative. In the case of the lower section of the logarithmic scale, short changes can be observed much better because of the better resolution.

Let us introduce a new parameter g which is quite similar to the parameter d in the case of linear scales. Here g shows the ratio of the window length of two neighbour scales and $N_0 = L_{min}$. If the number of requested CSA rows (m) and L_{max} are also predefined input parameters, one have to find the optimal g value:

$$g = \sqrt[m-1]{\frac{L_{max}}{L_{min}}} \tag{15}$$

Based on the latter (16) defines the N_j array for a logarithmic scale.

$$N_j = \left[L_{min} g^j\right] \quad \text{for } j : 0 \ldots (m-1) \tag{16}$$

2.3. Peak Detection. Several mathematical methods are available to detect peaks in data series. It is a difficult question, which one is the best choice to use in our application. We chose a specialised method. In the case of the SSPRT it is a special property that the neighbour rows in the output are results of overlapped sum calculation with different window sizes. Because of the applied rectangular windows another important property is that all rows are scaled, delayed, and low-pass filtered versions of the original input data. The presented 2D output data of SSPRT can be also computed as an output array of a special low-pass filtering filter bank. CSA I and CSA III, which have been presented in a previous section of this paper, represent different ways of calculation to get the same result. These two methods follows the same logic of calculation as techniques for FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filtering respectively. The CSA II method is also similar to FIR techniques as CSA I. The former represents just an optimized way to get the same output.

Another important question may arise if one develops a peak detection method for 2D data arrays. Do we need all local peaks? In our case there is a time instant where two peaks can be found in the same column. The value of these peaks represents the probability of the corresponding subsections to be an event, while unchanged sections of the input (contains only background noise) result in negative LLRs and events (changed subsections) result in positive LLRs. If a peak has higher maximum value than another, the applied window has longer overlap to the detected event. In such cases the longer event with higher peak value has to be stored and lower peaks in the same column can be ignored.

In the case of our example series illustrated by Figure 1, three local maximum points can be found. The coordinates

of these points in the three-dimensional Descartes coordinate system are [12; 1; 2], [17; 3; 4], [17; 6; 5]. Based on the latter considerations just two of the peaks are relevant [12; 1; 2] and [17; 6; 5]. These points correspond to overlapping subsections of the original signal so we should decide which one to store as a final decision. Similarly to the case of overlapped subsections which have been detected at the same time instance, now we can do the same comparison and decision procedure for the applied window length and the peak magnitude. Based on probability ratio test the final decision is that the signal contains a changed subsection which starts at the 11^{th} time instance and has a length of seven samples.

Now let us see some mathematical equations and how this logic has been implemented as a sequential evaluation method. In the following the methodology is discussed in four main steps.

Step 1. At first step, the maximum points of the sequentially calculated columns of CSA have to be found. At all time instances, the maximum values are searched and the coordinates of the resulting maximum point are stored in a new array. These parameter sets (time instance or column index; row index; maximum value of the analysed column) are noted as $M_i[i, r_i, v_i]$.

$$x_i \longrightarrow M_i\left[i, r_i, v_i\right] \tag{17}$$

$$\max_{0\le l\le m-1} S_i^l = S_i^{r_i} = v_i \tag{18}$$

In the M_i series the coordinates of the maximum point and the corresponding value can be stored. However it is practical to drop the row index and use the corresponding sliding window size instead. This parameter describes the calculation process better and can be useful in later steps if already has been stored. During sequential event detection methods the length of an event can only be determined when the event has already ended and the used window size is a key parameter at this point if the start date of the event (b_i) is needed.

$$M_i\left[i, r_i, v_i\right] \longrightarrow \widehat{M}_i\left[b_i, w_i, v_i\right] \tag{19}$$

$$b_i = i - \left(w_i - 1\right) \tag{20}$$

$$w_i = N_{r_i} \tag{21}$$

Step 2. The decision logic has to be sequential. It means we only know the values before and at the examined time instance. First, possible local maximum points in the series of the cumulated LLR values have to be determined. If the current value is greater than or equal to the value of the previous iteration, a partial decision can be stated that the point can be a local maximum (LM) point of the series.

$$d_i^{LM} = \begin{cases} 1 & if\ v_i \ge v_{i-1} \\ 0 & if\ v_i < v_{i-1} \end{cases} \tag{22}$$

where d_i^{LM} is the Boolean result of the decision about the i^{th} input data which could be a local maximum. Boolean outputs as 0 and 1 represent false and true values respectively.

Second, the evaluation of the applied SPRT rules should be an alert if the analysed subset of the data series could correspond to an event.

$$d_i^{LLR} = \begin{cases} 1 & if\ v_i \ge A \\ 0 & if\ v_i < A \end{cases} \tag{23}$$

where d_i^{LLR} is the result of the local decision of the applied SPRT logic. The threshold parameter A discussed earlier at the introduction of SPRT can be configured to zero if one follows the evaluation methodology of TSPRT but can also be chosen to any positive value, which affects the accuracy and the sensitivity of the classical SPRT methodology.

If both partial decisions result true, the actual $\widehat{M}_i$ data cluster has to be stored for later use. The d_i^Has the holding decision at time instance i can be determined as $d_i^H = d_i^{LLR} \bullet d_i^{LM}$.

HM_i is a new series, which holds $\widehat{M}_i$ according to the d_i^H decision. HM_i can be determined point by point using the following equations:

$$\widehat{M}_i\left[b_i, w_i, v_i\right] \longrightarrow HM_i\left[hb_i, hw_i, hv_i\right] \tag{24}$$

$$HM_i = \begin{cases} HM_{i-1} & if\ d_i^H = 0 \\ \widehat{M}_i & if\ d_i^H = 1 \end{cases} \tag{25}$$

Step 3. The goal of this step is to find the endpoint of the detected events. Now the decision result at time instance i will be signed as D_i.

$$D_i = \begin{cases} d_{i-1}^H & if\ d_i^H = 0 \\ 0 & if\ d_i^H = 1 \end{cases} \tag{26}$$

Any time if the D_i results true, it can be stated that a detected event or statistical change just ended. This is not the final decision yet, because these alerts can also sign tiny event parts, which can also be a part of a long change. It must be cleared that the possible correspondence to a longer event can only be determined later.

$$HM_i\left[hb_i, hw_i, hv_i\right] \longrightarrow DM_i\left[db_i, dw_i, dv_i\right] \tag{27}$$

$$DM_i = \begin{cases} DM_{i-1} & if\ d_i^H = 0 \\ HM_i & if\ d_i^H = 1 \end{cases} \tag{28}$$

Step 4. In this step the previously detected possible events are evaluated. These events can correspond to the same event or they can be registered as individual events. To determine this property, a watchdog logic has been used. The watchdog must restart its operation if the actual d_i^H becomes true. The initial value when restart is needed equals always the actual hb_i value. The watchdog (wdg_i) decrements its stored value by one if d_i^H false and alerts if the stored value reaches the

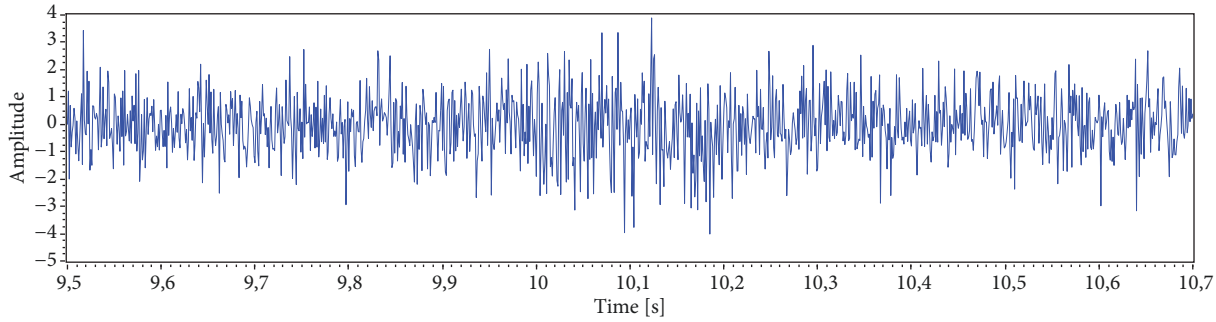

Figure 2: Time signal of the generated Gaussian background noise with additional Gaussian noise between 10 and 10.2 seconds.

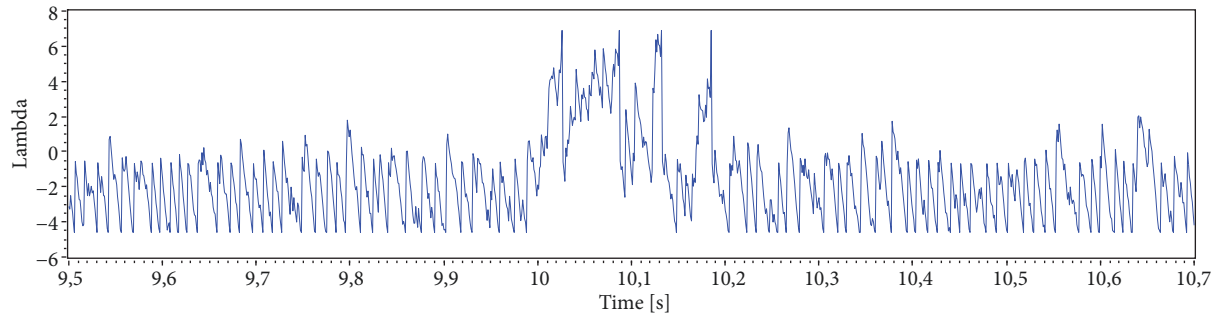

Figure 3: Lambda values resulted by the common SPRT evaluation method.

zero level. This alert is signed now as FD_i which means the final decision at time instance i.

$$wdg_i = \begin{cases} wdg_{i-1} - 1 & if\ d_i^H = 0 \\ hw_i & if\ d_i^H = 1 \end{cases} \quad (29)$$

$$FD_i = \begin{cases} 1 & if\ wdg_i = 0 \\ 0 & if\ wdg_i \neq 0 \end{cases} \quad (30)$$

Once a final decision result has been determined as true, the actual property values stored in DM_i correspond to the most possible event detected in the last section of the source signal. These results are signed as P_i as event properties after final decision at time instance i.

$$DM_i\,[db_i, dw_i, dv_i] \longrightarrow P_i\,[pb_i, pw_i, pv_i] \quad (31)$$

$$P_i = \begin{cases} invalid & if\ FD_i = 0 \\ DM_i & if\ FD_i = 1 \end{cases} \quad (32)$$

In the case of the example series shown in Figure 1, the final decision about the detected event parameters is as follows:

$$P_i\,[pb_i, pw_i, pv_i] = P_{24}\,[11, 7, 5] \quad (33)$$

This means that a decision could be made at the 24^{th} time instance, that a statistical change has been occurred in the source signal starting at the 11^{th} time instance, and its most probable duration is seven samples.

3. Using the SSPRT Method for Analysing Generated Test Signal

The applicability of the developed SSPRT methodology has been tested on several types of test signals. The method seemed to be working as expected in the case of generated test signals and on real life measurement signals as well. The following data series and figures are just representative examples for the evaluation of a generated test signal. Detailed discussion of applications will be topics for later publication.

The following example in Figure 2 shows a tiny section of the generated signal that contains Gaussian white noise with unit RMS (Root Mean Square) as constant background and periodically contains additional Gaussian noise with the same power and 0.2 seconds duration. The sections with additional noise are the simulated events which have to be detected using the SSPRT methodology. The additional noise results slightly in increased power of the signal between 10 and 10.2 seconds, and it still have Gaussian distribution during the event. If one takes a look at the generated time signal which is shown by Figure 2, it can be stated for the first sight that a common threshold test cannot be used for reliable event detection and also the start and end points of the events cannot be determined accurately. If the common SPRT evaluation method has been chosen we get the data series as Lambda values shown by Figure 3. The common SPRT is able to detect the simulated event as the decision function has reached the level of 6.9 more times. It can be seen that event detection works well, if LLR values have been processed using the common evaluation method, but how to get exact duration of the event remains a question. In the case

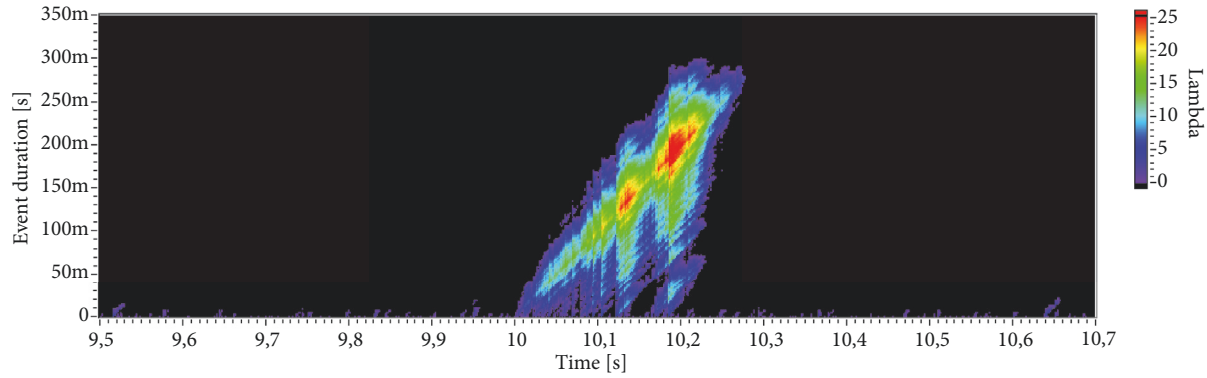

Figure 4: Linear scaled CSA output. Peak coordinates: [10.191; 0.192; 28.622].

of the novel SSPRT evaluation method we need to apply one of the above suggested CSA calculation methods to the LLR values. The fixed size cumulative sum calculation results in the highest value, if the length of the applied window fits to the event duration and if these two have full overlap. It can be easily demonstrated. In the case of a too short window, not all event-related positive LLR values are used for sum calculation, and too long windows occur that background noise related negative LLR values are taken into account.

Figure 4 illustrates the resulting probability surface where the highest peak shows the most probable duration end endpoint of the detected event. Determined duration of the statistical change has been 192 ms and its probable start time 9.999 s. These are extremely good estimations for the 200 ms long simulated event beginning at 10 s.

Local maximum points of the resulting three-dimensional surface can be easily found by several peak searching algorithms which can be applied to the presented data of the above colormap. The previously described mathematical background gives a possible solution to realize an effective peak searching method especially for the CSA output.

4. Practical Applications of SSPRT

SSPRT can be a versatile tool for any applications, where changes of the noisy input signal have to be detected accurately and the length of events is a key parameter for further evaluation. In the case of any practical applications where common SPRT evaluation has been applied successfully on sampled data series, the novel SSPRT methodology will provide at least that accurate event detection. Furthermore SSPRT shows directly the most probable duration which can help in event classification or source identification.

In the case of AE applications only well optimized sequential algorithms can be used for real-time evaluation. Steel under heat or pressure stresses emits AE signals, which can be detected using AE sensors. Experiments have been planned and carried out using a material testing simulator shown in Figure 5. Detailed description of the experimental setup and the PXIe-6363 based data acquisition system can be found in reference [16, 21]. The results showed that the SPRT method is absolutely suitable for AE event detection during

Figure 5: The Gleeble 3800 thermomechanical simulator [6].

fatigue test. Figure 6 clearly demonstrates the relationship between the applied loading force and the rate of detected AE events. The test specimen produced much more ultrasonic bursts during pulling forces than in the case of compression.

One of the detected bursts can be seen in Figure 7 and the corresponding CSA result of the SSPRT evaluation in Figure 8. In the case of the evaluation methodology introduced by reference [16] smoothing was applied to the LLR values before locating start and end time of events. It seems to work well, but this way averaging parameters must be fixed before evaluation. The disadvantage of the moving average method is that rapid internal changes during an event cannot be observed; this is also mentioned in that article. SSPRT could be a better tool for analysing such events' internal structure. If we take a look at the colormap, the highest peak shows that the detected event is 237.8μs long, starts at 264.9μs, and lasts for 502.7μs. The resulting data can be also displayed as a three-dimensional surface, which provide new opportunities for further studies. Other AE applications are for example failure-monitoring of slide bearings [22] or localisation of initial cracks in windscreen glass [23], where SSPRT could be a great tool for accurate event detection, localisation, and analysis.

Based on the above, SSPRT is well suited for detection of rare events in noisy environment. This is guaranteed by the hypothesis test theory, and it is possible to calculate the probability of correct and incorrect decisions. If events

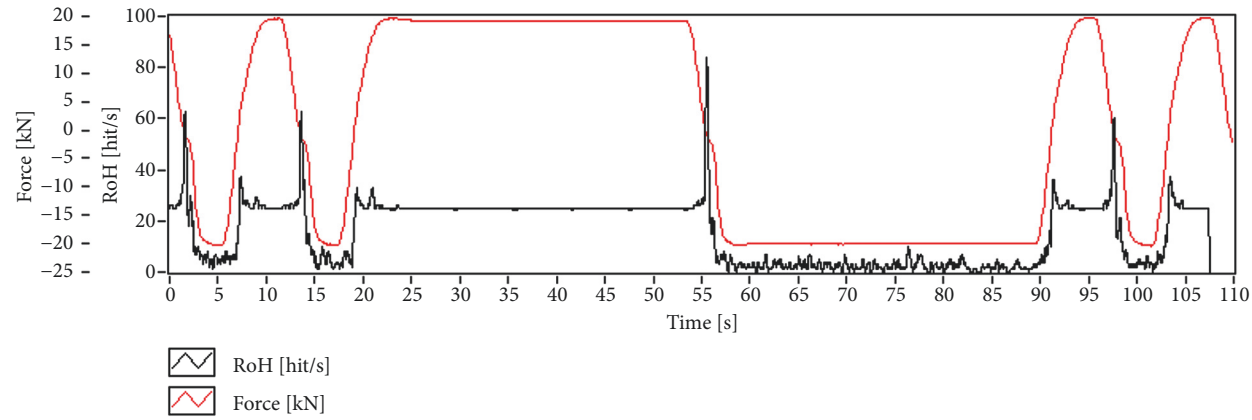

Figure 6: Rate of detected events (black) and the applied tensile and compression forces (red) during fatigue test.

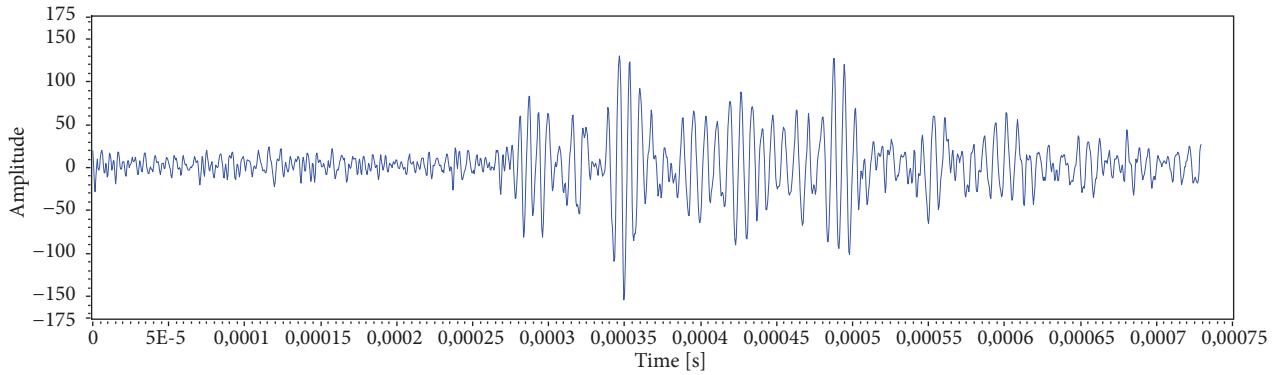

Figure 7: One of the detected ultrasonic bursts detected during fatigue test.

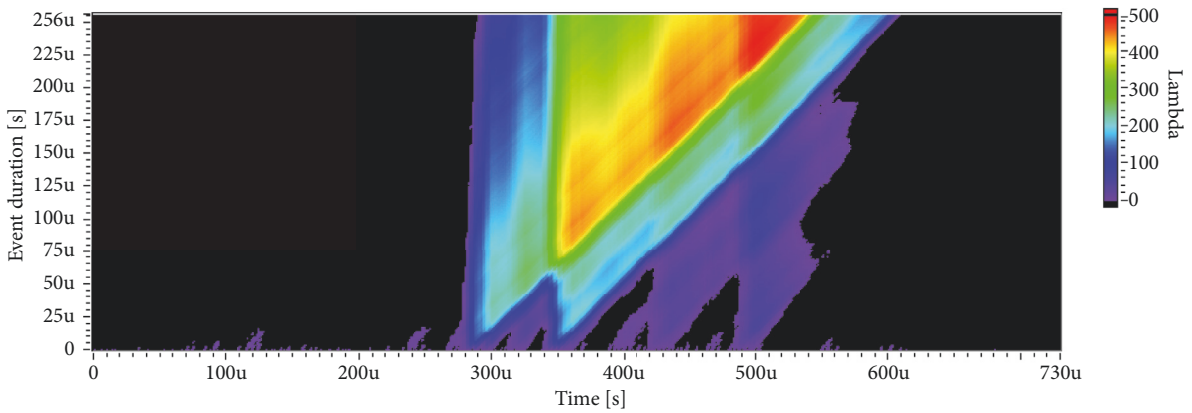

Figure 8: CSA result of detected AE event [502.7; 237.8; 490.4].

occur more frequently than expected by the theory or if the distribution does not match the hypotheses, the SSPRT will still be able to detect events. It remains also possible to analyse internal structure of events. The disadvantage of periodically appearing changes is that decision making can be done at a lower significance level. By SSPRT evaluation of vibration acceleration of an internal combustion engine, it is easy to detect combustion processes of the cylinders. If the SSPRT method can reliably detect all cylinder activities, further conclusions can be drawn. For example combustion frequency is proportional to the engines rotational speed. By displaying and processing the resulting three-dimensional data, detailed information can be obtained about the performance of combustion processes in each cylinder. Knowing the exact position of the crankshaft gives the possibility of resampling signals by angular rotation and then cyclostationary diagnostic tools can be easily applied. The results of such algorithms can be displayed in polar

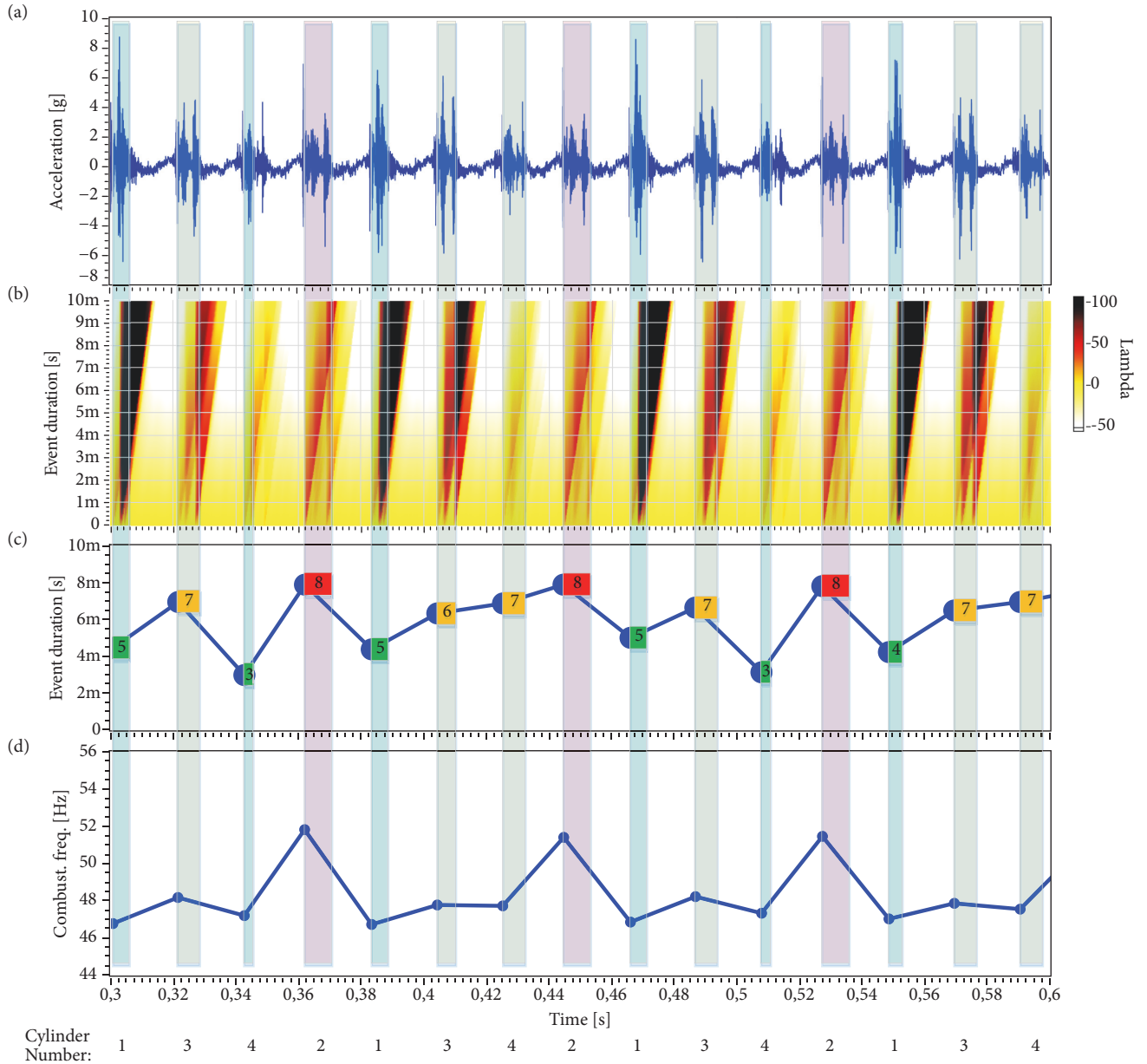

FIGURE 9: Synchronous display of a four-cylinder gas motor measurement and SSPRT results: measured vibration acceleration of the engines surface (a), linear scaled CSA calculation results (b), determined event duration (c), determined combustion frequency (d).

diagrams, which offer further possibilities of measurement evaluation.

In Figure 9 the raw vibration acceleration data series and the derived results appear simultaneously. Event lengths resulted by SSPRT are illustrated as transparent vertical bars. Depending on the event length, different colored bars appear in the graph: short events are green (2 to 5 ms); yellow bars sign medium length (6 to 7 ms); the longest ones appear in red (8 ms). In the (c) graph the determined combustion frequency can be seen. This frequency can be easily converted to rotational speed. The experimental setup contained a high precision encoder as reference sensor. Average speed of the gas motor was 1455 rpm which equals the 48.5 Hz average combustion frequency resulted by SSPRT.

5. Conclusions

This article presents a new technique, which enhances the evaluation method of the classical sequential probability ratio test. This novel method is based on realization of parallel fixed size cumulative sum calculations with different predefined sizes on the calculated log likelihood ratio. The resulting data is called cumulative sum array and the values can be visualized as a three-dimensional surface. This paper has shown three alternative ways, how the same cumulative sum array values can be calculated. Some special advantages and disadvantages of these alternative ways of the calculation have been also discussed. Many peak detection methods could be used to evaluate the resulting data of

the cumulative sum array; however, the authors suggest specialised peak detection algorithm. It has been developed to evaluate the log likelihood ratio based cumulative sum array sequentially. If a local peak has been found, the coordinates of the peak show the queried properties of the event. Rows correspond to a given event length, columns show when an event has been ended. Maximum values of the peaks give information about strength and significance. The proposed method can be readily used in practice. The successful applicability of the novel evaluation method has been first demonstrated on generated test signals where the noisy background contained short sections with additional noise. Thereafter, practical examples have been presented through acoustic emission and vibration diagnostic applications. The method can also be useful by cumulated sum array visualization on color maps. Graphical representation makes it much easier to compare detected events via giving a good visual overlook of the events' internal structure.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research reported in this paper was supported by the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of Artificial Intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI/FM).

References

[1] A. R. Huebner and A. D. Fina, "The stochastically curtailed generalized likelihood ratio: a new termination criterion for variable-length computerized classification tests," *Behavior Research Methods*, vol. 47, no. 2, pp. 549–561, 2015.

[2] Y. Ou, Z. Wu, M. B. Khoo, and N. Chen, "A rational sequential probability ratio test control chart for monitoring process shifts in mean and variance," *Journal of Statistical Computation and Simulation*, vol. 85, no. 9, pp. 1765–1781, 2015.

[3] B. Liu, J. Lan, and X. R. Li, "A 2-SPRT based approach to multiple-model hypothesis testing for multi-distribution detection," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3221–3236, 2016.

[4] H. Kim and D. R. Jeske, "Truncated SPRTs with application to multivariate normal data," *Sequential Analysis. Design Methods & Applications*, vol. 36, no. 2, pp. 251–277, 2017.

[5] S. Laxminarayan, M. J. Buller, W. J. Tharion, and J. Reifman, "Human core temperature prediction for heat-injury prevention," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 883–891, 2015.

[6] M. Berta, M. Szutyányi, A. Bencze, M. Hron, and R. Pánek, "Automatic ELM detection using gSPRT on the COMPASS tokamak," *Fusion Engineering and Design*, vol. 123, pp. 950–954, 2017.

[7] N. I. Spanos, J. S. Sakellariou, and S. D. Fassois, "Exploring the limits of the truncated SPRT method for vibration-response-only damage diagnosis in a lab-scale wind turbine jacket foundation structure," *Procedia Engineering*, vol. 199, pp. 2066–2071, 2017.

[8] X. Li, J. Cai, H. Zuo, X. Chen, H. Mao, and Y. Xu, "Reliability sequential compliance method for a partially observable gear system subject to vibration monitoring," *Journal of Vibroengineering*, vol. 19, no. 5, pp. 3313–3334, 2017.

[9] S. Li and X. Wang, "Fully distributed sequential hypothesis testing: algorithms and asymptotic analyses," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 64, no. 4, part 1, pp. 2742–2758, 2018.

[10] A. Bencze, M. Berta, A. Buzas et al., "Characterization of edge and scrape-off layer fluctuations using the fast Li-BES system on COMPASS," Plasma Physics, New York, 2019, https://arxiv.org/abs/1901.08043v2.

[11] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 231, no. 694-706, pp. 289–337, 1933.

[12] A. Wald, "Sequential tests of statistical hypotheses," *Annals of Mathematical Statistics*, vol. 16, no. 2, pp. 117–186, 1945.

[13] N. Mukhopadhyay and B. M. de Silva, "Theory and applications of a new methodology for the random sequential probability ratio test," *Statistical Methodology*, vol. 5, no. 5, pp. 424–453, 2008.

[14] G. Manhertz, G. Gardonyi, and G. Por, "Managing measured vibration data for malfunction detection of an assembled mechanical coupling," *The International Journal of Advanced Manufacturing Technology*, vol. 75, no. 5-8, pp. 693–703, 2014.

[15] T. Dobján, S. Pletl, T. Deák, L. Doszpod, and G. Pór, "Identification of the place and materials of knocking objects in flow induced vibration," *Acta Cybernetica*, vol. 20, no. 1, pp. 53–67, 2011.

[16] G. Manhertz, G. Csicso, G. Gardonyi, and G. Por, "Real-time acoustic emission event detection with data evaluation for supporting material research," in *in Proccedings of the 31st Conference of the European Working Group on Acoustic Emission (EWGAE)*, pp. 1–8, 2014.

[17] A. Tartakovsky, I. Nikiforov, and M. Basseville, *Sequential analysis: Hypothesis testing and changepoint detection*, 2014.

[18] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, pp. 100–114, 1954.

[19] D. Nikovski and A. Jain, "Fast adaptive algorithms for abrupt change detection," *Machine Learning*, vol. 79, no. 3, pp. 283–306, 2010.

[20] M. Harris, S. Sengupta, and J. D. Owens, "Parallel prefix sum (scan) with cuda mark," *Gpu Gems 3*, pp. 1–24, 2007.

[21] G. Por, G. Csicso, Zs. Danka et al., "Acoustic events detected during tensile testing of TWIP steels," in *Proceedings of the 11th European Conference on NDT*, pp. 1–10, 2014.

[22] I. Baran, M. Nowak, and W. Darski, "Application of acoustic emission in monitoring of failure in slide bearings," *JAE: Journal of Acoustic Emission*, vol. 25, pp. 341–347, 2007.

[23] G. Manthei, C. Alter, and S. Kolling, "Localisation of initial cracks in laminated glass using acoustic emission analysis – Part I," in *Proceedings of the In 31st Conference of the European Working Group on Acoustic Emission (EWGAE)*, pp. 1–9, 2014.