BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **20**, No 5
Special issue on Innovations in Intelligent Systems and Applications
Sofia • 2020 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2020-0038

Self-Similar Decomposition of Digital Signals

Kiril M. Alexiev[1], *Teodor G. Toshkov*[1], *Dimiter P. Prodanov*[1,2]

[1]*IICT, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria*
[2]*EHS and NERF, IMEC, Leuven, Belgium*
E-mails: alexiev@bas.bg teodor.toshkov95@gmail.com Dimiter.Prodanov@imec.be

***Abstract*:** *Traditionally, the engineers analyze signals in the time domain and in the frequency domain. These signal representations discover different signal characteristics and in many cases, the exploration of a single signal presentation is not sufficient. In the present paper, a new self-similar decomposition of digital signals is proposed. Unlike some well-known approaches, the newly proposed method for signal decomposition and description does not use pre-selected templates such as sine waves, wavelets, etc. It is realized in time domain but at the same time, it contains information about frequency signal characteristics. Good multiscale characteristics of the algorithm being proposed are demonstrated in a series of examples. It can be used for compact signal presentation, restoration of distorted signals, event detection, localization, etc. The method is also suitable for description of highly repetitive continuous and digital signals.*

***Keywords*:** *Self-similarity, digital signal decomposition.*

1. Introduction

Sensors are the devices by which a process, a physical phenomenon or an object is observed. These devices encode the physical quantity being measured into signals. The signals as output of the sensors contain not only useful information, but also a lot of additional information related to the sensor device, namely: the process of measurement, transformation of one physical quantity into another, integration, sampling (a set of time instants at which the signal has value), digitization (the act of conversion of analog values to digital ones), time delay of the sensor output from the actual timing of phenomena being sensed, electronic or other type of noise in the sensor. Another part of the output information is generated by external factors such as noise due to the effects of the external environment such as electric and magnetic fields among many. All this requires careful selection of the way the output signal is presented in order to fulfill both requirements: 1) transmit information about the observed process as accurately as possible; 2) reduce the influence of undesirable factors on the useful information.

The purpose of the signal analysis is to collect, retrieve and understand the information carried by the signals. It involves different type of signal processing: in time domain, in frequency domain, and/or time-frequency domain.

In the time domain, the signal is represented as variation of the amplitude through time. Analyzing signals in the time domain, we are able to detect signal events, eliminate/reduce interfering factors and amplify preferred signal features.

However, sometimes the most interesting information is how many times an event occurs in the recorded period. The Fourier representation theorem [1] gives the mathematical foundation for extracting the signal frequency content. The signal is presented as a sum of sin waves with given amplitude, period/frequency and phase. In the frequency domain, we observe the signal amplitude/power versus frequency. However, spectral representation is associated with the loss of time information for signal events in the data being processed. In other words, this means that a part of signal (its cut in time) of a certain shape generates the same spectrum no matter where it is – at the beginning of the investigated signal, in the middle, or somewhere at the end of the signal, the signal spectrum displays the total frequency properties throughout the interval. Therefore, the spectral representation of the signal does not provide information on the instantaneous (local) frequency characteristics of the signal.

Another very important statement is that the main characteristics in each signal can be represented by a spectrum with a limited range. The Nyquist – Kotelikov theorem states that for a complete representation of a particular signal, the sampling frequency must be at least twice the frequency of the portion of the signal with the highest bandwidth. A valuable tool for spectrum analysis implementation is the Fast Fourier Transform (FFT) algorithm [2]. FFT is computationally efficient and provides acceptable results for discrete signals of limited length, but it has some fundamental drawbacks. The first one is narrow frequency resolution. It is in a reciprocal relationship to the sampling time interval. In other words, the spectral resolution does not depend in any way on the type of signals under consideration but only on the sampling rate. The second limitation is related to the "leakage" of the energy from the main lobe of the spectral band to the side lobes. This leads to distortion of the other spectral bands. One possible solution to this problem, or at least to reduce these distortions in the spectrum, is the implementation of window functions. This decreases the leakage at the expense of a lower frequency resolution [3]. The disadvantages of FFT are particularly pronounced in the processing of short signals. To avoid this, the signal is often artificially lengthened by zeros or copies of itself to facilitate the resolution of the problems described above. Another disadvantage of operating in the spectral domain is the problem, related to the presentation of discontinues functions (i.e. with an infinite slope at certain time instants). In this case, an infinitely large number of terms is required in order to represent properly these signals.

In addition to the standard signal representation by a sum of periodic sine waves, there is also a signal representation by local support functions. Wavelet transform is one example for signal presentation by a certain orthonormal series generated by a wavelet. The main problem of wave analysis is the choice of a basis function. Appropriate representation could only be realized if we knew the properties of the signal being observed. With this signal representation, we get information about the frequency of waves used over time (the representation of signals in the time-

frequency domain). The Heisenberg uncertainty principle applies to this signaling space – the more selective waves have a narrower compact support (less selective over time).Wavelets are essentially equivalent to a short-time FFT with a Gaussian window function. Their advantages are better description of specific non-sinusoidal signals and lower computer load.

In a number of cases, the above methods are unable to extract sufficiently descriptive information. Often, the search for "useful" information in huge amounts of data can be aided by segmenting the data. The key to successful data segmentation is to define correctly the boundaries of the segments. To this end, a more general model of signal pattern is usually created [4]. It includes various information such as signal shape, different signal statistics, autocorrelation of the signal, signal coherence, etc. [5].

Another important feature is the signal texture. It is used to describe regions with a repeating structure. Texture patterns can be predefined or adaptively selected from the signal being analyzed.

The approach proposed in this article attempts to combine the positive characteristics of the methods listed above – a clear description of the signal (as its presentation in time domain), obtaining an idea of signal frequency properties (as in frequency domain), extracting the texture of the signal (like texture analysis). To do this, the signal is segmented into parts with monotonous behavior. Then these segments are compared for similarity. If there are similar distinct sets of consecutive segments the segments from these sets are merged into larger ones. This process is repeated until the whole signal is divided into distinct repetitive segments. The proposed approach does not require pre-selected patterns. In addition, the presence of abrupt sudden changes in the signal derivative does not disturb the algorithm. This enables the newly proposed Self Similar Decomposition (SSD) method to be applied to digitally encoded signals. They can be presented in a very compressed form.

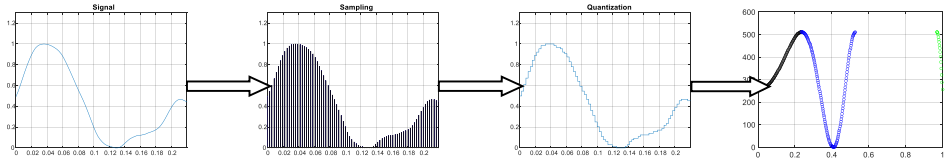
The rest of the paper is organized as follows: A description of SSD algorithm is given in the next Section 2. The algorithm is explained on a sinusoidal function. Then the possible similarity metrics are discussed. The section ends with a description of the proposed segmentation procedure. Some experimental results are presented in Section 3. The selected examples demonstrate the main properties of the SSD algorithm. Section 4 discusses noise vulnerability of suggested algorithm and points out three main direction for SSD implementation. The computer load for real time implementation is estimated. The final 5-th Section gives a brief summary and conclusions.

2. Self-similar decomposition of signals

A signal is generated by a sensor in order to describe a measured physical process. Although the sensor signal and the observed process behavior are closely related, some distinctions are stemming from non-ideal acquisition process and they have to be considered, so as to correctly estimate the process parameters. How they can be reflected will be revealed in the next subsection. Usually, the sensor output is voltage that varies over time. The analogue sensor voltage is sampled in time by a predefined rate in order not to miss important changes in the process observed. The sampling procedure converts the continuous independent variable (usually it is time) of sensor

signal to discrete. Then the acquired measurements are quantized by Analog-to-Digital Convertors (ADC) using a limited number of bits. The number of used bits determines the distance between adjacent quantization levels for signal presentation and influences the quantization error. The digital output data of ADC serves as an input of the SSD Algorithm.

The place of the SSD Algorithm in the general scheme of signal processing is shown on Fig. 1.



1. Raw sensor signal 2. Sampling 3. Quantization 4. SSD

Fig. 1. General scheme of signal processing with SSD Algorithm

The proposed algorithm for signal description is an attempt to combine the representation of the signal in both time and frequency domains. The basic idea is the same as for frequency and wavelet analyzes – to decompose the signal into multiple repeating fragments of itself. The difference is that no pre-set functions are used – sinusoids (in frequency analysis) or different basic wavelets (in wavelet analysis). Using some patterns of signal is not a new idea. There are two different approaches to describing signals. The first of them uses a set of a priori defined signal forms or dictionaries [4, 6, 7]. The received signal is usually divided into equal parts (at regular time intervals). Each part is classified according to its proximity to any of the templates. The second approach stems from fractal analysis, which seeks out the self-similarity of a signal on different scales. This approach has limited application due to the relatively small number of fractal signals in practice.

When looking for similarity, one of the most important steps is splitting the signal (segmenting). Different approaches have been described in the literature [4, 5, 7, 8]. The most popular between them are similarity-based thresholding, histogram-based thresholding and template matching. All of them presume a priori knowledge of some of signal characteristics that strongly restricts the flexibility of the chosen approach.

The approach proposed in the present work can be considered as an extension of the fractal approach to signals of general form. No information is needed in advance. The proposed algorithm also attempts to group similar parts into connected regions.

2.1. SSD Algorithm description

The structure of SSD Algorithm is shown on Fig. 2.

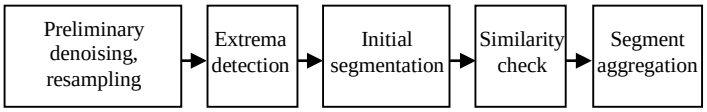


Fig. 2. Steps of SSD Algorithm

The first step of the algorithm is preliminary denoising and resampling, if necessary.

Denoising is one of the most important stages of the algorithm. The noise inevitably accompanies the measurement process. It can be generated by random or unpredictable outside sources in measurement process, unwanted sensor signal disturbance caused by imperfect sensor electronics, by inappropriate sampling rate or wrongly chosen ADC. The presence of high levels of noise creates artificial extrema, which makes it difficult to detect actual ones, or at least leads to the localization of extrema with significant error. Many methods have been developed to minimize signal noise [9].

The classic analog noise filtering approaches are implemented in the form of low pass, high pass or band-pass/band-reject filters. The most popular among them are Bessel, Butterworth, Chebyshev filters, to mention a few.

Digital filters provide significantly better performance in comparison with analog ones. The easiest way to implement a digital filter is by convolving the input signal with a filter kernel. This type of digital filters is also called FIR filter because of finite response from a filter kernel on a delta impulse.

Another type of digital filter is the recursive one, which uses one or more previously calculated points beside the points from the input measurement stream. The recursive filter design considers a suitable choice of a set of recursion coefficients. The impulse response of a recursive filter is infinitely long (IIR filter). However, when the amplitude of filter kernel drops below the quantization noise or round off error, the remaining part can be ignored.

The model-based filters are a big group of filters, using a priori information about observed process or system behavior. The parameters of the model equations are dynamically estimated in time maximizing the model consistency against received measurements. The most popular among these filters are AR, ARMA, ARIMA, Kalman, Extended Kalman filters and their numerous modifications.

The abundance of filtering algorithms described above offers an opportunity for an optimal choice of the processing algorithm for given sensor signal and a specific type of noise. After denoising, the resulting output signal has to reflect the behavior of the observed process as accurately as possible.

In order to enhance the accuracy in the segments' borders estimation, an upsampling and/or interpolation is performed. Different interpolation schemes can be applied. Linear interpolation is the best one for singularity signals (pulse train signals, rectangular, sawtooth, etc.). Spline interpolation is more suitable for smoother signals. Other types for interpolation can also be applied. The choice of suitable interpolation scheme may increase the computational cost, but higher accuracy of peak localization will be achieved. It is important to note that only an appropriate selection of interpolating functions and a suitable upsampling rate can improve the accuracy of the segmentation process.

The upsampling indirectly resolves the problem with non-uniformly sampled signals. Non-uniform sampling occurs often in real life due to imperfect sensors, bad communication lines (loss of measurements), external disturbances, mismatched clocks, etc.

Before we start with the second step of SSD Algorithm, let us reveal the main idea of the algorithm. It is based on the hypothesis that the most important points in the observed signal from information point of view are its extrema. The information can be encoded in a signal in many different ways and in different domains. There are countless signal modulation techniques. Sometimes the information is hidden in the frequency domain, when the periodicity or the phase of some processes are described. But regardless the way of information presentation, it is encoded into the changes of the signal. The changes of a signal can be determined only by comparing the values of many points in the signal (their relationship). If we consider any two neighboring points in the signal (the smallest set to compare), there are only three possible states – preserving the same value, growing or decreasing of the signal. The set of three neighboring points is a richer one, giving means to detect the transition of the signal from one state to another. We consider the signal state-change points as the most important information carriers and the proposed algorithm is built entirely on their use. These points are signal extrema, playing the key role for signal segmentation – they are the border points of the segments. The signal behavior on each segment remains the same – there is no change in the signal state. The accuracy of signal extrema localization is important for the quality of the segmentation.

In the third step of the algorithm, the signal is divided into non-overlapping parts. Preparations for this action have been made at the previous step, when the signal extrema were determined. These extrema serve as boundary points of the segments. The proposed segmentation algorithm is characterized by two features:

1) The length of the segment is determined only by the time between boundary extrema of this segment.

2) The signal in each segment is characterized by its monotonicity. This fact directly follows from the fact that the segment is a signal bounded by two adjacent extrema: either from one minimum to one maximum or from one maximum to one minimum.

Two more segments are added to the segments thus defined – the first part of the signal (before the first extremum) and the last part of the signal (after the last extremum). The second property of the segments also applies to them – they are also monotonic functions.

The original signal can be easily recovered from the segments. Recovery algorithms can use either the start time of the corresponding segment or the order of the segments. The first algorithm is trivial, but the second should be explained in detail. Some denotations are involved for explanation of signal reconstruction by list of its segments. Let $S = \{s_1^{m_1}, s_2^{m_2}, \dots, s_n^{m_n}\}$, denote the set of segments describing the signal. A segment $s_i^{m_i}$ is indexed in the set S with its number i and the number of repetitions m_i. The signal can be reconstructed unambiguously if it is described by the list of its segments: $j_1, j_2, \dots, j_k$, where $j_i \in \{1, 2, \dots, n\}$, and $k = m_1 + m_2 + \cdots + m_n$.

Finding the repeatability of a segment is the subject of the fourth step of the algorithm. This is the most delicate step of the algorithm, which determines the degree of compression and on the other hand the quality of signal recovery after its decomposition. There are different approaches for determining the similarity estimation of two segments: matched filtering, correlation analysis, subtracting

signals, beam formation, etc. [10] to mention at least the main ones. The emphasis in the development of SSD is on its multiscale properties both in the magnitude of the signal amplitude and over time.

The last step of the SSD algorithm consists of segment aggregation. It is applied over repeated two or more adjacent segments. These successive segments are combined as a supersegment and in the set describing the given signal are written as one element. In this way all found similar segments of double, triple, quadruple, ... *n*-tuples are aggregated.

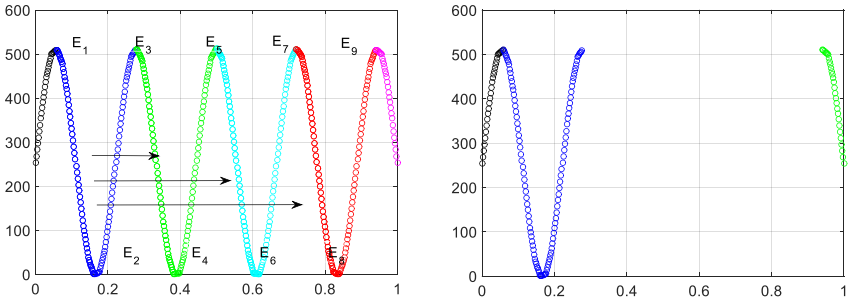
2.2. Recursive SSD Algorithm description on a demonstration example of sinusoidal signal

The recursive mode of operation of the SSD Algorithm is demonstrated on a sinusoidal signal shown in Fig.3. Due to the use of a synthetic signal, noise filtering and resampling is not required, and this step is omitted. In the next step, all the extrema of the function, denoted by E, are determined. The index of the extrema denotes their sequential number in the signal. The extrema define the initial set of segments: $S_1 = \{(E_1, E_2), (E_2, E_3), \dots, (E_{n-1}, E_n)\}$. In this description, the index of the set of segments means the level (depth) of the recursive procedure. Each segment is described by the ordered pair of extrema (E_1, E_2). For our demonstration example, the set is $S_1 = \{(E_1, E_2), (E_2, E_3), \dots, (E_8, E_9)\}$. The fourth step of the algorithm consists in finding similar segments, according to a given criterion. The search for close segments is performed in the set S_1, selecting the segments sequentially according to their order in the signal. In our example, the process of determining close segments begins with the first detected segment (E_1, E_2). When the criterion for similarity of the tested segment is not met even once for its pairing with all other segments in the set of segments, the check cycle ends, and this segment is left as an unique segment. Then the algorithm moves on to the next element of the set of segments. However, if the criterion is met, a new set of segments is formed at the next recursion level. This is done in several steps.

First, the similar elements are indexed as used so that they are no longer used for checks at the same recursion level. In our example, four close segments are found $(E_1, E_2), (E_3, E_4), (E_5, E_6), (E_7, E_8)$(see Fig.3a) and based on them, a new set of segments is formed.

The second step is to define the segments in the new recursion set. It consists of non-indexed (unused elements) that immediately follow the found similar elements. In our example, $S_2 = \{(E_2, E_3), (E_4, E_5), (E_6, E_7), (E_8, E_9)\}$ (Fig. 3a). For this set, the recursion procedure is called again, which looks for similar elements in the new segment set.

These steps are repeated until the check passes all segments in the original set of segments or the set of non-indexed segments becomes empty. For our example, the similarity of the segments in S_2 is also detected and thus the all procedure is closed due to the empty set of non-indexed segments.



(a) Segmentation process (b) Three primitive shapes describe the signal

Fig. 3. An example for SSD Algorithm for sinusoidal function

Summarizing the above, the first pass of the recursive procedure ends with the following initial results: 1) Detects similar segments $(E_1, E_2), (E_3, E_4), (E_5, E_6), (E_7, E_8)$. 2) The set of segments $S_2 = \{(E_2, E_3), (E_4, E_5), (E_6, E_7), (E_8, E_9)\}$ is formed and is subject to further analysis. Then the same recursive procedure is called again with S_2 passed as input parameters. In the second call (at Level 2) of the recursive procedure, similar actions are performed but with the new set of segments. In our example, similar segments are searched for, starting with the first segment in the set according order in the signal (E_2, E_3). If the algorithm does not find similar to this one, it is declared as a unique one and proceeds to the next segment. In the sinusoidal signal example, however, the algorithm finds three similar segments $(E_4, E_5), (E_6, E_7)$, and (E_8, E_9). When a similarity is found, each recursion level except the first one starts a procedure for aggregation of those successive segments for which the corresponding similar sequence of segments were found. In our example there are 4 such "supersegments" – $(E_1, E_3), (E_3, E_5), (E_5, E_7), (E_7, E_9)$.

The result of the operation of the algorithm on a sinusoidal signal is shown on Fig. 3b. The sinusoidal signal is described by three segments: the first of the segments represents the beginning of the signal to the first extremum, the second segment is actually a "supersegment" and describes the sine wave from one maximum to the next maximum (completely unambiguously determines the sinusoidal signal), the third segment is the part of the signal after the last extremum to the end.

2.3. Similarity measures

The proximity measure of two segments determines whether two segments can be represented by only one of them or not. Similarity as a measure is defined in mathematics on the basis of congruence – congruent objects are those that can be represented as linear transformations of one another, i.e., translation, rotation and reflection. When working with real signals, however, because of the presence of different interferences and noise, the signals can only be compared to some degree of similarity. In addition, the present work seeks such proximity measures that allow correct work on different scales, i.e., we are looking for similar, not identical segments. For this case of similarity, we speak of congruence up to a dilation. When we talk about the similarity of signals with precision to scale, we must take into

account both the scale of the amplitude of the signal and the scale in time (signal shrinking).

There are many methods for similarity estimation. The most popular of them is correlation analysis. It detects linear functional dependency only. In the case of nonlinear dependency, the methods using multi-moment measures are applied: the mutual information approach [10] and its generalizations, such as dual total correlation [11], excess entropy [12], binding information [13], etc. In [14] the authors suggest a copula for more complex dependency analysis.

In the present work we limit search only for proximity of multiscale signals. The linear regression analysis is an effective tool for solving this problem. The correlation coefficient is an estimate for the proximity of the two segments, and the slope of the linear regression line determines the scale factor. However, the direct application of this method will lead to a significant error due to the phase shift or different time scale. To overcome this shortcoming, the present algorithm resembles the measurements in such a way that there is a correspondence between the measurements in the two compared segments. Two cases are considered:

1) Resampling without rescaling over time. In this case, it makes sense to compare segments of equal duration (over time). Therefore, the length of the segments (the number of measurements in them) is checked in advance and if it is different, the segments are considered to be different. In this way, a significant acceleration of the algorithm is achieved. When the same number of points is discovered, a resampling is performed, which equalizes the moments of time in which the values of the signals in the two segments are compared.

2) Resampling with rescaling over time. We will introduce some notations. Let $\mathrm{seg}_1(t)$ and $\mathrm{seg}_2(t)$ denote the first and second segments for similarity check. They are defined in time as follows: $\mathrm{seg}_1(t)$, $t \in [t_{b_1}, t_{e_1}]$ and $\mathrm{seg}_2(t)$, $t \in [t_{b_2}, t_{e_2}]$. In the general case, two segments can differ both in phase and in number of measurements (they can be of different duration). Let us consider the first segment as a basic and coordinate the measurements of the second segment to those of the basic one. This is realized by resampling the measurements of the second segment with a rating $m_{t_{1,2}} = \frac{t_{e_1} - t_{b_1}}{t_{e_2} - t_{b_2}}$. After resampling, the two segments are ready for similarity check.

Different criteria for similarity check can be applied. Let us look at them in detail. Firstly the points from the two processed segments are denoted as: in segment $\mathrm{seg}_1(t)$ as $s_{1_1}, s_{1_2}, \dots, s_{1_N}$ and the points of the signal in segment $\mathrm{seg}_2(t)$ as $\mathrm{s}_{2_1}, \mathrm{s}_{2_2}, \dots, \mathrm{s}_{2_\mathrm{N}}$.

The Euclidean distance is the most popular one. It is computed as follows:

$$d_2 = \sqrt{\sum_{i=1}^{N} \left(s_{2_i} - s_{1_i}\right)^2}.$$

Two other modifications have also been often used – Manhattan distance and supreme norm. Manhattan distance is L_1 norm and requires fewer computations:

$$d_1 = \sum_{i=1}^{N} |s_{2_i} - s_{1_i}|.$$

Supreme norm (Chebyshev distance) is determined by the maximal distance of corresponding measurements:

$$d_{\max} = \max|s_{2_i} - s_{1_i}|, \ i = 1, \dots, N.$$

The Minkowski distance [16] is a generalization of the distances, described above:

$$d_p = \sqrt[p]{\sum_{i=1}^{N} (s_{2_i} - s_{1_i})^p}.$$

If p=1, Minkowski distance transforms to Manhattan distance, while if p=2, it becomes Euclidean and when $p \to \infty$ it will be equivalent to Chebishev distance.

All these distances are classical ones, but they cannot be applied when one of the compared signals is scaled or shifted.

The Hausdorff distance [17, 18, 19] on finite sets is defined by:

$$d_{\mathrm{H}} = \max_{s_{1_i} \in S_1} \left\{ \min_{s_{2_j} \in S_2} \left\{ d\left(s_{1_i}, s_{2_j}\right) \right\} \right\}.$$

The Hausdorff distance might allow some shift between measurements, belonging to the compared segments; some scaling and skew might also be allowed.

If we want to consider the scale of the data, normalization has to be applied. The Mahalanobis distance criterion estimates the closeness, regarding normally distributed data:

$$d_{\mathrm{M}} = \sqrt{\frac{(\overline{s_2} - \overline{s_1})^2}{\frac{1}{N}\sum_{i=1}^{N} (s_{1_i} - \overline{s_1})(s_{2_i} - \overline{s_2})}}.$$

In calculation of Mahalanobis distance, the amplitude scale is normalized by weighting the difference between the means of the two segments' covariance matrix. Therefore, the Mahalanobis distance is scale invariant and it is a unitless measure, but the scale is measured using standard deviation. However, the problem of neglecting the presence of a constant composite in the measurements of one of the segments remains.

Correlation analysis can also be used to estimate the similarity of two segments. It is applied over normalized functions. It is important to note that the normalization leads to multiscale capabilities of correlation analysis. Proportionally modified signals will be identified as being the same. The main parameter that is calculated is the correlation coefficient. A correlation coefficient with a value equal to 1 shows a perfect match (similarity) of the compared segments. Two uncorrelated signals will have a correlation coefficient equal to 0. Correlation analysis can also be applied to detect symmetric signals. Signals with mirror symmetry will have a correlation coefficient equal to -1. To calculate this coefficient, four correlation metrics are used – Pearson, Spearman, point-biserial and Kendall rank correlation.

The Pearson correlation coefficient is calculated as

$$R = \frac{N \sum_{i=1}^{N} s_{1_i} s_{2_i} - \left(\sum_{i=1}^{N} s_{1_i}\right)\left(\sum_{i=1}^{N} s_{2_i}\right)}{\sqrt{\left[N \sum_{i=1}^{N} s_{1_i}^2 - \left(\sum_{i=1}^{N} s_{1_i}\right)^2\right]\left[N \sum_{i=1}^{N} s_{2_i}^2 - \left(\sum_{i=1}^{N} s_{2_i}\right)^2\right]}}.$$

The Pearson correlation criterion is preferable for normally distributed $s_{1_1}, s_{1_2}, \dots, s_{1_N}$ and $s_{2_1}, s_{2_2}, \dots, s_{2_N}$. The biggest Pearson correlation criterion downside is that $R = 0$ does not imply independence. The implementation of the Pearson approach for similarity estimation avoids this drawback thanks to the fact that similarity estimation does not concern with the independence of data, but in similarity only.

The Spearman correlation coefficient is the usual Pearson correlation coefficient, applied to the ranked measurements. The proposed segmentation procedure assures that the compared segments' points are ranked, reducing Spearman correlation coefficient to Pearson criterion, given by the expression:

$$\rho = \frac{\sum_{i=1}^{N}(s_{1_i} - \overline{s_1})(s_{2_i} - \overline{s_2})}{\sqrt{\sum_{i=1}^{N}(s_{1_i} - \overline{s_1})^2 \sum_{i=1}^{N}(s_{2_i} - \overline{s_2})^2}},$$

here i denotes the index of measurements with tied ranks.

Point-biserial correlation coefficient can be considered as equivalent to Pearson correlation coefficient when one of the sequences obeys some additional restrictions. That is why it cannot be considered here.

Kendall rank correlation is measured using tau coefficient (τ) and it estimates the similarity of the orderings of the data in two sequences, ranked by each of the quantities. It could not be applied because of the monotonicity of the compared sequences, which will always give Kendal correlation coefficient equal to 1 or –1 (if time scale is considered). The only exception will occur when sequences contain constant parts, but even in this case, when the measurements are ranked by value and by time, the Kendal correlation coefficient will take one of the two values: 1 or –1.

Regardless of which distance metric or correlation coefficient is used, a detection threshold has to be set. The detection threshold has to guarantee affordable false alarm rate when the hypothesis for similarity is true. The algorithms that apply normalization are preferable for similarity estimation.

2.4. Signal segmentation

One of the fundamental problems of digital signal processing is signal segmentation. Signal segmentation can be considered as dividing the signal into parts using some criterion. The quality of segmentation is determined mainly by the proper choice of criterion and the precision of segmentation. The adequately selected criterion will preserve desirable characteristics of the signal in segments. The suitable choice of segmentation criterion for signal decomposition on self-similar parts is also extremely important. The simplest segmentation methods use a preselected temporal interval to determine a segment of a specified length [20]. The more complicated algorithms look for points, where the signal changes. These points are characterized by unique features or by a change in the statistical parameters of the signal. Both approaches can use a priori defined models or adaptive estimates of the signal parameters. Almost all articles devoted on adaptive segmentation use the construction of a moving "window". The main drawback of the moving "window" approach is hidden in the unfeasible requirement for continuous (smooth) movement. To overcome this problem, a more complicated criterion for segmentation is applied. It

estimates the segment border and the phase shift simultaneously [17, 18]. In spite of this, the realization of the algorithm becomes ineffective.

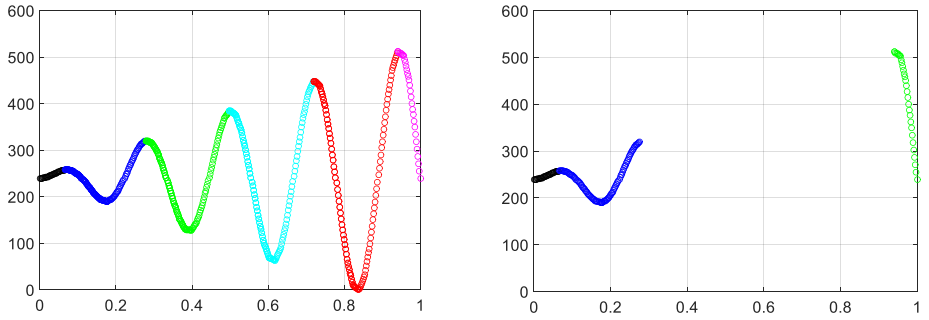
The proposed here SSD algorithm avoids these negative aspects by performing segmentation without using fixed-length windows or constructing any models. The signal partitioning is defined by the detected extrema. The number of the initial (smallest) segments corresponds to the number of the found extrema minus one. The parts of the signal before the first found extremum and after the last one are ignored, because the first and the last measurement could not be considered feature points. The formation of a set from initial segments is only the first step of the segmentation procedure. After that, the similarity search is carried out and a sequential consolidation/integration of the smallest segments in bigger ones takes place.

The segmentation process gives quite accurate results when the signal to noise ratio and the sampling rate are high enough. When these conditions are violated (small signal-to-noise ratio and/or low sampling rate), it will be especially difficult to estimate the segments' borders precisely and the whole procedure may fail. In order to alleviate the noise influence, a low-pass filter has to be applied. An appropriate choice of a low-pass filter will remove the noise partially or entirely and will make the signal smooth with no noise-generated extrema. The low-pass filtering in the proposed algorithm may be regarded as an analogue of anti-aliasing filter, restricting the bandwidth of a signal in order to satisfy the sampling theorem over a given band.

3. Experimental results

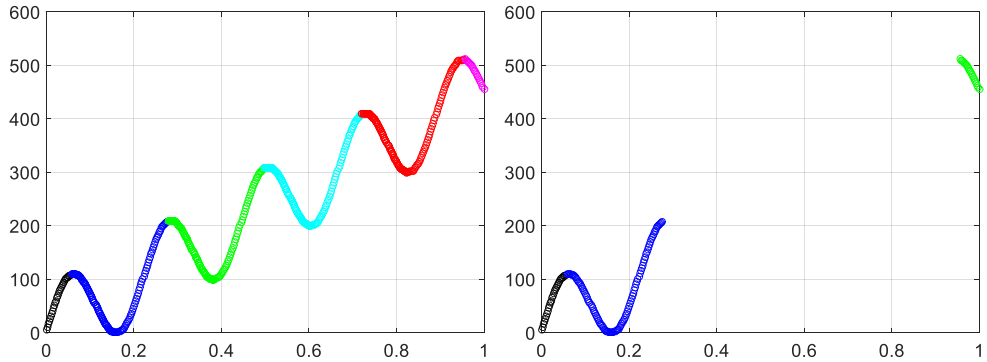
Several examples have been selected for consideration in this section to demonstrate the basic properties of the proposed SSD Algorithm. Most signals are specially synthesized and without noise. Finally, the SSD Algorithm is applied to a real ECG signal to show its capabilities for working with real signals.

After the demonstration of the algorithm on a sinusoidal signal in the descriptive part, here we will start the presentation of the algorithm on a sinusoidal signal with a linearly increasing amplitude (Fig. 4a). Despite the complex presentation of the signal, the SSD decomposes it again only with the help of a library of only three segments (Fig. 4b), demonstrating good scalability.



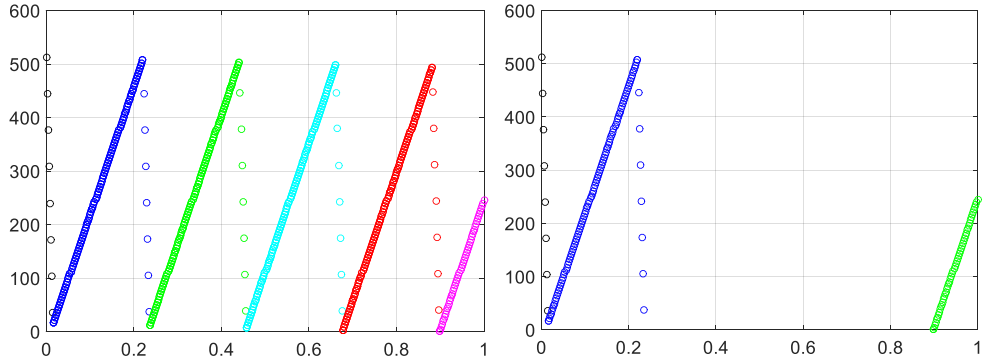
(a) Segmentation (b) Four primitive shapes describe the signal

Fig. 4. SSD Algorithm for sinusoidal function with linearly increasing amplitude



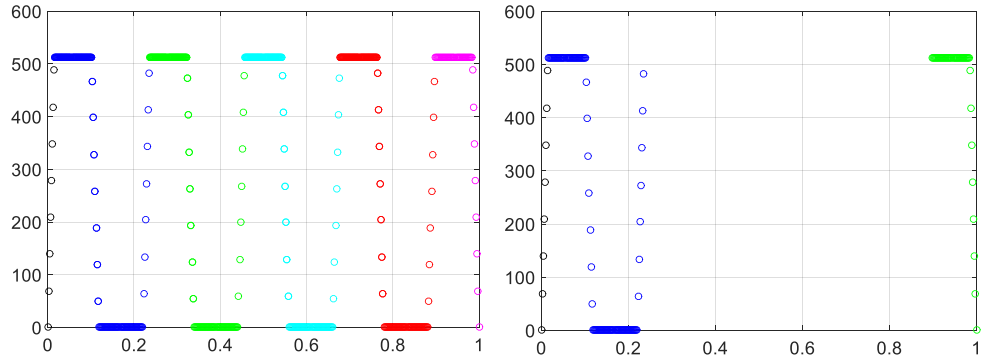
(a) Segmentation (b) Three primitive shapes describe the signal

Fig. 5. SSD Algorithm for sinusoidal with linear trend



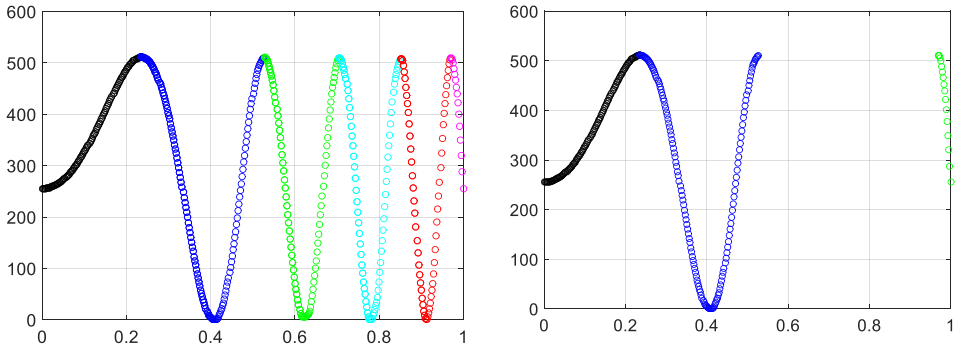
(a) Segmentation (b) Three primitive shapes describe the signal

Fig. 6. SSD Algorithm for sawtooth signal



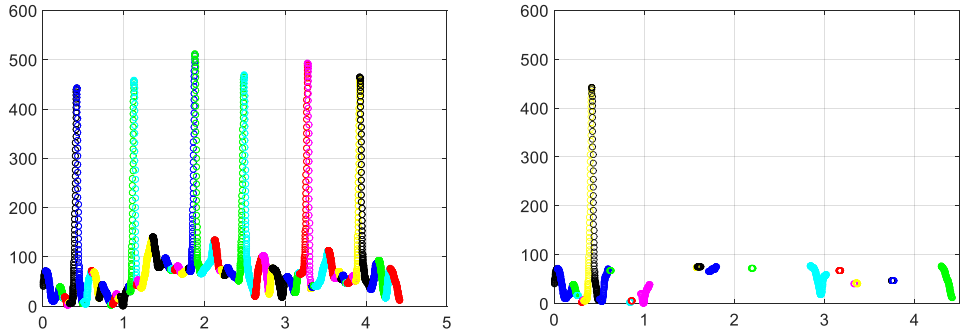
(a) Segmentation (b) Three primitive shapes, describing function

Fig. 7. SSD Algorithm for rectangular pulse train



(a) Segmentation (b) Four primitive shapes, describing function

Fig. 8. SSD Algorithm for sinusoidal function with time scale



(a) Segmentation (b) Primitive shapes, describing an ECG signal

Fig. 9. SSD Algorithm for an ECG signal

The following figure (Fig. 5) shows a sinusoidal signal with a constant trend. Despite the presence of a trend, the SSD Algorithm again compresses the signal only in a library of three segments (Fig. 5b).

Then two examples of digital signals are displayed. These signals are particularly unsuitable for description in the frequency domain due to the presence of interruptions in the signal derivative. A special feature in the processing of digital signals with SSD is the use of linear interpolation in resampling. Fig. 6 shows the decomposition of a sawtooth signal in a library of only three segments. Fig. 7 presents the results of the processing of a rectangular pulse train. The result is again with optimal performance - only three primitives. The examples clearly show the effective representation of digital signals, which are particularly convenient for SSD due to the high repeatability and limited set of the signal waveforms.

The next demonstration example is selected specifically with a frequency modulated sinusoidal signal. Such a signal requires rescaling over time. The SSD Algorithm again decomposes the signal only in a library of three primitives (Fig. 8).

The last example on Fig. 9 shows a SSD of a real bioelectrical signal, in this case ECG, which fluctuates and has noise. The algorithm successfully identifies the elements of the QRS complex (rising and falling slopes).

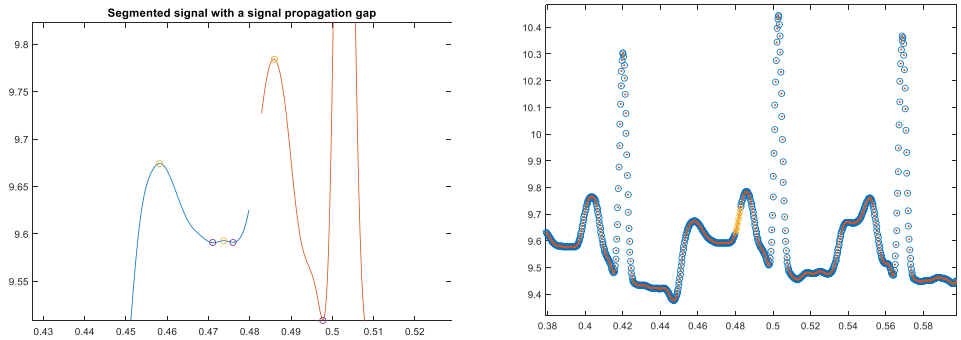
4. Discussion

The presented experimental results demonstrate the effectiveness of SSD in the decomposition of repetitive signals. The SSD performance is especially good when working with digital signals. So far, the question of the sensitivity of the algorithm to the presence of noise has not been discussed. The general scheme of decomposition application assumes noise filtering at the system input (Fig. 2, step 1). This is due to the fact that the presence of noise in the signal degrades the operation of the algorithm simultaneously in two of its steps. The first deterioration is at the stage of signal extrema detection. The presence of noise degrades the accuracy of extremum localization. Moreover, high noise levels can lead to an exponential increase in the number of extrema, which will completely destroy the operation of the algorithm. The second vulnerability to the presence of noise in the signal is in determining the similarity of the segments. In this case, the standard signal processing methods apply an adaptive threshold selection (according to the noise level). This approach is feasible for implementation in the current algorithm as well.

There are three main directions for further development of the algorithm.

The **first direction** is the signal representation in a compact form. The algorithm allows storing the most important signal features. A very high compression ratio can be achieved, especially for artificial (man-made) signals. Furthermore, it is due to say there are no limitations, given by the form of signals (continuous, singular). The signal presentation in this format contains simultaneously the main signal features and their local frequency. The algorithm is recursive and easily realizable even in real-time. Required computational resources are minimal and the performance speed is high. The encoded signal describes the texture of the original one.

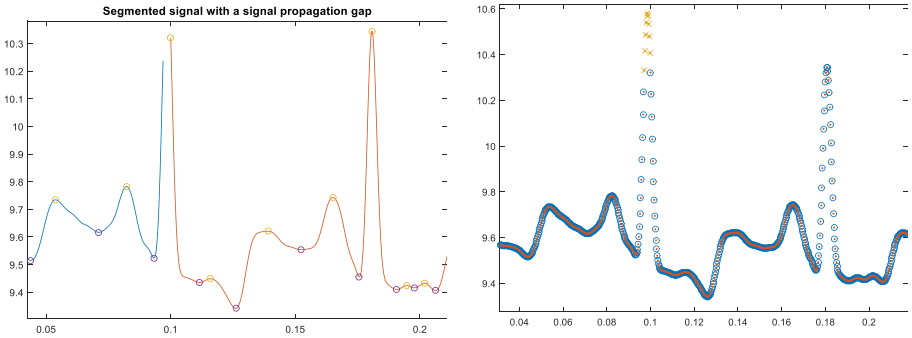
The **second direction** is algorithm implementation for the restoration of signals that have been distorted in some way. The telecommunications in ad hoc networking, IoT, satellite systems, etc. are the targeted areas for application. When transmitting data over more or less insecure channels, there is often a loss or damage to the transmitted information. Error-correcting encoding helps to recover information, especially if the losses are in single bits or in small packets of bits. To this end, it is assumed that a sufficient surplus of data will be sent to restore the original data in the event of damage or loss of a certain amount of data. Recovering information for larger packets of bits requires a much larger resource that not every system or communication channel can afford. The newly proposed algorithm provides means for signal restoration using the information about the signal texture. The packet of lost information can contain extremal points or not. When the lost information concerns only one segment (Fig. 10a) it can be recovered using the preserved information of the closest (similar) to it another segment (Fig. 10b). If the lost information in the segment predominate over the remaining correctly transmitted information in the segment, one could also use neighboring segments to find the closest preserved signal texture.



(a) Segmentation of a signal with losses (b) Restoration of signal

Fig. 10. Restoration of a signal with losses in a segment

When the packet of lost information contains one or more extremal point (Fig. 11a) the surrounding segments are included and the signal is restored by filling the gap with the closest to its vicinity preserved signal texture (Fig. 11b). It is important to note that here the closeness is regarded again up to scale.



(a) Segmentation of a signal with losses (b) Restoration of signal

Fig. 11. Restoration of a signal with losses in two segments

The SSD algorithm can also be applied for event detection and separation of signals (the **third direction** of development). Creating a library of mostly used segments or groups of segments, we can divide a signal into different parts, characterized with some specificity or different system/process behavior. The unique segments (without similar others) inform us about something unusual that happened at that moment. Thus, it may be used for registering events.

The description of the SSD algorithm shows its low requirements for computer resources. The computer load for filtration remains hidden so far. We will use the available literature data to calculate the required computer resource for filtration [15]. The filtering in time domain requires the performance of $O(n^2)$ real multiplications and $O(n^2)$ real additions to perform a convolution. When the filtering is realized in the frequency domain, FFT is performed on the signal and the convolution function, followed by $O(n)$ multiplications and inverse Fourier transform of the obtained results. As a result, the total complexity of the frequency domain filtering is

$O(n \times \log n)$. When using a wavelet transform for filtering, the computational complexity can be reduced to $O(n)$.

Summarizing the above, we can say that the computational complexity of SSD is highest when applying filtering in the time domain $O(n^2)$, less when filtering in the frequency domain $O(n \times \log n)$ and most effective when using wavelet filtering – $O(n)$. Analyzing this data, however, we must take into account the effectiveness of these algorithms to eliminate different types of noise.

5. Conclusion

This article proposes an approach for multi-scale associative signal decomposition. The approach is adaptive and does not work with pre-selected basic functions for signal description such as sinusoids, wavelets, etc. The proposed description is based on self-similar signal decomposition. The signal decomposition is performed in the time domain, but simultaneously an information is obtained about the repeatability of the basic elements, i.e., a frequency information is also gained. The proposed signal representation can be applied to the automatic detection of signals, for the purposes of classification, event localization, compression, etc. The approach is particularly effective when applied to repetitive signals of artificial origin like digital ones.

References

1. Fourier, J. The Analytical Theory of Heat. New York, Cosimo, Inc., 2007, ISBN: 978-1-60206-107-1, Originally Published in 1822.
2. Cooley, J. W., J. W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. – Mathematics of Computation, Vol. **19**, No 90, pp. 297-301. DOI:10.1090/S0025-5718-1965-0178586-1, ISSN 0025-5718.
3. Krivosheew, V. I. Contemporary Methods for Digital Signal Processing (Digital Spectral Analysis). Educational-Methodical Material for the Advanced Training Program "Modern Mobile Digital Communication Systems, Problems of Noise Immunity and Information Protection". Nijnii Novgorod, 2006 (in Russian).
4. Besl, P. J., R. C. Jain. Segmentation through Variable-Order Surface Fitting. – IEEE Trans. on PAMI, Vol. **IO**, March 1988, No 2, pp. 167-192.
5. Allen, R. L., D. W. Mills. Signal Analysis Time, Frequency, Scale, and Structure. Copyright © 2004 by the Institute of Electrical and Electronics Engineers, Inc., ISBN: 0-471-23441-9.
6. Bhanu, B. Representation and Shape Matching of 3-D Objects. – IEEE Trans. on PAMI, Vol. **PAMI-6**, May 1984, No 3, pp. 340-350.
7. Hebert, M., J. Ponce. A New Method for Segmenting 3-D Scenes into Primitives. – In: Proc. of 6th Int. Conf. Pattern Recognition, Munich, West Germany, 19-22 October 1982, pp. 836-838.
8. Panagiotakis, C., G. Tziritas. Signal Segmentation and Modelling Based on Equipartition Principle. – In: Proc. of 16th International Conference on Digital Signal Processing. Print ISBN: 978-1-4244-3297-4, CD-ROM ISBN: 978-1-4244-3298-1, DOI: 10.1109/ICDSP.2009.5201105.
9. Smith, S. W. The Scientist and Engineer's Guide to Digital Signal Processing. Second Edition. Copyright © 1997-1999 by California Technical Publishing. ISBN 0-9660176-6-8.
10. Kennedy, H. L. A New Statistical Measure of Signal Similarity. – In: Proc. of Conference: Information, Decision and Control, 2007, IDC'07. DOI: 10.1109/IDC.2007.374535.

11. Han, T. S. Nonnegative Entropy Measures of Multivariate Symmetric Correlations. – Information and Control, Vol. **36**, 1978, pp. 133-156.
12. Olbrich, E., N. Bertschinger, N. Ay, J. Jost. How Should Complexity Scale with System Size? – The European Physical Journal B – Condensed Matter and Complex Systems, 2008, DOI: 10.1140/epjb/e2008-00134-9.
13. Abdallah, S. A., M. D. Plumbley. A Measure of Statistical Complexity Based on Predictive Information. ArXiv e-prints. arXiv:1012.1890v1, 2010.
14. Low, R. K. Y., J. Alcock, R. Faff, T. Brailsford. Canonical Vine Copulas in the Context of Modern Portfolio Management: Are They Worth It? – Journal of Banking & Finance, Vol. **37**, 2013, No 8. DOI:10.1016/j.jbankfin.2013.02.036.
15. Kreyszig, E. Introductory Functional Analysis with Applications. John Wiley and Sons, 1978. ISBN: 0-471-50731-8.
16. Chew, L. P., K. Kedem. Getting Around a Lower Bound for the Minimum Hausdorff Distance. Computational Geometry, Theory and Applications, Vol. **10**, 1988, No 3, pp. 197-202.
17. Chew, L. P., K. Kedem, S. Schirra. On Characteristic Points and Approximate Decision Algorithms for the Minimum Hausdorff Distance. Technical Report MPI-I-94-150, Max-Planck-InstitutfürInformatik, Saarbrücken, 1994, DE. 10 p.
18. Dubuisson, M. P., A. K. Jain. Modified Hausdorff distance for object matching. – In: Proc. of IAPR Int. Conf. on Pattern Recognition (ICPR'94, Jerusalem, IS), Vol. **A**, pp. 566-568.
19. Rabiner, L., R. Schafer. Theory and Applications of Digital Speech Processing. 1st Edition. ISBN-13: 978-0136034285, ISBN-10: 0136034284.
20. Pan, V. Y. The Trade-off between the Additive Complexity and the Asynchronicity of Linear and Bilinear Algorithms. – Information Processing Letters, Vol. **22**, 1986, No 1, pp. 11-14. DOI:10.1016/0020-0190(86)90035-9.

Received: 31.01.2020; Second Version: 18.05.2020; Accepted: 20.05.2020