

# Fuzzy Extreme Analysis in Time- and Frequency Domains

Włodzimierz Pogribny, Marcin Drzycimski, and Zdzisław Drzycimski

**Abstract**—Very often, we are interested in the shape of a signal or an envelope of the signal or its spectrum. Classical extreme analysis (CEA) produces too many minor details of a signal shape. This fact, in the context of telecommunications, does not allow us to ensure a significant coefficient of a signal compression. Another option for signal analysis, based on the Delta Modulation (DM), lacks sufficient dynamic range at a relatively low sampling rate. One more “true envelope” manner is based on the cepstral analysis, and requires too many operations.

Other methods are used for the definition of a signal envelope. The discrete Hilbert transform (DHT) is only expedient to capture an envelope of a narrowband signal and, moreover, requires too many mathematical operations. Other manners based on decimation, as well as the use of a signal rectifying followed by low-pass filtering do not always ensure sufficient accuracy and signal compression coefficient.

Fuzzy EA (FEA) is free from similar drawbacks. Its first and second differences are compared with no zero limits, and that allows us to take into consideration only major details of the signal or spectrum shape. Consequently we obtain both an envelope of wideband signal, and a signal significant compression in real time. This article focuses on FEA features connected with the aforementioned tasks. Apart from the FEA algorithm, the article outlines some methods of signal reconstruction after FEA in both domains, and the structure of the FEA specialized processor. FEA application in both domains is demonstrated through examples.

**Keywords**—Extreme analysis, envelope extraction, signal compression.

## I. INTRODUCTION

**I**N order to conduct a signal shape analysis, both CEA [1]–[4] and DM [5] are frequently used. Of these, the first and second methods are connected with the signal shape analysis.

The first method, which is based on the CEA, operates by comparing differences between signal adjacent samples to zero. That leads us to obtain too many minor details which do not allow us to identify the main features of the signal shape. This limits the speed of signal processing as well as its compression and transmission.

The second method uses different kinds of low-bit DM, especially the Adaptive Differential Pulse Code Modulation (ADPCM). This allows us to accurately pick out the signal extremes, but only by using a sampling frequency which exceeds Nyquist’s. Otherwise the procedure does not ensure a sufficient dynamic range [5].

On the other hand, for the definition of a signal envelope, the following 4 methods are often used: DHT [6], [7], the

decimation [8], the low-pass filtering after signal rectifying [9] and “True-Envelope” estimator [10].

The classical operation is based on analog HT algorithm [6]:

$$\tilde{x}(t) = \frac{1}{\pi} p \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau \quad (1)$$

Hence direct DHT algorithm is obtained in the form [6], [7]:

$$\tilde{x}_i = \sum_{m=-N}^N x_m h_{i-m} \quad (2)$$

where  $x(t)$  and  $\{x_m\}_{m=-N}^N$  – input signal in analog and discrete forms,  $\tilde{x}(t)$  and  $\{\tilde{x}_m\}_{m=-N}^N$  – results of analog HT and DHT,  $p$ –principal value of the analog convolution,  $\{h_m\}$  – impuls response which is connected with one of the analog HT convolution [7]. Here  $x_m = x(mT)$  and  $T$  – a sampling period. On the basis of this algorithm, an analytical signal is obtained:  $z_i = x_i + j\tilde{x}_i$ . Hence its instantaneous amplitude in discrete and analog forms is:

$$\{|z_i|\}_{i=-N}^N = \left\{ \sqrt{x_i^2 + \tilde{x}_i^2} \right\}_{i=-N}^N, \quad |z(t)| = \sqrt{x^2(t) + \tilde{x}^2(t)} \quad (3)$$

If the input signal is narrowband, for example  $x(t) = A(t) \sin \omega t$ , and its envelope  $A(t)$  is changing more slowly than the signal itself, then, taking into consideration its HT result  $\tilde{x}(t) = A(t) \cos \omega t$  and using (3), we immediately obtain the envelope:  $|z(t)| = A(t)$ . This also means that it is extremely problematic to achieve a calculation of reasonable accuracy for a broadband signal envelope. Apart from this, DHT requires an inconveniently large number of mathematical operations: about  $(2N + 1)^2$  in the direct case (2) and about  $(2N + 1) \log_2(2N + 1)$  when FFT is used instead of the direct algorithm [6], [7]. This fact limits the application of DHT in real time systems.

Based on decimation, the next option [8] is very simple but not accurate enough, especially when high a compression factor is needed.

The method which uses low-pass filtering after signal rectifying [9] does not always ensure sufficient accuracy of an envelope, especially when sharp signal changes take place.

The last “True-Envelope” method [10] applies to the frequency domain only. It is based on an iterative cepstral technique, and performs a band-limited interpolation of the prominent spectral peaks. This requires many complicated operations and iterations, again limiting its implementation in fast-acting real-time systems.

An alternative is the Fuzzy EA technique, which does not suffer from any of the previously mentioned drawbacks. It is expedient to apply it to study and to compress different kinds of random signals in both time and frequency domains as well as to define their envelopes. As opposed to CEA, its limits are changeable, unequal to zero, and are established in relation to a signal character. Being assigned separately for the first and second differences, they define the “fuzzy” signal rise, constancy and diminution, that provides the researcher with appropriate maxima and minima. Aforementioned differences and limits allow us to take into consideration only the major details of the signal or spectrum shape. This also enables us to obtain both an envelope (even an envelope of a wideband signal), and significant compression of the signal in real time. In connection with the fact that until now, no direct comparison of FEA effects in both domains has been made, the purpose of this article is to study FEA features connected with the abovementioned tasks. Apart from the FEA algorithm usage, the article outlines some techniques of signal reconstruction after FEA in both domains, and the structure of an FEA specialized processor. FEA application in both domains is demonstrated through examples.

## II. FEA ALGORITHM

Extreme analysis is based on backwards differences  $\nabla x_i$  and  $\nabla x_{i+1}$  [1] among three current adjacent samples of a function  $x(t)$ . This can be depicted for  $x_{i-1}, x_i, x_{i+1}$  as:

$$\nabla x_i = x_i - x_{i-1}, \quad \nabla x_{i+1} = x_{i+1} - x_i \quad (4)$$

These differences are sometimes called “increasing differences”.

Focusing only on the analysis of signs of the successive differences (4) allows us to identify the severe rise, changelessness, and diminution of a signal as well as its rigorous and unrigorous maxima and minima. However, in many cases this procedure is accompanied by the detection of a lot of tiny details which play an insignificant role. They disturb the study of the main fragments of a signal or spectrum shape, and at the same time result in an accumulation of needless data.

Our approach is based on the analysis not only of the signs of the first and second differences, but also their values, comparing those with non zero setting boundaries. Then we can define a signal rise, diminution, and its “fuzzy” changelessness within the boundaries. On the other hand, it allows us to sharpen the recognition of “distinct” and “indistinct” rigorous and unrigorous maxima and minima.

For this, it should be expedient to depict a distinct rise (5), fuzzy changelessness (6) and distinct diminution (7) of the signal as follows:

$$\nabla x_i > \epsilon, \quad \nabla x_{i+1} > \epsilon, \quad (5)$$

$$|\nabla x_i| \leq \epsilon^{(\epsilon)}, \quad |\nabla x_{i+1}| \leq \epsilon^{(\epsilon)}, \quad (6)$$

$$\nabla x_i < -\epsilon, \quad \nabla x_{i+1} < -\epsilon, \quad (7)$$

where  $\epsilon, \epsilon^{(s)}$  – set a priori boundaries of “fuzziness”.

On the basis of those differences we can depict all fuzzy extremes. Then, distinct rigorous maximum (8) and minimum (9) of a signal are represented as follows:

$$\nabla x_i > \epsilon, \quad \nabla x_{i+1} < -\epsilon \quad (8)$$

$$\nabla x_i < -\epsilon, \quad \nabla x_{i+1} > \epsilon \quad (9)$$

In addition to these, we have four fuzzy distinct unrigorous maxima and minima. In the case of the inequalities

$$\nabla x_i > \epsilon, \quad |\nabla x_{i+1}| \leq \epsilon^{(s)} \quad (10)$$

or

$$|\nabla x_{i+1}| \leq \epsilon^{(s)}, \quad \nabla x_{i+1} < -\epsilon \quad (11)$$

or

$$\nabla x_i < -\epsilon, \quad |\nabla x_{i+1}| \leq \epsilon^{(s)} \quad (12)$$

or

$$|\nabla x_i| \leq \epsilon^{(s)}, \quad \nabla x_{i+1} > \epsilon \quad (13)$$

we have: distinct unrigorous maximum following the signal rise (10), distinct unrigorous maximum following the signal changelessness (11), distinct unrigorous minimum following the signal diminution (12), and distinct unrigorous minimum following the signal changelessness (13). Using these inequalities provides us with a general algorithm of the fuzzy extremes finding:

$$\begin{aligned} & \forall \nabla x \exists E(((\nabla x_i > \epsilon) \wedge (\nabla x_{i+1} < -\epsilon)) \\ & \quad \vee ((\nabla x_i < -\epsilon) \wedge (\nabla x_{i+1} > \epsilon)) \\ & \quad \vee ((\nabla x_i > \epsilon) \wedge (|\nabla x_{i+1}| \leq \epsilon^{(s)})) \\ & \quad \vee (|\nabla x_i| \leq \epsilon^{(s)} \wedge (\nabla x_{i+1} < -\epsilon)) \\ & \quad \vee ((\nabla x_i < -\epsilon) \wedge (|\nabla x_{i+1}| \leq \epsilon^{(s)})) \\ & \quad \vee (|\nabla x_i| \leq \epsilon^{(s)} \wedge (\nabla x_{i+1} > \epsilon))) \mapsto E \end{aligned} \quad (14)$$

Here  $E$  means the appearance of the extreme,  $\mapsto$  is a symbol of a sequence. The first conjunction of inequalities in brackets depicts the occurrence of distinct rigorous maximum, the second one means the appearance of distinct rigorous minimum. The other conjunctions describe distinct unrigorous extremes: maximum following the signal rise, maximum after the signal fuzzy changelessness, minimum before the signal fuzzy changelessness and minimum after the signal fuzzy changelessness.

We should note that the algorithm (14) is a universal one, and independent of the method of digital representation of the signal samples. Therefore Delta Modulation steps may be used instead of differences [5].

For a more accurate signal analysis, as well as its reconstruction after FEA, it is expedient to use the additional analysis of the secondary signal backwards differences value:

$$\nabla^2 x_{i+1} = \nabla x_{i+1} - \nabla x_i = x_{i+1} - 2x_i + x_{i-1} \quad (15)$$

which is accompanied by the appearance of “indistinct” unrigorous fuzzy extremes. When the condition

$$|\nabla^2 x_{i+1}| > \delta^{(n)} \quad (16)$$

takes place, where  $\delta^{(n)}$  – sets a priori boundary of “indistinctness” of a signal rise or diminution, then indistinct unrigorous fuzzy maximum or minimum of a signal takes place in the sample  $x_i$ . In fact, indistinct unrigorous fuzzy extremes characterize some side bulges and dents of a signal shape. Obviously, this method, in comparison to the one based only on the first differences (14), leads to an increase in the extremes’ number but at the same time it allows us to obtain a more exact signal analysis, as well as its reconstruction after FEA.

We can see that the CEA strict rules used to describe the character of the signal’s changes (including its extremes [1]–[4]) are derived from the abovementioned inequalities (5)–(14) if both  $\epsilon$  and  $\epsilon^{(s)}$  equal to 0.

Similarly, for frequency domain, a signal sample’s differences  $\{\nabla x_k\}$  in (14) can be replaced with its consecutive spectrum component differences  $\{\nabla X_k = \nabla X(k)\}$ , where  $\nabla X(k) = X(k) - X(k-1)$ , and in effect, the algorithm can be written down as:

$$\begin{aligned} & \forall \nabla X \exists E (((\nabla X_i > \epsilon) \wedge (\nabla X_{i+1} < -\epsilon)) \\ & \quad \vee ((\nabla X_i < -\epsilon) \wedge (\nabla X_{i+1} > \epsilon)) \\ & \quad \vee ((\nabla X_i > \epsilon) \wedge (|\nabla X_{i+1}| \leq \epsilon^{(s)})) \\ & \quad \vee ((|\nabla X_i| \leq \epsilon^{(s)}) \wedge (\nabla X_{i+1} < -\epsilon)) \\ & \quad \vee ((\nabla X_i < -\epsilon) \wedge (|\nabla X_{i+1}| \leq \epsilon^{(s)})) \\ & \quad \vee ((|\nabla X_i| \leq \epsilon^{(s)}) \wedge (\nabla X_{i+1} > \epsilon))) \mapsto E \end{aligned} \quad (17)$$

Secondary spectrum differences have the form:

$$\begin{aligned} \nabla^2 X_i &= \nabla X_i - \nabla X_{i-1} = X_i - 2X_{i-1} + X_{i-2} \\ \nabla^2 X_{i+1} &= \nabla X_{i+1} - \nabla X_i = X_{i+1} - 2X_i + X_{i-1} \end{aligned} \quad (18)$$

Indistinct, unrigorous fuzzy extreme occurs when the following condition is met

$$|\nabla^2 X_{i+1}| > \epsilon^{(\max)} \quad (19)$$

where boundary  $\epsilon^{(\max)}$  corresponds to  $\delta^{(n)}$  in the time domain (16). Besides this, the extreme takes place in the spectrum component  $X_i$ . In that way we can depict “bulges” and “dents” in the spectrum. These “indistinct” extremes improve the quality of the analysis as well as the spectrum reconstruction after FEA.

Obviously, gaining only an insignificant number of the distinct and indistinct extremes and distances between them instead of all samples of a signal realization leads to a significant signal compression. That compression is very important for data transmission and recording tasks. We can depict a compression factor for FEA result in the following form:

$$C = \frac{Nc_s}{nc_s + (n-1)c_i} \quad (20)$$

where  $N$  – general number of input signal samples,  $n$  – total number of the extremes,  $c_s$  – code word length of the values of extremes and  $c_i$  – code word length of the intervals between extremes. The extreme values and intervals allow us to reconstruct a signal accurately by using appropriate techniques.

Now let us consider a particular event, where FEA is used to extract a signal envelope. For that, we propose the use of a twofold EA. Firstly, a modulated narrow-band signal with zero constant component is rectified and, afterwards, its strict maxima are defined on the basis of CEA. The maxima represent the envelope samples. Next, these samples undergo FEA according to (14) and (16) so as to obtain “distinct” and “indistinct” extremes as main details of an envelope. These extremes are the basic samples of the envelope, and, after corresponding interpolating, enable us to reconstruct it. Additionally, this leads to an envelope significant compression with minimal loss of accuracy.

For the frequency domain, the first step is connected with both FFT algorithm to define amplitude, and phase spectra of a signal and CEA to extract envelopes of those spectra. The second one is based on FEA that is carried out on both envelopes according to (17) and (19).

### III. SIGNAL AND ENVELOPE RECONSTRUCTION AFTER FEA

Here we consider two methods of signal and envelope reconstruction after FEA. The first approach is based on a linear interpolation of the extreme samples combined with low pass filtering of the interpolation result afterwards. The second is connected with a non-linear approximation of a signal on the basis of these samples.

A linear interpolator leads to rough reconstruction of a signal after FEA according to the algorithm [5]:

$$\hat{x}_k = x_{i-1}^{(e)} + \frac{x_i^{(e)} - x_{i-1}^{(e)}}{r_i} k \quad (21)$$

where:  $\hat{x}_k = \hat{x}(kT)$  –  $k$ -th sample of a signal in  $i$ -th segment of the interpolation; factor  $r_i$  is a difference between numbers of the both border samples which correspond to the  $i$ -th and  $i-1$ -th extremes. Besides this,  $r_i + 1$  is a total number of samples in this segment whilst  $k = \overline{0, r_i}$  – a current sample number on the segment;  $x_{i-1}^{(e)}$  and  $x_i^{(e)}$  – the value of next extremes after FEA on the borders of  $i$ -th segment of the interpolation. Let us note that the first segment begins with an initial sample  $x_0$  of a signal, that is, the first extreme is  $x_0^{(e)} = x_0$ . The linear interpolator generates samples and fills an interval between two sequential extremes with the samples. Therefore, that rough reconstructed signal is connected with all the extremes and the intermediate samples as well as with the number  $\{r_i\}$  of periods  $T$  between the extremes.

Afterwards, that signal is entered to a smoothing digital filter with either a constant or adaptive impulse response (IR). The first case is simpler and is based on a convolution operation of a signal with the inalterable IR:

$$y_n = \sum_{m=0}^{M-1} \hat{x}_{n-m} h_m \quad (22)$$

where  $\{y_n\}$  – the output of filtering;  $\{\hat{x}_m\}$  – a signal after linear interpolation;  $\{h_m\}_{m=0}^{M-1}$  – weight factors of IR.

On the other hand, a general adaptive filtering algorithm on the basis of an error analysis in the time domain can be

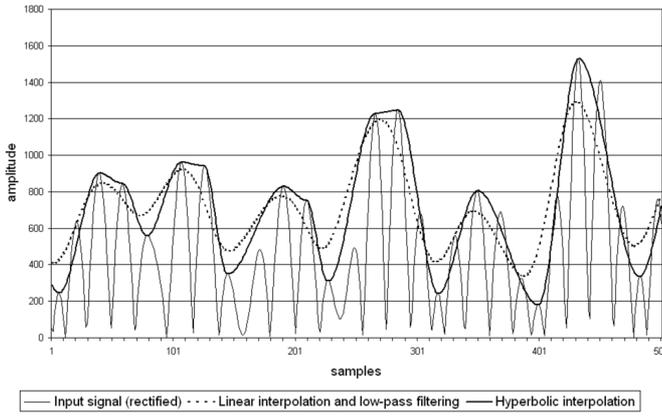


Fig. 1. Comparison of the use of linear and hyperbolic interpolations to an envelope reconstruction. Band-pass signal is in the range 300-600Hz, envelope extremes were obtained using the FEA method. In both cases the compression factor is 119,97.

represented as follows [11]:

$$y_n = \sum_{m=0}^{M-1} \hat{x}_{n-m} p(m, \sigma_y) h_m \quad (23)$$

where  $\{p(m, \sigma_y)\}$  – individual adaptation coefficients for the corresponding weight factors, and  $\sigma_y$  – mean square error (MSE) on a set of the  $n$  extremes:

$$\sigma_y = \left( \frac{1}{n-1} \sum_{r=1}^n (x_r^{(e)} - y_r)^2 \right)^{1/2} \quad (24)$$

The adaptation ability of the filtering consists of corresponding changes of an IR's weight factors under the influence of MSE. In the simplest case, the adaptation coefficient can be common for all weight factors and has the form  $p(\sigma_y)$ . Then formula (22) is transformed into the following:

$$y_n = p(\sigma_y) \sum_{m=0}^{M-1} \hat{x}_{n-m} h_m \quad (25)$$

In this case, the adaptation process consists in choosing the whole corresponding IR.

A decision about changing the weight factors is made on the basis of a criterion  $\sigma_y > / < \sigma_{ass}$  where  $\sigma_{ass}$  is the assigned MSE of a signal reconstruction.

For example, if  $\sigma_y \leq \sigma_{ass}$  then  $p(\cdot) = 1$  and values of  $\{h_m\}_{m=0}^{M-1}$  will not be changed.

When  $\sigma_y > \sigma_{ass}$  two events can take place. Firstly, if  $\sum_{r=1}^n \text{sgn}(x_r^{(e)} - y_r) > 0$  then the weight factor values should be increased, e.g. by the choice  $p(\cdot) = 1, 5$ . Secondly, in the opposite case, when  $\sum_{r=1}^n \text{sgn}(x_r^{(e)} - y_r) < 0$ , the values should be decreased, e.g. by setting  $p(\cdot) = 0, 7$ .

In some cases it is expedient to base the filtering adaptation on an error analysis in the frequency domain [12]. For that, we should operate on a power spectrum error of the reconstructed signal that allows us, analogously to the algorithms (24) and (25), to change the IR in an appropriate way. The filtering

TABLE I  
ENVELOPE RECONSTRUCTION AFTER FEA IN THE TIME DOMAIN (1ST TEST SIGNAL)

$\epsilon, \epsilon^{(s)}$	$\delta^{(n)}$	Compression	
		factor $C$	SNR [dB]
0	-	14,87	14,74
0,0002	0,001	6,50	41,27
0,0005	0,010	11,59	20,18
0,0007	0,002	8,11	32,25
0,0010	0,002	8,79	30,59
0,0020	0,004	11,27	24,58
0,0020	0,008	13,20	21,04
0,0020	0,040	16,46	18,66

effectiveness, in both cases, depends mainly on the character of the signal and its spectrum. Some drawbacks of the depicted algorithms are the complication of the necessary hardware realization, as well as the occurrence of a delay connected with the filtering process duration.

Better approximation results in comparison to the algorithm depicted by (21) and (22) can be obtained using a non-linear interpolation [13], [14]. In the work [13] the hyperbolic interpolation for the non-linear approximation has been proposed as optimal. For that interpolation we assigned the next initial conditions. Values of a function  $\hat{x}_k = \hat{x}(kT)$  in the points of neighbouring  $(i-1)$ -th and  $i$ -th extremes are  $x_{i-1}^{(e)}$  and  $x_i^{(e)}$ . The signal  $\hat{x}(t)$  derivatives in these points equal 0. This allows us to obtain a system of four equations, hence we can write down the expression to the hyperbolic interpolation:

$$\hat{x}_k = -2(x_i^{(e)} - x_{i-1}^{(e)}) \left( \frac{k}{r_i} \right)^3 + 3(x_i^{(e)} - x_{i-1}^{(e)}) \left( \frac{k}{r_i} \right)^2 + x_{i-1}^{(e)} \quad (26)$$

All denotations in the formula (26) are the same as in (21).

We can demonstrate the advantage of this technique via the example of the reconstruction of an envelope of rectified modulated sine signal, Fig. 1. after FEA, the linear interpolation together with low-pass filtering was used, as well as hyperbolic one.

We can see that the hyperbolic interpolation leads to more accurate signal reconstruction than the linear one together with low-pass filtering (21) at the same compression factors. Additionally, it is both simpler and faster because of its foundation on simple mathematical operations only.

#### IV. SIMULATION METHOD AND RESULTS

The effectiveness of the FEA method was examined in the context of its accuracy and compression factor (20), in depending on the boundaries of "fuzziness"  $\epsilon, \epsilon^{(s)}$  and  $\delta^{(n)}$ . As an example, below are listed some of the results of studying the extraction of two test signal envelopes, and reconstruction on the basis of the abovementioned algorithms in comparison with the well-known decimation method [8] by using different decimation factors. Additionally, a signal-noise ratio (SNR) was used for the evaluation of the accuracy of an envelope reconstruction as follows:

$$\text{SNR} = 10 \lg \frac{\Psi_x^2}{\sigma_x^2} \quad (27)$$

TABLE II  
AMPLITUDE SPECTRUM ENVELOPE RECONSTRUCTION AFTER FEA IN  
FREQUENCY DOMAIN (1ST TEST SIGNAL)

$\epsilon, \epsilon^{(s)}$	$\delta^{(n)}$	Compression factor $C$	SNR [dB]
0	-	1,41	19,74
0,02	2	4,68	30,47
0,05	5	5,56	25,18
0,10	5	7,79	24,27
0,15	0,3	10,51	23,14
0,20	0,4	14,63	20,73
0,30	0,6	21,33	18,14
0,50	2,5	29,52	17,25

where mean power of an envelope is:

$$\Psi_x^2 = \frac{1}{n} \sum_{r=0}^{n-1} (x_r^{(e)})^2 \quad (28)$$

and an approximation MSE is:

$$\sigma_x = \sqrt{\frac{1}{n} \sum_{r=0}^{n-1} (x_r^{(e)} - \hat{x}_r)^2} \quad (29)$$

Here  $n$  is a quantity of an envelope maxima  $\{x_i^{(e)}\}_{i=0}^n$  determined on the basis of rigorous CEA,  $\{\hat{x}_i\}_{i=0}^n$  - the samples of the envelope reconstructed after FEA at the same points.

The first test signal was English speech which was filtered in a band 300-500 Hz and, afterwards, rectified. Its duration was 1 second and it consisted of 8000 samples.

Table I presents SNR values for its envelope which was extracted and processed using FEA and hyperbolic interpolation in the time domain. The simulation was carried out using different  $\epsilon, \epsilon^{(s)}$  and  $\delta^{(n)}$  values to study their impact on the compression factor  $C$  and SNR. As Table I shows, the SNR values in the range of 30-32 dB can be acquired at good compression factors between 7 and 9. As was to be expected, further widening of the "fuzziness" boundaries increases the compression factor by worsening the SNR.

Tables II and III present the effects of the use of FEA in the frequency domain for the processing of amplitude and phase spectra respectively. Similarly to the previous example, SNR values for envelope extraction and reconstruction after FEA are presented for different  $\epsilon, \epsilon^{(s)}$  and  $\delta^{(n)}$  values to show the their impact on the compression factor and SNR. In this case,

TABLE III  
PHASE SPECTRUM ENVELOPE RECONSTRUCTION AFTER FEA IN  
FREQUENCY DOMAIN (1ST TEST SIGNAL)

$\epsilon, \epsilon^{(s)}$	$\delta^{(n)}$	Compression factor $C$	SNR [dB]
0	-	1,43	12,61
0,1	0,2	1,06	50,68
0,7	2	1,35	22,90
1,0	2	1,52	20,82
1,5	3	1,88	14,75
2,0	4	2,21	13,24

TABLE IV  
ENVELOPE EXTRACTION USING DECIMATION (1ST TEST SIGNAL)

Decimation factor	SNR [dB]
2	28,99
5	14,27
10	7,01
20	6,72
30	6,17
40	5,72
50	4,51

FEA confirms its effectiveness for amplitude spectrum when the compression factor of an amplitude spectrum is around 5, while SNR is around 31dB and 10 when SNR is around 23 dB. Better compression factor (around 11 or better) can be achieved, but this causes the SNR drop.

As the simulation results show, in most cases the use of FEA for processing a phase spectrum gives a low compression and SNR. Therefore FEA is inexpedient for processing a phase spectrum with a large number of extremes.

Table IV presents SNR values for a well-known method of envelope extraction based on a decimation algorithm in the time domain [8]. The SNR values are given for different decimation factors.

It is noticeable that generally, in this case, worse SNR takes place than when FEA was used. It should be noted that even though Nyquist's criterion was met for decimation factors 2 and 5, the SNR corresponding values were still low. Obviously, low-pass pre-filtering would slightly improve SNR for higher decimation factors (10 and more) as the aliasing effect would be eliminated.

Fig.2 shows an example of an envelope extracting of the first test signal with the use of the CEA in the time domain.

Fig.3 shows the envelope reconstruction after both FEA and decimation in the time domain to compare them.

Fig. 4 illustrates an envelope restoration of the first test signal after FEA in the frequency domain. In this case, the

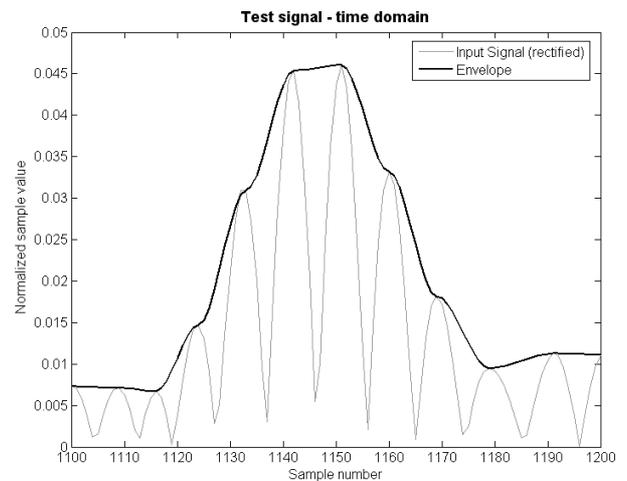


Fig. 2. First test signal envelope extracting with using rigorous EA (CEA) in the time domain.

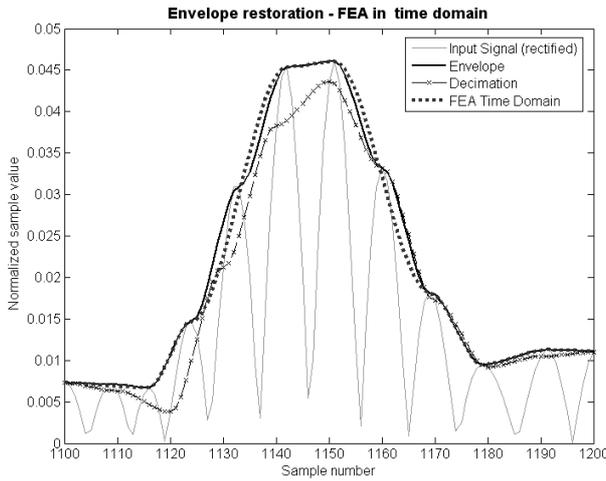


Fig. 3. Envelope restoration of the first test signal after FEA and after decimation in the time domain.

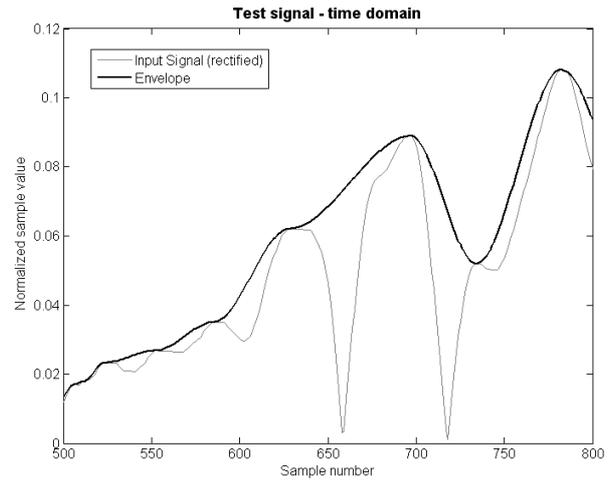


Fig. 5. Second test signal envelope extraction with the use of rigorous EA (CEA) in the time domain.

signal envelope was reconstructed after IFFT of its amplitude and phase spectra, which had before undergone the FEA in this domain. In all cases a hyperbolic approximation for an envelope reconstruction was used. The reconstruction, after FEA, of an amplitude spectrum (FEA FQ Domain (amplitude)) is accompanied by a compression factor  $C = 10,54$  when  $SNR = 23,14$  dB and after FEA of a phase spectrum (FEA FQ Domain (phase)) correspondingly by  $C = 1,88$  if  $SNR$  is  $14,75$  dB. On the other hand, in the time domain, the envelope reconstruction after FEA is connected with  $C = 10$ , when  $SNR$  is  $25,94$  dB, whilst after decimation the values  $C = 10$  and  $SNR = 7,01$  dB take place.

The second test signal had a heartbeat character. Its form in the time domain after rectifying, as well as its extracted envelope after CEA and a hyperbolic approximation, are shown in Fig.5.

Table V shows the simulation output of the second test sig-

nal processing in the time domain. The very good compression factor  $C$  (18 and more) with high  $SNR$  (50 dB and more) should be noted here. It is the effect of more “stable” signal envelope and the choice of suitable coefficients of fuzziness. The difference between classical EA (when  $\epsilon$  and  $\epsilon^{(s)}$  are equal to 0 and without  $\delta^{(n)}$ ) and FEA is noticeable: FEA provides better accuracy because of the use of unrigorous extremes.

Tables VI and VII show simulation outputs of this signal processing in the frequency domain. For amplitude spectrum, compression factor  $C$  as well as  $SNR$  are worse than for the first test signal in the same domain. This is contrary to the better result of the second test signal FEA processing in the time domain. As it was expected, there is a significant improvement when FEA is used for processing a phase spectrum, due to the calmer shape character of that spectrum.

In comparison to the result of the previous test signal processing (shown in Table IV), the decimation algorithm gave

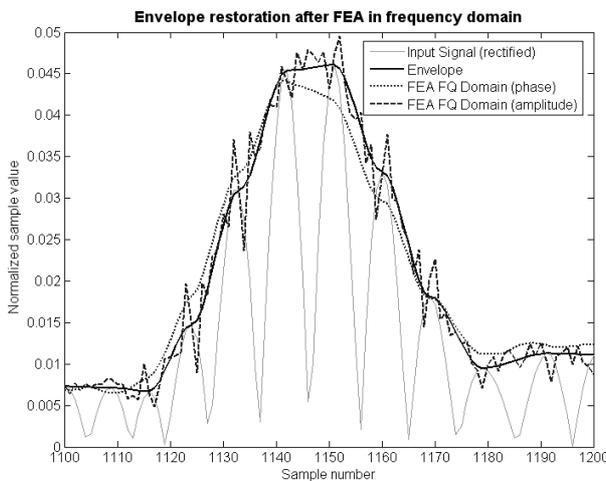


Fig. 4. Envelope restoration of the first test signal after FEA in the frequency domain.

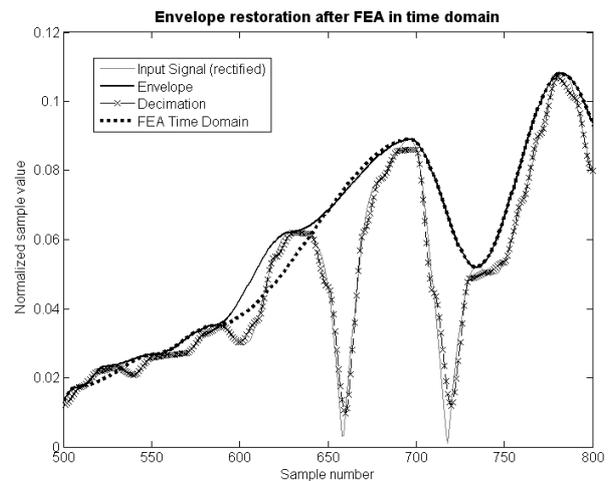


Fig. 6. Envelope restoration of the second test signal: after FEA in the time domain and decimation.

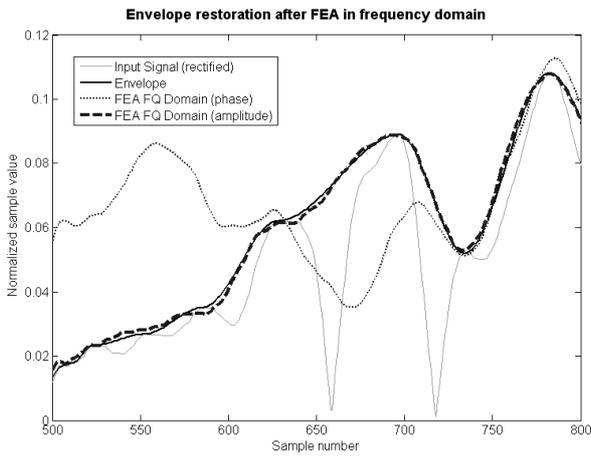


Fig. 7. Envelope restoration of the second test signal after FEA in the frequency domain.

much better results for the second test signal. As table VIII shows, it allows us to obtain SNR= 53 dB for the compression factor  $C = 2$  and 28.5 dB for  $C = 10$ . Similar to the previous case, SNR drops at higher decimation factors.

Fig.6 and Fig.7 illustrate an effect of the second test signal envelope reconstruction after FEA in both domains, as well as after the decimation. In both cases, hyperbolic approximation was used to restore an envelope. In these examples, the amplitude spectrum envelope reconstruction after FEA (FEA FQ Domain (amplitude)) is accompanied by compression factor  $C = 9,36$  when SNR is 25,82 dB and after FEA of a phase spectrum (FEA FQ Domain (phase)) correspondingly 8,92 and 15,43 dB. On the contrary, the envelope reconstruction after FEA in the time domain is characterized by  $C = 19,04$  and SNR= 50,85 dB whilst after decimation there are  $C = 10$  and SNR= 28,49 dB.

## V. FEA ANALYZER

The structure of a specialized processor realizing FEA algorithm is shown in Fig.8. It consists of: analog to digital converter A/D, first and second delay elements on one sampling period delay, first subtractor SUB for obtaining first differences, second subtractor for the calculation of second differences, two comparators responsible for the detection of extremes, an OR gate, a switch SW for transmitting all extreme

TABLE V  
ENVELOPE RECONSTRUCTION AFTER FEA IN THE TIME DOMAIN (2ND TEST SIGNAL)

$\epsilon, \epsilon^{(s)}$	$\delta^{(n)}$	Compression factor $C$	SNR [dB]
0	-	26,32	12,76
0,0002	0,001	18,10	51,55
0,0002	0,002	18,18	50,93
0,0005	0,002	18,52	50,93
0,0007	0,002	18,78	50,91
0,0010	0,002	19,05	50,85
0,0020	0,010	19,51	40,90
0,0300	0,010	21,51	38,68

TABLE VI  
AMPLITUDE SPECTRUM ENVELOPE RECONSTRUCTION AFTER FEA IN FREQUENCY DOMAIN (2ND TEST SIGNAL)

$\epsilon, \epsilon^{(s)}$	$\delta^{(n)}$	Compression factor $C$	SNR [dB]
0	-	1,34	17,97
0,02	2	4,16	42,98
0,05	2	5,56	46,79
0,15	0,33	7,82	42,13
0,2	0,4	8,60	25,85
0,2	5	9,58	25,78
0,2	3	9,36	25,82
0,3	3	10,71	19,74
0,5	5	13,27	17,79
1	5	15,36	11,09

TABLE VII  
PHASE SPECTRUM ENVELOPE RECONSTRUCTION AFTER FEA IN FREQUENCY DOMAIN (2ND TEST SIGNAL)

$\epsilon, \epsilon^{(s)}$	$\delta^{(n)}$	Compression factor $C$	SNR [dB]
0	-	1,33	5,73
0,1	0,2	2,41	55,44
0,5	2	4,83	18,58
0,7	2	5,73	17,57
1	2	6,91	20,50
1,5	3	8,92	15,43
2	4	10,06	13,13

samples  $\{x_{extr}\}$ , a counter CT for calculating the distance between extremes, and a clock CLK for controlling the whole structure.

After PCM procedure, input signal samples  $\{x_i\}$  are forwarded to the first subtractor unit. The subtractor calculates the first difference value  $\nabla x_i = x_i - x_{i-1}$  using current  $x_i$  and delayed  $x_{i-1}$  samples. Next, this result is forwarded to the second subtractor to obtain the second difference value analogously. The first and the second differences are then compared with  $\epsilon, \epsilon^{(s)}$  and  $\delta^{(n)}$  – set a priori boundaries of the “fuzziness” and “indistinctness” of a signal change to detect distinct and indistinct fuzzy extrema. When either extreme is detected, an appropriate control signal is generated by the corresponding comparator. Then, via the OR gate, this signal turns on the switch which passes an extreme sample  $x_{extr}$  to the first output of the processor. Apart from this, that signal influences the counter action that causes the passing of extreme distances to the second output of the processor. After that counter is reset.

## VI. SUMMARY

As the abovementioned experiments show, FEA is an effective means of signal analysis and compression, as well as envelope extraction. It is expedient for use in processing both narrowband and wideband signals in the time and frequency domains. Its peculiarities follow from its operating not only on signs, but also on values of the first and second differences by using different boundaries for comparison. Thanks to the fact that it picks up only major details of the signal or spectrum shape, FEA allows us to obtain a significant compression of a signal at high accuracy of its reconstruction afterwards.

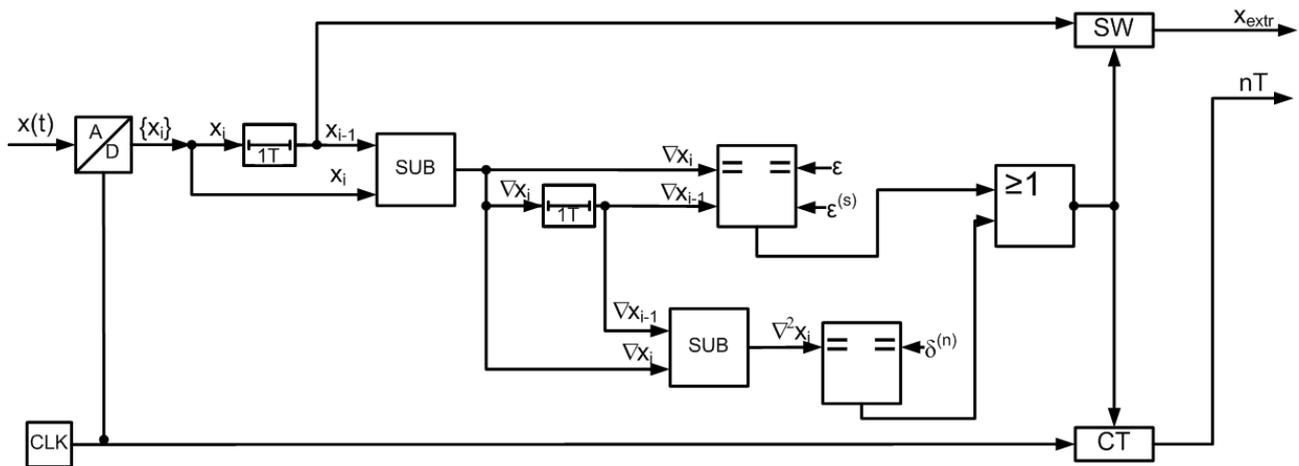


Fig. 8. Block scheme of FEA specialized processor working in the time domain.

In particular, FEA leads to the much better capture of an envelope than the classic method based on direct decimation. For example, for the same envelope FEA in the time domain allows us to obtain  $\text{SNR} \approx 41$  dB and compression factor 6.5, as opposed to the direct decimation with decimation factor 5 when SNR is about 14dB. FEA demonstrates its advantages in the frequency domain as well – for the processing of an amplitude and phase spectra.

It should be noted that the compression coefficient of FEA sufficiently depends on a signal character. For longer and more “steady” parts of the input signal (e.g. voice signal with a lot of “silent” fragments) compression coefficient can be reached even to around 120 for the time domain, still assuring better SNR than the classic envelope extraction method. Further optimisation of the compression ratio is possible by the reduction of the number of bits used to transmit the extreme distances.

It has been demonstrated that in most cases, hyperbolic approximation should be recommended for signal reconstruction after FEA. Its accuracy is very close to the one of signal reconstruction on the basis of linear interpolation together with an adaptive low-pass filtering at the same compression factors. However, the hyperbolic approximation is simpler and faster because it is based solely on simple mathematical operations.

The structure of a specialized processor for FEA is simple, economical, and fast. That makes it attractive to use in different real-time systems. Its universality promotes its use

in a wide range of applications including telecommunications, physics, and medicine.

## REFERENCES

- [1] G. Korn and T. Korn, *Mathematical Handbook*. New York: McGraw Hill, 1968.
- [2] J. Cacko, M. Bily, and J. Bukovec, *Random Processes: Measurement, Analysis and Simulation*. New York: Elsevier Science Publishing Company, 1988.
- [3] A. Rosenfeld and E. Johnston, “Angel Detection on Digital Curves,” *IEEE Transactions on Computers*, vol. C-22, pp. 875–878, 1973.
- [4] C. H. Teh and R. T. Chin, “On the Detection of Dominant Points on Digital Curves,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, pp. 859–872, 1989.
- [5] V. O. Pohribnyi (W. Pogribny), Z. Drzycimski, and I. Zielinski, “Compression of Images with Use of Differential Methods and Extreme Analysis,” in *Mathematics of Data/Image Coding, Compression, and Encryption*, M. S. Schmalz, Ed., 1998, pp. 176–184, Proceedings of SPIE, vol. 3456.
- [6] A. D. Poularikas, *The Transforms and Applications Handbook*, 2nd ed. Boca Raton FL: CRC Press LLC, 2000.
- [7] A. V. Oppenheim and R. Schaffer, *Discrete-time Signal Processing*, 2nd ed. Upper Saddle River NJ: Prentice-Hall, Inc., 1998.
- [8] R. G. Lyons, *Understanding Digital Signal Processing*. New York: Addison Wesley Longman, 1997.
- [9] B. Boulet and L. Chartrand, *Fundamentals of Signals and Systems*. Boston: Da Vinci Engineering Press., 2005.
- [10] A. Robel, F. Villavincenzo, and X. Rodet, “On Cepstral and All-pole Based Spectral Envelop Modeling With Unknown Model Order,” *Pattern Recognition Letters*, vol. 28, no. 11, pp. 1343–1350, 2007.
- [11] W. Pogribny, Z. Drzycimski, and J. Zalewski, “Adaptacyjny sposób odtwarzania sygnałów,” 2003, patent UP RP No. 186190, IntCl H03M 7/00.
- [12] —, “Adaptacyjny sposób odtwarzania sygnałów,” 2003, patent UP RP No. 186192, IntCl H03M 7/00.
- [13] W. Pogribny, Z. Drzycimski, and M. Drzycimski, “Fuzzy Extreme Analysis of Random Signals,” in *Proceedings of the 6th International Conference QRM*, Oxford, 2007, pp. 151–156.
- [14] W. Pogribny and M. Drzycimski, “Fuzzy Extreme Analysis for Signal Compression,” in *ICSES 2008 Conference Proceedings*, Kraków, 2008, pp. 31–34.

TABLE VIII  
ENVELOPE EXTRACTION WITH THE USE OF DECIMATION (2ND TEST SIGNAL)

Decimation factor	SNR [dB]
2	53,51
5	38,33
10	28,49
20	14,51
30	10,83
40	9,68
50	7,41