Oleh TSURKAN[1,2], Ihor KUPCHUK[1], Yurii POLIEVODA[1], Oleksandr WOZNIAK[1], Yaroslav HONTARUK[1], Yrii PRYSIAZHNIUK[1]

Vinnytsia National Agrarian University (1), Separated structural unit «Ladyzhyn Professional College of Vinnytsia National Agrarian University» (2)
ORCID. 1. 0000-0002-7218-0026, 2. 0000-0002-2973-6914, 3. 0000-0002-2485-0611, 4. 0000-0002-0986-6869, 5. 0000-0002-7616-9422, 6. 0000-0002-3351-3676

doi:10.15199/48.2022.11.08

Digital processing of one-dimensional signals based on the median filtering algorithm

***Abstract**. The article is devoted to solving the problem of effective digital signal processing of the information system for monitoring the indicators of chemical processes of a biogas plant. Considers the principles of digital signal processing, general provisions of digital filtering, existing methods of noise filtering in electrical signals, the median filters of one-dimensional signals are studied in details. In order to increase the speed of information processing, a median filtering algorithm based on difference matrices using the threshold saturation function was developed.*

Streszczenie. *Artykuł poświęcony jest rozwiązaniu problemu efektywnego cyfrowego przetwarzania sygnałów systemu informatycznego monitorowania wskaźników procesów chemicznych biogazowni. Rozważa zasady cyfrowego przetwarzania sygnałów, ogólne przepisy filtrowania cyfrowego, istniejące metody filtrowania szumów w sygnałach elektrycznych, szczegółowo badane są filtry medianowe sygnałów jednowymiarowych. W celu zwiększenia szybkości przetwarzania informacji opracowano algorytm filtrowania medianowego oparty na macierzach różnic z wykorzystaniem funkcji progowej nasycenia.* (***Cyfrowe przetwarzanie sygnałów jednowymiarowych w oparciu o algorytm filtrowania medianowego***)

Keywords: digital signal processing, digital filtering, median filters, filters of one-dimensional signals, threshold saturation function.
Słowa kluczowe: mechanizm rozdrabniania paszy z łodyg, napęd hydrauliczny, model matematyczny, stany nieustalone, rozdzielacz

Introduction

Refusal of agricultural enterprises of Ukraine from the use of traditional fuel [1] and provision of energy needs through biogas [2, 3] is a priority task for the energy security of Ukraine [4, 5]. It is known that the productivity of methane generation significantly depends on the creation of favorable conditions for the occurrence of chemical reactions in the bioreactor [2, 6], therefore, monitoring of indicators of chemical processes and timely adjustment of the parameters [7, 8, 9] of a biogas plant is an important task. For this, a whole complex of measuring equipment, sensors and signal transmitters is used [7, 10].

The key task of processing physical signals is to extract interference and obtain reliable information [11, 12]. This information is usually present in the amplitude of the signal (absolute or relative) [13], in frequency or spectral composition [14], in phase, or in the relative time dependences of several signals [15]. The key problem of processing physical signals is the problem of obtaining useful information. Filtering is used to remove interference. The concept of filtering always implies some notion of a “perfectly accurate” signal. Such a signal is the purpose of filtering.

The aim of the work is to study the process of processing of electrical signals by using filtering and to develop an algorithm for fast median filtration. Research objectives defined by the purpose require: analysis of digital noise filtering in electrical signals; development of software for fast processing of signal values with a median filter; conduct a detailed study of the median filter, experimentally test the software-implemented median filter with different apertures at different levels of fluctuation interference.

Analysing the ways of problem solution

In many cases, it is more efficient to use a median filter compared to linear filters, because the linear processing procedures are optimal for steady or Gaussian distribution of interference, which is impossible in real signals [13, 16]. The median of the sequence $x_1, x_2, ..., x_n$ at odd *n* is the average value of the member of the series, which is obtained by ordering this sequence in ascending (or descending) order [17]. For even *n*, the median is defined as the arithmetic mean of two averages of an ordered sequence.

The median filter is a window filter that sequentially slides along the signal array, and returns at each step one of the elements that fall into the window (aperture) of the filter. The output signal y_k of the sliding median filter, for the current count k is formed from the input time series x_{k-1}, x_k, x_{k+1} [17, 18]:

(1) $$y_k = med(x_{k-n}, x_{k-n+1}, x_{k-1}, x_k, x_{k+1}, x_{k+n-1}, x_{k+n}),$$

where $med(x_1, x_m, x_{2n+1}) = x_{n+1}$, x_m – elements of the variation series, which is ranked in ascending order:

$$x_1 = min(x_1, x_2, \dots, x_{2n+1}) \le x_2 \le x_3 \le \dots \le x_{2n+1} = max(x_1, x_2, \dots, x_{2n+1}).$$

Thus, the median filtering replaces the values of the samples in the center of the aperture with the median value of the initial samples inside the aperture of the filter. Due to this feature, the median filters at the optimally selected aperture can maintain sharp distortions of objects without distortion, suppressing weakly correlated interference [15, 19]. The initial and final filtering conditions are the final values of the signals, or the median that fits within the aperture.

The attenuation of statistical noise by median filters due to their nonlinearity is considered only at a qualitative level. It is also impossible to clearly distinguish the effect of median filters on signal and noise [18, 20]. If the values of the elements of the sequence of numbers {*xi*} in the filter aperture are independent and equally distributed (IED) random variables with an average value $mx = m + z$,, then the mathematical expectation is $M\{z\} = 0$ and hence $M\{z\} = 0$.

Let $F(x)$ and $f(x) = F'(x)$ – distribution functions and probability density of quantities x. According to probability theory, the distribution $y = med(x_1, \dots, x_n)$, for large *n* is approximately normal $N(m_t, \sigma_n)$, where m_t – theoretical median, which is determined by the condition $F(m_t)$= 0.5, while the dispersion of the distribution is [17]:

(2) $$\sigma^2 = 1/(n4f^2(m_t)).$$

These results are valid for both one-dimensional and two-dimensional filtering, if *n* is chosen equal to the number of points in the filter aperture. If *f (x)* is symmetric with *m*, then the distribution of the medians will also be symmetric to *m*, thus, the formula:

(3) $M\{med(x_{1,...,}x_n)\} = M\{x_i\} = m.$

If the random variables *x* are IED and are evenly distributed on the interval [0,1], then we can find the exact value of the median variance by the formula [7]:

(4) $\sigma_n^2 = 1/(4(n+2)) = 3_x/(n+2).$

If the random variables *x* are independent, equally distributed with the normal distribution *N (m,), then mt = m.* Modified median variance formula for small odd values of *n*:

(5) $\sigma_n^2 \approx \pi\sigma^2(2n - n + \pi).$

The value of the noise dispersion for random variables in the sliding *n*-window of arithmetic mediation (first-order MNC filter) is σ^2/n. This means that for normal white noise with equal values of *n* windows of the median filter and the sliding mediation filter, the noise variance at the output of the median filter is approximately 57% greater than that of the sliding filter. For the median filter to give the same dispersion as the sliding mediation, its aperture must be 57% larger.

Thus, with exponential (modulo) noise density distribution:

(6) $f(x) = (\sqrt{2}/\sigma\, exp(-\sqrt{2}|x-m|/\sigma)),$

the noise dispersion after the median filter is 50% less than after the moving average filter. The limiting case of such distributions is the pulse noise, which is suppressed by the median filters with the greatest efficiency.

When registering, processing and exchanging data in modern measuring and information systems, signal flows in addition to the useful signal $s(t-\tau_k)$ and fluctuation noise $q(t)$ usually contain pulsed fluxes $g(t) = \sum k\delta\,(t-\tau_k)$ of different intensity with a regular or chaotic structure:

(7) $x(t) = s(t-\tau_0) + g(t) + q(t).$

Suppose the noise $q(t)$ is a statistical process with zero mathematical expectation, a useful signal $s(t-\tau_0)$ has an unknown time position $\tau_0 \in [0,T]$, and the flow of noise pulses $q(t)$ looks like:

(8) $$g(t) = \sum_{k=1}^{K} \varepsilon_k a_k g(t-\tau_k),$$

where a_k – the amplitude of the pulses in the flow; τ_k – unknown time position of pulses $\varepsilon_k = 1$ with probability p_k and $\varepsilon_k = 0$ with probability p_k.

This impulse interference problem corresponds to the Bernoulli flow. When applied to a stream $x(t)$ of sliding median filtering with window *N* deductions (*N* - odd) median filter completely eliminates single pulses spaced at least half of the filter aperture, and suppresses pulse interference, if the number of pulses within the aperture does not exceed $N-1)/2$. In this case, when $p_k = p$ for all interference pulses, the probability of interference suppression is determined by:

(9) $$R(p) \sum_{m=0}^{\frac{N-1}{2}} C_N^m p^m (1-p)^{N-p}.$$

If the probability of error is not very high, then the median filtering, even with a sufficiently small aperture, will significantly reduce the number of errors.

Research results

Calculation of ADC parameters to ensure efficient operation of filters.

Median filters are widely used for image processing and for them adaptive algorithms of order selection and weighting of coefficients are defined [7, 9]. Similar algorithms have been developed for the analysis of wave fields, and for vibroseismic harmonic signals, median filters of pair orders have been developed [17].

When selecting an information signal in a broadband transmission system, the median filter must operate in the mode of recursive aperture formation so that some of the values that fall into the filter aperture are the output signals for the previous nodes. Determination of the values of y_j for this mode is carried out in accordance with the expression:

(10) $$y_j = \sum_{i=0}^{N-1} x_i,$$

where y_j – the value of the output signals of the recursive filter for the previous nodes; $x_i = \begin{cases} y_i, i = 0, N-2 \\ x_i, i = N-1 \end{cases}$, x_i – values of the input file that fall into the aperture.

Since it is advisable to record the values coming from the communication channel using an ADC, it is necessary to determine its basic parameters. The minimum number of calculations that must be recorded to clear the information signal from noise by the median filter is defined as $\frac{N-1}{2} \cdot k_m$, where $k_m \geq 1$ – signal sampling margin factor:

(11) $$T_{ADC} \leq \frac{2 \cdot k_m}{(N-1) \cdot \upsilon \cdot k_\upsilon} - T_{WR} - T_{RD}.$$

where T_{ADC} – the duration of the ADC conversion cycle; T_{WR} – the duration of the program cycle of starting the ADC using the interface circuit until the signal "Start"; T_{RD} – the duration of the program cycle of reading data from the ADC from the moment of determining the signal "End of conversion" to the moment of writing data to memory.

The ADC reference voltage must not be less than the input signal $\hat{x}(t)$ taking into account the quantization error. For evaluation calculations, you can use a simplified formula:

(12) $U_{0ADC} \geq \Delta U_{ADC} + \nu \geq |U_c| + U_\xi;$

(13) $U_{0.ADC} \cdot \left(1 - \frac{1}{N_{ADC}}\right) \geq |U_c| \cdot \left(\nu + \frac{1}{\sqrt{h^2}}\right).$

For a binary analog-to-digital converter, the final formula for determining the number of ADC bits, regardless of the type of signal:

(14) $$n_{ADC} \geq log_2\left(\frac{U_{0.ADC}}{U_{0.ADC} - |U_c| \cdot \left(\nu + \frac{1}{\sqrt{\square^2}}\right)}\right) + 1.$$

Thus, during the time [0, *T*] the input of the median filter receives a set of signals $\hat{x}(t)$. The informative signal *x* (*t* - *τ*) has an unknown time location *τ* ∈ [0, *T*], and this interval has many elements of the section with a delay. The pulse flow *χ* (t) has the form:

(15) $\chi(t) = \sum_{j=0}^{L-1} \kappa_{\chi.j} \cdot U_{\chi.j} \cdot f\left(t - \tau_{\chi.j}\right),$

where $U_{\chi.j}$ – pulse amplitude in the flow *χ* (t); τ_j – its temporary location; $\kappa_{\chi.j}$ – impulse noise factor, which is equal to one with probability p_χ and zero - with probability (1 - p_χ).

This problem of interference corresponds to the Bernoulli flow, for which there are no more than *L* points on the interval [0, *T*]. The statistics of each point is characterized by a partial density:

(16) $s_j(\tau_\chi) = p_{\chi.j} \cdot w_j(\tau_\chi),$

where $p_{\chi.j}$ – the probability of the *j-th* pulse; $w_j(\tau_\chi)$ – distribution of moments of their appearance.

In case that the rationing condition is met $\int_0^T w_j(\tau_\chi) d\tau_\chi = 1$, at p_χ = 1(in the time interval [0, *T*] all *L* pulses are available) and w_j (τ_χ) = *δ* (*τ* - τ_j), the flux *χ*(*t*) is defined as a deterministic pulse noise. If the combined signal $\hat{x}(t)$ is sampled over time with an interval Δ*T* and these samples are subjected to sliding recursive median filtering with an aperture *N*, then taking into account the generating function Θ (*z*) of the Bernoulli flow:

(17) $\Theta(z) = \prod_{i=0}^{N-1}\left(1 + p_{\chi.i} \cdot (z-1)\right) = \sum_{i=0}^{N} p_{\chi.i} \cdot z^i,$

where $p_{\chi.i} = \frac{1}{i!} \cdot \frac{\partial\Theta(z)}{\partial z^i}\Big|_{z=0}$.

The probability of removing the impulse interference:

(18) $$p_{np} = \sum_{i=0}^{\frac{N-1}{2}} p_i,$$

or in the case of equality of all p_j , for the Bernoulli flow:

(19) $$p_{np}^{(B)} = \sum_{i=0}^{\frac{N-1}{2}} C_N^i \cdot p_\chi^i \cdot \left(1-p_\chi\right)^{N-i}.$$

If $p_j \Big/ \sum_{j=0}^{L-1} p_j << 1$, then the flow in its properties is close to the Poisson flow, and:

(20) $$p_{np}^{(B)} = \sum_{i=0}^{\frac{N-1}{2}} C_N^i \cdot p_\chi^i \cdot \left(1-p_\chi\right)^{N-i}.$$

where $\Lambda = \sum_{j=0}^{N-1} p_j \int_0^T w_j(\tau_\chi) d\tau_\chi = \sum_{j=0}^{N-1} p_j = \lambda \cdot N$;

$\lambda = \frac{1}{N}\sum_{j=1}^{N-1} p_j$ – the average intensity of the Poisson flow within the aperture of the sliding recursive median filter.

The value of $N \cdot \Delta T$ characterizes the time interval on which N samples are taken. Then:

(21) $$p_{np}^{(P)} = \sum_{i=0}^{\frac{N-1}{2}} \frac{\Lambda^i}{i!} \cdot e^{-\Lambda}.$$

The results of calculations of the probabilities of pulse interference extraction from the probability pχ for recursive median filters with different apertures are shown in Fig. 1. In the case of using Haar functions as a carrier, the critical situation will be not only the case when the amplitudes of the informative signals and pulse interference coincide, but the signals are in antiphase, because in this case only the absence of signal (*one digit code combination*) will be recorded.

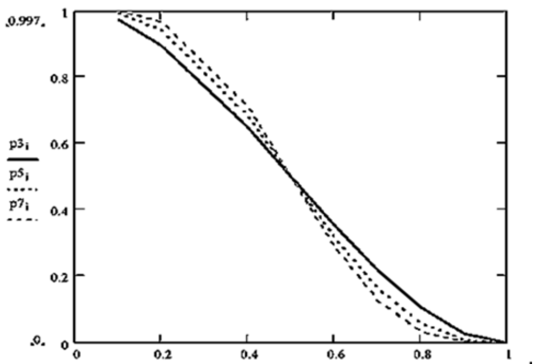

Fig. 1. Calculations of the probabilities of removing the pulse noise p_N from the probability p_χ for recursive median filters with apertures N = 3, 5, 7.

An error may occur if the interference completely mimics the additional Haar function used for this channel.

Development of an algorithm for fast processing of values by a median filter

For the median filter, a fast processing algorithm is quite simply implemented, which is based on the construction of difference matrices using the saturation threshold function $F_{ij} = f(x_i - xj)$, in which

(22) $$f(\Delta x) = \begin{cases} 1, \Delta x \geq 0 \\ 0, \Delta x < 0 \end{cases}.$$

For a filter with aperture N = 5 at the first five values, the matrix F_0 will look like (23):

(23) $$\begin{matrix} f(x_0-x_0) & f(x_1-x_0) & f(x_2-x_0) & f(x_3-x_0) & f(x_4-x_0) \\ f(x_0-x_1) & f(x_1-x_1) & f(x_2-x_1) & f(x_3-x_1) & f(x_4-x_1) \\ f(x_0-x_2) & f(x_1-x_2) & f(x_2-x_2) & f(x_3-x_2) & f(x_4-x_2) \\ f(x_0-x_3) & f(x_1-x_3) & f(x_2-x_3) & f(x_3-x_3) & f(x_4-x_3) \\ f(x_0-x_4) & f(x_1-x_4) & f(x_2-x_4) & f(x_3-x_4) & f(x_4-x_1) \end{matrix}$$

or in generalized form:

(24) $$F_0 = \left\| \begin{matrix} F_{00} & F_{10} & F_{20} & F_{30} & F_{40} \\ F_{01} & F_{11} & F_{21} & F_{31} & F_{41} \\ F_{02} & F_{12} & F_{22} & F_{32} & F_{42} \\ F_{03} & F_{13} & F_{23} & F_{33} & F_{43} \\ F_{04} & F_{14} & F_{24} & F_{34} & F_{44} \end{matrix} \right\|.$$

Offset by one position along a series of values gives the matrix F_1:

(25) $$F_1 = \left\| \begin{matrix} F_{11} & F_{21} & F_{31} & F_{41} & \vdots & F_{51} \\ F_{12} & F_{22} & F_{32} & F_{42} & \vdots & F_{52} \\ F_{13} & F_{23} & F_{33} & F_{43} & \vdots & F_{53} \\ F_{14} & F_{24} & F_{34} & F_{44} & \vdots & F_{54} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ F_{15} & F_{25} & F_{35} & F_{45} & & F_{55} \end{matrix} \right\|,$$

in which you only need to calculate the nine values located in the selected area.

The sum of the differences of the values of F_{ij} by columns:

(26) $$F_i = \sum_{j=0}^{N} F_{ij} = \sum_{j=0}^{N} f\left(x_i - x_j\right),$$

shows the number of definition by value and allows you to sort the registered values of x_j by value: 1 corresponds to the minimum, N – the maximum, and $\frac{(N+1)}{2}$ – the median value.

Thus, to implement the algorithm for fast processing of values by the median filter, it is necessary to perform a number of actions:

- form a matrix F_0 for the first N registered values according to the selected filter aperture;
- calculate the value of $F_{0.j}$ for each of the columns of the matrix F_0;
- select the required value from the first N registered;
- for the matrix F_n we should determine the values of $F_{(n+j)(n+N-1)}$ and $F_{(n+N-1)(n+j)}$ at $0 \leq j < N$;
-from the pre-calculated values $F_{n.j\pm1}$ we should delete values $F_{(n-1)(n+j)}$ and $F_{(n+j)(n-1)}$;
- to the column $F_{(n+j)}$ we should add values to the F_n matrix $F_{(n+j)(n+N-1)}$.

The action continues until all values have been processed.

This algorithm allows, if necessary, instead of the average to choose another ranked element from the minimum to the maximum and can be easily implemented on any processor, including single-chip microcontrollers.

Software development

Based on the general principles of median filtering described above, programmatically, the operation of the median filter can be divided into three stages, which are performed cyclically for each value of the input array. So in a cycle for each value of an array, at first the window with the set size is formed (where the size of a window should not be even), then this window is sorted in ascending order, and from the sorted window the average value which is the result of application of filtering for the set element of entering array is chosen. The obtained value is written accordingly in the resulting array. Each of these three stages is implemented by a separate function: "*Window formatting*", *"Array Sort"*, *"Medium element"*.

"Window formatting" of a given size involves the acceptance as arguments of the window itself and its size. Having a pointer to the current element of the array, in order to form a window you need to copy from the array to the window, n / 2 elements to the left of the current and the same to the right.

The "Array Sort" function is produced by the permutation method. Neighboring elements are swapped until no permutation is performed in one pass of the entire array. The number of such passes is equal to the number of elements in the array. As a result, sorting is implemented as a nested loop: the inner one is responsible for permuting neighboring elements if they are not in ascending order, and the outer loops through the array until the permutation stops.

The "Medium element" function involves setting a pointer to the beginning of the window, which is then shifted in a loop until the middle element is reached. Passing through all the elements of the resulting array in a loop with a postcondition, the program outputs the value of the filtering result to the specified text file.

Program implements a bidirectional, closed connected list to store the values of the input signal. An element of the development list is a structure of type mass, containing three fields. A data field of type double is to store the value of the signal and the fields next and prev, which are pointers to the mass type, to implement the relationship of the current list item with the previous and next items, respectively.

It is advisable to organize the list dynamically, which will allow you to store data of any predetermined size, which will save memory.

The filtering program will be implemented as a separate class – there is a *«massiv»* class. Its members are a «start» pointer of type *«mass»* to the beginning of the list, a *«current»* pointer of the same type to the current element in question, and *«length»* the number of elements in the list. The private methods of the class will store information about the input and output signals, and the public methods will provide a convenient interface for working with them.

The *void add (double val)* method adds an element to the top of the list. As an argument, it takes a value of type *double* – the value of the signal on a certain count. If the list does not yet contain any elements, ie *length* is zero, then the memory is allocated at the beginning of the list, which is indicated by the *start* pointer and the list is looped. If the *length* is greater than zero, the new value is inserted at the beginning of the list, pre-allocating memory for it.

The *void clean* method clears the list. In a loop, passing the list, each current element is deleted, storing in the buffer variable a pointer to the next element.

The method *void outf (ofstream * fo)* performs in a loop on an elemental output of the contents of the list in a file stream of type *ofstream*.

The list is sorted by the *void sort* method. In which the method of sorting by permutations is realized.

For greater clarity of the filtering result, the *massiv* class has a public method- *void plot (int col)*. This method, using the *graphics.h* function library, builds a signal based on specified counts stored in a list. As an argument, the *plot* function takes an integer value of *col*, which specifies the color of the graph. This is necessary when building multiple graphs on one screen, for better resolution. First, the current point is moved to the center of the screen, and then using the *lineto (int, int)* function of the *graphics.h* library, the values of the readings stored in the list are combined by lines.

The *void Child method (massiv*ch, int window)* selects a window of values from this list into a *ch-list* of window length, which is passed to the function as an argument. The list *ch* is formed from *(window-1) / 2* elements that are to the left of the current element and has the same number of elements to the right of the current element.

The *double middle* method returns a list item numbered *(length-1) / 2*, which is the median value after sorting the list. For this purpose it is necessary to pass in a cycle the first number *(length-1) / 2* elements of the list shifting the pointer. In the main function of the program *void main*, two instances of the *massiv* class are created: *m* – saves the input signal, *res* – the output signal. The text menu is implemented in the loop. By scanning the keyboard with the *getch* function and analyzing the codes of the pressed keys that are stored in the *what* variable, the *switch* structure implements the selection of menu items.

If you press the «1» key, which has the code 59, the program prompts the user to enter a file name that contains a set of values of the input signal. Then the values are read one by one from the file and by the add (double) method are added to the list m.

When you press the «2» key, which has the code 50, the program prompts you to add one value to the list. The user enters the value of the count from the keyboard, which is added to the list by the add (double) method.

When you press the «3» key, the signal contained in the list of variable m is filtered. To do this, alternately in the loop for each element of the list m, a window of a given size is selected and stored in the variable t, which is also an instance of the class massiv. The middle value of the list t is determined by the middle method, is added to the resulting list res.

Pressing the «4» key displays the data from the resulting list in a text file.

Experimental verification of the obtained results

To assess the effectiveness of the developed program, it is advisable to generate a discrete signal of a sufficiently complex shape, to expose it to additive white noise of different amplitude and to process the median filter at the selected aperture. To perform calculations, we will set the values of the filter aperture *N* = 3 and *N* = 5. The generated test signal is written to a file, from which the program then reads the data. The signal is selected bipolar with an amplitude of 12 V, the code combination format is one byte. The results of the calculations at values of the filter aperture *N* = 3 are summarized in table1.

In order to analyze the data obtained, you need to plot the graph of input and output signals for each aperture with the appropriate noise ratio. The plotted graphs for each aperture (*N* = 3 and *N*) are presented in Fig. 2 and Fig. 3.

Table 1. Amplitudes of input and output (cleaned) signals of the median filter with an aperture of N = 3

Correlation: U_c/U_ξ = 20		Correlation: U_c/U_ξ = 10		Correlation: U_c/U_ξ = 6.	
U_{in}	U_{out}	U_{in}	U_{out}	U_{in}	U_{out}
-12.44	-11.81	-12.09	-12.09	-11.47	-11.65
-11.71	-11.71	-12.34	-12.29	-11.65	-11.47
-11.61	-11.71	-12.29	-12.29	-10.90	-10.90
-11.90	-11.61	-10.80	-10.80	-10.90	-10.90
12.06	11.88	10.93	10.93	11.34	11.05
11.88	11.88	12.55	12.14	11.05	11.05
11.81	11.88	12.14	12.55	10.83	10.83
11.89	11.89	12.94	12.14	10.35	10.83
11.95	11.89	11.53	11.85	13.02	12.27
11.74	11.95	11.85	11.85	12.27	12.27
12.47	12.12	13.11	12.81	10.38	10.78
12.12	12.46	12.81	12.81	10.78	10.48
12.46	12.12	11.25	11.56	10.48	10.78
11.80	12.04	11.56	11.37	10.89	10.89
12.04	12.02	11.37	11.37	12.18	11.63

Correlation:: U_c/U_ξ = 5		Correlation:: U_c/U_ξ = 4	
U_{in}	U_{out}	U_{in}	U_{out}
-12.93	-11.63	-9.01	-12.74
-10.38	-12.18	-13.72	-10.61
-12.18	-12.18	-10.61	-11.76
-12.69	-12.18	-11.76	-10.61
9.97	9.97	13.69	12.44
10.51	10.51	12.44	12.44
12.24	11.50	9.29	9.97
11.50	11.50	9.97	9.47
11.23	11.50	9.47	9.97
11.53	11.53	10.14	10.14
11.77	11.53	12.30	11.38
10.95	11.77	11.38	11.38
13.86	12.46	11.04	11.38
12.46	13.82	11.41	11.41
13.82	12.46	11.71	11.41
11.18	11.18	10.68	10.68
-14.32	-9.60	-9.00	-9.00

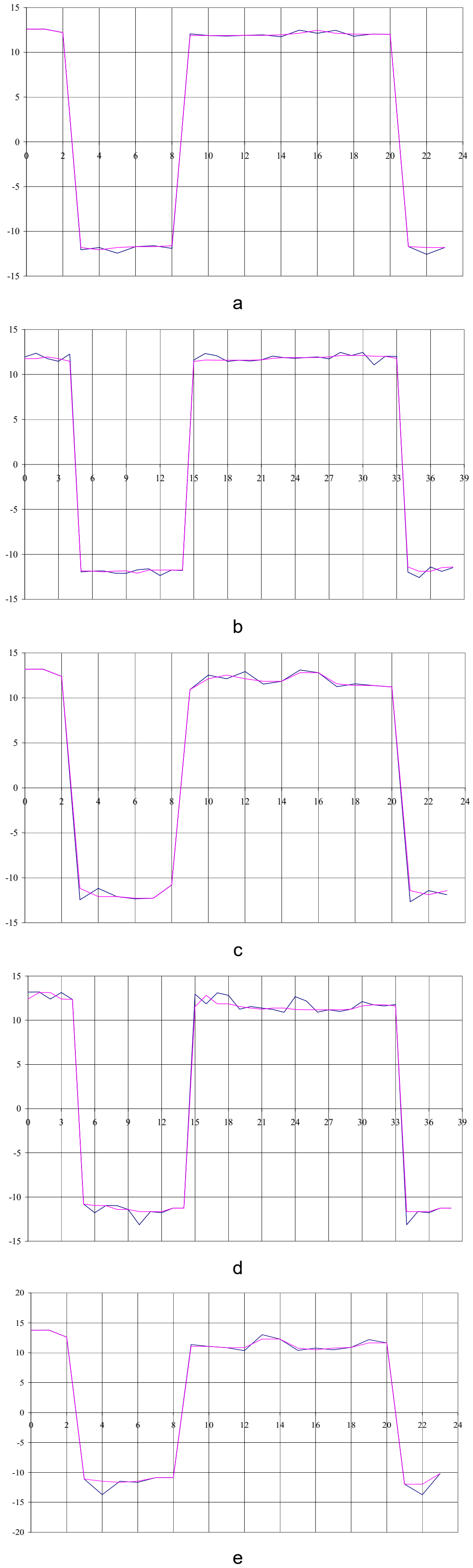

Fig. 2. The results of filtering a rectangular pulse by the median filter: The blue line shows the input signal, by pink is shown the signal after filtering: a) the aperture of the filter 3, the signal/obstacle ratio of 5%; b) 5, 5%; c) 3, 10%; d) 5, 10%; e) 3, 15%

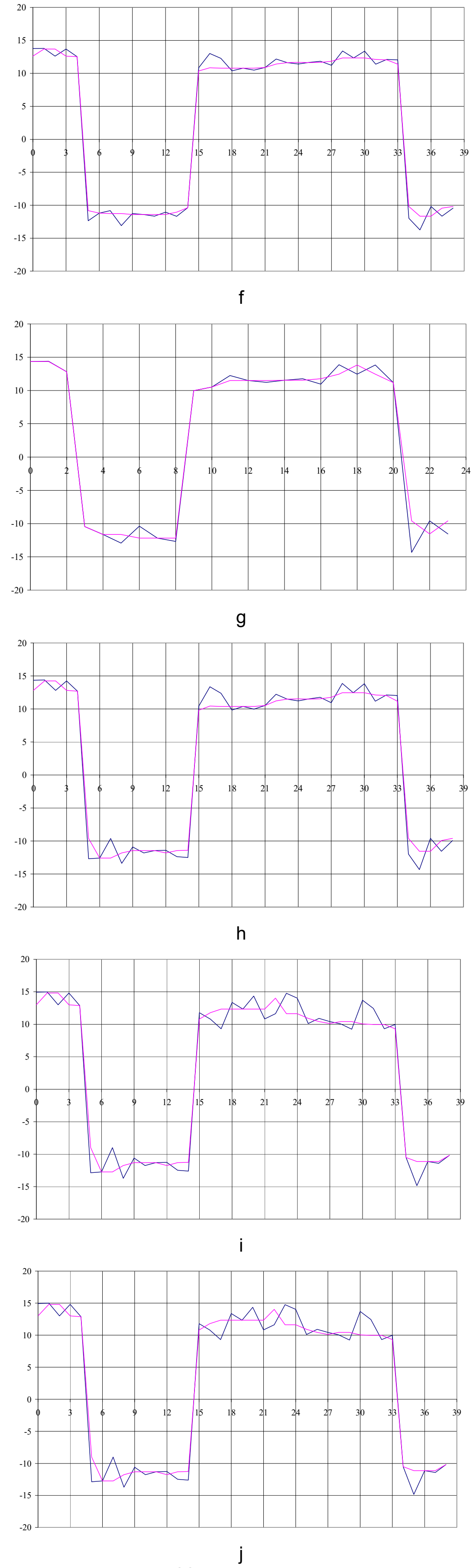

Fig. 3. The results of filtering a rectangular pulse by the median filter: The blue line shows the input signal, by pink is shown the signal after filtering: f) 5, 15%; g) 3, 20%; h) 3, 20%; i) 3, 25%; j) 5, 25%, respectively.

The graphs (Fig. 2, Fig. 3) confirm the high efficiency of the median filters, the theoretical study of which was conducted above.

Conclusions

The classification of existing digital filters is carried out. For further development, a median filter was selected, which belongs to the class of heuristics and is one of the most effective in filtering signals from impulse noise and white noise.

Highlighting the advantages and disadvantages, the review of existing software that implements the median filter is made. It is established that the urgent task is to increase the processing speed and reduce resource costs in the implementation of such filters.

In order to increase the speed of information processing, a median filtering algorithm based on difference matrices using the saturation threshold function has been developed. The software that implements the proposed algorithm is developed. Schemes of the main program, reading of values of a signal from a file, filtering, sorting of data on amplitude, a choice of a window of elements, a choice of the registered values are presented. An experimental verification of the implemented median filters with different apertures at different levels of pulse noise was performed. The conditions of data registration and ADC parameters to ensure efficient operation of filters are also defined.

In many cases, the use of a median filter is more effective compared to others due to the nonlinearity of the characteristic. The most promising is the use of such filters for impulse interference, but this filter also gives positive results for white noise.

Funding. *This research was supported and funded by the Ministry of Education and Science of Ukraine under grant № 0122U000844.*

Authors: TSURKAN Oleh – *Dr. Eng. Sc., Associate Professor, Faculty of Engineering and Technology, Vinnytsia National Agrarian University, director of Separated structural unit «Ladyzhyn Professional College of Vinnytsia National Agrarian University» (24321, 5 P. Kravchyka str., Ladyzhyn, Vinnytsia region, Ukraine), e-mail: tsurkan_ov76@ukr.net;* KUPCHUK Ihor – *PhD of Engineering, Associate Professor, Deputy Dean for Science, Faculty of Engineering and Technology, Vinnytsia National Agrarian University (21008, 3 Sonyachna str., Vinnytsia, Ukraine), e-mail: kupchuk.igor@i.ua;* POLIEVODA Yurii – PhD of Engeneering, Associate Professor, Faculty of Engineering and Technology, Vinnytsia National Agrarian University (21008, 3 Sonyachna str., Vinnytsia, Ukraine), e-mail: vinyura36@gmail.com; VOZNYAK Oleksandr – PhD of Engeneering, Associate Professor, Faculty of Engineering and Technology, Vinnytsia National Agrarian University (21008, 3 Sonyachna str., Vinnytsia, Ukraine), e-mail: alex.voz1966@gmail.com;HONTARUK Yaroslav – *PhD of Economic, Senior Lecturer, Deputy Dean for Science, Faculty of Management and Law, Vinnytsia National Agrarian University (21008, 3 Sonyachna str., Vinnytsia, Ukraine), e-mail: e050122015@gmail.com;* PRYSIAZHNIUK Yrii – *PhD Student (Engeneering), Research Fellow, Faculty of Engineering and Technology, Vinnytsia National Agrarian University (21008, 3 Sonyachna str., Vinnytsia, Ukraine), e-mail: prisaznuku13@gmail.com*.

REFERENCES

[1]. Kaletnik G., Honcharuk I., Yemchyk T., Okhota Yu. The world experience in the regulation of the land circulation, *European Journal of Sustainable Development*, 9 (2020), nr 2, 557-568. https://doi.org/10.14207/ejsd.2020.v9n2p557

[2]. Mazur V., Tkachuk O., Pantsyreva H., Kupchuk I., Mordvaniuk M., Chynchyk O. Ecological suitability peas (Pisum Sativum) varieties to climate change in Ukraine, Agraarteadus, 32 (2021), nr 2, 276-283. https://doi.org/10.15159/jas.21.26

[3]. Palamarchuk V., Krychkovskyi V., Honcharuk I., Telekalo N. The modeling of the production process of high-starch corn hybrids of different maturity groups. European Journal of Sustainable Development, 10 (2021), nr 1, 584-598. https://doi.org/10.14207/ejsd.2021.v10n1p584

[4]. Palamarchuk V., Telekalo N. The effect of seed size and seeding depth on the components of maize yield structure. Bulgarian Journal of Agricultural Science, 24 (2018), nr 5, 785–792.

[5]. Kupchuk I., Yaropud V., Hraniak V., Poberezhets Ju., Tokarchuk O., Hontar V., Didyk A. Multicriteria compromise optimization of feed grain grinding process. Przegląd Elektrotechniczny. 97 (2021), nr 11, 179-183. https://doi.org/10.15199/48.2021.11.33

[6]. Kuznietsova I., Bandura V., Paziuk V., Tokarchuk O., Kupchuk I. Application of the differential scanning calorimetry method in the study of the tomato fruits drying process, *Agraarteadus*, 31 (2020), nr 2, 173–180. https://doi.org/10.15159/jas.20.14

[7]. Vedmitskyi Y. G., Kukharchuk V. V., Hraniak V. F., New nonsystem physical quantities for vibration monitoring of transient processes at hydropower facilities, integral vibratory accelerations, *Przeglad Elektrotechniczny*, 93 (2017), nr.3, 69-72

[8]. Yaropud V., Hunko I., Aliiev E., Kupchuk I. Justification of the mechatronic system for pigsty microclimate maintenance. *Agraarteadus*. 32 (2021), nr 2, 212–218. https://doi.org/10.15159/jas.21.21

[9]. Borysiuk D., Spirin A., Kupchuk I., Tverdokhlib I., Zelinskyi V., Smyrnov Ye., Ognevyy V. The methodology of determining the place of installation of accelerometers during vibrodiagnostic of controlled axes of wheeled tractors, *Przegląd Elektrotechniczny*, 97 (2021), nr 10, 44-48. https://doi.org/10.15199/48.2021.10.09

[10]. Gunko I., Hraniak V., Yaropud V., Kupchuk I., Rutkevych V. Optical sensor of harmful air impurity concentration, *Przegląd Elektrotechniczny*, 97 (2021), nr 7, 76-79. https://doi.org/10.15199/48.2021.07.15

[11]. Hrushetskyi S., Yaropud V., Kupchuk I., Semenyshena R., The heap parts movement on the shareboard surface of the potato harvesting machine, *Bulletin of the Transilvania University of Braşov. Series II: Forestry, Wood Industry, Agricultural Food Engineering,* 14 (2021), nr 1. 127-140. https://doi.org/10.31926/but.fwiafe.2021. 14.63.1.12

[12]. Solona O., Kovbasa V., Kupchuk I. Analytical study of soil strain rate with a ploughshare for uncovering slit. *Agraarteadus*. 2020. Vol. 31, №2. P. 212–218. https://doi.org/10.15159/jas.20.22

[13]. Kovbasa V., Solona O., Deikun V., Kupchuk I. Functions derivation of stresses in the soil and resistance forces to the motion of a plough share for cavity creation. *UPB Scientific Bulletin, Series D: Mechanical Engineering,* 83 (2021), nr 3, 305–318

[14]. Kukharchuk V., Katsyv S., Hraniak V. Analysis of dependency between current harmonics coefficient and load, as well as filter parameters for asymmetrical network modes, *Przeglad Elektrotechniczny*, 96 (2020), nr. 9, 103-107. https://doi.org/10.15199/48.2020.09.22

[15]. Paziuk V., Vyshnevskiy V., Tokarchuk O., Kupchuk I. Substantiation of the energy efficient schedules of drying grain seeds. *Bulletin of the Transilvania University of Braşov, Series II: Forestry, Wood Industry, Agricultural Food Engineering*, 63 (2021), nr. 14, 137–146. https://doi.org/10.31926/but.fwiafe.2021.14.63.2.13

[16]. Honcharuk I., Kupchuk I., Solona O., Tokarchuk O., Telekalo N., Experimental research of oscillation parameters of vibrating-rotor crusher, *Przeglad Elektrotechniczny*, 97 (2021), 3. 97–100. https://doi.org/10.15199/48.2021.03.19

[17]. , Semenov A. A., Baraban S. V., Semenova O. O., Voznyak O. M., Vydmysh A. A., Yaroshenko L. V Statistical Express Control of the Peak Values of the Differential-Thermal Analysis of Solid Materials. Solid State Phenomena. 2019. Vol. 291. P. 28-41. https://doi.org/ 10.4028/www.scientific.net/SSP.291.28

[18]. Solona O., Kupchuk I. Dynamic synchronization of vibration exciters of the three-mass vibration mill, *Przegląd Elektrotechniczny*, 1 (2020), nr. 3, 163–167. https://doi.org/10.15199/48.2020.03.35

[19]. Matvijchuk V., Shtuts A., Kolisnyk M., Kupchuk I., Derevenko I. Investigation of the tubular and cylindrical billets stamping by rolling process with the use of computer simulation. *Periodica Polytechnica Mechanical Engineering*, 66 (2022), nr 1, 51-58. https://doi.org/10.3311/PPme.18659

[20]. Rutkevych V., Kupchuk I., Yaropud V., Hraniak V., Burlaka S. Numerical simulation of the liquid distribution problem by an adaptive flow distributor. *Przegląd Elektrotechniczny*, 98 (2022), nr 2, 64-69. https://doi.org/10.15199/48.2022.02.13