UDC 004.93:004.94
DOI: 10.15587/1729-4061.2024.318558

DEVISING A COMPUTER METHOD TO RECOGNIZE AND ANALYZE SPECTROMETRIC SIGNALS PARAMETERS

Sergiy Reva
PhD*
Denys Tsybliyev
Corresponding author
PhD Student*
E-mail: dtsibliev@gmail.com
*Department of Computer Systems and Robotics
V. N. Karazin Kharkiv National University
Svobody sq., 4, Kharkiv, Ukraine, 61022

The object of this study is computerized systems for measuring the parameters of spectrometric signals digitized using special hardware. The task addressed in the research is to improve the process of filtering the usable pulse signal from noise and increase the accuracy of measuring pulse parameters by devising a new method of analysis. In order to verify the performance of the new method in comparison with several already known ones, input data arrays with predetermined parameters were prepared using computer simulation. A special algorithm was also developed to verify each detected pulse. As a result, the main characteristics of the methods, such as signal recognition accuracy and data processing speed, were obtained for several scenarios with different durations of modeling process and different pulse generation intensities. Comparative performance metrics were provided for all described software analysis methods. Ultimately, in the studied scenarios, the devised method showed better recognition ability than the considered alternative methods.

The key features of the proposed method are the use of software filters built on the basis of the application of Fast Discrete Fourier Transform (FDFT) algorithms and further computer processing of the signal using a mechanism for correcting the amplitudes of superimposed pulses. This makes it possible to filter the signal from noise without significantly changing the usable component and to more accurately determine the amplitudes in case of their frequent superposition. In practice, the devised method could be used to improve existing and design new computer systems of spectral analysis

Keywords: computer analysis of spectrometric signals, digital signal filtering, computer simulation, recognition algorithms, fast discrete Fourier transform

Received 09.10.2024
Received in revised form 28.11.2024
Accepted 12.12.2024
Published date 27.12.2024

How to Cite: *Reva, S., Tsybliyev, D. (2024). Devising a computer method to recognize and analyze spectrometric signals parameters. Eastern-European Journal of Enterprise Technologies, 6 (9 (132)), 86–96. https://doi.org/10.15587/1729-4061.2024.318558*

1. Introduction

Over a long time, classical methods of spectral analysis of X-ray and gamma radiation were based on the use of analog electronics. However, owing to the active development of computer technologies, computer methods of recording and measuring spectra have become widely used. Leading companies working in the field of designing spectrometric equipment today offer a large number of proposals for signal digitizers, which are actively used in modern spectrometric systems. In general, the principle of their operation involves recognition according to certain criteria and digital recording of individual frames (time segments), during which the usable signals of the detector are registered.

To acquire digital data for spectral analysis, special equipment is used, which usually includes a detector (scintillation or semiconductor), a signal digitizer, a computer or a distributed computer data processing system, and the radiation source itself. The spectrometric signal produced by the detector is a sequence of pulses that are generated at random time points and have different amplitudes (Fig. 1). It is they, or rather their parameters, that carry coded useful information about the materials and processes under investigation [1]. However, the random nature of the appearance of pulses often leads to their mutual overlap (the so-called pile-up effect), as a result of which their amplitude and duration change, and this, in turn, leads to the distortion of primary information. It is almost impossible to restore it completely by analog spectrometry methods, that is, the resulting spectrum always has certain errors.

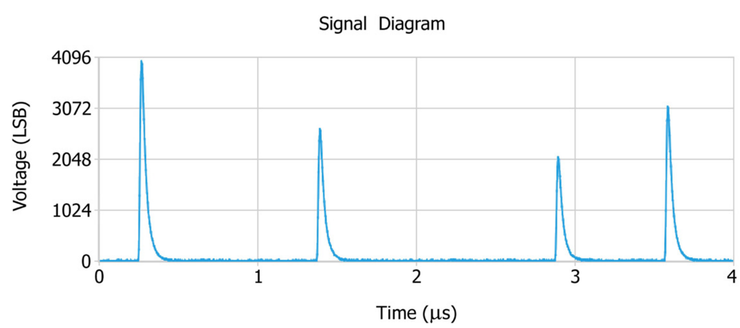


Fig. 1. An example of software generation of a fragment of the detector's output signal

Unlike analog measurement methods, in which signal processing is carried out exclusively by hardware, computer analysis provides additional opportunities. This includes not only the use of mathematical methods for processing the digitized signal but also its logical analysis in order to restore partially lost information, which could improve the reliability of the final result. However, it is worth noting that simpler methods that work quickly may not show the best recognition accuracy, especially when pulses are often superimposed. In turn, methods that perform complex additional processing of the digitized signal and show better recognition accuracy may work more slowly. It is logical to seek to improve both of these parameters or to optimize the ratio of recognition accuracy and speed of analysis methods.

In addition, there is a problem of objective evaluation of the efficiency of this or that computer processing method since there is no possibility of obtaining completely reliable input data due to the randomness and unpredictability of the processes at the input of the detector. Experimental research methods are quite expensive and evaluate the effectiveness of computer processing only by indirect parameters for clearly defined experimental conditions. This problem could be solved with the help of computer simulation of digital input data with previously known parameters and verification of the correctness of pulse recognition based on the results of the analysis.

Therefore, both devising computer research methods and introducing new methods for computer analysis of spectrometric signals, which would improve the accuracy or speed of obtaining the final result, are relevant tasks. Computer simulation of reference input data and the introduction of criteria and algorithms for the verification of digital processing results are also important related directions. In practice, this could be used for testing purposes when designing new and improving existing computerized systems for spectral analysis.

2. Literature review and problem statement

Current research in the field of computer analysis of spectrometric signals often emphasizes the importance of mechanisms for reducing the influence of electrical noise in the data processing process and is aimed at improving the efficiency of analysis methods. Thus, in [2], a method for recognizing the pulses of a digitized signal received from a scintillation detector was described. Several possible options for filtering the signal from electrical noise using various filters, such as moving average, Chebyshev, Bessel, Butterworth, etc., were also considered. But it is worth noting that the use of the above-mentioned filters could also modify the signal itself in the direction of slowing down the growth, changing the amplitude or in some other way. This could lead to deviations in the determination of the amplitude or the exact time of the appearance of the pulse, especially in the case of short pulses with a sharp front and decline. The possibility of filtering the signal from electrical noise without significant modification of the usable component being analyzed remains an open question in the cited study.

Paper [3] reports a clustering algorithm for recognizing the parameters of digitized signals from a detector for radiation particles of various types (neutrons, gamma quanta). The principle of its operation is generally based on determining template waveforms of several types of pulses and the use of matched filter technique in the analysis process. However, the effectiveness of this approach needs further study, especially with a high probability of pulse overlap, because the shape of the signal could be significantly changed in this case and the recognition accuracy could be low.

In [4], an algorithm was presented to reduce the influence of the pile-up effect on the resulting spectrum. The algorithm is based on the sparse approximation of the signal to separate superimposed pulses and the use of Least Absolute Shrinkage and Selection Operator (LASSO) regression analysis methods. The reported approach made it possible to reduce the influence of the pile-up effect and to improve the resulting investigated spectrum in the energy range of 0–1000 keV; however, the artifacts detected in the range of 1500 keV and above require further analysis. Since the performance of the proposed algorithm was tested on real spectra, the analysis of the results of its operation on simulated known input data with the addition of verification would likely make it possible to better investigate the accuracy of pulse recognition by this method.

In works [5, 6], several methods of analysis of spectrometric signals were described at once, such as the method of Maximum, Sum, Fitting, and Deconvolution. Each of them contains its specific approach to reduce the effect of electrical noise on recognition results. The mechanisms of mathematical modeling of the form of individual pulses were considered, the results of the work of the above methods on simulated digital data were obtained, and the dependence of the recognition of the number of pulses on the detector load level was visualized on the diagrams. In addition, spectra were acquired using each of the methods and their comparative characteristics were given. It is worth noting that the Fitting and Deconvolution methods showed the best data analysis results. In general, the studies provided detailed performance characteristics of the methods described, such as the total number of detected pulses, the full width at half maximum for peaks (FWHM) of different energies, the noise impact, and the processing time. Taking into account the fact that at high loading levels (the number of pulses per unit time) several pulses could overlap, forming one with a large amplitude, the verification of the recognition results by each method remains an unresolved issue. That is, checking whether the recognized pulse matches the one that was generated during the simulation, which is important to objectively determine the recognition accuracy. In addition, the results reported in these works show that at high loading levels, the efficiency of analysis methods decreases due to the superposition of pulses, so the issue of increasing recognition accuracy remains relevant.

There are also studies that analyze spectrometric signals using neural networks and artificial intelligence technologies, in particular, using Convolutional Neural Network (CNN) [7, 8]. However, this technology was used to solve the problems of distinguishing one type of radiation quanta from others (for example, neutrons from gamma quanta), and not for recognizing individual pulses and measuring their parameters.

Thus, the issue of the possibility of implementing electrical noise filtering with minimal changes to the digital signal itself and increasing recognition accuracy by devising a new analysis method remains unresolved. In addition, reviewed works do not provide criteria for assessing the reliability of recognized pulses, which would allow for an objective assessment of the accuracy of their recognition when using different methods for digital signal processing.

3. The aim and objectives of the study

The aim of our study is to improve accuracy in recognizing the parameters of spectrometric signals by devising an analysis method with an improved mechanism for filtering the digitized signal from electrical noise and additional processing of pulses during superposition. This will make it possible to improve the work of existing computerized systems for analyzing the parameters of spectrometric signals (and visualizing their quantitative and qualitative characteristics) or newly designed hardware and software complexes.

To achieve this goal, the following tasks were set:

– to build reference input data for analysis methods by computer simulation of digital data with an artificial, idealized distribution of pulse amplitudes, and develop an algorithm for verification of recognition accuracy;

– to implement an approach to filtering the spectrometric signal from noise without significant changes in the usable component and develop a pulse recognition algorithm with tracking of the level of the processed signal and correction of pulse amplitudes during superposition;

– to evaluate the effectiveness of developed algorithms by researching and comparing the implemented method with several existing ones, namely: the Maximum method and the Fitting method; get verified metrics of their effectiveness.

4. The study materials and methods

The object of our research is computerized systems for measuring the parameters of spectrometric signals digitized with the help of hardware.

The main hypothesis of the research assumes that the application of Discrete Fourier Transform (DFT) algorithms could make it possible to implement software filtering of the digitized signal from electrical noise without significantly changing the usable component. The hypothesis also suggests that there is a way to monitor the level of the processed signal for the purpose of recognizing individual pulses and correcting their amplitudes when superimposed, which could improve the accuracy of analysis.

In the course of the study, it was assumed that the random number generation mechanism of the standard library in the C++ programming language would make it possible to generate electrical noise with a distribution that sufficiently corresponds to a normal distribution.

The following simplifications were adopted. The value of the baseline of the signal (that is, the level of the electrical signal under the condition that there are no pulses in the current time period) is zero. The maximum level of electrical noise during simulation and in the process of signal analysis is fixed and must be set manually in the program interface.

The theoretical foundations of the research are based on the use of principles of system analysis, mathematical modeling and computer simulation, numerical methods, methods and algorithms for analytical processing and intelligent analysis of large data sets. In addition, Discrete Fourier Transform (DFT) algorithms, information visualization methods, and our original algorithms were used.

The evaluation of the effectiveness of the application of certain methods for digital processing of spectrometric signals should be the degree of reliability of the results and their correspondence to the real signals that were registered by the detector during the experiment. However, the problem is that it is not known exactly which spectral components of the radiation led to the generation of a signal of this particular shape, and whether a single pulse is the result of the registration of one gamma quanta, or it was formed as a result of the superposition of several responses of the detector to gamma quanta with other energies. In addition, the presence of noise in the measurement channel could distort the amplitude and time of occurrence of pulses in the output signal of the detector. Determining the influence of these factors on the formation of the spectrum and its reliability, as well as the effectiveness of digital methods of information recovery, cannot apply because of the lack of guaranteed input data of the experiment. That is why it was decided to investigate various methods for digital processing of spectrometric data based on an artificially generated sample signal with fully known parameters of each of the pulses that make up its composition.

As part of the research, a program was developed that implements the possibility of computer simulation of data [9], which in terms of format correspond to the output data of the digitizer, obtained during experiments. The application provides an opportunity to simulate a signal close to the real one with a previously specified law of distribution of pulse amplitudes, or to load data obtained directly with the help of a digitizer. However, in contrast to a well-defined spectral composition, during simulation, the program provides a random distribution of pulses in time, which brings the result closer to the data of a real experiment and creates some uncertainty in the form of the output signal. It was also possible to simulate based on special template files previously recorded using spectrometric equipment. They are tabular energy distribution functions (pulse amplitudes) and are a numerical representation of various energy spectra. If necessary, one could also use specially built (idealized) distribution functions. When developing the application, spectrum files obtained with a 12-bit analog-to-digital converter (ADC) with 4096 channels were used as templates. Mathematical models and a modeling algorithm based on deterministic final amplitude values, which are described in more detail in [10], were used to model a digital signal with the required amplitude distribution law according to template files.

The program contains implementation of two existing methods for determining the parameters of spectrometric signals (Maximum method and Fitting method). A new analysis method called Tracking using proprietary algorithmic approaches was also developed. The results of the above-mentioned methods are visualized in the form of numerical parameters, time diagrams of signals, and histograms of spectra built on the basis of analysis. The application is developed in the C++ programming language (standard 14, 17) [11, 12] using the QT library [13]. This library was chosen because it contains a wide set of graphical components for data visualization and also allows for cross-platform software development.

In order to comprehensively check the operation of the analysis methods, digital input data were simulated according to the prepared template file, and each of the described methods was launched and their effectiveness metrics were obtained. Measurements were carried out according to three scenarios on different arrays of data, simulated with the following parameters:

1. The detector load level (pulse generation intensity) is 10^5 pulses per second, the duration of the experiment is 3.072 seconds, the total number of simulated pulses is 307,200.

2. The loading level of the detector is 10^6 pulses per second, the duration of the experiment is 0.3072 seconds, the total number of simulated pulses is 307200.

3. The detector load level is 10^7 pulses per second, the duration of the experiment is 0.03072 seconds, the total number of simulated pulses is 307200.

The electrical noise imposed on the signal was modeled according to a normal distribution with a mathematical expectation parameter equal to zero and a standard deviation equal to eight values of the digitizer's least significant bit (LSB). At the end, for each of the analysis methods, information about the duration of its operation (in seconds), the total and verified number of recognized pulses, and the accuracy of the method (in percent) was measured and outputted. To verify the correctness of the recognition, a verification algorithm was developed that compares the generated and recognized pulses for coincidence with the permissible deviation ranges that could be specified in the program interface. During this comprehensive inspection, the possible amplitude deviation range was set equal to three times the maximum value of the electrical noise level. The verified accuracy of the analysis method was determined by the ratio of the number of verified pulses to the total number of simulated ones. Measurements were performed on a computer with an Intel Core i7 1185G7 processor @ 3.00 GHz, 16 Gb RAM, Windows 11 Pro (USA).

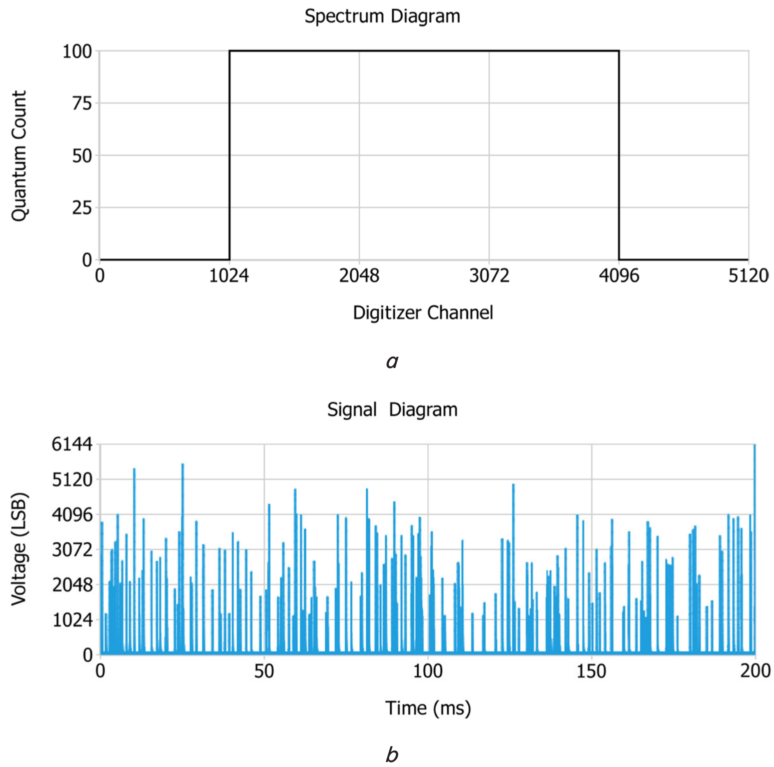


Fig. 2. Modeling according to a template file with a linear spectrum: *a* — distribution of pulse amplitudes; *b* — part of the simulated digital signal

5. Results of investigating spectrometric signal analysis methods

5. 1. Compilation of reference input data by means of computer simulation and development of an algorithm for verification of detected pulses

To check the effectiveness of spectrometric analysis methods, first of all it was necessary to compile reference input data with known parameters. This would make it possible at the end of the analysis to compare the generated and recognized pulses and to draw objective conclusions about the recognition accuracy. Such data were compiled by computer simulation of a digital pulse signal corresponding to the signal obtained during the registration of gamma quanta in a scintillation detector, based on a specially prepared template file with an artificial idealized linear spectrum. Simulation based on it made it possible to simulate a signal that contains the same number of pulses (100 per channel) with amplitude values from 1024 LSB to 4095 LSB. Thus, on the spectral diagram, this distribution looks like a step function shown in Fig. 2, *a*. The X-axis displays the numbers of the ADC channels (from 0 to 5120), and the Y-axis displays the number of pulses with the corresponding amplitude registered in these channels.

Fig. 2, *b* shows a diagram that visualizes a small part of the simulated digital signal, because the total number of pulses is quite large. But, as could be seen from the figure, separate signals appeared in this interval, the amplitude of which exceeds the value of 4095 LSB. They were formed due to the mutual superposition of two or more pulses, the periods of existence of which partially or completely coincide.

In order to objectively assess the accuracy of recognition by one or another method, a special verification algorithm was developed. This algorithm compares simulated and software-detected pulses (time of appearance, amplitude) for coincidence with certain permissible ranges of deviation. Verification is an important component because the pulses could overlap, especially at high loading levels, and the overall recognized pulse may not exactly correspond to what was generated during the simulation. Below is the software implementation of the verification algorithm in the C++ programming language, which illustrates its principle of operation:

```
    void VerifyDetectedPulses(
    const std::vector<SimulatedPulse>& simulated_pulses,
    const std::vector<DetectedPulse>& detected_pulses,
    std::vector<DetectedPulse>& verified_pulses)
{
    verified_pulses.clear();

    auto dp_iter = detected_pulses.cbegin();
    for(const auto& sim_pulse : simulated_pulses)
    {
        if(dp_iter == detected_pulses.cend())
                break;

        if(dp_iter->StartIdx > sim_pulse.Time +
        DigSamplesRange)
                continue;

        while(dp_iter != detected_pulses.cend() &&
                sim_pulse.Time > DigSamplesRange &&
                dp_iter->StartIdx < sim_pulse.Time -
        DigSamplesRange)
```

```
        {
                ++dp_iter;
        }

        if(dp_iter == detected_pulses.cend())
                break;

        auto dp_iter_copy = dp_iter;

        while(dp_iter_copy != detected_pulses.cend() &&
        IsValueInRange(dp_iter_copy->StartIdx,
        sim_pulse.Time, DigSamplesRange))
        {
                if(IsValueInRange(
        dp_iter_copy->AmplitudeCorrected(),
        sim_pulse.Amplitude,
        AmplitudeRange))
        {
                verified_pulses.push_back(
                *dp_iter_copy);
                if(dp_iter == dp_iter_copy)
                        ++dp_iter;

                break;
                }

        ++dp_iter_copy;
        }
    }
}
```

Verification is performed in the VerifyDetectedPulses method, where simulated_pulses is an array of simulated pulses, detected_pulses is an array of detected pulses, verified_pulses is the resulting array of verified pulses, DigSamplesRange and AmplitudeRange are the permissible ranges of deviation in registration time and amplitude for simulated and detected pulses, IsValueInRange is an auxiliary method that checks whether the passed value falls within the passed range. The above pulse arrays are sorted in ascending order by the time of pulse appearance.

5. 2. Implementation of an approach to filtering the spectrometric signal from noise without significant changes in the usable component and development of a pulse recognition algorithm with amplitude correction during superposition

The main approach to filtering the signal from noise in the devised method is based on the use of Discrete Fourier Transform (DFT) [14]. The application of direct DFT transforms a discrete signal into elementary harmonic components with different frequencies, and is calculated from the following formula [15]:

$$S[k]=\sum_{n=0}^{N-1} s[n]\exp\left(-\frac{2\pi ikn}{N}\right), \quad (1)$$

for $n=0, 1, \ldots, N-1$, $k=0, 1, \ldots, N-1$, where N is the size of the array of digitized signal values $s[n]$. The DFT result, $S[k]$ is a sequence of complex numbers (of the form $a+ib$) that is a function of discrete frequency components k, not time.

Filtering is performed by processing the $S[k]$ array and cutting off the high-frequency component, which in this case is electrical noise. To this end, it is necessary to set to zero the elements of the array $S[k]$, which have a magnitude value$\left(\sqrt{a^2+b^2}\right)$ below a certain threshold M_{tr} (algorithm parameter, in the software implementation of the devised method, the value was set to 500). If one applies the inverse Discrete Fourier Transform (IDFT) after that, one could get the value of the filtered signal $sf[n]$ from the following formula:

$$sf[n]=\frac{1}{N}\sum_{n=0}^{N-1} S[k]\exp\left(\frac{2\pi ikn}{N}\right). \quad (2)$$

It is worth noting that DFT transformations according to formulas (1) and (2) are very expensive in terms of calculations because they have a computational complexity of $O(N^2)$. However, there is a Fast Discrete Fourier Transform (FDFT) algorithm [16], which has a computational complexity of $O(N\log N)$ and works much faster, especially on large data sets. One of the possible options for implementing the FDFT algorithm in C++ could be found in [17]. The main condition for its use is that the size of the array to be transformed must be a multiple of 2 or must be padded with zeros at the end to reach the size of such a multiple. Fig. 3 shows the result of applying the filter using the method described above and the visualization of the shape of the original and filtered signal.

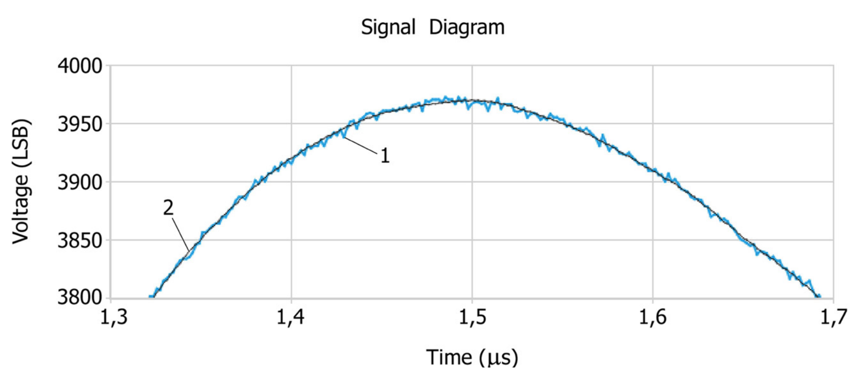


Fig. 3. General view of the filtered signal using FDFT algorithms: 1 – original signal; 2 – filtered signal

After applying filtering, recognition of individual pulses could be performed by sequentially analyzing the level of the processed signal for rising or falling, determining the beginning and end of each pulse, and correcting their amplitudes during superposition.

The detailed principle of operation of the devised method, which includes both the signal filtering stage and pulse recognition, consists of the following steps:

1. Splitting the entire array of signal values into separate consecutive uniform segments that are a multiple of 2 in order to optimize calculations. The last segment is padded with zero values to the desired size if necessary. The following steps are performed sequentially for each segment.

2. Filtering the array of digital values of the signal $s[i]$ by the algorithm described above based on FDFT and obtaining the array of values of the filtered signal $sf[i]$.

3. Pass through the $sf[i]$ array and analyze the previous and current value of the signal level, monitoring the signal state parameter (SignalState), which may take the following values: Baseline (the signal does not exceed the baseline value by the maximum value of the electrical noise), SignalGrowth (growth signal), SignalEquality (signal equality), SignalDescend (signal decline). Possible state transitions, transition conditions, and associated pulse recognition operations are depicted on the UML [18] state diagram [19] in Fig. 4. It should be noted separately that the detected pulse is added to the array of new detected pulses only if its amplitude exceeds the minimum threshold $A_{\min}$, which is a parameter of the algorithm. In the program, this parameter was set to 50 LSB. Otherwise, the found pulse is considered a continuation of the current pulse. This is done in order to reject false detections, which may be caused by random deviations of the falling component of the pulse signal.

4. For more correct processing of cases of overlapping pulses, an additional correction is performed when calculating the final value of the amplitude of the detected pulse. This is performed by subtracting the value of the simulated falling signal component of the previous pulses, calculated from the following formula [5]:

$$s_i' = s_i - \sum_{k=0}^{j-1} A_k p(i - t_k), \qquad (3)$$

where s_i' is the corrected value of the signal, s_i is the original value of the signal, A_k and t_k are the amplitude and time of appearance of the k-th previous pulse, respectively, $p(t)$ is the shape of the pulse generated by one registered quantum (a detailed description could be found in [5]) .

After the software implementation of the new method and the analysis of the signals simulated according to the linear pattern (Fig. 2, *a*), diagrams of the distribution functions of the pulse amplitudes were constructed. These diagrams visualize the distribution of amplitudes of simulated, recognized, and verified pulses at load levels of 10^5 (Fig. 5, *a*), 10^6 (Fig. 5, *b*), 10^7 (Fig. 5, *c*) pulses per second.

The diagrams of the distribution of pulse amplitudes recognized using the devised method show a slight deviation from the distribution line of simulated pulses at the loading level of 10^5 (Fig. 5, *a*), as well as at the level of 10^6 (Fig. 5, *b*) pulses per second. The calculated verified recognition accuracy for these scenarios was 99.52 % and 94.95 %, respectively. At a load level of 10^7 (Fig. 5, *c*) pulses per second, the deviation is significant, and the accuracy of the analysis was 54.26 %.

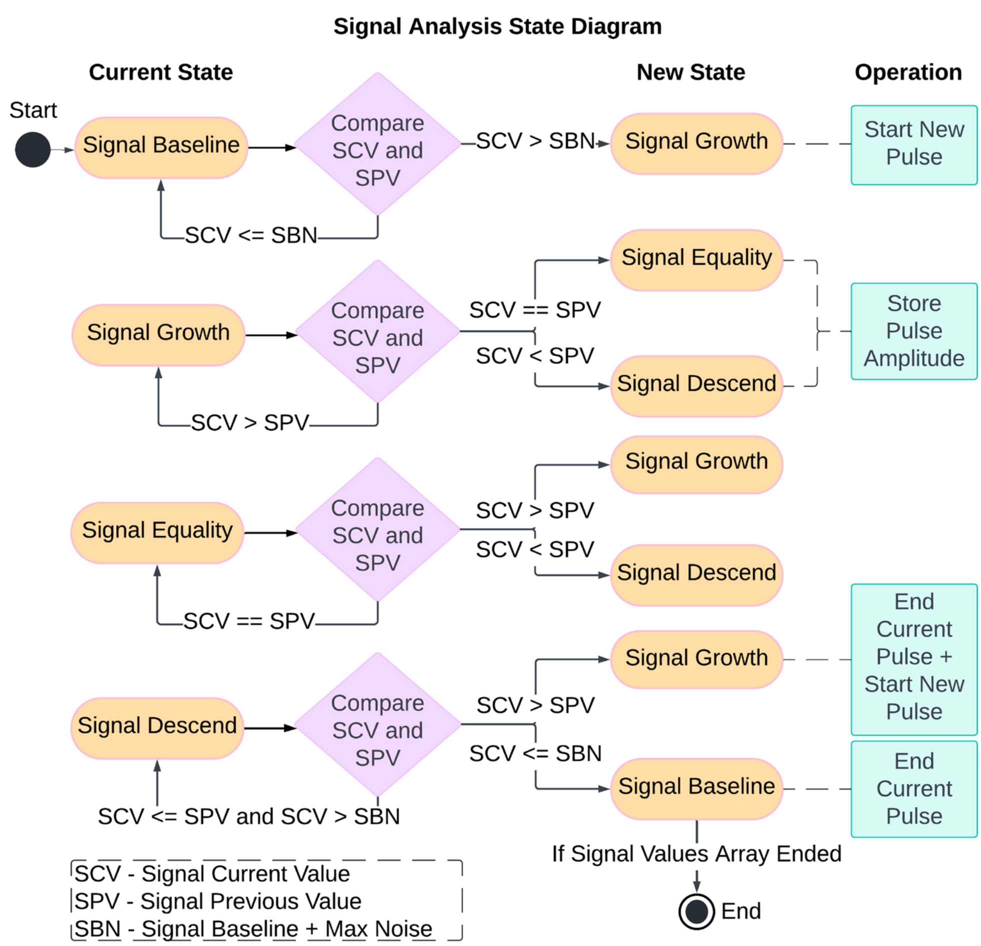


Fig. 4. UML diagram of states and operations in the process of analysis using a new method

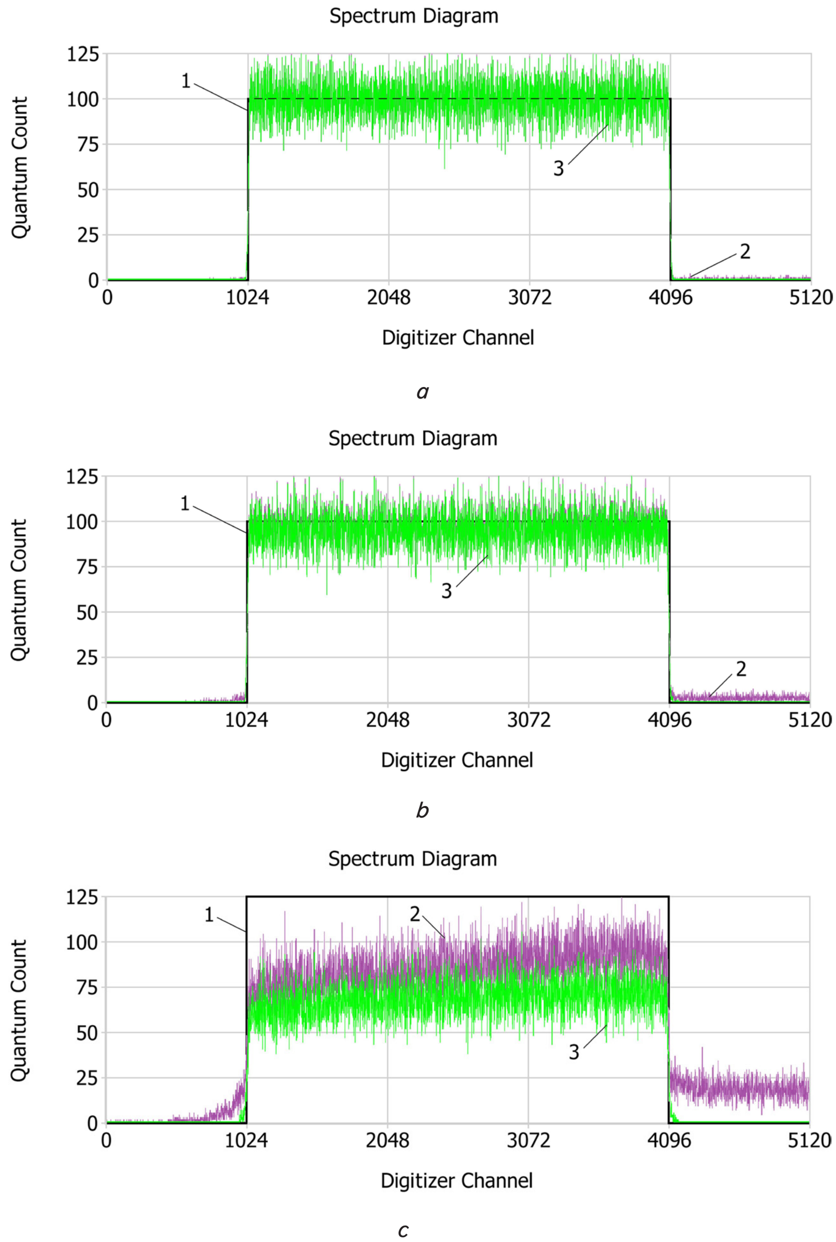


Fig. 5. Amplitude distribution diagrams: 1 – simulated; 2 – recognized; 3 – verified pulses obtained by the devised method at load levels (pulses per second): *a* – 10^5; *b* – 10^6; *c* – 10^7

5. 3. Evaluating the effectiveness of the implemented method by comparison with the existing methods of Maximum and Fitting, obtaining the final verified metrics of their effectiveness

In order to compare the effectiveness of the devised and already known methods of spectral analysis, two existing methods – Maximum and Fitting – were implemented in the course of the study, and metrics of their effectiveness were also obtained. The Maximum method is one of the simplest and fastest, and the Fitting method has shown some of the best results of the analysis before. A detailed description of these methods could be found in [5].

Visualization of the results of the Maximum method, namely the diagram of the distribution of the amplitudes of the simulated and recognized pulses at different loading levels, is shown in Fig. 6.

As could be seen from the diagram in Fig. 6, *a*, the distribution of the amplitudes of pulses recognized by the Maximum method showed a relatively small deviation from the horizontal distribution line at a load level of 10^5 pulses per second. The software-calculated verified recognition accuracy in this case was 97.57 %. However, at higher loading levels (diagrams in Fig. 6, *b*, *c*), the deviation from the idealized distribution line becomes significant. In the case of data analysis simulated with a loading level of 10^6 pulses per second, the verified recognition accuracy decreased to 77.77 %, and at a loading of 10^7 it was only 5.98 %.

The Fitting method performs more complex processing and makes it possible to distinguish superimposed pulses. Fig. 7 shows the distribution diagrams of the amplitudes of simulated, recognized, and verified pulses by this method at different loading levels.

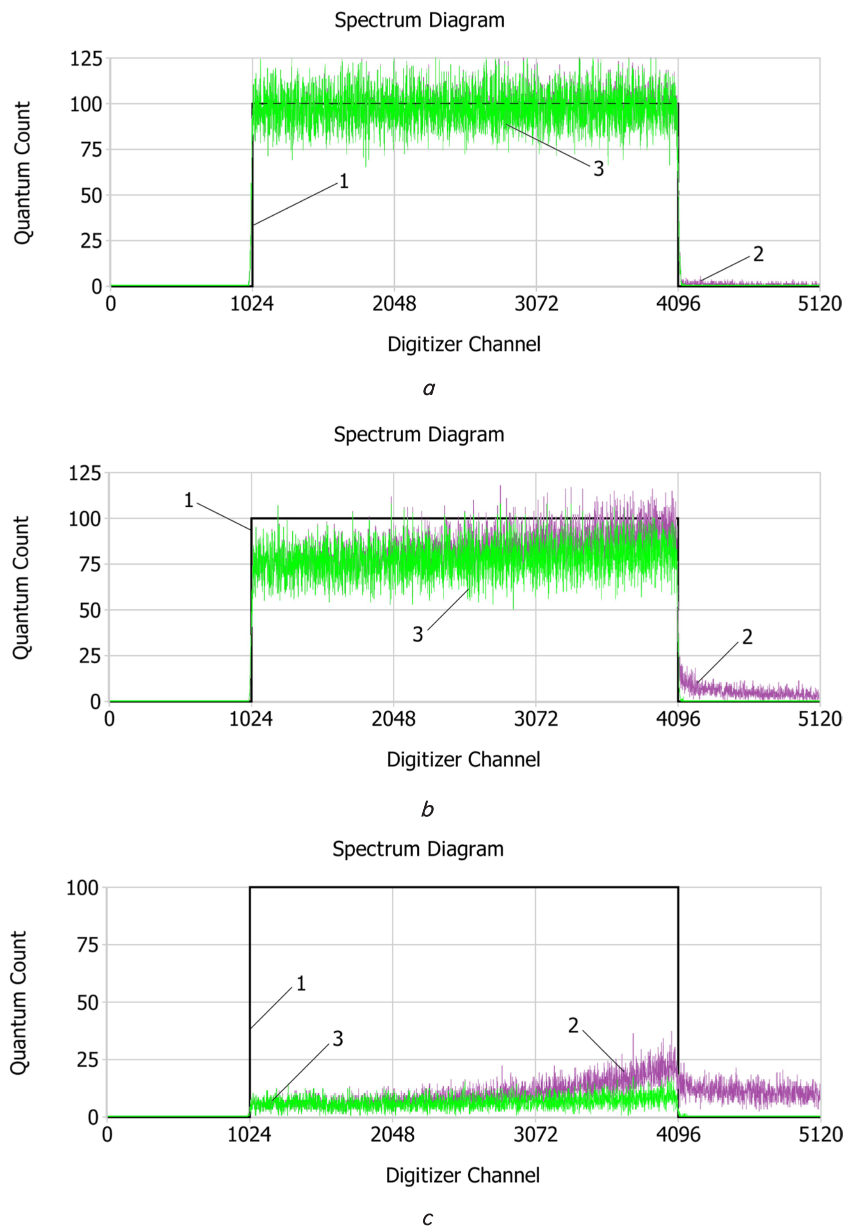


Fig. 6. Amplitude distribution diagrams: 1 — simulated; 2 — recognized; 3 — verified pulses obtained by the Maximum method at load levels (pulses per second): *a* — 10^5; *b* — 10^6; *c* — 10^7

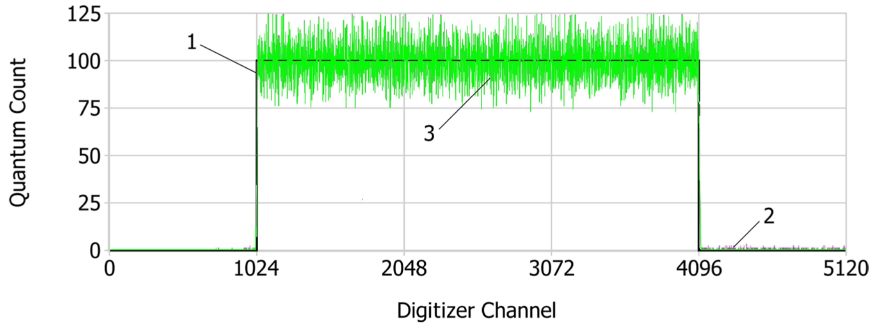
In general, the distribution diagrams of the pulse amplitudes recognized by the Fitting method (Fig. 7, *a*–*c*) are visually similar to those obtained as a result of the analysis by the new method. However, the verified accuracy of recognition is somewhat lower and was: 99.47 % in the case of a load level of 10^5 pulses per second, 94.79 % at a load level of 10^6, 53.82 % at a load level of 10^7.

The final comparative performance characteristics of the new and existing considered recognition methods are given in Tables 1–3. These tables contain the program-calculated and output results of the described analysis methods on the input data, simulated under three scenarios with different loading levels and duration of the experiments, which were described in detail in chapter 4.

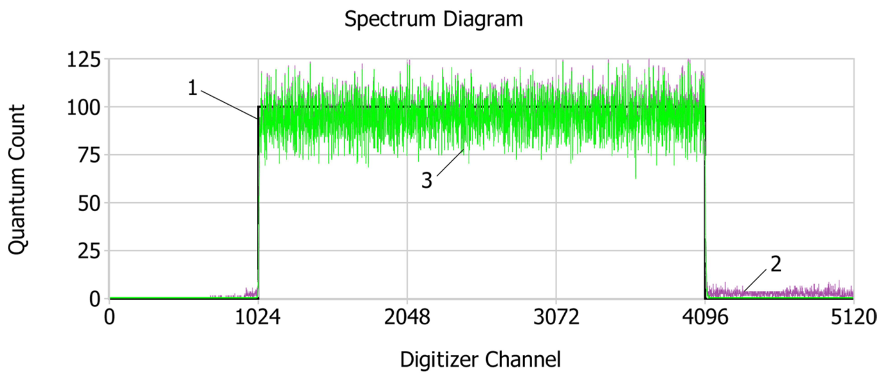
Table 1

Results of the analysis methods on the input data simulated with a load level of 10^5 pulses per second

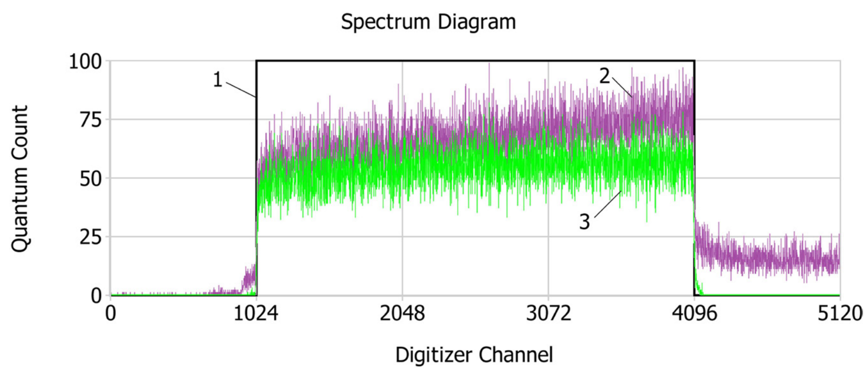
Method ID	Number of simulated pulses	Number of pulses recognized in total	Number of verified pulses	Method operating time (sec)	Verified method accuracy (%)
Maximum	307200	302519	299739	1.616	97.57
Fitting	307200	306460	305583	8.193	99.47
Our method	307200	306546	305739	41.629	99.52



a



b



c

Fig. 7. Amplitude distribution diagrams: 1 – simulated; 2 – recognized; 3 – verified pulses obtained by the Fitting method at load levels (pulses per second): *a* – 10^5; *b* – 10^6; *c* – 10^7

Table 2

Results of the analysis methods on the input data simulated with a load level of 10^6 pulses per second

Method ID	Number of simulated pulses	Number of pulses recognized in total	Number of verified pulses	Method operating time (sec)	Verified method accuracy (%)
Maximum	307200	263108	238907	0.245	77.77
Fitting	307200	299983	291192	3.246	94.79
Our method	307200	300408	291693	6.965	94.95

Table 3

Results of the analysis methods on the input data simulated with a load level of 10^7 pulses per second

Method ID	Number of simulated pulses	Number of pulses recognized in total	Number of verified pulses	Method operating time (sec)	Verified method accuracy (%)
Maximum	307200	61284	18381	0.061	5.98
Fitting	307200	244711	165350	6.791	53.82
Our method	307200	248356	166701	1.124	54.26

It is worth noting that the devised Tracking method demonstrated better verified recognition accuracy compared to the considered existing methods in all three scenarios that were investigated. In addition, Tables 1–3 also contain information about data processing time. The duration of the analysis by the simple method of Maximum was, as expected, the shortest. The Fitting method showed a better speed of operation than the devised method on data simulated with a loading parameter of 10^5 and 10^6 pulses per second (Tables 1, 2); however, the new method worked faster at a load of 10^7 pulses per second (Table 3).

6. Discussion of results based on investigating the methods of analysis of spectrometric signals

In contrast to previous studies [2, 5], in which the total number of recognized pulses and the resulting spectrum were mainly determined, in this work the concept of verified accuracy of the analysis method was introduced. This was done by computer simulation of the reference input data (Fig. 2) and implementation of the verification algorithm, which made it possible to objectively assess the accuracy of recognition by new and existing methods.

As could be seen from the diagrams in Fig. 5–7, *a*, and the comparative results of the analysis methods given in Table 1, the devised and considered known methods showed a fairly high recognition accuracy at a load level of 10^5 pulses per second. This is explained by the relatively small probability of pulse overlap in such a scenario.

A feature of the newly devised method, in comparison with some well-known ones [2], is the technique for filtering the signal from noise without significant modification of the usable component. Also, unlike existing simple methods of analysis (Maximum, Sum), the devised Tracking method, like the Fitting method, is capable of better analyzing impulses with relatively frequent overlap. After all, in the process of determining the level of their amplitudes, this processing method implements a mechanism for taking into account the influence of the signal value of the falling component of the previous pulses. Therefore, on the diagrams of the distribution of amplitudes at the load level of 10^6, shown in Fig. 5, *b*, the new method demonstrates less deviation from the linear distribution and a fairly high accuracy of recognition (Table 2). In the case of a load of 10^7, the verified accuracy of the Tracking method decreased to 54.26 % (Table 3). This result is explained by the fact that at a given loading level, the probability of superposition of pulses increases greatly, and often two or more of them superimpose in such a way as to form a single pulse with a large amplitude. In conclusion, the comparative results given in Tables 1–3, show that the devised method has made it possible to achieve the main goal of the study and improve the accuracy of recognition in the considered scenarios.

But it is worth noting that the pulses of the signal simulated in this study have a very sharp rising edge and fall, so the filtering mechanism by the devised method of the upper part of the pulses may not have a significant effect. In some cases, hardware filters are additionally used in the electronic equipment to record the signal from the detector, which form pulses with a slower rise and fall at the output (an example of such a signal was shown in Fig. 3). It could be assumed that a new analysis method could show better results on such digital data.

One limitation of this study is that the analysis was performed exclusively on a simulated digital signal, with a linear idealized pulse amplitude distribution and simulated load levels of 10^5, 10^6 and 10^7 pulses per second. Therefore, a further direction of research is to check the operation of the described methods on real experimental data obtained from the digitizer, as well as on simulated idealized data close to real spectra.

The result of our research is the possibility for improving accuracy in recognizing the parameters of spectrometric signals in the process of computer processing of digitized experimental data. This effect could be achieved by using the devised Tracking method.

In addition, the software developed in the research process could be used for further testing and improvement of computer methods for processing and analyzing digitized spectrometric data.

7. Conclusions

1. With the help of computer simulation, based on specialized template files, reference digital input data with a known distribution of pulse amplitudes (linear) were compiled for further analysis by various methods. A verification algorithm was also developed, which made it possible to compare arrays of generated and recognized pulses after analysis and determine what percentage of pulses detected by a certain method coincide with those that were simulated. This made it possible to more objectively assess the accuracy of recognition by each of the methods (especially under conditions of frequent superposition of pulses), unlike previous studies, in which only the total number of recognized pulses was determined.

2. An approach to filtering the spectrometric signal from electrical noise with minimal changes in the usable component has been implemented, based on the use of Fast Discrete Fourier Transform, and an algorithm for recognizing pulses with amplitude correction during their superimposition was developed. After testing on simulated data, the new algorithm showed fairly high recognition accuracy at load levels of 10^5 and 10^6 pulses per second (about 99.5 % and 94.9 %, respectively) under the considered conditions of the experimental simulation. With a load level of 10^7 and a high probability of pulse superposition, the verified recognition accuracy was about 54.2 %.

3. In order to evaluate the effectiveness of the developed algorithms and comparative analysis, two existing methods of spectral analysis were programmatically implemented: the Maximum method and the Fitting method. Their data processing speed was calculated, recognition accuracy was verified, and comparative characteristics of all considered methods were obtained. In conclusion, the new method showed 17–48 **%** better recognition accuracy than the simple Maximum method at high load levels. In addition, in the investigated scenarios, the devised method has made it possible to slightly improve the recognition accuracy compared to the existing Fitting method (by 0.05–0.4 **%**).

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available, either in numerical or graphical form, in the main text of the manuscript.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Knoll, G. F. (2010). Radiation Detection and Measurement. John Wiley & Sons, 864.
2. Wolszczak, W., Dorenbos, P. (2018). Time-resolved gamma spectroscopy of single events. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 886, 30–35. https://doi.org/10.1016/j.nima.2017.12.080
3. Pollastrone, F., Cardarilli, G. C., Riva, M., Costa Pereira, R., Fernandes, A., Cruz, N. et al. (2019). A clustering algorithm for scintillator signals applied to neutron and gamma patterns identification. Fusion Engineering and Design, 146, 2110–2114. https://doi.org/10.1016/j.fusengdes.2019.03.117
4. Lopatin, M., Moskovitch, N., Trigano, T., Sepulcre, Y. (2012). Pileup attenuation for spectroscopic signals using a sparse reconstruction. 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel, 1–5. https://doi.org/10.1109/eeei.2012.6377045
5. Khilkevitch, E. M., Shevelev, A. E., Chugunov, I. N., Iliasova, M. V., Doinikov, D. N., Gin, D. B. et al. (2020). Advanced algorithms for signal processing scintillation gamma ray detectors at high counting rates. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 977, 164309. https://doi.org/10.1016/j.nima.2020.164309
6. Shevelev, A. E., Khilkevitch, E. M., Lashkul, S. I., Rozhdestvensky, V. V., Altukhov, A. B., Chugunov, I. N. et al. (2016). High performance gamma-ray spectrometer for runaway electron studies on the FT-2 tokamak. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 830, 102–108. https://doi.org/10.1016/j.nima.2016.05.075
7. Yoon, S., Lee, C., Won, B.-H., Hong, S.-B., Seo, H., Kim, H.-D. (2022). Fast neutron-gamma discrimination in organic scintillators via convolution neural network. Journal of the Korean Physical Society, 80 (5), 427–433. https://doi.org/10.1007/s40042-022-00398-x
8. Yoon, S., Lee, C., Seo, H., Kim, H.-D. (2023). Improved fast neutron detection using CNN-based pulse shape discrimination. Nuclear Engineering and Technology, 55 (11), 3925–3934. https://doi.org/10.1016/j.net.2023.07.007
9. Law, A. M., Kelton, W. D. (2000). Simulation Modeling and Analysis. McGraw-Hill, 760.
10. Reva, S., Tsybliyev, D. (2023). Mathematical models and algorithms of computer modeling of spectrometric signals. Bulletin of V. N. Karazin Kharkiv National University, Series «Mathematical Modeling. Information Technology. Automated Control Systems», 58, 64–74. Available at: https://periodicals.karazin.ua/mia/article/view/23502
11. Meyers, S. (2014). Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. O'Reilly Media, Inc. Addison-Wesley, 320.
12. Drozdek, A. (2001). Data Structures and Algorithms in C++. Brooks/Cole, 650.
13. QT Framework Official Website. Available at: https://www.qt.io/product/framework
14. Wahab, M. F., Gritti, F., O'Haver, T. C. (2021). Discrete Fourier transform techniques for noise reduction and digital enhancement of analytical signals. TrAC Trends in Analytical Chemistry, 143, 116354. https://doi.org/10.1016/j.trac.2021.116354
15. Bracewell, R. N. (2000). The Fourier Transform and its Applications. New Delhi: McGraw-Hill Education, 640.
16. Heideman, M. T., Johnson, D. H., Burrus, C. S. (1985). Gauss and the history of the fast Fourier transform. Archive for History of Exact Sciences, 34 (3), 265–277. https://doi.org/10.1007/bf00348431
17. Press, W., Teukolsky, S., Vetterling, W., Flannery, B. (2007). Numerical recipes. The Art of Scientific Computing. Cambridge University Press, 1256. Available at: https://www.cambridge.org/us/universitypress/subjects/mathematics/numerical-recipes/numerical-recipes-art-scientific-computing-3rd-edition
18. OMG Unified Modeling Language Specification. Version 2.5 (2015). Available at: https://www.omg.org/spec/UML/2.5/PDF
19. Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Addison Wesley Professional, 736.