USING OF THE MATRIX REPRESENTATION OF ALGORIGTHMS OF THE FAST FOURIER AND HARTLEY TRANSFORMS IN THE PROBLEMS OF “SLIDING” SPECTRUM ANALYSIS

T.V. ZINCHENKO

Matrix formulae for calculating of the fast Fourier and Hartley transforms for a sequence of radar signal readings are obtained. The formulae obtained allow us to implement the “sliding” (or “galloping”) spectrum analysis with rational using of information about the spectrum of a previous step for each new position of a time window. The matrix formulae of the Hartley transforms with an arbitrary base are obtained. These formulae are a generalization of the well-known Hartley formula for splitting of the initial sequence into the two sequences, namely, with the even and odd numbers.

Keywords: radar signals, digital processing, fast Fourier transform, fast Hartley transform, spectrum analysis, “sliding” signal processing.

Introduction. To solve the problems of spectral analysis, in digital signal processing devices it can be used the principle of the “sliding” (“running” or “galloping”) window. This principle is implemented on the basis of the discrete Fourier transform (DFT) or the discrete Hartley transform (DHT). To reduce the time of these transforms, it is used a variety of algorithms of the fast transforms, namely, the algorithms of the fast Fourier transforms (FFT) and the algorithms of the fast Hartley transforms (FHT).

Methods. The DFT of a finite sequence $\{x(n)\}$, $0 \le n \le N$-1, is defined by the formula [1]:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j \cdot 2\pi nk/N} = \sum_{n=0}^{N-1} x(n) \cdot W^{nk} \qquad (1)$$

where $W = e^{-j \cdot 2\pi/N}$, k = 0, 1, 2 ..., (N-1). In formula (1) the sequence W^{nk} is a periodic sequence with the period N, i.e. $W^{(n+mN)(k+lN)} = W^{nk}$, $W_N^2 = W_{N/2}$, $W_N^{k+N/2} = -W_N^k$, m, l = 0, ±l, ... The DFT $X(k)$ can be read as follows

$$X(k) = \begin{cases} X_1(k) + W_N^k \cdot X_2(k), & 0 \le k \le \dfrac{N}{2} - 1 \\ X_1\left(k - \dfrac{N}{2}\right) - W_N^{k-\frac{N}{2}} \cdot X_2\left(k - \dfrac{N}{2}\right), & 0 \le k \le \dfrac{N}{2} - 1 \end{cases}. \qquad (2)$$

where $X_1(k)$ and $X_2(k)$ are the N/2-point DFT of the sequence of even elements $x_1(n) = x(2n)$ and the N/2-point DFT of the sequence of odd elements $x_2(n) = x(2n+1)$, respectively.

The process of calculation of the FFT with the decomposition of the processed signal sequence into the even and odd parts (2) is called the decimation-in-time.

If it is used an another way, namely, the decimation-in-frequency than the input sequence is decomposed into the two sequential parts

$$X(k) = \sum_{n=0}^{N/2-1} x(n) \cdot W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) \cdot W_N^{nk} = \sum_{n=0}^{N/2-1} x(n) \cdot W_N^{nk} + \sum_{n=N/2}^{N-1} x\left(n + \frac{N}{2}\right) \cdot W_N^{\left(n+\frac{N}{2}\right)k} =$$

$$= \sum_{n=0}^{N/2-1}\left[x_1(n)+e^{-j\pi k}\cdot x_2(n)\right]\cdot W_N^n \quad . \tag{3}$$

The even and odd samples of the DFT of the input sequence $X(2k)$ and $X(2k+1)$ are the N/2-point DFTs of the sequences $f(n)$ and $g(n)$

$f(n)=x_1(n)+x_2(n)$, $g(n)=[x_1(n)+x_2(n)]\cdot W_N^n$, $n=0,1,\ldots,(N/2-1)$,

which can be read by the formulae

$$X(2k)=\sum_{n=0}^{N/2-1}\left[x_1(n)-x_2(n)\right]\cdot W_N^{2n} = \sum_{n=0}^{N/2-1} f(n)\cdot W_{N/2}^n,$$

$$X(2k+1)=\sum_{n=0}^{N/2-1}\left[x_1(n)-x_2(n)\right]\cdot W_N^{n(2k+1)} = \sum_{n=0}^{N/2-1} g(n)\cdot W_{N/2}^n. \tag{4}$$

Theory/calculation. If a sequence of N samples of the input signal is rows arranged in a matrix of dimension $[L\times M]$, where L and M are the number of rows and columns, respectively than the number n of the current sample can be represented as $n = Ml + m$, where l is the current row number and m is the current column number. Moreover, in this case, the current number of an element of the DFT output signal matrix can be read as $k = Lr + s$. In this case, the formula of the DFT takes the following form [1]:

$$X(k)=X(s,r)=\sum_{m=0}^{M-1}\sum_{l=0}^{L-1}x(l,m)\cdot W^{(M+m)(L+s)}=\sum_{m=0}^{M-1}\left(W^L\right)^{mr}\cdot\left[W^{m}\cdot\sum_{l=0}^{L-1}x(l,m)\cdot\left(W^M\right)^{ls}\right] \tag{5}$$

According to the representation (5), the sequence of operations is as follows:

1) calculate the L-point DFTs with the transform kernel W^M for all columns (computation of the inner sum);

2) multiply each element of the result of step 1 by the twiddle factor W^{ms};

3) calculate the M-point DFTs for all rows, resulting from performing of step 1 and step 2, with the transform kernel W^L (computation of the inner sum).

Changing of the order of summation in (5) on the reverse order leads to the following expression

$$X(s,r)=\sum_{l=0}^{L-1}\left(W^M\right)^{ls}\cdot\sum_{m=0}^{M-1}\left[x(l,m)\cdot W^{m}\right]\cdot\left(W^L\right)^{mr} \quad . \tag{6}$$

Taking into account (6), the following procedure of calculation of the $X(k)$ can be used

1) multiply the signal samples $x(1, r)$ by the twiddle factor W^{ms};

2) calculate the M-point DFTs for all rows with the transform kernel W^{ms} (computation of the inner sum);

3) calculate the L-point DFTs for all columns with the transform kernel W^M.

Differences in the calculations according to (5) and (6) correspond to the difference between the FFT (the base 2 is used) with the decimation-in-time and decimation-in-frequency, respectively. If the decimation-in-time is used than the multiplication to the twiddle factors is followed by the main operations of the DFT. In the case of the decimation in frequency, the multiplication follows after the operations of the DFT.

The sequence of operations in (5) can be represented in a matrix form, and such representation is suitable for the practical implementation of the algorithm. For a fixed value s_0 (the line number), it can write down

$$X(s_0,r)=\left[X^T(l,m)\cdot W_{S_0M}(l)\right]^T\cdot W_{S_0L}(m,r)=R^T(m,1)\cdot W_{S_0L}(m,r)=G_{S_0}(1,r), \tag{7}$$

where $R(m,1)=X^T(l,m)\cdot W_{S_0M}(l)$, $R(m,1)$ is the column-matrix of dimension $[M\times 1]$,

$X(s_0,r)$ is the row-matrix of the DFT output signals of dimension $[1\times L]$,

$W_{S_0L}(m,r)=\left\|\left(W^{Lr+S_0}\right)^m\right\|$ is the matrix of the twiddle factors of dimension $[M\times L]$, $r=\overline{0,(L-1)};\ m=\overline{0,(M-1)}$,

$X(l,m)=\|x(l,m)\|$ is the matrix of values of the input samples of dimension $[L\times M]$, $l=\overline{0,(L-1)}$,

$W_{S_0M}(l)=\left\|\left(W^{S_0M}\right)^l\right\|$ is the column-matrix of the twiddle factors of dimension $[L\times 1]$, $l=\overline{0,(L-1)}$.

The sequence of operations in (6) is represented in a matrix form for the fixed value s_0 by the formula

$$X(s_0,r)=W^T_{S_0M}(l)\cdot\left[X(l,m)\cdot W_{S_0L}(m,r)\right]=W^T_{S_0M}(l)\cdot P(l,r)=G_{S_0}(1,r), \qquad (8)$$

where $P(l,r)=X(l,m)\cdot W_{S_0L}(m,r)$, $P(l,r)$ is the matrix of dimension $[L\times L]$.
According to the expressions (7) and (8), the sequences of operations are presented by Fig. 1 and Fig. 2, respectively.

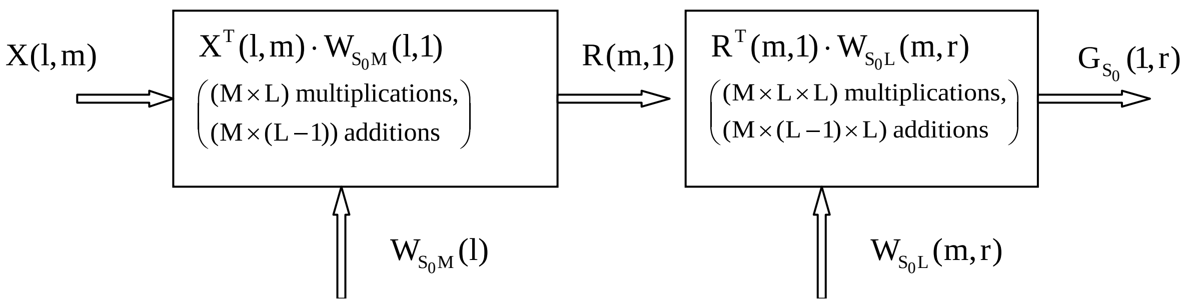

Fig.1. The scheme of calculation of the row-matrix of DFT output signals of dimension $[1\times L]$ with the decimation-in-time.

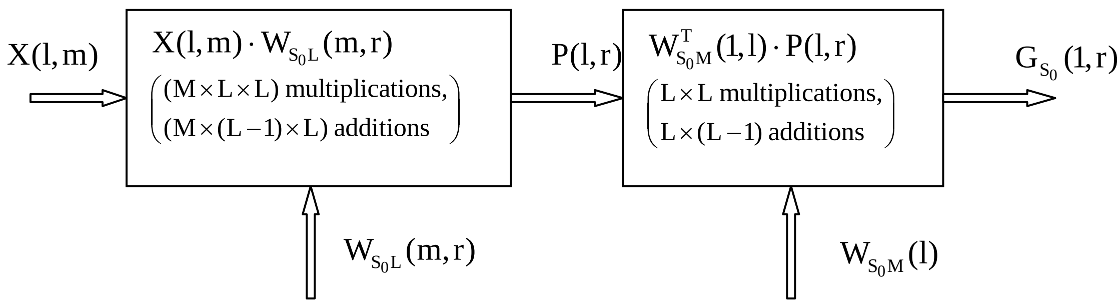

Fig.2. The scheme of calculation of the row-matrix of DFT output signals of dimension $[1\times L]$ with the decimation-in-frequency.

An FFT representation as a product of the matrices (7) or (8) makes it relatively easy to count the number of elementary operations in the implementation of the FFT, and, what is the most importantly, to identify those operations (that digital information), the result of which can be used at each step of the “sliding” FFT.

The “sliding” or “galloping” *N*-point DFT of the input discrete signal $x(n)$ is given by

$$F(n,k)=\frac{1}{N}\sum_{m=0}^{N-1} s(n-m)\cdot \mathrm{e}\left(-xi\frac{2\pi m}{N}p\right), \qquad (9)$$

where m, k = 0, 1, …, (N-1).

For a fixed number n, the function $F(n, k)$ is being the DFT in the variable m of the interval $x(n\text{-}m)$ of the data stream $x(n)$. If n = 0, 1, 2, ... than the FFT is a “sliding” FFT. If the step Δn is greater than 2, the FFT is a “galloping” FFT. The function $F(n, k)$ it is convenient to call the current Fourier spectrum.

Let us schematically consider in a matrix form an offset of the time window containing N samples, representing the samples by their numbers in the incoming time sequence. For example, if $N = 16 = 4\times4$ and $\Delta n = 1$ than

1st window				*2nd window*				*3rd window*			
1	2	3	4	2	3	4	5	3	4	5	6
5	6	7	8	6	7	8	9	7	8	9	10
9	10	11	12	10	11	12	13	11	12	13	14
13	14	15	16	14	15	16	17	15	16	17	18

Fig. 3. Representation in a matrix form of the numbers of input time sequence of the “sliding” FFT.

The columns 2-4 of the first window are the columns 1-3 of the second window, the columns 2-4 of the second window are the columns 1-3 of the third window etc. This means that for each step (for the next time window position) the values (M-Δn) of the L-point column FFTs (M is the number of columns, L is the number of rows) can be used with the previous step. As one can see from Fig. 3, for the “sliding” FFT the data processing of the function $x(n)$ should be carried out on columns according to the formulae (5) or (7) (see Fig.1).

Another possible scheme can be realized by using the values of the row FFTs. For instance, if $N = 16 = 4\times4$ and $\Delta n = 4$ than

1 window				*2 window*				*3 window*			
1	2	3	4	5	6	7	8	9	10	11	12
5	6	7	8	9	10	11	12	13	14	15	16
9	10	11	12	13	14	15	16	17	18	19	20
13	14	15	16	17	18	19	20	21	22	23	24

Fig. 4. Representation in a matrix form of the numbers of input time sequence of the “galloping” FFT with the step Δn=4.

When the window of information processing is moved than the rows 2-4 of the first window are the rows 1-3 of the second window, the rows 2-4 of the second window are the rows 1-3 of the third window, etc. This means that for each step (for the next position of the time window), the values (L-1) of the M-point row DFTs can be used with the previous step. According to Fig. 4, for the “galloping” FFT with the step Δn = 4, the data processing of the function $x(n)$ should be carried out in rows by using the formulae (6) or (8) (see Fig.2).

Thus, fixing the length N for a certain time window and a step of displacement Δn, as well as comparing the features of the algorithms (5) and (6) (with the decimation-in-time or decimation-on-frequency, respectively), one can choose (relatively simple) the most high-speed schematic decision of the FFT calculation in a “sliding” window. When the sliding procedures are implemented, the selection decision, which algorithm must be given priority (with processing by rows or columns) should be made depending on the step of sliding. For example, if the step of sliding $\Delta n = 1$ (see Fig.3) than it should be selected the algorithms (5) and (7) – the processing by columns. If the step of sliding $\Delta n = 4$ (see Fig.4) than it should be selected the algorithms (6) and (8) – the processing by rows.

For the case $N = 16$, let us arrange the input values in a matrix of dimension 4×4. In this case $L = M = 4; l = 0,1,2,3; m = 0,1,2,3; r = 0,1,2,3; s = 0,1,2,3$. The matrices are

$$W_{S_0L}(m,r) = \begin{pmatrix} W^0 & W^{(4+S_0)\cdot 0} & W^{(8+S_0)\cdot 0} & W^{(12+S_0)\cdot 0} \\ W^{S_0\cdot 1} & W^{(4+S_0)\cdot 1} & W^{(8+S_0)\cdot 1} & W^{(12+S_0)\cdot 1} \\ W^{S_0\cdot 2} & W^{(4+S_0)\cdot 2} & W^{(8+S_0)\cdot 2} & W^{(12+S_0)\cdot 2} \\ W^{S_0\cdot 3} & W^{(4+S_0)\cdot 3} & W^{(8+S_0)\cdot 3} & W^{(12+S_0)\cdot 3} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ W^{S_0} & -j\cdot W^{S_0} & -W^{S_0} & j\cdot W^{S_0} \\ W^{2S_0} & -W^{2S_0} & W^{2S_0} & -W^{2S_0} \\ W^{3S_0} & j\cdot W^{3S_0} & -W^{3S_0} & -j\cdot W^{3S_0} \end{pmatrix}$$

For the various values of $s_0 = 0,1,2,3$, the corresponding matrices are

$$W_{0L} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix}, \quad W_{1L} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ e^{-j\frac{\pi}{8}} & -j\cdot e^{-j\frac{\pi}{8}} & -e^{-j\frac{\pi}{8}} & j\cdot e^{-j\frac{\pi}{8}} \\ e^{-j\frac{2\pi}{8}} & -e^{-j\frac{2\pi}{8}} & e^{-j\frac{2\pi}{8}} & -e^{-j\frac{2\pi}{8}} \\ e^{-j\frac{3\pi}{8}} & j\cdot e^{-j\frac{3\pi}{8}} & -e^{-j\frac{3\pi}{8}} & -j\cdot e^{-j\frac{3\pi}{8}} \end{pmatrix},$$

$$W_{2L} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ e^{-j\frac{\pi}{4}} & -j\cdot e^{-j\frac{\pi}{4}} & -e^{-j\frac{\pi}{4}} & j\cdot e^{-j\frac{\pi}{4}} \\ -j & j & -j & j \\ e^{-j\frac{3\pi}{4}} & j\cdot e^{-j\frac{3\pi}{4}} & -e^{-j\frac{3\pi}{4}} & -j\cdot e^{-j\frac{3\pi}{4}} \end{pmatrix}, \quad W_{3L} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ e^{-j\frac{3\pi}{8}} & -j\cdot e^{-j\frac{3\pi}{8}} & -e^{-j\frac{3\pi}{8}} & j\cdot e^{-j\frac{3\pi}{8}} \\ e^{-j\frac{3\pi}{4}} & -e^{-j\frac{3\pi}{4}} & e^{-j\frac{3\pi}{4}} & -e^{-j\frac{3\pi}{4}} \\ -e^{-j\frac{\pi}{8}} & -je^{-j\frac{\pi}{8}} & e^{-j\frac{\pi}{8}} & j\cdot e^{-j\frac{\pi}{8}} \end{pmatrix}.$$

The column-matrix of the twiddle factors $W_{S_0M}(l) = \left\| \left(W^{S_0M}\right)^l \right\|$ for $N = 16$ is

$$\left\| \left(W^{S_0M}\right)^l \right\| = \left(W^{S_0M\cdot 0} \quad W^{S_0M\cdot 1} \quad W^{S_0M\cdot 2} \quad W^{S_0M\cdot 3}\right)^T = \left(1 \quad W^{4S_0} \quad W^{8S_0} \quad W^{12S_0}\right)^T.$$

For the values $s_0 = 0,1,2,3$, the corresponding column-matrices are

$$W_{0M} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad M_{1M} = \begin{pmatrix} 1 \\ -j \\ -1 \\ j \end{pmatrix}, \quad W_{2M} = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}, \quad W_{3M} = \begin{pmatrix} 1 \\ j \\ -1 \\ -j \end{pmatrix}.$$

The algorithms (5) and (6) suggest a different number of performed operations. The algorithm (5) suggests the *N* of the *L*-point transforms (sums), the *N* of the *M*-point transforms, and the *N* multiplications by the twiddle factors. In the algorithm (6), it is implemented the *N* of the *L*-point transforms (sums), the *N*×*L* of the M-point transforms, and the *N*×*L* multiplications by the twiddle factors. More operations of the algorithm (6) can be explained by the fact that the inner sum has more factors than it has in the algorithm (5), and it is a function of three variable parameters instead of two.

Results. Thus, using the uniform matrix approach to the FFT algorithm described in [1], it is easy to synthesize algorithms allowing us to implement the operation of the “running” (or “galloping”) spectral analysis with the optimal using of information about the spectrum of the previous position of the time window. Moreover, in the case of the matrix approach, in the algorithm it is contained the main grouping fragments allowing to efficiently parallelize of the computational process.

References

1. *Rabiner L.R., Gold B.* Theory and Application of digital signal processing. New Jersey: Prentice Hall, Inc, 1975.

Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов.– М.: Мир, 1978. – 848 с.

2. *Зинченко Т.В.* Матричное представление алгоритмов быстрых преобразований Фурье и Хартли в задачах «скользящего» спектрального анализа // Кибернетика и высокие технологии XXI века: XIII международная научно-техническая конференция, 14-15 мая 2012 г.- Россия, г. Воронеж, 2012. – С. 85-94.

3. *Jansen T., B. Rymon-Lipinski B., Hanssen N., Keeve E.* Fourier volume rendering on the GPU using a split-stream-FFT / Proceedings of the Vision, Modeling, and Visualization Conference. 2004.

4 . *Sumanaweera T., Liu D.* Medical image reconstruction with the FFT / GPU Gems 2, Pharr M., Ed. Addison-Wesley, 2005.

Zinchenko Tatyana Volodymyrivna, Ph.D. (1988), associate professor (2002), leading researcher, Institute “Quantum-Radar” (till 2006), lecturer in the National University of Food Technologies (Kyiv). Fields of research: synthesis of adaptive antenna arrays, processing of radar data, optimization methods in economics.

Along with the Fourier transform, it is possible to use the Hartley transform in the digital signal processing tasks [2]. The Hartley transform is a real-valued function of a real argument. The DHT for the samples of the function $x(i)$, $i = 0, 1, \ldots, (N\text{-}1)$ can be defined as follows

$$H(k) = \frac{1}{N}\sum_{i=1}^{N-1} x(i)\cdot c\ (2\pi k\ /sN),\ k = 0, 1, \ldots, (N\text{-}1), \qquad (10)$$

where casβ = cosβ + sinβ.

The Hartley transform is symmetric; the inverse transform is as follows

$$x(i) = \sum_{k=1}^{N-1} x(i)\cdot c\ (2\pi ks/\ N) \qquad (11)$$

The Hartley transform can be represented as the sum of the even and odd components $H(k) = E(k) + O(k)$, wherein

$$E(k) = (H(k) + H(N\text{-}k))/2,\ O(k) = (H(k\text{-}\ H(N\text{-}k))/2.$$

As it was shown by R.N. Bracewell in [2], there is a relationship between the Fourier and Hartley transforms. If the Fourier transform is represented as $F(k) = R(k) + i\,X(k)$ than

$$R(k) = (H(k) + H(N-k))/2,$$
$$X(k) = -(H(k) - H(N-k))/2\ . \qquad (12)$$

The spectral density Z^2 is

$$Z^2(k) = R^2(k) + X^2(k) = [H^2(k) + H^2(N - k)]/2. \qquad (13)$$

To estimate the phase of the Fourier transform, one can compute the value

$$\arg F(k) = \mathrm{arctg}\left[\frac{H(N-k)-H(k)}{H(N-k)+H(k)}\right] = \mathrm{arctg}\left[-\frac{O(k)}{E(k)}\right]. \qquad (14)$$

In work [2], it is described the principles of construction of the FHT algorithms. If the input N-element sequence $x(i)$ is decomposed into the two sequences with the even and odd numbers (decimation-in-time) — $\{x(0), 0, x(2), 0, x(4), ...\}$, $\{x(1), 0, x(3), 0, x(5), ...\}$, and for them is calculated the Hartley transforms $H_1(k)$ and $H_2(k)$ with the period N than the Hartley transform of the input sequence can be calculated by the formula

$$H(k) = H_1(k) + H_2(k)\cdot\cos(2\pi k/N) + H_2(N - k)\cdot\sin(2\pi k/N). \qquad (15)$$

The values $H_1(k)$ and $H_2(k)$ can be obtained by the repeated decomposition until, for example, of the four-element sequences consisting of the two-element segments; the conversion of the two-element sequence includes two operations of addition and no any operations of multiplication.

It is suggested that if $N = 2^P$ than the computation of the FHT requires performing of the number of operations proportional to $N\times P$ [2]. As it is in the case of the FFT, in order to increase the speed of computation of the FHT algorithm, it should be used the dimension $N = 4^P$ with the basic four-element DHT [2].

In a number of scientific publications, it has suggested that the FHT is faster than the FFT. Such conclusion could be obtained if we compare the transforms of the real-valued data sequence by the FHT and FFT, but such approach is incorrect. The FFT is an ideal tool for the data processing in the complex form. The FHT can be used for processing of arrays of the real numbers (for instance, for imaging).

From the viewpoint of uniform matrix approach, the formula for the FHT, which is similar to the formula (5) for the FFT, is given by

$$H(k) = H(s,r) = \sum_{m=0}^{M-1}\sum_{l=0}^{L-1} x(l,m)\cdot \mathrm{cas}\left[\frac{2\pi}{N}(Ml+m)\cdot(Lr+s)\right] =$$
$$= \sum_{m=0}^{M-1}\sum_{l=0}^{L-1} x(l,m)\cdot \mathrm{cas}\left[\frac{2\pi}{N}(Mls + ms + Lmr)\right]. \qquad (16)$$

In the equations (5) and (16), it is expressed the main difference between the Fourier and Hartley transforms in terms of the synthesis of fast algorithms, namely, the separability of the Fourier transform kernel that provides a variety of realization according to the characteristics and features of tasks and the inseparability of the Hartley transform kernel that substantially limits of the possible realizations. The conversion of the cas-function leads the equation (16) to the following form

$$H(s,r) = \sum_{l=0}^{L-1}\left[\mathrm{cas}\left(\frac{2\pi}{N}Mls\right)\sum_{m=0}^{M-1} x(l,m)\cdot\cos\left(\frac{2\pi}{N}m(Lr+s)\right)\right] +$$
$$+\sum_{l=0}^{L-1}\left[\mathrm{cas}\left(-\frac{2\pi}{N}Mls\right)\sum_{m=0}^{M-1} x(l,m)\cdot\sin\left(\frac{2\pi}{N}m(Lr+s)\right)\right]. \qquad (17)$$

Let us show that the Hartley formula (15) is a particular case of (17) with $M = 2$. In this case, the input data matrix has two columns ($m = 0, 1$) and $L = N/2$ rows. If one sets the source array elements by their numbers than the elements in the matrix are arranged as follows

0	1
2	3
4	5
……	
N-2	*N*-1

The first and second columns consist of the elements with the even and odd numbers, respectively. Transform the expression (17), taking into account the notations adopted in the transition to the matrix representation of the input sequence (see the formula (5))

$$\begin{aligned}
H(s,r) &= \sum_{l=0}^{L-1}\left[\cos\left(\frac{2\pi}{N}Ml\right)\cdot\left(x(l,0)+x(l,1)\cos\left(\frac{2\pi}{N}(Ln+s)\right)\right)\right]sr + \\
&+ \sum_{l=0}^{L-1}\left[\cos\left(-\frac{2\pi}{N}Ml\right)\cdot x(l,1)\sin\left(\frac{2\pi}{N}(Ln+s)\right)\right]r = \\
&= \sum_{l=0}^{L-1}\cos\left(\frac{2\pi}{N}2ln\right)\cdot x(l,0)+\cos\left(\frac{2\pi}{N}(Ln+s)\right)\sum_{l=0}^{L-1}\cos\left(\frac{2\pi}{N}2ln\right)\cdot x(l,1)+ \\
&+ \sin\left(\frac{2\pi}{N}(Ln+s)\right)\sum_{l=0}^{L-1}\cos\left(-\frac{2\pi}{N}2ln\right)\cdot x(l,1)= \\
&= H_1(k)+\cos\left(\frac{2\pi}{N}k\right)\cdot H_2(k)+\sin\left(\frac{2\pi}{N}k\right)\cdot H_2(N-k)=H(k), \qquad (18)
\end{aligned}$$

where $H_1(k)$ and $H_2(k)$ are the Hartley transforms with the period N of the sequences consisting of the elements of the columns {$x(0)$, 0, $x(2)$, 0, $x(4)$, 0, ...} and {$x(1)$, 0, $x(3)$, 0, $x(5)$, 0, ...}, respectively.

Thus, it was proven that the Hartley formula (15) is a particular case of the matrix algorithm (17).

By analogy with the FFT on the base 4, which in some cases it is considered preferable from the standpoint of the number of operations [1], the practical interest has the algorithm of the Hartley transform on the base 4. Represent the elements of the source array by their numbers in a matrix with such four columns

0	1	2	3
4	5	6	7
8	9	10	11
	……		
N-4	*N*-3	*N*-2	*N*-1

For the case $M = 4$, from the formula (17) it can be obtained

$$\begin{aligned}
H(k) &= H_0(k)+H_1(k)\cdot\cos\left(\frac{2\pi}{N}k\right)+H_1(N-k)\cdot\sin\left(\frac{2\pi}{N}k\right)+H_2(k)\cdot\cos\left(\frac{2\pi}{N}2k\right)+ \\
&+ H_2(N-k)\cdot\sin\left(\frac{2\pi}{N}2k\right)+H_3(k)\cdot\cos\left(\frac{2\pi}{N}3k\right)+H_3(N-k)\cdot\sin\left(\frac{2\pi}{N}3k\right). \qquad (19)
\end{aligned}$$

In (19) $H_0(k)$, $H_1(k)$, $H_2(k)$, and $H_3(k)$ are the Hartley transforms of the sequences consisting of the elements of the columns {$x(0)$,0,0,0,$x(4)$,0,0,0,$x(8)$,0...}, {$x(1)$,0,0,0,$x(5)$,0,0,0,$x(9)$,0...}, $x(2)$,0,0,0,$x(6)$,0,0,0,$x(10)$,0...}, and {$x(3)$,0,0,0,$x(7)$,0,0,0,$x(11)$,0...}, respectively. Combining of the Hartley transforms of the separable columns is realized by means of the two multiplications and one addition by each column. Such procedure significantly increases the total number of multiplications. In general, in the case of the matrix representation of the input sequence of dimension $N = L \times M$, the number of operations of multiplication Km and addition Ka are

$$Km = (N\text{-l}) \times [2(M-1) + MK_{Lm}],\ Ka = (N\text{-l}) \times [2(M-1) + MK_{La}] + M-1,$$

where K_{Lm} and K_{La} are the number of multiplications and additions, respectively in calculating of the Hartley transform of a single column (with L elements). The number of operations is counted taking into account the assumption that in the calculation of $H(0)$ (when $k = 0$), the operations of multiplication are not used. For $N = 16$ and $M = L = 4$ we have $Km = 90$ and $Ka = 125$. For the algorithm (18), this is suitably 82 and 105. Although the algorithm (19) requires more operations, it allows substantial opportunity of the parallezation of the computational process, which makes it attractive for the implementation.

Differences between the transforms are evident in the implementation of the "running" procedures by using the matrix algorithm.

The "running" FHT is given by

$$H(n,k) = \frac{1}{N}\sum_{m=0}^{N-1} s(n-m)\cdot \mathrm{cas}\left(\frac{2\pi m}{N}\right), \quad \text{where } m, k = 0, 1, \ldots, (N\text{-}1). \qquad (20)$$

In work [3], it is described a matrix algorithm of realization of the "running" FHT (the base 2 is used) with the decimation-in-time. This algorithm is based on the principle of repeating decomposition of the input sequence into the two parts.

As it is followed from the relation (16), unlike to the case of the FFT, the calculation of the FHT permits only the column processing of samples presented in a matrix form. The calculation of the number of operations showed that the practical utility has only the algorithms of computation of the Hartley transform on the base 2 or 4. As it is in the case of the column FFT data processing presented in the form of a matrix with 2 or 4 columns (see Fig. 3), to calculate the current spectrums in the "sliding" window, it should be used the methods of using of the spectrum values from the previous step.

Results. Thus, using the uniform matrix approach to the FFT algorithm described in [1], it is easy to synthesize algorithms allowing us to implement the operation of the "running" (or "galloping") spectral analysis with the optimal using of information about the spectrum of the previous position of the time window. Moreover, in the case of the matrix approach, in the algorithm it is contained the main grouping fragments allowing to efficiently parallelize of the computational process.

The general formula for calculating of the Hartley transform with the arbitrary base for the data sequence presented in a matrix form is obtained. It is shown that the well-known Hartley formula using splitting of the input sequence into the two parts, namely, with the even and odd numbers is a particular case of the formula obtained. It is proven that the algorithm with the base no more than four is a practically suitable.

References

3. *Rabiner L.R., Gold B.* Theory and Application of digital signal processing. New Jersey: Prentice Hall, Inc, 1975.

Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов.– М.: Мир, 1978. – 848 с.

4. *Bracewell R.N.* The Hartley transform. New York: Oxford University Press, 1986.

Брейсуэлл Р. Н. Преобразование Хартли. –М.: Мир, 1990. – 175 с.

5. *Зинченко Т.В.* Матричное представление алгоритмов быстрых преобразований Фурье и Хартли в задачах «скользящего» спектрального анализа // Кибернетика и высокие технологии XXI века: XIII международная научно-техническая конференция, 14-15 мая 2012 г.- Россия, г. Воронеж, 2012. – С. 85-94.

4. *Jansen T., B. Rymon-Lipinski B., Hanssen N., Keeve E.* Fourier volume rendering on the

GPU using a split-stream-FFT / Proceedings of the Vision, Modeling, and Visualization Conference. 2004.
5. *Sumanaweera T., Liu D.* Medical image reconstruction with the FFT / GPU Gems 2, Pharr M., Ed. Addison-Wesley, 2005.

Zinchenko Tatyana Volodymyrivna, Ph.D. (1988), associate professor (2002), leading researcher, Institute “Quantum-Radar” (till 2006), lecturer in the National University of Food Technologies (Kyiv). Fields of research: synthesis of adaptive antenna arrays, processing of radar data, optimization methods in economics.