

REVISED FAST FOURIER TRANSFORM

The problem of realisation of the Discrete Fourier Transform in on-line is analysed because of non-efficient consuming a time for a new recalculation of spectrum samples if one discrete-time signal sample or even some small portion of samples in period are replaced by new sample or by new samples, respectively. Using Fast Fourier Transform (FFT) procedure it is assumed that some signal samples in the respective period available for processing digitally are updated by a sensor in real time. It is urgent for every new sample that emerges to have a new spectrum. The ordinary recalculation of spectrum samples even with highly efficient Cooley-Tukey FFT algorithm is not suitable due to speedy varying in time real process to be observed. The idea is that FFT procedure should not be recalculated with every new sample, it is needed just to modify it when the new sample emerges and replaces the old one. We retrieve the recursive formulas for FFT algorithms that refer to the spectrum samples modification. In a case of appearing one new sample, the recursive algorithm calculates a new spectrum samples by simple addition of a residual between an old and new samples, multiplied on respective row of Fourier ‘code’ matrix, to a vector of old spectrum samples. An example of 8-point FFT is presented.

Keywords: digital signal processing, discrete Fourier transform, fast Fourier transform.

NOMENCLATURE

DSP is a digital signal processing;
 DFT is a discrete Fourier transform;
 FFT fast Fourier transform;
 $\{x(n)\}$ is a discrete-time real valued signal or sequence of real numbers;
 L is a length of a real valued signal;
 n is a number of signal sample;
 $X(\omega)$ is a Fourier transform of a discrete-time signal;
 j is the imaginary unit;
 ω is an angular frequency;
 $X(k)$ is a spectrum sample;
 k is a number of spectrum sample;
 N is a general number of spectrum or signal samples;
 W_N^{kn} is a periodic function with only N different values;
 $x_{old}(l)$ is an old l -th sample of real valued signal;
 $x_{new}(l)$ is a new l -th sample of real valued signal;
 $X_{old}(k)$ is an old k -th sample of spectrum;
 $X_{new}(k)$ is a new k -th sample of spectrum.

INTRODUCTION

The continuous-time Fourier series are broadly used in theory as well as in practice where functions are continuous. DFT can be treated as its discrete-time counterpart. DFT has also been implemented digitally in the area of filter synthesis, image processing, various audio and video signal developments, and many types of spectrum analyzers that compute sampled power spectra and frequency response functions. The properties of ordinary DFT are accurately described. On the other hand, it is known that ordinary DFT involves a lot of redundant calculations. Therefore, usually, ordinary DFT algorithm is replaced by highly efficient computer procedures, known as FFT algorithms. Also there is a considerable amount of literature available on DFT and FFT, mentioned here just a few (e.g. [1–9]) that are coupled with DSP. However, some problems, encountered with FFT applications to measured samples of signals, are not generally understood [3], especially, analysing varying in the time processes, e.g. if some portion of samples or even

one sample in the given period is replaced by new samples or one sample, respectively, and for each such real time case we have to obtain a new spectrum. Therefore, it is needed to modify DFT and FFT in order to recalculate only some products of the Fourier ‘code matrix’ with the respective samples replaced. The next section introduces the statement of the problem to be solved. In Section 3 we worked out the recursive equations that allow to modify the ordinary FFT procedure. Examples are presented in Section 4. Section 5 contains conclusions.

1 PROBLEM STATEMENT

Consider a discrete-time finite duration real-valued signal $\{x(n)\}$ of length L (i.e., $\{x(n)\} = 0$ for $n < 0$ and $n \geq L$) that has the Fourier transform

$$X(\omega) = \sum_{n=0}^{L-1} x(n)e^{-j\omega n} \quad \forall \omega \in \overline{0, 2\pi}, \quad (1)$$

where the upper and lower indices in the summation reflect the fact that $\{x(n)\} = 0$ outside the range of $0 \leq n \leq L-1$. Here j is the imaginary unit. When we sample $\{X(\omega)\}$ at equally frequencies $\omega_k = 2\pi k / N \quad \forall k \in \overline{0, N-1}$, with $N \geq L$, the resultant samples are as follows:

$$X\{(k)\} = X\left(\frac{2\pi k}{N}\right) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn / N}. \quad (2)$$

For convenience the upper index in the sum has been increased from $L-1$ to $N-1$ since $\{x(n)\} = 0$ for $n \geq L$. Here N is the general number of samples of the basic real valued signal $\{x(n)\}$ under consideration. The relation in eq. (2) is called DFT of $\{x(n)\}$ and is used for transforming the sample sequence $\{x(n)\}$ into a sequence of frequency samples $X\{(k)\}$ of length N . Rewriting eq. (2) in the form

$$X\{(k)\} = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad \forall k \in \overline{0, N-1}, \quad (3)$$

$$W_N^{kn} = e^{-j2\pi kn/N}, \quad (4)$$

one can realize that the integer product kn repeats for different combinations of k and n , and that W_N^{kn} is a periodic function with only N different values. Therefore, various fast and efficient DFT algorithms without redundant calculations were worked out [1–9]. Frequently, the FFT is computed by decimating the sample sequence $\{x(n)\}$ into sub-sequences until 2-point DFT's remain.

Assume that the frequencies analysis of real-time streaming sensor data $\{x(n)\}$ is needed. The aim of the paper is to work out an recursive approach that would update the spectrum samples given in eq. (3) as fast as possible with a sensor's sample that emerges, without anew recalculation of spectrum samples by FFT algorithm.

2 REVIEW OF THE LITERATURE

Recursive methods are important for the property that observations of time-varying phenomenon by their use can be processed by computer in real time. Hence, they may be applied in on-line monitoring and analysis of generally time-varying processes, and also combined with on-line control strategies to produce adaptive control algorithms. Some of them were used to the parametric identification of nonlinear Wiener systems [10]. Recursive procedures can be effective by processing various characteristics of stationary as well as nonstationary random processes and systems [11].

It is well-known that the most important area of DSP includes searching for different characteristics of signals and systems in frequency domain. Here more popular are DFT and various FFT procedures. However, as it is emphasized in [3], that the use of the DFT with the digitized signals are not generally understood. For example, ordinary DFT requires that in the given period N all samples ought to be fixed. Indeed, the author of this paper has not found any published work that addresses the recursive calculation of the DFT or FFT on an N -point complex valued function, when the samples of the varying in the time signal are observed by sensor. On the other hand, it is known that the processes, functioning in real life are dynamic and time-varying.

Therefore, it is important to work out procedures based on ordinary DFT that allow us to find spectrum samples, when some samples of discrete-time signal in the respective period, available for processing digitally, are updated by a sensor in real time.

3 MATERIALS AND METHODS

It is not efficient to recalculate the basic spectrum samples anew, if only one signal sample or even a small portion of new samples emerges continuously, especially, when speed is a main issue. Then, the computation time can become prohibitive, in spite of the fact that FFT requires only $N \log_2 N$ complex multiplications and complex additions to compute each of the N spectral samples. In such a case, it is important to work out an approach for modifying FFT in order to decrease the calculation time significantly. Let us retrieve now recursive formulas for recalculating the basic spectrum samples $X(k) \forall k \in \overline{0, N-1}$

partly, when a new sample $x_{new}(l)$ appears in the given N samples of a signal $x(n) \forall n \in \overline{0, N-1}$ while the respective old one vanishes. For real valued $x(n) \forall n \in \overline{0, N-1}$ eq.(3) can be rewritten as

$$X_{old}(k) = X_{old} \left(\frac{2\pi k}{N} \right) = \sum_{n=0}^{N-1} x_{old}(n) W_N^{kn}, \quad (5)$$

or

$$X_{new}(k) = X_{new} \left(\frac{2\pi k}{N} \right) = \sum_{n=0}^{N-1} x_{new}(n) W_N^{kn}, \quad (6)$$

if only the old and new samples of the sequence $\{x(n)\}$ are used, respectively. Here $x_{old}(l), x_{new}(l) \forall l \in \overline{0, N-1}$ are l -th old and new samples, $X_{old}(k), X_{new}(k) \forall k \in \overline{0, N-1}$ are values of the old and new samples in frequency domain, correspondingly.

Suppose now that in eq. (6) all the new samples are equivalent to the old ones, except, the sample $x_{new}(l)$. Then, we can rewrite eq. (6) as follows

$$X_{new}(k) = \sum_{n=0}^{l-1} x_{old}(n) W_N^{kn} + x_{new}(l) W_N^{kl} + \sum_{n=l+1}^{N-1} x_{old}(n) W_N^{kn} \quad \forall k \in \overline{0, N-1}.$$

Subtracting the values $X_{old}(k)$ from $X_{new}(k) \forall k \in \overline{0, N-1}$ we obtain the relationship of the form

$$\begin{bmatrix} X_{new}(0) - X_{old}(0) \\ X_{new}(1) - X_{old}(1) \\ \vdots \\ X_{new}(N-2) - X_{old}(N-2) \\ X_{new}(N-1) - X_{old}(N-1) \end{bmatrix} = [x_{new}(l) - x_{old}(l)] \begin{bmatrix} W_N^{l0} \\ W_N^{l1} \\ \vdots \\ W_N^{l(N-2)} \\ W_N^{l(N-1)} \end{bmatrix}.$$

It can also be rewritten in the recursive form

$$\begin{bmatrix} X_{new}(0) \\ X_{new}(1) \\ \vdots \\ X_{new}(N-2) \\ X_{new}(N-1) \end{bmatrix} = \begin{bmatrix} X_{old}(0) \\ X_{old}(1) \\ \vdots \\ X_{old}(N-2) \\ X_{old}(N-1) \end{bmatrix} + [x_{new}(l) - x_{old}(l)] \begin{bmatrix} W_N^{l0} \\ W_N^{l1} \\ \vdots \\ W_N^{l(N-2)} \\ W_N^{l(N-1)} \end{bmatrix}, \quad (7)$$

assuming that a new sample $x_{new}(l)$ emerges and replaces the old one $x_{old}(l)$.

Suppose now that in eq. (6) all the new samples are equivalent to the old ones, except a some portion of new samples $x_{new}(l), x_{new}(l+1), \dots, x_{new}(l+p-2), x_{new}(l+p-1)$, that appears in the given N samples of signal $\{x(n)\}$, while the respective portion of the old samples vanishes. In such a case, the final expression can be rewritten recursively

$$X_{new}(k) = X_{old}(k) + \sum_{m=l}^{l+p-1} [x_{new}(m) - x_{old}(m)] W_N^{km} \quad \forall k \in \overline{0, N-1},$$

or in extended form

$$\begin{bmatrix} X_{new}(0) \\ X_{new}(1) \\ \vdots \\ X_{new}(N-2) \\ X_{new}(N-1) \end{bmatrix} = \begin{bmatrix} X_{old}(0) \\ X_{old}(1) \\ \vdots \\ X_{old}(N-2) \\ X_{old}(N-1) \end{bmatrix} + [x_{new}(l) - x_{old}(l)] \begin{bmatrix} W_N^{l0} \\ W_N^{l1} \\ \vdots \\ W_N^{l(N-2)} \\ W_N^{l(N-1)} \end{bmatrix} + \dots + [x_{new}(v) - x_{old}(v)] \begin{bmatrix} W_N^{v0} \\ W_N^{v1} \\ \vdots \\ W_N^{v(N-2)} \\ W_N^{v(N-1)} \end{bmatrix} \quad (8)$$

Here $v = l + p - 1$.

4 EXPERIMENTS

Let us describe now two experiments that are carried out while analysis of recursive FFT. In both experiments we consider the discrete-time periodic signal

$$x(n) = \{\dots, 24, 8, 12, 16, 20, 6, 10, 14, \dots\}. \quad (9)$$

By inspection, the period $N = 8$. DFT is computed according to

$$X(k) = \sum_{n=0}^7 x(n) \exp(-j \frac{2\pi}{8} nk), \forall k \in \overline{0,7}. \quad (10)$$

or in a form

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & a(1-j) & -j & b(1+j) & -1 & b(1-j) & j & a(1+j) \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & b(1+j) & j & a(1-j) & -1 & a(1+j) & -j & b(1-j) \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & b(1-j) & -j & a(1+j) & -1 & a(1-j) & j & b(1+j) \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ 1 & a(1+j) & j & b(1-j) & -1 & b(1+j) & -j & a(1-j) \end{bmatrix} \begin{bmatrix} 24 \\ 8 \\ 12 \\ 16 \\ 20 \\ 6 \\ 10 \\ 14 \end{bmatrix}, \quad (11)$$

using Fourier ‘code’ matrix. Here $a=0.7071$ and $b = -a$. Afterwards, the spectrum samples $X(0), X(1), \dots, X(7)$ are determined by FFT using *Matlab* as follows: *fft*([24, 8, 12, 16, 20, 6, 10, 14], 8). The spectrum samples are: $X(0)=110$, $X(1)=4 - 4.83j$, $X(2)=22 + 16j$, $X(3)=4 - 0.83j$, $X(4)=22$, $X(5)=4 + 0.83j$, $X(6)=22 - 16j$, $X(7)=4 + 4.83j$.

In the first experiment we change in (11), firstly, fourth sample, and, secondly, the sixth one. Suppose that a new fourth sample ‘25’ comes in (9), and the old one ‘20’ goes out. Then, the system of linear complex valued equations (10) is of the form

$$\begin{bmatrix} X_{new}(0) \\ X_{new}(1) \\ X_{new}(2) \\ X_{new}(3) \\ X_{new}(4) \\ X_{new}(5) \\ X_{new}(6) \\ X_{new}(7) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & a(1-j) & -j & b(1+j) & -1 & b(1-j) & j & a(1+j) \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & b(1+j) & j & a(1-j) & -1 & a(1+j) & -j & b(1-j) \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & b(1-j) & -j & a(1+j) & -1 & a(1-j) & j & b(1+j) \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ 1 & a(1+j) & j & b(1-j) & -1 & b(1+j) & -j & a(1-j) \end{bmatrix} \begin{bmatrix} 24 \\ 8 \\ 12 \\ 16 \\ 25 \\ 6 \\ 10 \\ 14 \end{bmatrix}.$$

Afterwards, the spectrum samples $X_{new}(0), X_{new}(1), \dots, X_{new}(7)$ are recalculated anew by *Matlab*: *fft*([24, 8, 12, 16, 25, 6, 10, 14], 8). Their values now are: $X_{new}(0)=115$, $X_{new}(1)=-1 - 4.83j$, $X_{new}(2)=27 + 16j$, $X_{new}(3)=-1 - 0.83j$, $X_{new}(4)=27$, $X_{new}(5)=-1 + 0.83j$, $X_{new}(6)=27 - 16j$, $X_{new}(7)=-1 + 4.83j$. The same spectrum samples $X_{new}(0), X_{new}(1), \dots, X_{new}(7)$ can be determined recursively by eq. (7) as follows

$$\begin{bmatrix} X_{new}(0) \\ X_{new}(1) \\ X_{new}(2) \\ X_{new}(3) \\ X_{new}(4) \\ X_{new}(5) \\ X_{new}(6) \\ X_{new}(7) \end{bmatrix} = \begin{bmatrix} 110 + (5) \cdot 1 \\ 4 - 4.83j + (5) \cdot (-1) \\ 22 + 16j + (5) \cdot 1 \\ 4 - 0.83j + (5) \cdot (-1) \\ 22 + (5) \cdot 1 \\ 4 + 0.83j + (5) \cdot (-1) \\ 22 - 16j + (5) \cdot 1 \\ 4 + 4.83j + (5) \cdot (-1) \end{bmatrix} = \begin{bmatrix} 115 \\ -1 - 4.83j \\ 27 + 16j \\ -1 - 0.83j \\ 27 \\ -1 + 0.83j \\ 27 - 16j \\ -1 + 4.83j \end{bmatrix}.$$

It is obvious, that in both cases we obtain the same set values of $X(0), X(1), \dots, X(7)$, respectively. Suppose now that just after finishing recursive calculations the new sixth sample with values ‘5’ come in the set of samples (9) and the respective old one with values ‘10’ go out. Then, the previous values of $X_{new}(0), X_{new}(1), \dots, X_{new}(7)$ can be treated now as old ones, i.e. $X_{old}(0) = X_{new}(0), X_{old}(1) = X_{new}(1), \dots, X_{old}(6) = X_{new}(6), X_{old}(7) = X_{new}(7)$, respectively. The current values of $X(0), X(1), \dots, X(7)$ are treated as $X_{new}(0), X_{new}(1), \dots, X_{new}(7)$. They can be obtained by the recursive formula

$$\begin{bmatrix} X_{new}(0) \\ X_{new}(1) \\ X_{new}(2) \\ X_{new}(3) \\ X_{new}(4) \\ X_{new}(5) \\ X_{new}(6) \\ X_{new}(7) \end{bmatrix} = \begin{bmatrix} 115 + (-5) \cdot 1 \\ -1 - 4.83j + (-5) \cdot j \\ 27 + 16j + (-5) \cdot (-1) \\ -1 - 0.83j + (-5) \cdot (-j) \\ 27 + (-5) \cdot 1 \\ -1 + 0.83j + (-5) \cdot j \\ 27 - 16j + (-5) \cdot (-1) \\ -1 + 4.83j + (-5) \cdot (-j) \end{bmatrix} = \begin{bmatrix} 110 \\ -1 - 9.83j \\ 32 + 16j \\ -1 + 4.17j \\ 22 \\ -1 - 4.17j \\ 32 - 16j \\ -1 + 9.83j \end{bmatrix}.$$

Let us check now the previous recursive FFT by the ordinary one using *Matlab*: *fft*([24, 8, 12, 16, 20, 6, 5, 14], 8). The spectrum samples are:

$$\begin{aligned} X(0) &= 110, X(1) = -1 - 9.83j, X(2) = 32 + 16j, \\ X(3) &= -1 + 4.17j, X(4) = 22, X(5) = -1 - 4.17j, \\ X(6) &= 32 - 16j, X(7) = -1 + 9.83j. \end{aligned} \quad (12)$$

It is assumed in the second experiment that both samples in (9) emerge at the same time. They appear just after calculations performed with initial set of samples (9). The new fourth and sixth samples with values '25' and '5' come in (9) and the respective old ones with values '20' and '10' go out. Then, eq. (11) obtains the form

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & a(1-j) & -j & b(1+j) & -1 & b(1-j) & j & a(1+j) \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & b(1+j) & j & a(1-j) & -1 & a(1+j) & -j & b(1-j) \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & b(1-j) & -j & a(1+j) & -1 & a(1-j) & j & b(1+j) \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ 1 & a(1+j) & j & b(1-j) & -1 & b(1+j) & -j & a(1-j) \end{bmatrix} \begin{bmatrix} 24 \\ 8 \\ 12 \\ 16 \\ 25 \\ 6 \\ 5 \\ 14 \end{bmatrix}$$

Accordinging eq. (8) recursive 8-point FFT expression is

$$\begin{bmatrix} X_{new}(0) \\ X_{new}(1) \\ X_{new}(2) \\ X_{new}(3) \\ X_{new}(4) \\ X_{new}(5) \\ X_{new}(6) \\ X_{new}(7) \end{bmatrix} = \begin{bmatrix} 110 + (5) \cdot 1 + (-5) \cdot 1 \\ 4 - 4.83j + (5) \cdot (-1) + (-5)j \\ 22 + 16j + (5) \cdot 1 + (-5) \cdot (-1) \\ 4 - 0.83j + (5) \cdot (-1) + (-5) \cdot (-j) \\ 22 + (5) \cdot 1 + (-5) \cdot 1 \\ 4 + 0.83j + (5) \cdot (-1) + (-5) \cdot j \\ 22 - 16j + (5) \cdot 1 + (-5) \cdot (-1) \\ 4 + 4.83j + (5) \cdot (-1) + (-5) \cdot (-j) \end{bmatrix} = \begin{bmatrix} 110 \\ -1 - 9.83j \\ 32 + 16j \\ -1 + 4.17j \\ 22 \\ -1 - 4.17j \\ 32 - 16j \\ -1 + 9.83j \end{bmatrix} \quad (13)$$

Thus, the results (13) given by the recursive FFT of the form (7) are coincident with the results (12) that are obtained by ordinary FFT using Matlab standard function *fft*. Let us analyse now the recursive FFT using the special form

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} G(0) + W_8^0 H(0) \\ G(1) + W_8^1 H(1) \\ G(2) + W_8^2 H(2) \\ G(3) + W_8^3 H(3) \\ G(0) - W_8^0 H(0) \\ G(1) - W_8^1 H(1) \\ G(2) - W_8^2 H(2) \\ G(3) - W_8^3 H(3) \end{bmatrix}$$

of ordinary FFT for the given period N . Here

$$G(k) = \sum_{n=0}^3 x(2n)W_4^{kn}, \quad H(k) = \sum_{n=0}^3 x(2n+1)W_4^{kn}, \\ \forall k = 0, 1, 2, 3.$$

Suppose that a new fourth sample '25' comes in the set, given by the initial eq. (9), and the old sample '20' goes out. At the same moment, every value of $G(k) \forall k \in \overline{0,3}$ changes, while corresponding value of $H(k) \forall k \in \overline{0,3}$ remains the same. Then, one can obtain

$$\begin{bmatrix} X_{new}(0) \\ X_{new}(1) \\ X_{new}(2) \\ X_{new}(3) \\ X_{new}(4) \\ X_{new}(5) \\ X_{new}(6) \\ X_{new}(7) \end{bmatrix} = \begin{bmatrix} G(0) + (5) \cdot 1 + W_8^0 H(0) \\ G(1) + (5) \cdot (-1) + W_8^1 H(1) \\ G(2) + (5) \cdot 1 + W_8^2 H(2) \\ G(3) + (5) \cdot (-1) + W_8^3 H(3) \\ G(0) + (5) \cdot 1 - W_8^0 H(0) \\ G(1) + (5) \cdot (-1) - W_8^1 H(1) \\ G(2) + (5) \cdot 1 - W_8^2 H(2) \\ G(3) + (5) \cdot (-1) - W_8^3 H(3) \end{bmatrix} = \begin{bmatrix} X_{old}(0) \\ X_{old}(1) \\ X_{old}(2) \\ X_{old}(3) \\ X_{old}(4) \\ X_{old}(5) \\ X_{old}(6) \\ X_{old}(7) \end{bmatrix} + 5 \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

This relationship is coincident with the formula (7), too.

5 RESULTS

It follows from the 8-point DFT example with real-valued samples set (9), that recursive FFT on an 8-point complex valued function, requires 8 operations (here operation is defined as one complex valued multiplication plus an one complex valued addition [3]) if one new sample comes in, and twice more both operations, if a portion of two samples emerges.

6 DISCUSSION

Thus, N -point DFT requires N complex-valued operations if one new sample emerges in a period, and $2N$ operations if a portion of two samples goes in, respectively. On the other hand, direct computation of the DFT on an N -point complex valued function requires M operations to recalculate all N spectrum samples. Calculations increase even four times if twice increases the general number of samples to be processed. The Cooley-Tukey algorithm, that belongs to the class of FFT algorithms, takes approximately $N \log_2 N$ operations [12, 13]. It is known [4], that for small values of N (say, 32 to 128) the FFT is important. For large values of N (1024 and above), the FFT is incredibly more efficient. For example, FFT is even hundred times faster than DFT, when $N = 1024$. Nevertheless, recalculation of spectrum samples by FFT is not only nonrational but also nonefficient if one sample emerges replacing an old one. In such a case, recursive calculation by (7) is much more effective. In order to change old spectrum samples the recursive FFT requires only 1024 operations on an 1024-point complex valued function, while the ordinary FFT requires 10 times more by anew their recalculation.

CONCLUSIONS

For discrete-time signals the DFT coefficient values have been proposed to recursively determine if one new signal sample or new portion of samples emerge in the given period of a realization replacing the old sample or old portion of samples, respectively. The number of operations for their speedy calculating is essentially reduced by the original recursive expression in comparison with the ordinary DFT or FFT equations (2), (3), respectively, used in the case of fixed values of samples $x(n) \forall n \in \overline{0, N-1}$ in a fixed period N . An example, presented here, has shown us the efficiency of the recursive approach, too. Therefore, it is not rational to recalculate frequency samples by ordinary DFT or even FFT algorithms if only one sample in the given period or if some small portion of samples is replaced by new sample or some new samples, respectively. The recursive FFT approach could be effective, especially, in real-time applications when speed of calculations is the main issue.

ACKNOWLEDGEMENTS

The work is supported by the state budget scientific research project of Institute of Mathematics and Informatics of Vilnius University «Analysis, recognition, optimization and control of nonlinear systems and signals of complex structure» (registration number 2AP1.46) and by an European Commission Funded Project «Comenius MP» running from November 2012 to October 2014 (№ 526315-LLP-2012-CY-COMENIUS-CMP).

REFERENCES

1. Deziel J. P. Applied introduction to digital signal processing / J. P. Deziel. – New Jersey : Prentice Hall, Inc., 2000. – 388 p.
2. Gonzalez R. C. Digitale Image Processing / R. C. Gonzalez. – New Jersey : Prentice Hall, Inc., 2007. – 976 p.
3. Oppenheim A. V. Discrete-time signal processing / A. V. Oppenheim, R. W. Shafer. – New Jersey : Prentice Hall, Inc., 2009. – 1120 p.
4. Proakis J. G. Digital signal processing. Principles, algorithms, and applications / J. G. Proakis, D. G. Manolakis. – New Jersey : Prentice Hall, Inc., 2006. – 1004 p.
5. Proakis J. G. Student manual for digital signal processing with Matlab / J. G. Proakis, V.K. Ingle. – New Jersey : Prentice Hall, Inc., 2006. – 264 p.

Пупейкіс Р.

Канд. техн. наук, доцент, старший сотрудник отдела процессов распознавания, Вильнюсский университет, Вильнюс, Литва

ИСПРАВЛЕННОЕ БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ

Проблема реализации дискретного преобразования Фурье в режиме он-лайн анализируется из-за неэффективных затрат времени для нового пересчета отсчетов спектра, если отсчет одного сигнала с дискретным временем или даже небольшая часть отсчетов в периоде заменены на новый отсчет или новые отсчеты, соответственно. Использование процедуры быстрого преобразования Фурье (БПФ) предполагает, что некоторые отсчеты сигнала в соответствующем периоде, доступные для цифровой обработки, обновляются с помощью датчика в режиме реального времени. Это актуально для каждого нового отсчета, который приводит к получению нового спектра. Обычный пересчет отсчетов спектра даже с высокоэффективным алгоритмом БПФ Кули-Тьюки не подходит из-за быстро изменяющегося во времени наблюдаемого реального процесса. Идея заключается в том, что процедура БПФ не должна пересчитываться с каждым новым отсчетом, нужно просто модифицировать его, когда новый отсчет появляется и заменяет старый. Получены рекуррентные формулы для алгоритмов БПФ, которые относятся к модификации отсчетов спектра. В случае возникновения одного нового отсчета, рекурсивный алгоритм вычисляет новые отсчеты спектра простым добавлением к вектору старых отсчетов спектра разности между старыми и новыми отсчетами, умноженной на соответствующий ряд матрицы «кода» Фурье. Приведен пример 8-точечного БПФ.

Ключевые слова: цифровая обработка сигналов, дискретное преобразование Фурье, быстрое преобразование Фурье.

Пупейкіс Р.

Канд. техн. наук, доцент, старший співробітник відділу процесів розпізнавання, Вільнюський університет, Вільнюс, Литва

ВИПРАВЛЕНЕ ШВИДКЕ ПЕРЕТВОРЕННЯ ФУР'Є

Проблема реалізації дискретного перетворення Фур'є в режимі он-лайн аналізується через неефективні витрати часу для нового перерахунку відліків спектру, якщо відлік одного сигналу з дискретним часом або навіть невелика частина відліків в періоді замінені на новий відлік або нові відліки, відповідно. Використання процедури швидкого перетворення Фур'є (ШПФ) припускає, що деякі відліки сигналу у відповідному періоді, доступні для цифрової обробки, оновлюються за допомогою датчика в режимі реального часу. Це актуально для кожного нового відліку, який призводить до отримання нового спектра. Звичайний перерахунок відліків спектру навіть з високоэффективним алгоритмом ШПФ Кулі-Тьюки не підходить через швидко мінливого у часі спостережуваного реального процесу. Ідея полягає в тому, що процедура ШПФ не повинна перераховуватися з кожним новим відліком, потрібно просто модифікувати його, коли новий відлік з'являється і замінює старий. Отримано рекуррентні формули для алгоритмів ШПФ, які відносяться до модифікації відліків спектру. У разі виникнення одного нового відліку, рекурсивний алгоритм обчислює нові відліки спектру простим додаванням до вектора старих відліків спектру різниці між старими і новими відліками, помноженої на відповідний ряд матриці «коду» Фур'є. Наведено приклад 8-точкового ШПФ.

Ключові слова: цифрова обробка сигналів, дискретне перетворення Фур'є, швидке перетворення Фур'є.

REFERENCES

1. Deziel J. P. Applied introduction to digital signal processing. New Jersey: Prentice Hall, Inc., 2000, 388 p.
2. Gonzalez R.C. Digitale Image Processing. New Jersey, Prentice Hall, Inc., 2007, 976 p.
3. Oppenheim A. V., Shafer R. W. Discrete-time signal processing. New Jersey, Prentice Hall, Inc., 2009, 1120 p.
4. Proakis J. G., Manolakis D. G. Digital signal processing. Principles, algorithms, and applications. New Jersey, Prentice Hall, Inc., 2006, 1004 p.
5. Proakis J. G., Ingle V. K. Student manual for digital signal processing with Matlab. New Jersey : Prentice Hall, Inc., 2006, 264 p.
6. Lyons R. G. Understanding digital signal processing. New Jersey, Prentice Hall, Inc., 2010, 984 p.
7. Richardson M. H. Fundamentals of the discrete Fourier transform, *Sound & Vibration Magazine*, 1978, March, pp. 1–8.

6. Lyons R. G. Understanding digital signal processing / R. G. Lyons. – New Jersey : Prentice Hall, Inc., 2010. – 984 p.
7. Richardson M. H. Fundamentals of the discrete Fourier transform / M. H. Richardson // *Sound & Vibration Magazine*. – 1978. – March. – P. 1–8.
8. Smith S. W. Digital signal processing. A practical guide for engineers and scientists / S.W. Smith. – San Diego : California Technical Publishing, 2003. – 640 p.
9. Pupeikis R. Vaizdu apdorojimo Matlab'o terpėje pagrindai / R. Pupeikis. – Vilnius : Technika, 2008. – 107 p.
10. Pupeikis R. Self-tuning minimum variance control of linear systems followed by saturation nonlinearities in a noisy frame / R. Pupeikis / *International Journal of Robust and Nonlinear Control*. – 2014. – Vol. 24, № 2. – P. 313–325. DOI: 10.1002/rnc.2888
11. Казлаускас К. Цифровые системы обработки данных. Монография./ К. Казлаускас, Р. Пупейкіс. – Вильнюс : Мокслас, 1991. – 220 с.
12. Cooley J. W. An algorithm for the machine calculation of complex Fourier series/ J. W. Cooley, J. Tuke // *Mathematics Computation*. – 1965. – Vol. 19 – P. 297–301.
13. Brigham E. Fast Fourier transform and its applications / E. Brigham. – New Jersey : Prentice Hall, Inc., 1988. – 446 p.

Article was submitted 15.12.2014.

After revision 25.12.2014.