

УДК 519.677

doi 10.26089/NumMet.v20r324

## СПЕКТРАЛЬНЫЙ АНАЛИЗ ДИСКРЕТНЫХ СИГНАЛОВ С ВЫСОКИМ ЧАСТОТНЫМ РАЗРЕШЕНИЕМ

О. В. Осипов<sup>1</sup>

Представлены алгоритмы прямого и обратного быстрого преобразования Фурье (БПФ), позволяющие обрабатывать дискретные сигналы с высоким частотным разрешением, в том числе с небольшим количеством отсчетов; получать амплитудно-частотные характеристики с длиной набора частот, большей, чем длина исходного дискретного сигнала. Временная сложность разработанных алгоритмов для прямого и обратного БПФ равна  $O(N \cdot R \cdot \log_2 N)$ , где  $R$  — частотное разрешение спектральной характеристики (отношение длины набора частот к длине  $N$  набора отсчетов сигнала). Разработанные методы позволят увеличить разрешающую способность отечественных систем цифровой обработки сигналов и могут быть реализованы в электронике и программном обеспечении для спектрального анализа.

**Ключевые слова:** быстрое преобразование Фурье (БПФ), спектральный анализ, высокое разрешение, сдвиг частоты, частотно-временное разрешение, проблемы цифровой обработки сигналов (ЦОС), численный итерационный алгоритм БПФ, прямое БПФ, обратное БПФ, амплитудно-частотная характеристика (АЧХ).

**Введение.** Существующие методы анализа сигналов на основе быстрого преобразования Фурье имеют существенный недостаток: при обработке коротких сигналов невозможно получать детальную АЧХ сигнала, а для того чтобы уменьшать интервал между частотами, приходится увеличивать длину выборки исходного сигнала. Это ограничение обычно связывают с принципом Гейзенберга, предполагая невозможность получения хорошего разрешения одновременно и по частоте, и по времени. Вопрос заключается не в том, возможно это или нет, а в том, какое минимальное количество вычислительного времени необходимо затратить для получения приемлемой (в смысле частотного разрешения) спектральной характеристики.

В некоторых случаях для решения проблемы переходят к вейвлет-преобразованиям [1, 2]. Способов увеличения частотного разрешения БПФ в открытых источниках предлагается сравнительно немного: полифазное БПФ [3], добавление нулей к сигналу для увеличения его длительности [4, с. 426]. В некоторых программных реализациях встречается такой грубый способ улучшения разрешения, как увеличение размера окна (длительности сигнала) посредством циклического повторения короткого сигнала. Однако такие методы являются неестественными, потому что изменяют исходный сигнал и процедуре БПФ подвергается уже передискретизированный сигнал. К тому же обратное преобразование без потерь в этих методах выполнить невозможно. Ниже описано, как можно получить АЧХ высокого разрешения без изменения сигнала, выполнив модификацию стандартного быстрого преобразования Фурье.

**1. Алгоритм БПФ с частотным сдвигом.** Сначала проведем анализ алгоритма БПФ с разрежением по времени [5, с. 132] и выясним, каким путем можно исправить описанный выше недостаток. Пусть имеется комплексный сигнал  $S$ , отсчеты которого  $S_i$  ( $i = 0, \dots, N-1$ ) заданы с интервалом  $\Delta t$  в моменты времени  $t_i$  начиная с  $t_0 = 0$ . Обозначим  $F$  — частота дискретизации сигнала,  $\nu$  — частота гармоники,  $j$  — мнимая единица. Далее будем работать с последовательностями  $S$ , длины которых являются степенью двойки.

Прямое дискретное преобразование Фурье, имеющее вид

$$\sum_{i=0}^{N-1} S_i e^{-2\pi j \cdot t_i \nu} = \sum_{i=0}^{N-1} S_i e^{-2\pi j \cdot i \Delta t \nu} = \sum_{i=0}^{N-1} S_i e^{-2\pi j (i \Delta t k F / N)}, \quad (1)$$

будем записывать в более простом виде, полагая  $F \Delta t = 1$ . Тогда преобразование (1) примет вид

$$A(S, N, k) = \sum_{i=0}^{N-1} S_i e^{-2\pi j (ik/N)},$$

<sup>1</sup> Белгородский государственный технологический университет им. В. Г. Шухова, ул. Костюкова, 46, 308012, г. Белгород; доцент, e-mail: osipov.fft@yandex.ru

где параметр  $k$  определяет частоту гармоники через выражение  $\nu = kF/N$ .

Алгоритм БПФ обычно конструируют [6] для расчета значений  $A(S, N, k)$  при  $k = 0, \dots, N-1$ . Поступим иначе — сдвинем набор частот на величину  $\Delta k$  и запишем одну операцию “бабочка” с разрежением по времени: для  $k = 0, \dots, N/2 - 1$

$$\begin{aligned}
 A(S, N, k + \Delta k) &= \sum_{i=0}^{N-1} S_i e^{-2\pi j i (k + \Delta k) / N} = \\
 &= \sum_{i=0}^{N/2-1} S_{2i} e^{-2\pi j i (k + \Delta k) / (N/2)} + e^{-2\pi j (k + \Delta k) / N} \sum_{i=0}^{N/2-1} S_{2i+1} e^{-2\pi j i (k + \Delta k) / (N/2)}, \\
 A(S, N, k + \Delta k + N/2) &= \sum_{i=0}^{N-1} S_i e^{-2\pi j i (k + \Delta k + N/2) / N} = \sum_{i=0}^{N/2-1} S_{2i} e^{-2\pi j i (k + \Delta k + N/2) / (N/2)} + \\
 &+ e^{-2\pi j (k + \Delta k + N/2) / N} \sum_{i=0}^{N/2-1} S_{2i+1} e^{-2\pi j i (k + \Delta k + N/2) / (N/2)} = \\
 &= \sum_{i=0}^{N/2-1} S_{2i} e^{-2\pi j i (k + \Delta k) / (N/2)} - e^{-2\pi j (k + \Delta k) / N} \sum_{i=0}^{N/2-1} S_{2i+1} e^{-2\pi j i (k + \Delta k) / (N/2)},
 \end{aligned} \tag{2}$$

где  $k = 0, \dots, N/2 - 1$ .

Видно, что при сдвиге частот операция “бабочка” так же сокращает количество вычислений вдвое, поэтому ее можно рекуррентно повторить для каждой суммы в (2) в отдельности. *Использование частотного сдвига является основой описываемого метода.* Обозначим поворачивающие множители как

$$W(k, N) = e^{-2\pi j k / N},$$

а четные  $S^{(0)}$  и нечетные  $S^{(1)}$  элементы последовательности  $S$  следующим образом:

$$S^{(0)} = \{S_{2i}\}_{i=0, \dots, N/2-1}, \quad S^{(1)} = \{S_{2i+1}\}_{i=0, \dots, N/2-1}.$$

Тогда (2) можно компактно записать в виде

$$\begin{aligned}
 A(S, N, k + \Delta k) &= A\left(S^{(0)}, \frac{N}{2}, k + \Delta k\right) + W(k + \Delta k, N) \cdot A\left(S^{(1)}, \frac{N}{2}, k + \Delta k\right), \\
 A\left(S, N, k + \Delta k + \frac{N}{2}\right) &= A\left(S^{(0)}, \frac{N}{2}, k + \Delta k\right) - W(k + \Delta k, N) \cdot A\left(S^{(1)}, \frac{N}{2}, k + \Delta k\right).
 \end{aligned}$$

где  $k = 0, \dots, N/2 - 1$ .

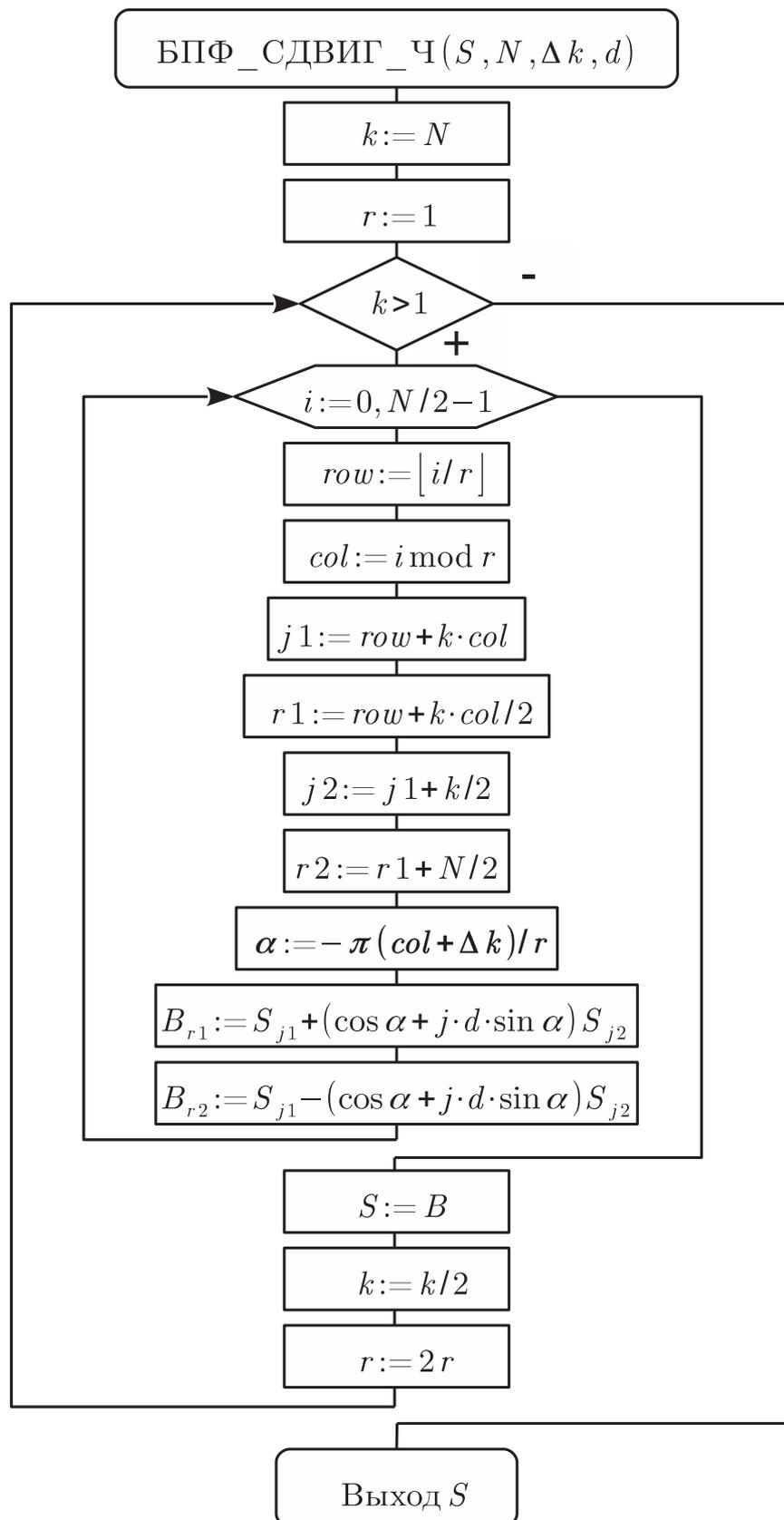
Рекуррентным образом продолжим разложение: для  $k = 0, \dots, N/4 - 1$

$$\begin{aligned}
 A\left(S, \frac{N}{2}, k + \Delta k\right) &= A\left(S^{(0)}, \frac{N}{4}, k + \Delta k\right) + W\left(k + \Delta k, \frac{N}{2}\right) \cdot A\left(S^{(1)}, \frac{N}{4}, k + \Delta k\right), \\
 A\left(S, \frac{N}{2}, k + \Delta k + \frac{N}{4}\right) &= A\left(S^{(0)}, \frac{N}{4}, k + \Delta k\right) - W\left(k + \Delta k, \frac{N}{2}\right) \cdot A\left(S^{(1)}, \frac{N}{4}, k + \Delta k\right); \\
 &\dots\dots\dots \\
 A(S, 2, \Delta k) &= A(S_0, 1, \Delta k) + W(\Delta k, 2) \cdot A(S_1, 1, \Delta k), \\
 A(S, 2, \Delta k + 1) &= A(S_0, 1, \Delta k) - W(\Delta k, 2) \cdot A(S_1, 1, \Delta k), \quad A(S, 1, \Delta k) = S_0,
 \end{aligned}$$

где  $k = 0, \dots, N/4 - 1$ .

Последовательность действий позволяет вычислить БПФ для набора частот  $k + \Delta k$  при  $k = 0, \dots, N - 1$ . В дальнейшем, чтобы получать спектральную характеристику с высоким разрешением, имеет смысл задавать  $\Delta k < 1$ . Для этого необходимо выполнить БПФ несколько раз с различными  $\Delta k$ . К примеру, чтобы увеличить частотное разрешение в 4 раза, надо вычислить БПФ с  $\Delta k$ , равными 0, 1/4, 2/4, 3/4.

Изобразим в виде блок-схемы (рис. 1) итерационный алгоритм для вычисления БПФ со сдвигом по частоте, на основе которого в дальнейшем построим алгоритмы прямого и обратного БПФ с высоким частотным разрешением.

Рис. 1. Блок-схема алгоритма итерационного БПФ со сдвигом по частоте  $\Delta k$

Алгоритм, представленный на рис. 1, является модификацией обычного итерационного алгоритма БПФ. Дополнительный массив  $B$  введен для организации перестановок результатов промежуточных вычислений операцией “бабочка”. Результаты комплексных сложений (вычитаний) записываются в массив  $B$  на позиции  $r1, r2$ , смещенные относительно позиций  $j1, j2$  таким образом, чтобы после окончания алгоритма результат был записан в массиве  $S = B$  в нужном порядке (возрастания частот). Это сделано из соображений минимизации размера алгоритма (объема программного кода), чтобы избавиться от обратной перестановки битов. Платой за это является использование в 2 раза большего количества памяти. Ключевым на рис. 1 является выражение для вычисления угла поворота  $\alpha$ . Оно записано таким образом, чтобы обеспечить частотный сдвиг. Итерации внутреннего цикла можно выполнять независимо друг от друга, т.е. возможно его полное распараллеливание на  $N$  потоков. Параметр  $d \in \{1, -1\}$  отвечает за направление БПФ:  $d = 1$  — прямое,  $d = -1$  — обратное. Результатом работы алгоритма является массив значений

$$A(S, N, k + \Delta k), \quad k = 0, \dots, N - 1,$$

записанных на место исходного сигнала  $S$ .

Алгоритмическая сложность процедуры **БПФ\_СДВИГ\_Ч** составляет  $O(N \log_2 N)$  операций. Реализация данного алгоритма на языке C++ с использованием шаблонного класса `std::complex<double>` выглядит следующим образом:

```

1 void БПФ_СО_СДВИГОМ_ПО_ЧАСТОТЕ_ИТЕРАЦИОННЫЙ(complex<double>* S,
2         int N, double dk, int d)
3 {
4     int k = N, r = 1, i, j1, j2, r1, r2, row, column;
5     double angle;
6     ~// Выделение памяти для дополнительного массива
7     complex<double>* B = new complex<double>[N];
8     while(k > 1)
9     {
10        for (i=0; i < N/2; i++)
11        {
12            row    = i / r;
13            column = i % r;
14            j1 = row + k * column;
15            r1 = row + k * column / 2;
16            j2 = j1 + k / 2;
17            r2 = r1 + N / 2;
18            angle = -M_PI * (column + dk) / r; ~// Угол поворота
19            complex<double> W(cos(angle), d * sin(angle));
20            ~// Операция "Бабочка"
21            B[r1] = S[j1] + W * S[j2];
22            B[r2] = S[j1] - W * S[j2];
23        }
24        memcpy(S, B, N * sizeof(complex<double>)); ~// S := B
25        k /= 2;
26        r *= 2;
27    }
28    delete []B;
29 }
```

Вышеприведенная подпрограмма умышленно сделана с двумя циклами (вместо трех, как во многих реализациях БПФ) и не оптимизирована для большей наглядности. Вычисление частных и остатков от делений целесообразней заменить на побитовые операции, память выделить заранее, а значения тригонометрических функций вычислить предварительно и сохранить в отдельном массиве. Самое главное качество этого алгоритма — возможность его распараллеливания. Вложенный цикл специально организован так, что его итерации абсолютно не зависят друг от друга, а значит, можно оптимизировать его выполнение с помощью OpenCL, OpenMP или векторных инструкций процессора (как в работах [7, 8]).

**2. Прямое БПФ высокого разрешения.** Задача заключается в вычислении дискретного преоб-

разования Фурье сигнала  $S$  длиной  $N$

$$\bar{A}(S, N, m) = \sum_{i=0}^{N-1} S_i e^{-2\pi j \cdot t_i \nu_m}, \quad \nu_m = \frac{mF}{NR}, \quad m = 0, \dots, NR-1, \quad N = 2^p, \quad p, R \in \mathbb{N} \quad (3)$$

для набора частот длиной  $NR$  с использованием БПФ. Здесь длина набора частот  $\nu_m$  в  $R$  раз больше длины набора отсчетов  $S_i$ . Представим  $m = Rk + r$  ( $k = 0, \dots, N-1$ ;  $r = 0, \dots, R-1$ ) и запишем (3) в виде

$$\bar{A}(S, N, m) = \sum_{i=0}^{N-1} S_i e^{-2\pi j i m / NR} = \sum_{i=0}^{N-1} S_i e^{-2\pi j i (kR+r) / (NR)} = \sum_{i=0}^{N-1} S_i e^{-2\pi j i (k+r/R) / N},$$

или учитывая, что  $k = \lfloor m/R \rfloor$ ,  $r = m \bmod R$ :

$$\bar{A}(S, N, m) = A\left(S, N, \left\lfloor \frac{m}{R} \right\rfloor + m \bmod R\right). \quad (4)$$

Из уравнения (4) следует, что для получения амплитуд всего набора частот длиной  $NR$  необходимо выполнить процедуру **БПФ\_СДВИГ\_Ч**  $R$  раз (количество возможных значений  $m \bmod R$ ). Рассмотрим это обстоятельство подробнее. Для первого набора частот, удовлетворяющих условию  $m \bmod R = 0$ , рассчитывается БПФ с  $\Delta k = 0$ :

$$\bar{A}(S, N, kR) = A(S, N, k).$$

Для частот  $m = kR + 1$  необходимо при расчете БПФ сдвинуть частоту  $k$  на  $\Delta k = 1/R$ :

$$\bar{A}(S, N, kR + 1) = A(S, N, k + 1/R).$$

Повторяя этот процесс далее до последнего набора частот ( $m \bmod R = R - 1$ ), получим

$$\bar{A}(S, N, (k+1)R - 1) = A(S, N, k + (R-1)/R).$$

Изобразим алгоритм в виде блок-схемы (рис. 2).

На выходе процедуры **БПФ\_ВР** получим искомого значения (3), записанные в массив  $A$  длиной  $NR$ . Так как процедура **БПФ\_СДВИГ\_Ч** выполняется  $R$  раз (рис. 2), алгоритмическая сложность **БПФ\_ВР** равна  $O(NR \log_2 N)$ . В качестве примера на рис. 3 представлена схема расчета АЧХ сигнала  $S$  длиной  $N = 4$  с 4-кратным частотным разрешением.

В соответствии с блок-схемой на рис. 2 приведем реализацию прямого БПФ с высоким разрешением на языке C++:

```

1 void БПФ_С_ВЫСОКИМ_РАЗРЕШЕНИЕМ(complex<double>* S, int N,
2                                 complex<double>* A, int R)
3 {
4     int i, r;
5     complex<double>* T = new complex<double>[N];
6     for (r=0; r < R; r++)
7     {
8         for (i=0; i < N; i++)
9             T[i] = S[i]; // T := S
10        БПФ_СО_СДВИГОМ_ПО_ЧАСТОТЕ_ИТЕРАЦИОННЫЙ(T, N, double(r) / R, 1);
11        for (i=0; i < N; i++)
12            A[R*i+r] = T[i];
13    }
14    delete []T;
15 }

```

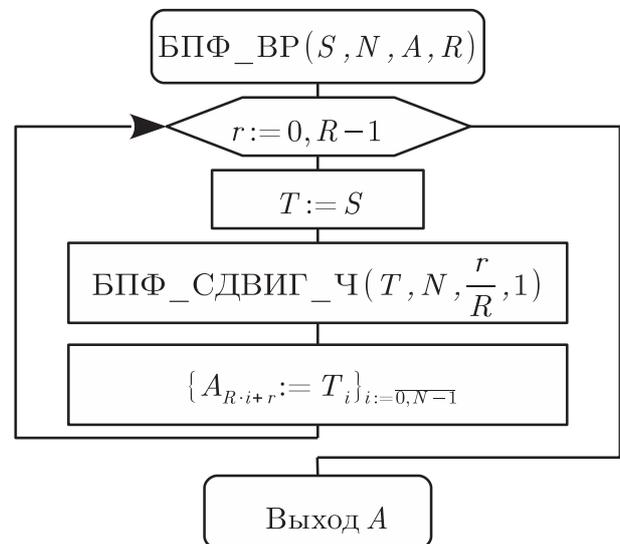


Рис. 2. Общий алгоритм прямого БПФ с  $R$ -кратным разрешением

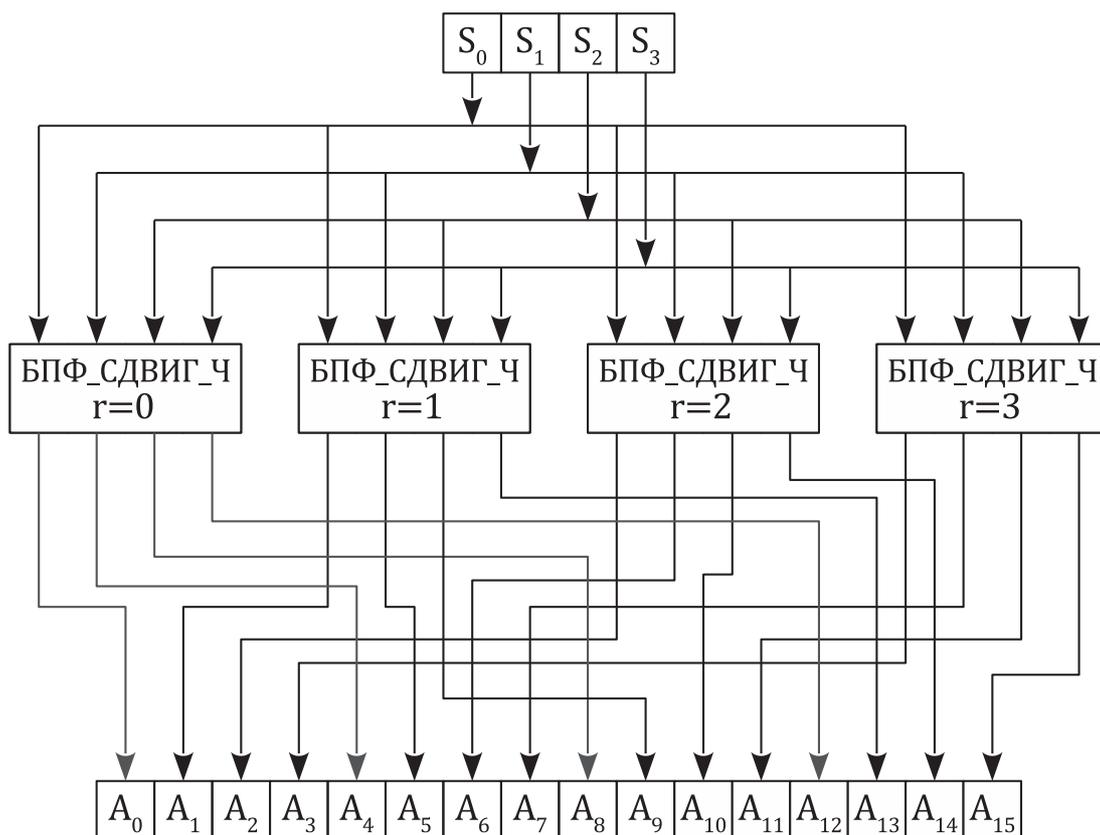


Рис. 3. Схема прямого БПФ с  $N = 4$ ,  $R = 4$

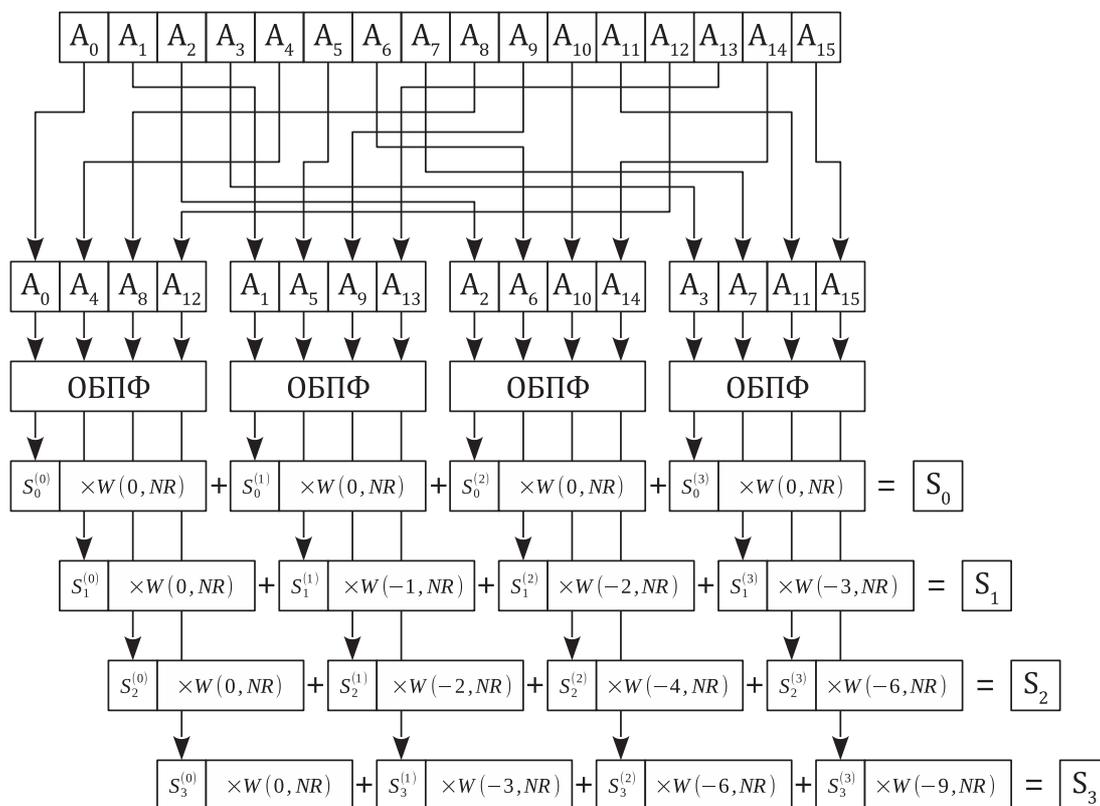


Рис. 4. Схема обратного БПФ с  $N = 4$ ,  $R = 4$

Отметим, что возможно получить такой же результат (набор амплитуд (3)), используя распространенный алгоритм БПФ с добавлением нулей в конец исходного сигнала  $S$ . Однако вычислительная сложность тогда будет  $O(NR \log_2(NR))$ , а необходимое количество памяти —  $NR$  ячеек. Предлагаемый метод имеет сложность  $O(NR \log_2(N))$  и использует  $N$  ячеек памяти внутри процедуры **БПФ\_СДВИГ\_Ч**. Прирост производительности обеспечивается не столько уменьшением количества комплексных умножений, сколько уменьшением потребляемой памяти, потому что обрабатываемые массивы при небольших размерах сигналов  $S$  целиком умещаются в кэш-память первого уровня  $L1$ . Данный алгоритм можно легко распараллелить, выполняя одновременно несколько БПФ с различными значениями  $r$  в отдельных потоках.

**3. Обратное БПФ высокого разрешения.** Пусть в результате прямого преобразования Фурье (3) или иным способом получен набор амплитуд  $\bar{A}_m$  ( $m = 0, \dots, NR - 1$ ). Необходимо восстановить сигнал  $S_i$ , выполнив обратное преобразование Фурье: для  $i = 0, \dots, N - 1$

$$S_i = \frac{1}{NR} \sum_{m=0}^{NR-1} \bar{A}_m e^{2\pi j m i / (NR)}. \quad (5)$$

Выполним  $R$ -кратное разрежение по частоте последовательности (5):

$$\begin{aligned} S_i &= \frac{1}{NR} \sum_{m=0}^{NR-1} \bar{A}_m e^{2\pi j m i / (NR)} = \\ &= \frac{1}{NR} \sum_{r=0}^{R-1} \sum_{k=0}^{N-1} \bar{A}_{kR+r} e^{2\pi j (kR+r) i / (NR)} = \\ &= \frac{1}{NR} \sum_{r=0}^{R-1} e^{2\pi j r i / (NR)} \sum_{k=0}^{N-1} \bar{A}_{kR+r} e^{2\pi j k i / N}. \end{aligned}$$

Если обозначить элементы ОБПФ каждой из  $r$  последовательностей в виде

$$\begin{aligned} S_i^{(r)} &= \sum_{k=0}^{N-1} \bar{A}_{kR+r} e^{2\pi j k i / N}, \quad \text{то} \\ S_i &= \frac{1}{NR} \sum_{r=0}^{R-1} S_i^{(r)} e^{2\pi j r i / (NR)} = \\ &= \frac{1}{NR} \sum_{r=0}^{R-1} S_i^{(r)} W(-ri, NR). \end{aligned}$$

Для решения задачи (5) необходимо разбить  $\bar{A}$  на  $R$  последовательностей каждая длиной  $N$ , как показано на рис. 4. Для каждой последовательности выполняется обратное БПФ без сдвига по частоте и умножение на поворачивающие множители  $W(-ri, NR)$ .

Таким образом, чтобы восстановить сигнал, необходимо так же выполнить обратное БПФ  $R$  раз. Для этого воспользуемся функцией **БПФ\_СДВИГ\_Ч** (рис. 1) с параметрами  $\Delta k = 0, d = -1$ . Может показаться, что при восстановлении сигнала информация о гармониках избыточна, так как длина последовательности  $\bar{A}$  больше длины  $S$ . Однако в некоторых прикладных задачах, например при обработке аудиосигналов (удаление шумов, создание эквалайзера), такая операция имеет смысл и позволяет редактировать частотно-временную характеристику с большей точностью, чем стандартное БПФ с равными длинами  $A$  и  $S$ . На рис. 5 представлена блок-схема алгоритма обратного быстрого преобразования Фурье с высоким частотным разрешением.

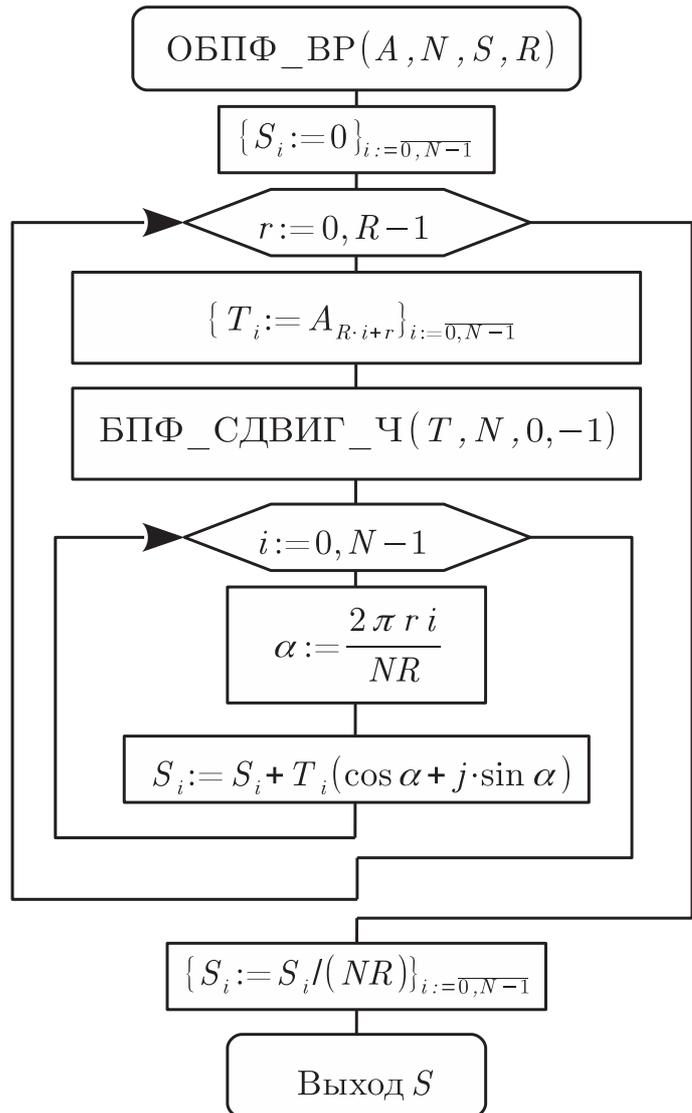


Рис. 5. Общий алгоритм обратного БПФ с  $R$ -кратным разрежением

На языке C++ алгоритм ОБПФ с высоким частотным разрешением, изображенный на рис. 5, может быть реализован следующим образом:

```

1 void ОБПФ_С_ВЫСОКИМ_РАЗРЕШЕНИЕМ(complex<double>* A, int N,
2                                   complex<double>* S, int R)
3 {
4     int i, r;
5     complex<double>* T = new complex<double>[N];
6     for (i=0; i < N; i++)
7         S[i] = 0;
8     for (r=0; r < R; r++)
9     {
10        for (i=0; i < N; i++)
11            T[i] = A[R*i + r];
12        БПФ_СО_СДВИГОМ_ПО_ЧАСТОТЕ_ИТЕРАЦИОННЫЙ(T, N, 0, -1);
13        for (i=0; i < N; i++)
14        {
15            double angle = 2 * PI * r * i / (N * R); // Угол поворота
16            complex<double> W(cos(angle), sin(angle));
17            S[i] += T[i] * W;
18        }
19    }
20    for (i=0; i < N; i++)
21        S[i] /= (N * R);
22    delete []T;
23 }

```

**4. Обратимость прямого и обратного дискретных преобразований Фурье с высоким разрешением.** Численные эксперименты показывают, что преобразования (3) и (5) обратимы с точностью до ошибок округления вещественных чисел, вносимых сопроцессором при выполнении операций с плавающей запятой. Можно доказать обратимость математически, подставив (3) в (5):

$$\begin{aligned}
 S_i &= \sum_{m=0}^{NR-1} \frac{1}{NR} \left( \sum_{\hat{i}=0}^{N-1} S_{\hat{i}} e^{-2\pi j \hat{i} m / (NR)} \right) e^{2\pi j i m / (NR)} = \\
 &= \frac{1}{NR} \sum_{m=0}^{NR-1} \sum_{\hat{i}=0}^{N-1} S_{\hat{i}} e^{-2\pi j (\hat{i}-i) m / (NR)} = \frac{1}{NR} \sum_{\hat{i}=0}^{N-1} S_{\hat{i}} \sum_{m=0}^{NR-1} e^{-2\pi j (\hat{i}-i) m / (NR)}.
 \end{aligned}
 \tag{6}$$

Обозначим  $h = \hat{i} - i$ ,  $M = NR$  и рассмотрим отдельно выражение

$$E(M, h) = \sum_{m=0}^{M-1} e^{-2\pi j h m / M}.$$

Очевидно, что  $E(M, 0) = M$ . Рассмотрим случай  $h \neq 0$ . Сгруппируем в  $E(M, h)$  слагаемые с номерами  $m$  и  $m + M/2$ :

$$E(M, h) = \sum_{m=0}^{M/2-1} \left( e^{-2\pi j h m / M} + e^{-2\pi j h (m+M/2) / M} \right) = \sum_{m=0}^{M/2-1} \left( e^{-2\pi j h m / M} (1 + e^{-2\pi j h / 2}) \right).$$

Вынесем сумму за скобки и продолжим разложение:

$$\begin{aligned}
 E(M, h) &= (1 + e^{-2\pi j h / 2}) \sum_{m=0}^{M/2-1} e^{-2\pi j (h m / 2) / (M/2)} = (1 + e^{-2\pi j h / 2}) E\left(\frac{M}{2}, \frac{h}{2}\right) = \\
 &= (1 + e^{-2\pi j h / 2}) (1 + e^{-2\pi j h / 4}) E\left(\frac{M}{4}, \frac{h}{4}\right) = \dots = \\
 &= E\left(\frac{M}{N}, \frac{h}{N}\right) \prod_{p=1}^{\log_2 N} \left(1 + e^{-2\pi j h / (2^p)}\right).
 \end{aligned}
 \tag{7}$$

Поскольку  $-N < h < N$  и  $N = M/R$ , то среди множителей в (7) обязательно найдется нулевой. Если  $h \bmod 2 = 1$ , то  $e^{-\pi j h} = -1$  при  $p = 1$ , а значит,  $E(M, h) = 0$ . Если  $h \bmod 2 = 0$ , то при последовательном увеличении  $p$  обязательно найдется целое  $h_0 = h/(2^{p-1})$ , такое, что  $h_0 \bmod 2 = 1$ .  $e^{-\pi j h_0} = -1$ , если  $h_0$  — нечетное.

Из этого следует, что  $E(M, h) = 0$  при  $-N < h < N$  и  $h \neq 0$ . Значит, в уравнении (6) вторая сумма не равна нулю только при  $\hat{i} = i$ ; следовательно

$$S_i = \frac{1}{M} \sum_{\hat{i}=0}^{N-1} S_{\hat{i}} \sum_{m=0}^{M-1} e^{-2\pi j(\hat{i}-i)m/M} = \frac{1}{M} S_i M = S_i.$$

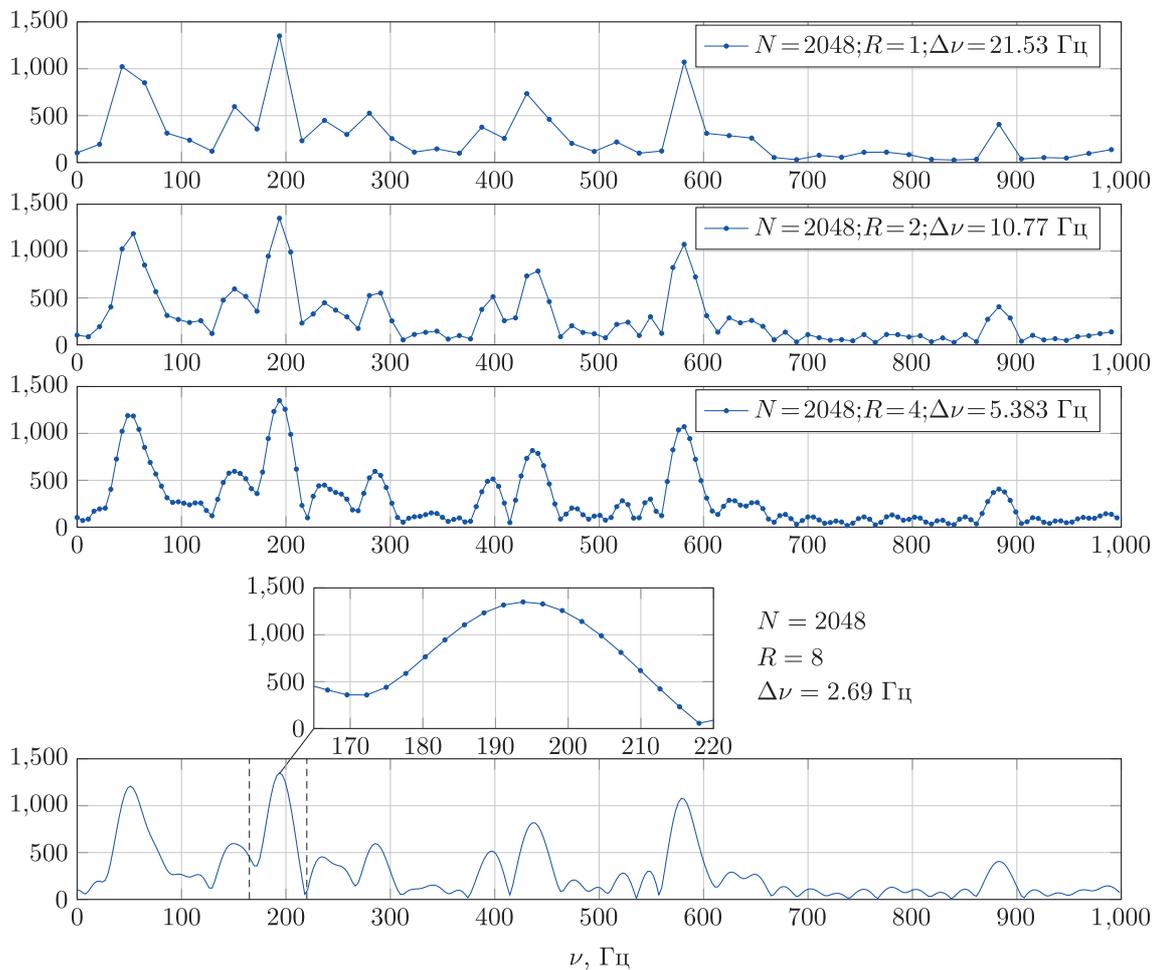


Рис. 6. Спектры акустического сигнала с различным разрешением  $R$

## 5. Результаты численных экспериментов.

**Эксперимент 1.** Построим АЧХ акустического сигнала длительностью  $2048/44100 \approx 0.046$  сек., взятого из звуковой дорожки (wav-файла) с частотой дискретизации 44 100 Гц и с различным разрешением  $R$  (рис. 6). Из рис. 6 понятен смысл разработанного алгоритма прямого преобразования Фурье с высоким частотным разрешением (раздел 2). С ростом числа  $R$  на график добавляются дополнительные точки и спектр сигнала сглаживается. Для наглядности изображена только часть спектра: до 1000 Гц (из 22 050 Гц возможных).

**Эксперимент 2.** Для пилообразного сигнала, изображенного на рис. 7, вычислим преобразование Фурье двумя способами: аналитическим (рис. 8а) и численным (рис. 8б) с использованием алгоритма, описанного в разделе 2.

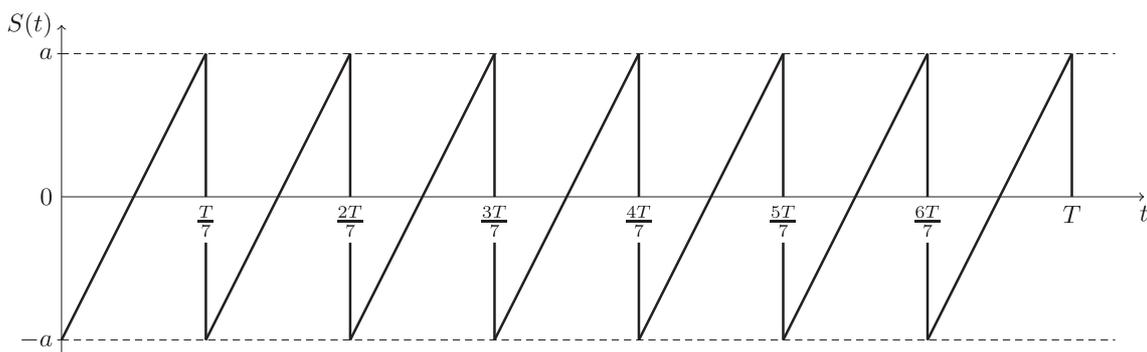


Рис. 7. График функции сигнала  $S = S(t)$

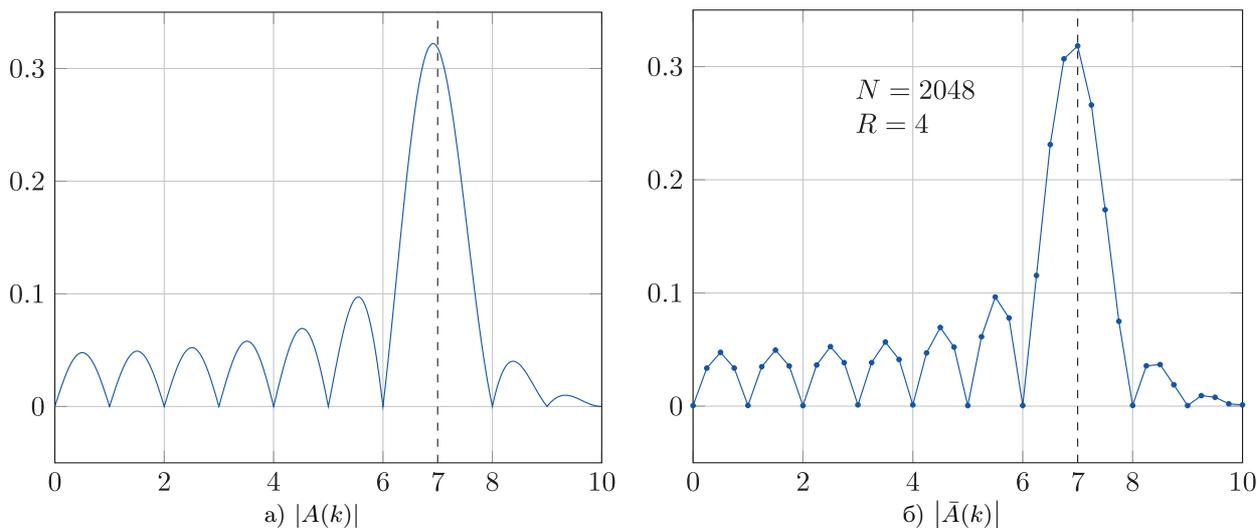


Рис. 8. Спектр сигнала  $S(t)$ , вычисленный аналитически (а) и численно (б)

Запишем преобразование Фурье сигнала  $S(t)$  на отрезке  $[0, T]$  аналитически:

$$\begin{aligned} \text{Im}(A(k)) &= \int_0^T S(t) \sin(-2\pi kt) dt = -\frac{an}{2\pi^2 k^2 T} \sin(2\pi kT) + \\ &+ a \cdot \frac{\cos(2\pi kT) + 1}{2\pi k} + \frac{a}{\pi k} \sum_{i=1}^{n-1} \cos(2\pi kTi/n), \\ \text{Re}(A(k)) &= \int_0^T S(t) \cos(-2\pi kt) dt = -\frac{an}{2\pi^2 k^2 T} \cdot (1 - \cos(2\pi kT)) + \\ &+ \frac{a \sin(2\pi kT)}{2\pi k} + \frac{a}{\pi k} \sum_{i=1}^{n-1} \sin(2\pi kTi/n), \end{aligned} \tag{8}$$

где  $n$  — целое количество гребней на отрезке  $[0, T]$ . Аналитическое выражение (8) и график функции для него при заданном  $n$  можно получить с помощью программы на языке MATLAB:

```

1 syms t k T a p
2 n = 7 % Количество отрезков (гребней)
3 fs = 0 % Интеграл по синусам (мнимая часть)
4 fc = 0 % Интеграл по косинусам (действительная часть)
5 % Функция имеет разрывы, поэтому необходимо интегрировать по частям
6 for i = 1:n
7     f = sym('2*n*a*(t - i*T/n) / T - a')
```

```

8     f = subs(f, {'n', 'i'}, {n, i - 1})
9     % Символьное интегрирование i-го отрезка
10    fs = fs + int(f * sym('sin(-2*p*k*t)'), t, T*(i - 1)/n, T*i/n)
11    fc = fc + int(f * sym('cos(-2*p*k*t)'), t, T*(i - 1)/n, T*i/n)
12    end
13    A = sqrt(fs*fs + fc*fc) % Функция зависимости амплитуды от частоты A(k)
14    A = subs(A, {T, a, p}, {1, 1, pi})
15    A = simplify(A)
16    ezplot(A, 0, 10), ylabel('A(k)'), xlabel('k, Герц')

```

На рис. 8а представлен график функции  $F(k) = |A(k)|$  при  $T = 1$ ,  $a = 1$ ,  $k \in [0, 10]$ . Дискретная функция  $\bar{F}(k) = |\bar{A}(k)|$ , полученная с использованием БПФ при  $N = 2048$  и  $R = 4$  для того же сигнала  $S(t)$ , изображена на соседнем рис. 8б. На этом же рисунке видно, что на одну единицу по оси  $k$  приходится по  $R = 4$  точки. При увеличении  $R$  вырастет количество точек на графике, следовательно улучшится разрешение спектральной характеристики  $|\bar{A}(k)|$ .

В таблицу сведены значения средних квадратических отклонений (СКО)

$$\sqrt{\frac{2}{NR} \sum_{i=0}^{NR/2-1} \left( \left| A\left(\frac{i}{R}\right) \right| - \left| \bar{A}\left(\frac{i}{R}\right) \right| \right)^2}$$

при различных параметрах  $N$  и  $R$ . Отметим, что при вычислении СКО используется только первая половина спектра ( $NR/2$  точек), так как вторая половина спектра ввиду специфики самого алгоритма БПФ является отражением первой и не соответствует аналитическим значениям функции  $A(k)$ .

Зависимость СКО аналитического и численного преобразований Фурье сигнала  $S(t)$  при различных  $N$ ,  $R$

$R \setminus N$	8	32	128	512	2048	8192
1	0.20729	0.049803	0.012804	0.0032661	$8.1489 \times 10^{-4}$	$2.0380 \times 10^{-4}$
2	0.1833	0.042143	0.011062	0.0029066	$7.3585 \times 10^{-4}$	$1.7697 \times 10^{-4}$
4	0.17837	0.038375	0.010336	0.0026853	$6.96 \times 10^{-4}$	$1.6515 \times 10^{-4}$
8	0.177	0.037263	0.010083	$2.6034 \times 10^{-3}$	$6.8235 \times 10^{-4}$	$1.6116 \times 10^{-4}$
16	0.17666	0.036953	0.0099936	$2.5759 \times 10^{-3}$	$6.7803 \times 10^{-4}$	$1.5987 \times 10^{-4}$
32	0.17657	0.036874	0.0099671	$2.5671 \times 10^{-3}$	$6.7671 \times 10^{-4}$	$1.5947 \times 10^{-4}$
64	0.17655	0.036854	0.0099602	$2.5644 \times 10^{-3}$	$6.7631 \times 10^{-4}$	$1.5935 \times 10^{-4}$

Из таблицы видно, что с ростом значений  $R$ ,  $N$  СКО уменьшается. При увеличении  $N$  и постоянном  $R$  дискретный сигнал приближается к  $S(t)$ , поэтому СКО в этом случае уменьшается быстрее. СКО уменьшается стабильно, что говорит о хорошей корреляции спектров  $|A(k)|$  и  $|\bar{A}(k)|$  даже при больших значениях  $R$  и  $N$ .

Наоборот, увеличение  $R$  при постоянном  $N$  не добавляет точек в исходный дискретный сигнал, т.е. не приближает этот сигнал к аналитической функции  $S(t)$ , поэтому СКО уменьшается незначительно. Такое поведение, вообще говоря, не требуется, и при других тестовых данных (другой функции сигнала) СКО в этом случае может не уменьшаться.

**Заключение.** Необходимость в описанных методах возникла при решении задачи определения точной высоты тона звучания струнных музыкальных инструментов в режиме реального времени. Полученная точность расчета приблизительно равна 0.5–1.0 Гц при частоте дискретизации 44 100 Гц,  $512 \leq N \leq 4096$  и  $R$  порядка нескольких десятков (с использованием языка C++). На компьютере средней мощности при загрузке вещественного сопроцессора такая точность оказалась вполне приемлемой даже без использования специальных оптимизаций БПФ (например, распараллеливание). Созданные алгоритмы можно использовать в задачах, требующих высокую точность спектральной характеристики при исследовании дискретных сигналов и другой физической природы — электромагнитных, радиосигналов.

Разработанные методы позволяют улучшить точностные характеристики отечественного [9] программного и аппаратного обеспечения для проведения спектрального анализа различных физических процессов.

Основные алгоритмы очень просты для реализации практически на любом языке программирования, а также на ПЛИС [10, 11].

## СПИСОК ЛИТЕРАТУРЫ

1. *Зубаков А.П.* Фурье и вейвлет-преобразования в проблеме распознавания речи // Вестник Тамбовского университета. Серия: Естественные и технические науки. 2010. **15**, № 6. 1893–1899.
2. *Голубинский А.Н., Астахов Р.А.* О частотно-временном разрешении кратковременного Фурье-анализа и непрерывного вейвлет-анализа при обработке речевых сигналов // Вестник Воронежского института МВД России. 2013. № 3. 4–11.
3. <http://www.dsplib.ru/content/polyphasefft/polyphase.html>
4. *Рабинер Л., Гоулд Б.* Теория и применение цифровой обработки сигналов. М.: Мир, 1978.
5. *Гадзиковский В.И.* Цифровая обработка сигналов. М.: СОЛОН-Пресс, 2013.
6. *Выдрин Д.Ф., Абзалылова Ю.Р., Вдовин А.К.* Быстрое преобразование Фурье в цифровой обработке сигналов // Теория и практика современной науки. 2017. № 2. 161–163.
7. *Божж Г.О.* Оптимизация ассемблерного кода БПФ, ориентированного на обработку OFDM сигналов сотовых сетей связи // Экономика и качество систем связи. 2017. № 4. 40–51.
8. *Бурцев А.А.* Оптимизация алгоритмов быстрого преобразования Фурье для специализированного векторного сопроцессора с учетом иерархической структуры памяти // Труды научно-исследовательского Института системных исследований РАН. 2017. **7**, № 4. 83–95.
9. *Ишин П.А.* Оптимизация преобразования Фурье под архитектуру “Эльбрус” // Современные информационные технологии и ИТ-образование. 2011. № 7. 683–691.
10. *Галанина Н.А., Иванова Н.Н.* Вычислительные аспекты быстрого преобразования Фурье и вопросы его реализации на ПЛИС // Вестник Чувашского университета. 2018. № 3. 172–181.
11. *Капитанов А.А., Остапенков П.С.* Реализация высокоскоростного блока быстрого преобразования Фурье на программируемых логических интегральных схемах // Вестник Московского энергетического института. 2015. № 2. 92–97.

Поступила в редакцию  
20.04.2019

---

## Spectral Analysis of Discrete Signals with High Frequency Resolution

O. V. Osipov<sup>1</sup>

<sup>1</sup> *Shukhov Belgorod State Technological University; ulitsa Kostyukova 44, Belgorod, 308012, Russia; Ph.D., Associate Professor, e-mail: osipov.fft@yandex.ru*

Received April 20, 2019

**Abstract:** Algorithms of direct and inverse fast Fourier transforms are discussed. These algorithms allow one to process discrete signals with high frequency resolution, including with a small number of frequency samples, and to receive the frequency responses with a set length of frequencies greater than the length of the original discrete signal. The time complexity of the developed algorithms for the direct and inverse FFT is  $O(NR \log_2 N)$ , where  $R$  is the frequency resolution of the spectral characteristic (the ratio of the length of a set of frequencies to the length  $N$  of a set of signal samples). The developed methods allow one to increase the resolution of systems of digital signal processing and can be implemented in electronic devices and in software for spectral analysis.

**Keywords:** fast Fourier transform (FFT), spectral analysis, high resolution, frequency shift, time-frequency resolution, digital signal processing (DSP) problems, numerical iterative FFT algorithm, forward FFT, inverse FFT, amplitude-frequency characteristic.

### References

1. A. P. Zubakov, “Fourier and Wavelet-Transformations in the Problem of Speech Identification,” *Vestn. Tambov Gos. Univ. Ser. Estestven. Tekhnich. Nauki* **15** (6), 1893–1899 (2010).

2. A. N. Golubinsky and R. A. Astashov, "On Time-and-Frequency Resolution of the Short-Term Fourier Analysis and the Continuous Wavelet Analysis at Speech Signals," *Vestn. Voronezh Inst. Ross. MVD*, No. 3, 4–11 (2013).
3. Polyphase FFT. <http://www.dsplib.ru/content/polyphasefft/polyphase.html>. Cited June 22, 2019.
4. L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, 1975; Mir, Moscow, 1978).
5. V. I. Gadzikovskii, *Digital Signal Processing* (SOLON Press, Moscow, 2013) [in Russian].
6. D. F. Vydrin, Yu. R. Abzalilova, and A. K. Vdovin, "Fast Fourier Transformation at Digital Signal Processing," *Teoriya Praktika Sovremen. Nauki*, No. 2, 161–163.
7. G. O. Bokk, "Optimization of the FFT Assembler Code Oriented on the Processing of OFDM Signals of Cellular Communication Networks," *Ekonomika Kachestvo Sistem Svyazi*, No. 4, 40–51 (2017).
8. A. A. Burtsev, "Optimization of Algorithms of fast Fourier Transform for the Specialized Vector Coprocessor Taking into Account Hierarchical Structure of Memory," *Tr. Inst. Sistemn. Issled., Ross. Akad. Nauk* **7** (4), 83–95 (2017).
9. P. A. Ishin, "Optimization of Fourier Transform for Elbrus Architecture," *Sovremen. Inform. Tekhnol. IT-Obrazov.*, No 7, 683–691 (2011).
10. N. A. Galanina and N. N. Ivanova, "Computing Aspects of Fast Fourier Transform and Issues of Its FPGA Realization," *Vestn. Chuvash Gos. Univ.*, No. 3, 172–181 (2018).
11. A. A. Kapitanov and P. S. Ostapenkov, "The High-Speed Floating-Point Fast Fourier Transforms Core Implementation Based on Field-Programmable Gate Arrays," *Vestn. Mosk. Gos. Energetich. Inst.*, No. 2, 92–97 (2015).