

## Алгоритм аппроксимации дискретного сигнала небольшим количеством гармоник с убывающими амплитудами

**О. В. Осипов**

Белгородский государственный технологический университет имени В. Г. Шухова,  
кафедра программного обеспечения вычислительной техники и автоматизированных систем,  
Белгород, Российская Федерация

ORCID: 0000-0002-5812-5338, e-mail: [osipov.fft@yandex.ru](mailto:osipov.fft@yandex.ru)

**Аннотация:** Предложен алгоритм аппроксимации произвольного дискретного сигнала тригонометрическим полиномом с убывающими по амплитуде гармониками, имеющий алгоритмическую сложность  $O(NR(L + \log_2 N))$ , где  $L$  — длина полинома,  $N$  — длина набора отсчетов исходного сигнала,  $NR$  — длина частотного базиса алгоритма быстрого преобразования Фурье (БПФ). Представлены блок-схемы разработанных алгоритмов, исходные тексты программ на языке Python, результаты численных экспериментов. Разработанные алгоритмы могут быть применены для улучшения отечественных технологий в области электроники и программного обеспечения, а также включены в учебные программы инженерных специальностей.

**Ключевые слова:** тригонометрический полином, метод последовательного вычитания гармоник, быстрое преобразование Фурье (БПФ), высокое разрешение, тригонометрическая аппроксимация, метод наименьших квадратов, цифровая обработка сигналов (ЦОС), амплитудный спектр сигнала, анализ данных, растекание спектра.

**Для цитирования:** Осипов О.В. Алгоритм аппроксимации дискретного сигнала небольшим количеством гармоник с убывающими амплитудами // Вычислительные методы и программирование. 2024. 25, № 3. 274–291. doi 10.26089/NumMet.v25r321.

## An algorithm for approximating a discrete signal with a small number of harmonics with decreasing amplitudes

**Oleg V. Osipov**

Shukhov Belgorod State Technological University, Belgorod, Russia

ORCID: 0000-0002-5812-5338, e-mail: [osipov.fft@yandex.ru](mailto:osipov.fft@yandex.ru)

**Abstract:** An algorithm for approximating an arbitrary discrete signal by a trigonometric polynomial with decreasing harmonics in amplitude is proposed. It has an algorithmic complexity of  $O(NR(L + \log_2 N))$ , where  $L$  is the length of the polynomial,  $N$  is the length of the set of samples of the original signal, and  $NR$  is the length of the frequency basis of the fast Fourier transform (FFT) algorithm. The flowcharts of the developed algorithms, the source texts of Python programs, and the results of numerical experiments are presented. The developed algorithms can be applied to improve



domestic technologies in the field of electronics and software, as well as included in the curricula of engineering specialties.

**Keywords:** trigonometric polynomial, sequential harmonic subtraction method, fast Fourier transform (FFT), high resolution, trigonometric approximation, least squares method, digital signal processing (DSP), the amplitude spectrum of the signal, data analysis, spectrum spreading.

**For citation:** O. V. Osipov, “An algorithm for approximating a discrete signal with a small number of harmonics with decreasing amplitudes,” Numerical Methods and Programming. 25 (3), 274–291 (2024). doi 10.26089/NumMet.v25r321.

**1. Введение.** Для аппроксимации периодических сигналов повсеместно используются алгоритмы быстрого преобразования Фурье (БПФ) [1, с. 256; 2; 3]. БПФ обладает тем недостатком, что оно подвержено эффекту растекания спектра [4, 5]. Данный эффект выражается в том, что сигнал аппроксимируется набором гармоник, равномерно распределенным по всему набору частот, и даже обычный синусоидальный сигнал будет аппроксимирован суммой гармоник, большая часть из которых на самом деле отсутствует в сигнале. На избыточный и неестественный характер разложения Фурье указывает, например, автор работ [6, 7]. Существенная часть природных сигналов имеет одну или несколько выраженных гармоник. В связи с этим хотелось бы иметь алгоритм, который будет аппроксимировать периодический сигнал небольшим количеством простых гармоник, т.е. раскладывать сигнал только на его собственные гармоники.

Вопрос представления периодических сигналов как можно меньшим количеством синусоидальных функций время от времени поднимается при решении прикладных задач цифровой обработки сигналов. Одной из ранних статей, в которой рассматривался вопрос об аппроксимации сигнала одной гармонической модельной функцией с оптимальными параметрами частоты, фазы и амплитуды, является работа [8]. В ней фаза и амплитуда гармоники вычисляются в результате решения задачи минимизации среднеквадратичного отклонения (СКО) этой гармоники от исходного сигнала при фиксированной частоте, которая подбирается перебором по всему частотному базису БПФ. Позднее в работах [6, 7] предпринята попытка решить задачу минимизации по всем трем параметрам приближенными численными методами без ограничения на значение частоты. На практике такой итерационный метод применим для некоторого рода сигналов, но не гарантирует нахождение глобального минимума и может остановиться на частоте, соответствующей пику на спектрограмме вблизи начального приближения. Рассматриваемые в данной работе методы можно считать развитием методов [8], в которые вносятся два улучшения: частотный базис БПФ расширяется в  $R$  раз и обеспечивается возможность представлять сигнал несколькими гармониками.

Рассматриваемый в работах [8, 9] метод тригонометрической аппроксимации основан на минимизации квадратичного функционала. Основы аппроксимации периодических функций с известным периодом тригонометрическими полиномами методом наименьших квадратов изложены в справочном руководстве по небесной механике [10, с. 649]. Близким рассматриваемому методу аппроксимации является метод Прони [11, с. 365], который представляет сигнал набором затухающих экспонент. Предлагаемый метод, в отличие от метода Прони, раскладывает сигнал в ряд убывающих по амплитуде синусоид.

В теории машинного обучения в последнее время разработаны методы так называемой “разреженной” аппроксимации сигналов, в которых вместо ортогонального базиса используется словарь функций, который выбирается индивидуально для каждой задачи. С использованием подобных словарей разработаны различные алгоритмы аппроксимации и последующего сжатия “разреженных” сигналов [12] и изображений [13]. Для текущей задачи такие методы плохо применимы, поскольку большое количество гармонических функций с разными частотами потребует хранения объемного словаря и соответствующей разреженной матрицы, что снизит скорость вычислений.

Предлагаемый в данной работе метод можно назвать *методом последовательного вычитания гармоник*. Далее будет рассматриваться алгоритм, который выделяет в сигнале  $L$  самых сильных гармоник. Сигналы, имеющие, к примеру, две выраженные частоты, можно будет аппроксимировать суммой двух гармоник, более сложные сигналы — заданным числом  $L$  простых гармоник. В качестве базиса будет использоваться набор неортогональных тригонометрических функций.

**2. Постановка задачи.** Имеется действительный периодический дискретный сигнал  $C$ , представленный отсчетами в моменты времени  $t_i = i\Delta t$  ( $i = 0, \dots, N - 1$ ,  $N$  является степенью двойки) с частотой дискретизации  $F$ . Необходимо приблизить  $C$  тригонометрическим полиномом

$$S_i = \sum_{l=1}^L (a_l \cos(2\pi\omega_l t_i) + b_l \sin(2\pi\omega_l t_i)) \quad (1)$$

при заданном количестве  $L$  находимых простых гармоник. Искомыми являются параметры гармоник — действительные коэффициенты  $a_l$ ,  $b_l$  и частоты  $\omega_l$  ( $l = 1, \dots, L$ ), которые выбираются из набора частот  $\{kF/(NR)\}_{k=0, \dots, NR-1}$ . Такой набор частот обусловлен выбором алгоритма БПФ с высоким частотным разрешением [14], который понадобится для промежуточных вычислений. Данный алгоритм, который будет рассмотрен ниже, является усовершенствованием классического БПФ и имеет входной параметр  $R$  — частотное разрешение. Чем больше значение  $R$ , тем точнее будет выбираться частота искомой гармоники  $\omega_l$  на  $l$ -м шаге алгоритма и, следовательно, будет лучше аппроксимация. Так как  $\Delta t$  — временной интервал между двумя соседними замерами, то  $F\Delta t = 1$ . Если обозначить  $\omega_l = \lambda_l F/(NR)$ , то можно частоты  $\omega_l$  и моменты времени  $t_i$  в (1) выразить через целые числа  $\lambda_l$  и  $i$  соответственно:

$$S_i = \sum_{l=1}^L \left( a_l \cos\left(\frac{2\pi i \lambda_l}{NR}\right) + b_l \sin\left(\frac{2\pi i \lambda_l}{NR}\right) \right).$$

Для краткости далее введем обозначение:  $\theta = 2\pi/(NR)$ .

**3. Основная идея алгоритма.** Предлагается последовательно исключать из дискретного сигнала самые сильные гармоники, а их параметры  $a$ ,  $b$ ,  $\lambda$  сохранять в массиве троек  $G$ .

Рассмотрим алгоритм по шагам.

1. Инициализируется счетчик искомых гармоник:  $l = 0$ . Положим  $S^{(0)} = C$ . Под переменной  $l$  следует понимать номер слоя сигналов  $S^{(0)}, \dots, S^{(L)}$ . Каждый следующий сигнал  $S^{(l+1)}$  будет получен исключением из текущего сигнала  $S^{(l)}$  найденной гармоники.

2. Для каждого  $k = 0, \dots, NR/2 - 1$  отдельно решаются задачи минимизации СКО сигнала  $S^{(l)}$  от простой гармоники  $G(x, y, k) = x \cos(ik\theta) + y \sin(ik\theta)$  при неизвестных  $x, y$ :

$$O(x, y, k) = \sum_{i=0}^{N-1} (S_i^{(l)} - G(x, y, k))^2 \rightarrow \min.$$

Обозначим  $x_k, y_k$  — значения параметров  $x$  и  $y$ , при которых функция  $O$  достигает минимума при заданном  $k$ , а  $J_k = O(x_k, y_k, k)$  — само значение СКО в точке минимума.

3. В сигнале  $S^{(l)}$  находится самая сильная гармоника. Индекс  $k_{\min}$  минимального элемента последовательности  $\{J_k\}_{k=0, \dots, NR/2-1}$  выбирается частотой самой сильной гармоники и вместе с ее амплитудой запоминается в массиве  $G$ :

$$G_l = (\lambda_l, a_l, b_l), \quad \lambda_l = k_{\min}, \quad a_l = x_{k_{\min}}, \quad b_l = y_{k_{\min}}.$$

Самой сильной гармоникой считается та, среднеквадратичное отклонение которой от сигнала  $S^{(l)}$  минимально.

4. После решения задач минимизации коэффициенты  $a_l$  и  $b_l$  становятся известны и найденная гармоника полностью “вычитается” из сигнала:

$$S_i^{(l+1)} = S_i^{(l)} - a_l \cos\left(\frac{2\pi i \lambda_l}{NR}\right) - b_l \sin\left(\frac{2\pi i \lambda_l}{NR}\right), \quad i = 0, \dots, N - 1. \quad (2)$$

Переменная  $l$  увеличивается на единицу и шаги 2–4 итеративно повторяются еще  $L - 1$  раз для сигналов  $S^{(1)}, \dots, S^{(L-1)}$ . По окончании вычислений в  $S^{(L)}$  будет записан остаточный сигнал.

**4. Определение амплитуды и фазы отдельной гармоники в сигнале.** Необходимо решить несколько отдельных подзадач. Для начала определим, какой вклад вносит  $k$ -я гармоника (гармоника с частотой  $k$ ) в периодический сигнал  $S$ . Каким может быть минимальное возможное отклонение  $S$  от простой гармоники  $x \cos(ik\theta) + y \sin(ik\theta)$ ? Подстрочные индексы  $k$  пока опустим для удобства и будем



считать, что  $x, y$  — искомые параметры  $k$ -й гармоники,  $j$  — мнимая единица. Дан сигнал  $S$ , а также известно его дискретное преобразование Фурье (ДПФ) для данного  $k$ :

$$A = \sum_{i=0}^{N-1} S_i \cos(ik\theta) - j \cdot \sum_{i=0}^{N-1} S_i \sin(ik\theta) = \sum_{i=0}^{N-1} S_i e^{-j \cdot ik\theta}.$$

Рассмотрим задачу минимизации СКО  $S$  от простой гармоники с заданной частотой  $k$  ( $0 \leq k < NR/2$ ):

$$F(x, y) = \sum_{i=0}^{N-1} (S_i - x \cos(ik\theta) - y \sin(ik\theta))^2 \rightarrow \min.$$

Искомые параметры гармоники  $x$  и  $y$  определяются из решения системы

$$\frac{\partial F(x, y)}{\partial x} = 0, \quad \frac{\partial F(x, y)}{\partial y} = 0. \tag{3}$$

Выражение для гармоники преобразуем с помощью формулы Эйлера в комплексную форму, чтобы упростить дальнейшие тригонометрические преобразования:

$$x \cos(ik\theta) + y \sin(ik\theta) = \frac{x + j \cdot y}{2} e^{-j \cdot ik\theta} + \frac{x - j \cdot y}{2} e^{j \cdot ik\theta}. \tag{4}$$

Найдем производные (3):

$$\begin{aligned} \frac{\partial F}{\partial x} &= -2 \sum_{i=0}^{N-1} (\cos(ik\theta) (S_i - x \cos(ik\theta) - y \sin(ik\theta))) = \\ &= -2 \sum_{i=0}^{N-1} (S_i \cos(ik\theta)) + 2 \sum_{i=0}^{N-1} \left( \left( \frac{x + j \cdot y}{2} e^{-j \cdot ik\theta} + \frac{x - j \cdot y}{2} e^{j \cdot ik\theta} \right) \cos(ik\theta) \right), \\ \frac{\partial F}{\partial y} &= -2 \sum_{i=0}^{N-1} (\sin(ik\theta) (S_i - x \cos(ik\theta) - y \sin(ik\theta))) = \\ &= -2 \sum_{i=0}^{N-1} (S_i \sin(ik\theta)) + 2 \sum_{i=0}^{N-1} \left( \left( \frac{x + j \cdot y}{2} e^{-j \cdot ik\theta} + \frac{x - j \cdot y}{2} e^{j \cdot ik\theta} \right) \sin(ik\theta) \right). \end{aligned} \tag{5}$$

Обозначим точку минимума функции  $F(x, y)$  как  $(\tilde{x}, \tilde{y})$ , запишем ее в комплексной форме:  $z = \tilde{x} + j \cdot \tilde{y}$ , приравняем производные (5) к нулю и сократим полученную систему, учитывая, что левые слагаемые являются частями комплексной амплитуды  $A$ :

$$\begin{aligned} 2 \operatorname{Re}(A) &= \sum_{i=0}^{N-1} ((ze^{-j \cdot ik\theta} + \bar{z}e^{j \cdot ik\theta}) \cos(ik\theta)), \\ -2 \operatorname{Im}(A) &= \sum_{i=0}^{N-1} ((ze^{-j \cdot ik\theta} + \bar{z}e^{j \cdot ik\theta}) \sin(ik\theta)). \end{aligned} \tag{6}$$

Умножим последнее уравнение на мнимую единицу и сложим отдельно левые и правые части (6):

$$2\bar{A} = \sum_{i=0}^{N-1} ((ze^{-j \cdot ik\theta} + \bar{z}e^{j \cdot ik\theta}) e^{j \cdot ik\theta}) = zN + \bar{z} \sum_{i=0}^{N-1} e^{j \cdot 2ik\theta}. \tag{7}$$

Выразим из последнего уравнения  $z$  и  $\bar{z}$ :

$$z = \frac{2(\bar{A}N - A\bar{Q})}{N^2 - |Q|^2}, \quad \bar{z} = \frac{2(AN - \bar{A}\bar{Q})}{N^2 - |Q|^2}, \quad \text{где } Q = \sum_{i=0}^{N-1} e^{j \cdot 2ik\theta}.$$

Искомые амплитуды гармоники определяются из  $z$  простыми выражениями:  $\tilde{x} = \operatorname{Re}(z)$ ,  $\tilde{y} = \operatorname{Im}(z)$ . Теперь точка минимума функции СКО найдена и все параметры гармоники известны. Полезно было бы

знать значение самой функции  $F$  в точке минимума. Если действовать путем прямой подстановки, то объем вычислений функции  $F$  окажется сравним с вычислением ДПФ для  $S$ . Поэтому лучше упростить выражение для  $F(\tilde{x}, \tilde{y})$ :

$$F(\tilde{x}, \tilde{y}) = \sum_{i=0}^{N-1} \left( S_i - \frac{\tilde{x} + j \cdot \tilde{y}}{2} e^{-j \cdot i k \theta} - \frac{\tilde{x} - j \cdot \tilde{y}}{2} e^{j \cdot i k \theta} \right)^2 = -2\tilde{x} \operatorname{Re}(A) + 2\tilde{y} \operatorname{Im}(A) + \frac{z^2}{4} \bar{Q} + \frac{\bar{z}^2}{4} Q + \frac{z\bar{z}}{2} N + \sum_{i=0}^{N-1} S_i^2. \quad (8)$$

Сумму, содержащую  $z$ , сократим отдельно:

$$\begin{aligned} \frac{z^2}{4} \bar{Q} + \frac{\bar{z}^2}{4} Q + \frac{z\bar{z}}{2} N &= 2\operatorname{Re} \left( \frac{z^2}{4} \bar{Q} \right) + \frac{z\bar{z}}{2} N = \\ &= \operatorname{Re} \left( \frac{z^2}{2} \bar{Q} + \frac{z\bar{z}}{2} N \right) + j \cdot \operatorname{Im} \left( \frac{z\bar{z}}{2} N \right) = \operatorname{Re} \left( \frac{z}{2} (z\bar{Q} + \bar{z}N) \right). \end{aligned}$$

Из уравнения (7) следует, что  $z\bar{Q} + \bar{z}N = 2A$ , тогда

$$\operatorname{Re} \left( \frac{z}{2} (z\bar{Q} + \bar{z}N) \right) = \operatorname{Re}(zA) = \tilde{x} \operatorname{Re}(A) - \tilde{y} \operatorname{Im}(A).$$

Подставим последнее выражение в уравнение (8) и получим итоговое выражение для  $F(\tilde{x}, \tilde{y})$ :

$$F(\tilde{x}, \tilde{y}) = -\tilde{x} \operatorname{Re}(A) + \tilde{y} \operatorname{Im}(A) + \sum_{i=0}^{N-1} S_i^2.$$

Возвратим подстрочные индексы  $k$ . Значения аргументов функции  $F(x, y)$ , при которых она достигает минимума при заданном  $k$ , обозначим как  $x_k, y_k$ . Тогда исчерпывающее решение задачи минимизации для  $k$ -й гармоники ( $0 < k < NR/2$ ) запишется в виде:

$$z_k = \frac{2(\bar{A}_k N - A_k Q_{2k})}{N^2 - |Q_{2k}|^2}, \quad A_k = \sum_{i=0}^{N-1} S_i e^{-2\pi j \frac{ik}{NR}}, \quad Q_k = \sum_{i=0}^{N-1} e^{2\pi j \frac{ik}{NR}},$$

$$z_k = x_k + j \cdot y_k, \quad J_k = F_k(x_k, y_k) = -x_k \operatorname{Re}(A_k) + y_k \operatorname{Im}(A_k) + \sum_{i=0}^{N-1} S_i^2.$$

При  $k = 0$  выражение для  $z_k$  вычисляется иначе, потому что  $N^2 - |Q_0|^2 = 0$ . Оно представляет собой свободный член и определяется вместе с  $J_0$  так:

$$z_0 = \frac{1}{N} \sum_{i=0}^{N-1} S_i, \quad J_0 = \sum_{i=0}^{N-1} S_i^2 - \frac{1}{N} \left( \sum_{i=0}^{N-1} S_i \right)^2.$$

Можно вычислить параметры гармоники в середине спектра:

$$z_{NR/2} = \frac{1}{N} \sum_{i=0}^{N/2-1} (S_{2i} - S_{2i+1}), \quad J_{NR/2} = \sum_{i=0}^{N-1} S_i^2 - \frac{1}{N} \left( \sum_{i=0}^{N/2-1} (S_{2i} - S_{2i+1}) \right)^2.$$

Если необходимо, выражение для самой гармоники можно записать через полную амплитуду и начальную фазу:

$$x_k \cos \left( \frac{2\pi ik}{NR} \right) + y_k \sin \left( \frac{2\pi ik}{NR} \right) = \sqrt{x_k^2 + y_k^2} \cos \left( \frac{2\pi ik}{NR} - \varphi_k \right),$$

где  $\varphi_k = \operatorname{arctg}(y_k/x_k)$  — начальная фаза  $k$ -й гармоники.



**5. Вычитание гармоник из спектра.** Согласно основному алгоритму, описанному в начале, найденные гармоники необходимо исключать из сигнала по формуле (2). Поскольку  $z_k$  выражается через значения амплитуды  $A_k$ , требуется на каждом шаге алгоритма сопровождать сигнал  $S^{(l)}$  его спектром  $A^{(l)}$ . Это можно делать с помощью БПФ напрямую:

$$A_k^{(l)} = \sum_{i=0}^{N-1} S_i^{(l)} e^{-2\pi j \frac{ik}{NR}}.$$

Но в таком случае потребуется вычислять БПФ заново на каждом шаге алгоритма. Чтобы этого не делать, выразим с помощью тригонометрических преобразований значения каждого следующего спектра  $A_k^{(l+1)}$  через значения предыдущего спектра  $A_k^{(l)}$ . Подставим вместо  $S_i^{(l+1)}$  формулу (2) в спектр  $A_k^{(l+1)}$ :

$$A_k^{(l+1)} = \sum_{i=0}^{N-1} S_i^{(l+1)} e^{-2\pi j \frac{ik}{NR}} = \sum_{i=0}^{N-1} \left( S_i^{(l)} - a_l \cos\left(\frac{2\pi i \lambda_l}{NR}\right) - b_l \sin\left(\frac{2\pi i \lambda_l}{NR}\right) \right) e^{-2\pi j \frac{ik}{NR}},$$

где  $a_l, b_l, \lambda_l$  — амплитуда и частота гармоники, найденной на  $l$ -м шаге. Раскроем скобки в последнем выражении и представим гармонику в форме Эйлера (4), как это сделали ранее:

$$\begin{aligned} A_k^{(l+1)} &= \sum_{i=0}^{N-1} S_i^{(l)} e^{-2\pi j \frac{ik}{NR}} - \sum_{i=0}^{N-1} \left( a_l \cos\left(\frac{2\pi i \lambda_l}{NR}\right) + b_l \sin\left(\frac{2\pi i \lambda_l}{NR}\right) \right) e^{-2\pi j \frac{ik}{NR}} = \\ &= A_k^{(l)} - \sum_{i=0}^{N-1} \left( \frac{a_l + j \cdot b_l}{2} \exp\left(-2\pi j \cdot \frac{i \lambda_l}{NR}\right) + \frac{a_l - j \cdot b_l}{2} \exp\left(2\pi j \cdot \frac{i \lambda_l}{NR}\right) \right) e^{-2\pi j \frac{ik}{NR}} = \\ &= A_k^{(l)} - \frac{a_l + j \cdot b_l}{2} \sum_{i=0}^{N-1} \exp\left(-2\pi j \cdot \frac{i(\lambda_l + k)}{NR}\right) - \frac{a_l - j \cdot b_l}{2} \sum_{i=0}^{N-1} \exp\left(2\pi j \cdot \frac{i(\lambda_l - k)}{NR}\right) = \\ &= A_k^{(l)} - \frac{a_l + j \cdot b_l}{2} Q_{-(\lambda_l + k)} - \frac{a_l - j \cdot b_l}{2} Q_{\lambda_l - k}. \end{aligned}$$

При таком способе вычислений в процессе вычитания гармоник при больших  $L$  из-за ошибок округления может возникнуть расхождение между сигналом  $S^{(l)}$  и его спектром  $A^{(l)}$ , поэтому при программировании алгоритма лучше использовать типы двойной точности. Чтобы ошибки не накапливались, можно периодически пересчитывать спектр напрямую с помощью БПФ.

**6. Вычисление суммы комплексных экспонент.** Для получения значений  $Q_k$  достаточно вычислить БПФ единичного сигнала. Можно также получить их аналитически. Преобразуем сумму  $Q_k$  следующим образом:

$$\begin{aligned} Q_k &= \sum_{i=0}^{N-1} e^{2\pi j \frac{ik}{NR}} = \sum_{i=0}^{N/2-1} \left( e^{2\pi j \frac{ik}{NR}} + e^{2\pi j \frac{(i+N/2)k}{NR}} \right) = \\ &= \sum_{i=0}^{N/2-1} e^{2\pi j \frac{ik}{NR}} \left( 1 + e^{2\pi j \frac{k}{2R}} \right) = \left( 1 + e^{2\pi j \frac{k}{2R}} \right) \sum_{i=0}^{N/2-1} e^{2\pi j \frac{ik}{NR}}. \end{aligned}$$

Аналогично перегруппируем последнюю сумму:

$$\sum_{i=0}^{N/2-1} e^{2\pi j \frac{ik}{NR}} = \left( 1 + e^{2\pi j \frac{k}{4R}} \right) \sum_{i=0}^{N/4-1} e^{2\pi j \frac{ik}{NR}}.$$

Выполняя то же действие над следующими суммами, получим в конечном итоге:

$$Q_k = \sum_{i=0}^{N-1} e^{2\pi j \frac{ik}{NR}} = \prod_{\eta=1}^{\log_2 N} \left( 1 + e^{2\pi j \frac{k}{2^\eta R}} \right).$$

**7. Алгоритм БПФ с высоким частотным разрешением.** На рис. 1 представлена блок-схема алгоритма БПФ, который будет использован в основном алгоритме для вычисления амплитуд  $A_k$  произвольного сигнала  $S$ . Данный алгоритм подробно описан в работе [14, с. 131]. Входные параметры подпрограммы БПФ\_ВР следующие: сигнал  $S$  длиной  $N$  ( $N$  обязательно должно быть степенью двойки), частотное разрешение  $R$  (целое положительное), направление  $d \in \{-1, 1\}$ ,  $\Delta k$  — частотный сдвиг. При  $d = 1$  выполняется прямое преобразование Фурье, при  $d = -1$  — обратное. Действие алгоритма заключается в вычислении элементов массива  $A$  по формуле:

$$A_k = \sum_{i=0}^{NR-1} S_i \exp\left(-j \cdot 2\pi d \cdot \frac{i(k + \Delta k)}{NR}\right), \quad k = 0, \dots, NR - 1. \quad (9)$$

Представленный на блок-схеме алгоритм нетрудно реализовать практически на любом языке программирования. В листинге 1 представлена такая реализация в виде функции `fft_hr` на языке Python. Алгоритм составлен таким образом, что можно полностью распараллелить по  $NR$  потокам внутренних

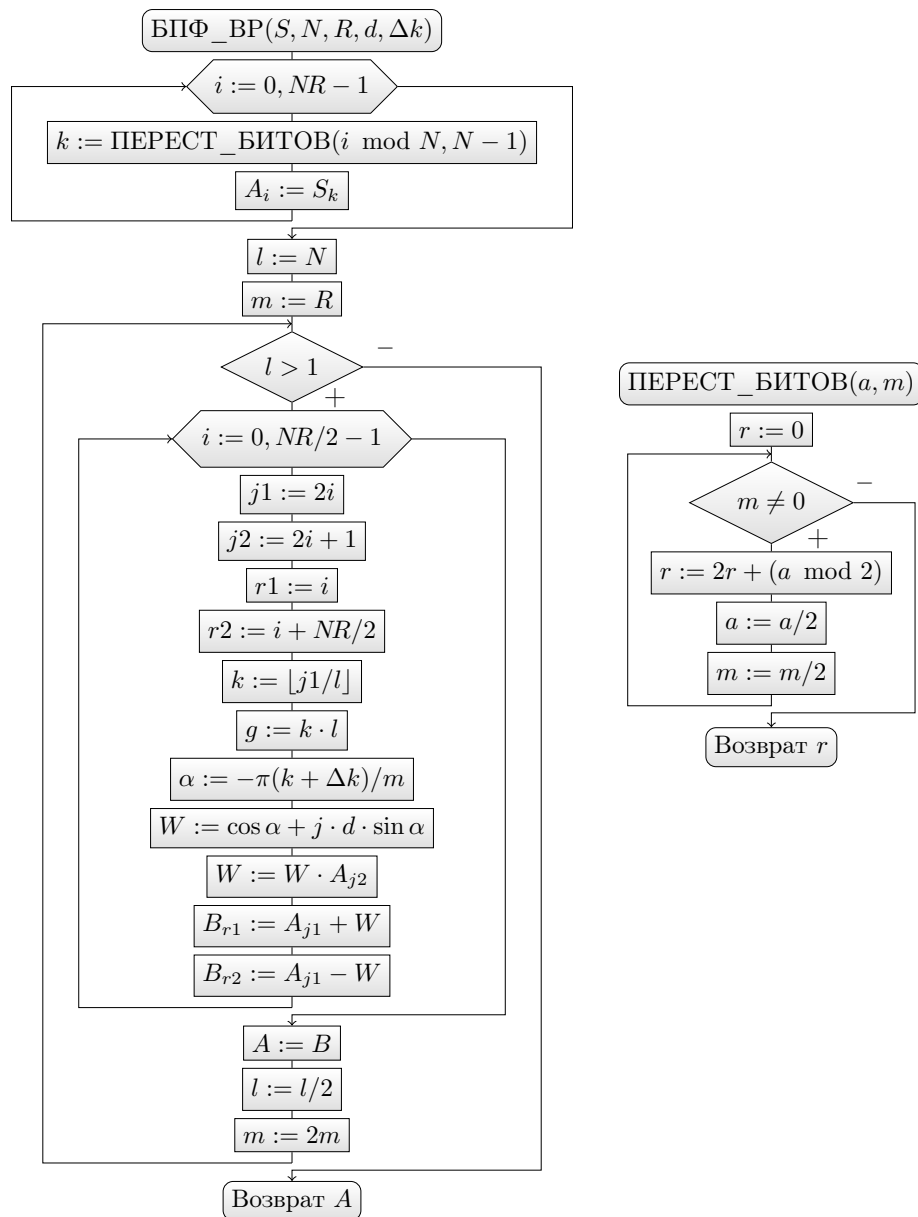


Рис. 1. Блок-схема алгоритма БПФ с высоким частотным разрешением  
 Fig. 1. Block diagram of the FFT algorithm with high frequency resolution



Листинг 1. Реализация алгоритма БПФ с высоким частотным разрешением на языке Python  
 Listing 1. Implementation of the FFT algorithm with high frequency resolution in Python

```

1  # Обратная перестановка битов
2  def inverse(a, m):
3      r = 0
4      while m:
5          r = 2 * r + a % 2
6          a //= 2
7          m //= 2
8      return r
9  # Быстрое преобразование Фурье
10 def fft_hr(S, R, d, dk=0):
11     N = len(S)
12     NR = N * R
13     A, B = [complex()] * NR, [complex()] * NR
14     for i in range(NR):
15         k = inverse(i % N, N - 1)
16         A[i] = S[k]
17     l = N
18     m = R
19     while l > 1:
20         for i in range(NR//2):
21             j1 = 2 * i
22             j2 = 2 * i + 1
23             r1 = i
24             r2 = i + NR//2
25             k = j1 // l
26             g = k * l
27             angle = -math.pi * (k + dk) / m
28             W = complex(math.cos(angle), d * math.sin(angle))
29             W = W * A[j2]
30             B[r1] = A[j1] + W # Бабочка
31             B[r2] = A[j1] - W
32     A = B.copy()
33     l //= 2
34     m *= 2
35     return A
    
```

цикл (строки с 20 по 31), потому что каждая итерация цикла обрабатывает “свои” элементы массивов  $A$  и  $B$ . Это видно по значениям, которые присваиваются переменным  $j1, j2, r1, r2$ . Подобное распараллеливание можно реализовать на языке OpenCL [15] с использованием барьеров для синхронизации переходов к следующим итерациям внешнего цикла (с предусловием, пока  $l > 1$ ).

Сопроцессор сравнительно медленно вычисляет значения тригонометрических функций ( $\sin, \cos$ ). Вторая возможная оптимизация (при условии, что  $\Delta k = 0$ ) заключается в подготовке массива  $W$ , в который заранее будут записаны значения поворачивающих множителей:

$$W_g = \cos(\alpha) + j \cdot d \cdot \sin(\alpha), \quad \alpha = -\pi g / (NR), \quad g = 0, \dots, NR - 1.$$

Тогда вычисление  $W$  (строка 28) нужно будет заменить на обращение к элементу массива  $W_g$ . Переменная  $g$  специально объявлена для индексации элементов такого массива.

Алгоритм **БПФ\_ВР**, как и многие другие алгоритмы БПФ, использует обратную перестановку битов, но только для инициализации массива  $A$ . Она выполняется путем  $R$ -кратного повторения и перетасовки значений исходного сигнала  $S$ . Для вычисления индексов элементов массива используется функция ПЕРЕСТ\_БИТОВ( $a, m$ ), которая возвращает обратную перестановку битов индекса  $a$  длиной  $\log_2(m + 1)$  битов. Временная сложность представленного алгоритма **БПФ\_ВР** равна  $O(NR \cdot \log_2 N)$ .



**8. Реализация основного алгоритма аппроксимации.** В соответствии с предложенным в начале статьи подходом, а также приведенными выражениями для решения задачи минимизации СКО и вычитания гармоник, построим основной алгоритм аппроксимации. На рис. 2 представлена блок-схема алгоритма, который аппроксимирует сигнал  $S$  длиной  $N$  полиномом из  $L$  убывающих по амплитуде гармоник. Поскольку значения  $Q_k$  повторяются с периодом  $NR$ , т.е.  $Q_k = Q_{k+nNR}$  для любых целых  $k$  и  $n$ , то в программе достаточно составить массив  $Q$  длиной  $NR$  только для частот от 0 до  $NR - 1$ . Поэтому в блок-схеме при вычислении индексов элементов этого массива используется операция сложения по

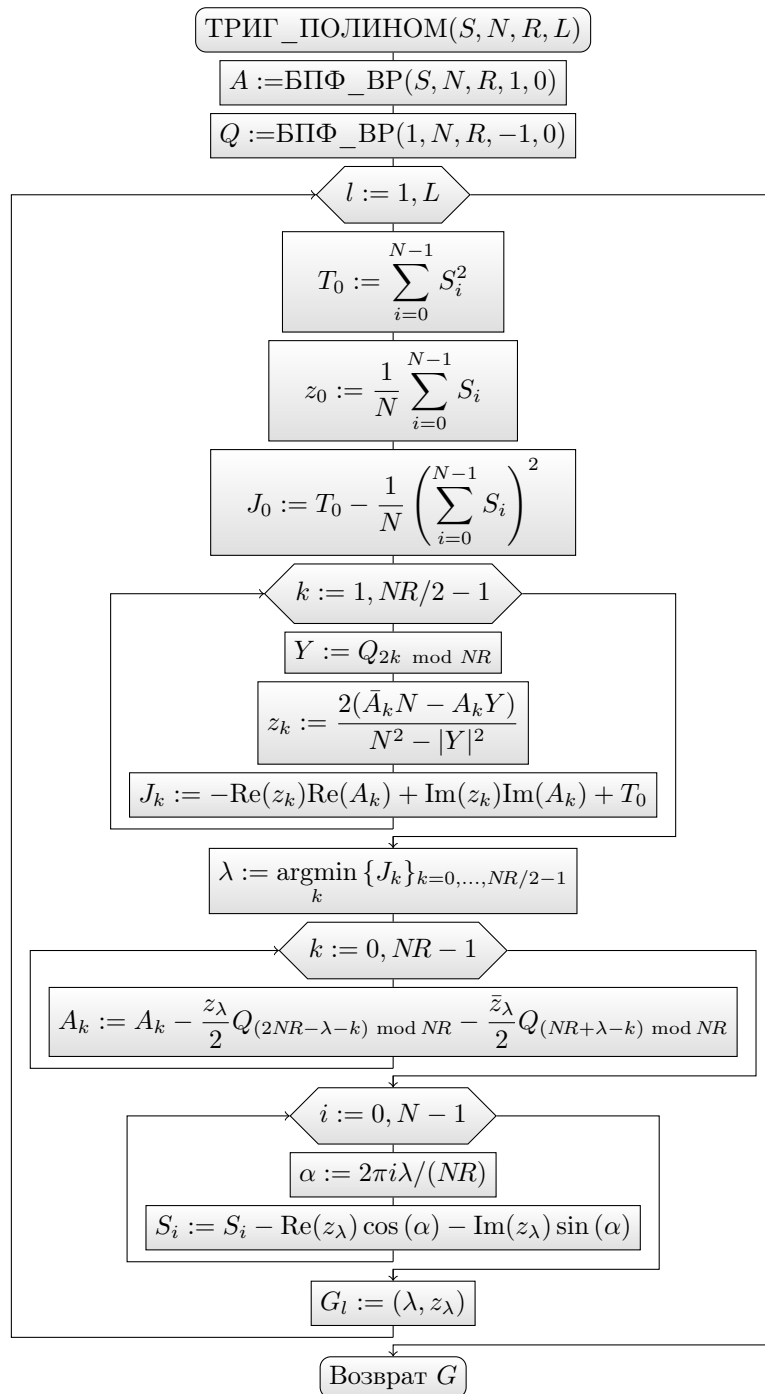


Рис. 2. Блок-схема основного алгоритма аппроксимации сигнала тригонометрическим полиномом  
 Fig. 2. Block diagram of the basic algorithm for signal approximation by a trigonometric polynomial



модулю  $NR$ . Сам массив  $Q$  при этом вычисляется как БПФ единичного сигнала. Входной параметр  $R$ , как и раньше, определяет длину частотного базиса и частотное разрешение БПФ. Результатом работы алгоритма является последовательность  $G$ , состоящая из  $L$  пар, в каждой из которых записана частота ( $\lambda$ ) и амплитуда ( $z_\lambda$ ) гармоники. Можно оптимизировать алгоритм, если обрабатывать только значащую часть спектра. Для этого нужно ограничить сверху индекс  $k$  в циклах до нужного значения. Например, если заранее известно, что искомые гармоники находятся в части спектра до 5000 Гц из 44100 Гц, то достаточно ограничиться частотой  $k_{\max} = 5000/44100 \cdot NR$ .

Рассмотрим кратко возможные модификации и улучшения представленного метода. Если исходный сигнал сильно смещен относительно оси абсцисс, то можно заранее перенести его таким образом, чтобы сумма всех отсчетов сигнала равнялась нулю. Это необязательно, потому что, если смещение большое, то алгоритм найдет его сам. Возможна также модификация алгоритма, в которой смещение относительно оси абсцисс корректируется на каждой итерации  $l$  таким образом, чтобы выполнялось равенство:  $\sum_{i=0}^{N-1} S_i^{(l)} = 0$ . Само смещение при этом накапливается как амплитуда гармоники с частотой  $k = 0$ . Но как показывают вычислительные эксперименты, такая модификация незначительно влияет на аппроксимацию. Можно на каждой  $l$ -ой итерации алгоритма дополнительно уточнять параметры найденной гармоники, в том числе частоту  $\lambda_l$ . Для этого в окрестности найденной точки  $G_l$  нужно найти минимум (3) функции  $F(x, y, k)$  по всем трем параметрам с помощью метода градиентного спуска.

В листинге 2 приведена программа на языке Python для аппроксимации произвольного дискретного сигнала  $S$  тригонометрическим полиномом  $G$  длиной  $L$ .

Листинг 2. Исходный текст подпрограммы для аппроксимации сигнала тригонометрическим полиномом  
 Listing 2. The source text of the subroutine for signal approximation by a trigonometric polynomial

```

1 def trigonometric_polynom(S, R, L):
2     N = len(S)
3     NR = N * R
4     G = []
5     A = fft_hr(S, R, 1) # БПФ сигнала S
6     Q = fft_hr([1] * N, R, -1) # БПФ единичного сигнала
7     for l in range(L):
8         T0 = numpy.sum(S**2)
9         Z = [complex()] * (NR//2)
10        J = [float()] * (NR//2)
11        Z[0] = numpy.sum(S) / N
12        J[0] = T0 - numpy.sum(S) ** 2 / N
13        # Вычисление СКО сигнала от всех простых гармоник
14        for k in range(1, NR//2):
15            Y = Q[2 * k % NR]
16            Z[k] = 2 * (N * A[k].conjugate() - Y * A[k])
17            Z[k] /= (N*N - Y.real*Y.real - Y.imag*Y.imag)
18            J[k] = -Z[k].real * A[k].real + Z[k].imag * A[k].imag + T0
19        h = numpy.argmin(J) # Частота самой сильной гармоники
20        # Пересчет спектра путем вычитания самой сильной гармоники из спектра
21        for k in range(NR):
22            A[k] -= Z[h]/2 * Q[(2*NR - h - k) % NR]
23            A[k] -= Z[h].conjugate()/2 * Q[(NR + h - k) % NR]
24        # Вычитание гармоники из сигнала
25        for i in range(N):
26            angle = 2 * math.pi * i * h / NR
27            S[i] -= Z[h].real * math.cos(angle)
28            S[i] -= Z[h].imag * math.sin(angle)
29        G.append([h, Z[h]])
30    return G
    
```

**9. Эксперимент 1.** Аппроксимируем искусственно образованный двумя гармониками сигнал:

$$S_i^{(0)} = 440 \sin(2\pi \cdot 400 \cdot i\Delta t) + 680 \sin(2\pi \cdot 710 \cdot i\Delta t), \quad \Delta t = 1/F, \quad N = 2048, \quad F = 44100 \text{ Гц.}$$

Частотное разрешение для экспериментов выберем равным  $R = 64$ . В листинге 3 представлен исходный текст программы для данного численного расчета.

Листинг 3. Исходный текст программы для аппроксимации искусственного сигнала с двумя гармониками  
 Listing 3. The source code of the program for the approximation of an artificial signal with two harmonics

```

1 N = 2048      # Количество отсчетов
2 F = 44100    # Частота дискретизации
3 R = 64      # Частотное разрешение
4 L = 2       # Количество искомых гармоник
5 t = numpy.linspace(0, (N - 1) * 1/F, N)
6 S = 440 * numpy.sin(2 * math.pi * 400.0 * t) + \
7     680 * numpy.sin(2 * math.pi * 710.0 * t)
8 G = trigonometric_polynom(S.copy(), R, L)
9 S2 = [float()] * N
10 for g in G:
11     v = g[0] * F / (N*R)
12     S2 += g[1].real * numpy.cos(2 * math.pi * v * t)
13     S2 += g[1].imag * numpy.sin(2 * math.pi * v * t)
14     print("%f*cos(2*%f*pi*%f*t)+%f*sin(2*%f*pi*%f*t)" \
15           % (g[1].real, v, g[1].imag, v))
16 fig, ax = plt.subplots()
17 ax.plot(t, S, t, S2)
18 ax.legend(['Исходный сигнал', 'Восстановленный сигнал'])
19 plt.show()
20 # Рисование спектрограммы аппроксимированного сигнала
21 w = numpy.linspace(0, F * (1 - 1 / (N*R)), N*R)
22 A = fft_hr(S2, R, 1)
23 fig2, ax2 = plt.subplots()
24 ax2.plot(w, 2 / N * numpy.abs(A))
25 ax2.legend(['Амплитудный спектр восстановленного сигнала'])
26 plt.xlim([0, 1000])
27 plt.show()
    
```

В результате аппроксимации сигнал был восстановлен до следующей формы:

$$\tilde{S}(t) = 1,728 \cos(2\pi \cdot 709,923 \cdot t) + 683,298 \sin(2\pi \cdot 709,923 \cdot t) - 2,997 \cos(2\pi \cdot 400,047 \cdot t) + 439,919 \sin(2\pi \cdot 400,047 \cdot t).$$

Продемонстрируем работу алгоритма на графиках. На рис. 3 представлен амплитудный спектр  $|A(\omega)|$  исходного сигнала  $S^{(0)}$  (верхний график), полученный с помощью БПФ (9). Показана часть спектра до 1000 Гц из 44100 рассчитанных. После первой итерации алгоритма сильная гармоника исключается из сигнала  $S^{(0)}$ , тем самым этот сигнал преобразуется в  $S^{(1)}$ . На среднем графике показана соответствующая сигналу  $S^{(1)}$  спектрограмма, на которой видно, что гармоника с частотой 709,923 Гц исчезла вместе с боковыми лепестками. Серым цветом закрашена удаленная часть спектра. После второй итерации алгоритма была исключена и вторая гармоника, что показано на нижнем графике. Видно, что остаточный сигнал  $S^{(2)} = S^{(0)}(t) - \tilde{S}(t)$  не превышает по амплитуде 8 единиц, а СКО сигнала  $S^{(0)}$  от сигнала  $\tilde{S}$  составляет  $\sigma = 5,85$ .

**10. Эксперимент 2.** Рассмотрим сигнал, состоящий из двух гармоник, частоты которых близки друг к другу:

$$S_i = 500 \sin(2\pi \cdot 550 \cdot i\Delta t) + 500 \sin(2\pi \cdot 575 \cdot i\Delta t), \quad \Delta t = 1/F, \quad N = 2048, \quad F = 44100 \text{ Гц.}$$

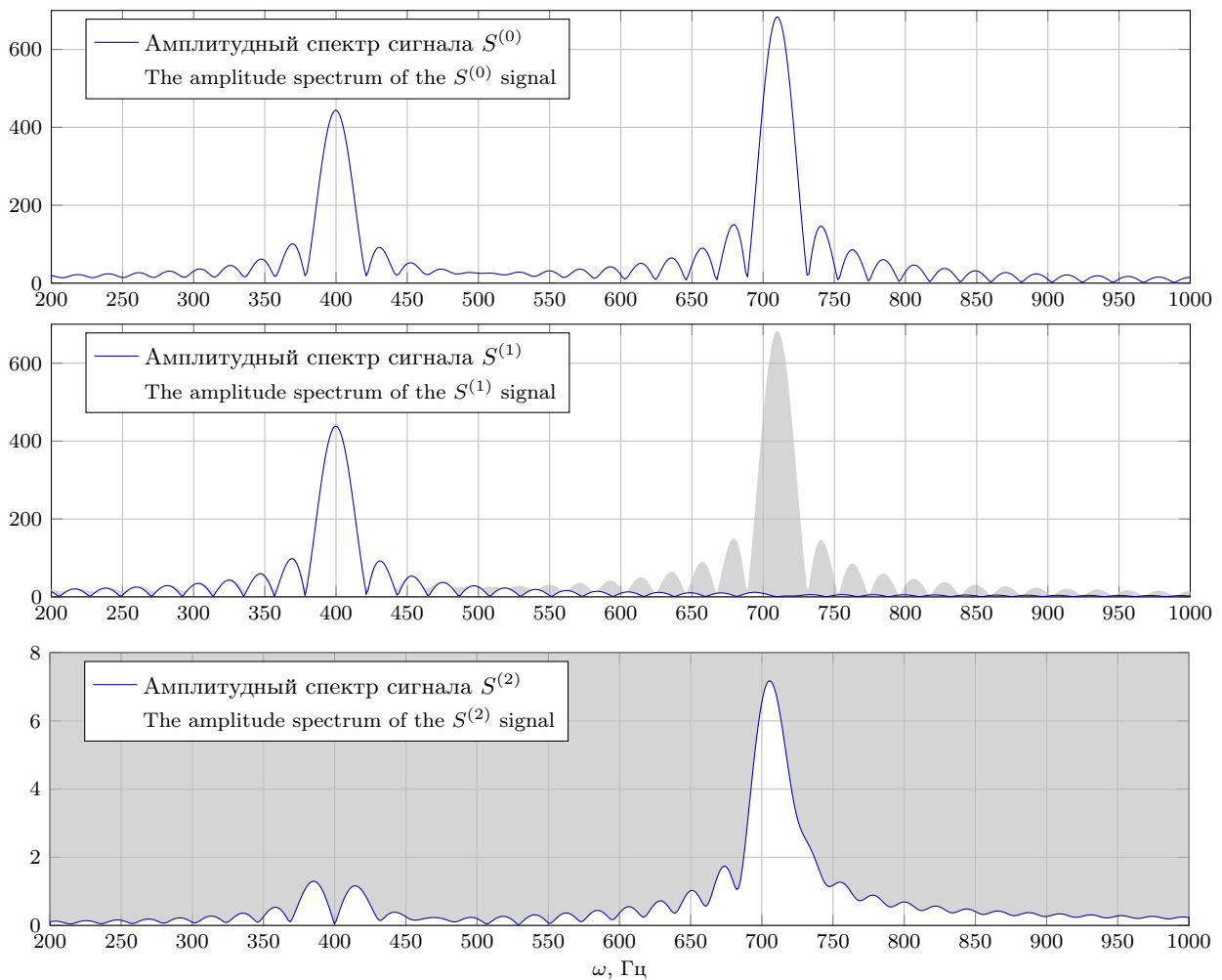


Рис. 3. Амплитудные спектры последовательности сигналов с разным числом гармоник

Fig. 3. Amplitude spectra of a sequence of signals with a different number of harmonics

При аппроксимации этого сигнала найденные частоты близки друг к другу, но не совсем совпадают с исходными (550 и 575 Гц):

$$\tilde{S}(t) = 572,365 \cos(2\pi \cdot 547,751 \cdot t - 66,585^\circ) + 462,607 \cos(2\pi \cdot 576,35 \cdot t - 100,276^\circ).$$

Это объясняется тем, что найденные частоты соответствуют пикам на спектрограмме исходного сигнала, которые изначально вычисляются неточно, потому что сигнал очень короткий. Если уменьшать в исходном сигнале  $S$  разность по частоте этих двух гармоник, то на спектрограмме они вовсе будут сливаться в одну и исходные частоты восстановить не удастся. Можно аппроксимировать данный сигнал  $S$  большим числом гармоник  $L$ , что хотя бы уменьшит СКО  $\sigma$  аппроксимированного сигнала от исходного сигнала  $S$ . В табл. 1 представлены значения  $\sigma$  с ростом числа гармоник  $L$ . В каждой строке табл. 1 выписаны параметры найденной на соответствующем шаге алгоритма гармоники.

Исходный текст программы для проведения данного эксперимента представлен в листинге 4. Для формирования таблицы в ней необходимо подключить пакет **tabulate**.

**11. Эксперимент 3.** На рис. 4 (верхний график) представлен фрагмент акустического сигнала из 16-битной звуковой дорожки с такими же параметрами дискретизации, как и в предыдущем эксперименте. На среднем графике на рис. 4 показано, какую форму имеет этот же восстановленный пятью гармониками сигнал. На нижнем графике сигнал аппроксимирован 15 гармониками.

На рис. 5 в логарифмическом масштабе показано, как изменяется СКО  $\sigma$  исходного акустического сигнала от аппроксимированного с ростом числа гармоник при различной длине частотного базиса. При

Таблица 1. Значения СКО  $\sigma$  и параметры гармоник с ростом их числа  
 Table 1. The standard deviation  $\sigma$  and the parameters of harmonics with an increase in their number

Количество гармоник $L$ Harmonics count $L$	СКО $\sigma$ SD $\sigma$	Частота $\omega$ , Гц Frequency $\omega$ , Hz	Амплитуда $ z $ Amplitude $ z $	Начальная фаза $\varphi$ , град. Phase $\varphi$ , deg.
1	337,087	547,751	572,365	66,585
2	82,691	576,350	462,607	100,276
3	37,465	541,358	104,081	179,089
4	20,848	560,536	44,049	-85,150
5	16,401	576,350	18,242	155,852
6	11,557	596,873	16,481	96,968
7	8,396	532,947	11,248	164,761
8	6,725	554,816	7,129	126,508
9	5,429	617,061	5,619	84,369
10	3,574	513,432	5,796	145,949
11	3,073	638,931	2,584	86,231
12	2,615	490,553	2,289	132,267
13	2,364	574,331	1,583	93,586
14	2,108	659,454	1,515	73,921
15	1,890	468,684	1,324	129,355
16	1,734	597,546	1,069	122,348
17	1,593	530,592	0,969	104,162
18	1,443	679,642	0,953	57,764
19	1,297	448,496	0,900	142,595
20	1,179	554,144	0,766	113,344

Листинг 4. Исходный текст программы для аппроксимации искусственного сигнала с двумя близкими по частоте гармониками

Listing 4. The source code of the program for approximating an artificial signal with two harmonics close in frequency

```

1 N = 2048      # Количество отсчетов
2 F = 44100    # Частота дискретизации
3 R = 64       # Частотное разрешение
4 L = 20       # Количество искоемых гармоник
5 t = numpy.linspace(0.0, (N - 1) * 1/F, N)
6 v1, v2 = 550, 575 # Частоты гармоник
7 S = 500 * numpy.sin(2 * math.pi * v1 * t)
8 S += 500 * numpy.sin(2 * math.pi * v2 * t)
9 G = trigonometric_polynom(S.copy(), R, L)
10 S2 = numpy.array([0]*N, dtype=numpy.double)
11 line, table = 1, []
12 for g in G:
13     v = g[0] * F / (N*R)
14     phase = math.atan2(g[1].imag, g[1].real)
15     amplitude = numpy.abs(g[1])
16     S2 = S2 + amplitude * numpy.cos(2 * math.pi * v * t - phase)
17     table.append([line, math.sqrt(numpy.sum((S - S2)**2)/N), \
18                 v, amplitude, math.degrees(phase)])
19     line += 1
20 print(tabulate(table, headers=['L', 'СКО', 'Частота', 'Амплитуда', 'Фаза'], \
21             tablefmt='orgtbl', floatfmt=".3f"))
    
```

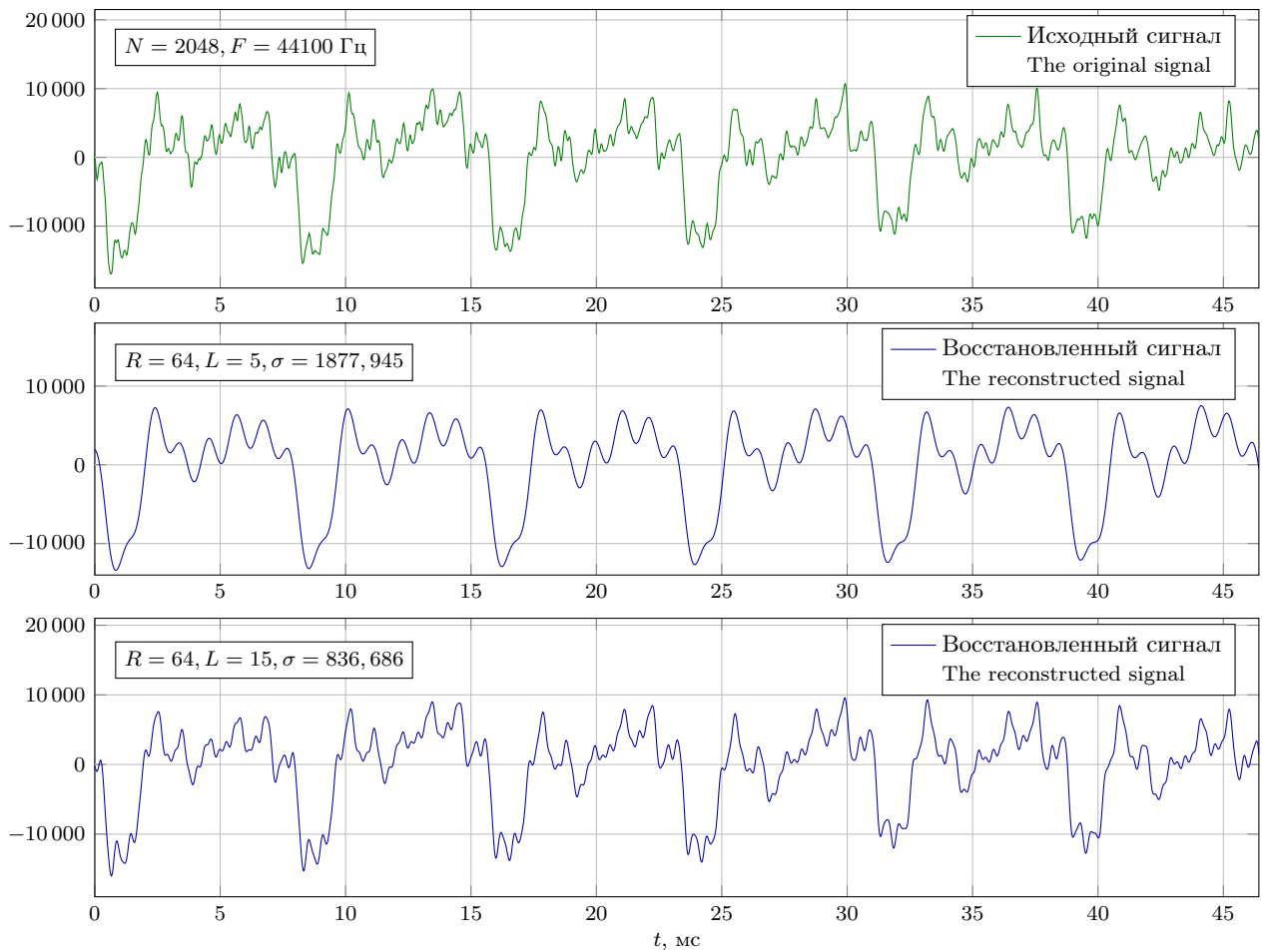


Рис. 4. Исходный сигнал и его аппроксимации с разным числом гармоник

Fig. 4. The original signal and its approximations with a different number of harmonics

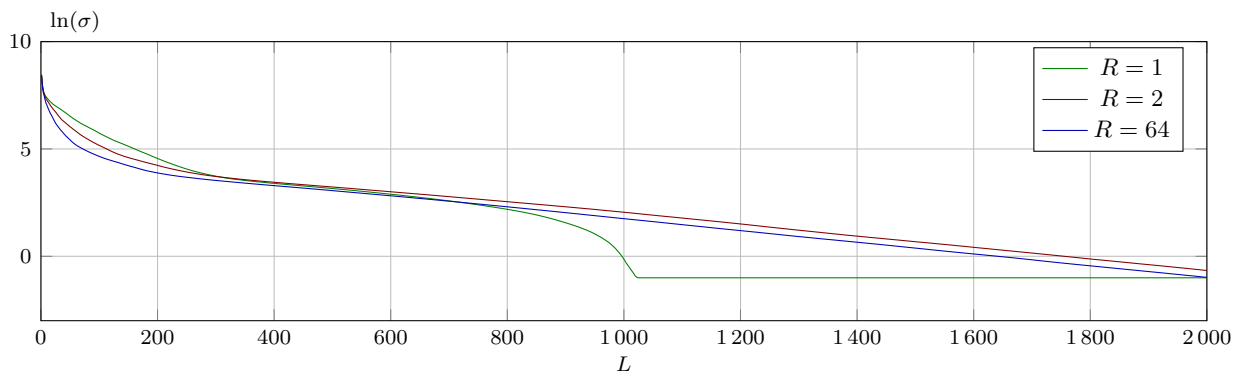


Рис. 5. СКО восстановленного сигнала от исходного с ростом числа гармоник  $L$

Fig. 5. The SD of the restored signal from the original one with an increase in the number of harmonics  $L$

$R = 1$  аппроксимация соответствует разложению Фурье. На графике видно, что при  $R = 64$  кривая  $\sigma(L)$  с ростом числа гармоник убывает все медленнее, пока не приобретает линейный характер. Сигнал хорошо аппроксимируется уже 200 гармониками. Дальнейшая аппроксимация включает незначащие гармоники с низкой амплитудой.

**12. Тригонометрическая аппроксимация в форме БПФ.** Рассмотрим отдельно одно обстоятельство. При исследовании действительных сигналов часто пользуются известными выражениями для

разложения функций в ряд Фурье [16, с. 59], стараясь избежать использования комплексных чисел. В работах [17–19] отмечено, что такие разложения приводят к появлению погрешностей. К тому же такое численное интегрирование трудоемко, так как требуется вычислять сумму ряда для каждой частоты отдельно. Т.е. такие преимущества БПФ как высокая скорость и малая временная сложность становятся недоступны в принципе. Заметим, что для действительного сигнала  $S$  с помощью БПФ может быть получена точная аппроксимация, которая не будет иметь мнимой части. Пусть

$$A_k = \sum_{i=0}^{N-1} S_i e^{-2\pi j \frac{ik}{NR}} \quad \text{и} \quad \text{Im}(S) = 0.$$

Такой спектр является симметричным относительно середины, потому что  $A_{NR/2+k} = \bar{A}_{NR/2-k}$  при  $1 < k < NR/2$ . Выполним обратное ДПФ [20, с. 277] и реконструируем сигнал, сгруппировав скобками отраженные части:

$$S_i = \frac{1}{NR} \sum_{k=0}^{NR-1} A_k e^{2\pi j \frac{ik}{NR}} = \frac{1}{NR} \left[ A_0 + (-1)^i A_{NR/2} + \sum_{k=1}^{NR/2-1} \left( A_{NR/2+k} e^{2\pi j \frac{i(NR/2+k)}{NR}} + A_{NR/2-k} e^{2\pi j \frac{i(NR/2-k)}{NR}} \right) \right].$$

Нетрудно убедиться, что  $e^{2\pi j \frac{i(NR/2+k)}{NR}} = e^{-2\pi j \frac{i(NR/2-k)}{NR}}$ , потому что  $e^{j(\pi+x)} = e^{-j(\pi-x)}$  для любого  $x$ . Используя известное выражение Эйлера (4), получим разложение сигнала  $S$  в тригонометрический ряд:

$$S_i = \frac{A_0 + (-1)^i A_{NR/2}}{NR} + \frac{2}{NR} \sum_{k=1}^{NR/2-1} \left( \text{Re}(A_k) \cos\left(2\pi \frac{ik}{NR}\right) - \text{Im}(A_k) \sin\left(2\pi \frac{ik}{NR}\right) \right).$$

Видно, что данная тригонометрическая аппроксимация не содержит мнимой единицы  $j$ .

**13. Анализ алгоритмической сложности.** Рассматриваемый в данной работе алгоритм перебирает на каждом шаге все  $NR$  гармоник и выбирает из них одну, которая больше всех минимизирует СКО аппроксимированного сигнала от исходного. Поскольку на каждой итерации ищется только одна гармоника, итерация алгоритма имеет линейную алгоритмическую сложность  $O(NR)$ . Общая сложность аппроксимации, не включая БПФ, равна  $O(LNR)$ , где  $L$  — общее количество гармоник, которые нужно найти. Аппроксимация была бы точнее, если бы мы искали на каждом шаге не одну, а две гармоники. Тогда нужно было бы перебрать все пары гармоник и найти такую, которая сильнее всех минимизирует СКО. Для отдельно взятой пары частот  $k_1, k_2$  минимизируемый функционал в таком случае будет иметь следующий вид:

$$F(x_1, y_1, x_2, y_2) = \sum_{i=0}^{N-1} (S_i - x_1 \cos(ik_1\theta) - y_1 \sin(ik_1\theta) - x_2 \cos(ik_2\theta) - y_2 \sin(ik_2\theta))^2 \rightarrow \min.$$

Дифференцируя данный функционал по всем неизвестным  $x_1, y_1, x_2, y_2$ , нетрудно свести задачу к решению двух линейных уравнений:

$$\begin{aligned} 2\bar{A}_{k_1} &= z_1 Q_0 + \bar{z}_1 Q_{2k_1} + z_2 Q_{k_1-k_2} + \bar{z}_2 Q_{k_1+k_2}, \\ 2\bar{A}_{k_2} &= z_1 Q_{k_2-k_1} + \bar{z}_1 Q_{k_1+k_2} + z_2 Q_0 + \bar{z}_2 Q_{2k_2}, \\ z_1 &= x_1 + j \cdot y_1, \quad z_2 = x_2 + j \cdot y_2. \end{aligned}$$

Поскольку при таком способе аппроксимации нужно перебирать все пары гармоник, на каждой итерации нужно будет решить  $(NR)^2$  таких уравнений. Следовательно, сложность одной итерации будет равна  $O((NR)^2)$ . Такая большая алгоритмическая сложность скорее всего будет неоправданной, но зато такой алгоритм будет точнее распознавать две близкие по частоте гармоники. Например, в эксперименте 2, если разность между частотами гармоник будет равна  $k_1 - k_2 = 1$  и  $k_1, k_2$  будут попадать в частотный базис, то алгоритм обнаружит их абсолютно точно, т.е. минимизирует СКО до нуля. При этом даже на



спектрограмме гармоник с частотами  $k_1, k_2$  изначально будут неразличимы. Можно дальше улучшать точность и ухудшать сложность — перебирать гармоники тройками, четверками и т.д. сочетаниями из  $NR$  по  $n$ . Тогда сложность будет стремиться к  $O((NR)^n)$ . Это наталкивает на мысль о том, что задача аппроксимации сигнала наименьшим количеством гармоник при заданном максимально допустимом СКО  $\sigma_{\max}$  и фиксированном частотном базисе является экспоненциально сложной и, скорее всего, даже  $NP$ -полной.

**14. Заключение.** Разработан метод, позволяющий получить приемлемую аппроксимацию дискретного сигнала тригонометрическим полиномом с убывающими по амплитуде гармониками за линейное время. С помощью него можно изучать естественные сигналы различной природы (акустические, электротехнические, радиосигналы и др.), поддающиеся спектральному анализу. Метод позволяет достаточно точно определить несущую частоту сигнала и получить его гармонические составляющие в различных прикладных задачах [21, 22]. Поскольку в основе метода лежит идея вычитания гармоник, то в задачах аудиоредактирования его можно использовать для удаления отдельных гармоник и обертонов из сигнала. Численные эксперименты показывают, что с помощью предложенных аппроксимаций можно искусственно синтезировать различные звуки, распознавать ноты, удалять цифровые водяные знаки и разрабатывать алгоритмы сжатия аудиоданных. Полученные в работе результаты позволяют улучшить отечественное программное и аппаратное обеспечение для цифровой обработки сигналов. Рекомендуется включение представленных методов в учебные программы инженерных специальностей.

### Список литературы

1. Сергиенко А.Б. Цифровая обработка сигналов. СПб.: Питер, 2002.
2. Кошелева Д.Д., Доронина А.В. Преобразование Фурье и быстрое преобразование Фурье // Инновации. Наука. Образование. 2021. № 38. 626–632.
3. Бахурин С.А. Алгоритм БПФ с прореживанием по времени. [https://ru.dsplib.org/content/fft\\_dec\\_in\\_time/fft\\_dec\\_in\\_time.html](https://ru.dsplib.org/content/fft_dec_in_time/fft_dec_in_time.html). (Дата обращения: 19 июля 2024).
4. Сикорский С.П., Иванчуков Е.В. Процесс цифровой обработки сигналов и преобразование Фурье при эффекте растекания спектра // Журнал научных и прикладных исследований. 2016. 1, № 11. 67–69.
5. Попов Н.М., Серов А.Н. Сравнительный анализ методов снижения эффекта “растекания спектра” // Автоматизация и измерения в машино-приборостроении. 2019. № 3. 48–53.
6. Дмитриев Е.В. Гармонические дискретные спектры и аппроксимация коротких процессов, сигналов, функций // Авиакосмическое приборостроение. 2006. № 3. 23–31.
7. Дмитриев Е.В. Обнаружение и различение сигналов в их аддитивной смеси путем расчета и анализа ее естественного спектра // Физика волновых процессов и радиотехнические системы. 2008. 11, № 2. 61–66. <https://elibrary.ru/item.asp?id=12931786>. (Дата обращения: 19 июля 2024).
8. Гетманов В.Г. Об уменьшении времени решения задачи дискретной тригонометрической аппроксимации // Автометрия. 1988. № 6. 108–110.
9. Гетманов В.Г., Дабагян Р.А., Сидоров Р.В. Исследование характеристик геомагнитных пульсаций методом локальных аппроксимаций // Геомагнетизм и аэрономия. 2016. 56, № 2. 209–216. doi 10.7868/S001679401602005X.
10. Абалакин В.К., Аксенов Е.П., Гребеников Е.А., Демин В.Г., Рябов Ю.А. Справочное руководство по небесной механике и астродинамике. М.: Наука, 1976.
11. Марпл С.Л. Цифровой спектральный анализ и его приложения. М.: Мир, 1990.
12. Граничин О.Н., Павленко Д.В. Рандомизация данных и  $L_1$ -оптимизация // Компьютерные инструменты в образовании. 2010. № 1. 4–13.
13. Князев Б.А., Черненький В.М. Сверточное разреженное представление изображений для анализа статических и динамических образов // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. 2014. № 11. 664–695. doi 10.7463/1114.0730860.
14. Осипов О.В. Итерационные алгоритмы БПФ с высоким частотным разрешением // Вычислительные методы и программирование. 2021. 22, № 2. 123–137. doi 10.26089/NumMet.v22r209.
15. Бурцев А.А. Оптимизация операции быстрого преобразования Фурье в среде OpenCL // Труды Института Системных Исследований РАН. 2022. 12, № 1–2. 11–27. doi 10.25682/NIISI.2022.1-2.0002.
16. Гетманов В.Г. Цифровая обработка сигналов. М.: НИЯУ МИФИ. 2010.



17. Поршнев С.В., Кусайкин Д.В. О точности тригонометрической интерполяции // Цифровая обработка сигналов. 2019. № 1. 43–51. <https://elibrary.ru/item.asp?id=41269684>. (Дата обращения: 19 июля 2024).
18. Зиятдинов С.И., Гурин Н.В. Анализ ошибок при тригонометрической интерполяции // Известия высших учебных заведений. Приборостроение. 2008. 51, № 5. 42–45.
19. Ефимов В.М., Резник А.Л. Об отсчетных функциях при восстановлении периодического сигнала и дисперсии ошибки тригонометрической интерполяции // Автометрия. 2005. 41, № 4. 3–14.
20. Осипов О.В. Спектральный анализ дискретных сигналов с высоким частотным разрешением // Вычислительные методы и программирование. 2019. 20, № 3. 270–282. doi 10.26089/NumMet.v20r324.
21. Васеева Т.В., Альтман Е.А. Алгоритм нахождения оптимальной несмещенной оценки амплитуды гармоники на основе метода корреляционных функций // Вестн. Ростовского гос. унив. путей сообщения. 2022. № 1. 168–176. doi 10.46973/0201-727X\_2022\_1\_168.
22. Чижма С.Н., Газизов Р.И. Метод спектрального анализа сигналов в системах электроснабжения // Омский научный вестник. 2013. № 1. 198–201.

Поступила в редакцию  
19 мая 2024 г.

Принята к публикации  
18 июля 2024 г.

### Информация об авторе

Олег Васильевич Осипов — к.ф.-м.н., доцент; Белгородский государственный технологический университет имени В. Г. Шухова, кафедра программного обеспечения вычислительной техники и автоматизированных систем, ул. Костюкова, 46, 308012, Белгород, Российская Федерация.

### References

1. A. B. Sergienko, *Digital Signal Processing* (Piter, St. Petersburg, 2002) [in Russian].
2. D. D. Kosheleva and A. V. Doronina, “Fourier Transform and Fast Fourier Transform,” *Innovatsii. Nauka. Obrazovanie*. No. 38, 626–632 (2021).
3. S. A. Bakhurin, “Time-Domain Decimation FFT Algorithm,” [https://ru.dsplib.org/content/fft\\_dec\\_in\\_time/fft\\_dec\\_in\\_time.html](https://ru.dsplib.org/content/fft_dec_in_time/fft_dec_in_time.html). Cited July 19, 2024.
4. S. P. Sikorskiy and E. V. Ivanchukov, “Digital Signal Processing and Fourier Transform in the Case of Spectral Spreading,” *Zh. Nauchnykh i Prikladnykh Issledovaniy*. 1 (11), 67–69 (2016).
5. N. M. Popov and A. N. Serov, “Comparative Analysis of Methods for Reducing the ‘Spectral Spreading’ Effect,” *Avtomatizatsiya Izmereniya v Mashinostroyenii*, No. 3, 48–53 (2019).
6. E. V. Dmitriev, “Harmonic Discrete Spectra and Approximation of Short Processes, Signals, and Functions,” *Aviakosmicheskoe Priborostroenie*, No. 3, 23–31 (2006).
7. E. V. Dmitriev, “Detection and Discrimination of Signals in Additive Mixture by Calculation and Analysis of Natural Spectrum,” *Fizika Volnovykh Protsessov i Radiotekhnicheskie Sistemy*. 11 (2), 61–66 (2008). <https://elibrary.ru/item.asp?id=12931786>. Cited July 19, 2024.
8. V. G. Getmanov, “On Reducing the Solution Time of the Discrete Trigonometric Approximation Problem,” *Avtometriya*, No. 6, 108–110 (1988).
9. V. G. Getmanov, R. A. Dabagyan, and R. V. Sidorov, “Studying Geomagnetic Pulsation Characteristics with the Local Approximation Method,” *Geomagnetism i Aeronomiya*, 56 (2), 209–216 (2016) [*Geomagn. Aeron.* 56 (2), 195–202 (2016)]. doi 10.1134/S0016793216020055.
10. V. K. Abalakin, E. P. Aksyonov, E. A. Grebenikov, et al., *Handbook of Celestial Mechanics and Astrodynamics* (Nauka, Moscow, 1976) [in Russian].
11. S. L. Marple, *Digital Spectral Analysis with Applications* (Prentice-Hall, Englewood Cliffs, 1987; Mir, Moscow, 1990).
12. O. N. Granichin and D. V. Pavlenko, “Data Randomization and  $L_1$ -Optimization,” *Komp’yuternye Instrumenty v Obrazovanii*, No. 1, 4–13 (2010).
13. B. A. Knyazev and V. M. Chernen’kiy, “Convolutional Sparse Coding for Static and Dynamic Images Analysis,” *Nauka i Obrazovanie*, No. 11, 664–695 (2014). doi 10.7463/1114.0730860.
14. O. V. Osipov, “Iterative FFT-Algorithms with High Frequency Resolution,” *Numerical Methods and Programming (Vychislitel’nye Metody i Programirovanie)*. 22 (2), 123–137 (2021). doi 10.26089/NumMet.v22r209.



15. A. A. Burtsev, “Optimization of Fast Fourier Transform Operation in OpenCL Environment,” Trudy Inst. Sistemnykh Issledovaniy RAN. **12** (1–2), 11–27 (2022). doi 10.25682/NIISI.2022.1-2.0002.
16. V. G. Getmanov, *Digital Processing of Signals* (NIYAU MPhI, Moscow, 2010).
17. S. V. Porshnev and D. V. Kusaikin, “On Accuracy of Trigonometric Interpolation,” Digital Signal Processing (Tsifrovaya Obrabotka Signalov), No. 1, 43–51 (2019). <https://elibrary.ru/item.asp?id=41269684>. Cited July 19, 2024.
18. S. I. Ziatdinov and N. V. Girin, “Analysis of Errors in Trigonometric Interpolation,” Izv. Vyssh. Uchebn. Zaved., Priborostroen. **51** (5), 42–45 (2008).
19. V. M. Efimov and A. L. Reznik, “Sampling Functions in Periodical Signal Reconstruction. The Variance of a Trigonometric Interpolation Error,” Avtometriya **41** (4), 3–14 (2005). [Optoelectron., Instrum. Data Process. **41** (4), 3–13 (2005)]. <https://sibran.ru/en/journals/issue.php?ID=120300>. Cited July 20, 2024.
20. O. V. Osipov, “Spectral Analysis of Discrete Signals with High Frequency Resolution,” Numerical Methods and Programming (Vychislitel’nye Metody i Programmirovaniye). **20** (3), 270–282 (2019). doi 10.26089/NumMet.v20r324.
21. T. V. Vaseeva and E. A. Altman, “Algorithm for Finding the Optimum Unbias Estimate of the Amplitude Harmonics Based on the Method of Correlation Functions,” Vestn. Rostov Gos. Univ. Putey Soobshchen., No. 1, 168–176 (2022). doi 10.46973/0201-727X\_2022\_1\_168.
22. S. N. Chizhma and R. I. Gazizov, “Method of Spectral Analysis of Signals in Power Supply Systems,” Omskiy Nauch. Vestn., No. 1, 198–201 (2013).

Received  
May 19, 2024

Accepted for publication  
July 18, 2024

#### Information about the author

Oleg V. Osipov — Ph.D., Associate Professor; Shukhov Belgorod State Technological University, ulitsa Kostyukova, 46, 308012, Belgorod, Russia.