Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Курсовой проект по курсу Численные методы
по теме: "Аппроксимация функций с использованием вейвлет-анализа"

Студентка: Е. Н. Безлуцкая
Преподаватель: Д. Л. Ревизников
Группа: М8О-307Б

Дата:
Оценка:
Подпись:

Москва, 2019

Постановка задачи

Взять некоторую периодическую функцию и аппроксимировать с помощью дискретного вейвлет-преобразования(ДВП). Получить, используя прямое ДВП, коэффициенты аппроксимации и коэффициенты детализации. Затем провести обратное преобразование, получить исходную функцию. Обнаружить локальные характеристики функции.

Описание

Введение

Известно, что преобразование Фурье обладает существенным недостатком, а именно неспособностью отслеживать частотно-временные характеристики сигнала с течением времени. Чтобы этот недостаток устранить, в качестве анализирующей функции нужно выбирать не гармоническую, растянутую по всей числовой оси, а некую функцию, локализованную на этой оси. Так, вейвлет-преобразования основаны на разложении по малым волнам, называемым вейвлетами, изменяющейся частоты и ограниченным во времени (в пространстве).

Кратномасштабное разложение

Функцию (сигнал) $f(x)$ часто проще анализировать, если представить ее в виде линейной комбинации функций из некоторой системы функций разложения $\{\phi(x)\}$

$$f(x) = \sum_k \alpha_k \phi_k(x), \tag{1}$$

где k – конечное или бесконечное множество целых значений, вещественные числа α_k – *коэффициенты разложения*, ϕ_k принимают вещественные значения и называются *функциями разложения*. Если разложение единственно, то функции $\phi_k(x)$ называются базисными функциями, а все множество функций разложения $\{\phi(x)\}$ называется базисом в том классе функций, которые могут быть представлены таким образом. Представимые в виде (1) функции образуют *пространство функций*

$$V = \overline{\underset{k}{Span}\{\phi(x)\}}$$

,которое называется замыканием линейной оболочки функций $\{\phi(x)\}$ или пространством, натянутым на систему функций $\{\phi(x)\}$.

Рассмотрим систему функций разложения, которая состоит из целочисленных сдвигов и двоичных изменений масштаба некоторой заданной вещественной квадратичноинтегрируемой функции $\phi(x)$. Таким образом, мы рассматриваем систему функций $\{\phi_{j,k}(x)\}$ вида

$$\phi_{j,k} = 2^{j/2}\phi(2^j x - k), \tag{2}$$

где $j,k \in \mathbb{Z}$ и $\phi(x) \in L^2(\mathbb{R})$. Здесь k определяет положение функции $\phi_{j,k}(x)$ на оси x, индекс j — ширину функции $\phi_{j,k}(x)$ вдоль оси x, а множитель $2^{j/2}$ регулирует высоту (амплитуду) функции. Функция $\phi(x)$ называется *масштабирующей функцией.*

Если мы ограничимся рассмотрением какого-нибудь одного фиксированного значения j в (2), скажем, $j = j_0$, то получаемая в результате система функций разложения $\phi_{j_0,k}$ будет подмножеством всей системы функций $\phi_{j,k}$. Тогда определим подпространство для $L^2(\mathbb{R})$ выражением

$$V_{j_0} = \overline{\underset{k}{Span}\{\phi(x)\}}. \tag{3}$$

Если $f(x) \in V_{j_0}$, то она может быть заисана в виде

$$f(x) = \sum_k \alpha_k \phi_{j_0,k}(x), \tag{4}$$

Часто используется, например, масштабирующая функция Хаара:

$$\phi(x) = \begin{cases} 1, & 0 \le x < 1 \\ 0, & \text{в остальных случаях.} \end{cases} \tag{5}$$

Если задана масштабирующая функция, удовлетворяющая КМА-условиям:

1. Масштабирующая функция и ее целые сдвиги ортогональны

2. Подпространства, натянутые на систему масштабирующих функций при низком разрешении (в мелком масштабе), содержатся в подпространствах, натянутых на систему масштабирующих функций при более высоком разрешении (в более крупном масштабе)

3. Единственной функцией, принадлежащей одновременно всем подпространствам V_j , является функция $f(x) = 0$

4. Любая функция может быть представлена с произвольной точностью

, то можно определить такую *вейвлет-функцию* $\psi(x)$(материнский вейвлет), что система функций, состоящих из целых сдвигов и двоичных изменений масштаба функции $\psi(x)$, порождает разность между двумя смежными КМА-подпространствами V_j и V_{j+1}. Определим систему вейвлетов

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k), \tag{6}$$

и каждое из пространств W_j оказывается натянутым на подсистему $\{\psi_{j,k}(x)\}$ при $k \in \mathbb{Z}$. Как и ранее, мы записываем

$$W_j = \overline{\underset{k}{Span}\{\psi_{j,k}(x)\}}. \tag{7}$$

и если $f(x) \in W_j$, то

$$f(x) = \sum_k \alpha_k \psi_{j,k}(x), \tag{8}$$

Подпространства, порождаемые масштабирующей функцией и вейвлет-функцией, связаны между собой соотношением

$$V_{j+1} = V_j + W_j, \tag{9}$$

Часто используется, например, вейвлет-функция Хаара:

$$\psi(x) = \begin{cases} 1, & 0 \le x < 0.5; \\ -1, & 0.5 \le x < 1; \\ 0, & \text{в остальных случаях.} \end{cases} \tag{10}$$

Дискретное вейвлет-преобразование

Подобно разложению в ряд Фурье разложение в вейвлет-ряд ставит в соответствие функции непрерывного аргумента некоторую последовательность коэффициентов. В том случае, когда подлежащая разложению функция является дискретной (т. е. последовательностью чисел), получаемая последовательность коэффициентов называется *дискретным вейвлет-преобразованием* функции f(x). Например, если $f(n) = f(x_0 + n\Delta x)$ для некоторых значений x_0, Δx и $n = 0, 1, 2, ..., M-1$ коэффициенты разложения $f(x)$ в вейвлет-ряд становятся коэффициентами *прямого* ДВП последовательности $f(n)$:

$$W_\phi(j_0, k) = \frac{1}{\sqrt{M}} \sum_n f(n) \phi_{j_0,k}(n), \tag{11}$$

$$W_\psi(j, k) = \frac{1}{\sqrt{M}} \sum_n f(n) \psi_{j,k}(n) \quad \text{для} \quad j \ge j_0. \tag{12}$$

В формулах выше
$\phi_{j_0,k}(n)$ и $\psi_{j,k}(n)$ – дискретные версии базисных функций $\phi_{j_0,k}(x)$ и $\psi_{j,k}(x)$
$W_\phi(j_0, k)$ – коэффициенты приближения
$W_\psi(j, k)$ – коэффициенты деталей

Дополнением к прямому будет обратное ДВП вида

$$f(n) = \frac{1}{\sqrt{M}} \sum_n W_\phi(j_0, k) \phi_{j_0,k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{+\infty} \sum_k W_\psi(j, k) \psi_{j,k}(n). \tag{13}$$

Обычно полагают $j_0 = 0$ и выбирают число M так, чтобы оно было степенью двойки (т.е. $M = 2^J$); при этом суммирование в уравнениях (11)—(13) производится по значениям $n = 0, 1, 2, ..., M-1$, $j = 0, 1, 2, ..., J-1$ и $k = 0, 1, 2, ..., 2^J - 1$. Для системы

Хаара дискретные аналоги масштабирующих функций и вейвлет-функций (т.е. базисных функций), участвующих в преобразовании, соответствуют строкам $M \times M$ матрицы преобразования Хаара $\mathbf{H}$.

Матрица $\mathbf{H}$ состоит из из базисных функций Хаара $h_k(z)$. Эти функции определены на непрерывном замкнутом интервале $z \in [0, 1]$ при $k = 0, 1, 2, ..., N - 1$, где $N = 2^n$. Чтобы получить $\mathbf{H}$, зададим целое k такое, что $k = 2^p + q - 1$,

где $0 \leq p \leq n - 1$,

$$q = \begin{cases} 0, 1 & \text{при } l = 0 \\ 1 \leq q \leq 2^p & \text{при } l \neq 0 \end{cases}$$

Тогда *базисные функции Хаара* будут

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, \quad z \in [0, 1]$$

и

$$h_k(z) = h_{pq}(z) = \frac{2^{p/2}}{\sqrt{N}} \begin{cases} 1 & \text{при } (q-1)/2^p \leq z \leq (q-0.5)/2^p \\ -1 & \text{при } (q-0.5)/2^p \leq z \leq q/2^p \\ 0 & \text{в остальных случаях, } z \in [0, 1] \end{cases}$$

Само преобразование состоит из M коэффициентов, минимальный масштаб равен нулю, а максимальный равен $J - 1$.

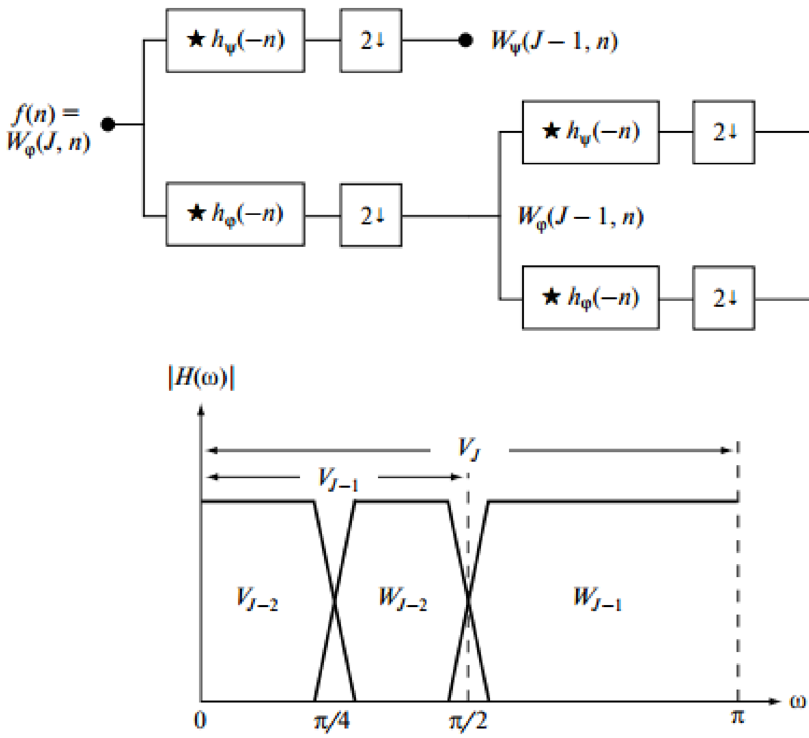
Быстрое вейвлет-преобразование

Быстрое вейвлет-преобразование (БВП) представляет собой эффективный метод реализации вычислений дискретного вейвлет-преобразования (ДВП), который использует взаимосвязь между коэффициентами ДВП соседних масштабов. Метод БВП называют также иерархическим алгоритмом Малла.

ДВП сигнала x получают применением набора фильтров. Сначала сигнал пропускается через низкочастотный (low-pass) фильтр. Одновременно сигнал раскладывается с помощью высокочастотного (high-pass) фильтра. В результате получаются детализирующие коэффициенты (после ВЧ-фильтра) и коэффициенты аппроксимации (после НЧ-фильтра).

Это разложение можно повторить несколько раз для дальнейшего увеличения частотного разрешения с дальнейшим прореживанием коэффициентов после НЧ и ВЧ-фильтрации. Это можно представить в виде двоичного дерева, где листья и узлы соответствуют пространствам с различной частотно-временной локализацией. Это дерево представляет структуру банка (гребёнки) фильтров.

Рис. 1: Двухступенчатый или двухмасштабный БВП-банк анализа и его свойства в отношении разделения частот



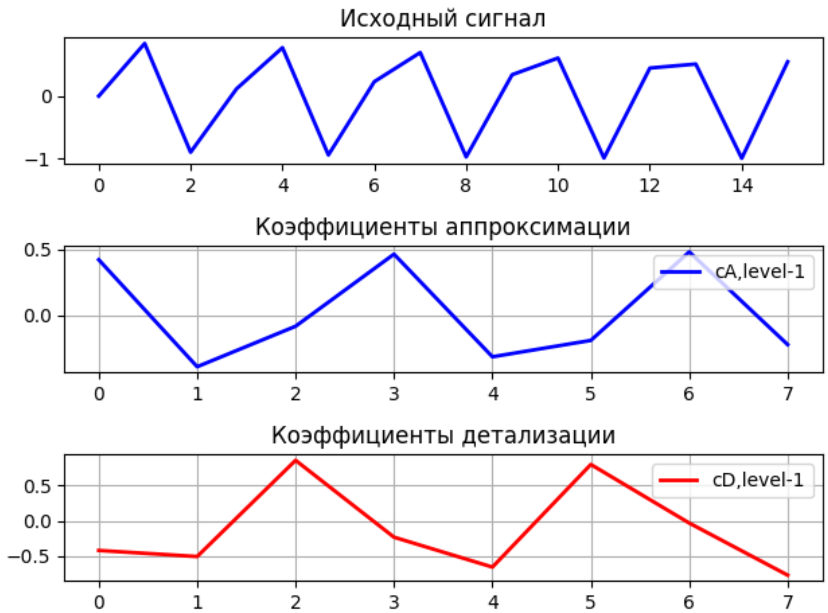
Результаты

Прямое ДВП

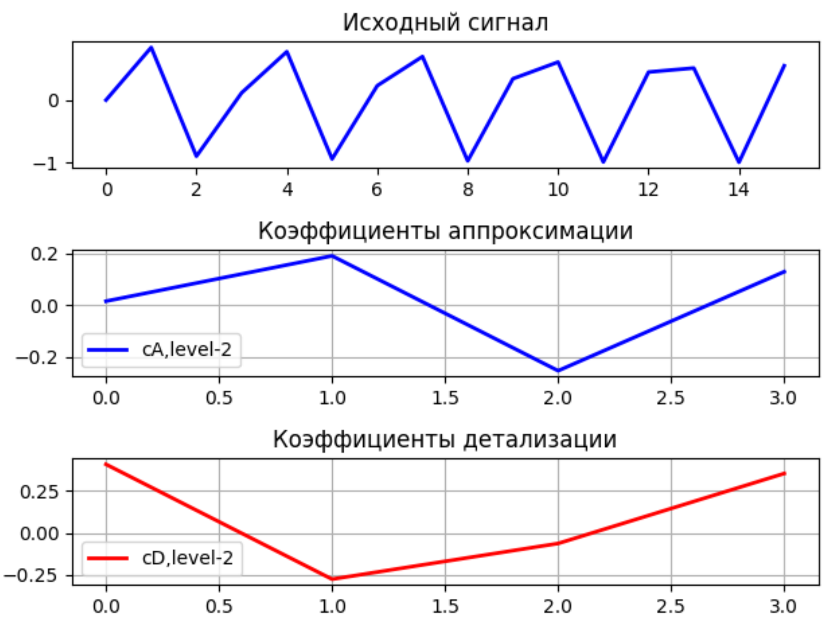
В качестве сигнала используется таблично заданная периодическая функция.

Для начала возьмем функцию со стационарным частотным спектром $f(x) = sin(2x)$ с масштабом $J = 4$, тогда число значений функции $M = 2^J = 16$

Разложение(уровень1)



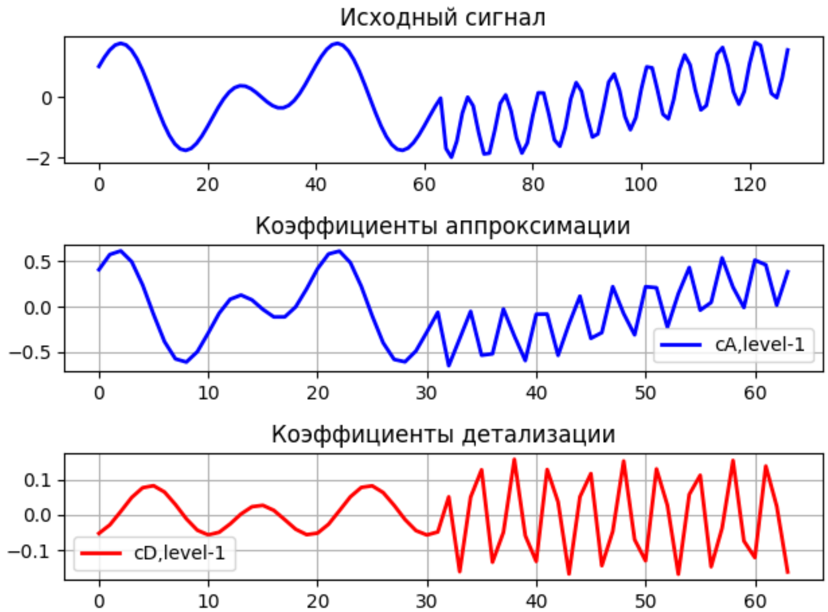
Разложение(уровень2)



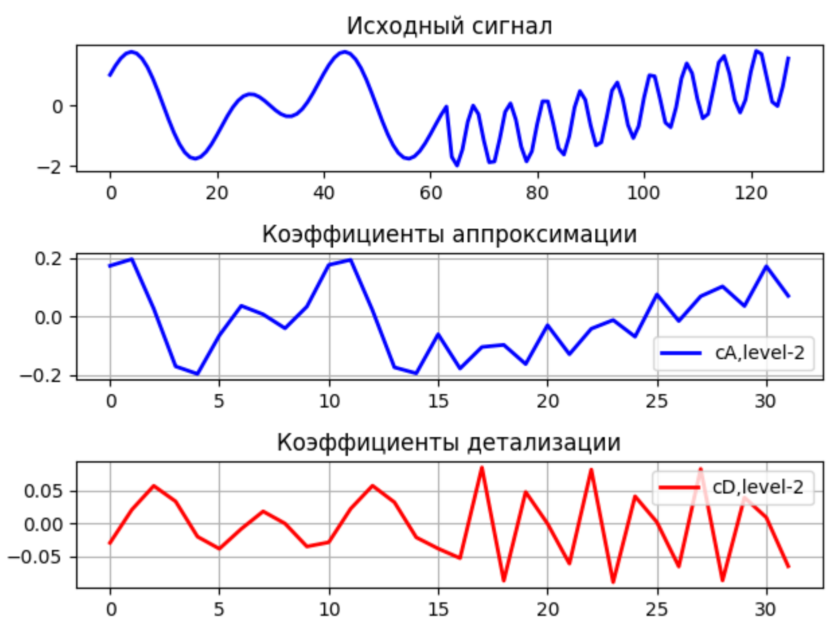
Дальнейшее разложение до максимального уровня(4) не имеет смысла, так как малое количество коэффициентов нам не даст никакой интересной информации. Для изучения особенностей вейвлет-преобразования нужно взять сигнал с нестационарным частотным спектром. Например, где, начиная с определенного x, частота функции начинается меняться. В данном примере для наглядности возьмем масштаб больше – $J = 7$. Возьмем в диапазоне x от 0 до 20 сигнал:

$$f(x) = \begin{cases} cos(x) + sin(2x) & \text{, при } x \leq 10 \\ cos(x) + sin(6x) & \text{, иначе.} \end{cases}$$

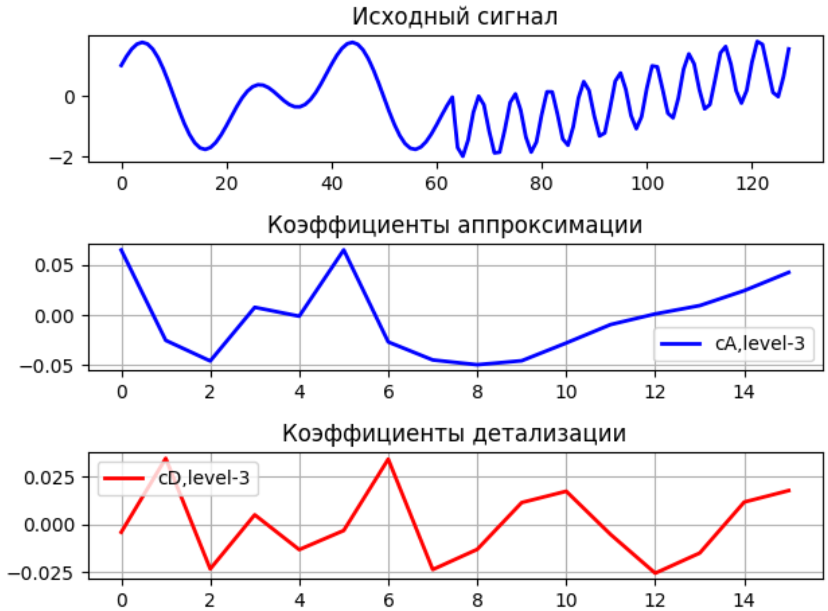
Разложение(уровень1)



Разложение(уровень2)



Разложение(уровень3)

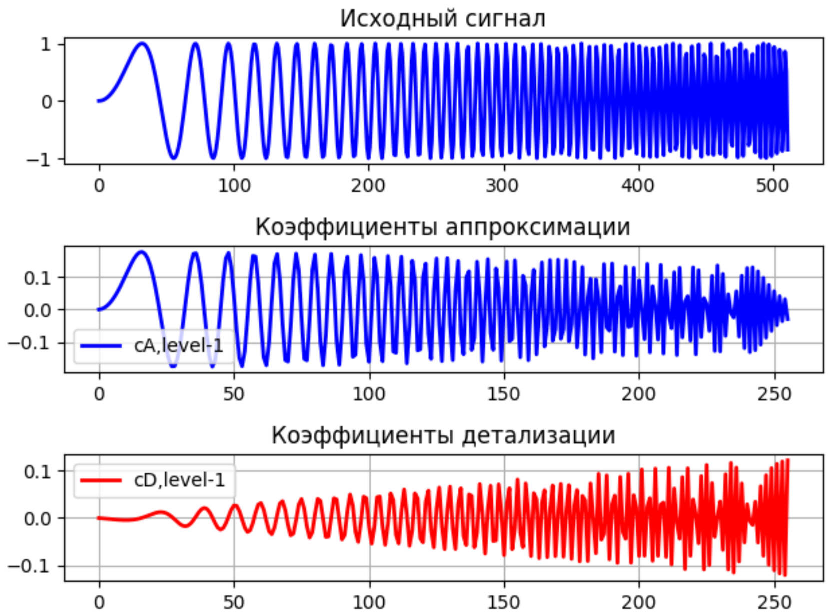


На первых уровнях разложения видно, что ДВП хорошо «отлавливает» спектральные характеристики. Это позволяет нам легко определить, в какой части сигнала произошло изменение частоты спектра.

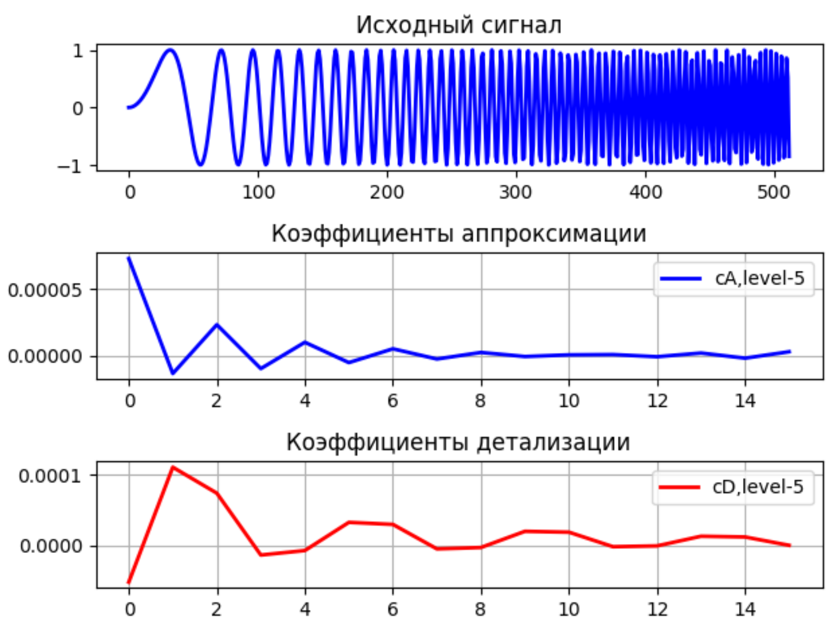
В следующем примере будем использовать сигнал с динамическим частотным спектром, который со временем увеличивается.

$f(x) = sin(x^2)$

Разложение(уровень1)

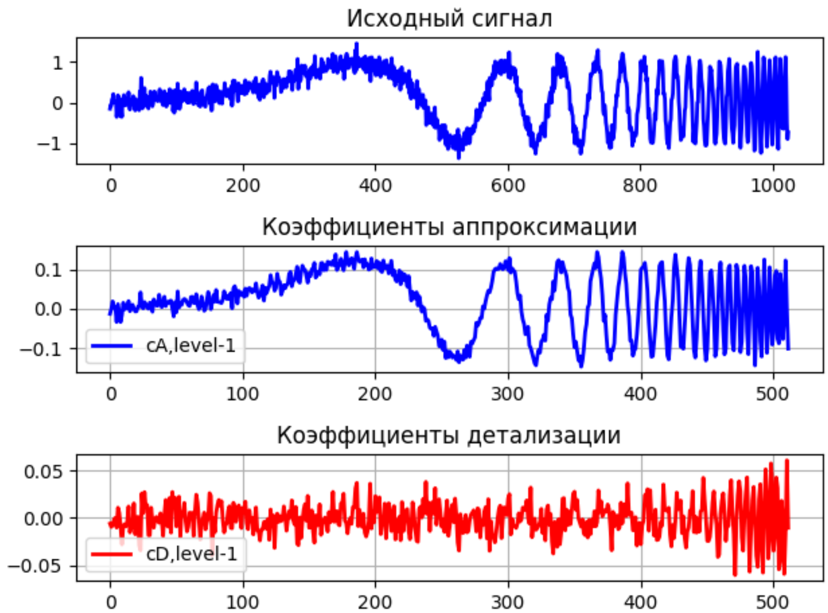


Разложение(уровень5)

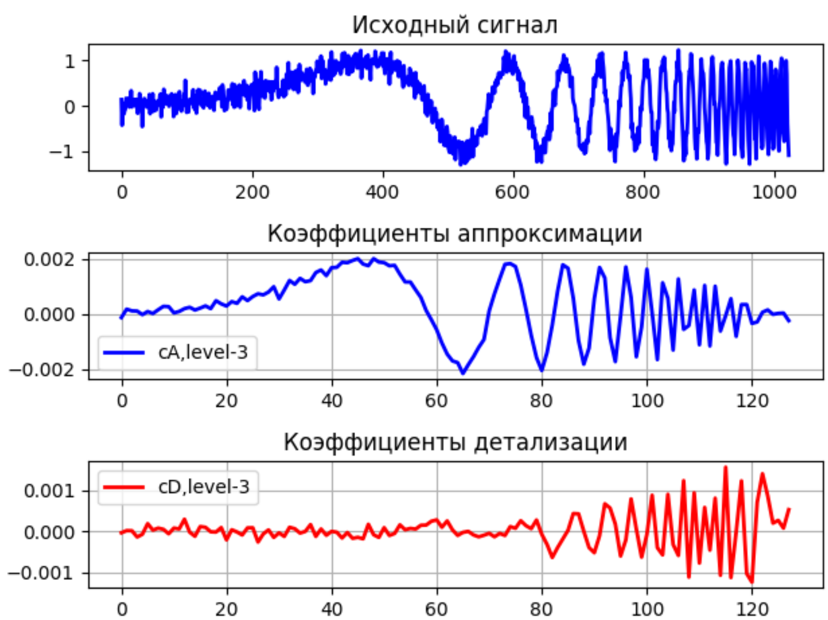


Можно заметить, что начиная с определеленного уровня, функция начинает «сглаживаться» в области высоких частот. Это подводит нас к тому, что с помощью вейвлет-преобразования можно удалять из сигнала шум.

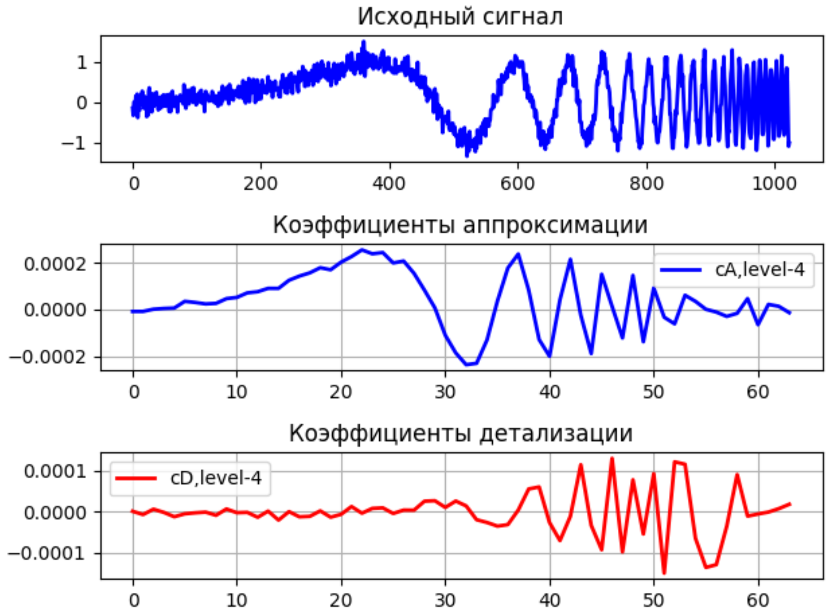
Разложение(уровень1)



Разложение(уровень3)



Разложение(уровень4)



При масштабе, равном десяти, уже на уровне три-четыре заметно удаление шума.

Обратное ДВП

Для демонстрации обратного ДВП возьмем простой короткий сигнал ($J = 3$) и получим исходную функцию из коэффициентов разного уровня:

```
Signal: [1, 2, 3, 4, 5, 6, 7, 8]
Coefficients in level = 1:  [array([ 2.121,   4.95 ,   7.778,  10.607]),
array([-0.707, -0.707, -0.707, -0.707])]
Signal by coefficients:  [ 1.  2.  3.  4.  5.  6.  7.  8.]

Coefficients in level = 2:  [array([  5.,  13.]), array([-2., -2.]),
array([-0.707, -0.707, -0.707, -0.707])]
Signal by coefficients:  [ 1.  2.  3.  4.  5.  6.  7.  8.]

Coefficients in level = 3:  [array([ 12.728]), array([-5.657]),
 array([-2., -2.]), array([-0.707, -0.707, -0.707, -0.707])]
Signal by coefficients:  [ 1.  2.  3.  4.  5.  6.  7.  8.]
```

Исходный код

```
import matplotlib.pyplot as plt
import numpy as np

SCALE = 3
N = 2 ** SCALE

def f1(x):
    return np.sin(2*x)

def f2(x):
    if x <= 10:
        return np.cos(x) + np.sin(2*x)
    else:
        return np.cos(x*0.3) + np.sin(6*x)

def f3(x):
    return np.sin(x**2)

def f4():
    return np.sin(2*np.pi*(2**np.linspace(2,10,N))*np.arange(N)/48000) + np.
    random.normal(0, 1, N) * 0.15

def check_level(level):
    max_level = int(np.log2(N))
    if level is None:
        return max_level
    elif level < 0:
        raise ValueError(
            "Level value of %d is too low . Minimum level is 0." % level)
    elif level > max_level:
        print(("Level value of {0} is too high. The maximum level value of {1}
     will be used.").format(level, max_level))
        return max_level
    return level

def dwt(data):
    size = len(data) // 2
    cA = np.zeros(size)
    cD = np.zeros(size)

    for i, j in zip(range(0, len(data), 2), range(size)):
        c = 2 * (data[i] + data[i + 1]) / np.sqrt(N)
        cA[j] = c

    for i, j in zip(range(0, len(data), 2), range(size)):
        c = 2 * (data[i] - data[i + 1]) / np.sqrt(N)
        cD[j] = c

    return cA, cD
```

```python

def wavedec(data, level=None):
    '''
    Multilevel 1D Discrete Wavelet Transform of data.

    Parameters
    ----------
    data: array_like
        Input data
    level : int, optional
        Decomposition level (must be >= 0). If level is None (default) then it
        will be calculated using the `dwt_max_level` function.

    Returns
    -------
    [cA_n, cD_n, cD_n-1, ..., cD2, cD1] : list
        Ordered list of coefficients arrays
        where `n` denotes the level of decomposition. The first element
        (`cA_n`) of the result is approximation coefficients array and the
        following elements (`cD_n` - `cD_1`) are details coefficients arrays.
    '''

    coeffs_list = []

    level = check_level(level)
    if level == 0:
        return [np.array(data)]
    a = data
    for i in range(level):
        a, d = dwt(a)
        coeffs_list.append(d)

    coeffs_list.append(a)
    coeffs_list.reverse()

    return coeffs_list

def idwt(a, d):
    res = []
    for i in range(len(a)):
        x = (a[i] + d[i]) * np.sqrt(N) / 4
        y = (a[i] - d[i]) * np.sqrt(N) / 4
        res.extend([x, y])
    return np.array(res)

def waverec(coeffs):
    """
    Multilevel 1D Inverse Discrete Wavelet Transform.

    Parameters
    ----------
```

```
    coeffs : array_like
        Coefficients list [cAn, cDn, cDn-1, ..., cD2, cD1]
    """
    if len(coeffs) < 1:
        raise ValueError(
            "Coefficient list too short (minimum 1 arrays required).")
    elif len(coeffs) == 1:
        # level 0 transform (just returns the approximation coefficients)
        return coeffs[0]

    a, ds = coeffs[0], coeffs[1:]

    for d in ds:
        a = idwt(a, d)

    return a
```

Выводы

Тема вейвлет-преобразований далась мне нелегло. Необходимо заложить в себе прочные математические основы для понимания той роли, которую играют вейвлет-методы и кратномасштабный анализ в обработке различных сигналов. В связи с этим некоторые вопросы так и остались у меня без ответа. Например, как выбрать подходящий вейвлет для исходного сигнала; почему при малых сжатиях вейвлет-преобразование уступает по качеству в сравнении с оконным Фурье-преобразованием. Доступных материалов на эту тему не так много, возможно, причиной является то, вейвлеты и вейвлет-преобразования являются сравнительно новыми средствами обработки данных.

Несмотря на сложность метода, вейвлет-преобразование широко используется для анализа сигналов. Помимо этого, оно находит большое применение в области сжатия данных. В чужих примерах мне удалось пронаблюдать сжатие изображений, удивительно, как просто это делается с помощью небольших python-скриптов. Также я познакомилась с библиотекой PyWavelets, на которую можно было ориентироваться при столкновении со сложностями в своем алгоритме разложения сигнала.

В дополнение к проделанной работе, мне бы хотелось познакомиться ближе с преобразованием Фурье и сжать самостоятельно несколько изображений. Но это уже выходящая за рамки данной работы история, которую исследовать я буду самостоятельно.

Список используемых источников

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Издание 3-е, исправленное и дополненное. Москва: Техносфера, 2012

2. `https://en.wikipedia.org/wiki/Discrete_wavelet_transform`