



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Учебное пособие

Учебно-методический комплект
по дисциплине
«Цифровая обработка сигналов»

Конспект лекции
«Быстрое преобразование Фурье»

В.В. Леонидов

1 Введение

На прошлой лекции мы изучили, что такое дискретное преобразование Фурье (ДПФ), изучили его свойства, подводные камни, способы улучшения результатов, а также частотно-временное преобразование Фурье.

Вспомним выражение для ДПФ:

$$X[m] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi nm/N} \quad (1)$$

Из него видно, что для расчёта N -точечного ДПФ требуется N^2 вычислений с комплексными числами. Получается ресурсозатратно, особенно если мы имеем дело с большим количеством отсчётов.

Учёным не нравилось делать много вычислений, и несколько человек (существуют разные мнения, кто был первым) предложили алгоритмы, позволяющие вычислять ДПФ, производя меньшее количество математических операций. Называются такие алгоритмы **Быстрое преобразование Фурье (БПФ)**.

Хочу сразу отметить, что БПФ возвращает абсолютно такие же результаты, что и ДПФ, использует в основе ту же формулу, только позволяет достичь этих результатов гораздо быстрее.

В рамках данной лекции рассмотрим два самых распространённых алгоритма БПФ – с прореживанием по времени и с прореживанием по частоте.

Итак, выше мы уже отметили, что для N -точечного ДПФ требуется N^2 операций комплексного умножения и сложения. А что, если разделить исходный сигнал на два длиной $N/2$ отсчётов каждый? Тогда получается, что для вычисления ДПФ от одной из половин потребуется $(N/2)^2 = N^2/4$ математических операций. Или $N^2/2$ для двух половин (если не считать процедуру разделения и объединения). Получается, эффективность вычислений возрастает в два раза! Но это ещё не предел. Мы можем делить наш сигнал до тех пор, пока не получим набор из сигналов, состоящих из двух отсчётов (рисунок 1).

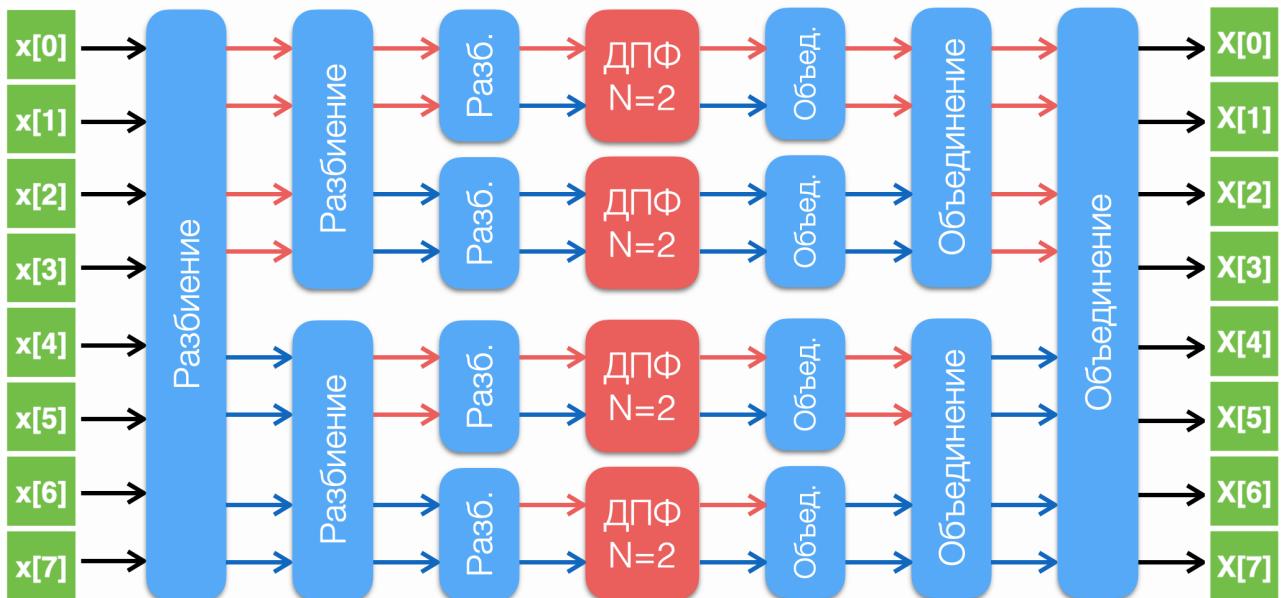


Рис. 1 – Разбиение сигнала до $N=2$

Теперь давайте немного модифицируем выражение (1). Пусть $W_N^{nm} = e^{-j2\pi nm/N}$, тогда:

$$X[m] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nm} \quad (2)$$

На примере сигнала из $N = 8$ отсчётов рассчитаем несколько значений W_N^{nm} :

$$\begin{aligned} W_8^0 &= e^0 = 1 \\ W_8^1 &= e^{-j2\pi/8} = e^{-j\pi/4} = \frac{1-j}{\sqrt{2}} = a \\ W_8^2 &= a^2 = \left(\frac{1-j}{\sqrt{2}}\right)^2 = \frac{1-2j+j^2}{2} = -j \\ W_8^3 &= a^3 = a^2 \cdot a = -a^* \\ W_8^4 &= a^4 = (a^2)^2 = -1 \\ W_8^5 &= a^5 = a^4 \cdot a = -a \\ W_8^6 &= a^6 = a^4 \cdot a^2 = j \\ W_8^7 &= a^7 = a^4 \cdot a^3 = a^* \\ W_8^8 &= a^8 = a^4 \cdot a^4 = 1 \end{aligned}$$

Можно заметить, что чётные и нечётные значения W_N^{nm} отличаются только знаком:

$$\begin{aligned} W_8^0 &= -W_8^4 \\ W_8^1 &= -W_8^5 \\ W_8^2 &= -W_8^6 \\ W_8^3 &= -W_8^7 \end{aligned}$$

Если произведение nm выходит за диапазон от 0 до 7, мы также увидим повторяемость этих значений.

Например, при $n = 6$, а $m = 7$:

$$W_8^{42} = a^{42} = (a^8)^5 \cdot a^2 = a^2 = -j$$

Следовательно, в процессе ДПФ вычисление одних и тех же значений повторяется несколько раз. Наша задача – оптимизировать процесс так, чтобы избежать выполнение повторяющихся операций.

2 Алгоритм БПФ с прореживанием по времени

Давайте разделим сигнал на отсчёты с чётными и нечётными номерами, как показано на рисунке 2. Эта операция называется **прореживание по времени**. Математически это можно записать следующим образом:

$$X[m] = \sum_{n=0}^{N/2-1} x[2n] \cdot W_N^{2nm} + \sum_{n=0}^{N/2-1} x[2n+1] \cdot W_N^{(2n+1)m} \quad (3)$$

Отметим, что:

$$W_N^{2nm} = e^{-j2\pi 2nm/N} = e^{-\frac{-j2\pi nm}{N/2}} = W_{\frac{N}{2}}^{nm} \quad (4)$$

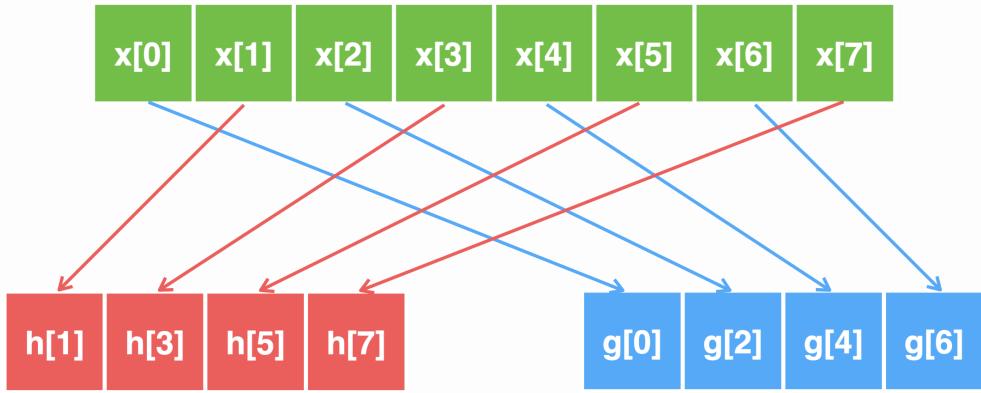


Рис. 2 – Прореживание сигнала по времени

$$W_N^{(2n+1)m} = W_N^{2nm} \cdot W_N^m = W_{\frac{N}{2}}^{nm} \cdot W_N^m \quad (5)$$

Тогда, выражение (3) принимает вид:

$$X[m] = \sum_{n=0}^{N/2-1} x[2n] \cdot W_{\frac{N}{2}}^{nm} + W_N^m \sum_{n=0}^{N/2-1} x[2n+1] \cdot W_{\frac{N}{2}}^{nm} \quad (6)$$

Это был расчёт ДПФ для первых $N/2$ отсчётов. Теперь запишем подобное выражение для второй половины отсчётов:

$$X \left[m + \frac{N}{2} \right] = \sum_{n=0}^{N/2-1} x[2n] \cdot W_N^{2n(m+\frac{N}{2})} + \sum_{n=0}^{N/2-1} x[2n+1] \cdot W_N^{(2n+1)(m+\frac{N}{2})} \quad (7)$$

Давайте теперь по очереди разбираться со всеми множителями:

$$W_N^{2n(m+\frac{N}{2})} = W_N^{2nm} \cdot W_N^{2n\frac{N}{2}} = W_{\frac{N}{2}}^{nm} \cdot e^{-j2\pi n \frac{N}{2}} = W_{\frac{N}{2}}^{nm} \cdot e^{-j2\pi n} \quad (8)$$

Т.к. $e^{-j2\pi n} = 1$, то выражение (8) принимает вид:

$$W_N^{2n(m+\frac{N}{2})} = W_{\frac{N}{2}}^{nm} \quad (9)$$

Рассмотрим следующий множитель:

$$W_N^{(2n+1)(m+\frac{N}{2})} = W_N^{2nm} \cdot W_N^{2nN/2} \cdot W_N^m \cdot W_N^{N/2} \quad (10)$$

Т.к.

$$W_N^{N/2} = e^{-j2\pi N} = e^{-j\pi} = -1, \quad (11)$$

то выражение (10) можно упростить:

$$W_N^{(2n+1)(m+\frac{N}{2})} = -W_{\frac{N}{2}}^{nm} \cdot W_N^m \quad (12)$$

Подставим результаты из (9) и (12) в уравнение (7) и получим:

$$X \left[m + \frac{N}{2} \right] = \sum_{n=0}^{N/2-1} x[2n] \cdot W_{\frac{N}{2}}^{nm} - W_N^m \sum_{n=0}^{N/2-1} x[2n+1] \cdot W_{\frac{N}{2}}^{nm} \quad (13)$$

Для простоты восприятия сделаем две замены:

$$G[m] = \sum_{n=0}^{N/2-1} x[2n] \cdot W_N^{nm} \quad (14)$$

$$H[m] = \sum_{n=0}^{N/2-1} x[2n+1] \cdot W_N^{nm} \quad (15)$$

Тогда, с учётом (14) и (15), сгруппируем уравнения (6) и (13):

$$\begin{aligned} X[m] &= G[m] + W_N^m \cdot H[m] \\ X\left[m + \frac{N}{2}\right] &= G[m] - W_N^m \cdot H[m] \end{aligned} \quad (16)$$

Полученное выражение (16) называется **графом «бабочка»**. Почему? Это станет ясно, если посмотреть на его графическое представление (рисунок 3). Данное выражение лежит в основе БПФ с прореживанием по времени.

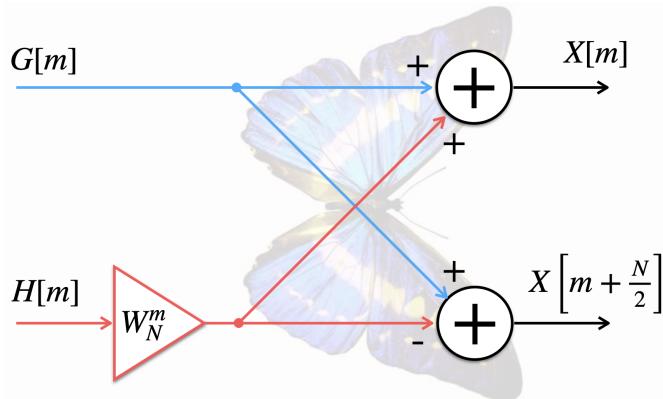
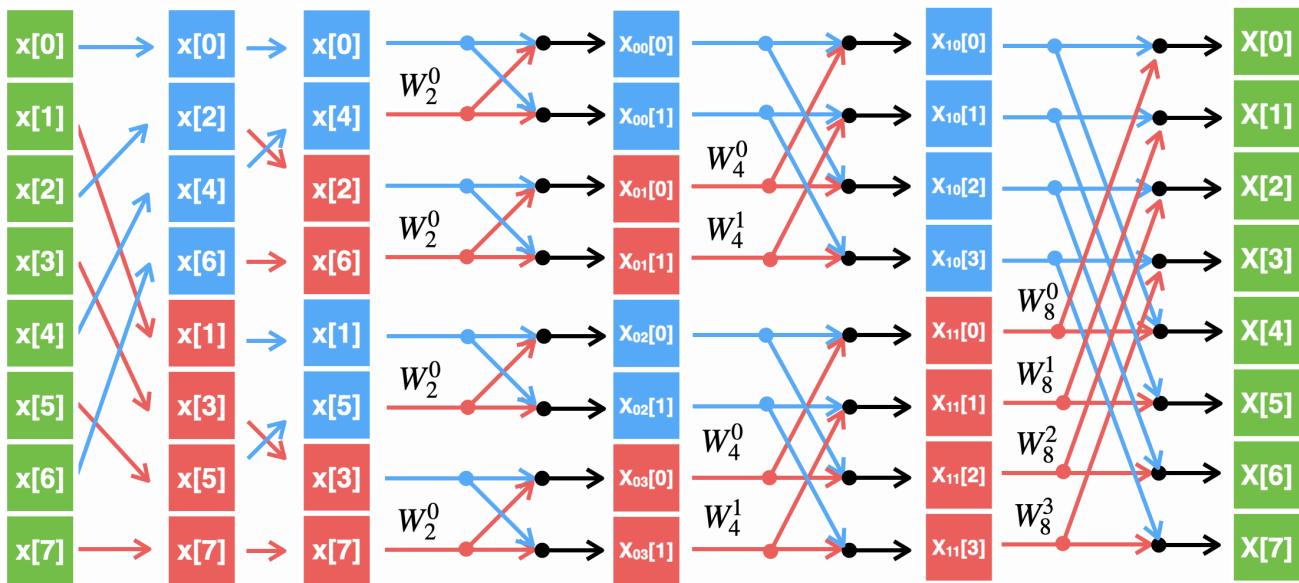


Рис. 3 – Граф «бабочка» для БПФ с прореживанием по времени

времени.

Таким образом, берём анализируемый сигнал и продолжаем разбивать его на сигналы, состоящие из чётных и нечётных индексов до тех пор, пока не получим набор сигналов из двух отсчётов. На рисунке 4 показан пример такого разбиения для сигнала из восьми отсчётов. Для вычисления ДПФ такого сигнала

Рис. 4 – Алгоритм БПФ с прореживанием по времени для $N=8$

требуется 3 стадии разбиения. На первом этапе получаем набор из четырёх ДПФ от сигналов из двух отсчётов:

$$X_{00}[0] = x[0] + W_2^0 x[4]$$

$$X_{00}[1] = x[0] - W_2^0 x[4]$$

$$X_{01}[0] = x[2] + W_2^0 x[6]$$

$$X_{01}[1] = x[2] - W_2^0 x[6]$$

$$X_{02}[0] = x[1] + W_2^0 x[5]$$

$$X_{02}[1] = x[1] - W_2^0 x[5]$$

$$X_{03}[0] = x[3] + W_2^0 x[7]$$

$$X_{03}[1] = x[3] - W_2^0 x[7]$$

Далее, на основе полученных результатов формируется два ДПФ для сигналов, состоящих из четырёх отсчётов:

$$X_{10}[0] = X_{00}[0] + W_4^0 X_{01}[0]$$

$$X_{10}[1] = X_{00}[1] + W_4^1 X_{01}[1]$$

$$X_{10}[2] = X_{00}[0] - W_4^0 X_{01}[0]$$

$$X_{10}[3] = X_{00}[1] - W_4^1 X_{01}[1]$$

$$X_{11}[0] = X_{02}[0] + W_4^0 X_{03}[0]$$

$$X_{11}[1] = X_{02}[1] + W_4^1 X_{03}[1]$$

$$X_{11}[2] = X_{02}[0] - W_4^0 X_{03}[0]$$

$$X_{11}[3] = X_{02}[1] - W_4^1 X_{03}[1]$$

И на последнем этапе формируются итоговые результаты восьмиточечного ДПФ:

$$X[0] = X_{10}[0] + W_8^0 X_{11}[0]$$

$$X[1] = X_{10}[1] + W_8^1 X_{11}[1]$$

$$X[2] = X_{10}[2] + W_8^2 X_{11}[2]$$

$$X[3] = X_{10}[3] + W_8^3 X_{11}[3]$$

$$X[4] = X_{10}[0] - W_8^0 X_{11}[0]$$

$$X[5] = X_{10}[1] - W_8^1 X_{11}[1]$$

$$X[6] = X_{10}[2] - W_8^2 X_{11}[2]$$

$$X[7] = X_{10}[3] - W_8^3 X_{11}[3]$$

ДПФ рассчитано!

3 Двоично-инверсная перестановка

Если количество отсчётов исходного сигнала $N = 2^n$, где n – целое положительное число, то его разбиение на сигналы, состоящие из чётных и нечётных индексов вплоть до последнего уровня можно сделать легко и за одну итерацию с помощью **двоично-инверсной перестановки**.

Для этого записываем индексы всех отсчётов в двоичной системе счисления, при этом должны быть записаны все n бит индекса, включая ведущие нули. Затем зеркально отображаем код каждого из этих двоичных чисел и записываем полученные результаты обратно в десятичную систему счисления. Готово! Если расположить элементы исходного массива в соответствии с полученными индексами, мы получим подряд идущие пары отсчётов для двухточечного ДПФ. Пример, как это работает для сигнала из восьми отсчётов, представлен в таблице 1. Можете сравнить с рисунком 4, действительно совпадает. Именно поэтому, рассматриваемые нами алгоритмы также называются БПФ по основанию два. У них есть одно небольшое ограничение – количество

Таблица 1 – Двоично-инверсная перестановка

Номер до перестановки	Двоичное представление	Двоично-инверсная перестановка	Номер после перестановки
0	000	000	1
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

отсчётов анализируемого сигнала должно быть равно степени двойки (например, 16, 32, 64, 128, 256 и т.д.). «Но как же быть, если, скажем, в моём сигнале всего 200 отсчётов?» – спросите вы. Ничего страшного, нужно просто дополнить сигнал нулевыми отсчётами, пока его длина не станет равна ближайшей степени двойки.

4 Алгоритм БПФ с прореживанием по частоте

Как вы догадались из названия раздела, прореживать можно не только время, но ещё и частоту. Граф «бабочка» для БПФ с прореживанием по частоте показан на рисунке 5. Подробно рассматривать данный алгоритм не будем, просто примем его как данность. Пример БПФ с прореживанием по частоте для сигнала,

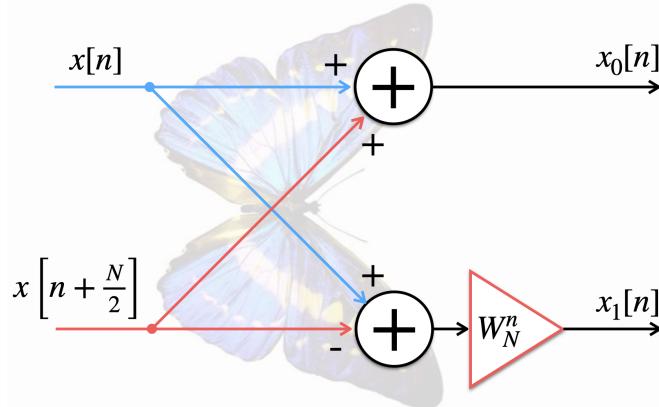


Рис. 5 – Граф «бабочка» для БПФ с прореживанием по частоте

состоящего из 8 отсчётов показан на рисунке 6. Прореживание (двоично-инверсная перестановка) в данном случае производится уже после вычисления ДПФ, т.е. по частотным отсчётам, отсюда и такое название. По количеству операций умножения данный алгоритм схож с алгоритмом прореживания по времени, поэтому эффективность их схожая. Какой метод выбрать – решать вам.

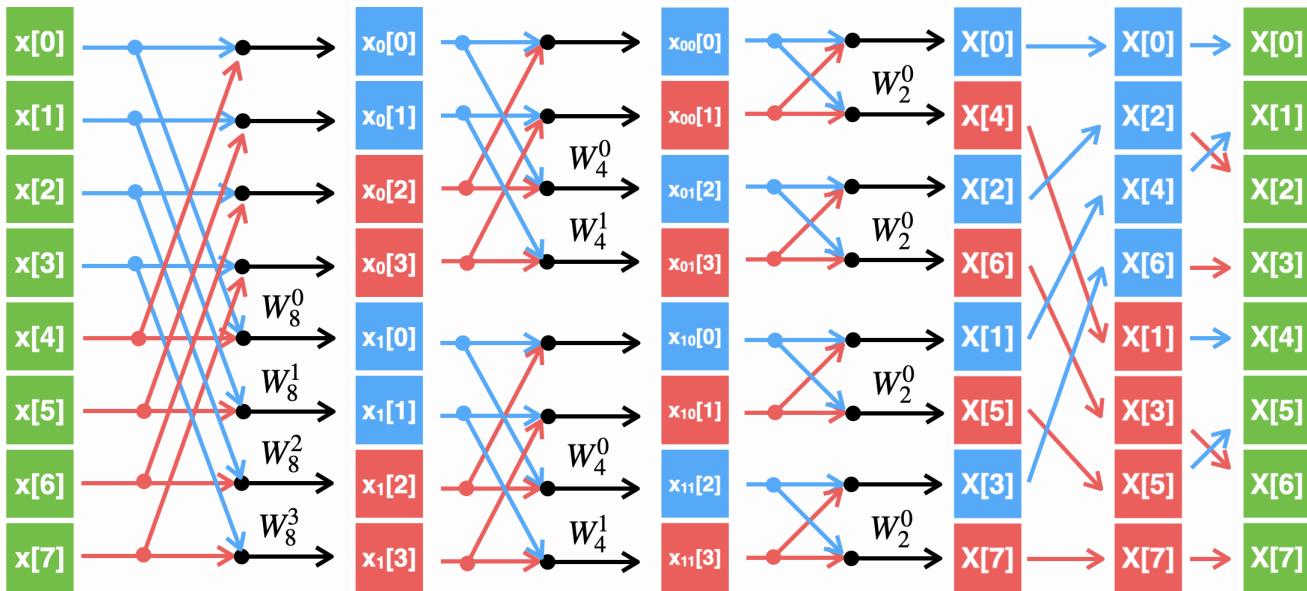


Рис. 6 – Алгоритм БПФ с прореживанием по частоте для N=8

5 Поворотные коэффициенты

В вышеупомянутых «бабочках» есть множитель W_N^m . Он называется **поворотным коэффициентом**. На нижнем уровне БПФ, когда имеем дело с двумя отсчётами, требуется всего один поворотный коэффициент $W_2^0 = 1$. Умножение на единицу считается тривиальной операцией, откуда следует, что на первом уровне не требуется ни одной операции умножения, только сложение и вычитание.

На втором уровне у нас получается два поворотных коэффициента $W_4^0 = 1$ и $W_4^1 = -j$. Умножение на $-j$ также делается просто: нужно поменять местами действительную и мнимую часть, а также инвертировать знак мнимой части. На этом уровне также не требуется операций умножения.

На следующем уровне требуется четыре поворотных коэффициента W_8^0 , W_8^1 , W_8^2 и W_8^3 , их значения мы рассчитывали в начале нашей статьи.

Давайте проиллюстрируем на графике все упомянутые выше поворотные коэффициенты (рисунок 7). Из графика видно, что на каждом следующем уровне количество поворотных коэффициентов увеличивается в два раза, при этом половина из них совпадает с поворотными коэффициентами предыдущего уровня.

Когда БПФ используется для решения каких-либо задач в устройствах цифровой обработки сигналов, количество отсчётов анализируемого сигнала как правило заранее известно и скорее всего не будет меняться в процессе работы устройства. Поэтому, чтобы не тратить время на вычисление W_N^m , их заранее рассчитанные значения для заданного количества отсчётов и всех уровней разбиения записывают в памяти в виде таблицы констант, которые в дальнейшем будут использоваться для умножения в графах «бабочка».

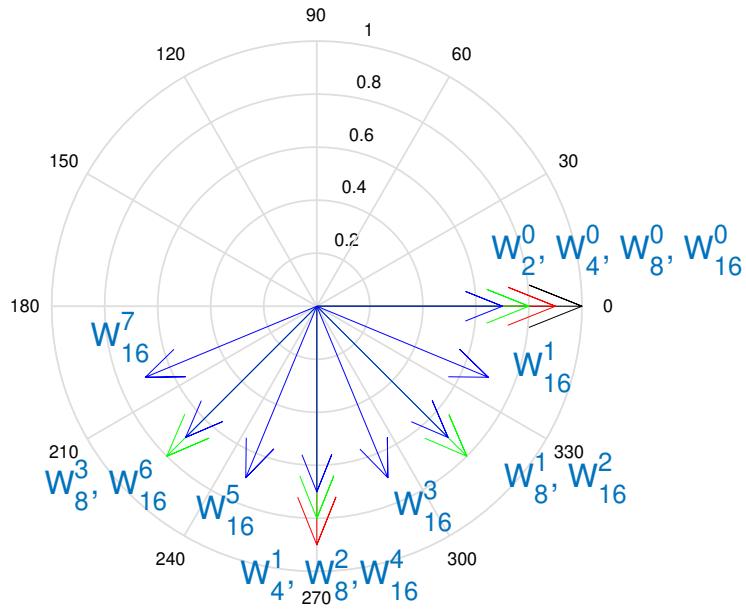


Рис. 7 – Поворотные коэффициенты

6 Выводы

БПФ – это всего лишь алгоритм эффективного вычисления ДПФ, поэтому результаты вычисления БПФ и ДПФ получаются абсолютно идентичны. Для вычисления ДПФ сигнала, состоящего из N отсчётов нам требуется N^2 вычислений с комплексными числами, в случае с БПФ – $\frac{N}{2} \log_2 N$ вычислений с комплексными числами.

Чтобы наглядно продемонстрировать эффективность алгоритма, давайте взглянем на таблицу 2.

Таблица 2 – Сравнение эффективности ДПФ и БПФ

Кол-во отсчётов, N	Количество вычислений с комплексными числами		Эффективность
	ДПФ	БПФ	
256	65 536	1 024	64:1
512	262 144	2 304	114:1
1024	1 048 576	5 120	205:1
2048	4 194 304	11 264	373:1
4096	16 777 216	24 576	683:1

Особенна заметен прирост эффективности с увеличением количества отсчётов.