

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Цифровая обработка сигналов»**  
**Тема: Частотный анализ полиномиальных приближений**

Студент гр. 9381	_____	Колованов Р.А.
Студент гр. 9381	_____	Семенов А.Н.
Преподаватель	_____	Середа А.-В. И.

Санкт-Петербург  
2022

### **Цель работы.**

Провести анализ частотных характеристик известных формул полиномиального сглаживания временных рядов.

### **Основные теоретические положения.**

Свойства любого фильтра однозначно определяют его частотная и фазовая характеристики. Они показывают, какое влияние фильтр оказывает на амплитуду и фазу различных гармоник обрабатываемого сигнала.

К наиболее известным типам нерекурсивных цифровых фильтров (НЦФ) относятся частотные фильтры, алгоритм которых для симметричных НЦФ, не изменяющих фазу сигналов, имеет вид:

$$y_k = \sum_{n=-N}^N h_n s_{k-n}$$

В качестве временного ряда рассматривается дискретный сигнал с шагом дискретизации, равным единице.

Под полиномиальным сглаживанием понимается аппроксимация в смысле МНК значений конечного (нечетного) числа элементов сглаживаемого ряда полиномом заданного порядка с присвоением среднему из этих элементов значения сглаживающего полинома в центре выбранного временного отрезка. Такой подход соответствует так называемому сглаживанию в скользящем окне.

### **Постановка задачи.**

Получить формулы для передаточных функций нерекурсивных фильтров, соответствующих полиномиальному сглаживанию дискретного сигнала для полиномов различного порядка и построить графики  $\tilde{H}(f)$ . Проинтерпретировать частотные свойства передаточных функций. Провести сопоставительный анализ частотных характеристик передаточных функций для различных степеней полиномов.

## Выполнение работы.

1. Вывести формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией по 3, 5, 7 и 9 точкам. Построить графики  $\tilde{H}(f)$ . Проинтерпретировать частотные свойства передаточных функций для различного количества точек.

Выведем формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией по  $2m + 1$  точкам:

Имеем входной сигнал  $s(t)$  и выходной сигнал  $y(t) = A + Bt$ . Для осуществления приближения (в смысле МНК) прямой линией по  $2m + 1$  точкам необходимо минимизировать следующее выражение:

$$F(A, B) = \sum_{k=-m}^m (s_k - y_k)^2 = \sum_{k=-m}^m (s_k - A - Bk)^2 \Rightarrow \min$$

Неизвестными коэффициентами в этой задаче являются коэффициенты  $A$  и  $B$ . Для нахождения коэффициентов продифференцируем и приравняем к нулю частные производные по  $A$  и  $B$ :

$$\begin{aligned} \begin{cases} \frac{\delta F(A, B)}{\delta A} = 0 \\ \frac{\delta F(A, B)}{\delta B} = 0 \end{cases} &\Leftrightarrow \begin{cases} -2 \sum_{k=-m}^m (s_k - A - Bk) = 0 \\ -2 \sum_{k=-m}^m (ks_k - kA - Bk^2) = 0 \end{cases} \Leftrightarrow \\ &\Leftrightarrow \begin{cases} - \sum_{k=-m}^m s_k + \sum_{k=-m}^m A + \sum_{k=-m}^m Bk = 0 \\ - \sum_{k=-m}^m ks_k + \sum_{k=-m}^m kA + \sum_{k=-m}^m Bk^2 = 0 \end{cases} \end{aligned}$$

Отсюда выражаем коэффициенты  $A$  и  $B$ :

$$\begin{cases} A = \frac{1}{2m+1} \sum_{k=-m}^m s_k \\ B = \frac{\sum_{k=-m}^m s_k}{\sum_{k=-m}^m k^2} \end{cases}$$

В итоге получаем:

$$y_0 = A = \frac{1}{2m+1} \sum_{k=-m}^m s_k = \frac{1}{2m+1} (s_{-m} + s_{-m+1} + \dots + s_{m-1} + s_m)$$

В общем случае:

$$\begin{aligned} y_n &= \frac{1}{2m+1} \sum_{k=-m+n}^{m+n} s_k = \\ &= \frac{1}{2m+1} (s_{-m+n} + s_{-m+1+n} + \dots + s_{m-1+n} + s_{m+n}) \end{aligned}$$

$$s_n = e^{i\omega n}$$

$$y_n = \frac{1}{2m+1} (e^{-mi\omega} + e^{(-m+1)i\omega} + \dots + e^{(m-1)i\omega} + e^{mi\omega}) e^{ni\omega} = H(\omega) e^{ni\omega}$$

$$H(\omega) = \frac{1}{2m+1} (1 + 2 \cos(\omega) + 2 \cos(2\omega) + \dots + 2 \cos(m\omega))$$

$$H(\omega) = \frac{\sin(\frac{(2m+1)\omega}{2})}{(2m+1) \sin(\frac{\omega}{2})}$$

$$H(\omega) = H(2\pi f) = \tilde{H}(f)$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией по 3 точкам:

$$\tilde{H}_3(f) = \frac{\sin(3\pi f)}{3 \sin(\pi f)}$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией по 5 точкам:

$$\tilde{H}_5(f) = \frac{\sin(5\pi f)}{5 \sin(\pi f)}$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией по 7 точкам:

$$\tilde{H}_7(f) = \frac{\sin(7\pi f)}{7 \sin(\pi f)}$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией по 9 точкам:

$$\tilde{H}_9(f) = \frac{\sin(9\pi f)}{9 \sin(\pi f)}$$

Графики для передаточных функций нерекурсивного фильтра на интервале  $f \in [0; 1]$  представлены рис. 1.

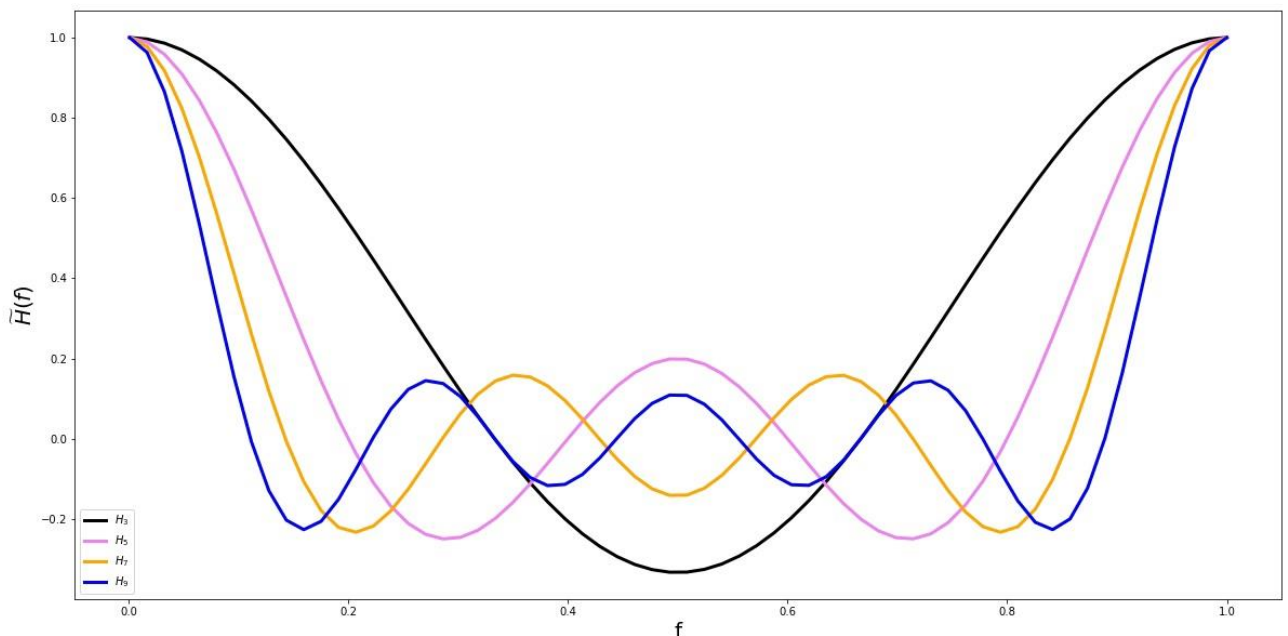


Рисунок 1 – Передаточные функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией по 3, 5, 7 и 9 точкам.

Из графика видно, что за один период каждая из передаточных функций имеет ровно столько экстремумов, сколько было использовано точек для ее построения. Следовательно, чем больше экстремумов имеется, тем больше колебаний получится. К тому же порядок гладкости увеличивается с увеличением количества точек для построения функции.

2. Вывести формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени по 7, 9, 11 и 13 точкам. Построить графики  $\tilde{H}(f)$ . Проинтерпретировать частотные свойства передаточных функций для различного количества точек.

Выведем формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени по  $2m + 1$  точкам:

Имеем входной сигнал  $s(t)$  и выходной сигнал  $y(t) = A + Bt + Ct^2$ . Для осуществления приближения (в смысле МНК) полиномом второй степени по  $2m + 1$  точкам необходимо минимизировать следующее выражение:

$$F(A, B, C) = \sum_{k=-m}^m (s_k - y_k)^2 = \sum_{k=-m}^m (s_k - A - Bk - Ck^2)^2 \Rightarrow \min$$

Неизвестными коэффициентами в этой задаче являются коэффициенты  $A$ ,  $B$  и  $C$ . Для нахождения коэффициентов продифференцируем и приравняем к нулю частные производные по  $A$  и  $C$ :

$$\begin{aligned} \begin{cases} \frac{\delta F(A, B, C)}{\delta A} = 0 \\ \frac{\delta F(A, B, C)}{\delta C} = 0 \end{cases} &\Leftrightarrow \begin{cases} -2 \sum_{k=-m}^m (s_k - A - Bk - Ck^2) = 0 \\ -2 \sum_{k=-m}^m (k^2 s_k - k^2 A - Bk^3 - Ck^4) = 0 \end{cases} \Leftrightarrow \\ &\Leftrightarrow \begin{cases} - \sum_{k=-m}^m s_k + \sum_{k=-m}^m A + \sum_{k=-m}^m Bk + \sum_{k=-m}^m Ck^2 = 0 \\ - \sum_{k=-m}^m k^2 s_k + \sum_{k=-m}^m k^2 A + \sum_{k=-m}^m Bk^3 + \sum_{k=-m}^m Ck^4 = 0 \end{cases} \end{aligned}$$

Отсюда получаем:

$$\left\{ \begin{array}{l} (2m+1)A + \frac{m(m+1)(2m+1)}{3}C = \sum_{k=-m}^m s_k \\ \frac{m(m+1)(2m+1)}{3}A + \frac{m(m+1)(2m+1)(3m^2+3m-1)}{15}C = \sum_{k=-m}^m k^2 s_k \end{array} \right.$$

Подставим  $\frac{m(m+1)(2m+1)}{3}C$  во второе уравнение:

$$\frac{m(m+1)(2m+1)}{3}A + \frac{3m^2+3m-1}{5} \left( \sum_{k=-m}^m s_k - (2m+1)A \right) = \sum_{k=-m}^m k^2 s_k$$

$\Leftrightarrow$

$$\begin{aligned} & \left( \frac{m(m+1)(2m+1)}{3} - \frac{(3m^2+3m-1)(2m+1)}{5} \right) A \\ &= \sum_{k=-m}^m k^2 s_k - \frac{3m^2+3m-1}{5} \sum_{k=-m}^m s_k \end{aligned}$$

Тогда:

$$A = \frac{\sum_{k=-m}^m k^2 s_k - \frac{3m^2+3m-1}{5} \sum_{k=-m}^m s_k}{\frac{m(m+1)(2m+1)}{3} - \frac{(3m^2+3m-1)(2m+1)}{5}}$$

В итоге получаем:

$$y_0 = A = \frac{\sum_{k=-m}^m k^2 s_k - \frac{3m^2+3m-1}{5} \sum_{k=-m}^m s_k}{\frac{m(m+1)(2m+1)}{3} - \frac{(3m^2+3m-1)(2m+1)}{5}}$$

Например, для 7 точек:

$$y_7 = A = \frac{\sum_{k=-3}^3 k^2 s_k - 7 \sum_{k=-3}^3 s_k}{28 - 49} = \frac{1}{21} \left( 7 \sum_{k=-3}^3 s_k - \sum_{k=-3}^3 k^2 s_k \right) =$$

$$\begin{aligned}
&= \frac{1}{21} (7s_{-3} + 7s_{-2} + 7s_{-1} + 7s_0 + 7s_1 + 7s_2 + 7s_3 - 9s_{-3} - 4s_{-2} - s_1 - 4s_2 \\
&\quad - 9s_3) = \frac{1}{21} (-2s_{-3} + 3s_{-2} + s_{-1} + 7s_0 + 6s_1 + 3s_2 + 2s_3) \\
y_7 &= \frac{1}{21} (-2s_{-3} + 3s_{-2} + s_{-1} + 7s_0 + 6s_1 + 3s_2 + 2s_3)
\end{aligned}$$

В общем случае:

$$\begin{aligned}
y_n &= \frac{1}{21} (-2s_{n-3} + 3s_{n-2} + s_{n-1} + 7s_n + 6s_{n+1} + 3s_{n+2} + 2s_{n+3}) \\
s_n &= e^{i\omega n} \\
y_n &= \frac{1}{21} (-2e^{-3i\omega} + 3e^{-2i\omega} + 6e^{-i\omega} + 7 + 6e^{i\omega} + 3e^{2i\omega} - 2e^{3i\omega}) e^{i\omega n} \\
&= H(\omega) e^{i\omega n} \\
H(\omega) &= \frac{1}{21} (-2e^{-3i\omega} + 3e^{-2i\omega} + 6e^{-i\omega} + 7 + 6e^{i\omega} + 3e^{2i\omega} - 2e^{3i\omega}) \\
H(\omega) &= \frac{1}{21} (7 + 12 \cos \omega + 6 \cos 2\omega - 4 \cos 3\omega) \\
H(\omega) &= H(2\pi f) = \tilde{H}(f)
\end{aligned}$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени по 7 точкам:

$$\tilde{H}_7(f) = \frac{1}{21} (7 + 12 \cos(2\pi f) + 6 \cos(4\pi f) - 4 \cos(6\pi f))$$

Аналогично можно найти  $\tilde{H}(f)$  для 9, 11 и 13 точек.

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени по 9 точкам:

$$\tilde{H}_9(f) = \frac{1}{231} (59 + 108 \cos(2\pi f) + 78 \cos(4\pi f) + 28 \cos(6\pi f) - 42 \cos(8\pi f))$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени по 11 точкам:

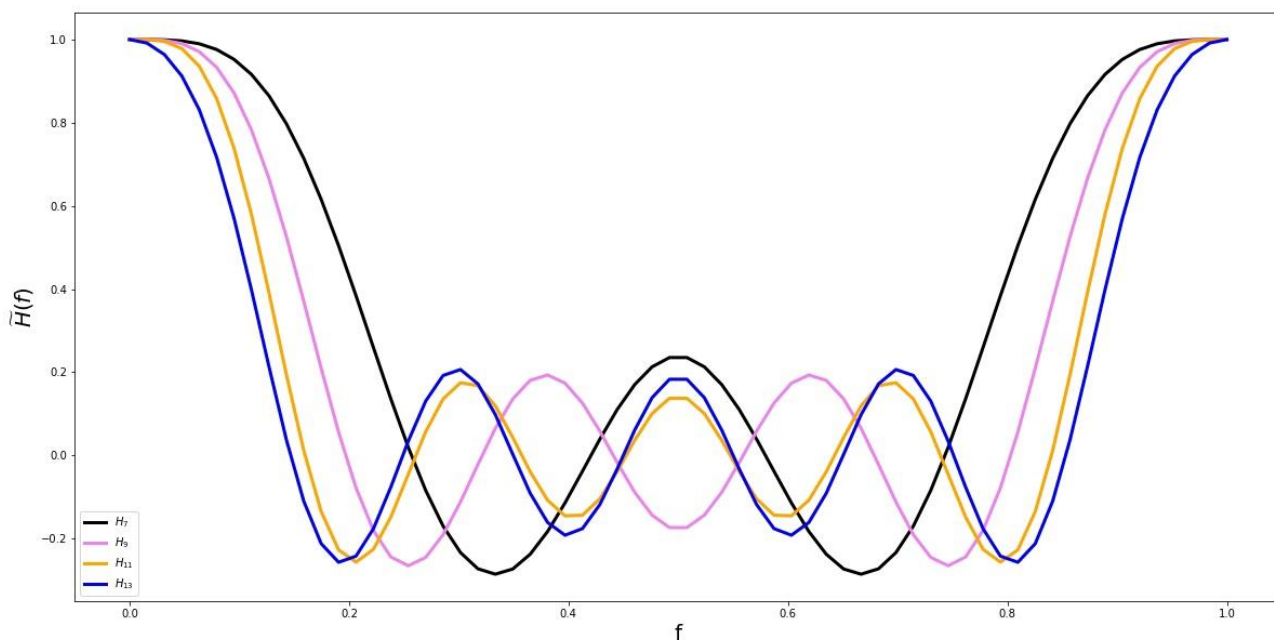


$$\tilde{H}_{11}(f) = \frac{1}{429} (89 + 168 \cos(2\pi f) + 138 \cos(4\pi f) + 88 \cos(6\pi f) + 18 \cos(8\pi f) - 72 \cos(10\pi f))$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени по 13 точкам:

$$\tilde{H}_{13}(f) = \frac{1}{143} (25 + 48 \cos(2\pi f) + 42 \cos(4\pi f) + 32 \cos(6\pi f) + 18 \cos(8\pi f) - 22 \cos(10\pi f))$$

Графики для передаточных функций на интервале  $f \in [0; 1]$  представлены рис. 2.



*Рисунок 2 – Передаточные функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени по 7, 9, 11 и 13 точкам.*

3. Вывести формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом четвёртой степени по 9, 11, 13 и 15 точкам. Построить графики  $\tilde{H}(f)$ . Проинтерпретировать частотные свойства передаточных функций для различного количества точек.

Имеем входной сигнал  $s(t)$  и выходной сигнал  $y(t) = A + Bt + Ct^2 + Dt^3 + Et^4$ . Для осуществления приближения (в смысле МНК) полиномом четвертой степени по  $2m + 1$  точкам необходимо минимизировать следующее выражение:

$$F(A, B, C) = \sum_{k=-m}^m (s_k - y_k)^2 = \sum_{k=-m}^m (s_k - A - Bk - Ck^2 + Dk^3 + Ek^4)^2 \Rightarrow \min$$

Посчитаем частные производные по  $A$ ,  $C$  и  $E$ , получим:

$$\begin{cases} \frac{\delta F(A, B, C, D, E)}{\delta A} = 0 \\ \frac{\delta F(A, B, C, D, E)}{\delta C} = 0 \\ \frac{\delta F(A, B, C, D, E)}{\delta E} = 0 \end{cases} \Leftrightarrow \begin{cases} -2 \sum_{k=-m}^m (s_k - A - Bk - Ck^2 - Dk^3 - Ek^4) = 0 \\ -2 \sum_{k=-m}^m (k^2 s_k - k^2 A - Bk^3 - Ck^4 - Dk^5 - Ek^6) = 0 \\ -2 \sum_{k=-m}^m (k^4 s_k - k^4 A - Bk^5 - Ck^6 - Dk^7 - Ek^8) = 0 \end{cases} \Leftrightarrow$$

$$\begin{cases} - \sum_{k=-m}^m s_k + \sum_{k=-m}^m A + \sum_{k=-m}^m Bk + \sum_{k=-m}^m Ck^2 + \sum_{k=-m}^m Dk^3 + \sum_{k=-m}^m Ek^4 = 0 \\ - \sum_{k=-m}^m k^2 s_k + \sum_{k=-m}^m k^2 A + \sum_{k=-m}^m Bk^3 + \sum_{k=-m}^m Ck^4 + \sum_{k=-m}^m Dk^5 + \sum_{k=-m}^m Ek^6 = 0 \\ - \sum_{k=-m}^m k^4 s_k + \sum_{k=-m}^m k^4 A + \sum_{k=-m}^m Bk^5 + \sum_{k=-m}^m Ck^6 + \sum_{k=-m}^m Dk^7 + \sum_{k=-m}^m Ek^8 = 0 \end{cases}$$

$$\begin{cases} - \sum_{k=-m}^m s_k + \sum_{k=-m}^m A + \sum_{k=-m}^m Ck^2 + \sum_{k=-m}^m Ek^4 = 0 \\ - \sum_{k=-m}^m k^2 s_k + \sum_{k=-m}^m k^2 A + \sum_{k=-m}^m Ck^4 + \sum_{k=-m}^m Ek^6 = 0 \\ - \sum_{k=-m}^m k^4 s_k + \sum_{k=-m}^m k^4 A + \sum_{k=-m}^m Ck^6 + \sum_{k=-m}^m Ek^8 = 0 \end{cases}$$

Отсюда:

$$\left\{ \begin{aligned} (2m+1)A + \frac{m(m+1)(2m+1)}{3}C + \frac{m(m+1)(2m+1)(3m^2+3m-1)}{15}F = \\ = \sum_{k=-m}^m s_k \\ \frac{m(m+1)(2m+1)}{3}A + \frac{m(m+1)(2m+1)(3m^2+3m-1)}{15}C + \\ + \frac{m(m+1)(2m+1)(1-3m+6m^3+3m^4)}{21}F = \sum_{k=-m}^m k^2 s_k \\ \frac{m(m+1)(2m+1)(3m^2+3m-1)}{15}A + \frac{m(m+1)(2m+1)(1-3m+6m^3+3m^4)}{21}C + \\ + \frac{m(m+1)(2m+1)(-3+9m-m^2-15m^3+5m^4+15m^5+5m^6)}{45} = \sum_{k=-m}^m k^4 s_k \end{aligned} \right.$$

Выразим  $C$  из первого уравнения системы нормальных уравнений:

$$C = \frac{-15A - 30Am + Em - 10Em^3 - 15Em^4 - 6Em^5 - 15 \sum_{k=-m}^m s_k}{5m(m+1)(2m+1)}$$

Подставим теперь  $C$  во второе и третье уравнение:

$$\left\{ \begin{aligned} & \frac{(-3-2m+12m^2+8m^3)(-35A+3Em(-2-m+2m^2+m^3))}{525} + \\ & + \frac{3m^2+3m-1}{5} \sum_{k=-m}^m s_k = \sum_{k=-m}^m k^2 s_k \\ & - \frac{(-3-2m+12m^2+8m^3)}{315} * \\ & * (3A(6m^2+6m-5) + Em(-6+m+12m^2+m^3-6m^4-2m^5)) + \\ & + \frac{1-3m+6m^3+3m^4}{7} \sum_{k=-m}^m s_k = \sum_{k=-m}^m k^4 s_k \end{aligned} \right.$$

Выразим из 2 уравнения  $E$  и подставим в 3 уравнение:

$$\left\{ \begin{aligned} E = 35 \cdot \frac{A(-3-2m+12m^2+8m^3) + (3-9m-9m^2) \sum_{k=-m}^m s_k + 15 \sum_{k=-m}^m k^2 s_k}{3m(-2-m+2m^2+m^3)(-3-2m+12m^2+8m^3)} \\ - 4A(45+18m-200m^2-80m^3+80m^4+32m^5) - \\ - 15 \cdot (12-50m-35m^2+30m^3+15m^4) \sum_{k=-m}^m s_k + \\ + 525 \cdot (-3+2m+2m^2) \sum_{k=-m}^m k^2 s_k = 945 \sum_{k=-m}^m k^4 s_k \end{aligned} \right.$$

И выразим из 3 уравнения A:

$$A = 15 \left( \frac{(12 + 5m(1 + m)(-10 + 3m(1 + m))) \sum_{k=-m}^m s_k}{4(-3 + 2m)(-1 + 2m)(1 + 2m)(3 + 2m)(5 + 2m)} \right. \\ \left. - \frac{35(-3 + 2m(1 + m)) \sum_{k=-m}^m k^2 s_k - 63 \sum_{k=-m}^m k^4 s_k}{4(-3 + 2m)(-1 + 2m)(1 + 2m)(3 + 2m)(5 + 2m)} \right)$$

В итоге получаем:

$$y_0 = A = 15 \left( \frac{(12 + 5m(1 + m)(-10 + 3m(1 + m))) \sum_{k=-m}^m s_k}{4(-3 + 2m)(-1 + 2m)(1 + 2m)(3 + 2m)(5 + 2m)} \right. \\ \left. - \frac{35(-3 + 2m(1 + m)) \sum_{k=-m}^m k^2 s_k - 63 \sum_{k=-m}^m k^4 s_k}{4(-3 + 2m)(-1 + 2m)(1 + 2m)(3 + 2m)(5 + 2m)} \right)$$

Для 9 точек:

$$y_9 = \frac{1}{429} \left( 179 \sum_{k=-4}^4 s_k - \frac{1}{4} \left( 185 \sum_{k=-4}^4 k^2 s_k - 9 \sum_{k=-4}^4 k^4 s_k \right) \right) \\ y_9 = \frac{1}{429} (15s_{-4} - 55s_{-3} + 30s_{-2} + 135s_{-1} + 179s_0 + 135s_1 + 30s_2 - 55s_3 \\ + 15s_4)$$

В общем случае:

$$y_n = \frac{1}{429} (15s_{n-4} - 55s_{n-3} + 30s_{n-2} + 135s_{n-1} + 179s_n + 135s_{n+1} \\ + 30s_{n+2} - 55s_{n+3} + 15s_{n+4}) \\ s_n = e^{i\omega n} \\ y_n = \frac{1}{429} (15e^{-4i\omega} - 55e^{-3i\omega} + 30e^{-2i\omega} + 135e^{-i\omega} + 179 + 135e^{i\omega} + 30e^{2i\omega} \\ - 55e^{3i\omega} + 15e^{4i\omega}) e^{i\omega n} = H(\omega) e^{i\omega n} \\ H(\omega) = \frac{1}{429} (15e^{-4i\omega} - 55e^{-3i\omega} + 30e^{-2i\omega} + 135e^{-i\omega} + 179 + 135e^{i\omega} \\ + 30e^{2i\omega} - 55e^{3i\omega} + 15e^{4i\omega})$$

$$H(\omega) = \frac{1}{429}(179 + 270 \cos \omega + 60 \cos 2\omega - 110 \cos 3\omega + 30 \cos 4\omega)$$

$$H(\omega) = H(2\pi f) = \tilde{H}(f)$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом четвертой степени по 9 точкам:

$$\begin{aligned} \tilde{H}_9(f) = \frac{1}{429} (179 + 270 \cos(2\pi f) + 60 \cos(4\pi f) - 110 \cos(6\pi f) \\ + 30 \cos(8\pi f)) \end{aligned}$$

Аналогично можно найти  $\tilde{H}(f)$  для 11, 13 и 15 точек.

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом четвертой степени по 11 точкам:

$$\begin{aligned} \tilde{H}_{11}(f) = \frac{1}{429} (143 + 240 \cos(2\pi f) + 120 \cos(4\pi f) - 20 \cos(6\pi f) \\ - 90 \cos(8\pi f) + 36 \cos(10\pi f)) \end{aligned}$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом четвертой степени по 13 точкам:

$$\begin{aligned} \tilde{H}_{13}(f) = \frac{1}{2431} (677 + 1200 \cos(2\pi f) + 780 \cos(4\pi f) + 220 \cos(6\pi f) \\ - 270 \cos(8\pi f) - 396 \cos(10\pi f) + 220 \cos(12\pi f)) \end{aligned}$$

Формула для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию полиномом четвертой степени по 15 точкам:

$$\begin{aligned} \tilde{H}_{15}(f) = \frac{1}{46\,189} (11\,063 + 20\,250 \cos(2\pi f) + 15\,000 \cos(4\pi f) \\ + 7\,510 \cos(6\pi f) - 330 \cos(8\pi f) - 5\,874 \cos(10\pi f) \\ + 5\,720 \cos(12\pi f) + 4\,290 \cos(14\pi f)) \end{aligned}$$

Графики для передаточных функций на интервале  $f \in [0; 1]$  представлены рис. 3.

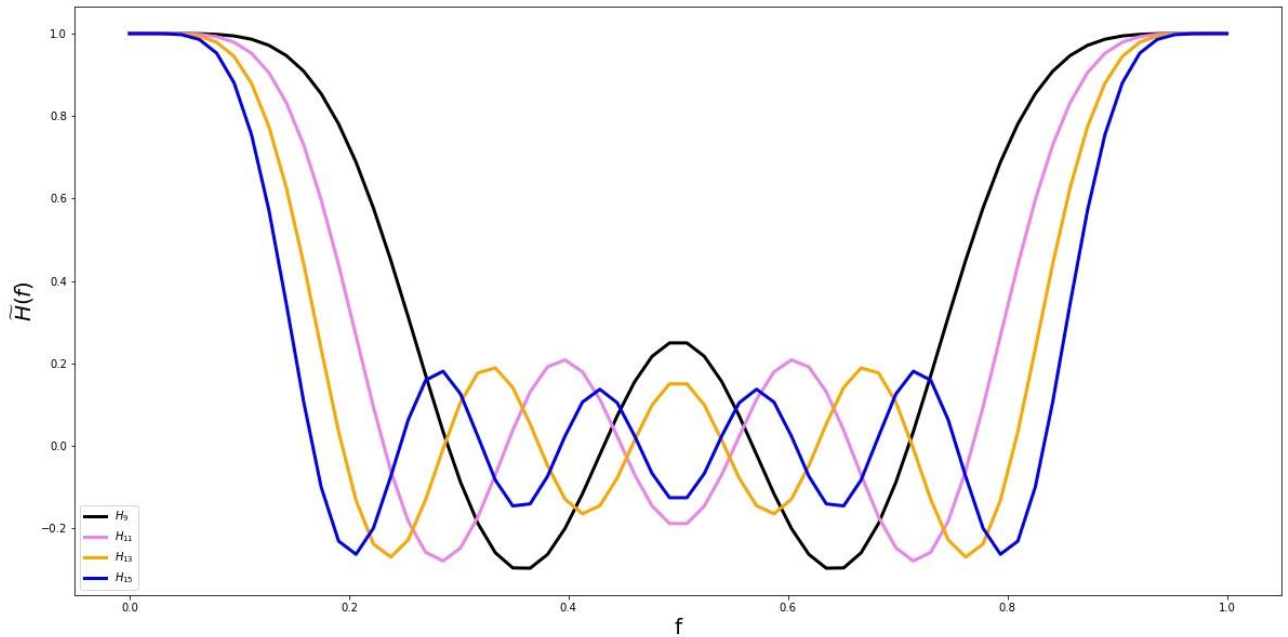


Рисунок 3 – Передаточные функции нерекурсивного фильтра, соответствующего сглаживанию полиномом четвертой степени по 7, 11, 13 и 15 точкам.

4. Вывести формулы для передаточной функции нерекурсивного фильтра, соответствующего сглаживанию по формулам Спенсера по 15 и 21 точкам. Построить графики  $\tilde{H}(f)$ . Проинтерпретировать частотные свойства передаточных функций для различного количества точек.

Для 15-ти точек имеем:

$$y_n = \frac{1}{320} (-3s_{n-7} - 6s_{n-6} - 5s_{n-5} + 3s_{n-4} + 21s_{n-3} + 46s_{n-2} - 67s_{n-1} + 74s_n + 67s_{n+1} + 46s_{n+2} + 21s_{n+3} + 3s_{n+4} - 5s_{n+5} - 6s_{n+6} - 3s_{n+7})$$

Для 21-ой точки имеем:

$$y_n = \frac{1}{350} (-s_{n-10} - 3s_{n-6} - 5s_{n-8} - 5s_{n-7} - 2s_{n-6} + 6s_{n-5} + 18s_{n-4} + 33s_{n-3} + 47s_{n-2} + 57s_{n-1} + 60s_n + 57s_{n+1} + 47s_{n+2} + 33s_{n+3} + 18s_{n+4} + 6s_{n+5} - 2s_{n+6} - 5s_{n+7} - 5s_{n+8} - 3s_{n+9} - s_{n+10})$$

Соответствующие передаточные функции:

$$\begin{aligned}\tilde{H}_{15}(f) &= \frac{1}{320} (74 + 134 \cos(2\pi f) + 92 \cos(4\pi f) + 42 \cos(6\pi f) + 6 \cos(8\pi f) \\ &\quad - 10 \cos(10\pi f) - 12 \cos(12\pi f) - 6 \cos(14\pi f)) \\ \tilde{H}_{21}(f) &= \frac{1}{350} (60 + 114 \cos(2\pi f) + 94 \cos(4\pi f) + 66 \cos(6\pi f) + 36 \cos(8\pi f) \\ &\quad + 12 \cos(10\pi f) - 4 \cos(12\pi f) - 10 \cos(14\pi f) - 10 \cos(16\pi f) \\ &\quad - 6 \cos(18\pi f) - 2 \cos(20\pi f))\end{aligned}$$

Графики для передаточных функций на интервале  $f \in [0; 1]$  представлены рис. 4.

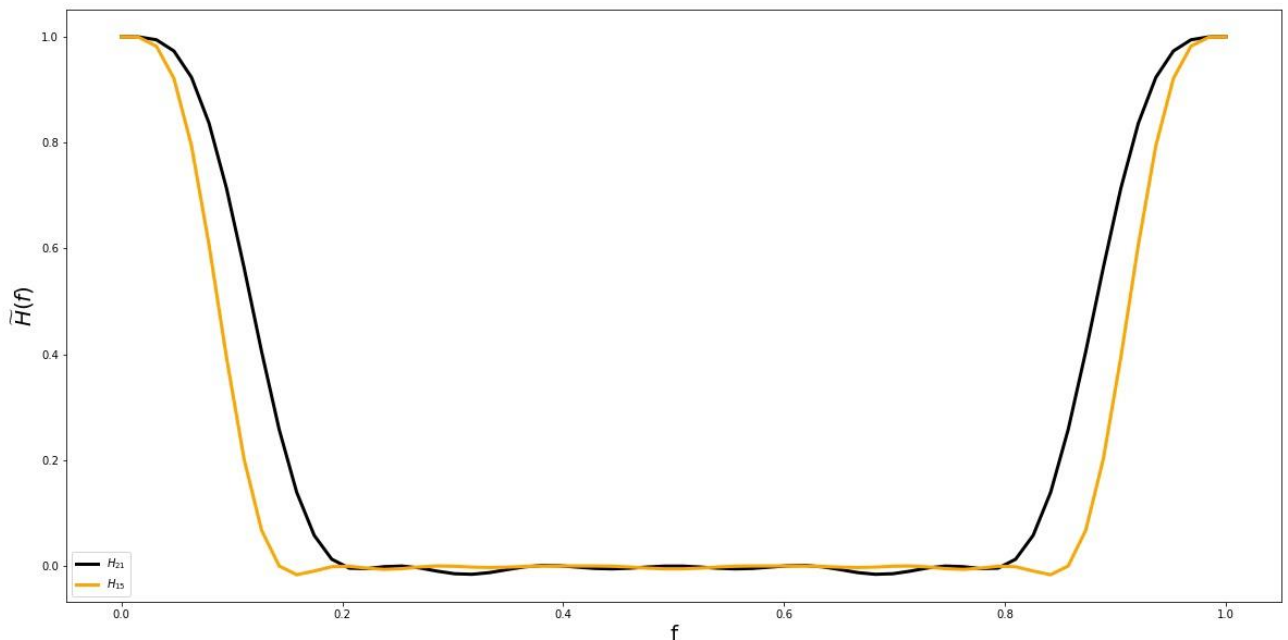


Рисунок 4 – Графики передаточной функции не рекурсивного фильтра, соответствующего сглаживанию по формулам Спенсера по 15 и 21 точкам.

5. Построить графики из предыдущих пунктов в логарифмической шкале (Дб). Объясните, чем отличаются данные графики от полученных ранее и объясните их смысл.

Построенные графики представлены на рисунках 5-8.

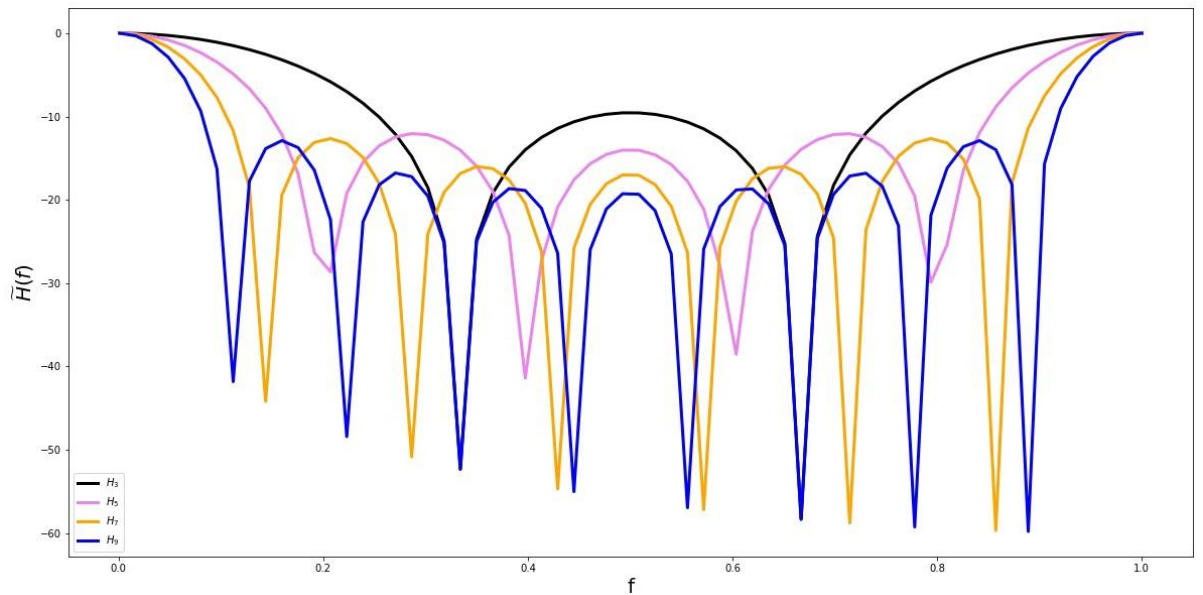


Рисунок 5 – Передаточные функции нерекурсивного фильтра, соответствующего сглаживанию прямой линией по 3, 5, 7 и 9 точкам в логарифмической шкале.

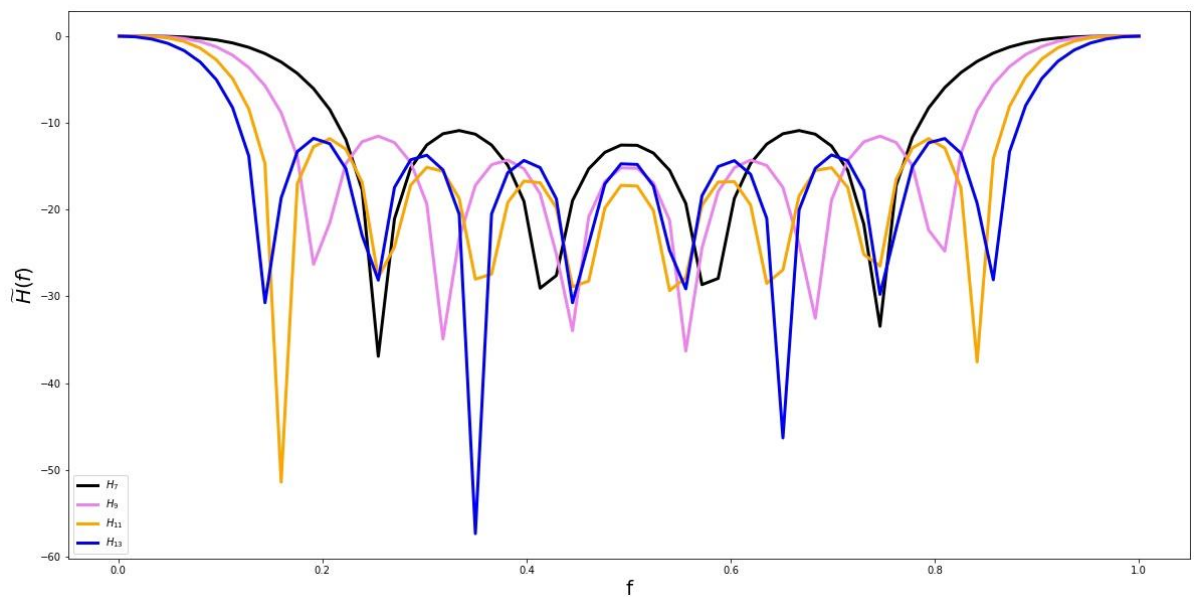
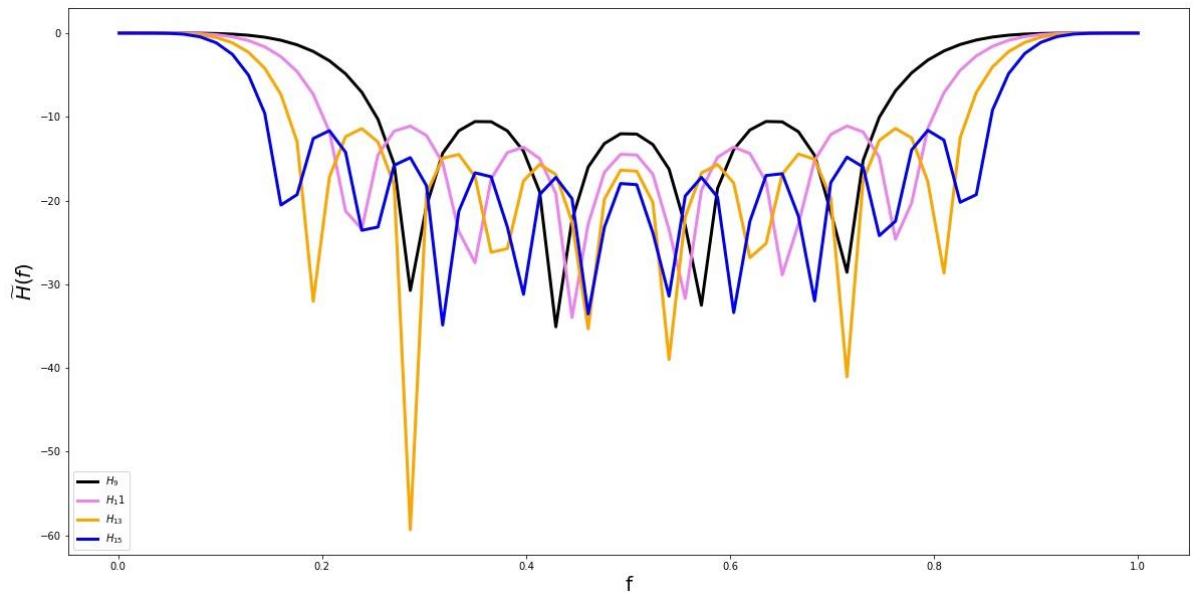
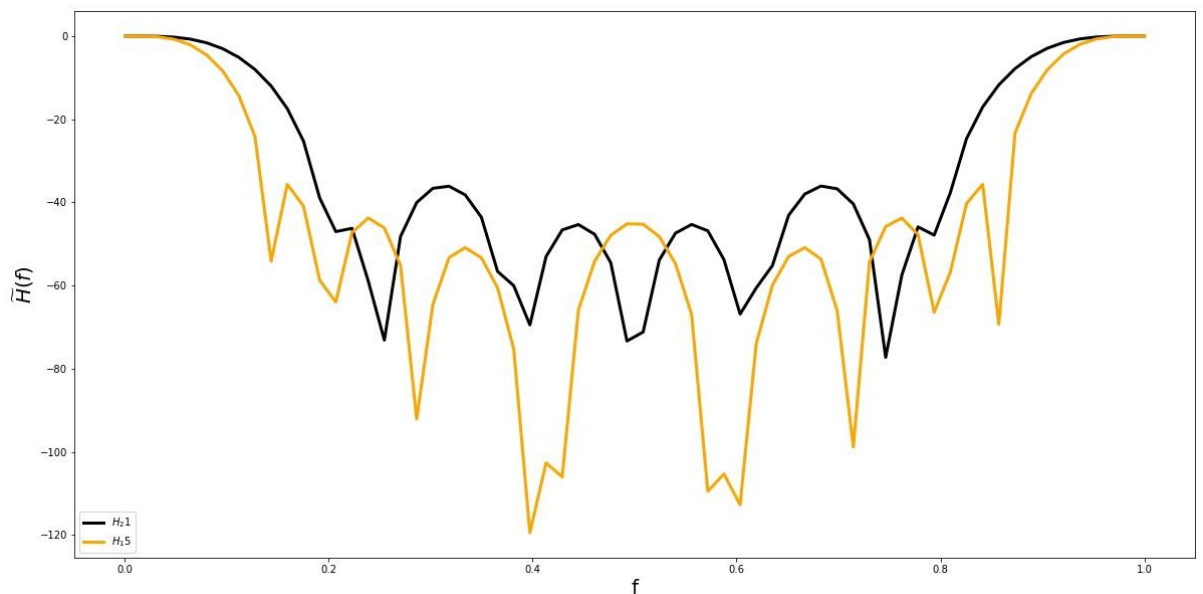


Рисунок 6 – Передаточные функции нерекурсивного фильтра, соответствующего сглаживанию полиномом второй степени по 7, 9, 11 и 13 точкам в логарифмической шкале.





*Рисунок 7 – Передаточные функции нерекурсивного фильтра, соответствующего сглаживанию полиномом четвертой степени по 7, 11, 13 и 15 точкам в логарифмической шкале.*



*Рисунок 8 – Графики передаточной функции нерекурсивного фильтра, соответствующего сглаживанию по формулам Спенсера по 15 и 21 точкам в логарифмической шкале.*

Графики передаточных функций (рис. 1 – 4), построенные ранее, недостаточно информативны, так как значения на высоких частотах настолько

малы, что их трудно различить. Поэтому лучше использовать логарифмы чисел  $H(w)$ . Для этой цели используются децибелы –  $20 \lg \left( \frac{|y_n|}{|s_n|} \right)$  Дб.

Графики на рис. 5 – 8 позволяют сделать вывод о том, что шум является высокочастотным, а сигнал низкочастотным. Формулы сглаживания пропускают одни частоты, а другие устраняют.

6. Провести сопоставительный анализ свойств передаточных функций, полученных при выполнении пп. 1–4.

Были получены графики передаточных функций нерекурсивных фильтров, соответствующих сглаживанию прямой линией, полиномом второй степени, полиномом четвёртой степени и формулам Спенсера (рис. 1 – 4).

Полученные графики подтверждают общее свойство всех передаточных функций – увеличение числа точек ведет к тому, что кривые спадают более быстро и величина последующих колебаний слегка уменьшается.

### **Выводы.**

В ходе выполнения лабораторной работы был проведен анализ частотных характеристик известных формул полиномиального сглаживания временных рядов полиномами второй степени, четвертой степени и с помощью формул Спенсера.

В процессе были выведены формулы для передаточных функций нерекурсивных фильтров, соответствующих полиномиальному сглаживанию дискретного сигнала для полиномов различного порядка и построены графики  $\tilde{H}(f)$ . Также были проинтерпретированы частотные свойства передаточных функций и проведен сопоставительный анализ частотных характеристик передаточных функций для различных степеней полиномов.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import numpy as np
import matplotlib.pyplot as plt
import math
pi = math.pi

plt.rcParams ['figure.figsize'] = [20, 10]

def task1():
    h_3 = lambda x: np.sin(3*math.pi*x)/(3*np.sin(x*math.pi))
    h_5 = lambda x: np.sin(5*math.pi*x)/(5*np.sin(x*math.pi))
    h_7 = lambda x: np.sin(7*math.pi*x)/(7*np.sin(x*math.pi))
    h_9 = lambda x: np.sin(9*math.pi*x)/(9*np.sin(x*math.pi))

    t = np.linspace(0.001, 1.0, 64)

    plt.plot(t, h_3(t), label=r'$H_3$', c = 'black', linewidth=3)
    plt.plot(t, h_5(t), label=r'$H_5$', c = 'violet', linewidth=3)
    plt.plot(t, h_7(t), label=r'$H_7$', c = 'orange', linewidth=3)
    plt.plot(t, h_9(t), label=r'$H_9$', c = 'blue', linewidth=3)
    plt.xlabel('f', fontsize=20)
    plt.ylabel(r'$\widetilde{H}(f)$', fontsize=20)
    plt.legend()
    plt.show()

def task2():
    h_7 = lambda x: 1 / 21 * (7 + 12 * np.cos(2*pi*x) + 6 * np.cos(4*pi*x) - 4 *
np.cos(6*pi*x))
    h_9 = lambda x: 1 / 231 * (59 + 108 * np.cos(2*pi*x) + 78 * np.cos(4*pi*x) +
28 * np.cos(6*pi*x) - 42 * np.cos(8*pi*x))
    h_11 = lambda x: 1 / 429 * (89 + 168 * np.cos(2*pi*x) + 138 * np.cos(4*pi*x)
+ 88 * np.cos(6*pi*x) + 18 * np.cos(8*pi*x) - 72 * np.cos(
10*pi*x))
    h_13 = lambda x: 1 / 143 * (25 + 48 * np.cos(2*pi*x) + 42 * np.cos(4*pi*x) +
32 * np.cos(6*pi*x) + 18 * np.cos(8*pi*x) - 22 * np.cos(10*pi*x))

    t = np.linspace(0, 1, 64)
```

```

plt.plot(t, h_7(t), label=r'$H_7$', c = 'black', linewidth=3)
plt.plot(t, h_9(t), label=r'$H_9$', c = 'violet', linewidth=3)
plt.plot(t, h_11(t), label=r'$H_{11}$', c = 'orange', linewidth=3)
plt.plot(t, h_13(t), label=r'$H_{13}$', c = 'blue', linewidth=3)
plt.xlabel('f', fontsize=20)
plt.ylabel(r'$\widetilde{H}(f)$', fontsize=20)
plt.legend()
plt.show()

def task3():
    h_9 = lambda x: 1 / 429 * (179 + 270 * np.cos(2*pi*x) + 60 * np.cos(4*pi*x)
- 110 * np.cos(6*pi*x) + 30 * np.cos(8*pi*x))
    h_11 = lambda x: 1 / 429 * (143 + 240 * np.cos(2*pi*x) + 120 * np.cos(4*pi*x)
- 20 * np.cos(6*pi*x) - 90 * np.cos(8*pi*x) + 36 * np.cos(10*pi*x))
    h_13 = lambda x: 1 / 2431 * (677 + 1200 * np.cos(2*pi*x) + 780 * np.cos(4*pi*x)
+ 220 * np.cos(6*pi*x) - 270 * np.cos(8*pi*x) - 396 * np.cos(10*pi*x) + 220 *
np.cos(12*pi*x))
    h_15 = lambda x: 1 / 46189 * (11063 + 20250 * np.cos(2*pi*x) + 15000 *
np.cos(4*pi*x) + 7510 * np.cos(6*pi*x) - 330 * np.cos(8*pi*x) - 5874 *
np.cos(10*pi*x) - 5720 * np.cos(12*pi*x) + 4290 * np.cos(14*pi*x))

    t = np.linspace(0, 1, 64)

    plt.plot(t, h_9(t), label=r'$H_9$', c = 'black', linewidth=3)
    plt.plot(t, h_11(t), label=r'$H_{11}$', c = 'violet', linewidth=3)
    plt.plot(t, h_13(t), label=r'$H_{13}$', c = 'orange', linewidth=3)
    plt.plot(t, h_15(t), label=r'$H_{15}$', c = 'blue', linewidth=3)
    plt.xlabel('f', fontsize=20)
    plt.ylabel(r'$\widetilde{H}(f)$', fontsize=20)
    plt.legend()
    plt.show()

def task4():
    h_21 = lambda x: 1/320 * (74 + 134*np.cos(2*math.pi*x) +
92*np.cos(4*math.pi*x) + 42*np.cos(6*math.pi*x) + 6*np.cos(8*math.pi*x) -
10*np.cos(10*math.pi*x) - 12*np.cos(12*math.pi*x) - 6*np.cos(14*math.pi*x))
    h_15 = lambda x: 1/350 * (60 + 114*np.cos(2*math.pi*x) +
94*np.cos(4*math.pi*x) + 66*np.cos(6*math.pi*x) + 36*np.cos(8*math.pi*x) +
12*np.cos(10*math.pi*x) - 4*np.cos(12*math.pi*x) - 10*np.cos(14*math.pi*x) -
10*np.cos(16*math.pi*x) - 6*np.cos(18*math.pi*x) - 2*np.cos(20*math.pi*x))

```

```

t = np.linspace(0, 1, 64)

plt.plot(t, h_21(t), label=r'$H_{21}$', c = 'black', linewidth=3)
plt.plot(t, h_15(t), label=r'$H_{15}$', c = 'orange', linewidth=3)
plt.xlabel('f', fontsize=20)
plt.ylabel(r'$\widetilde{H}(f)$', fontsize=20)
plt.legend()
plt.show()

def task5():
    def _task1():
        h_3 = lambda x: np.sin(3 * math.pi * x) / (3 * np.sin(x * math.pi))
        h_5 = lambda x: np.sin(5 * math.pi * x) / (5 * np.sin(x * math.pi))
        h_7 = lambda x: np.sin(7 * math.pi * x) / (7 * np.sin(x * math.pi))
        h_9 = lambda x: np.sin(9 * math.pi * x) / (9 * np.sin(x * math.pi))

        t = np.linspace(0.001, 1, 64)

        plt.plot(t, 20*np.log10(np.abs(h_3(t))), label=r'$H_3$', c = 'black',
linewidth=3)
        plt.plot(t, 20*np.log10(np.abs(h_5(t))), label=r'$H_5$', c = 'violet',
linewidth=3)
        plt.plot(t, 20*np.log10(np.abs(h_7(t))), label=r'$H_7$', c = 'orange',
linewidth=3)
        plt.plot(t, 20*np.log10(np.abs(h_9(t))), label=r'$H_9$', c = 'blue',
linewidth=3)
        plt.xlabel('f', fontsize=20)
        plt.ylabel(r'$\widetilde{H}(f)$', fontsize=20)
        plt.legend()
        plt.show()

    def _task2():
        h_7 = lambda x: 1 / 21 * (7 + 12 * np.cos(2*pi*x) + 6 * np.cos(4*pi*x) -
4 * np.cos(6*pi*x))
        h_9 = lambda x: 1 / 231 * (59 + 108 * np.cos(2*pi*x) + 78 * np.cos(4*pi*x)
+ 28 * np.cos(6*pi*x) - 42 * np.cos(8*pi*x))
        h_11 = lambda x: 1 / 429 * (89 + 168 * np.cos(2*pi*x) + 138 * np.cos(4*pi*x)
+ 88 * np.cos(6*pi*x) + 18 * np.cos(8*pi*x) - 72 * np.cos(
10*pi*x))
        h_13 = lambda x: 1 / 143 * (25 + 48 * np.cos(2*pi*x) + 42 * np.cos(4*pi*x)
+ 32 * np.cos(6*pi*x) + 18 * np.cos(8*pi*x) - 22 * np.cos(10*pi*x))

```

```

t = np.linspace(0.001, 1, 64)

plt.plot(t, 20*np.log10(np.abs(h_7(t))), label=r'$H_7$', c = 'black',
linewidth=3)
plt.plot(t, 20*np.log10(np.abs(h_9(t))), label=r'$H_9$', c = 'violet',
linewidth=3)
plt.plot(t, 20*np.log10(np.abs(h_11(t))), label=r'$H_{11}$', c =
'orange', linewidth=3)
plt.plot(t, 20*np.log10(np.abs(h_13(t))), label=r'$H_{13}$', c = 'blue',
linewidth=3)
plt.xlabel('f', fontsize=20)
plt.ylabel(r'$\widetilde{H}(f)$', fontsize=20)
plt.legend()
plt.show()

def _task3():
    h_9 = lambda x: 1 / 429 * (179 + 270 * np.cos(2*pi*x) + 60 * np.cos(4*pi*x)
- 110 * np.cos(6*pi*x) + 30 * np.cos(8*pi*x))
    h_11 = lambda x: 1 / 429 * (143 + 240 * np.cos(2*pi*x) + 120 *
np.cos(4*pi*x) - 20 * np.cos(6*pi*x) - 90 * np.cos(8*pi*x) + 36 * np.cos(10*pi*x))
    h_13 = lambda x: 1 / 2431 * (677 + 1200 * np.cos(2*pi*x) + 780 *
np.cos(4*pi*x) + 220 * np.cos(6*pi*x) - 270 * np.cos(8*pi*x) - 396 *
np.cos(10*pi*x) + 220 * np.cos(12*pi*x))
    h_15 = lambda x: 1 / 46189 * (11063 + 20250 * np.cos(2*pi*x) + 15000 *
np.cos(4*pi*x) + 7510 * np.cos(6*pi*x) - 330 * np.cos(8*pi*x) - 5874 *
np.cos(10*pi*x) - 5720 * np.cos(12*pi*x) + 4290 * np.cos(14*pi*x))

t = np.linspace(0.001, 1, 64)

plt.plot(t, 20*np.log10(np.abs(h_9(t))), label=r'$H_9$', c = 'black',
linewidth=3)
plt.plot(t, 20*np.log10(np.abs(h_11(t))), label=r'$H_{11}$', c = 'violet',
linewidth=3)
plt.plot(t, 20*np.log10(np.abs(h_13(t))), label=r'$H_{13}$', c =
'orange', linewidth=3)
plt.plot(t, 20*np.log10(np.abs(h_15(t))), label=r'$H_{15}$', c = 'blue',
linewidth=3)
plt.xlabel('f', fontsize=20)
plt.ylabel(r'$\widetilde{H}(f)$', fontsize=20)
plt.legend()
plt.show()

```

```

def _task4():
    h_21 = lambda x: 1 / 320 * (74 + 134 * np.cos(2 * math.pi * x) + 92 *
np.cos(4 * math.pi * x) + 42 * np.cos(
        6 * math.pi * x) + 6 * np.cos(8 * math.pi * x) - 10 * np.cos(10 *
math.pi * x) - 12 * np.cos(
        12 * math.pi * x) - 6 * np.cos(14 * math.pi * x))
    h_15 = lambda x: 1 / 350 * (60 + 114 * np.cos(2 * math.pi * x) + 94 *
np.cos(4 * math.pi * x) + 66 * np.cos(
        6 * math.pi * x) + 36 * np.cos(8 * math.pi * x) + 12 * np.cos(10 *
math.pi * x) - 4 * np.cos(
        12 * math.pi * x) - 10 * np.cos(14 * math.pi * x) - 10 * np.cos(16 *
math.pi * x) - 6 * np.cos(
        18 * math.pi * x) - 2 * np.cos(20 * math.pi * x))

    t = np.linspace(0.001, 1, 64)

    plt.plot(t, 20*np.log10(np.abs(h_21(t))), label=r'$H_{21}$', c = 'black',
linewidth=3)
    plt.plot(t, 20*np.log10(np.abs(h_15(t))), label=r'$H_{15}$', c = 'orange',
linewidth=3)
    plt.xlabel('f', fontsize=20)
    plt.ylabel(r'$\widetilde{H}(f)$', fontsize=20)
    plt.legend()
    plt.show()

_task1()
_task2()
_task3()
_task4()

task1()
task2()
task3()
task4()

```