

### Лабораторная работа №3

В процессе прохождения сигнала от источника до получателя, последовательность данных претерпевает изменения и преобразования. Эти преобразования обусловлены прохождением сигнала через различные вычислительные устройства, реализующие ту или иную математическую задачу. В задачах цифровой обработки сигналов данные проходят через цифровые цепи, которые называются *фильтрами*. Цифровые фильтры, как и аналоговые, обладают различными характеристиками. Как правило, эти характеристики выражаются в частотном представлении - амплитудно-частотная и фазо-частотная характеристики фильтра. Цифровые фильтры используются в основном для улучшения качества сигнала - для выделения сигнала из последовательности данных, либо для ухудшения нежелательных сигналов - для подавления определенных сигналов в приходящих последовательностях отсчетов.

На сегодняшний день цифровые фильтры применяются практически везде, где требуется обработка сигналов. Даже примитивные математические операции (умножение и сложение) можно представить в виде цифрового фильтра. Следовательно, всё устройства, которые нас окружают (компьютеры, телефоны, телевизоры, и т.д.) содержат ряд тех или иных цифровых фильтров. Например, в телевизорах используются фильтры для преобразования выводимого изображения. Настройка яркости - по сути фильтрация изображения.

Таким образом, цифровые фильтры применяются в широком спектре задач обработки сигналов: спектральный анализ, обработка аудио- и видео-данных, обработка речи, движения и т.д. Применительно к задачам ЦОС, фильтр - это некоторая математическая система, которая изменяет форму входного сигнала (амплитуду, фазу, частоту).

#### Преимущества цифровых фильтров:

- Возможность получения характеристик, которые невозможно получить аналоговыми методами (например, точную линейную ФЧХ).
- Цифровые фильтры стабильнее к внешним изменениям среды (температура, влажность, давление). Следовательно, цифровые фильтры не требуют постоянной калибровки.
- Простота перестройки частоты обработки сигнала.
- В зависимости от скорости обработки и частоты сигнала, один фильтр может одновременно обрабатывать данные с нескольких каналов.
- Данные до и после фильтра можно сохранить на любой носитель. Аналоговые сигналы записать сложнее (в любом случае требуется перевод в цифровую форму и обратно).
- Низкая потребляемая мощность и низкая стоимость относительно аналоговых фильтров.
- Повышенная точность вычисления (современные цифровые устройства не ограничены в точности вычисления).

- Простота проектирования фильтров со сложными частотными характеристиками.
- В задачах биомедицины применение аналоговых фильтров на очень низких частотах затруднительно, в отличие от цифровых фильтров.

### Недостатки цифровых фильтров

- **Ограниченная разрядность.** В процессе вычисления в цифровых фильтрах с конечной разрядностью накапливаются шумы квантования, шумы округления. Одна из типовых задач ЦОС - выбор подходящей разрядности фильтра.
- **Ограничение скорости обработки.** Как правило, аналоговые фильтры способны обрабатывать данные на очень больших частотах. Цифровые фильтры ограничены частотой дискретизации.
- **Аппаратное обеспечение.** Поскольку реальные сигналы в природе - непрерывные, для их обработки используются АЦП и ЦАП. От характеристик этих устройств также зависит качество получаемых цифровых (аналоговых) сигналов.

Сигналы и фильтры в зависимости способа представления характеризуются разными функциями:

Сигнал	Фильтр
отсчеты $x(n)$	импульсная характеристика $h(n)$
z-форма $X(z)$	передаточная функция $H(z)$
спектральная плотность $X(e^{j\phi})$	частотная характеристика $H(e^{j\phi})$
модуль спектра $ X(e^{j\phi}) $	АЧХ $ H(e^{j\phi}) $
аргумент спектра $\arg(X(e^{j\phi}))$	ФЧХ $\arg(H(e^{j\phi}))$

**Импульсной характеристикой**  $h(n)$  называется реакция цепи на входное воздействие в виде функции Дирака ( $\delta$ -функции). Она отражает влияние цепи на сигнал. В цифровых устройствах импульсная характеристика может быть конечной или бесконечной. Следовательно, существует два класса цифровых фильтров - с конечной импульсной характеристикой и с бесконечной импульсной характеристикой.

Цифровые фильтры описываются разностным уравнением:

$$y(k) = \frac{1}{a_0} \cdot (\sum_{k=0}^N b_k x(n - k) - \sum_{k=0}^M a_k y(n - k))$$

В этой формуле  $y(k)$  - выходное воздействие или отсчеты на выходе цифрового фильтра.  $x(k)$  - входной сигнал.  $a_k$  и  $b_k$  - множители (коэффициенты) цифровых звеньев на входе и выходе. Эти множители также называются коэффициентами числителя и знаменателя передаточной характеристики фильтра.

Передаточная характеристика фильтра имеет следующий вид:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + \dots + a_M z^{-M}}$$

где  $N$  и  $M$  - количество линий задержки для входного и выходного сигналов. В общем виде это формула БИХ-фильтра. Если знаменатель равен единице, то формула соответствует выражению для КИХ-фильтра (цифровой фильтр без обратной связи).

По реализации цифрового фильтра различают два метода: аппаратный и программный. Аппаратные фильтры реализуются на ПЛИС или специализированных сигнальных процессорах. Программные фильтры реализуются с помощью программ, выполняемых процессором или микроконтроллером.

Фильтры могут быть рекурсивными, когда выходной отсчет зависит от предыдущих выходных и от входного, т.е. в схеме присутствует обратная связь, или нерекурсивными, когда выходной отсчет зависит только от входных отсчетов.

### БИХ и КИХ фильтры

**КИХ фильтр** (англ. FIR — «finite impulse response») - это цифровой фильтр с конечной импульсной характеристикой. Импульсная характеристика такого фильтра ограничена во времени, то есть имеет счётное число коэффициентов. Начиная с определенного момента времени она становится равной нулю. Если на вход КИХ-фильтра подать единичный импульс, на выходе фильтра будет конечное число отсчетов. Как правило, ФЧХ фильтра с конечной импульсной характеристикой - линейна

В общем случае, КИХ-фильтры реализуются без обратных связей, то есть они нерекурсивные. Однако, с помощью математических преобразований можно привести фильтр к рекурсивной форме.

Разностное уравнение КИХ-фильтра:

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

где

- $y(n)$  - выходной дискретный сигнал (сумма взвешенных входных импульсов),
- $x(n)$  - входной дискретный сигнал (последовательность отсчётов),
- $h(n)$  - коэффициенты импульсной характеристики фильтра,
- $N$  - длина (порядок) фильтра.

Передаточная характеристика КИХ-фильтра:

$$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k}$$

Напомним, что операция  $z^{-k}$  - задержка последовательности на  $k$ -отсчетов.

**БИХ фильтр** (англ. IIR — «infinite impulse response») - это цифровой фильтр с бесконечной во времени импульсной характеристикой, то есть имеет очень большое или бесконечное число коэффициентов. БИХ фильтры также называют рекурсивными в связи с тем, что при их реализации используются обратные связи (сигнал с выхода фильтра через элементы задержки поступает на фильтр и вносит изменения сам в себя). Передаточная функция БИХ-фильтра имеет дробно-рациональный вид. Основные известные БИХ-фильтры: фильтр Чебышева, Баттерворта, Калмана, Бесселя и т.д.

Разностное уравнение БИХ-фильтра:

$$y(n) = \sum_{k=0}^N b_k x(n-k) - \sum_{k=1}^M a_k y(n-k)$$

Или

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_M x(n-M) + \dots - a_1 y(n-1) - a_2 y(n-2) - \dots - a_N y(n-N)$$

Как видно, выходной сигнал за счет обратных связей влияет сам на себя.

Передаточная характеристика БИХ-фильтра:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + \dots + a_M z^{-M}}$$

В отличие от КИХ фильтров, БИХ фильтры не всегда являются устойчивыми. Для устойчивости цифрового БИХ фильтра требуется, чтобы все полюса передаточной характеристики по модулю были строго меньше единицы (то есть лежали внутри единичной окружности на  $z$ -плоскости).

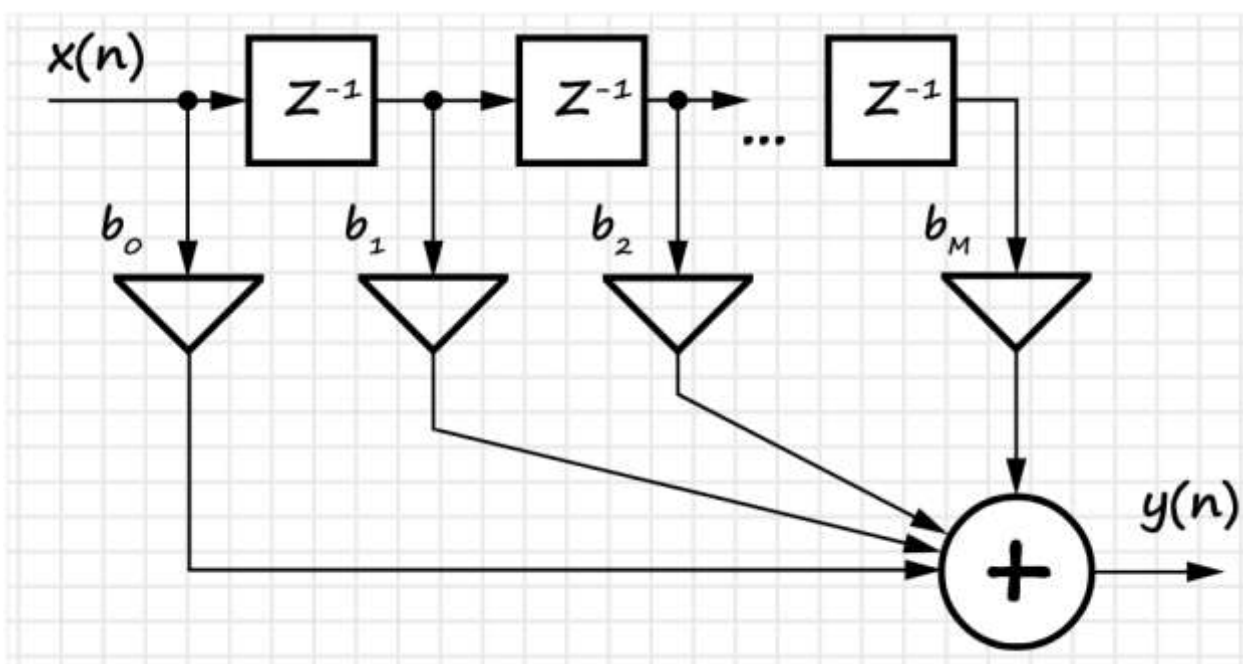
**Порядок фильтра** – максимальная степень в выражении передаточной функции  $H(z)$ .

Передаточная функция может быть реализована при помощи различных структурных схем. Наиболее распространены последовательные и параллельные структурные схемы. Если передаточную дробно-рациональную функцию рекурсивного фильтра представить в виде произведения передаточных функций первого и второго порядка, то структурная схема будет *последовательной*. Если передаточную функцию рекурсивного фильтра представить в виде суммы передаточных функций первого и второго порядка, то структурная схема будет *параллельной*.

Каноническая и прямая схема

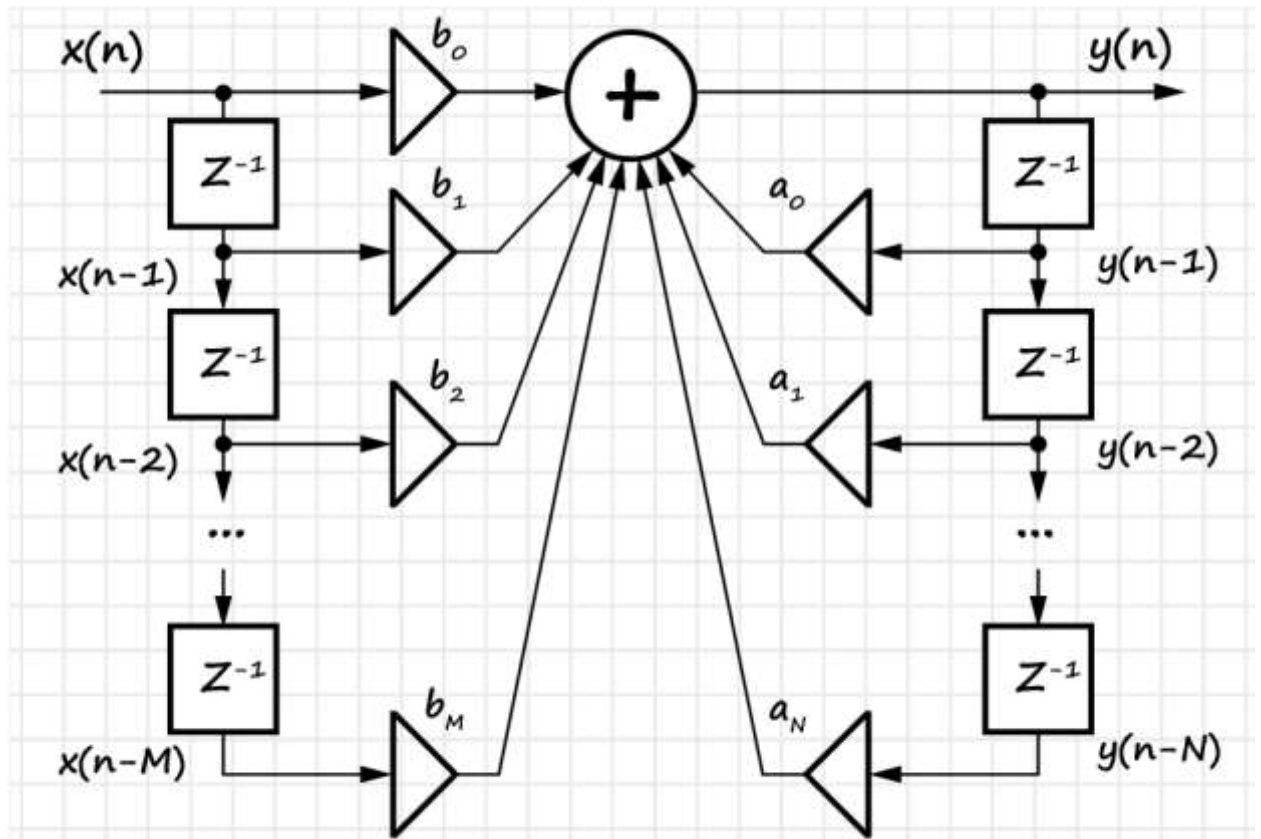
Структурную схему цифрового фильтра можно представить в прямой и канонической форме. Цифровые фильтры могут быть реализованы с использованием трех цифровых элементов: умножитель, сумматор и блок задержки. Очевидно, что умножители требуются для умножения отсчётов на коэффициенты  $a_k$  и  $b_k$ , а сумматор объединяет полученные результаты.

Структурная схема нерекурсивного КИХ-фильтра, согласно уравнению:



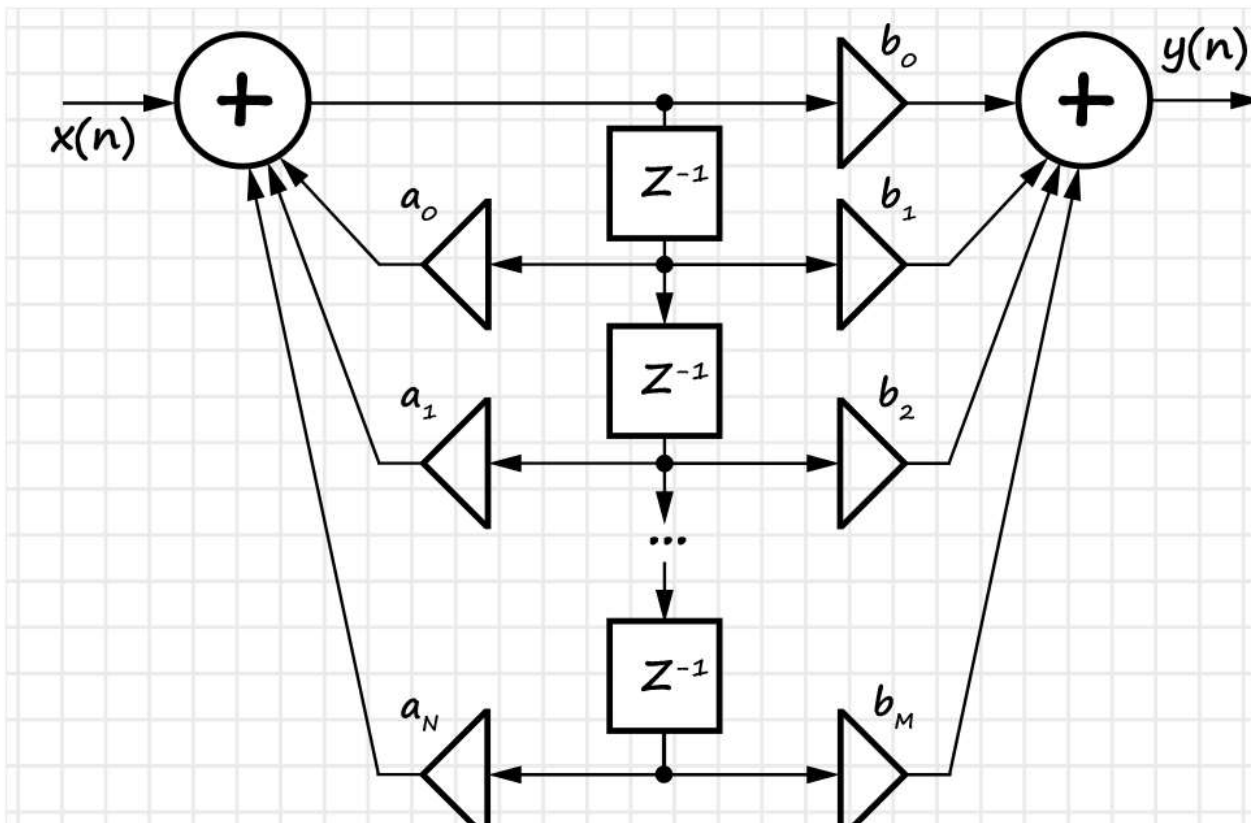
Количество линий задержки всегда на единицу меньше, чем число коэффициентов или порядок фильтра. Импульсная характеристика КИХ фильтра совпадает с набором коэффициентов этого фильтра.

Структурная схема БИХ-фильтра выглядит несколько сложнее из-за наличия обратных связей:



Количество линий задержки БИХ фильтра равно  $N+MN+M$ .

Структурная схема в **каноническом виде** минимизирует количество линий задержки, поскольку использует общие линии задержки. Входной сумматор - с обратными связями. Выходной сумматор - обобщает результат. Преобразование от одной формы к другой достаточно просто и тривиально



### Выбор фильтра

Перед разработчиками цифровых устройств часто встает задача выбора типа фильтра. Выбор между КИХ и БИХ фильтрами производится по следующим критериям:

- Фазовая характеристика БИХ фильтра - нелинейна. КИХ фильтры могут иметь строго линейную фазовую характеристику. Это означает, что такой фильтр не вносит искажений в форму сигнала.
- КИХ фильтры - устойчивы, т.к. реализуются по нерекурсивной форме. БИХ фильтры могут быть неустойчивыми.
- В реальных задачах стоит вопрос о выборе разрядности данных (округление, квантование). Эти процессы вносят меньший вклад в устройства без обратных связей, то есть в КИХ фильтры.
- Реализация АЧХ сложной формы или максимально прямоугольной формы потребует значительного числа коэффициентов КИХ фильтра. БИХ фильтры с этой задачей справляются лучше.
- У КИХ фильтров, как правило, нет эквивалентных аналоговых фильтров.
- КИХ фильтры позволяют легко получать требуемые характеристики (уровень затухания, неравномерности в полосе пропускания, частота среза и т.д.)
- БИХ фильтры существенно экономнее по количеству операций умножения, сложения и количеству линий задержки.

## БИХ-фильтры (SCIPY / MATLAB)

Ниже представлен список основных функций БИХ-фильтров для реализации в python с помощью пакета сигналов - `scipy.signal`. Большинство функций имеют аналогичные MATLAB-прототипы.

Function	Description
<code>iirdesign(wp, ws, gpass, gstop[, analog, ...])</code>	Функция для полного расчёта цифрового фильтра. Возвращает коэффициенты <code>a</code> , <code>b</code> , нули и полюсы и т.д.
<code>iirfilter(N, Wn[, rp, rs, btype, analog, ...])</code>	Реализация БИХ-фильтра выбранного типа и <code>N</code> -порядка. Возвращает коэффициенты <code>a</code> , <code>b</code> , нули и полюсы и т.д.
<code>butter(N, Wn[, btype, analog, output, fs])</code>	Фильтр Баттерворта. Реализует фильтр <code>N</code> -порядка и возвращает коэффициенты фильтра.
<code>buttord(wp, ws, gpass, gstop[, analog, fs])</code>	Возвращает минимальный порядок, требуемый для реализации фильтра
<code>cheby1(N, rp, Wn[, btype, analog, output, fs])</code>	Фильтр Чебышева 1 типа. Реализует фильтр <code>N</code> -порядка и возвращает коэффициенты фильтра.
<code>cheb1ord(wp, ws, gpass, gstop[, analog, fs])</code>	Возвращает минимальный порядок, требуемый для реализации фильтра
<code>cheby2(N, rs, Wn[, btype, analog, output, fs])</code>	Фильтр Чебышева 2 типа. Реализует фильтр <code>N</code> -порядка и возвращает коэффициенты фильтра.
<code>cheb2ord(wp, ws, gpass, gstop[, analog, fs])</code>	Возвращает минимальный порядок, требуемый для реализации фильтра



Function	Description
<code>ellip(N, rp, rs, Wn[, btype, analog, output, fs])</code>	Эллиптический БИХ-фильтр (Кауэра)
<code>ellipord(wp, ws, gpass, gstop[, analog, fs])</code>	Возвращает минимальный порядок, требуемый для реализации фильтра от значений входных аргументов
<code>bessel(N, Wn[, btype, analog, output, norm, fs])</code>	БИХ-фильтр Бесселя.
<code>iirnotch(w0, Q[, fs])</code>	<b>Заграждающий</b> фильтр. Возвращает коэффициенты а, b. Входные аргументы - частота сигнала, добротность и частота среза
<code>iirpeak(w0, Q[, fs])</code>	<b>Полосовой</b> фильтр. Функция возвращает коэффициенты а, b фильтра второго порядка. Входные аргументы - частота сигнала, добротность и частота среза

## КИХ фильтр Кайзера

Покажем прохождение зашумленного гармонического сигнала через КИХ фильтр.

## Расчеты

Импортируем библиотеки

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import lfilter, freqz, butter, firls, remez, firwin, firwin2, group_delay
from scipy.fftpack import fft, fftshift
```

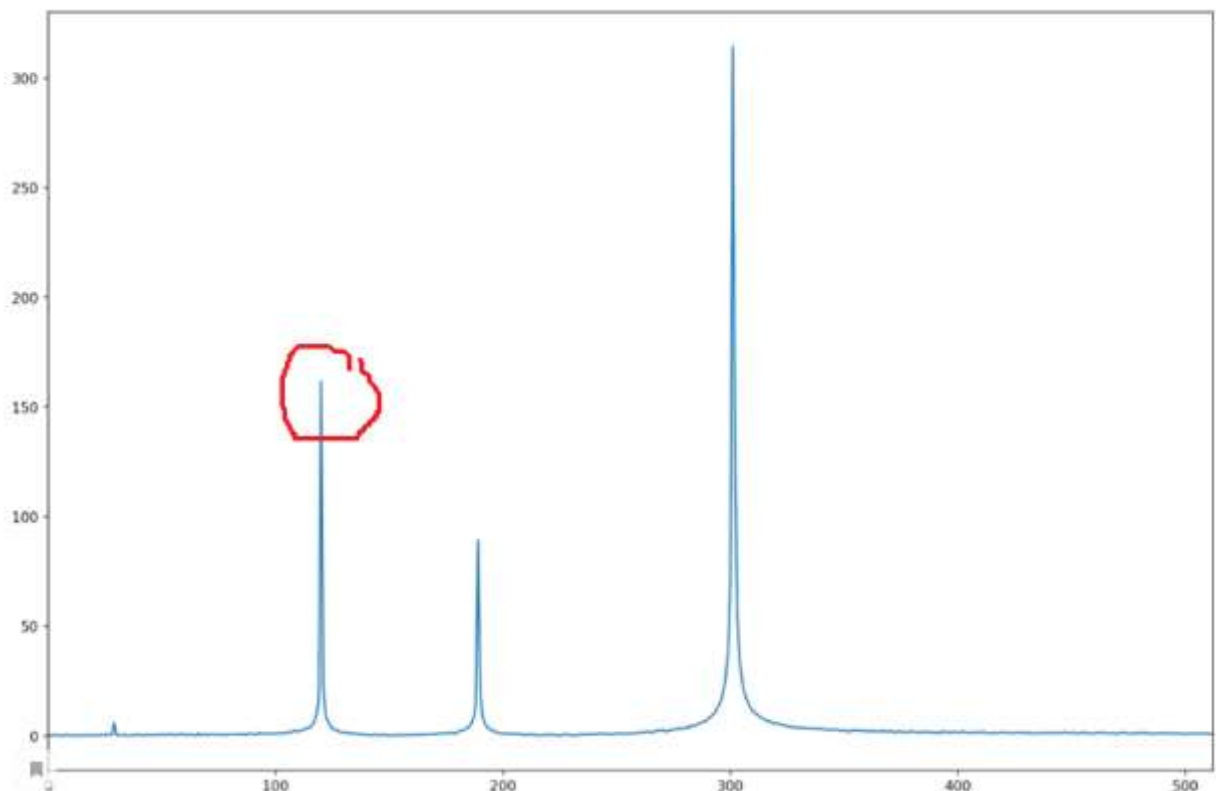
Читаем данные с файла

```
In [2]: data1 = np.genfromtxt("file3.txt", delimiter='\t', dtype=float)
```

Строим спектр

```
In [30]: clc_fft = np.abs(fft(data1, N))/N*2
plt.figure(figsize=(15, 10), dpi=200)
plt.plot(clc_fft)
plt.xlim([0, N/2])
```

Выделяем нужную частоту



## Применяем фильтрацию к сигналу

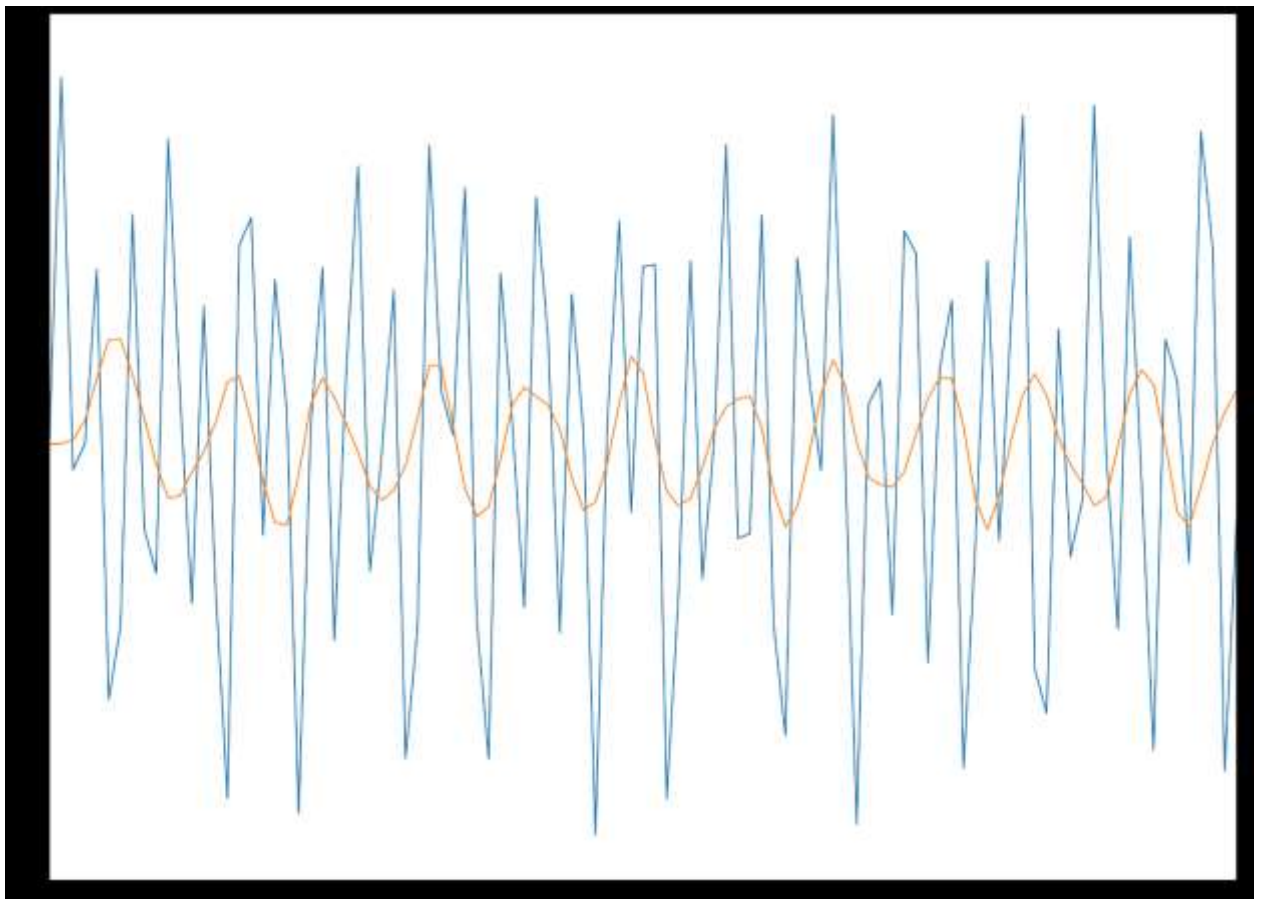
`firwin(numtaps, cutoff[, width, window, ...])` Реализация КИХ-фильтра N-порядка с применением оконной функции. За частота среза.

Метод `firwin`

`numtaps` - порядок фильтра,  
`cutoff` - частота среза относительно частоты дискретизации `fs`.  
`width` - если задано значение, то показывает ширину перехода от полосы пропускания до полосы подавления.  
`window` - оконная функция. Применяется в совокупности с методом `get_window` или напрямую задается название окна.  
`fs` - частота дискретизации или частота Найквиста. Если значение не задано, по умолчанию равно единице.

```
In [27]: # Kaiser FIR filter
taps = 10
h = firwin(taps, 0.13, window=('kaiser', 9))
y = lfilter(h, 1, data1)
plt.figure(figsize=(20, 15), dpi=200)
plt.plot(x)
plt.plot(y)
plt.xlim([0, 100])
```

## Получаем отфильтрованный сигнал



**Задание:** согласно номеру варианта загрузить сигнал из файла, отфильтровать сигналы для всех явных частот на спектре и построить графики выделенных сигналов. Параметры фильтров подобрать эмпирически.