**Лабораторная работа №6**

**Липатов Данила Вячеславович**

**МСМТ243**

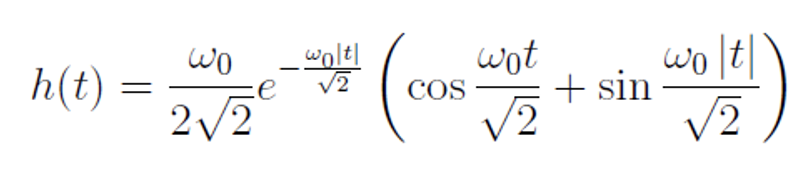
**Задача:**

Применить к модели вашего сигнала из ЛР 1 фильтр Пантелеева, запрограммировав свертку во временной области. Полуширину фильтра выбрать так, чтобы прошли низкочастотные гармоники, а высокочастотные были подавлены. Показать сравнение исходного и отфильтрованного сигналов, а также их спектров, отрисовав также АЧХ фильтра.

Сигнал из ЛР 1, как и ранее, задается следующим образом:

a = 18 # day  
b = 10 # month  
c = 2002 # year  
  
# Periods (in years)  
T1 = 0.5  
T2 = 1  
T3 = 4.6  
  
# Harmonic amplitudes, normalized  
A1 = (a / 31) \* 20  
A2 = (b / 12) \* 20  
A3 = ((c - 2000) / 50) \* 20  
N = 1024  
t = np.linspace(0, N \* 0.05, N)  
print(t)  
  
phi1 = np.pi / 2  
phi2 = np.pi / 2  
phi3 = np.pi / 2  
  
X\_model = (A1 \* np.cos(2 \* np.pi \* t / T1 + phi1) +  
 A2 \* np.cos(2 \* np.pi \* t / T2 + phi2) +  
 A3 \* np.cos(2 \* np.pi \* t / T3 + phi3))

Далее необходимо реализовать функцию, которая будет возвращать импульсную характеристику, которая далее будет использоваться в свертке. Функция имеет вид:



def pantelleev\_filter(t, omega\_0):  
 *"""  
 Pulse response of Panteleev filter.  
  
 Parameters:  
 t (array): Initial dates.  
 omega\_0 (int/float): half-width parameter ω0  
  
 Returns:  
 array:  
 - Impulse response of the filter  
 """* coef = omega\_0 / (2 \* np.sqrt(2))  
 exp\_part = np.exp(-omega\_0 \* np.abs(t) / np.sqrt(2))  
 cos\_part = np.cos(omega\_0 \* t / np.sqrt(2))  
 sin\_part = np.sin(omega\_0 \* np.abs(t) / np.sqrt(2))  
 return coef \* exp\_part \* (cos\_part + sin\_part)

И отрисуем графики исходного и отфильтрованного сигнала (рис. 1)

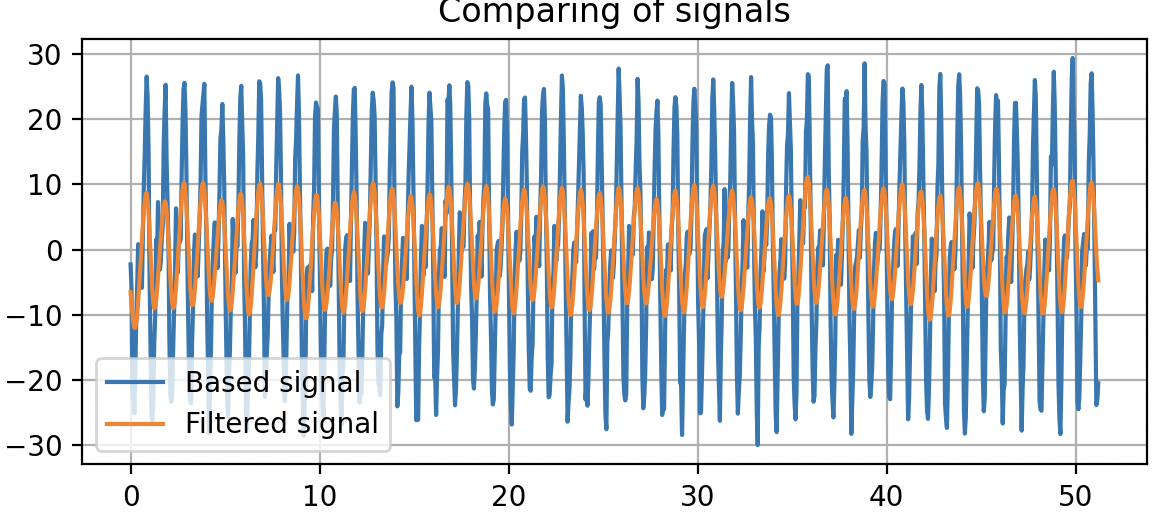


Рис. 1

Построим так же спектрограмму для исходного сигнала и отфильтрованного отдельно (рис.2 – рис.3)

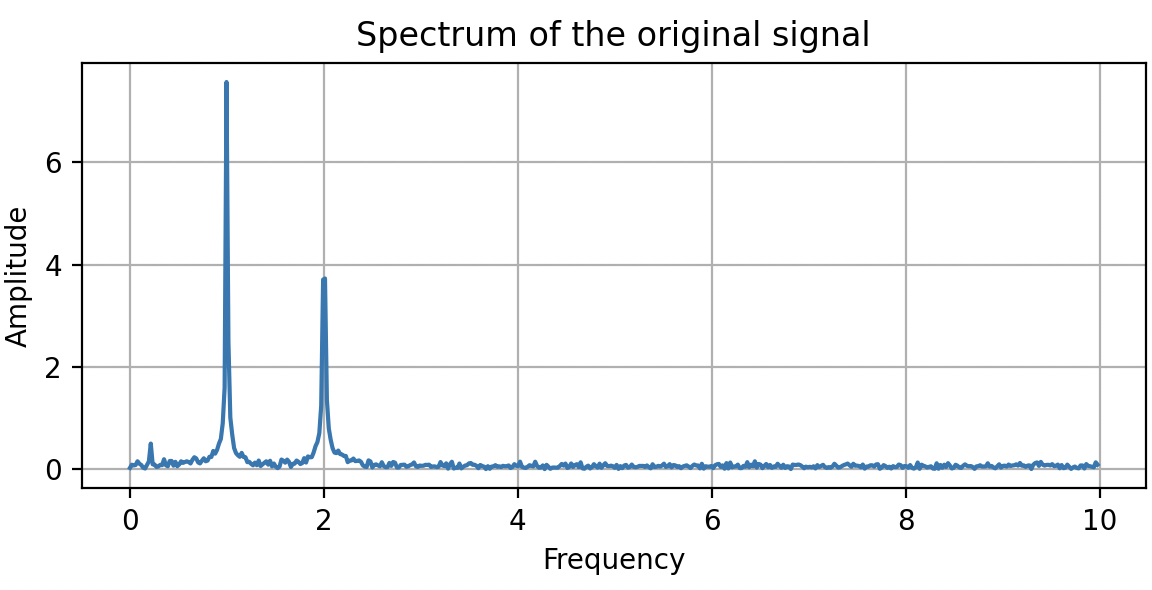


Рис. 2

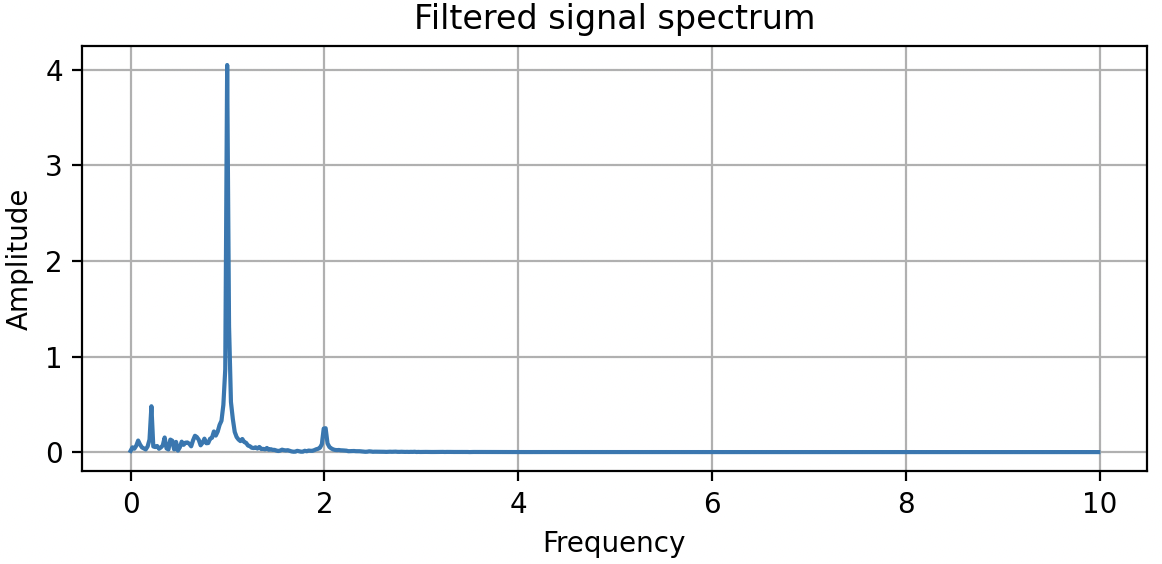


Рис. 3

Для того, чтобы было понятнее, насколько фильтр Пантелеева хорошо пропускает низкочастот. гармоники и занижает высокочастот. применим шум на исходный сигнал. Наложим теперь один на другой и посмотрим, что получится (рис. 4)

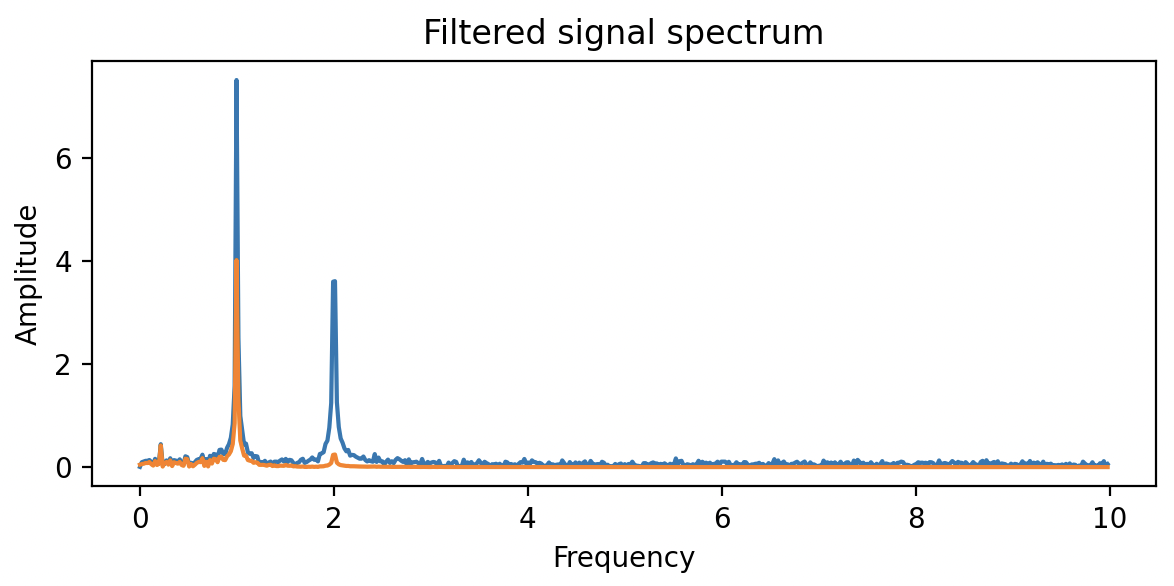
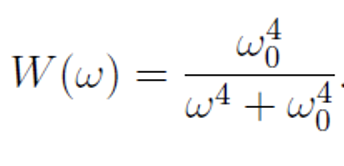


Рис. 4

В целом, мы получили, что фильтр и вправду занижает высокие частоты и пропускает низкие.

Для построения АЧХ возьмем формулу и перепишем ее в виде функции



def pantelleev\_ach(omega, omega\_0):  
 *"""  
 Amplitude-frequency response (AFR) of the Panteleev filter.  
 How does the amplitude 𝑊(𝜔) change with increasing frequency 𝜔  
  
 Parameters:  
 omega(t) (array): Initial dates.  
 omega\_0 (int/float): half-width parameter ω0  
  
 Returns:  
 array:  
 - AFR  
 """* return (omega\_0\*\*4) / (omega\*\*4 + omega\_0\*\*4)

Теперь отрисуем АЧХ (рис. 5)

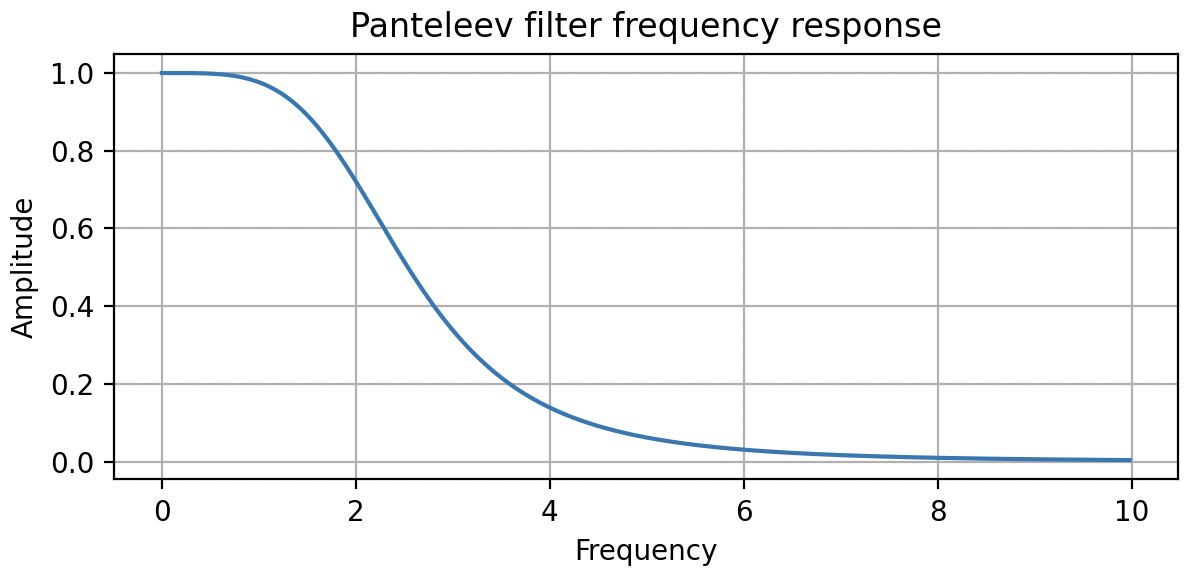


Рис. 5

**Итог**

В данном работе было реализовано применение фильтра Пантелеева и свертки, подбор ω0 - параметр полуширины. Для параметра полуширины = 6.5 мы получили данные результаты, если занизить или завысить данный параметр, то мы можем регулировать какие частоты будут пропускать через фильтр. Наглядно их можно посмотреть на спектрограмме, сопоставив спектрограммы отфильтрованного сигнала и исходного.