**Теория Фильтрации и Прогнозирование данных**

Липатов Данила МСМТ 243

Лабораторная работа №1

**Пункт 1**

Графики считанных данных из файла ‘AAMWPgfz0.050-year.dat’ X, Y относительно времени, выраженных в годах с шагом 18 суток или с шагом .

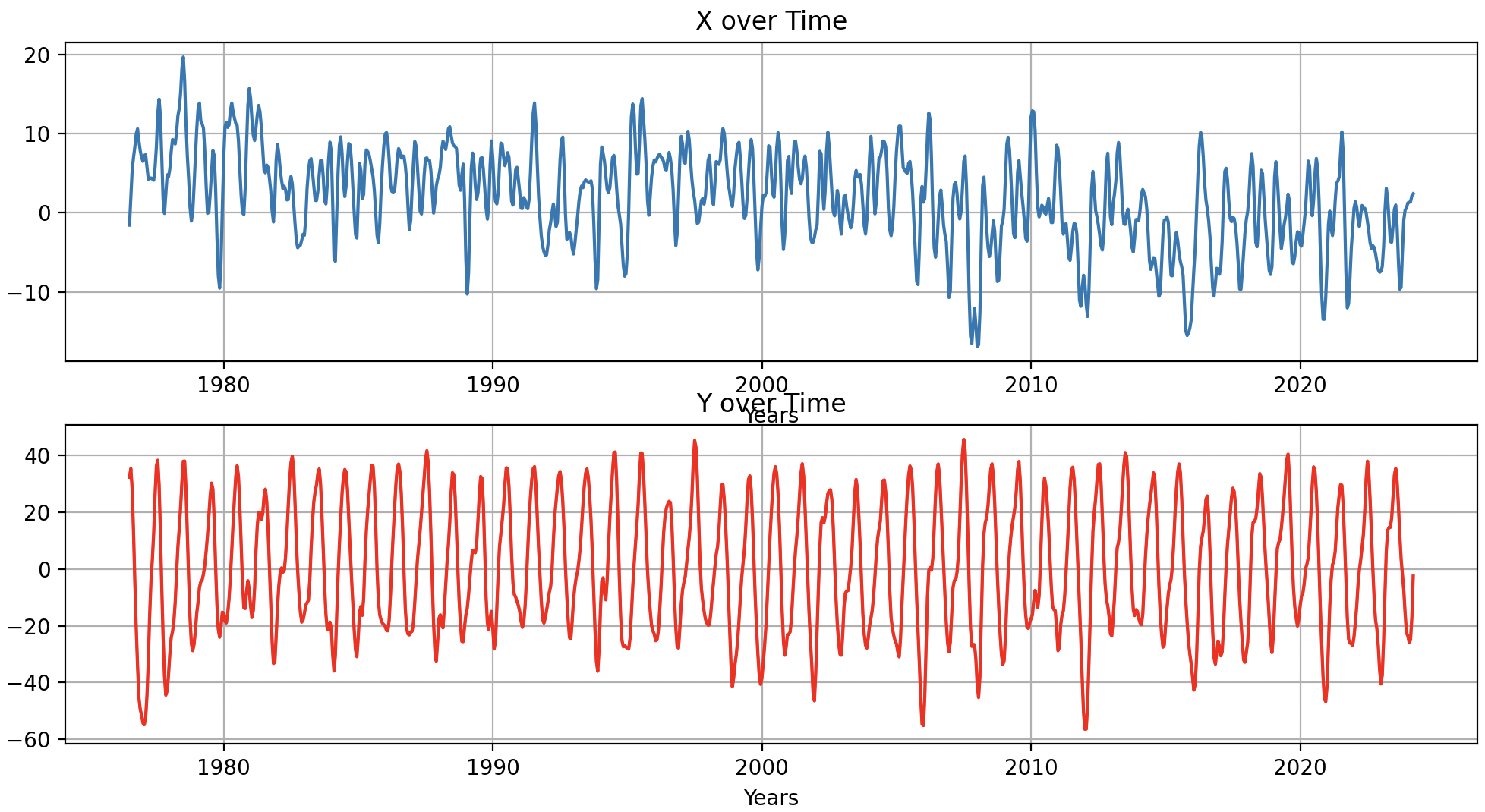


Рис. 1 Графики X , Y отн-но времени в годах.

Далее применялось БПФ на исходные данные относительно Х и Y и нормировались на кол-во элементов в массиве (т.к. данных по Х и по Y одинаковое кол-во, то нет разницы длину какого массива выбирать)

Код для БПФ на ЯП Python представлен ниже

X\_fft = np.fft.fft(X) / N

Y\_fft = np.fft.fft(Y) / N

freqs = np.fft.fftfreq(N, dT)

Полученные спектрограммы (ограниченные положительной частью) представлены на Рис. 2, стоит отметить, что частота выражена через циклическую частоту , где измеряется как рад / год (или для – цикл / год)

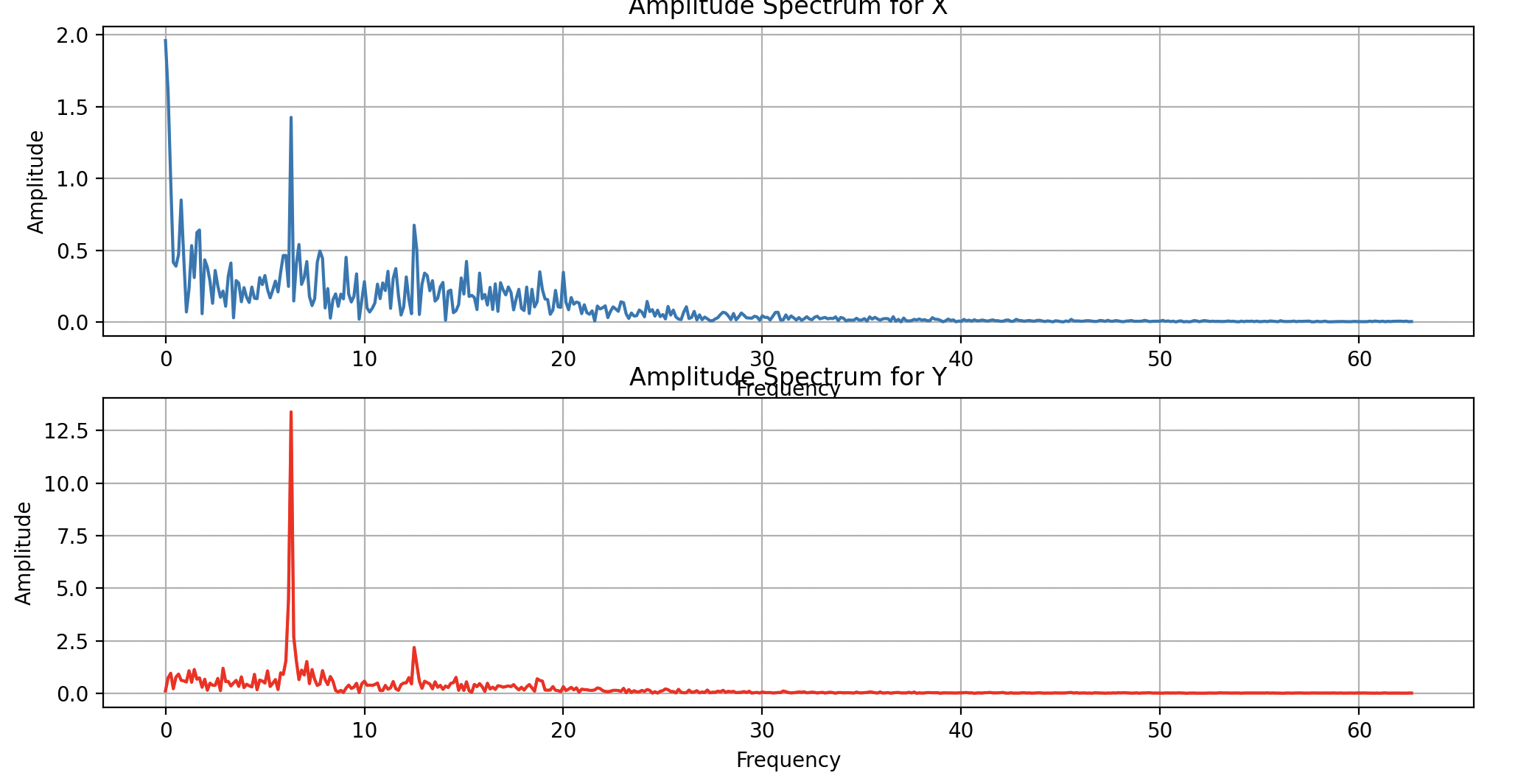


Рис. 2 Спектрограммы для Х и Y

На Рис.3 представлены графики периодограмм:

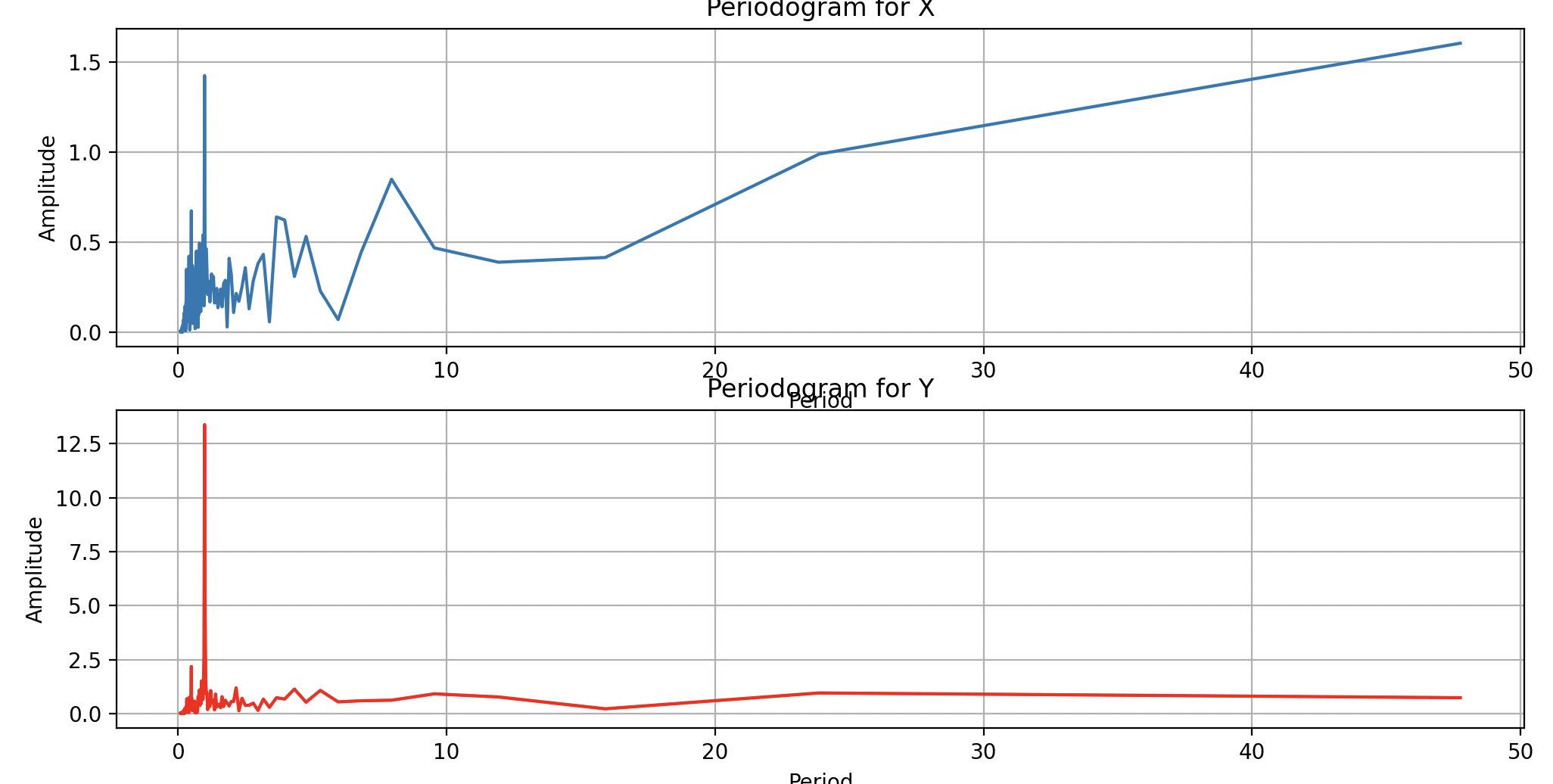


Рис. 3 Периодограммы для Х и Y

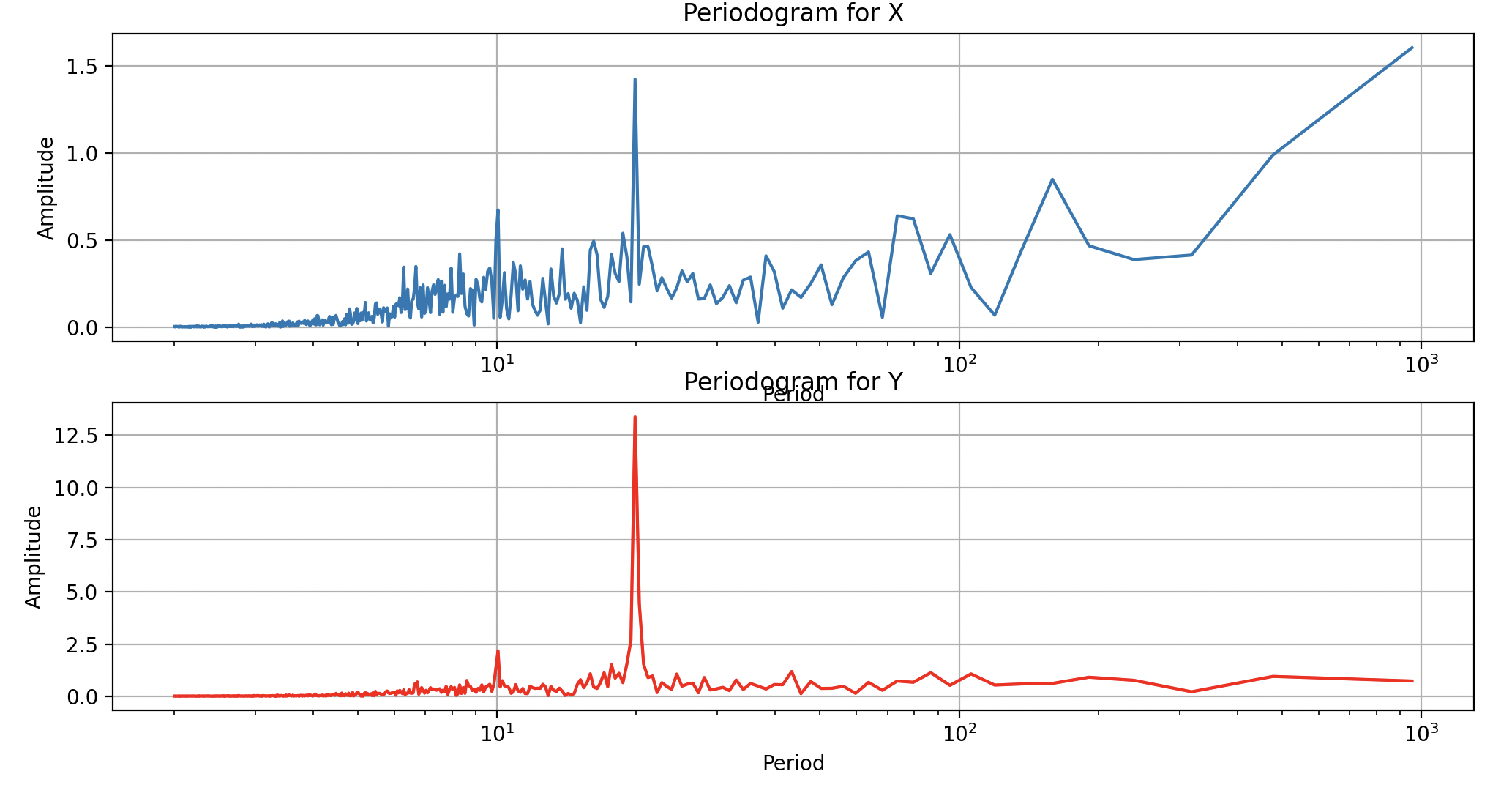


Рис. 3.2 Периодограммы для Х и Y с лог. шкалой

Далее необходимо было объединить X, Y в комлексный ряд и так же найти спектрограмму

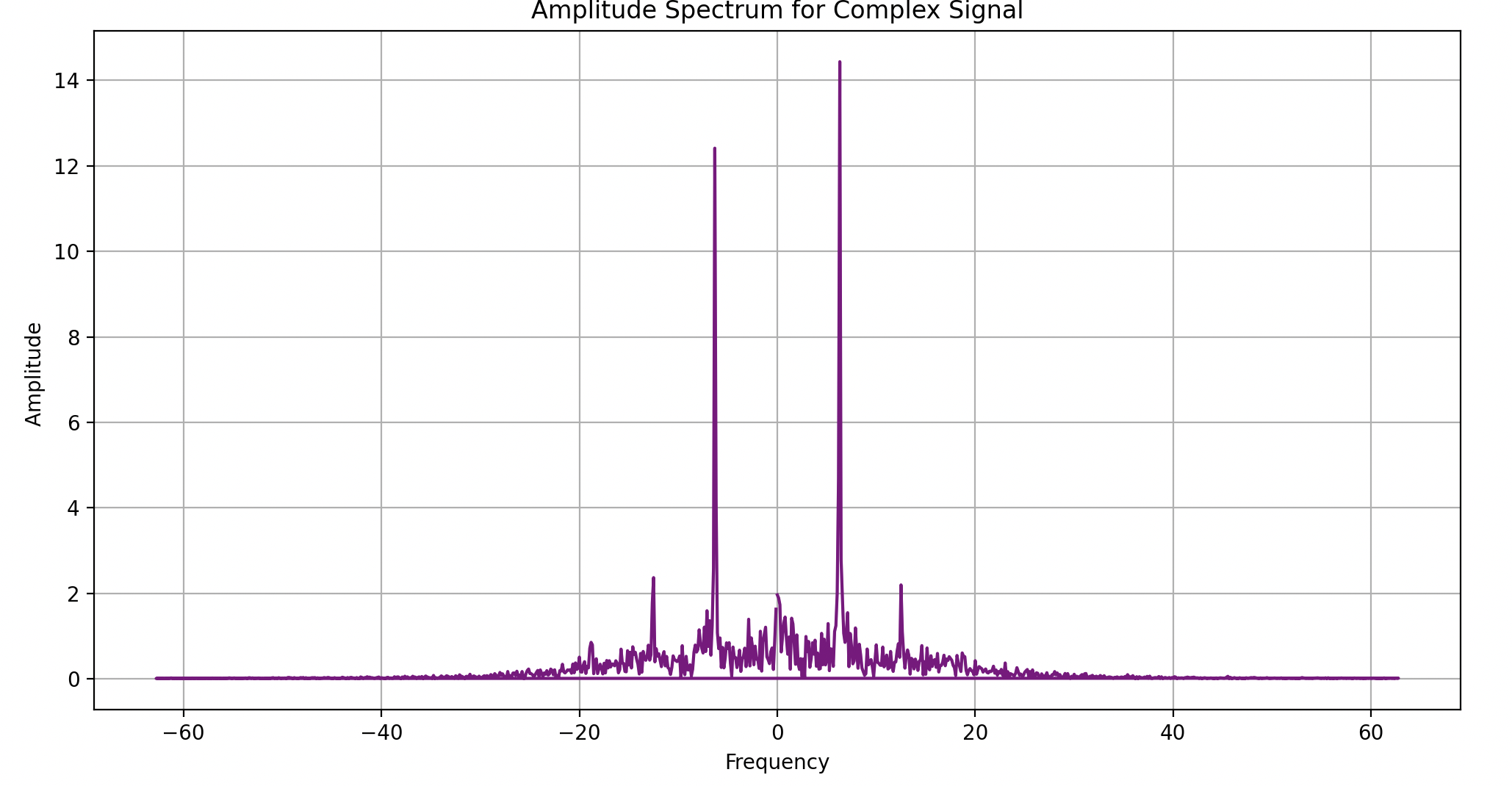


Рис. 4 Спектрограммы для комплексного ряда

Аналогично X, Y находим периодограмму комплексного ряда, представлен на Рис. 5

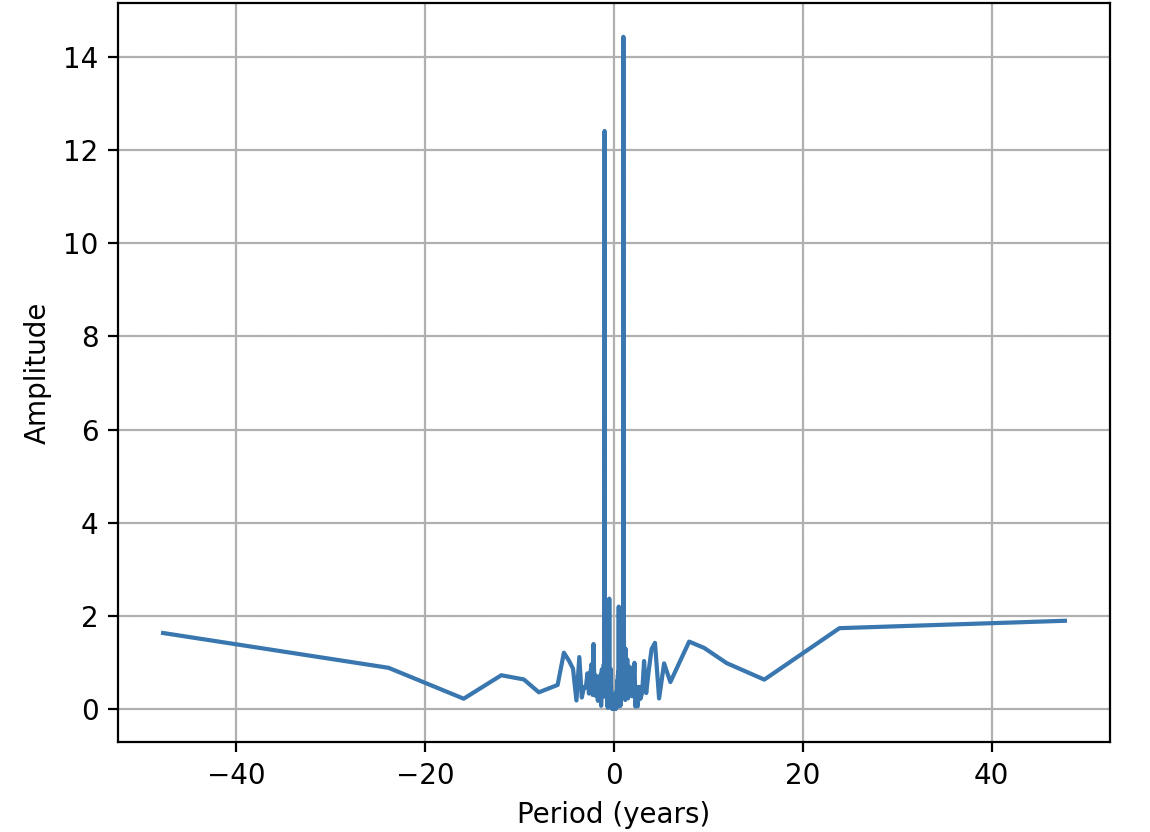


Рис. 5 Периодограммы для комплексного ряда

**Пункт 2**

Для нахождения Сk и С-k было принято найти максимальную амплитуду в представленной спектрограмме комплексного сигнала

Через формулу (1.6)

Находилось значение и его индекс в массиве, соотвественно:

fft\_k (9.5725+10.808i)

Dominant frequency: 6.31 rad / year

Amplitude for positive frequency (): 14.437969212443978

Откуда фаза получилась равной:

Phase for positive frequency (): 0.8459 rad

Аналогично для ретроградной

fft\_-k (-6.74906+10.4199i)

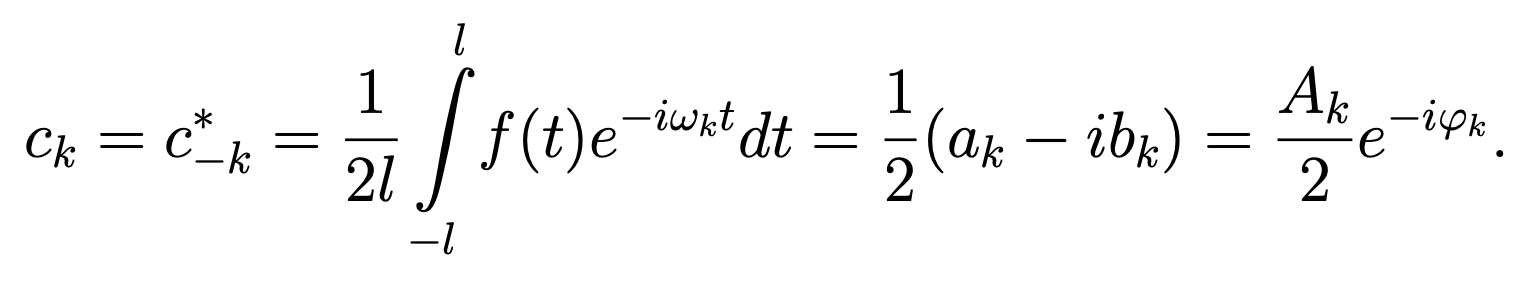
Dominant frequency: -6.32 rad / year

Amplitude for negative frequency (): 12.414712572558162

(Т.к. Re(z) < 0, Im(z) > 0)

Phase for negative frequency (): 2.14555676711411 rad

Найдем Сk и С-k по формуле 1.8, получим следующие результаты:



C\_k (4.81064 -5.36691i)

C\_-k (-3.34553 -5.22865i)

Оказывается, прямая реализация БПФ средствами доп модулей Python (numpy) считает элементы по определению r \* exp(i\*theta)

Если рассмотреть данные результаты как r / 2 \* exp(-i \* theta) из приведенных выше вычислений, то результаты сойдутся.

**Пункт 3**

Для моделирование собственного сигнала на основе входных данных( где a, b, c дата, месяц и год рождения)

Необходимо было отнормировать амплитуды гармоник, согласно задаче и отмасштабировать их.

# Модельный сигнал как сумма трёх гармоник  
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))

Где

# Периоды (в годах)  
T1 = 0.5  
T2 = 1  
T3 = 4.6  
  
# Амплитуды гармоник, нормализованные  
A1 = (a / 31) \* 20  
A2 = (b / 12) \* 20  
A3 = ((c - 2000) / 50) \* 20

phi1 = np.pi / 2  
phi2 = np.pi / 2  
phi3 = np.pi / 2

Получены следующие графики:

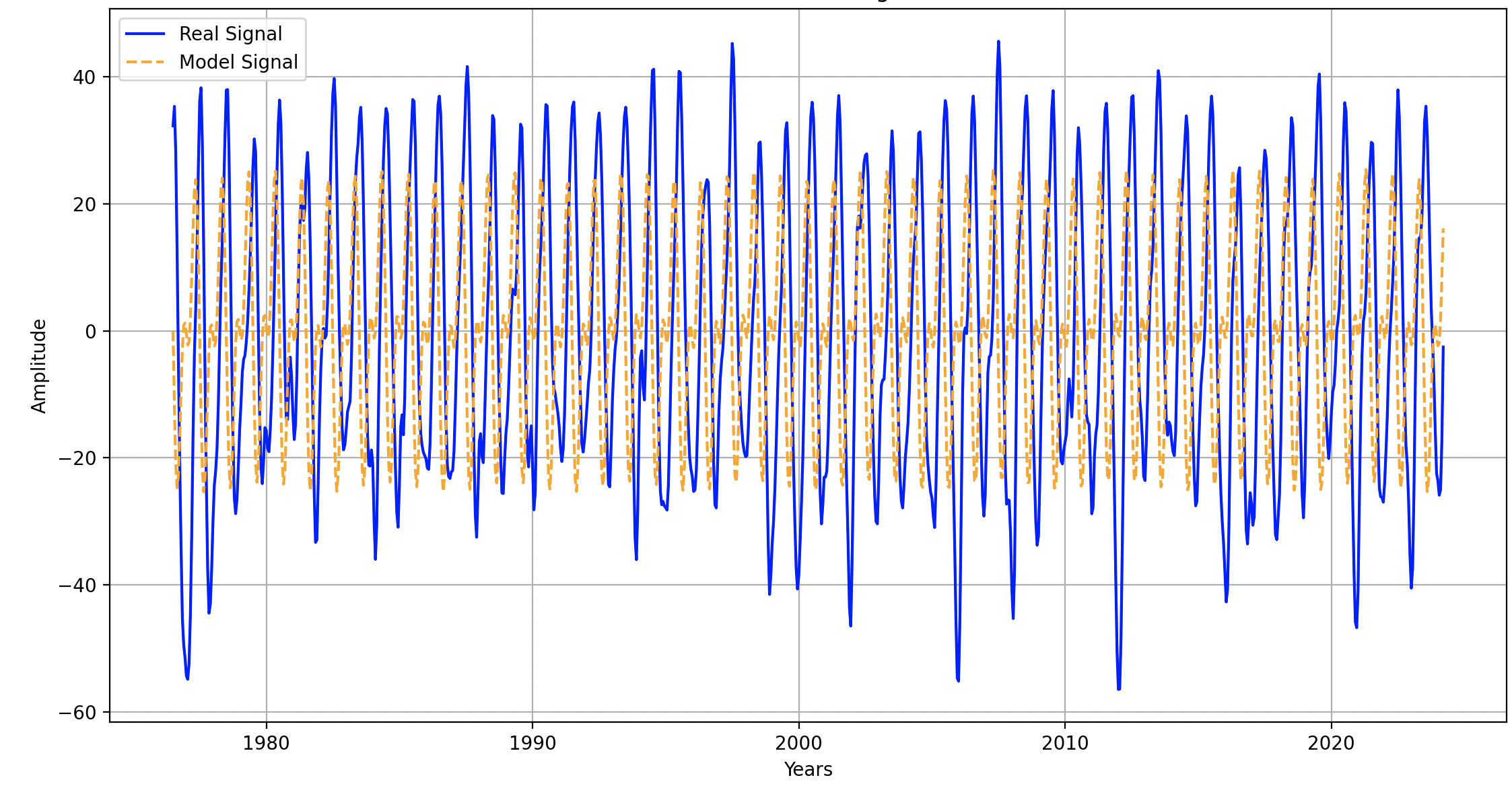


Рис. 6 Смоделированный сигнал против реального

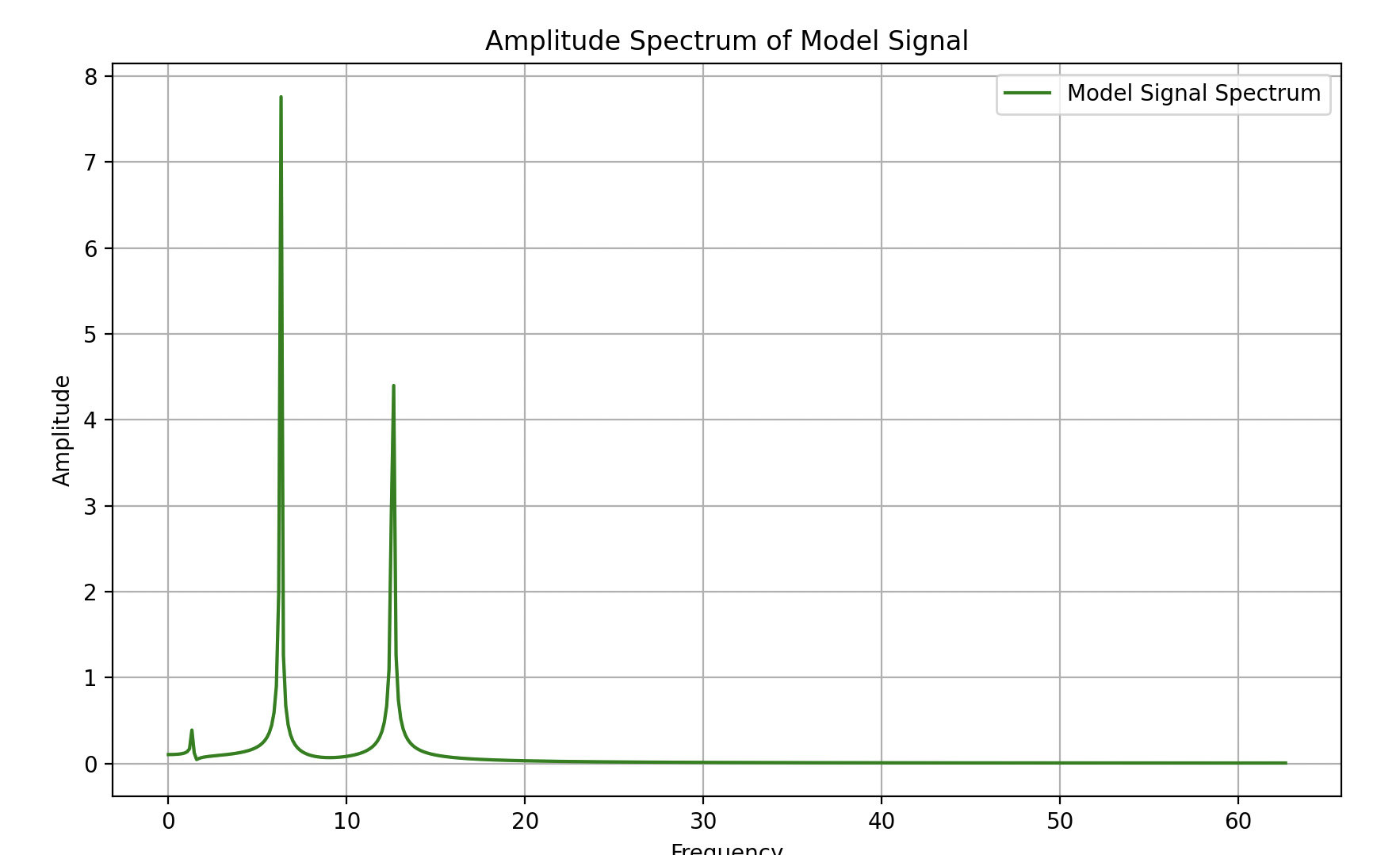


Рис. 7 Спектрограмма смоделированного сигнала

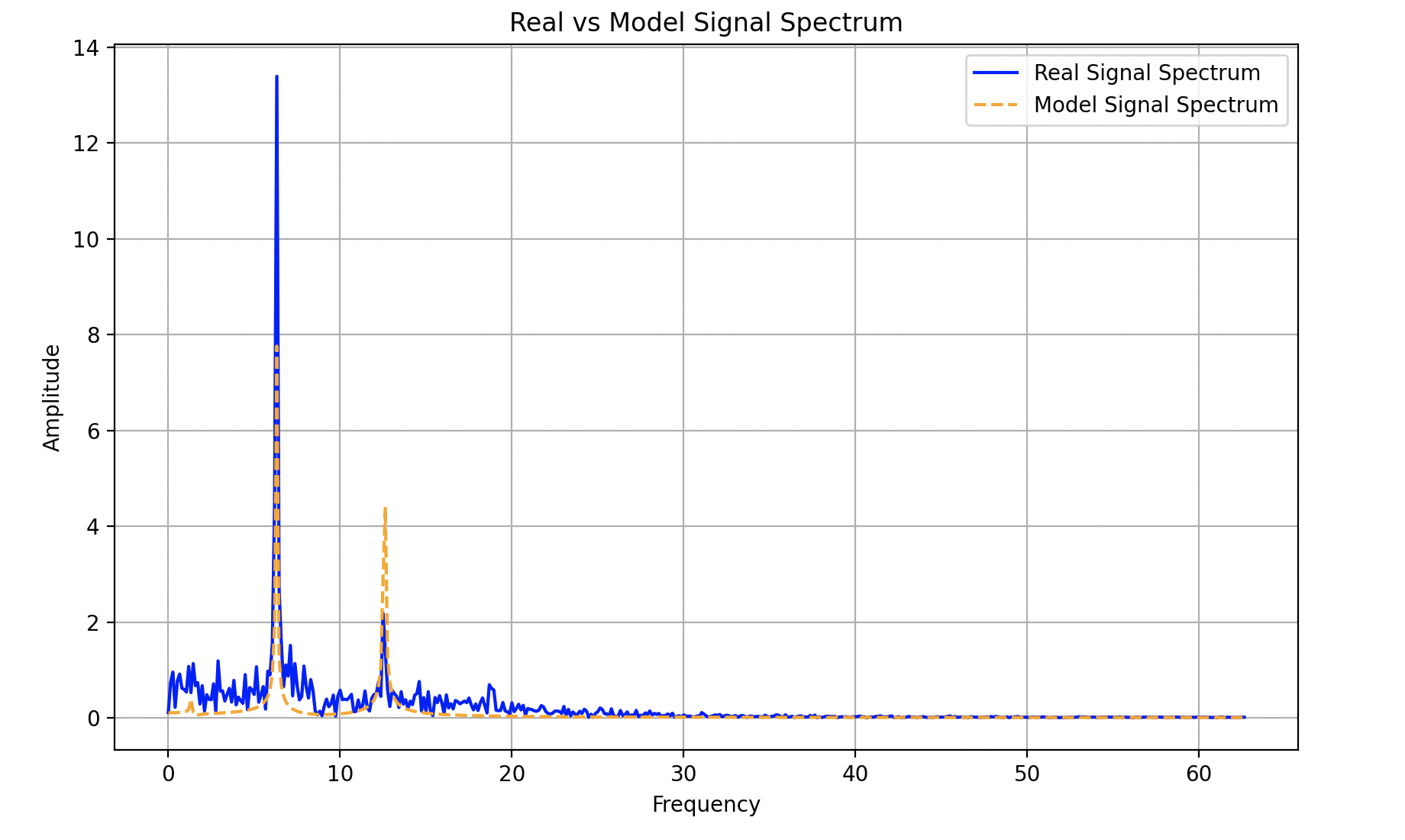


Рис. 8 Спектрограмма смоделированного сигнала, наложенного на реальный