Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Телекоммуникационные технологии

Отчет по лабораторной работе №3
Линейная фильтрация

Работу выполнил:
Балсутьев В.А.
Группа: 33501/4
Преподаватель:
Богач Н.В.

Санкт-Петербург
2017

Содержание

1. Цель работы

Изучить воздействие ФНЧ на тестовый сигнал с шумом.

2. Постановка задачи

Сгенерировать гармонический сигнал с шумом и синтезировать ФНЧ. Получить сигнал во временной и частотной областях до и после фильтрации. Сделать выводы о воздействии ФНЧ на спектр сигнала.

3. Теоретическая информация

Линейный фильтр (wiki) — динамическая система, применяющая некий линейный оператор ко входному сигналу для выделения или подавления определённых частот сигнала и других функций по обработке входного сигнала. Также довольно часто линейные фильтры называют линейные цепями(здесь и далее это синонимы).

Преобразование непрерывных сигналов в линейных цепях с постоянными параметрами может быть описано с помощью линейных дифференциальных уравнений с постоянными коэффициентами. Результатом интегрирования и дифференцирования гармонической функции некоторой частоты являются также гармонические функции той же частоты. Поэтому при подаче на вход линейной цепи гармонического сигнала.

$$x(t) = A_x e^{j(2\pi f t + \psi_x)}$$

на выходе цепи будет получен гармонический сигнал, отличающийся о входного лишь амплитудой и фазой:

$$y(t) = A_y e^{j(2\pi f t + \psi_y)}$$

Отношение выходного сигнала цепи к входному гармоническому сигналу произвольной частоты носит название частотной характеристики (ЧХ)$G(f)$:

$$G(f) = \frac{y(t)}{x(t)} = \frac{A_y}{A_x} e^{i(\psi_y - \psi_x)} = \mid G(f) \mid e^{i\psi(f)},$$

где модуль частотной характеристики $\mid G(f) \mid$ носит название амплитудно-частотной характеристики (АЧХ), а аргумент экспоненты $\psi(f)$ - фазо-частотная характеристика(ФЧХ). Если на вход цепи подается некоторое произвольное воздействие x(t), оно может быть разложено на гармонические составляющие с помощью преобразования Фурье:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{-j\pi f t} df$$

Некоторая гармоника $_f(t)$ частоты f, входящая в этот сигнал, имеет вид

$$x_f(t) = X(f) df e^{-j\pi f t}$$

Пройдя через линейную цепь, имеющую ЧХ $G(f)$, гармоника преобразуется в гармонику выходного сигнала:

$$y_f(t) = x_f(t) G(f) = X(f) G(f) df e^{-j\pi f t},$$

из чего следует, что спектр выходного сигнала $Y(f)$ равен произведению спектра входного сигнала цепи и ее частотной характеристики:

$$Y(f) = X(f) G(f)$$

Во временной области выходной сигнал цепи $y(t)$ может быть найден исходя из последней формулы с помощью обратного преобразования Фурье:

$$y(t) = \int_{-\infty}^{\infty} Y(f)e^{-j\pi ft}df$$

Таким образом, зная ЧХ линейной цепи, можно найти описание выходного сигнала цепи вначале частотной области, а затем и во времнной.

В данной работе будем использовать фильтр Баттерворта (wiki), для которого свойственно несколько осбенностей:

- АЧХ фильтра Баттерворта максимально гладкая на частотах полосы пропускания и снижается практически до нуля на частотах полосы подавления
- Фильтр Баттерворта — единственный из фильтров, сохраняющий форму АЧХ для более высоких порядков (за исключением более крутого спада характеристики на полосе подавления) тогда как многие другие разновидности фильтров (фильтр Бесселя, фильтр Чебышёва, эллиптический фильтр) имеют различные формы АЧХ при различных порядках.
- Амплитудно-частотная характеристика $G(\omega)$ фильтра Баттерворта n - го порядка может быть получена из передаточной функции $H(s)$:

  $$G^2(\omega) = (\mid H(j\omega) \mid)^2 = \frac{G_0^2}{1 + (\frac{\omega}{\omega_c})^2 n}$$

  где n - порядок фильтра, w_c - частота среза(частота на которой амплитуда равна - 3 dB), G_0 - коэффициент усиления по постоянной составляющей(усиление на нулевой частоте)

4. Ход выполнения работы

Положим сигнал гармоническим и зададим его формулой

$$s(t) = \sin(2, 4\pi t) \tag{1}$$

, далее прибавим к нему шумы, и получим зашумленный сигнал:

$$s'(t) = \sin(2, 4\pi t) + 1, 5\cos(9\pi t) + \frac{\sin(12\pi t)}{2} \tag{2}$$

Чистый Сигнал и его спектр в соответствии с формулой(1) принимают вид:

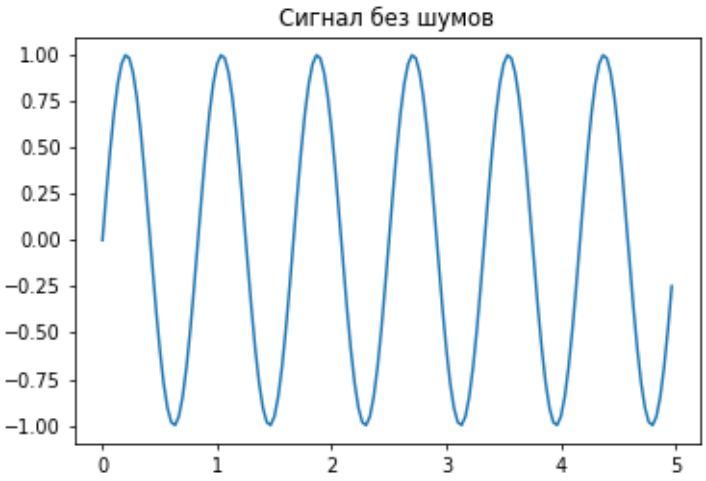

Рисунок 4.1. Чистый сигнал

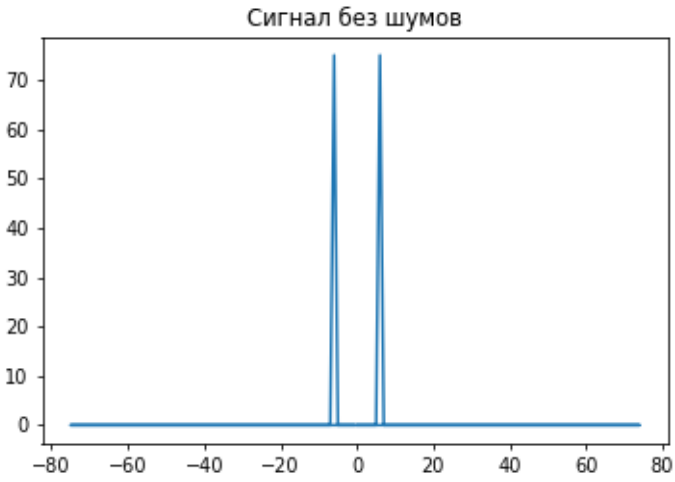

Рисунок 4.2. Спектр чистого сигнал

Реализуем вспомогательные функции для использования фильтра баттерворта и далее положим порядок фильтра $n = 6$ и частотой среза = 3,6. Выполним фильтрацию и построим отфильтрованный сигнал и зашумленный сигнал:

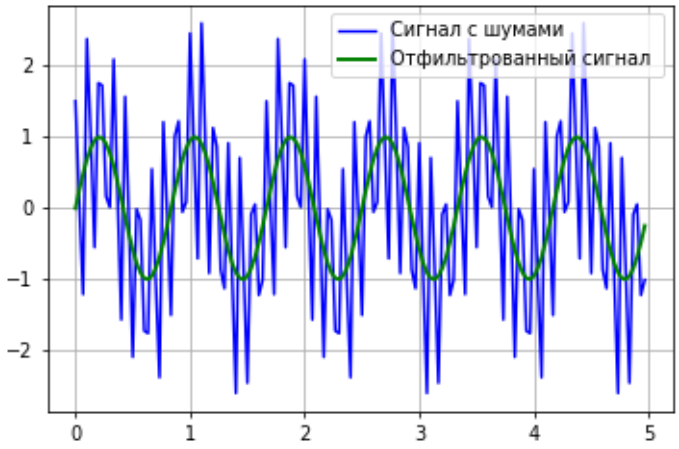

Рисунок 4.3. Сигналы до и после фильтрации

Рассмотрим спектр исходного зашумленного сигнала и отфильтрованного сигнала.

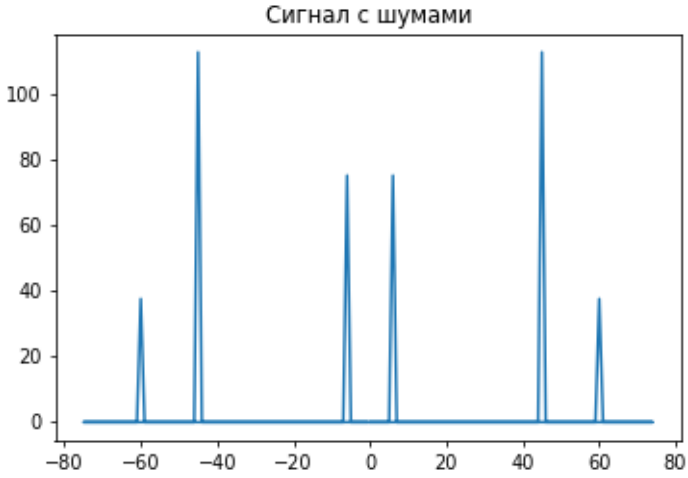

Рисунок 4.4. Спектр зашумленного сигнала

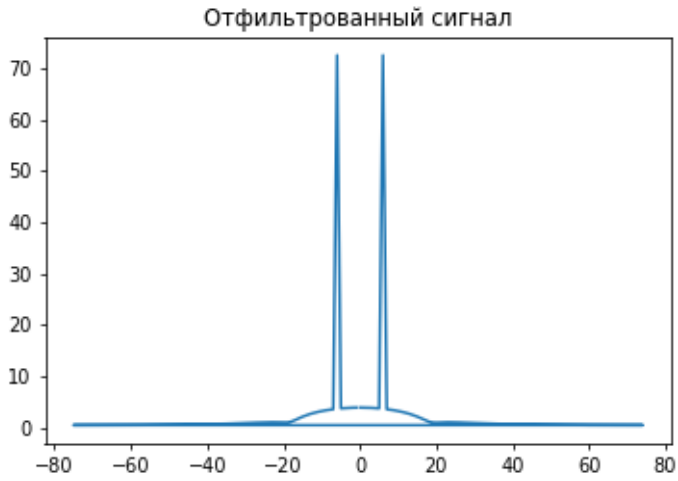

Рисунок 4.5. Спектр сигнала после фильтрации

Листинг 1: source01.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import butter, lfilter, freqz
from scipy.fftpack import fft, fftfreq

%matplotlib inline

def butter_lowpass(cutoff, fs, order=5):
    nyq = 0.5 * fs
    normal_cutoff = cutoff / nyq
    b, a = butter(order, normal_cutoff, btype='low', analog=False)
    return b, a

def butter_lowpass_filter(data, cutoff, fs, order=5):
    b, a = butter_lowpass(cutoff, fs, order=order)
    y = lfilter(b, a, data)
    return y


# Filter requirements.
order = 6
fs = 30.0       # sample rate, Hz
cutoff = 3.667  # desired cutoff frequency of the filter, Hz

# Demonstrate the use of the filter.
# First make some data to be filtered.
T = 5.0         # seconds
n = int(T * fs) # total number of samples
t = np.linspace(0, T, n, endpoint=False)


clear_data  = np.sin(1.2*2*np.pi*t)
mf_noize = 1.5*np.cos(9*2*np.pi*t) + 0.5*np.sin(12.0*2*np.pi*t)

fig = plt.figure()
plt.title('игналС без шумов')
plt.plot(t, clear_data)
fig.savefig('pictures/001_true_sig.png', dpi=70)
```

```
39 plt.show()
40
41
42 fig = plt.figure()
43 plt.title('игналС без шумов')
44 clear_dataf = fft(clear_data)
45 xdataf = fftfreq(n , 1.0 / n)
46 plt.plot(xdataf, np.abs(clear_dataf))
47 fig.savefig('pictures/002_true_sig_spec.png', dpi=70)
48 plt.show()
49
50 # "Noisy" data.  We want to recover the 1.2 Hz signal from this.
51 data = clear_data + mf_noize
52
53 fig = plt.figure()
54 plt.plot(t, data, 'b-', label='Сигнал с шумами')
55 plt.plot(t, clear_data, 'g-', linewidth=2, label='Отфильтрованный сигнал ')
56 plt.grid()
57 plt.legend()
58 fig.savefig('pictures/003_signals.png', dpi=70)
59 plt.show()
60
61 # Filter the data, and plot both the original and filtered signals.
62 y = butter_lowpass_filter(data, cutoff, fs, order)
63
64
65 fig = plt.figure()
66 plt.title('игналС с шумами')
67 dataf = fft(data)
68 xdataf = fftfreq(n , 1.0 / n)
69 plt.plot(xdataf, np.abs(dataf))
70 fig.savefig('pictures/004_spctr_noisy_signa.png', dpi=70)
71 plt.show()
72
73 fig = plt.figure()
74 plt.title('Отфильтрованный сигнал')
75 yf = fft(y)
76 xf = fftfreq(n , 1.0 / n)
77 plt.plot(xf, np.abs(yf))
78 fig.savefig('pictures/005_spctr_signa.png', dpi=70)
79 plt.show()
```

5. Выводы

Действительно, фильтр отметает все шумы с частотами, большими чем 1,2 и оставляет только требуемый нам сигнал(на которой наш фильтр и настроен). Так же после рассмотрения спектра чистого сигнала(4.2) и спектра сигнала после фильтрации (4.5) следует отметить очевидную особенность - фильтр формирует не идеальный сигнал, даже несмотря на максимально гладкую АЧХ фильтра Баттерворта и 6 порядок фильтра. Но все же отличия исходного спектра следует признать пренебрежимо малыми.

В результате данной работы мы рассмотрели процесс линейной фильтрации с помощью фильтра Баттерворта. Так же нам удалось на качественном уровне разобраться с влиянием линейных фильтров на зашумленный сигнал и его спектр.