

САНКТ-ПЕТЕРБУРГСКИЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО

---

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ

---

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ПРОГРАММНЫХ ТЕХНОЛОГИЙ

Отчет  
по лабораторной работе №3  
на тему: "Линейная фильтрация"

выполнила:  
Шевченко А.С.  
группа: 33501/1  
преподаватель:  
Богач Н.В.

Санкт-Петербург  
2018

## 1. Цель работы

Изучить воздействие ФНЧ на тестовый зашумленный сигнал.

## 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 часть

### 3.1 Преобразование непрерывных сигналов в линейных цепях (по В.С. Гутникову)

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 от 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)  $G(f)$ :

$$G(f) = \frac{y(t)}{x(t)} \Big|_{x(t)=A_x e^{j(2\pi f t + \psi_x)}}$$

Объединяя последние два уравнения получим:

$$G(f) = \frac{A_y}{A_x} e^{j(\psi_y - \psi_x)} = |G(f)| e^{j\psi(f)},$$

где  $\psi(f) = \psi_y - \psi_x$ . Модуль частотной характеристики  $|G(f)|$  носит название амплитудно-частотной характеристики (АЧХ), а ее аргумент  $\psi(f)$  — фазо-частотной характеристики (ФЧХ).

### 3.2 Линейная фильтрация

Фильтрация представляет собой одну из самых распространенных операций обработки сигналов. Цель фильтрации состоит в подавлении помех, содержащихся в сигнале, или в выделении отдельных составляющих сигнала, соответствующих тем или иным свойствам исследуемого процесса. Фильтрация сигнала заключается в целенаправленном изменении соотношения между различными компонентами спектра сигнала.

Фильтры - это устройства, изменяющие спектры сигналов. Фильтры классифицируются по виду входного и выходного сигнала (цифровые или аналоговые), по виду частотной характеристики (низких частот, верхних частот, полосовые), по виду импульсных характеристик (непрерывный, дискретный) и по протяженности импульсной характеристики (с конечной импульсной характеристикой, с бесконечной импульсной характеристикой).

Фильтр низких частот (ФНЧ), рассматриваемый в данной работе, пропускает низкочастотный составляющие спектра и задерживает высокочастотные.

## 4. Ход работы

Итак, необходимо сгенерировать гармонический сигнал, наложить шум и попытаться отфильтровать его через ФНЧ. Для генерации фильтра было использовано приложение Filter Designer в Матлабе. Код с описанием сигнала, его спектра и код полученного фильтра приведены ниже (Рис. 1):

```

1 - close all
2 - N1 = 512; %otscheti
3 - F0 = 1; %chastota
4 - Fs = 100; %chastota discretizatsii
5 - A = 1; % amplituda
6 - T = 4; %dlitelnost signala
7 - t = 0:1/Fs:T;
8 - df = Fs/N1;
9
10 - y = A*sin(2*pi*F0*t);
11 - figure;
12 - plot(t, y)
13 - xlabel('t');
14 - ylabel('y(t)');
15 - title('Sine Signal');
16 - ylim([-A-1 A+1]),grid
17
18 - ns_y = awgn(y, 20);%ns = noise
19 - figure;
20 - plot(t, ns_y)
21 - xlabel('t');
22 - ylabel('y(t)');
23 - title('Noised Signal');
24 - ylim([-A-1 A+1]),grid
25
26 - f = 0:df:Fs/2 - df; %massiv chastot spectra
27 - Y_ns_fft = abs(fft(ns_y, N1)); %amplitudi preobrazovania furie
28 - figure;
29 - plot(f, Y_ns_fft(1:length(f))), grid
30 - xlabel('Frequency (Hz)');
31 - ylabel('|Y(f)|');
32 - title('Amplitude Spectrum of Noised Signal');
33
34 - %FILTER_DESIGN Returns a discrete-time filter object.
35
36 - % MATLAB Code
37 - % Generated by MATLAB(R) 9.2 and the DSP System Toolbox 9.4.
38 - % Generated on: 14-Apr-2018 23:20:29
39
40 - % FIR Window Lowpass filter designed using the FIR1 function.
41
42 - % All frequency values are in Hz.
43
44 - Fpass = 2; % Passband Frequency
45 - Fstop = 4; % Stopband Frequency
46 - Dpass = 0.057501127785; % Passband Ripple
47 - Dstop = 0.0001; % Stopband Attenuation
48 - flag = 'scale'; % Sampling Flag
49
50 - % Calculate the order from the parameters using KAISERORD.
51 - [N,Wn,BETA,TYPE] = kaiserord([Fpass Fstop]/(Fs/2), [1 0], [Dstop Dpass]);
52
53 - % Calculate the coefficients using the FIR1 function.
54 - b = fir1(N, Wn, TYPE, kaiser(N+1, BETA), flag);
55 - Hd = dfilt.dfir(b);
56
57 - % [EOF]
58
59 - y_filt = filter(Hd, ns_y);
60 - figure;
61 - plot(t, ns_y, t, y_filt);
62 - xlabel('t');
63 - ylabel('y(t)');
64 - title('Noised and Filtered Signals');
65 - ylim([-A-1 A+1]),grid
66
67 - f = 0:df:Fs/2 - df; %massiv chastot spectra
68 - Y_filt_fft = abs(fft(y_filt, N1)); %amplitudi preobrazovania furie
69 - figure;
70 - plot(f, Y_filt_fft(1:length(f))), grid
71 - xlabel('Frequency (Hz)');
72 - ylabel('|Y(f)|');
73 - title('Amplitude Spectrum of Filtered Signal');
74

```

Рис. 1: Код скрипта для генерации зашумленного сигнала и его фильтрации

Далее мы будем менять степень зашумленности сигнала и смотреть, как справляется наш фильтр. Шум обеспечивается функцией `awgn(signal, noise)`, вместо `noise` подставим значение 20 - шум меньше, чем исходный сигнал:

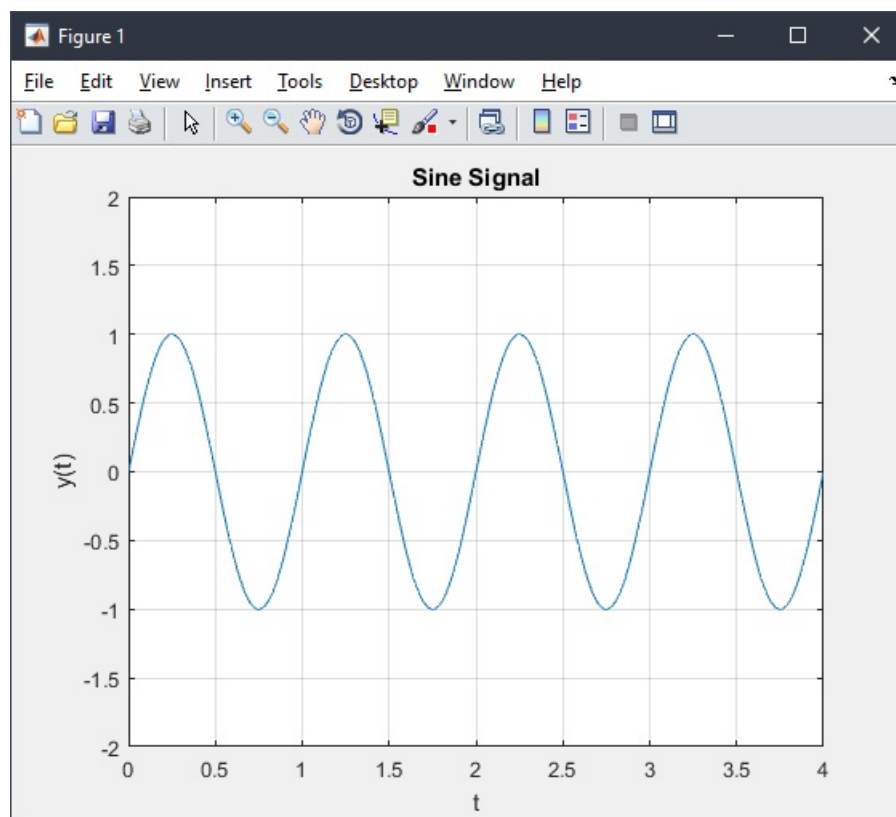


Рис. 2: Исходный сигнал

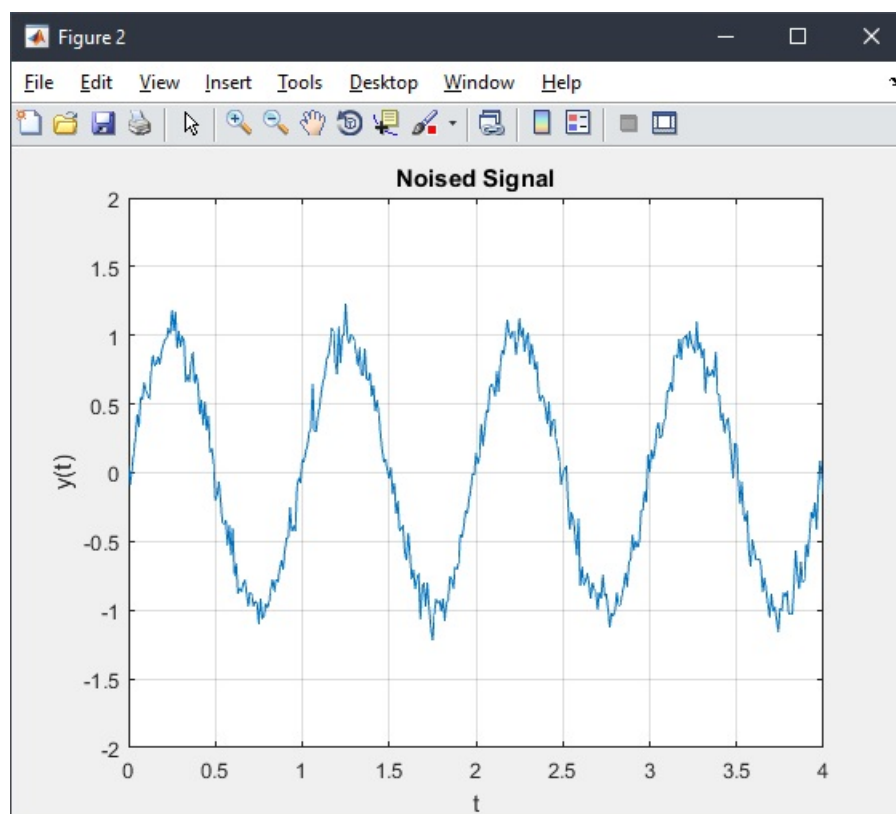


Рис. 3: Зашумленный сигнал

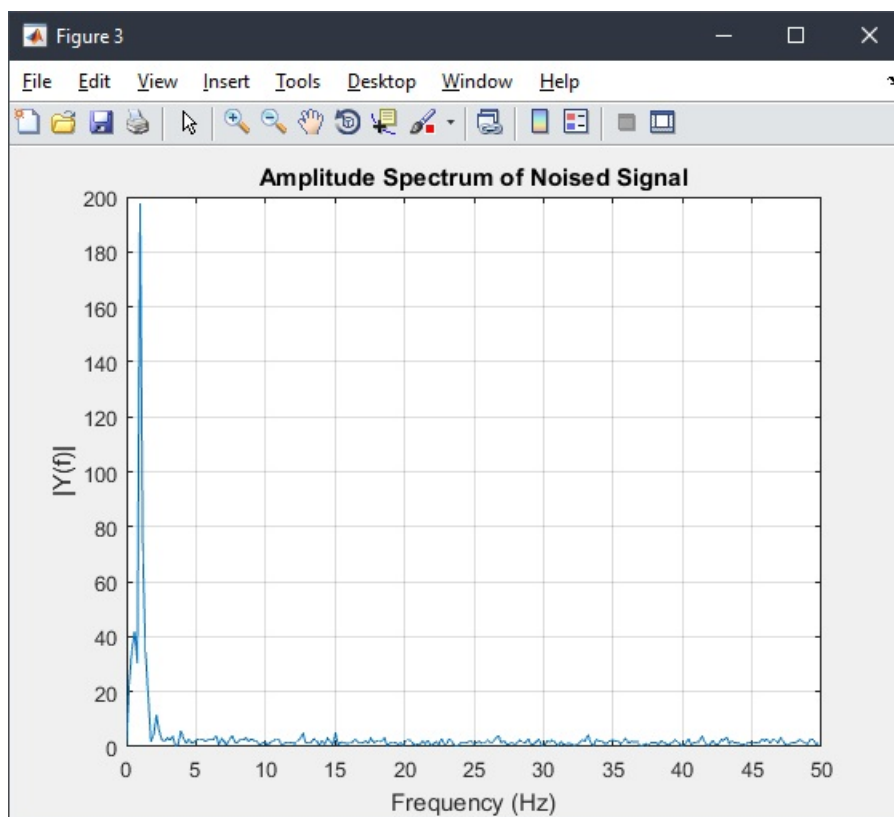


Рис. 4: Спектр зашумленного сигнала

Результаты фильтрации приведены на рисунках ниже:

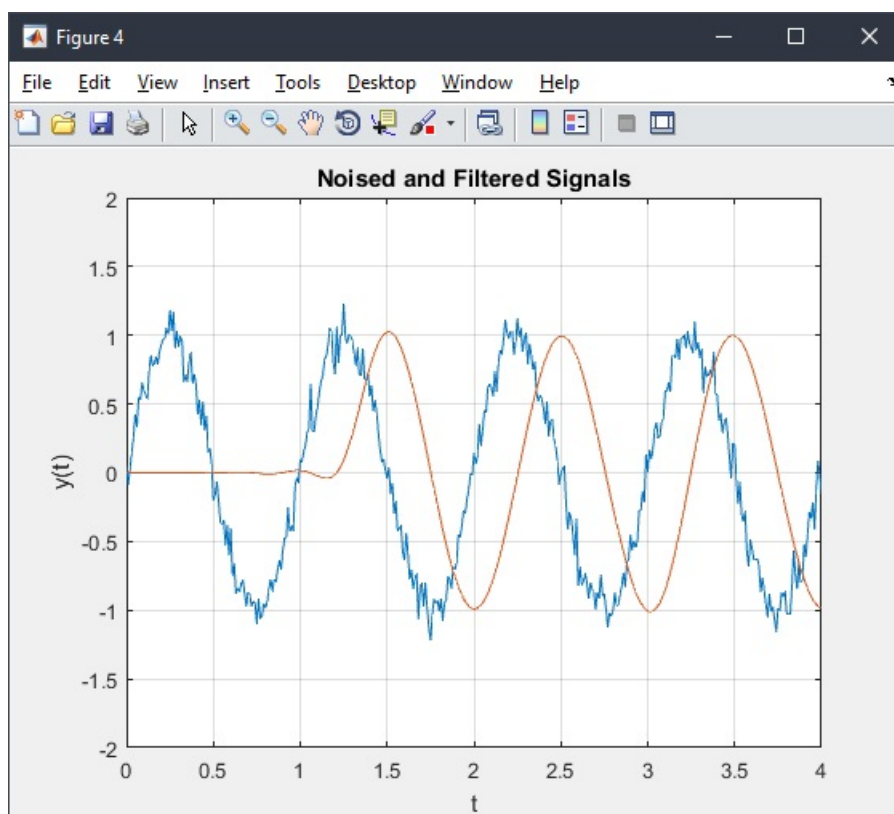


Рис. 5: Зашумленный сигнал (синий) и отфильтрованный сигнал (красный)

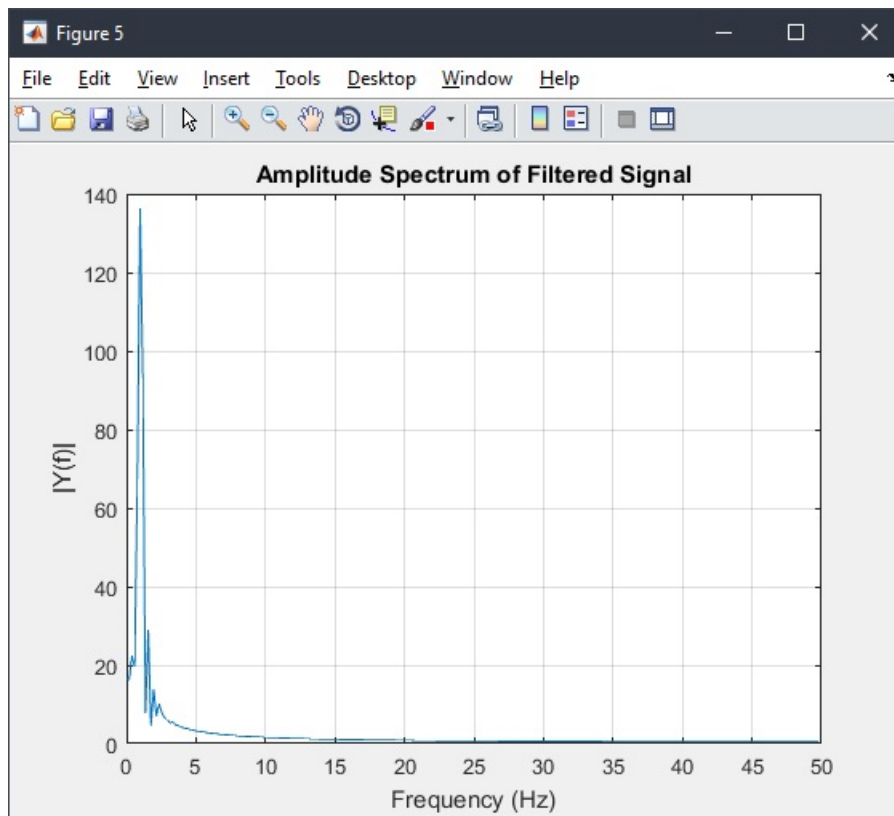


Рис. 6: Спектр отфильтрованного сигнала

Таким образом, при данной степени зашумленности сигнала, созданный нами фильтр справился хорошо. Теперь увеличим количество шума - подставим параметр -1. Знак минус значит, что шума больше, чем полезного сигнала.

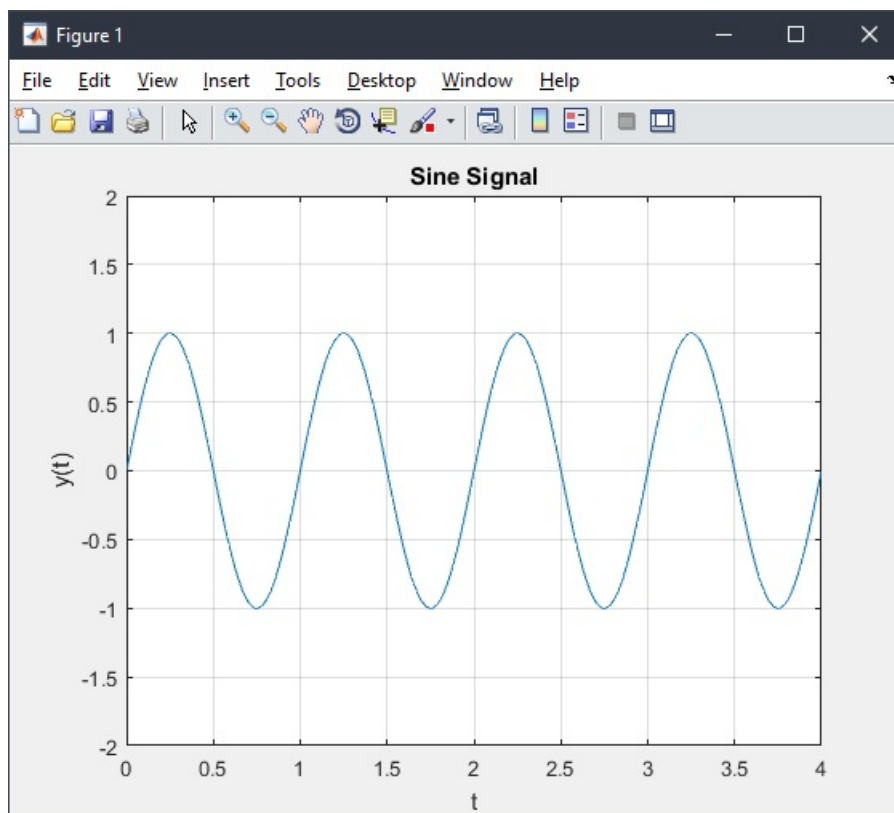


Рис. 7: Исходный сигнал

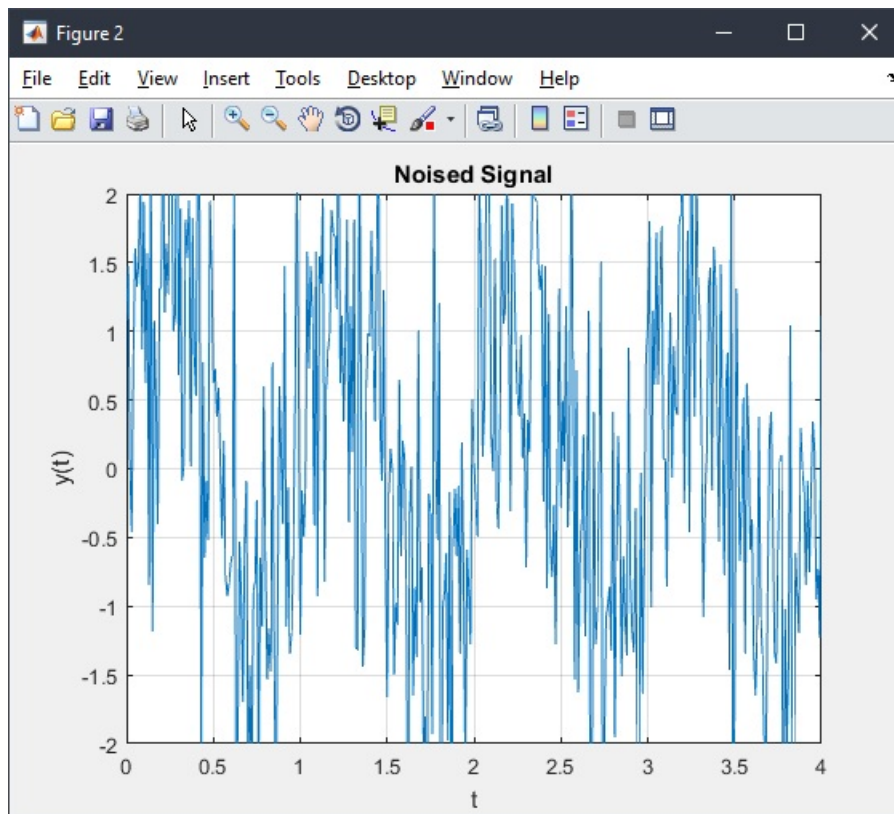


Рис. 8: Зашумленный сигнал

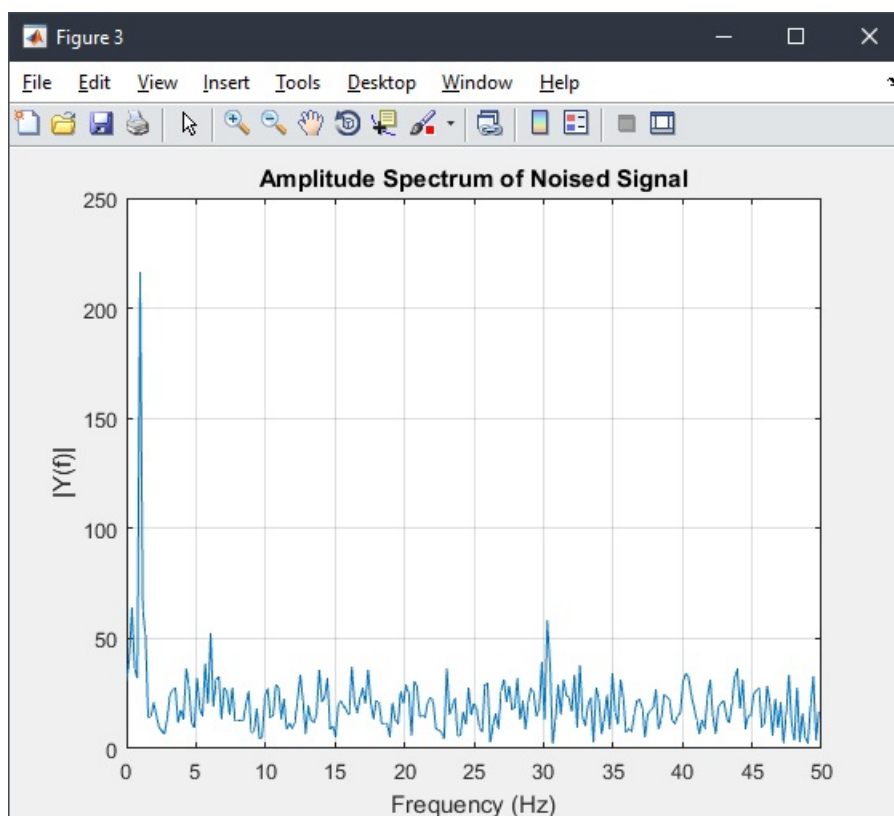


Рис. 9: Спектр зашумленного сигнала

Результаты фильтрации:

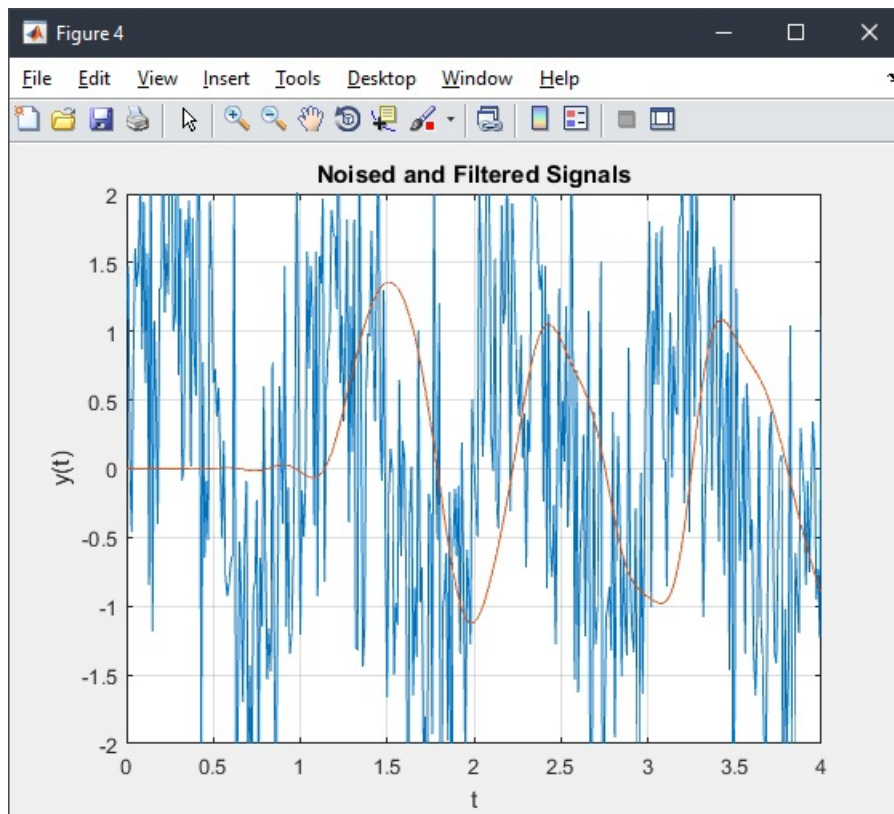


Рис. 10: Зашумленный сигнал (синий) и отфильтрованный сигнал (красный)

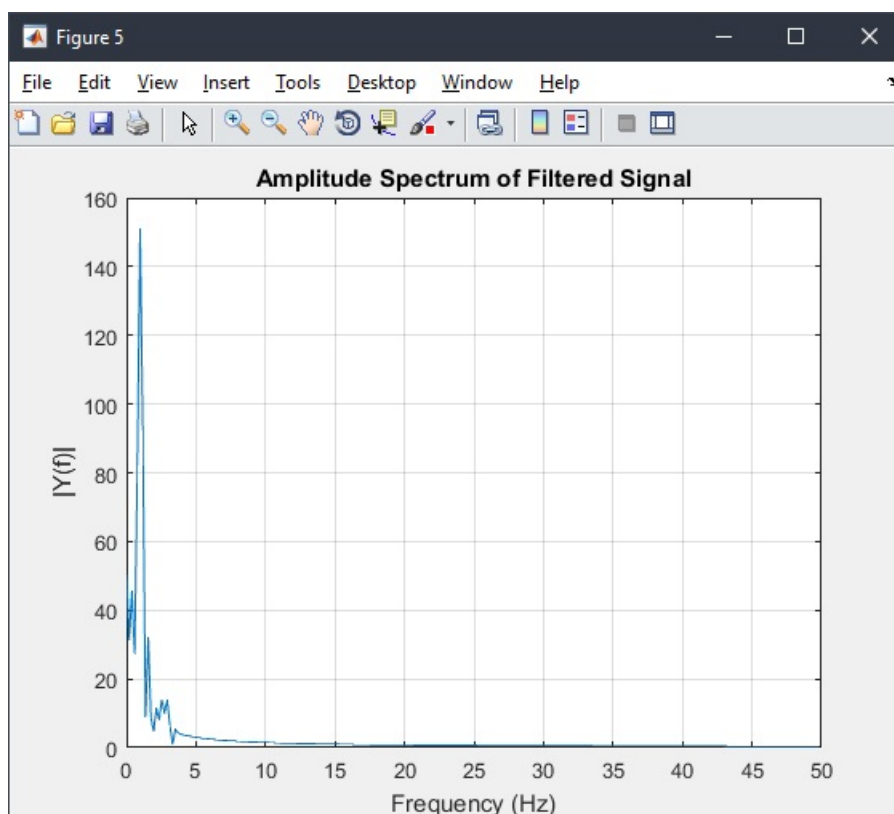


Рис. 11: Спектр отфильтрованного сигнала

После фильтрации такого сигнала на выходе мы видим сигнал, слегка напоминающий исходную синусоиду.

Теперь параметр зашумленности установим -10. Исходный сигнал после наложения такого шума совсем потеряет самого себя.



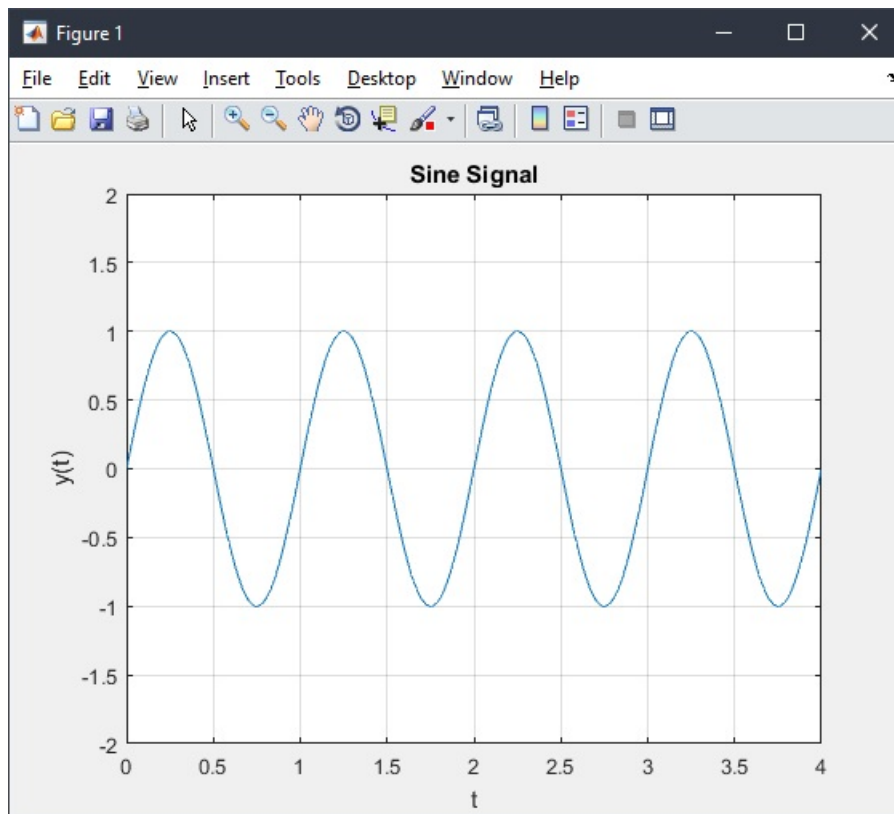


Рис. 12: Исходный сигнал

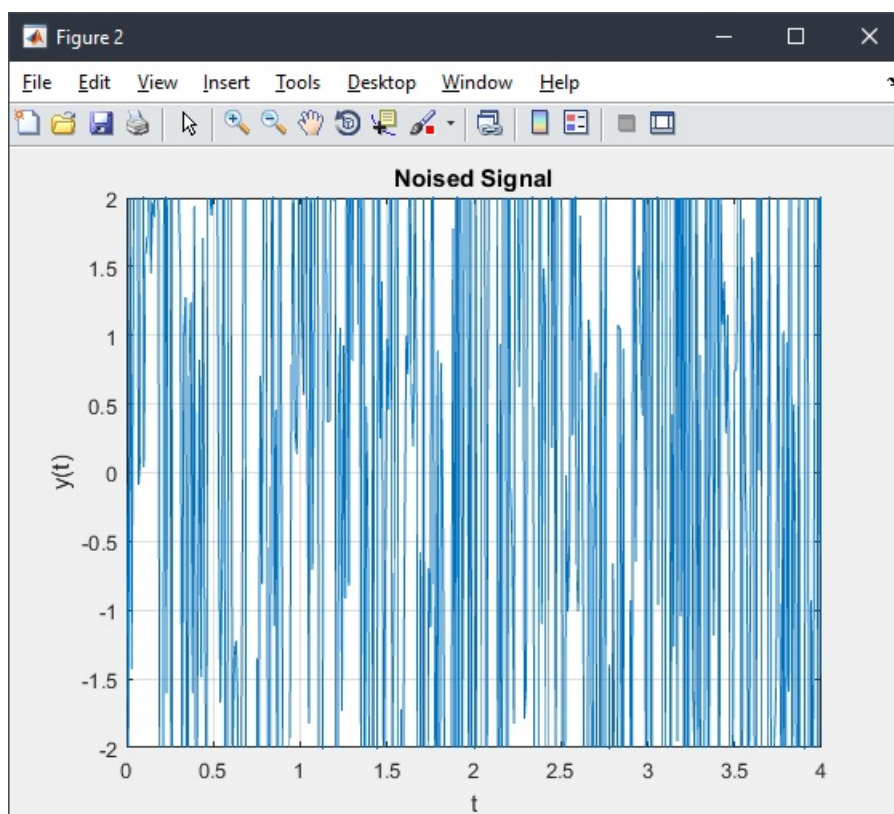


Рис. 13: Зашумленный сигнал

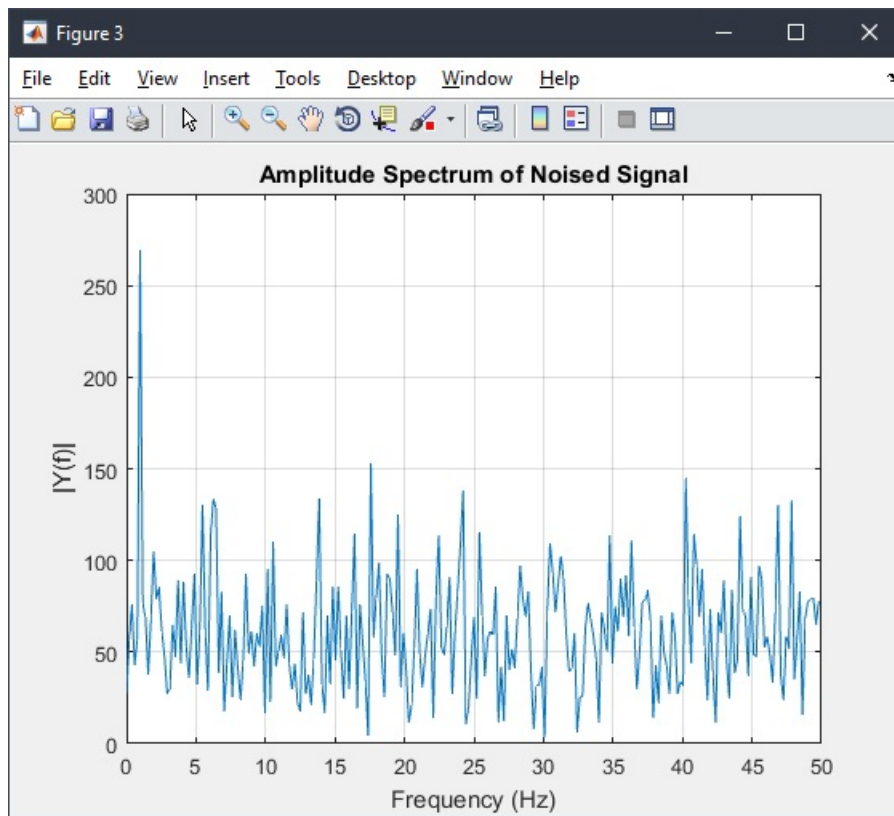


Рис. 14: Спектр зашумленного сигнала

Результаты фильтрации приведены на рисунках ниже:

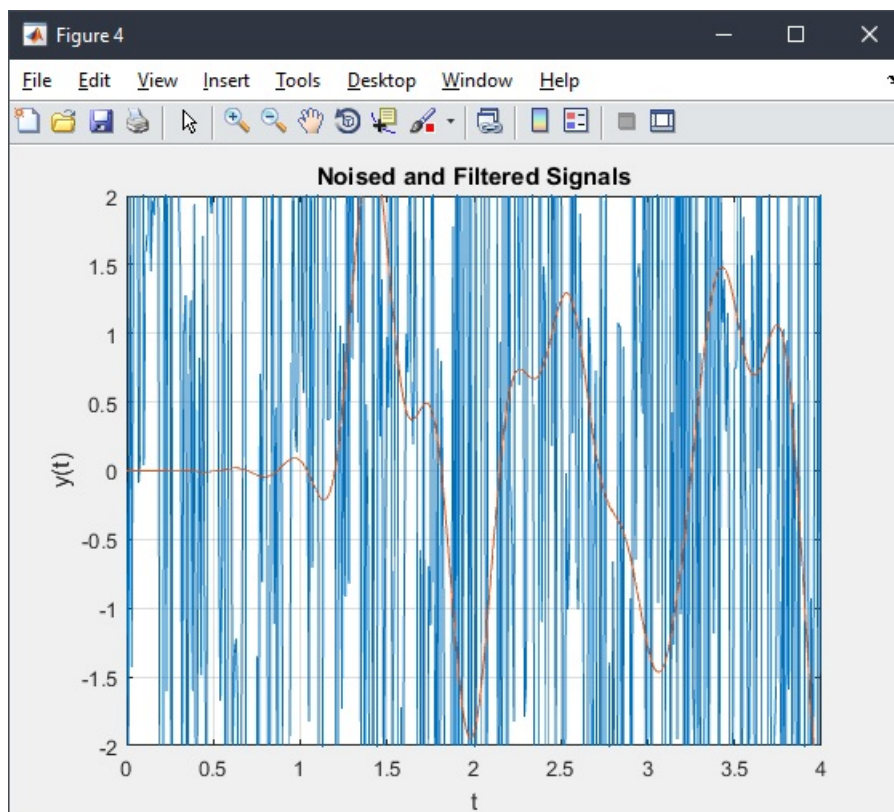


Рис. 15: Зашумленный сигнал (синий) и отфильтрованный сигнал (красный)

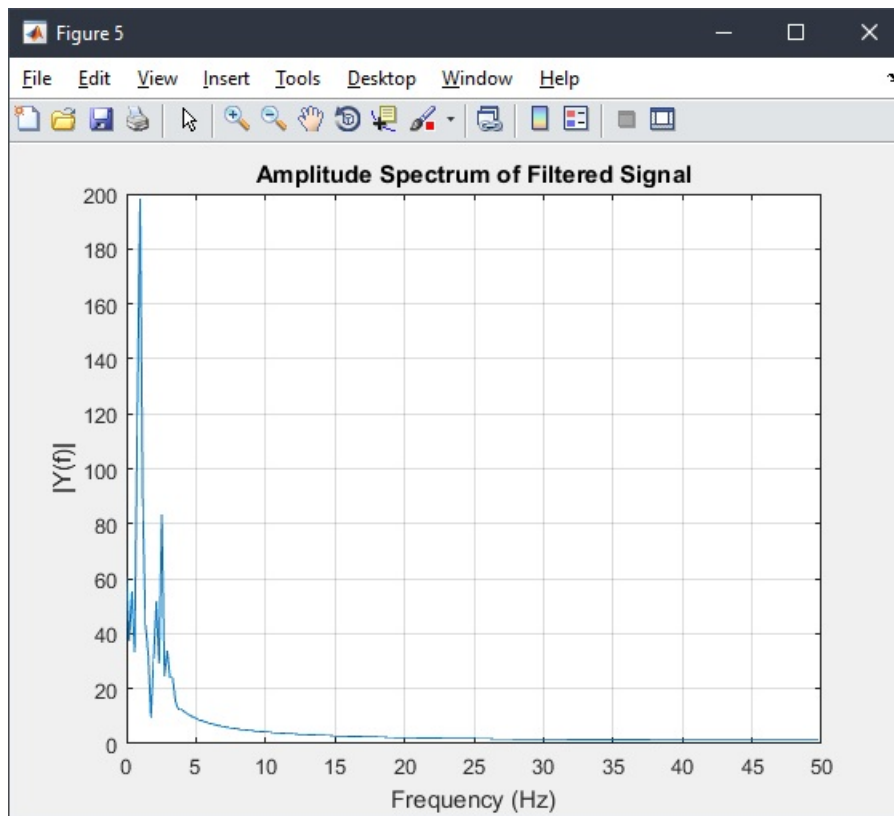


Рис. 16: Спектр отфильтрованного сигнала

Как видно, фильтр пытался справиться, что-то отфильтровал, что то не смог, но, к сожалению, с полученным после фильтрации сигналом уже работать нельзя.

## 5. Выводы

Таким образом, мы сгенерировали гармонический сигнал различной степени зашумленности и пропустили его через ФНЧ. Полностью сигнал отфильтровать не всегда представляется возможным, потому как мы использовали фильтр низких частот, а это значит, что задерживаются высокочастотные помехи, а низкочастотные пропускаются без проблем.

Процесс прохождения сигнала через линейную цепь можно описать так: исходный сигнал во временной области сворачивается со спектром частотной характеристики цепи  $G(f)$  - получается выходной сигнала.

Если на вход подается некоторый случайный сигнал, можно сначала найти его спектр, далее в линейной цепи он умножится на ее частотную функцию - так мы получим спектр выходного сигнала. Затем применив обратное преобразование Фурье, можно получить сам сигнал.