

Санкт-Петербургский политехнический университет Петра Великого
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе

Дисциплина: Телекоммуникационные технологии

Тема: Линейная фильтрация

Выполнил студент гр. 33501/2
Преподаватель

Миносян Э.К.
Богач Н.В.

Санкт-Петербург
25 апреля 2018 г.

0 Содержание

1	Цель работы	2
2	Постановка задачи	2
3	Теоретический раздел	2
3.1	Общие сведения о линейной фильтрации	2
4	Ход работы	3
4.1	Генерация гармонического сигнала с шумом	3
4.2	Фильтрация сигнала с шумом	7
5	Выводы	11

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 Теоретический раздел

3.1 Общие сведения о линейной фильтрации

Линейный фильтр — динамическая система, применяющая некий линейный оператор ко входному сигналу для выделения или подавления определённых частот сигнала и других функций по обработке входного сигнала. Линейные фильтры широко применяются в электронике, цифровой обработке сигналов и изображений, в оптике, теории управления и других областях.

Наиболее часто они используются для того, чтобы подавить нежелательные частоты входного сигнала или для того чтобы выделить нужную полосу частот в сигнале. Существует большое количество различных типов и модификаций линейных фильтров, в статье описаны наиболее распространённые.

Несмотря на природу фильтра — механическую, оптическую, электронную, программную или электрическую, а также на частотный диапазон, в котором они работают, математическая теория линейных фильтров универсальна и может быть применена к любому из них.

4 Ход работы

4.1 Генерация гармонического сигнала с шумом

Был написан код на языке MATLAB, который генерирует синусоидальный сигнал накладывает на него шум и выводит спектр

```
1 - A1 = 5; %Амплитуда 1
2 - f = 10;
3 - fs = 8400; % Частота Дискретизации
4 - t=0:1/8400:1; %Время
5 - y1 = A1*sin(2*pi*f*t);
6 - y = awgn(y1,10);
7 - figure;
8 - plot(t,y1);
9 - axis([0 1 -5 5])
10 - figure;
11 - plot(t,y);
12 - axis([0 1 -5 5])
13 - N=length(t);
14 - ffts1=abs(fft(y1,N)); % преобразование Фурье по модулю
15 - ffts1=2*ffts1./N;% нормализация
16 - ffts1(1)=ffts1(1)/2;% нормировка постоянной составляющей в спектре
17 - f1=0:fs/N:fs/2-fs/N;% вектор частот
18 - figure;
19 - plot(f1,ffts1(1:length(f1)),'m');%вывод спектра сигнала до наложения шума
20 - axis([0 100 0 5])
21 - title('Спектр сигнала до наложения шума');
22 - ffts2=abs(fft(y,N)); % преобразование Фурье по модулю
23 - ffts2=2*ffts2./N;% нормализация
24 - ffts2(1)=ffts2(1)/2;% нормировка постоянной составляющей в спектре
25 - f1=0:fs/N:fs/2-fs/N;% вектор частот
26 - figure;
27 - plot(f1,ffts2(1:length(f1)),'m');%вывод спектра зашумленного сигнала
28 - axis([0 100 0 5])
29 - title('Спектр зашумленного сигнала');
30 - t5=flt;
31 - y2=filter(t5,y);
32 - figure;
33 - plot(t,y2);
34 - axis([0 1 -5 5])|
35 - ffts3=abs(fft(y,N)); % преобразование Фурье по модулю
36 - ffts3=2*ffts3./N;% нормализация
37 - ffts3(1)=ffts3(1)/2;% нормировка постоянной составляющей в спектре
38 - f2=0:fs/N:fs/2-fs/N;% вектор частот
39 - figure;
40 - plot(f2,ffts3(1:length(f2)),'m');%вывод спектра отфильтрованного сигнала
41 - axis([0 100 0 5])
42 - title('Спектр отфильтрованного сигнала');
```

Рис. 4.1: Код на языке MATLAB

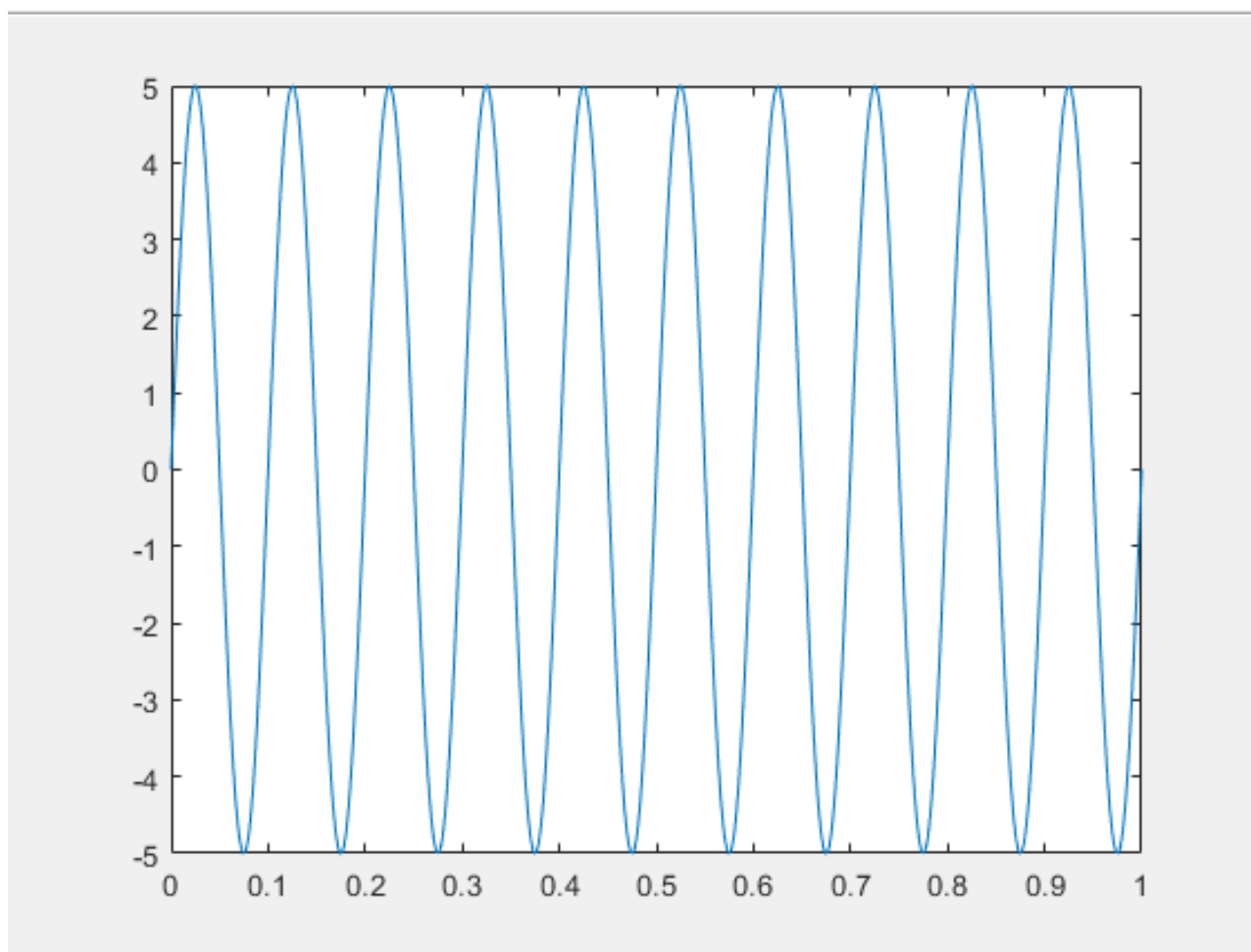


Рис. 4.2: Исходный сигнал

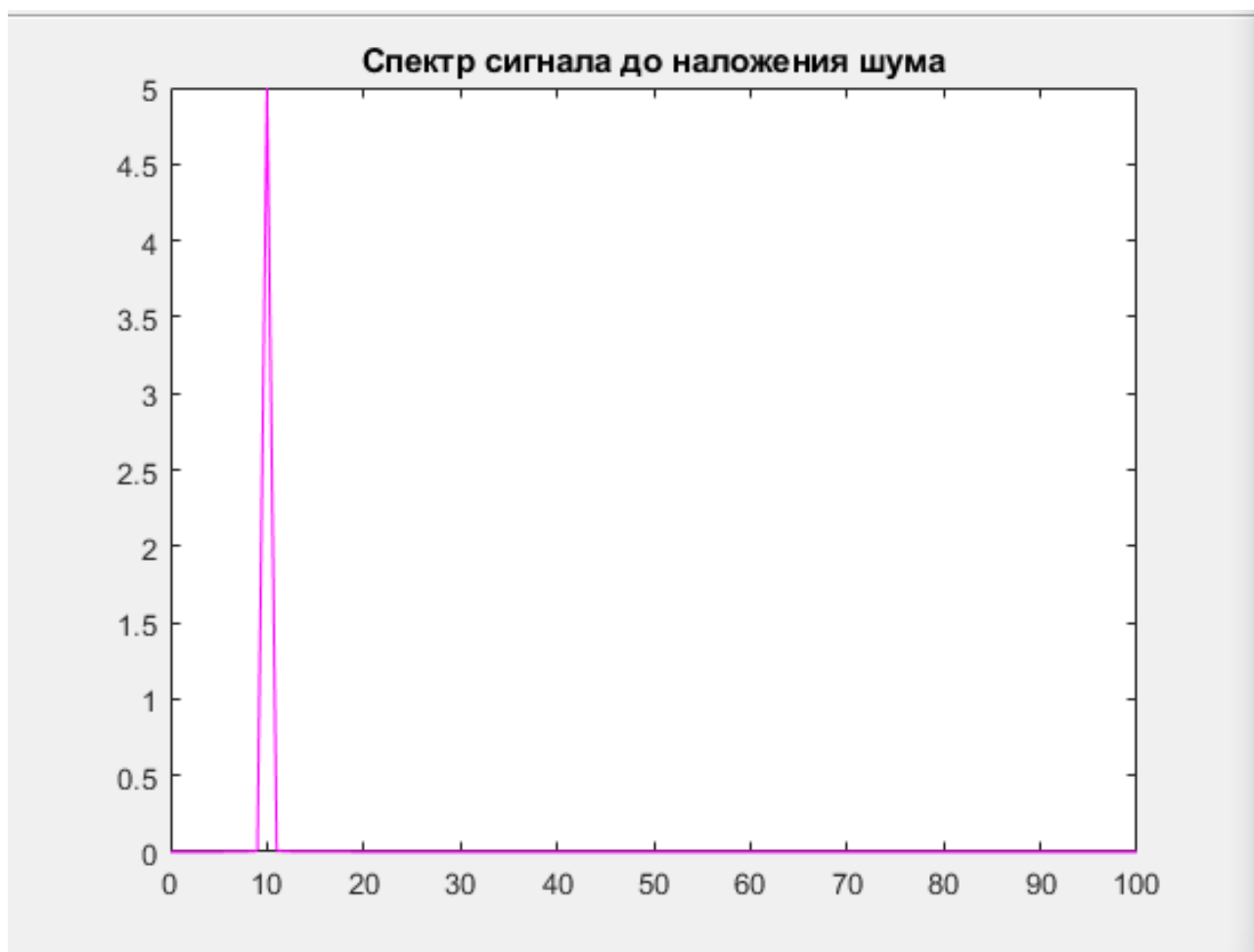


Рис. 4.3: Спектр исходного сигнала

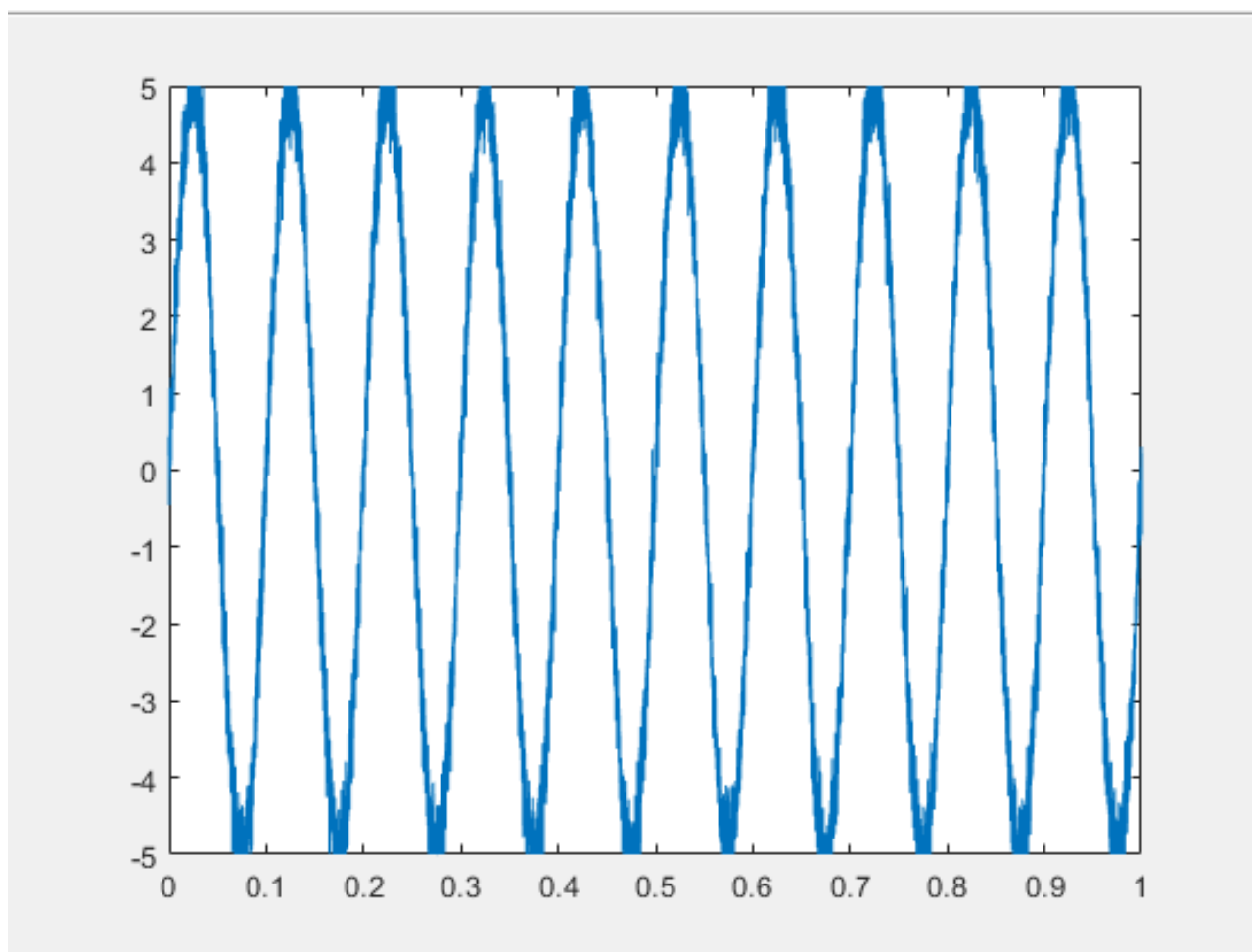


Рис. 4.4: Зашумленный сигнал.

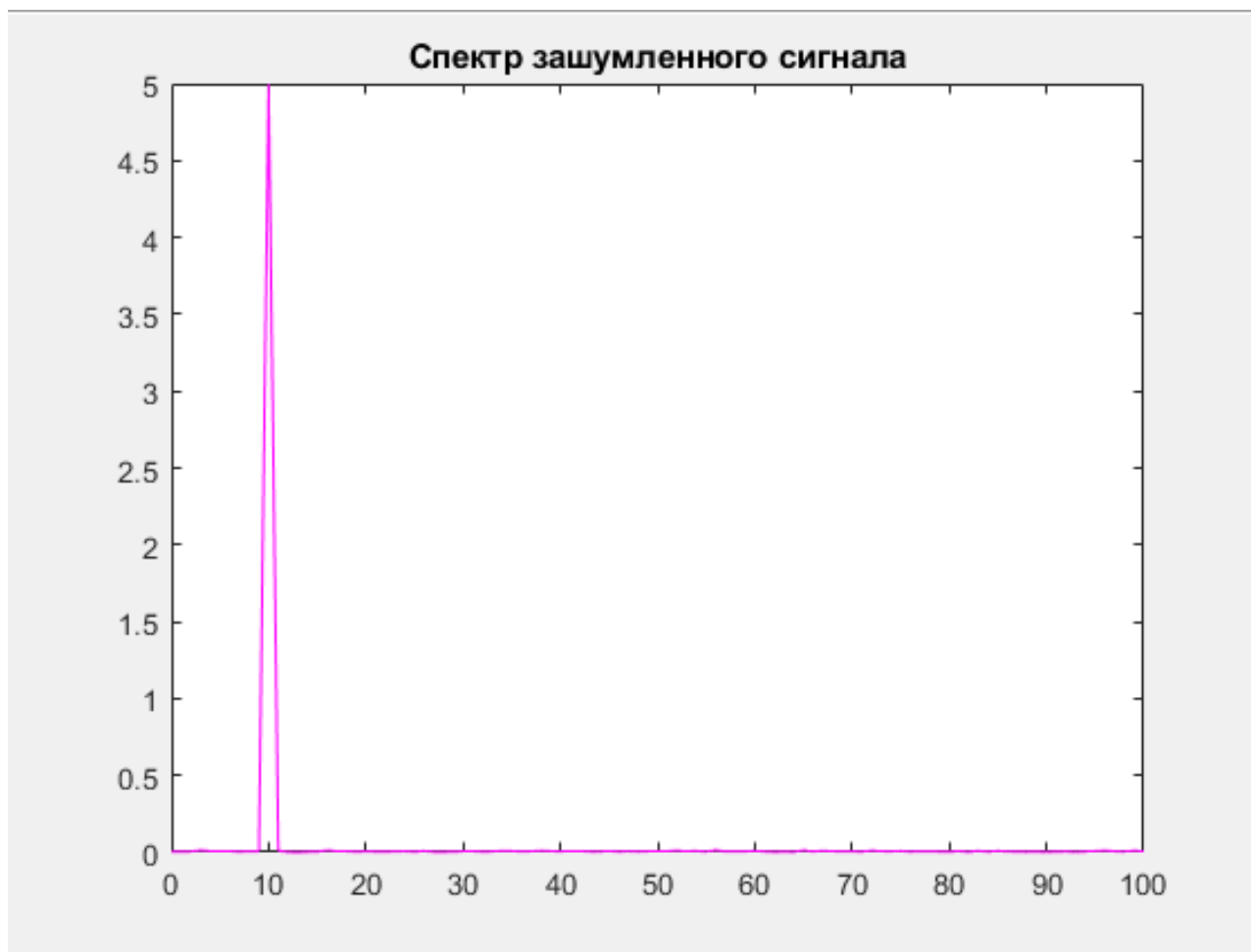


Рис. 4.5: Спектр зашумленного сигнала

4.2 Фильтрация сигнала с шумом

Была проведена фильтрация сигнала с шумом с использованием фильтра Kaiser. Данный фильтр был синтезирован с помощью средства Filter Design, встроенного в MATLAB.

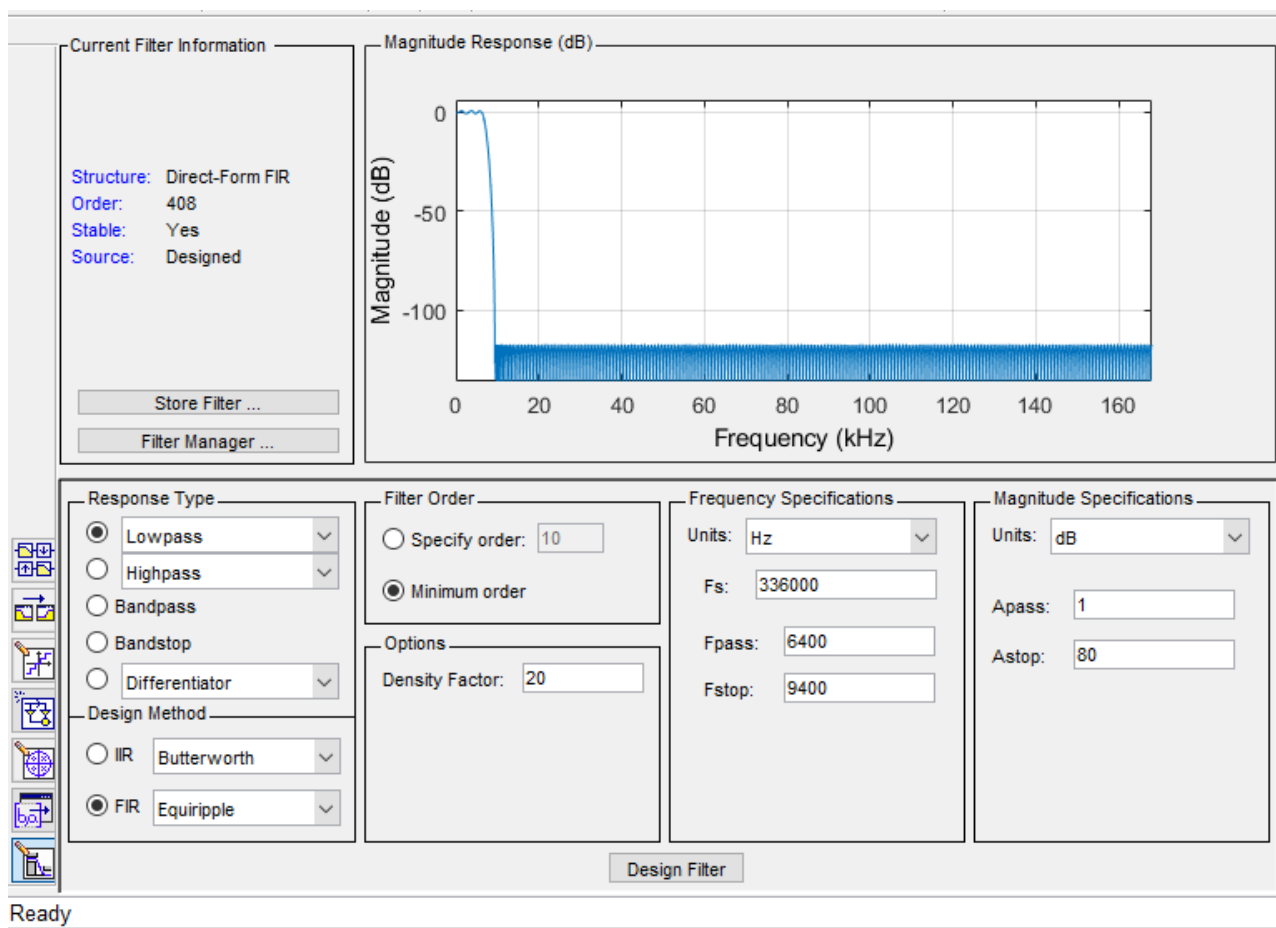


Рис. 4.6: Настройки фильтра в Filter Designer

```

function Hd = flt
%FLT Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.4 and DSP System Toolbox 9.6.
% Generated on: 20-May-2018 23:01:33

% Equiripple Lowpass filter designed using the FIRPM function.

% All frequency values are in Hz.
Fs = 336000; % Sampling Frequency

Fpass = 7400; % Passband Frequency
Fstop = 9400; % Stopband Frequency
Dpass = 0.057501127785; % Passband Ripple
Dstop = 0.0001; % Stopband Attenuation
dens = 20; % Density Factor

% Calculate the order from the parameters using FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fpass, Fstop]/(Fs/2), [1 0], [Dpass, Dstop]);

% Calculate the coefficients using the FIRPM function.
b = firpm(N, Fo, Ao, W, {dens});
Hd = dfilt.dffir(b);

% [EOF]

```

Рис. 4.7: Код сгенерированный средством Filter Design

```

10 - t5=flt;
11 - y2=filter(t5,y);
12 - figure;
13 - plot(t,y2);
14 - axis([0 1 -5 5])
15 - ffts2=abs(fft(y,N)); % преобразование Фурье по модулю
16 - ffts2=2*ffts2./N;% нормализация
17 - ffts2(1)=ffts2(1)/2;% нормировка постоянной составляющей в спектре
18 - f2=0:fs1/N:fs1/2-fs1/N;% вектор частот
19 - figure;
20 - plot(f2,ffts2(1:length(f2)),'m');%вывод спектра отфильтрованного сигнала
21 - axis([0 1000 0 5])

```

Рис. 4.8: Код MATLAB для использования фильтра к зашумленному сигналу.

В итоге был получен отфильтрованный сигнал

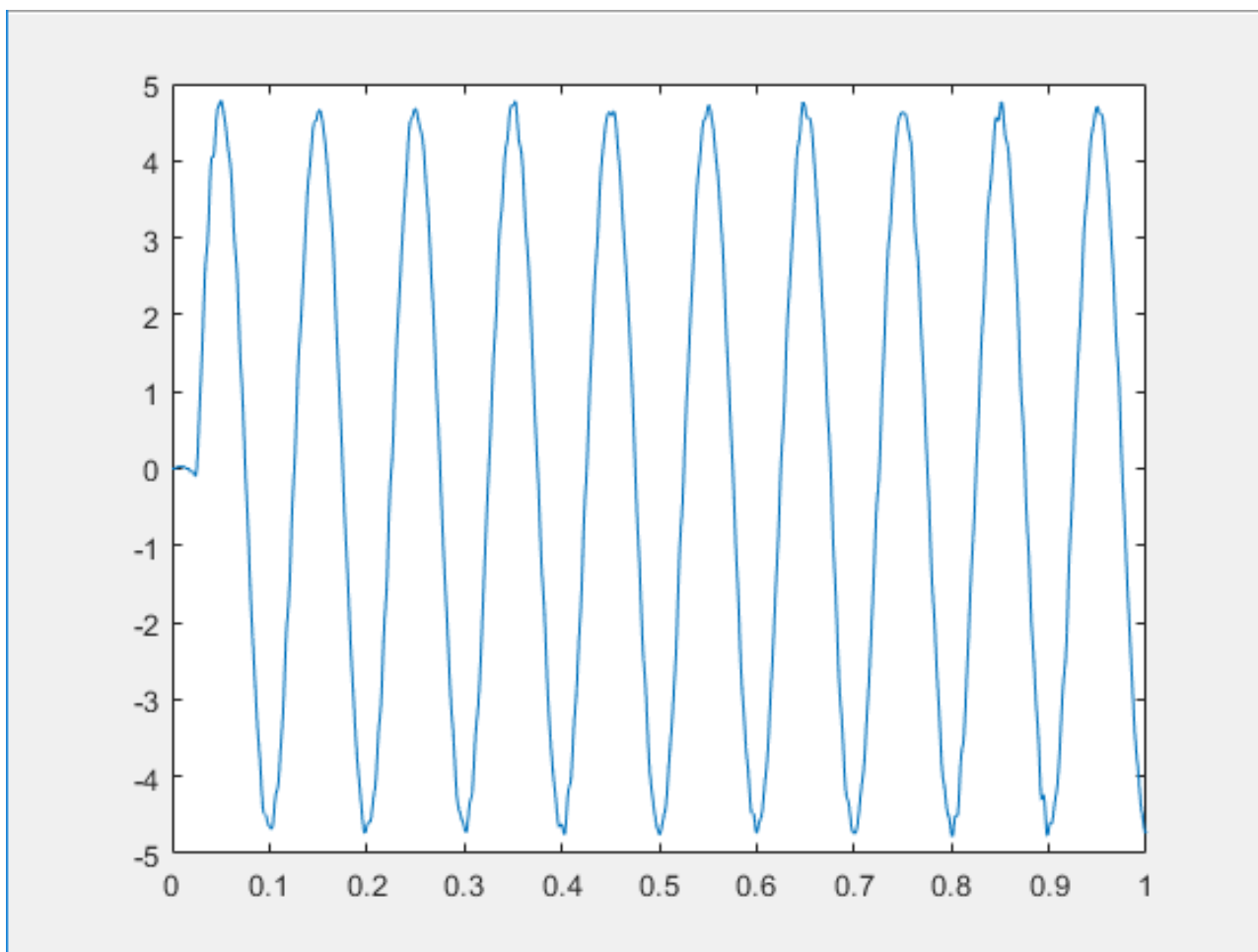


Рис. 4.9: Отфильтрованный сигнал.

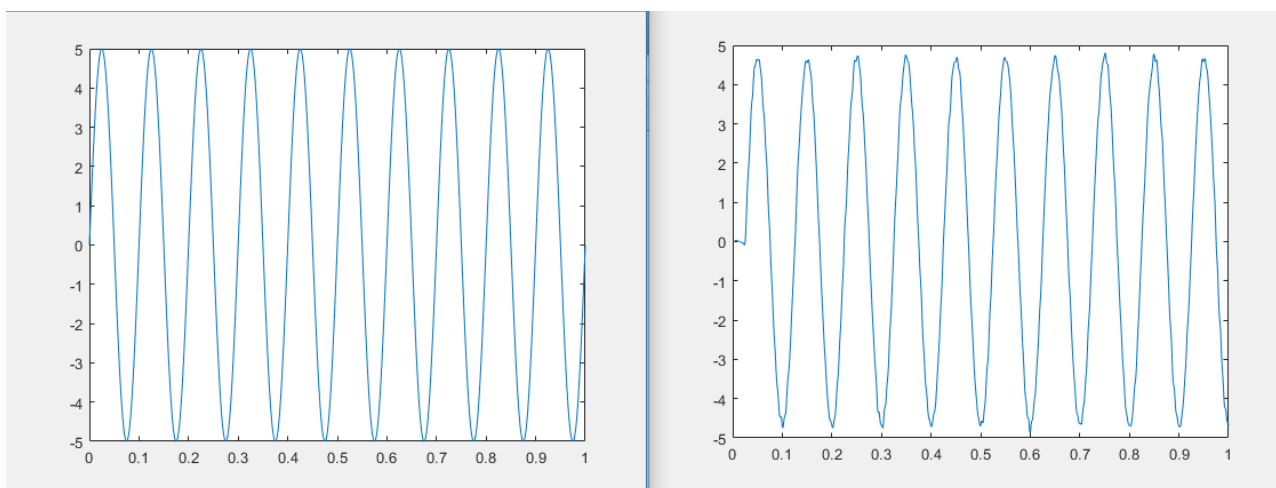


Рис. 4.10: Исходный сигнал и отфильтрованный сигнал.

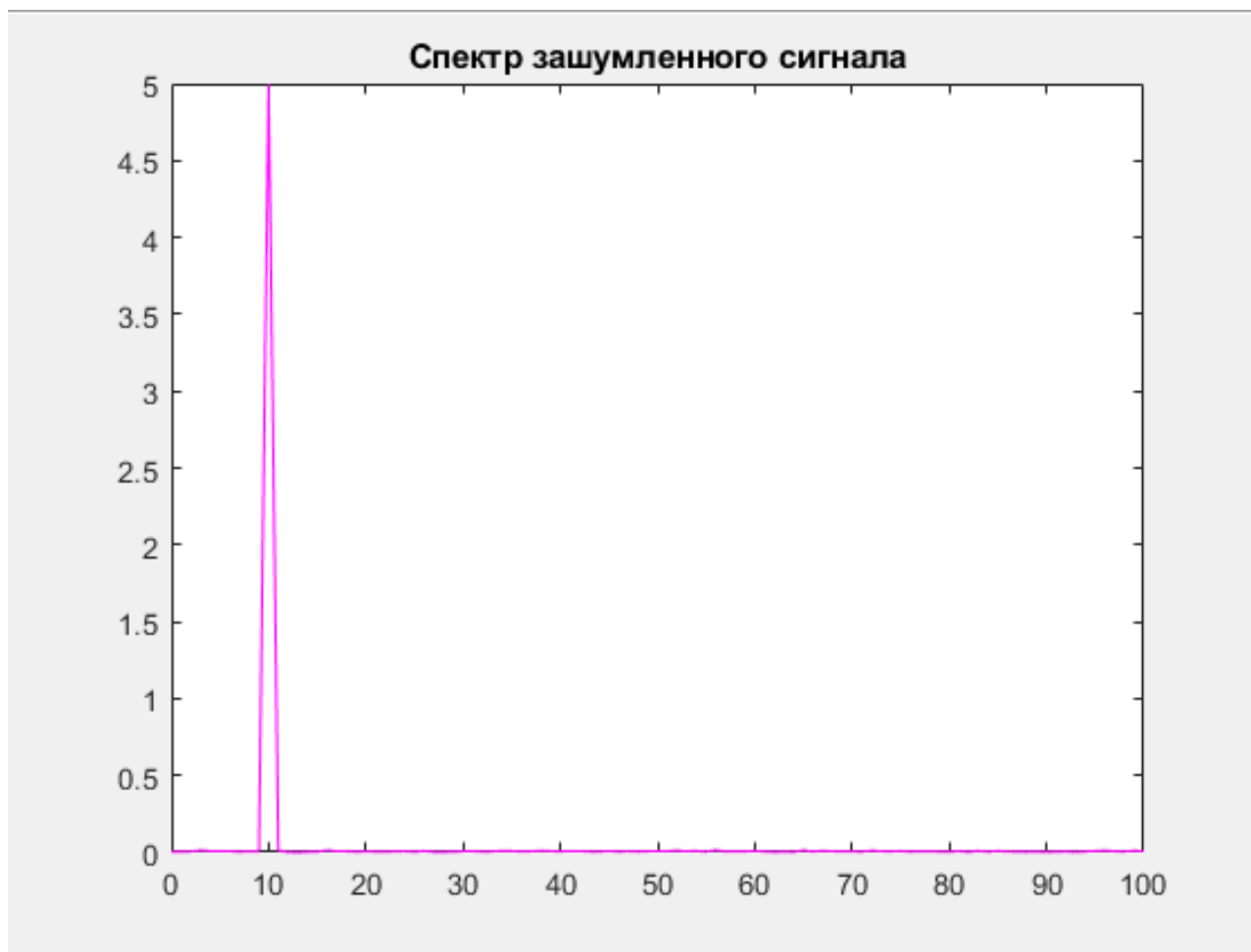


Рис. 4.11: Спектр отфильтрованного сигнала.

5 Выводы

В ходе работы была исследована работа фильтра с помощью средства Filter Design, встроенного в MATLAB. Было установлено, что фильтр работает привильно, так как отфильтрованный сигнал практически соответствует исходному. Имеются некоторые различия связанные с тем, что шум имеет низкие частоты, которые не удастся подавить.