

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Ст. преподаватель

должность, уч. степень, звание

подпись, дата

Е.К. Григорьев

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

ФИЛЬТРАЦИЯ СИГНАЛОВ В ПАКЕТЕ MATLAB

по курсу: МОДЕЛИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

Е.Д.Тегай

инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Изучение методов фильтрации сигналов в системах цифровой обработки сигналов с помощью пакета компьютерного моделирования MATLAB.

Индивидуальный вариант

Содержание индивидуального варианта продемонстрировано на рисунке 1, где для удобства необходимые данные выделены жёлтым цветом.

Номер варианта	Семейство фильтра	Порядок фильтра	Тип фильтра	f_{c1} , Гц	f_{c2} , Гц	Пульсации в полосе пропускания, Дб	Пульсации в полосе подавления, Дб
1	Бесселя	8	ФНЧ	100	-	-	-



2	Баттерворта	6	ФВЧ	400	-	-	-
3	Чебышева 1	5	РФ	150	300	1	-
4	Эллиптический	9	ФНЧ	75	-	0.5	1
5	Чебышева 2	8	ФВЧ	125	-	-	0.5
6	Бесселя	6	РФ	300	400	-	-
7	Баттерворта	7	ФНЧ	200	-	-	-
8	Чебышева 1	9	ФВЧ	300	-	0.5	-
9	Эллиптический	10	РФ	175	250	3	3.5
10	Чебышева 2	7	ФНЧ	100	-	-	0.5
11	Бесселя	6	ФВЧ	350	-	-	-
12	Баттерворта	5	РФ	200	400	-	-
13	Чебышева 1	7	ФНЧ	150	-	1	-
14	Эллиптический	8	ФВЧ	400	-	1	2
15	Чебышева 2	5	РФ	150	350	-	1
16	Бесселя	6	ФНЧ	120	-	-	-
17	Баттерворта	7	ФВЧ	150	-	-	-
18	Чебышева 1	4	РФ	300	500	0.5	-
19	Эллиптический	6	РФ	50	100	0.5	1

Рисунок 1 – Индивидуальный вариант

Ход работы

Для начала необходимо подготовить тестовые сигналы. Пусть один сигнал будет в полосе пропускания фильтра, а второй – в полосе подавления. Частота и фаза первого сигнала будет 60 Гц и 0 радиан, а частота и фаза второго – 15 Гц и $\pi/2$ радиан. Амплитуда каждого сигнала – 5 В. Частота

дискретизации – 1024 Гц. Итого тестовый сигнал будет представлять собой сумму первого и второго сигналов.

Код программы

```
% Параметры первого сигнала (в полосе пропускания фильтра)
f1 = 60; % Частота первого сигнала (в Гц)
phi1 = 0; % Фаза первого сигнала (в радианах)
A1 = 5; % Амплитуда первого сигнала (в вольтах)

% Параметры второго сигнала (в полосе подавления фильтра)
f2 = 15; % Частота второго сигнала (в Гц)
phi2 = pi/2; % Фаза второго сигнала (в радианах)
A2 = 5; % Амплитуда второго сигнала (в вольтах)

% Частота дискретизации
Fs = 1024; % Частота дискретизации (в Гц)

% Временной вектор
t = (0:(1/Fs):3-(1/Fs)); % Вектор времени от 0 до 3 секунд с шагом 1/Fs

% Генерация первого и второго сигналов
signal1 = A1 * sin(2*pi*f1*t + phi1);
signal2 = A2 * sin(2*pi*f2*t + phi2);

% Тестовый сигнал (сумма первого и второго сигналов)
test_signal = signal1 + signal2;

% Построение графиков
figure;
plot(t, signal1);
xlabel('Время (с)');
ylabel('Амплитуда');
title('Первый сигнал');
grid on;

figure;
plot(t, signal2);
xlabel('Время (с)');
ylabel('Амплитуда');
title('Второй сигнал');
grid on;

figure;
plot(t, test_signal);
xlabel('Время (с)');
ylabel('Амплитуда');
title('Тестовый сигнал');
grid on;
```

Графики

П

о

л

у

ч

е

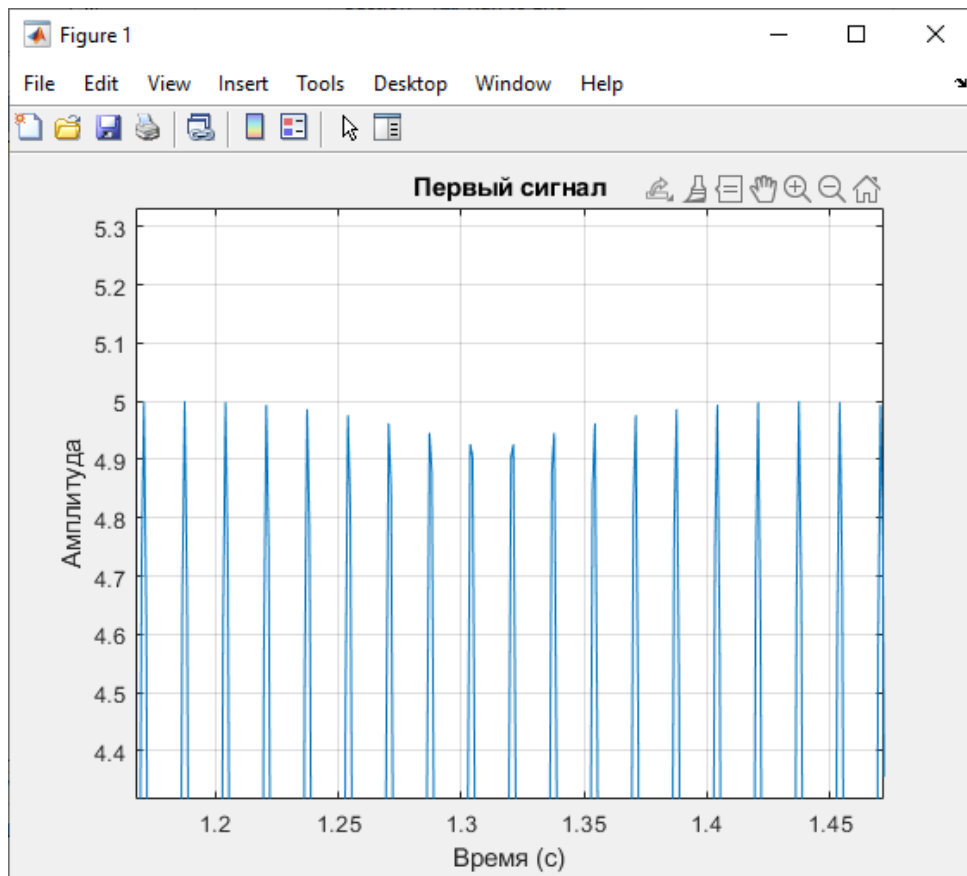


Рисунок 2 – Первый сигнал (приближённый вид)

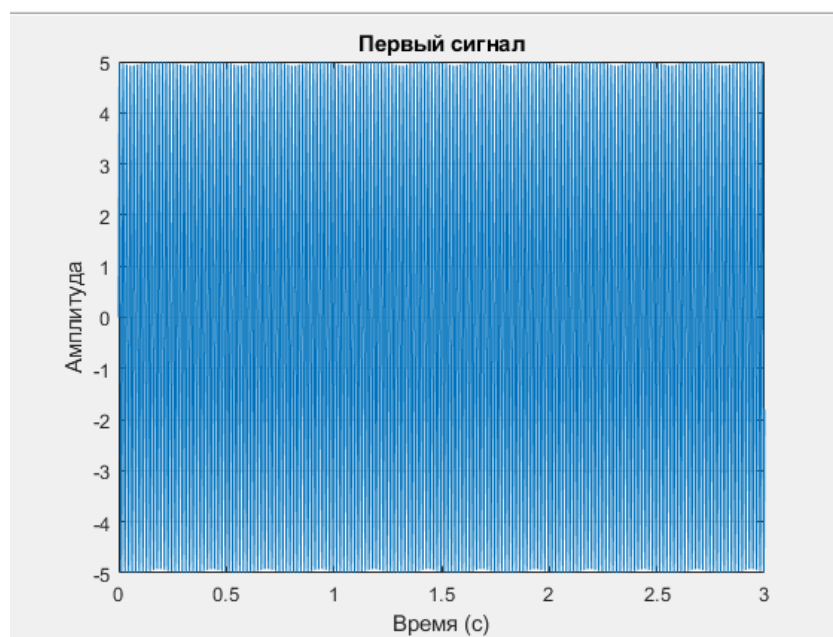


Рисунок 3 – Первый сигнал

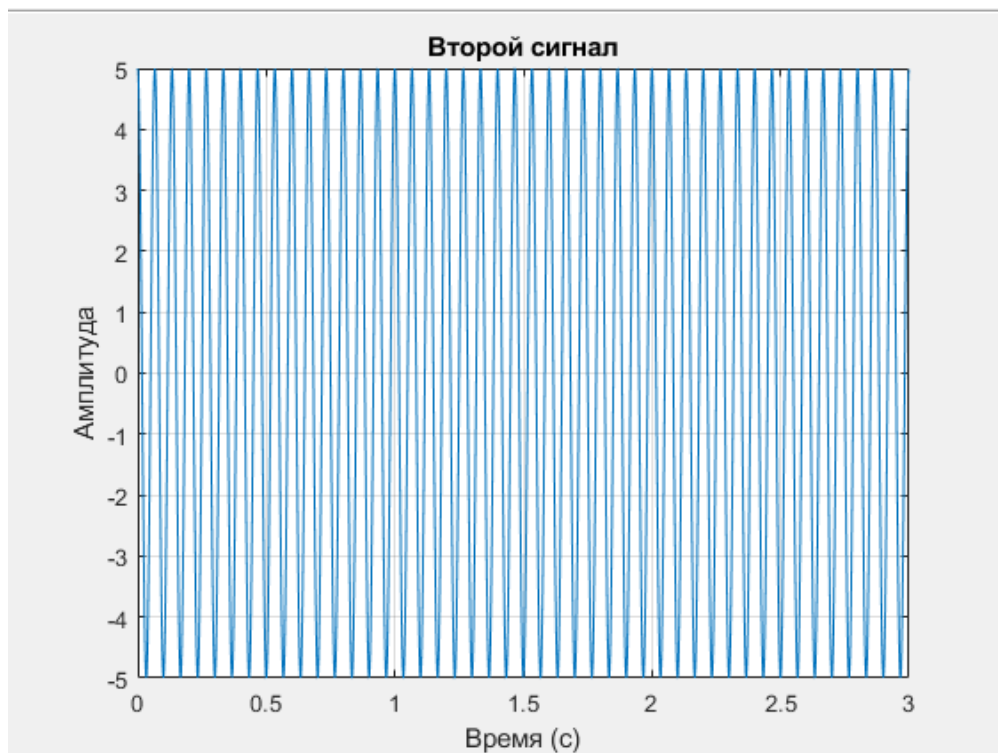


Рисунок 4 – Второй сигнал

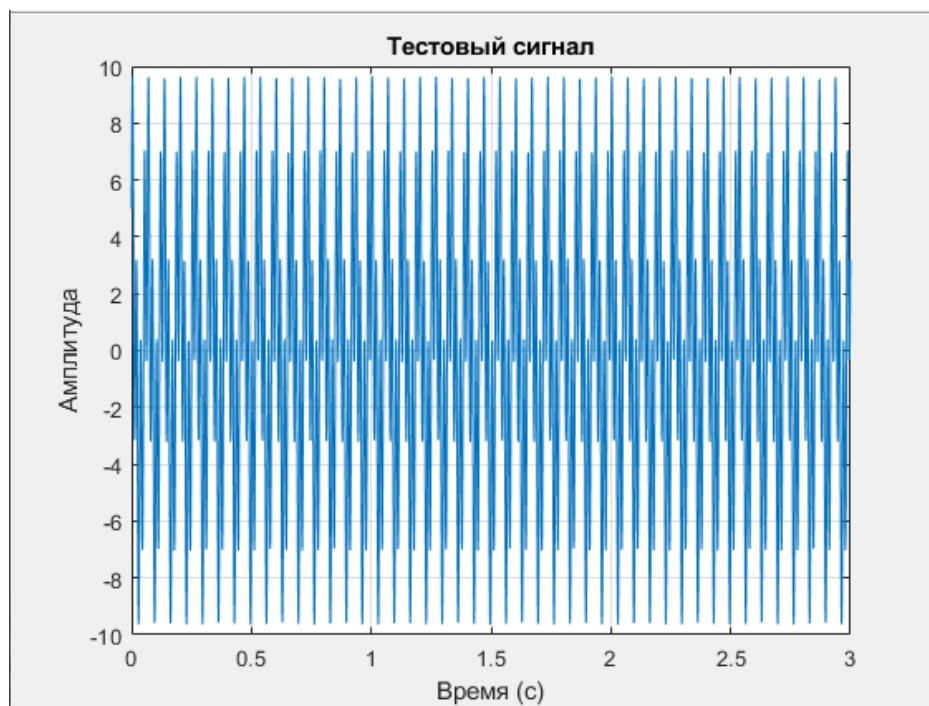


Рисунок 5 – Тестовый сигнал

Далее необходимо подготовить аналоговый ФНЧ прототип в соответствии с индивидуальным вариантом. Для этого воспользуемся функцией `ellipar`. Она используется для расчёта нулей (z) и полюсов (p), а также коэффициента усиления (k) аналогового прототипа эллиптического фильтра. В качестве аргументов данной функции передаются значения

порядка, максимальные и минимальные пульсации в полосах пропускания и подавления соответственно.

Далее, используя полученные нули, полюса и коэффициент усиления, можно построить аналоговый ФНЧ фильтр с помощью функции `zp2tf`, чтобы получить передаточную функцию в форме полиномов числителя и знаменателя. Полученный код продемонстрирован ниже.

Листинг кода

```
% ХАРАКТЕРИСТИКИ ФИЛЬТРА
N = 6; % Порядок фильтра
Rp = 0.5;
Rs = 1;
Fc1 = 50;
Fc2 = 100;

[z,p,k]=ellipap(N,Rp,Rs);
[b,a] = zp2tf(z,p,k);
```

Теперь необходимо получить графики частотных характеристик фильтра. Это делается с помощью функции `freqs`. Полученный код и график продемонстрированы ниже.

Листинг кода

```
% Параметры первого сигнала (в полосе пропускания фильтра)
f1 = 60; % Частота первого сигнала (в Гц)
phi1 = 0; % Фаза первого сигнала (в радианах)
A1 = 5; % Амплитуда первого сигнала (в вольтах)

% Параметры второго сигнала (в полосе подавления фильтра)
f2 = 15; % Частота второго сигнала (в Гц)
phi2 = pi/2; % Фаза второго сигнала (в радианах)
A2 = 5; % Амплитуда второго сигнала (в вольтах)

% Частота дискретизации
Fs = 1024; % Частота дискретизации (в Гц)

% Временной вектор
t = (0:(1/Fs):3-(1/Fs)); % Вектор времени от 0 до 3 секунд с шагом 1/Fs

% Генерация первого и второго сигналов
signal1 = A1 * sin(2*pi*f1*t + phi1);
signal2 = A2 * sin(2*pi*f2*t + phi2);

% Тестовый сигнал (сумма первого и второго сигналов)
test_signal = signal1 + signal2;

% ХАРАКТЕРИСТИКИ ФИЛЬТРА
N = 6; % Порядок фильтра
Rp = 0.5;
Rs = 1;
```

```

Fc1 = 50;
Fc2 = 100;

[z,p,k]=ellipap(N,Rp,Rs);
[b,a] = zp2tf(z,p,k);
figure()
freqs(b,a);
title('Частотная характеристика ФНЧ прототипа');

```

График

Полученный график показан на рисунке 6.

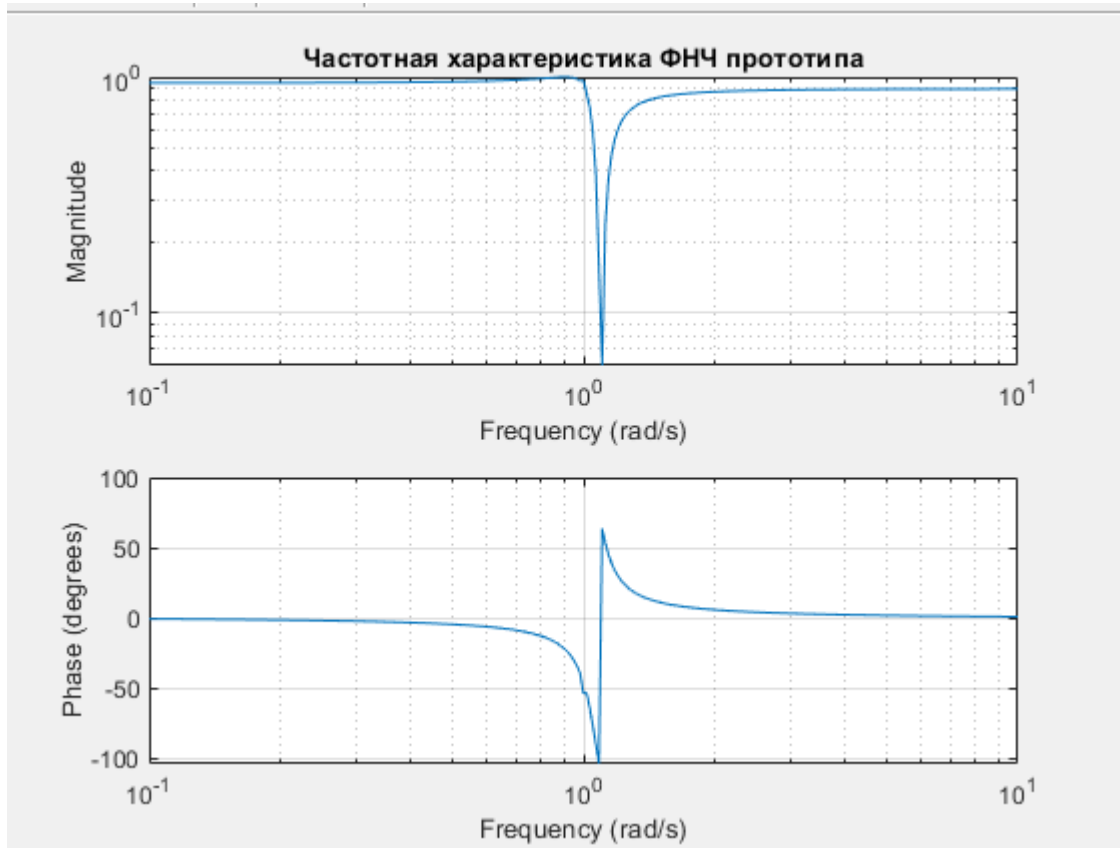


Рисунок 6 – Частотная характеристика ФНЧ прототипа

После этого необходимо преобразовать полученный прототип в фильтр с характеристиками, приведёнными в задании индивидуального варианта. Для начала необходимо перевести частоты в угловые. Помимо этого, ещё нужно вычислить частоту среза и ширину полосы подавления. Частота среза является квадратным корнем от произведения угловых частот, а ширина полосы подавления – их разностью. Преобразование реализуется с помощью функций:

- `lp2bs`, аргументами которой являются полученные ранее коэффициенты `a` и `b`, частота среза и ширина полосы подавления
- `tf2zp`, аргументами которой являются полученные ранее коэффициенты

- `zp2tf`, аргументами которой являются полученные ранее нули, полюсы передаточной функции, а также коэффициент масштабирования.

Итоговый код продемонстрирован ниже.

Листинг кода

```
% Угловые частоты
w1 = 2*pi*Fc1;
w2 = 2*pi*Fc2;

Fc = sqrt(w1*w2); % Частота среза
Bw = w2 - w1; % Ширина полосы подавления

% Преобразование в РФ
[bt,at] = lp2bs(b,a,Fc,Bw); % Преобразование коэффициентов
[new_z,new_p,new_k]=tf2zp(bt,at); % Преобразование передаточной функции в форму нулей и полюсов
[b_bs,a_bs] = zp2tf(new_z,new_p,new_k); % Обратное преобразование из формы нулей и полюсов в форму коэф. перед. фун.
```

Далее необходимо применить билинейное Z-преобразование. Это делается с помощью функции `bilinear`. В качестве аргументов ей передаются полученные ранее значения `new_z`, `new_p` и `new_k`, а также значение частоты дискретизации. Полученный код продемонстрирован ниже.

```
% Применение билинейного Z-преобразования
[dig_z, dig_p, dig_k] = bilinear(new_z, new_p,new_k, Fs);
```

Затем нужно полученный цифровой фильтр преобразовать в каскадную форму, состоящую из секций второго порядка. А затем нужно получить коэффициенты данного фильтра. Это делается с помощью функции `zp2tf`. Полученный код продемонстрирован ниже.

```
% Приведение к каскадной реализации
[b_kas, a_kas] = zp2tf(dig_z, dig_p, dig_k);
```

После этого строятся графики частотных характеристик полученного цифрового фильтра. Это делается с помощью функции `freqz`. Полученный код показан ниже. Результат продемонстрирован на рисунках 7 – 8.

Листинг кода

```
% Построение графика частотной характеристики цифрового фильтра
figure();
freqz(b_kas, a_kas, 512, Fs);
```



```

xlabel('Частота (Гц)');
ylabel('Амплитуда');
title('Частотная характеристика цифрового фильтра');
grid on;

% Построение графика частотной характеристики цифрового фильтра
figure();
[h, freq] = freqz(b_kas, a_kas, 512, Fs);
plot(freq, abs(h)); % Построение графика

hold on;
plot([Fc1, Fc1], [0, max(abs(h))], 'r--'); % Отметка частоты среза Fc1
plot([Fc2, Fc2], [0, max(abs(h))], 'g--'); % Отметка частоты среза Fc2
hold off;

xlabel('Частота (Гц)');
ylabel('Амплитуда');
title('Частотная характеристика цифрового фильтра');
legend('Амплитуда', 'Fc1', 'Fc2'); % Добавление легенды
grid on;

```

Графики

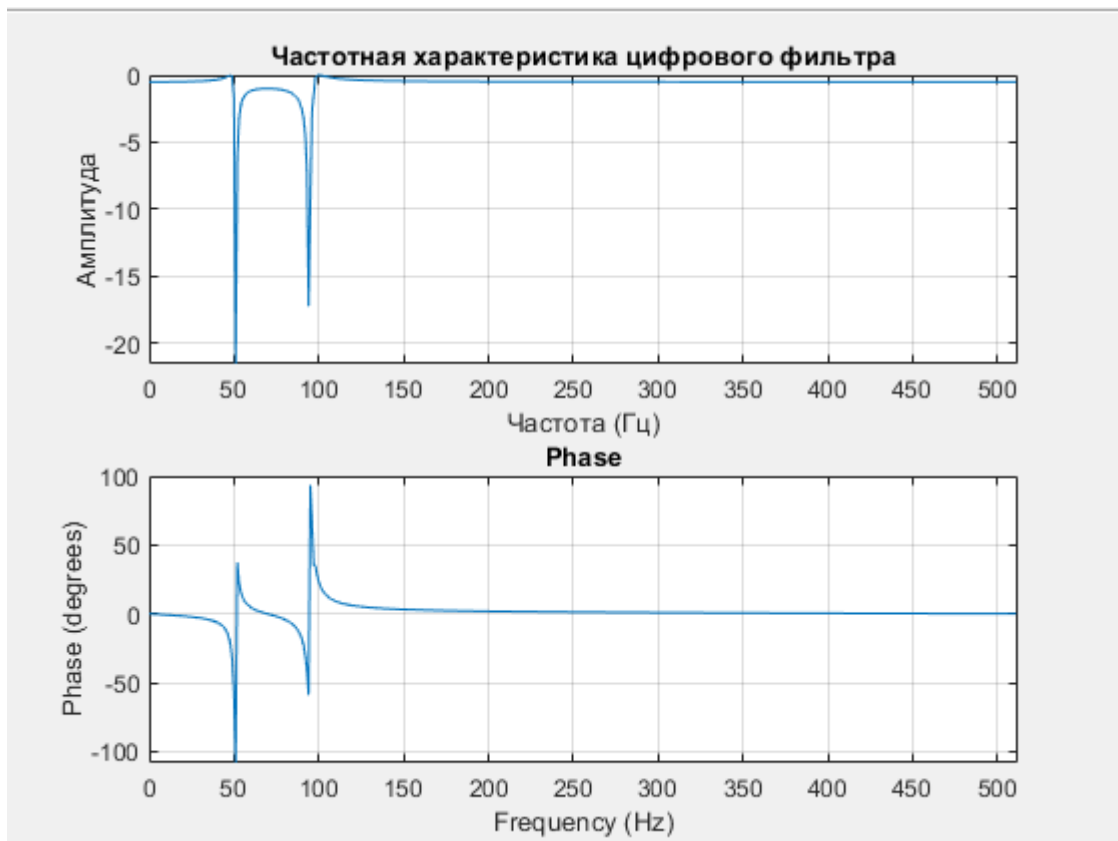


Рисунок 7 – Частотная характеристика цифрового фильтра

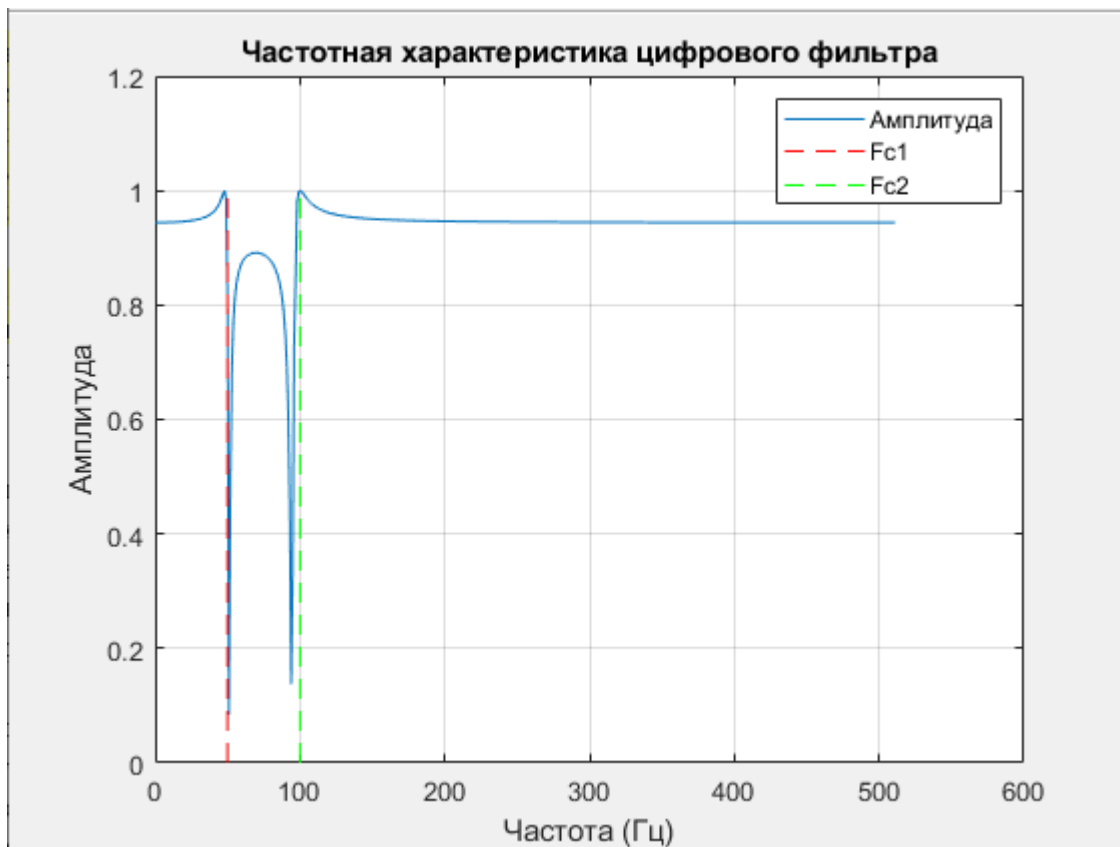


Рисунок 8 – Частотная характеристика (с отметкой)

Затем необходимо пропустить тестовый сигнал через полученный цифровой фильтр. Для удобства продублируем вывод графика исходного сигнала. Полученный код продемонстрирован ниже. На рисунках 9 – 10 показан сигнал до и после фильтрации.

Листинг кода

```
% Фильтрация сигнала
filtered_signal = filter(b_kas,a_kas,test_signal);
filtered_signal1 = filter(b_kas,a_kas,signal1);
%
figure()
plot(t,test_signal)
grid on
title ('Сигнал на входе фильтра')
xlabel('Время (с)')
ylabel('Амплитуда')

figure()
plot(t,filtered_signal)
grid on
title ('Сигнал на выходе фильтра')
xlabel('Время (с)')
ylabel('Амплитуда')

figure()
plot(t,signal1,'g')
grid on
```

```

hold on
plot (t,filtered_signal1,'r')
hold off
title ('Сигнал до (зелёный) и после (красный) фильтрации ')
xlabel('Время (с)')
ylabel('Амплитуда')

```

Графики

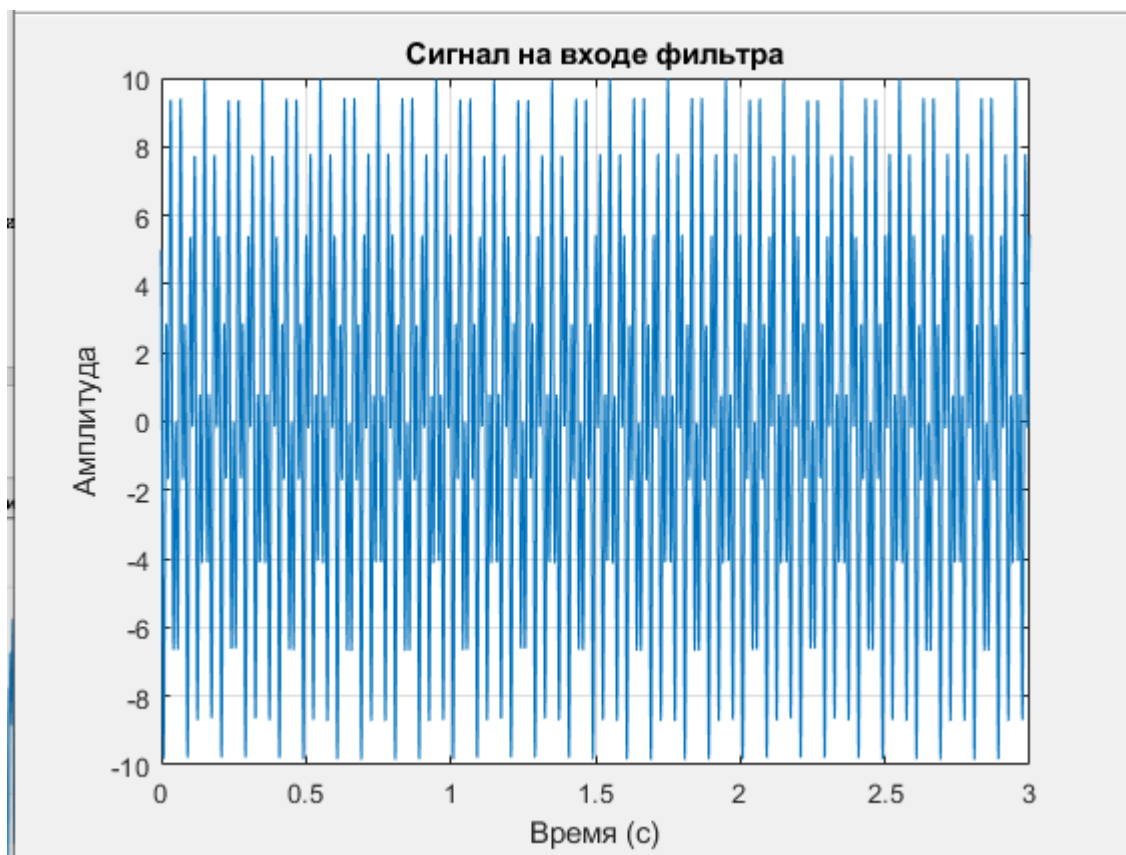


Рисунок 9 – Сигнал до фильтрации

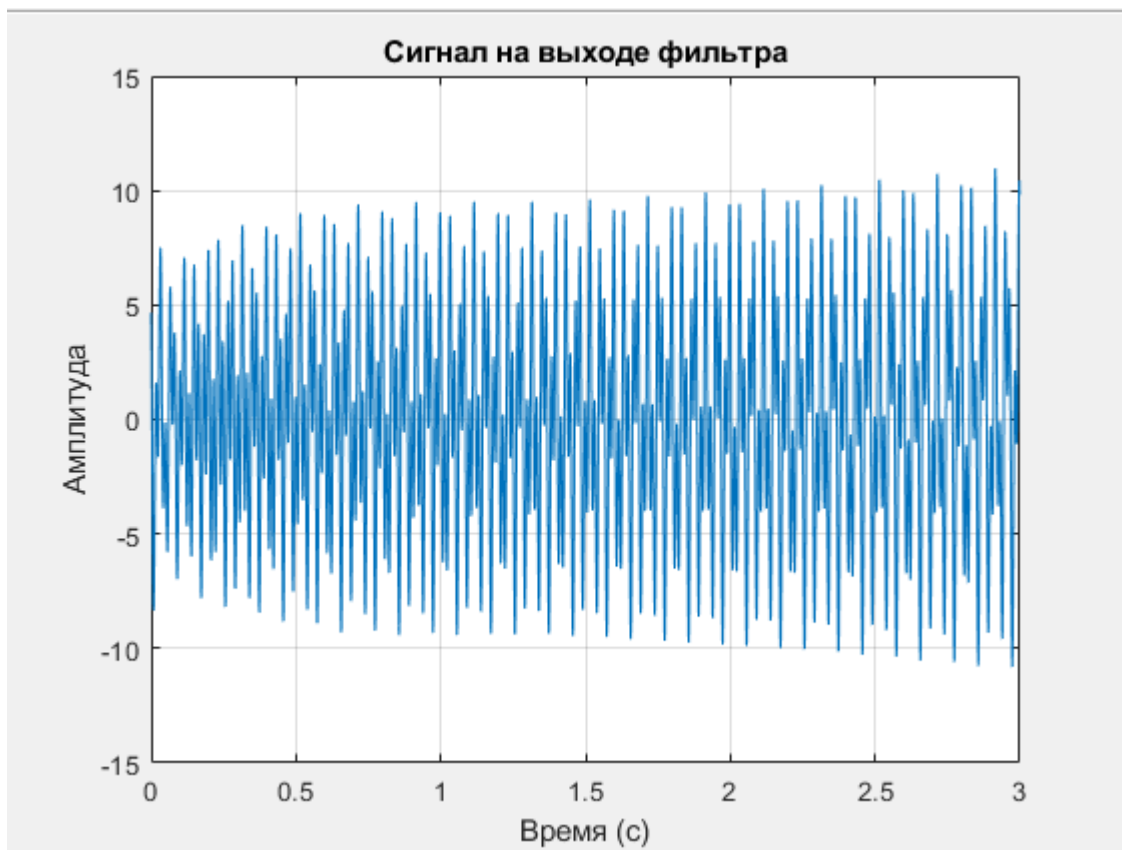


Рисунок 10 – Сигнал после фильтрации

Далее необходимо построить один график, где одновременно показан сигнал в полосе пропускания до и после фильтрации. Полученный код продемонстрирован ниже. На рисунке 11 показан полученный график.

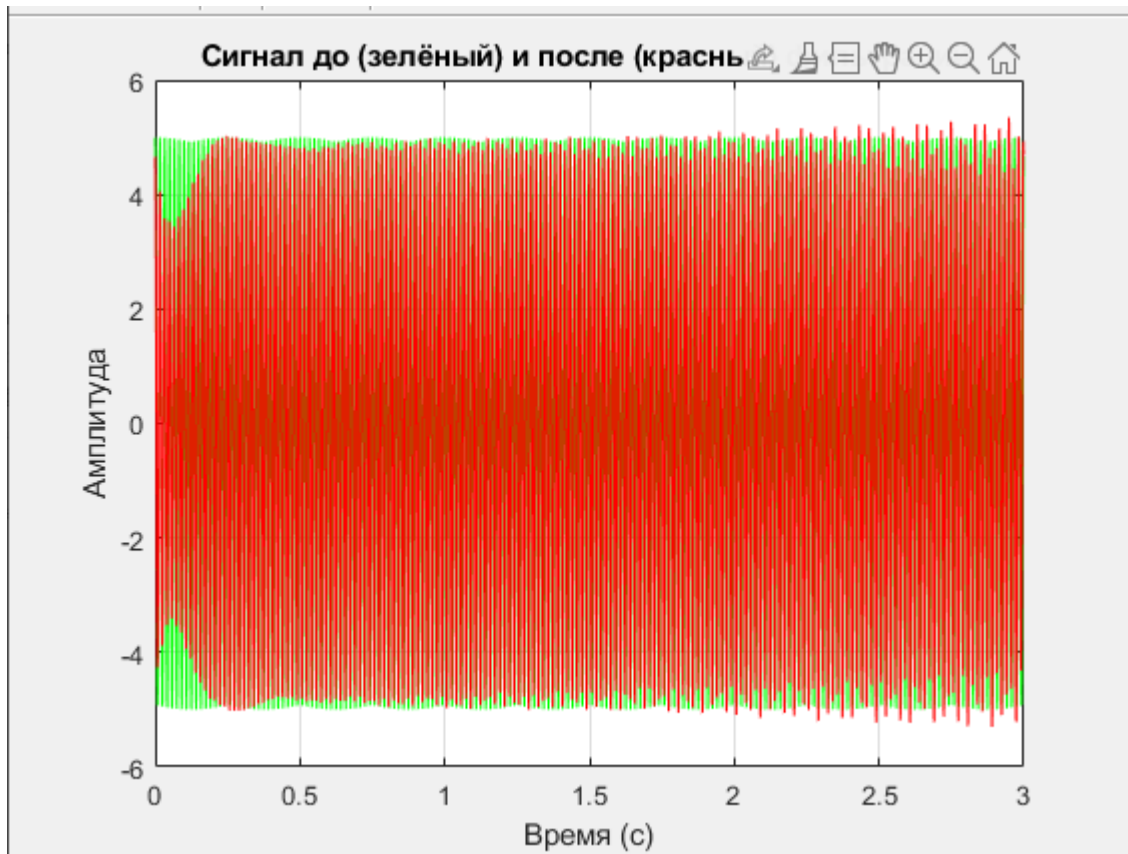


Рисунок 11 – Сигнал в полосе пропускания до и после фильтрации

Затем нужно построить графики групповой и фазовой задержки реализованного цифрового фильтра. Это делается с помощью функций графики показаны на рисунках 12 – 13.

Листинг кода

```
% Фазовая задержка
figure()
phasedelay(b_kas,a_kas,1024,'whole');
title('Фазовая задержка фильтра')
% Групповая задержка
figure()
grpdelay(b_kas,a_kas,1024);
title('Групповая задержка фильтра')
```

Графики

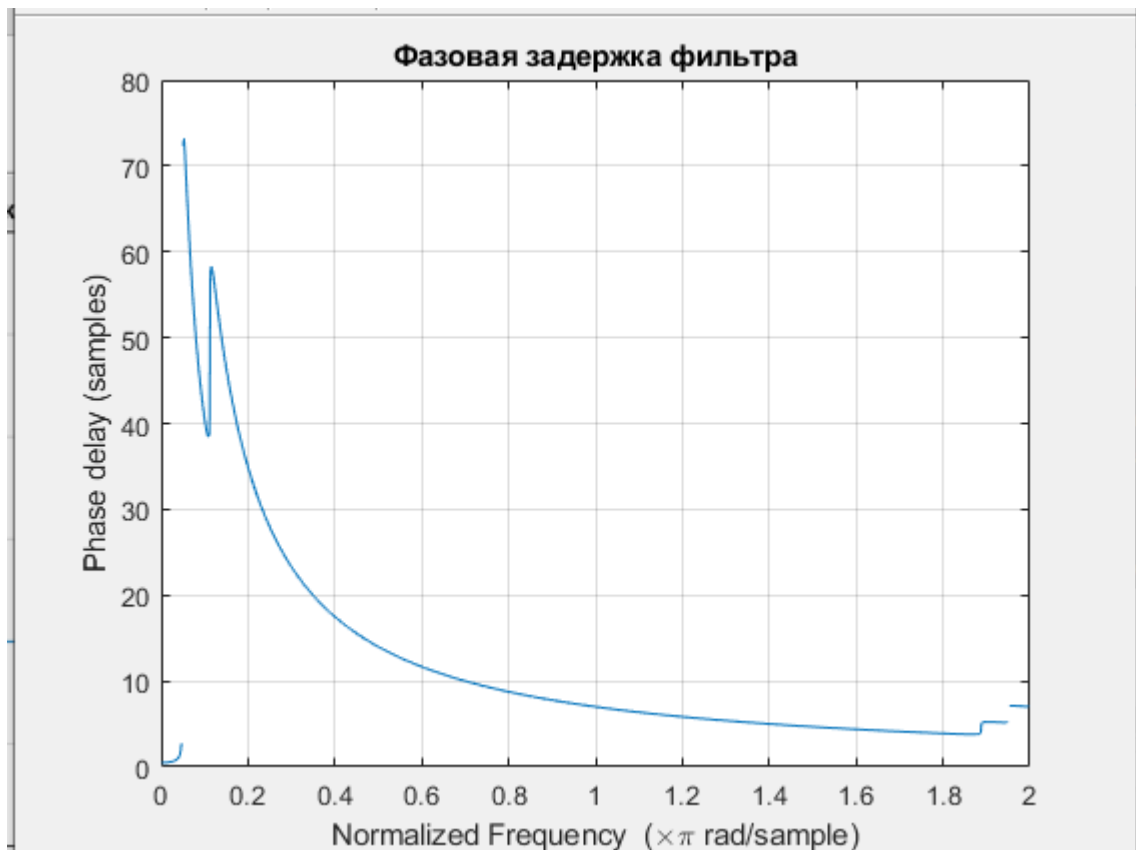


Рисунок 12 – Фазовая задержка

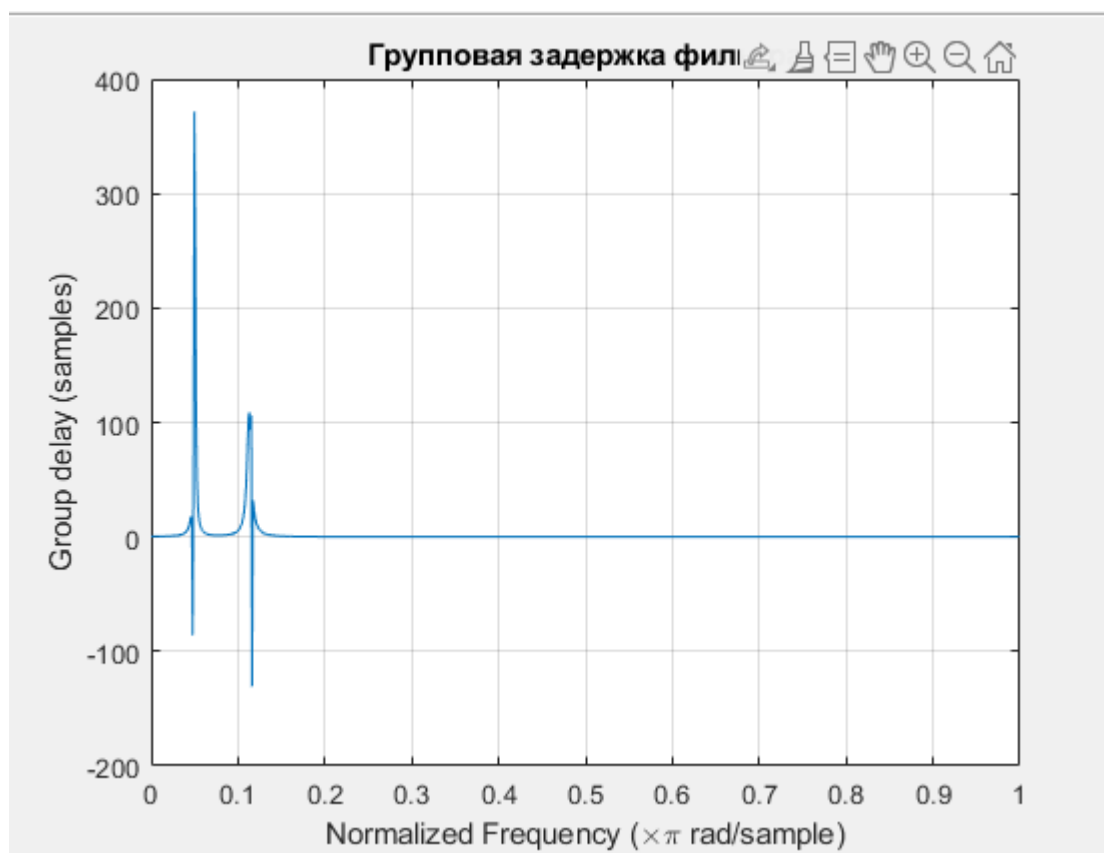


Рисунок 13 – Групповая задержка

После этого к тестовому сигналу нужно добавить аддитивный белый гауссовский шум с нулевым математическим ожиданием и среднеквадратическим отклонением, равным единице, а затем пропустить данный сигнал через фильтр. Код продемонстрирован ниже. На рисунках 14 –

Листинг кода

```
% Белый гауссовский шум
M = 0; % Мат. ожидание
sigma = 1; % Ср. отклонение
N=length(test_signal);
noise = normrnd(M, sigma, [1 N]);
noised_test=test_signal+noise; % Генерация сигнала с шумом
% Фильтрация сигнала с шумом
noised_out=filter(b_kas,a_kas,noised_test);
noised_signal1=filter(b_kas,a_kas,signal1);

figure()
plot(t,noised_test)
grid on
title ('Сигнал с шумом на входе фильтра')
xlabel('Время (с)')
ylabel('Амплитуда')

figure()
plot(t,noised_out)
grid on
title ('Сигнал с шумом на выходе фильтра')
xlabel('Время (с)')
ylabel('Амплитуда')

figure()
plot(t,signal1,'g')
grid on
hold on
plot (t,noised_signal1,'r')
hold off
title ('Сигнал в полосе пропускания фильтра (зелёным) и зашумлённый сигнал на выходе фильтра (красным)')
xlabel('Время (с)')
ylabel('Амплитуда')
```

Графики

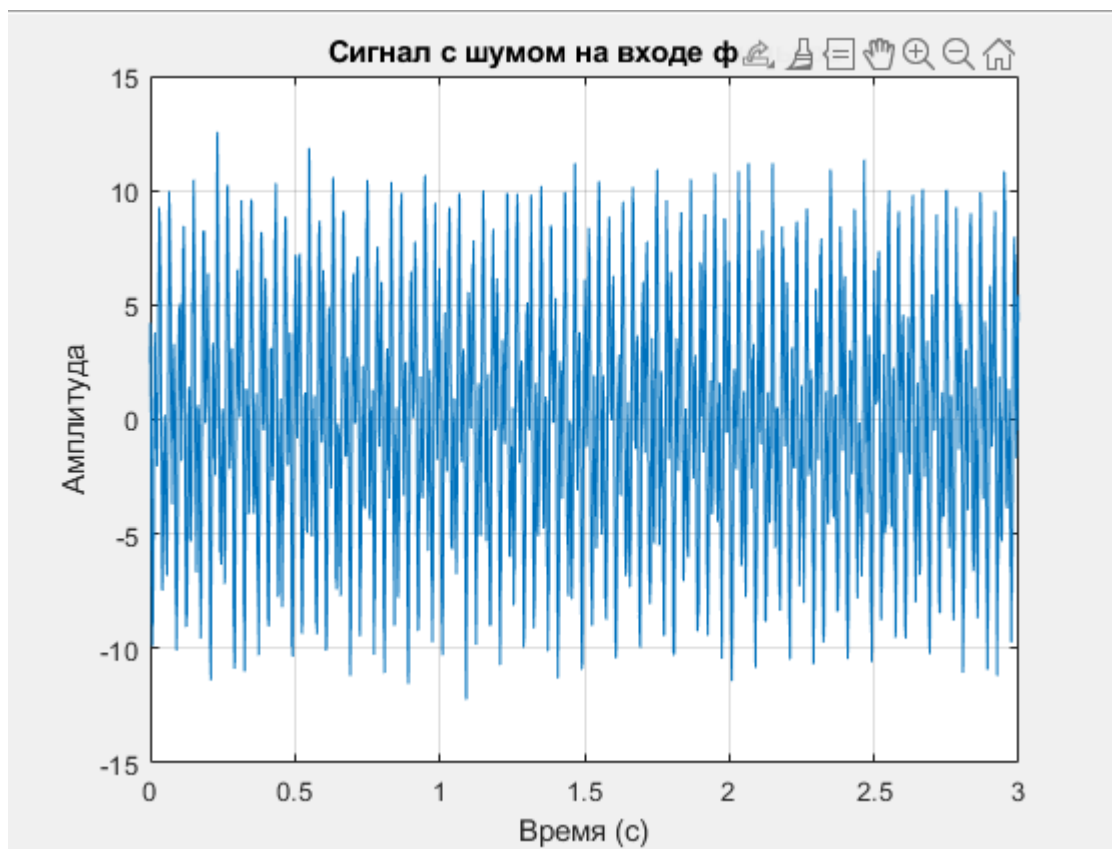


Рисунок 14 – Сигнал с шумом на входе

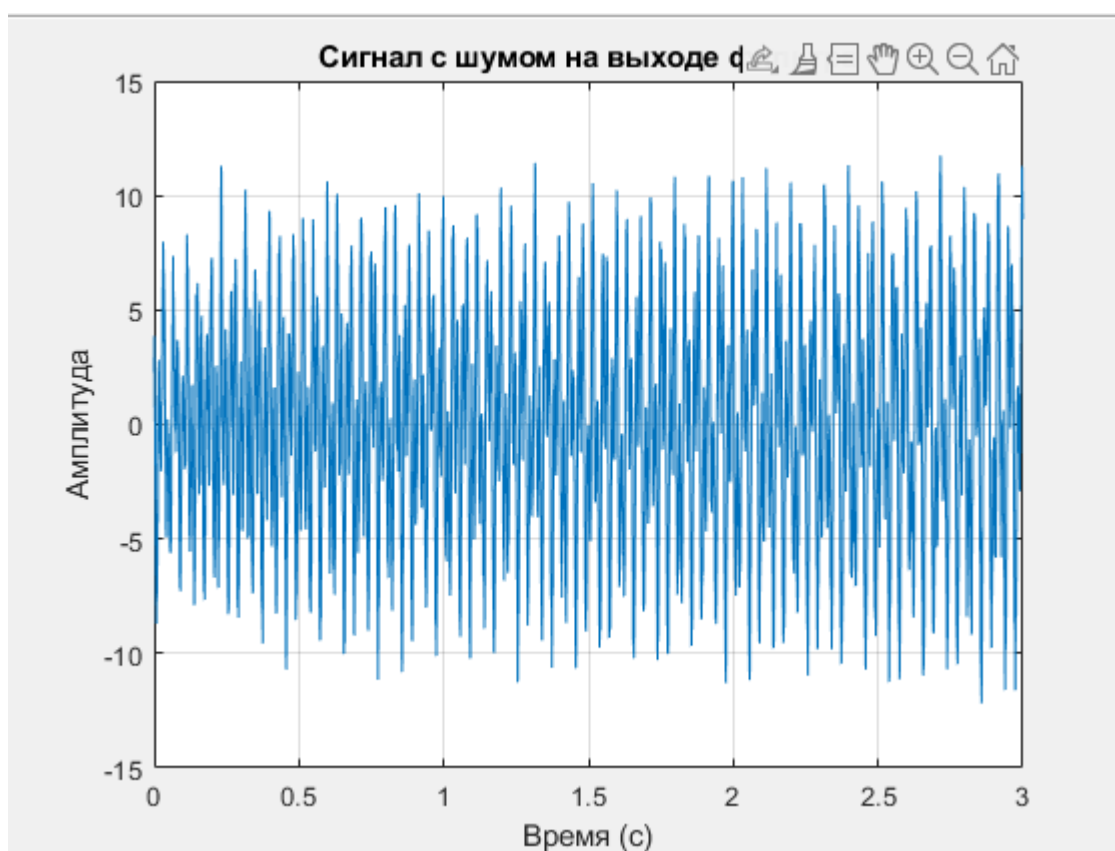


Рисунок 15 – Сигнал с шумом на выходе

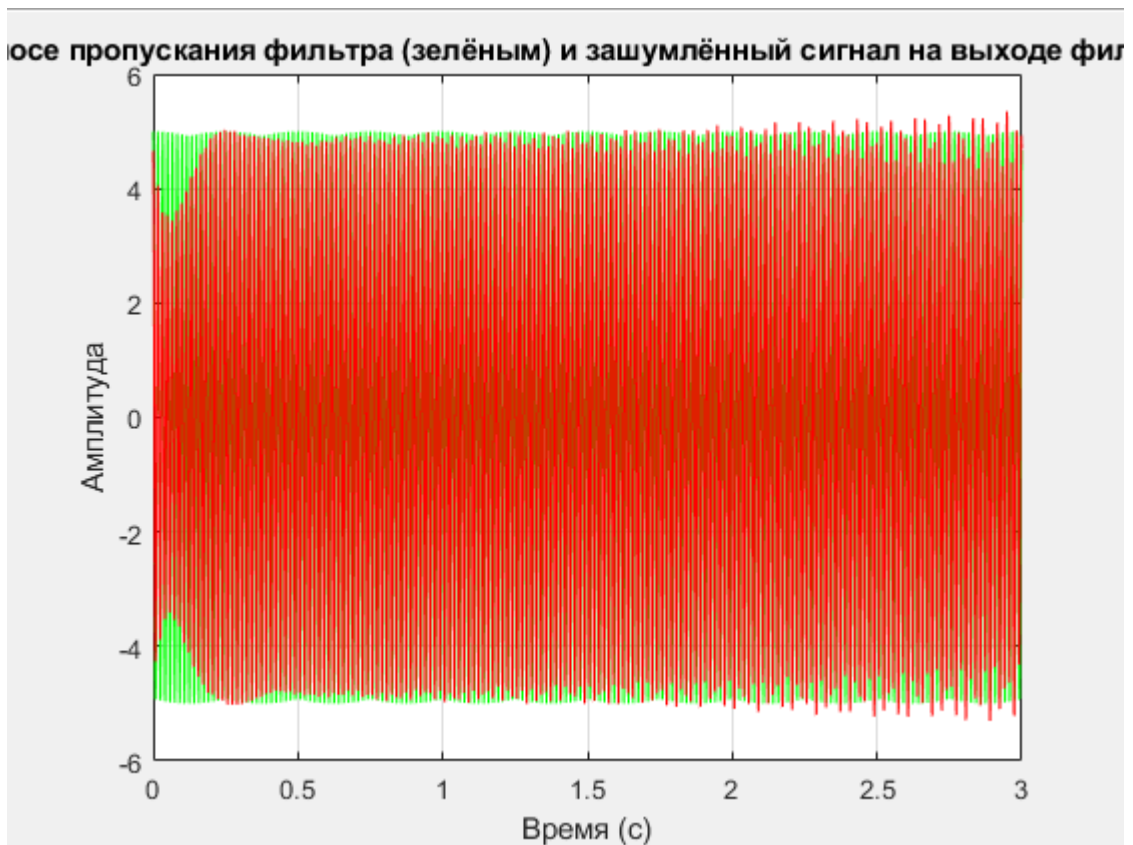


Рисунок 16 – Сигнал в полосе пропускания до и после фильтрации

Для реализации последнего пункта необходимо использовать полученный ФНЧ прототип и аналогично реализовать полосовой фильтр. Разработанный код показан ниже. На рисунках 17 – 19 продемонстрированы полученные графики.

Листинг кода

```
% Частоты
f1_1 = 80;
f2_1 = 150;

% Преобразование в угловые
w1_1 = 2*pi*f1_1;
w2_1 = 2*pi*f2_1;

% Частота среза и ширина полосы пропускания
Wo_1 = sqrt(w1_1*w2_1);
Bw_1 = w2_1 - w1_1;

% Преобразование в полосовой фильтр
[bt_1,at_1] = lp2bp(b,a,Wo_1,Bw_1);
[new_z_1,new_p_1,new_k_1]=tf2zp(bt_1,at_1);
[b_1,a_1] = zp2tf(new_z_1,new_p_1, new_k_1);

% Билинейное Z-преобразование
[z_dig_1,p_dig_1,k_dig_1]=bilinear(new_z_1,new_p_1,new_k_1,Fs);

% Каскадная реализация
[b_dig_1,a_dig_1] = zp2tf(z_dig_1,p_dig_1,k_dig_1);
```

```

% Фильтрация
new_noised_filtered_signal = filter(b_dig_1,a_dig_1,noised_test);

figure()
plot(t,new_noised_filtered_signal)
grid on
title ('Сигнал с шумом на выходе полосового фильтра')

figure()
plot(t,noised_test,'g')
hold on
plot(t,new_noised_filtered_signal,'r')
title({'Сигнал с шумом на входе фильтра(зелёным)', 'Сигнал с шумом на выходе фильтра (красным)'})

figure()
plot(t,new_noised_filtered_signal,'b')
grid on
hold on
plot (t,noised_out,'r')
hold off
title ({'Зашумлённый сигнал на выходе полосового фильтра (синим)', 'и зашумлённый сигнал на выходе фильтра (красным)'})

```

Графики

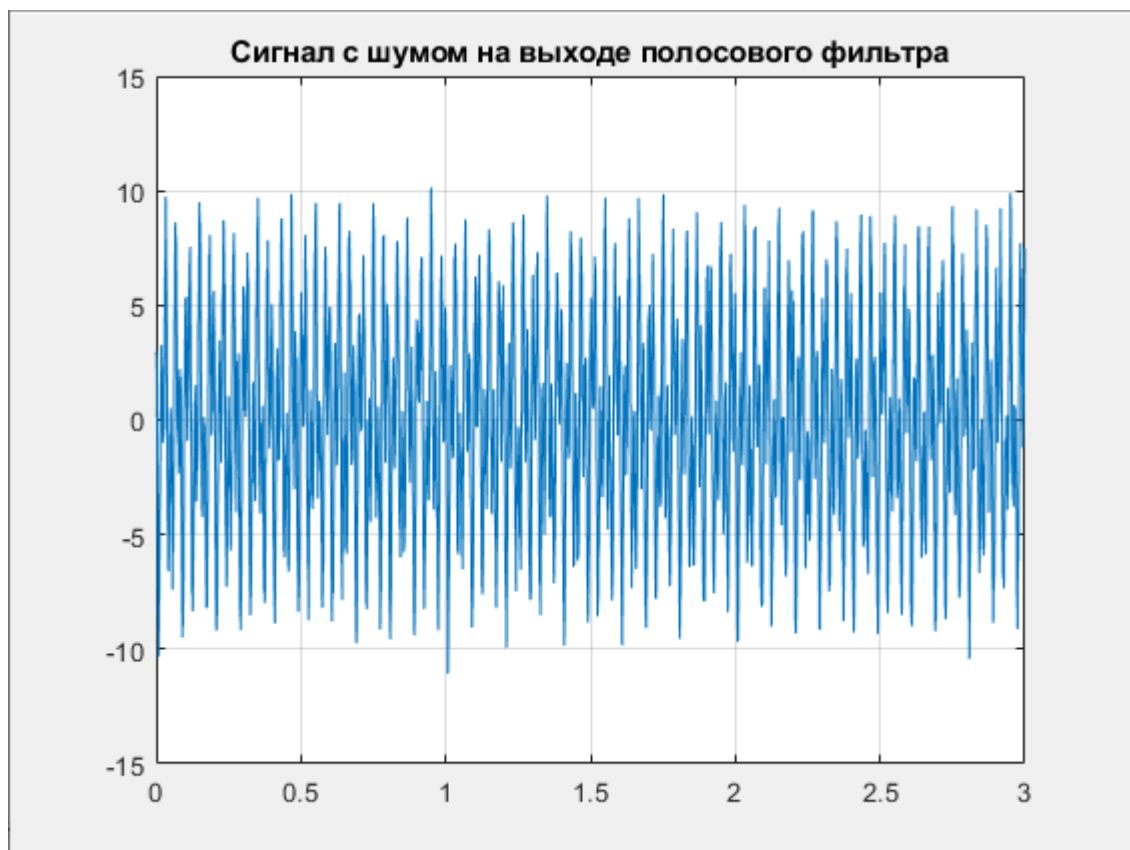


Рисунок 17 – Сигнал с шумом на выходе полосового фильтра

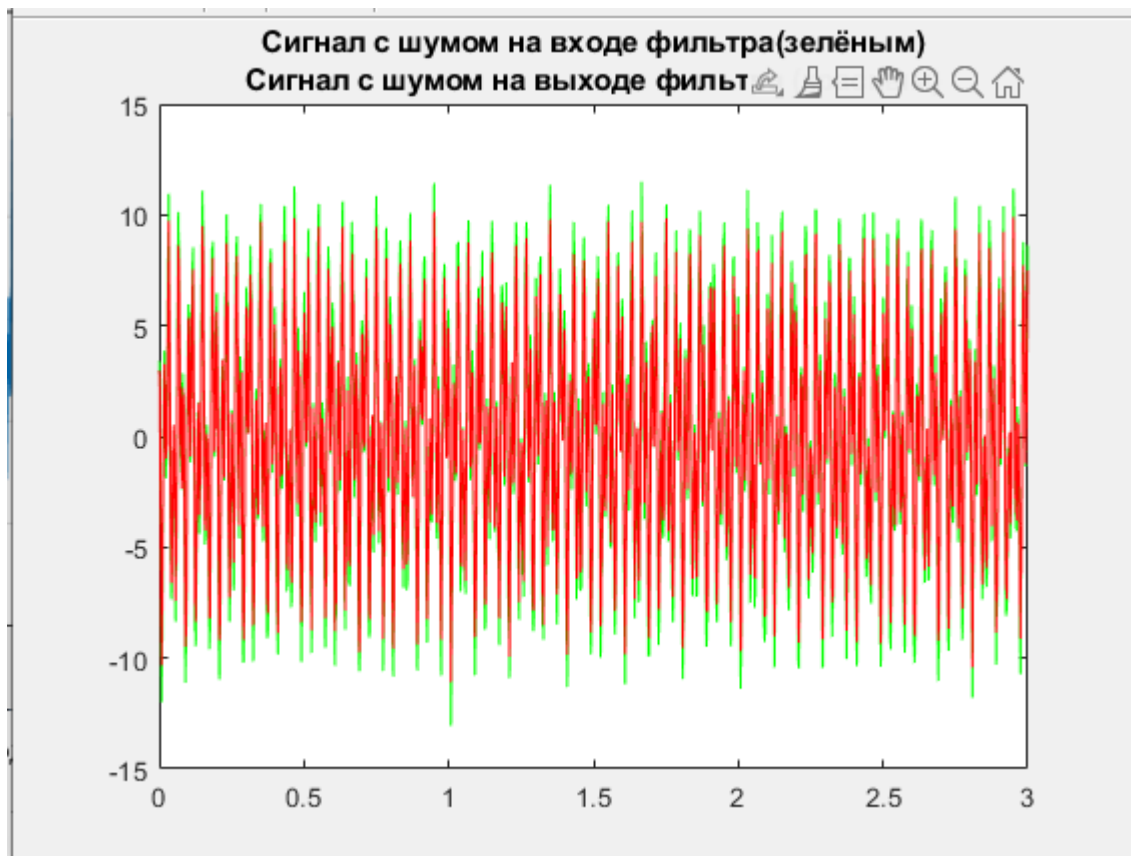


Рисунок 18 – Сигнал с шумом на входе и выходе фильтра

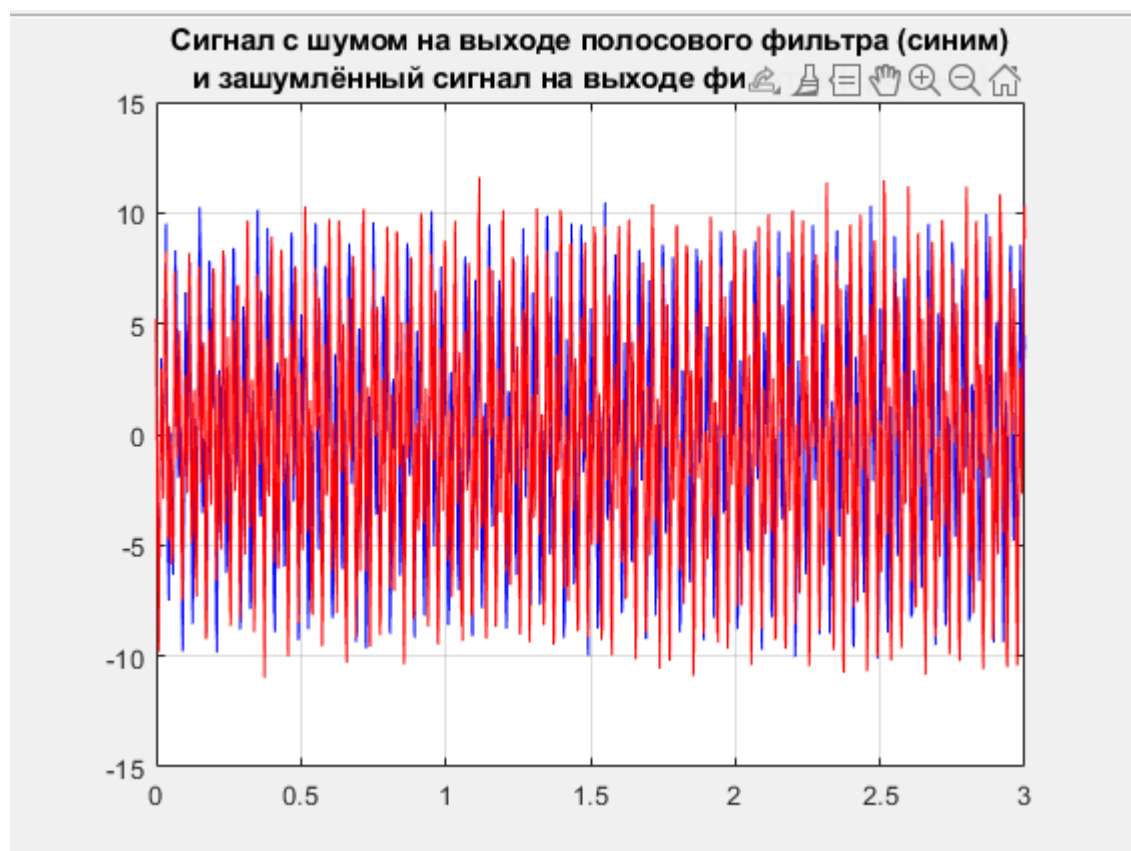


Рисунок 19 – Сигнал с шумом на входе и выходе полосового фильтра

Выводы

В данной лабораторной работе были изучены методы фильтрации сигналов в системах цифровой обработки сигналов с помощью пакета компьютерного моделирования MATLAB.