

Санкт-Петербургский Политехнический Университет
Институт Компьютерных Наук и Технологий
Кафедра Компьютерных Систем и Программных Технологий

Отчёт по лабораторной работе №3
на тему
Линейная фильтрация

Работу выполнил
Студент группы 33501/3
Степанов Е. О.
Преподаватель
Богач Н. В.

Санкт-Петербург
2018

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. Получите их спектры.

3. Теоретическая часть

Фильтрация сигнала (т.е. изменение его спектра), обычно предпринимается с целью увеличить отношение полезного сигнала к шумам и помехам, или же усилить какие-нибудь полезные качества сигнала.

Фильтры можно классифицировать по виду сигналов (аналоговые/цифровые), по виду частотной характеристики (фильтры нижних и верхних частот, полосно пропускающие, полосно-заграждающие и прочие), по виду их импульсной характеристики или по протяжённости импульсной характеристики.

Фильтра низких частот (ФНЧ) – один из видов аналоговых или электронных фильтров, эффективно пропускающий частотный спектр сигнала ниже некоторой частоты (частоты среза), и уменьшающий (подавляющий) частоты сигнала выше этой частоты. Степень подавления каждой частоты зависит от вида фильтра. Один из самых популярных фильтров такого типа - это фильтр Бесселя. Он имеет хорошую фазо-частотную характеристику. Его передаточная функция определяется формулой:

$$G(p) = \frac{n(0)}{n(p)},$$

где $n(p)$ - полином Бесселя, который м.б. найден на основе неравенств:

$$\begin{aligned} B_n(x) &= (2n-1)B_{n-1}(x) + x^2B_{n-2}(x); \\ B_1(x) &= x+1; \quad B_2(x) = x^2+3x+3. \end{aligned}$$

В отличие от фильтра нижних частот (НЧ), фильтр верхних частот пропускает частоты сигнала выше частоты среза, подавляя низкие частоты.

Реализация фильтров нижних частот может быть разнообразной, включая электронные схемы, программные алгоритмы, акустические барьеры, механические системы и т. д.

Полосный фильтр - пропускает составляющие сигнала только в определённой полосе частот.

Частота среза - частота, выше или ниже которой мощность выходного сигнала уменьшается по сравнению с мощностью в полосе пропускания.

4. Ход работы

Строим косинусоидальный сигнал в командном окне Matlab, добавляем в него шум и обрабатываем его:

```
mPi = 1.5;
f0 = 16;
t1=1;
%Строим сигнал.
t=0:pi/64:2*pi;
s = cos(2*mPi*f0*t);
plot(t, s);%Рисуем сигнал ДО зашумления.
%Dобавляем шум.
y = s+awgn(s, 2);plot(t, y);%Рисуем сигнал ПОСЛЕ зашумления.
dots = 1024;
```

```

%Спектры сигнала до и после зашумления.
plot(abs(fft(s, dots)))
plot(abs(fft(y, dots)))
%Значения фильтра.
koef_filter=myfilter;
koef_filter.Numerator
%Строим фильтр.
con=conv(s, koef_filter.Numerator);
t=1:1:length(con);
plot(t, con);
plot(abs(fft(con, length(con))));

```

Фильтр (myfilter.m):

```

function Hd = myfilter
Fs = 100; % Sampling Frequency
Fpass = 2; % Passband Frequency
Fstop = 49; % Stopband Frequency
Dpass = 0.057501127785; % Passband Ripple
Dstop = 0.0001; % Stopband Attenuation
flag = 'scale'; % Sampling Flag
% Calculate the order from the parameters using KAISERORD.
[N, Wn, BETA, TYPE] = kaiserord([Fpass Fstop]/(Fs/2), [10], [Dstop Dpass]);
% Calculate the coefficients using the FIR1 function.
b = fir1(N, Wn, TYPE, kaiser(N+1, BETA), flag);
Hd = dfilt.dfir(b);

```

На рисунке 1 представлен сигнал до и после зашумления, а на рисунке 2 - его спектры.

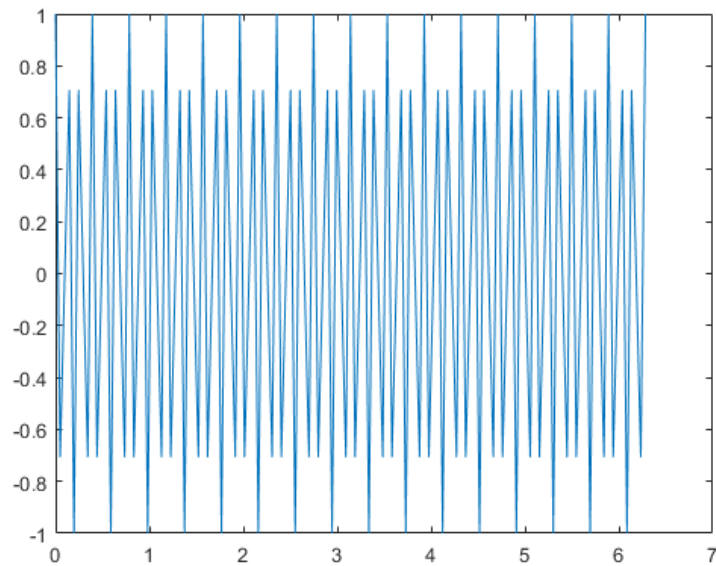


Рис. 1: Сигнал до зашумления.

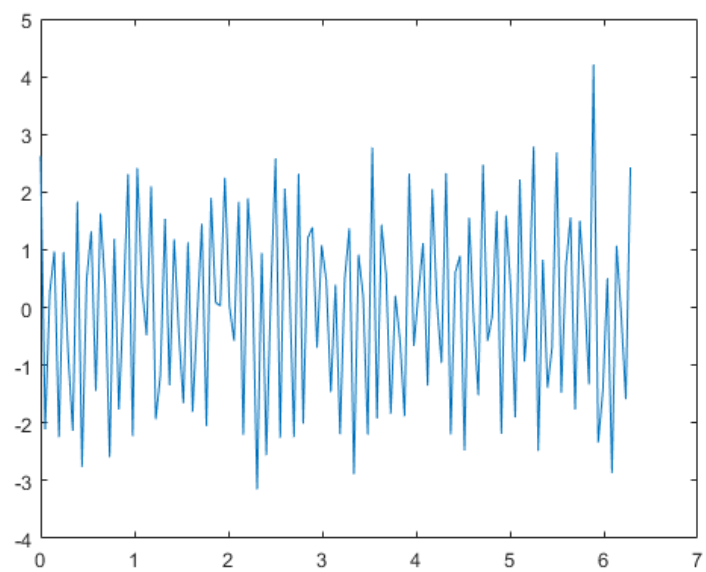


Рис. 2: Сигнал после зашумления.

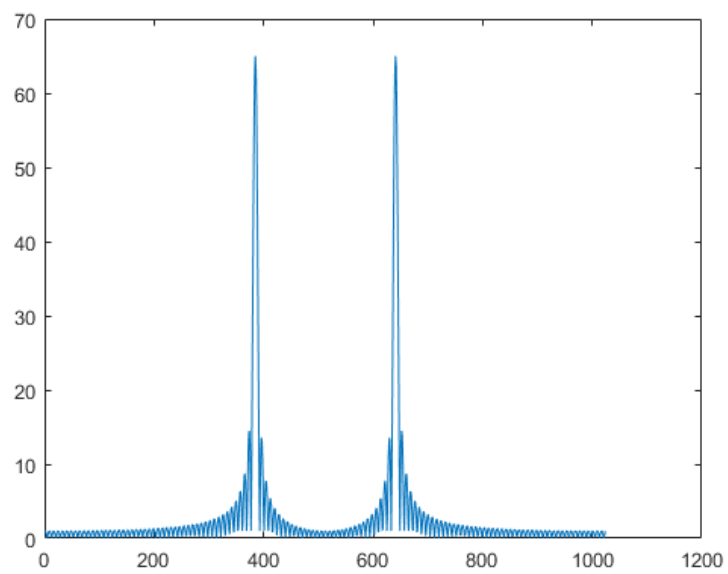


Рис. 3: Спектр сигнала до зашумления.

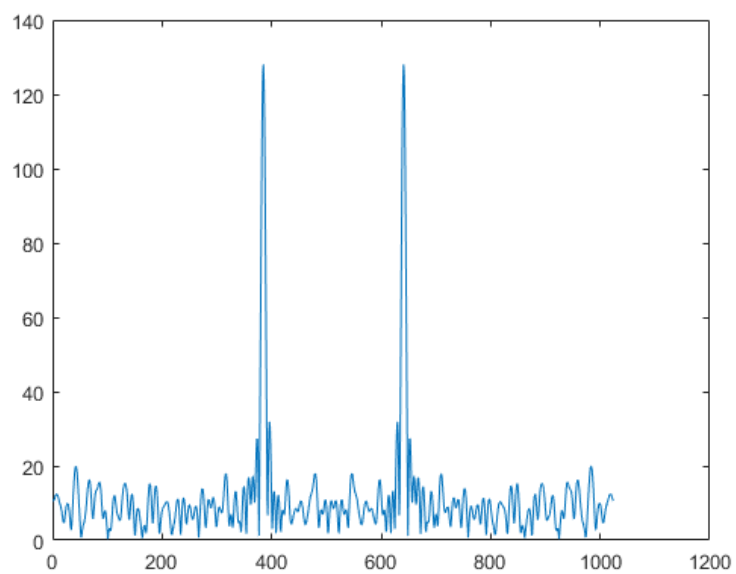


Рис. 4: Спектр сигнала после зашумления.

На рисунке 3 приведён сигнал после фильтрации, а так же его спектр.

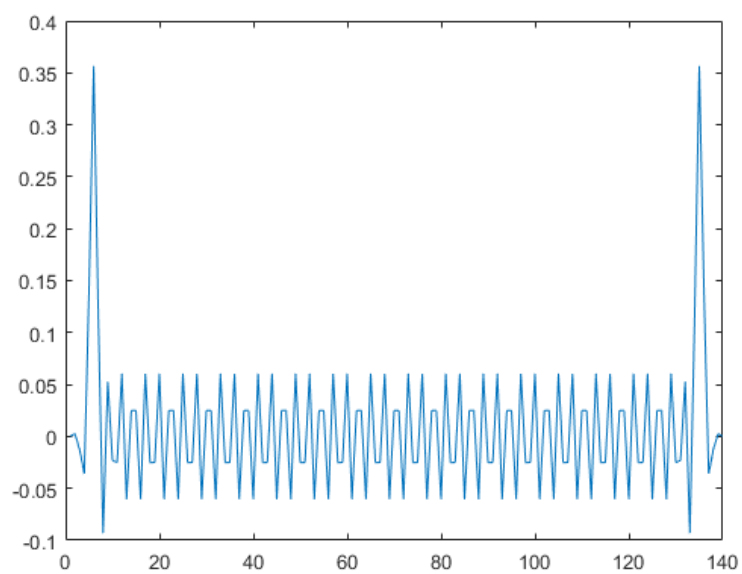


Рис. 5: Сигнал после прохождения через ФНЧ.

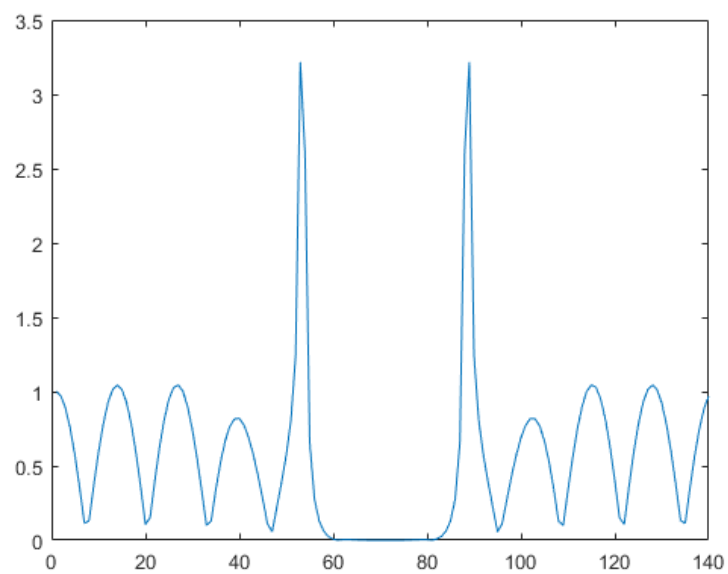


Рис. 6: Спектр после прохождения через ФНЧ.

Выполняем аналогичные действия в среде simulink

На рисунке ниже представлена используемая схема.

На рисунке 5 - сигнал до зашумления, после него и после фильтрации. На рисунке 6 - его спектры (соответственно).

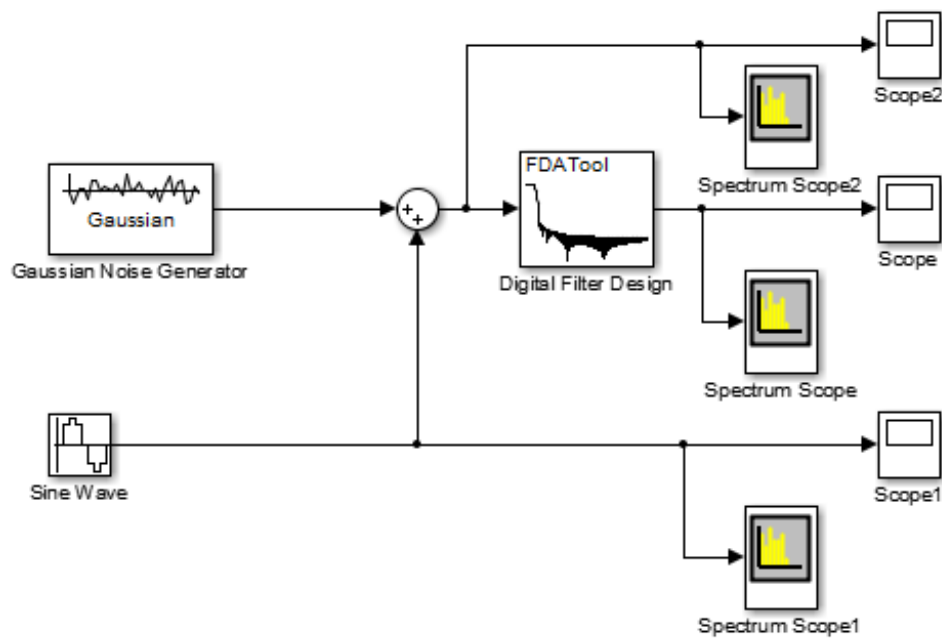


Рис. 7: Схема в simulink.

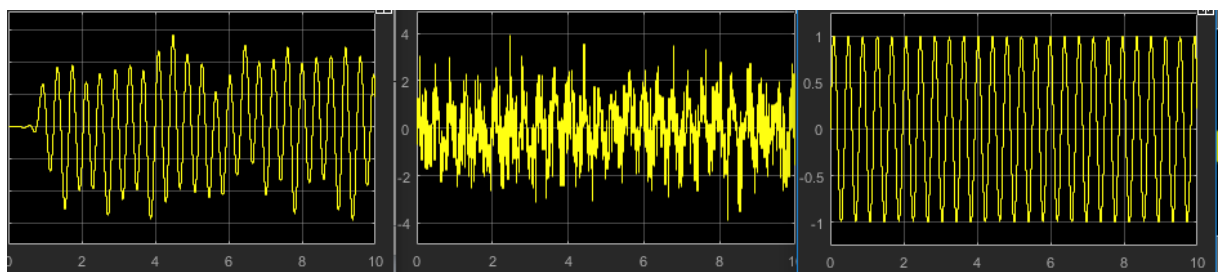


Рис. 8: Сигнал в simulink.

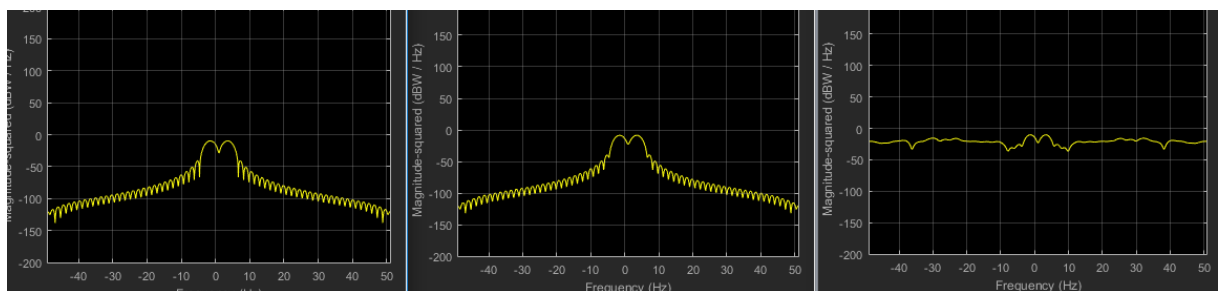


Рис. 9: Спектры сигнала в simulink.

Вывод

Применяя фильтрацию, мы устраняем большую часть помех и тем самым улучшаем качество сигнала. Но как видно по результатам работы, сигнал после фильтрации всё равно заметно отличается от исходного. Это объясняется тем, что часть шума может совпадать с сигналом (как итог - фильтр не убирает этот шум, и полезный сигнал претерпевает изменения).

Есть такое понятие, как идеальный ФНЧ - он полностью подавляет все частоты входного сигнала выше частоты среза и пропускает без изменений все частоты ниже частоты среза. Однако его невозможно реализовать на практике, поэтому все реальные фильтры можно лишь приближать к идеальному.