

Отчет по лабораторным работам по
дисциплине ТСС

Никитина А.В., Замотаева Ю.И.

2014

Содержание

1 Система верстки TeX и расширения LaTeX	4
1.1 Цель работы	4
1.2 Задача работы	4
1.3 Написание математических формул	4
1.4 Интерпретация результатов	4
1.5 Вывод	4
2 Визуализация сигналов во временной и частотной области	5
2.1 Цель работы	5
2.2 Постановка задачи	5
2.3 Справочные материалы	5
2.4 Теоретический материал	5
2.5 Ход работы	5
2.5.1 Код MATLAB для исходного сигнала	6
2.5.2 Результаты работы программы	6
2.5.3 Код MATLAB для зашумленного сигнала	7
2.5.4 Результаты работы программы	7
2.6 Вывод	9
3 Спектры простых сигналов	9
3.1 Цель работы	9
3.2 Постановка задачи	9
3.3 Справочные материалы	9
3.4 Теория	10
3.5 Полигармонический сигнал	10
3.5.1 Код Matlab	10
3.5.2 Модуляция в среде Simulink	14
3.6 Прямоугольный импульсный сигнал	15
3.6.1 Код Matlab	15
3.6.2 Модуляция в среде Simulink	17
3.7 Треугольный импульсный сигнал	18
3.7.1 Код Matlab	18
3.7.2 Модуляция в среде Simulink	20
3.8 Вывод	21
4 Линейная фильтрация	21
4.1 Цель работы	21
4.2 Теоретический материал	21
4.3 Ход работы	22
4.4 Моделирование в Simulink	25
4.5 Вывод	27
5 Аналоговая модуляция	27
5.1 Цель работы	27
5.2 Алгоритм работы	27
5.3 Теоретические сведения	28
5.4 Код MATLAB для п.1-3	28
5.5 Код MATLAB для п.4	32

5.6	Код MATLAB для п.5	34
5.7	Код MATLAB для п.6	36
5.8	Вывод	36
6	Частотная и фазовая модуляция	36
6.1	Цель работы	36
6.2	Постановка задачи	37
6.3	Справочные материалы	37
6.4	Теоретический материал	37
6.5	Ход работы	37
6.6	Выводы	43
7	Цифровая модуляция	43
7.1	Цель работы	43
7.2	Алгоритм работы	43
7.3	Теоретическая часть	43
7.4	Код MATLAB	45
7.4.1	BPSK modulation	45
7.4.2	PSK modulation	46
7.4.3	OQPSK modulation	47
7.4.4	GENQAM modulation	48
7.4.5	MSK modulation	49
7.5	Работа в среде Simulink	50
7.5.1	BPSK modulation	50
7.5.2	DBPSK modulation	51
7.5.3	M-FSK modulation	52
7.6	Вывод	53

1 Система верстки \TeX и расширения \LaTeX

1.1 Цель работы

Ознакомление с программой для верстки отчетов \TeX .

1.2 Задача работы

Научиться создавать титульный лист, разделы, списки и писать несложные формулы в среде верстки отчетов \TeX .

1.3 Написание математических формул

1.

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{25}{2 * k^3} = \frac{\pi}{2}$$

2.

$$x^3 + \sum_{x=1}^8 \frac{x}{2} = \sqrt{617}$$

3.

$$x(t) = A * \sin(w * t + f)$$

Формула интеграла:

$$\iint_{\frac{\sin(x)}{x} + y^2 = 1} f(x, y) dx dy$$

1.4 Интерпретация результатов

Среда верстки отчетов \TeX очень удобна для введения математических формул. Имеется множество команды, которые заметно облегчают ввод дробных выражений, констант, переменных, производных, интегралов, дифференциалов и других формул. Таким образом, можно без особы трудностей вводить формулы различной сложности.

1.5 Вывод

В данной лабораторной работе мы познакомились с системой создания и редактирования текстов \TeX и расширением \LaTeX . Рассмотрели команды этой среды, которые помогают составлять красивые и аккуратные отчеты. Таким образом, теперь мы без труда может написать отчет, который будет иметь титульный лист, главы, заголовки и различные математические формулы. Нам очень понравилась эта система, потому что она помогает каждому составить отчет быстро и удобно. Система полностью автоматизирована, от нас требуется только освоить основные команды и ознакомиться с дополнительной литературой.

2 Визуализация сигналов во временной и частотной области

2.1 Цель работы

Познакомиться со средствами генерации сигналов и визуализации их спектров.

2.2 Постановка задачи

В командном окне MATLAB и в среде Simulink промоделировать чистый синусоидальный сигнал, так же синусоидальный сигнал с шумом. Получить их спектры.

2.3 Справочные материалы

В.С. Гутников Фильтрация измерительных сигналов пп.1-2, 11-12

2.4 Теоретический материал

Ряд Фурье для периодических сигналов:

$s(t) = \sum_{k=-\infty}^{\infty} C_k e^{j2\pi k f_1 t}$, где $f_1 = \frac{1}{T_1}$; T_1 – период функции $s(t)$; C_k – постоянные коэффициенты.

На основе ряда Фурье можно вывести преобразование для непериодических сигналов, представив период равным ∞ :

$s(t) = \int_{-\infty}^{\infty} (\int_{-\infty}^{\infty} s(t) e^{-j2\pi f t} dt e^{j2\pi f t} df)$ Это соотношение называется преобразованием Фурье. Оно объединяет в себе прямое ПФ:

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi f t} dt$$

и обратное ПФ:

$$s(t) = \int_{-\infty}^{\infty} S(f) e^{j2\pi f t} df$$

Рассмотрим следующее свойство преобразования Фурье: умножение сигналов.

ПФ произведения двух функций равно свертке их ПФ:

$$\phi(t)\psi(t) = \Phi(f) * \Psi(f)$$

2.5 Ход работы

В среде MATLAB вначале моделируем чистый синусоидальный сигнал по следующей формуле:

$$y(t) = \sin(2\pi f t)$$

Затем моделируем зашумленный сигнал по следующей формуле:

$$y(t) = \sin(2\pi f t) + Arand()$$

Для выделения частот регулярных составляющих используем преобразование Фурье, которое реализовано встроенной в MATLAB функцией.

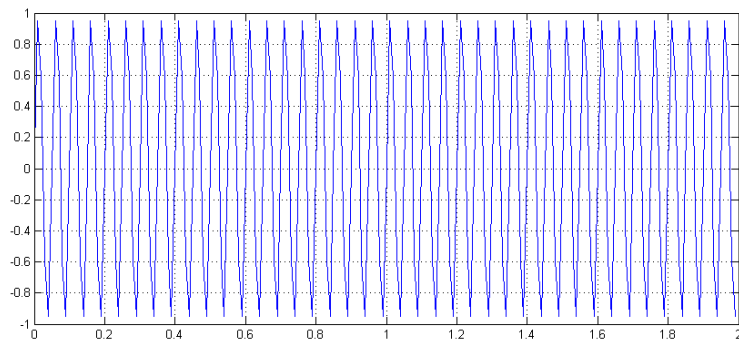
2.5.1 Код MATLAB для исходного сигнала

```
x=0:0.01:4*pi;
f0 = 20;
%исходный сигнал
y = sin(2*pi*f0*x);
figure(1)
plot(x(1:200),y(1:200))
grid
%спектр исходного сигнала
figure(2)
spectrum = fft(y,512);
normspectrum = spectrum.*conj(spectrum)/512;
f=100*(0:255)/512;
plot(f, normspectrum(1:256))
axis([0 max(F) 0 10])
grid
```

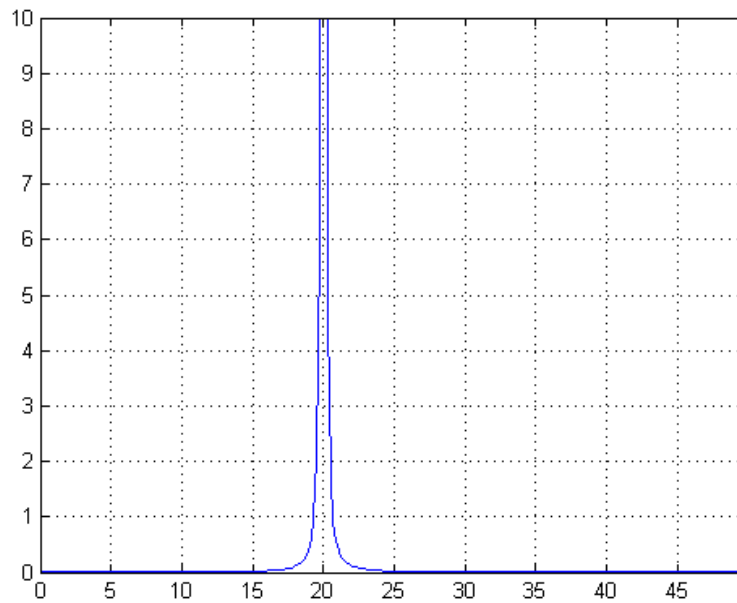
2.5.2 Результаты работы программы

В результате выполнения программы получились графики временной и частотной характеристик исходного синусоидального сигнала.

- Исходный сигнал



- Спектр исходного сигнала



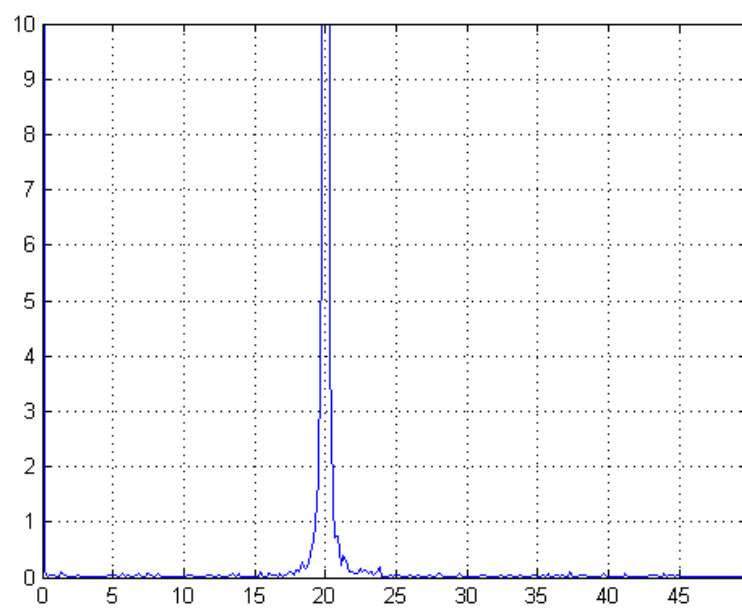
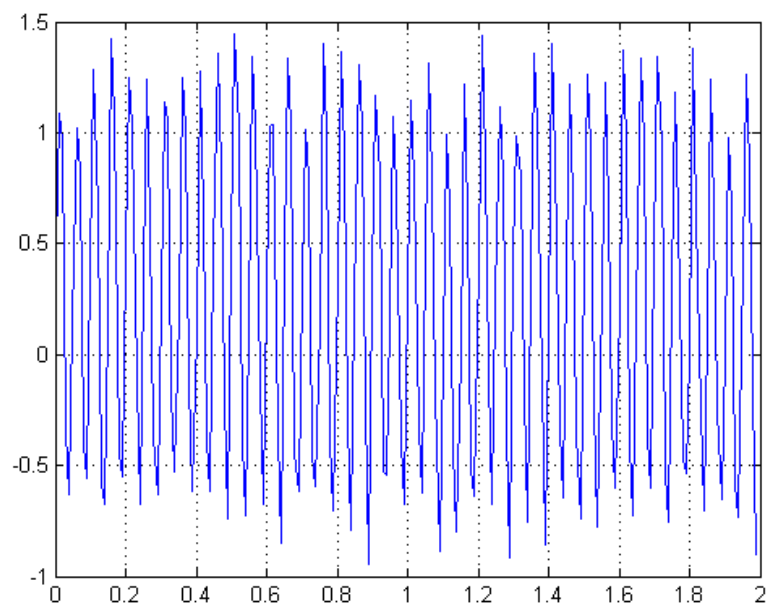
2.5.3 Код MATLAB для зашумленного сигнала

```
%зашумленный сигнал
ynoise = y+ 0.5*rand(size(x));
figure(3)
plot(x(1:200),ynoise(1:200));
grid
%спектр зашумленного сигнала
spectrum = fft(ynoise,512);
noisespectrum = spectrum.*conj(spectrum)/512;
figure(4)
plot(f, noisespectrum())
axis([0 max(f) 0 10])
grid
```

2.5.4 Результаты работы программы

В результате выполнения программы получились графики временной и частотной характеристик зашумленного синусоидальных сигнала.

- Зашумленный сигнал
- Спектр зашумленного сигнала



2.6 Вывод

В данной работе мы построили исходный синусоидальный сигнал, после чего получили его спектр. Далее мы построили зашумленный сигнал и его спектр. По рисункам видно, что спектр зашумленного сигнала очень напоминает спектр исходного сигнала, но является более неровным. Стоит пояснить возможные расхождения между теоретически ожидаемым и полученным спектром сигнала вследствие дискретности представления сигналов, а также их конечности во времени: конечность сигнала приводит к тому, что спектр сигнала представляет собой $\text{sinc}(x)$. Дискретизация сигнала приводит к тому, что в частотной области сигнал повторяется. Разъясним подробнее. В соответствии с теорией мы должны были получить спектр синусоидального сигнала в виде очень короткого импульса бесконечно большой длины. Но на практике на входе мы имеем сигнал ограниченной длины (наш бесконечный сигнал был умножен на прямоугольное окно), и поэтому его спектр сворачивается со спектром прямоугольного окна. Так как наш исходный сигнал был дискретизирован, он умножается на решетку дельта-импульсов. Спектром решетки дельта-импульсов является решетка дельта-импульсов. Поэтому спектр нашего сигнала сворачивается с решеткой дельта-импульсов. Так мы получаем спектр в виде нескольких импульсов.

3 Спектры простых сигналов

3.1 Цель работы

Получить представление о тестовых сигналах во временной и частотной областях. Реализовать операцию свертки сигналов.

3.2 Постановка задачи

В командном окне MATLAB и в среде Simulink промоделировать следующие тестовые сигналы:

- полигармонический сигнал $y(t) = \sum_{n=0}^{N-1} \cos(nt)$
- прямоугольный импульсный сигнал $y(t) = \Pi(t, T_i)$
- треугольный импульсный сигнал $y(t) = \Delta(t, T_i)$

Получить их спектры.

3.3 Справочные материалы

В.С. Гутников. Фильтрация измерительных сигналов пп.3-6, 13-14

3.4 Теория

Спектр - это функция, показывающая зависимость интенсивности различных гармоник в составе сигнала от частоты этих гармоник. Спектр периодического сигнала - это дискретный спектр. Известно, что спектр произведения двух сигналов равен свёртке спектров этих сигналов, а спектр свёртки сигнала равен произведению их спектров.

Прямоугольный импульс с единичной амплитудой мы будем обозначать символом $\Pi(t, T_i)$. Здесь первый аргумент обозначает положение импульса на горизонтальной оси: середина импульса соответствует нулю этого аргумента. Второй аргумент - ширину импульса.

Спектр прямоугольного импульса представляется простым выражением:

$$\Phi(f) = \int_{-\infty}^{\infty} \Pi(t, T_i) e^{-2\pi f t} dt = \int_{-\frac{T_i}{2}}^{\frac{T_i}{2}} e^{-2\pi f t} dt = \frac{\sin(\pi f T_i)}{\pi f}$$

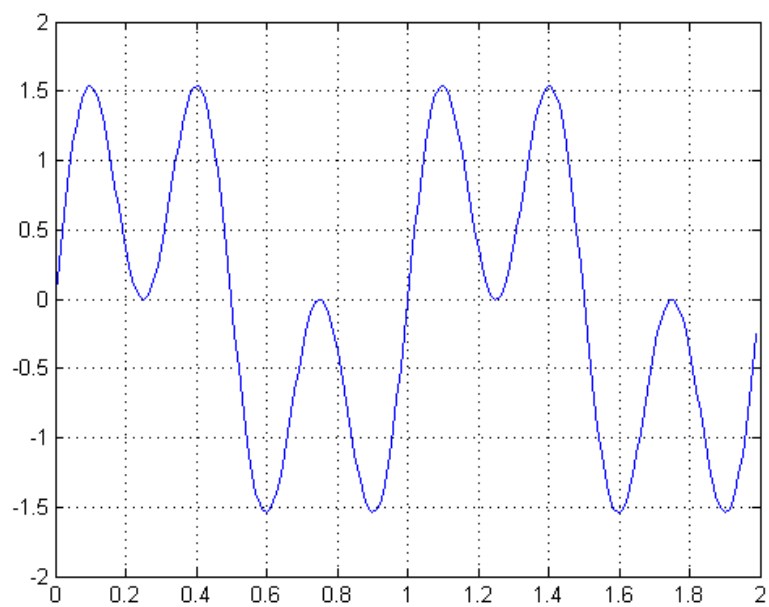
Треугольный симметричный импульс, середина которого соответствует $t=0$, длительность равна T_i , а амплитуда - единице, мы будем обозначать символом $\Delta(t, T_i)$. Такой импульс можно представить в виде свёртки двух одинаковых прямоугольных импульсов длительностью $T_i/2$:

$$\Delta(t, T_i) = \frac{2}{T_i} \Pi(t, \frac{T_i}{2}) * \Pi(t, \frac{T_i}{2}).$$

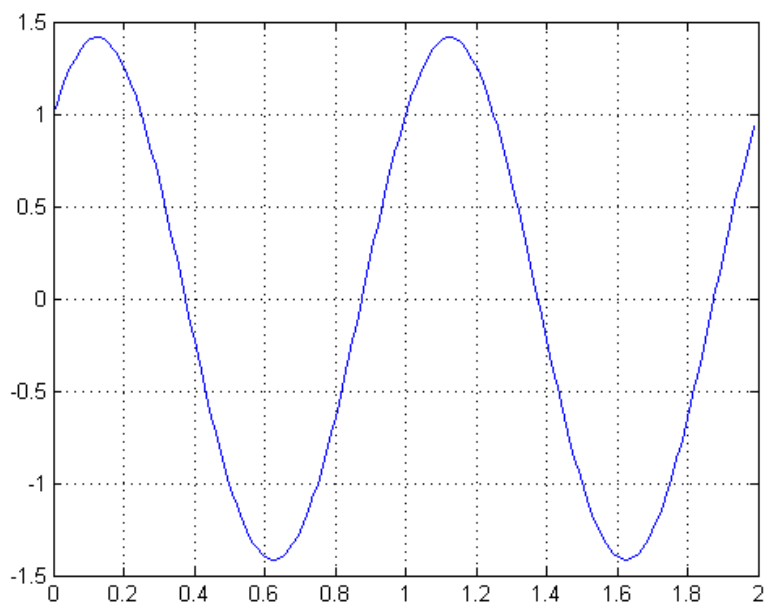
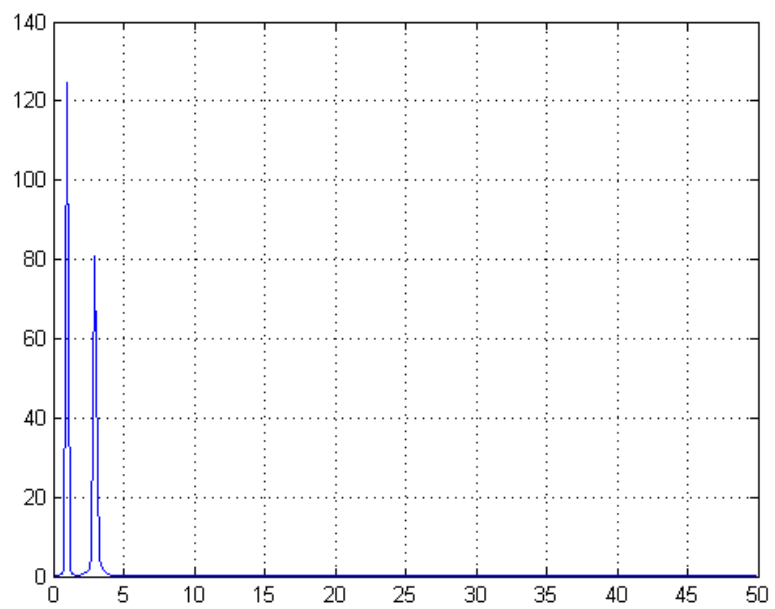
3.5 Полигормонический сигнал

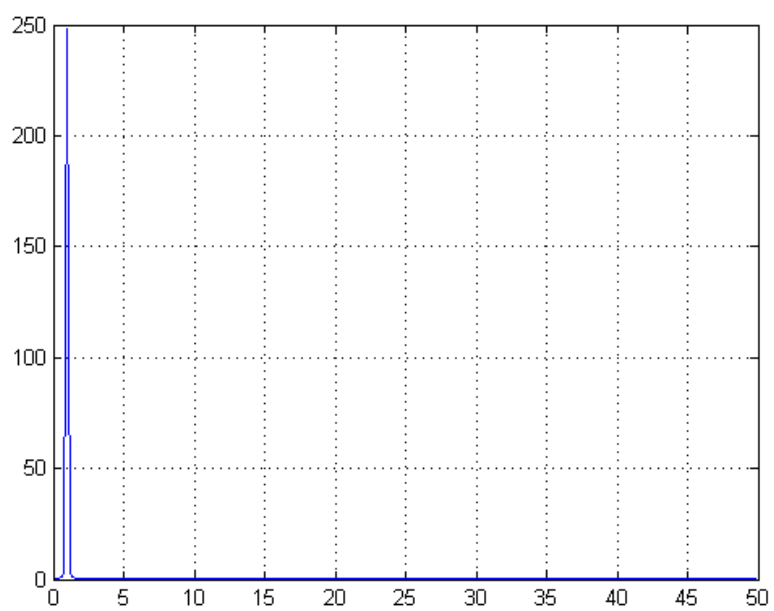
3.5.1 Код Matlab

```
x = 0:0.01:4*pi;
f=100*(0:255)/512;
figure
y1 = sin(2*pi*x)+sin(2*pi*3*x);
plot(x(1:200),y1(1:200))
grid
figure
s1 = fft(y1,512);
ss1 = s1.*conj(s1)/512;
plot(f,ss1(1:256))
grid
figure
y2 = sin(2*pi*x)+cos(2*pi*x);
plot(x(1:200),y2(1:200))
grid
figure
s2 = fft(y2,512);
ss2 = s2.*conj(s2)/512;
plot(f,ss2(1:256))
grid
```



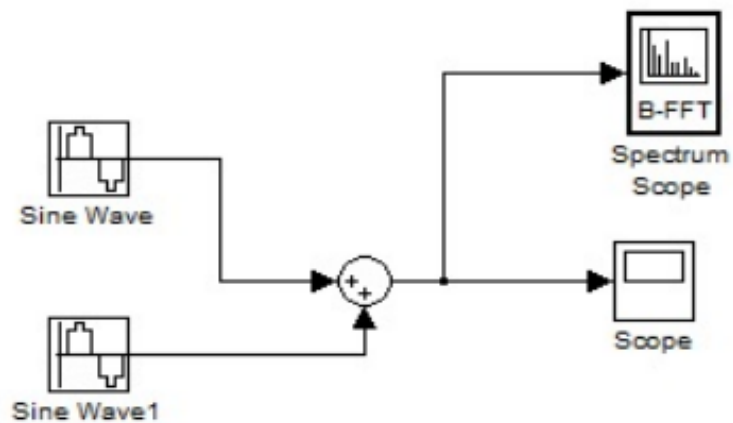
- полигармонический сигнал $\sin(x) + \sin(3x)$
- спектр данного полигармонического сигнала
- полигармонический сигнал $\sin(x) + \cos(x)$
- спектр данного полигармонического сигнала



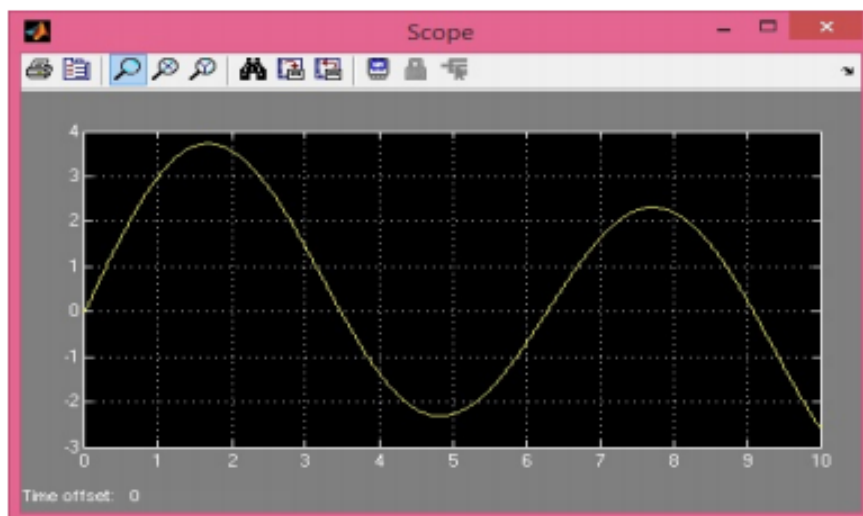


3.5.2 Модуляция в среде Simulink

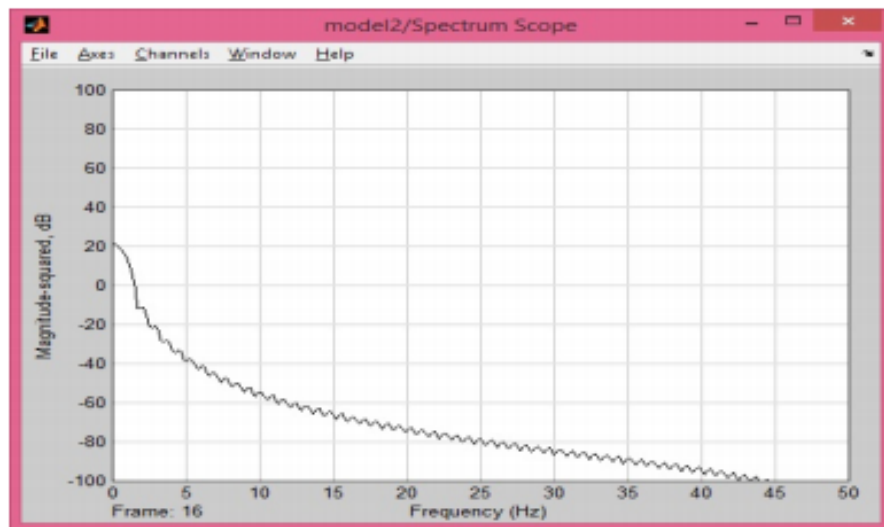
- модель



- сигнал



- спектр

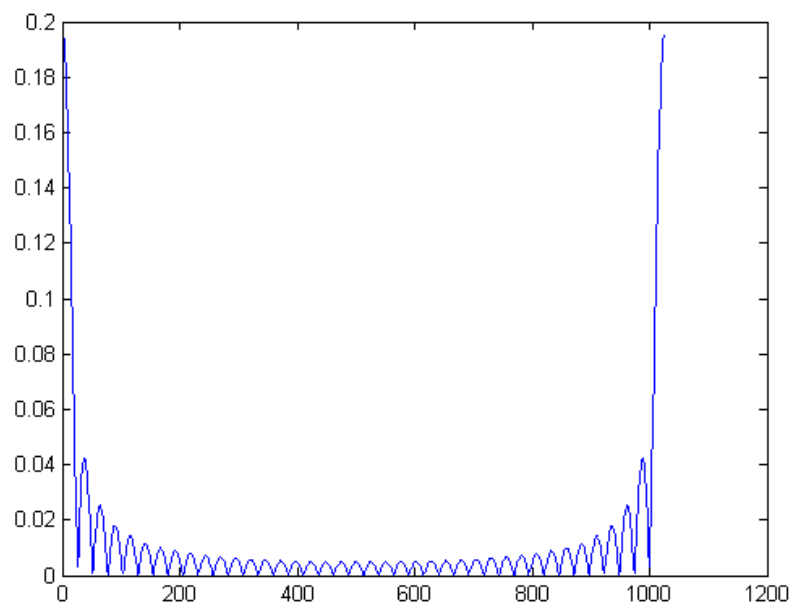
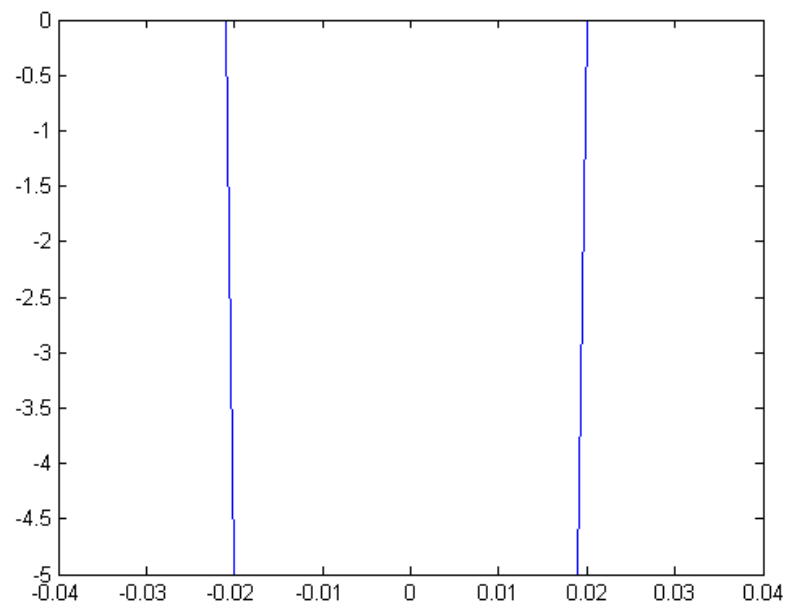


3.6 Прямоугольный импульсный сигнал

3.6.1 Код Matlab

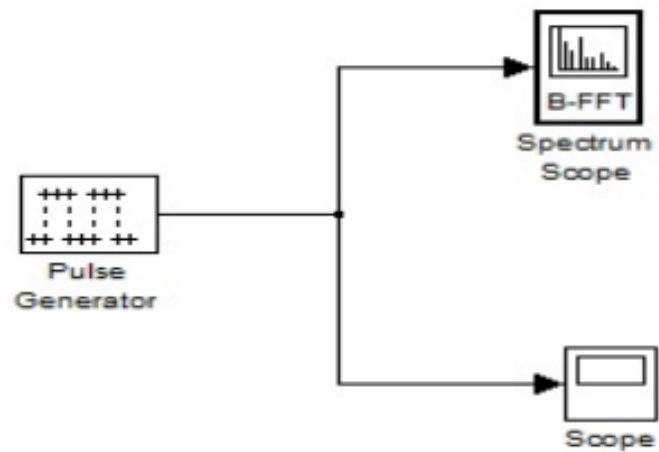
```
t=-0.04:1/1000:0.04;
y4=-5*rectpuls(t,0.04);
plot(t,y4);
figure;
spectrum = abs(fft(y4,1024))/1024;
plot(spectrum);
```

- Прямоугольный импульсный сигнал
- Спектр прямоугольного сигнала

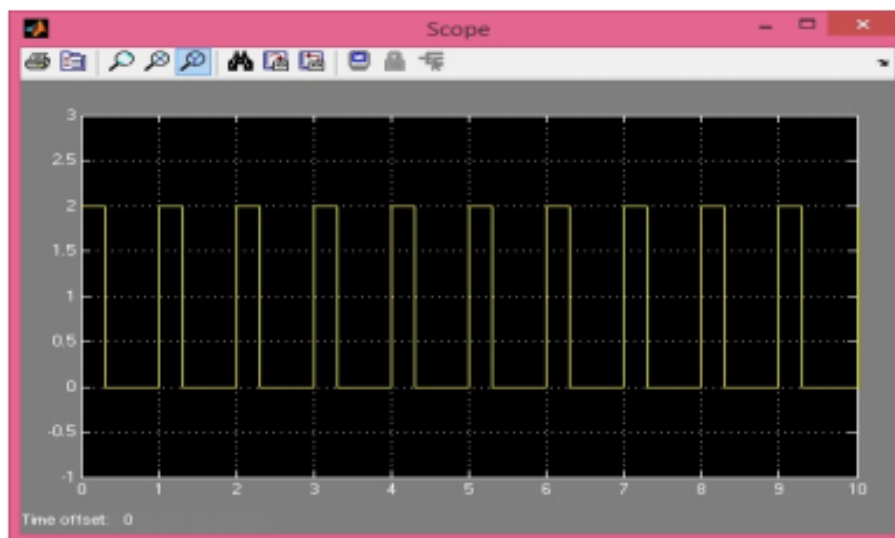


3.6.2 Модуляция в среде Simulink

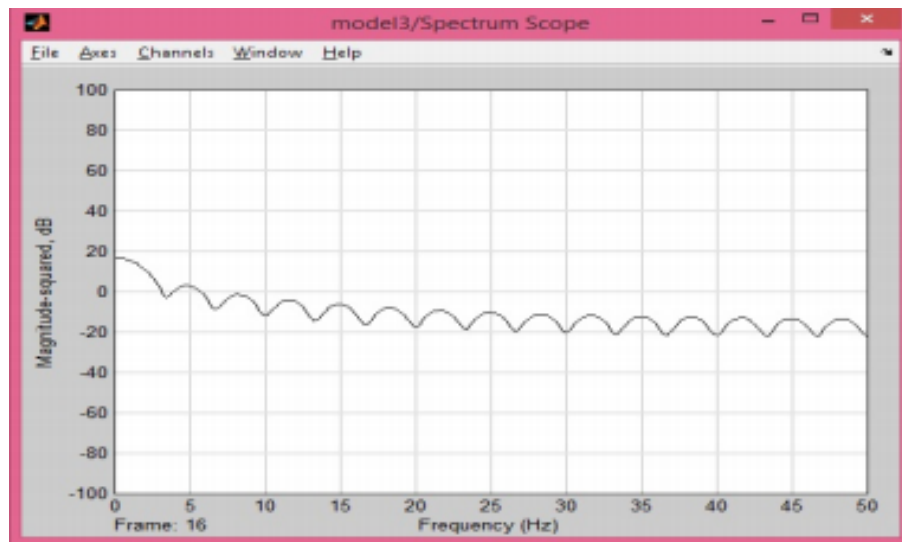
- модель



- СИГНАЛ



- спектр

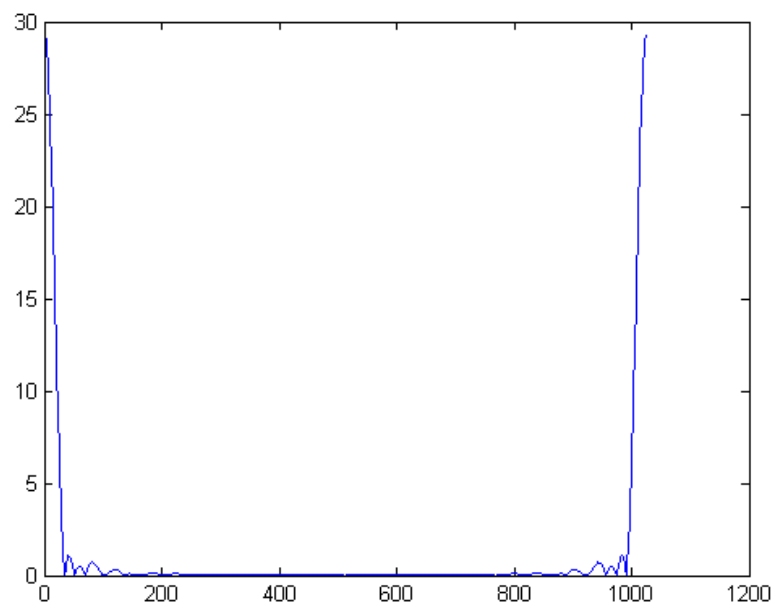
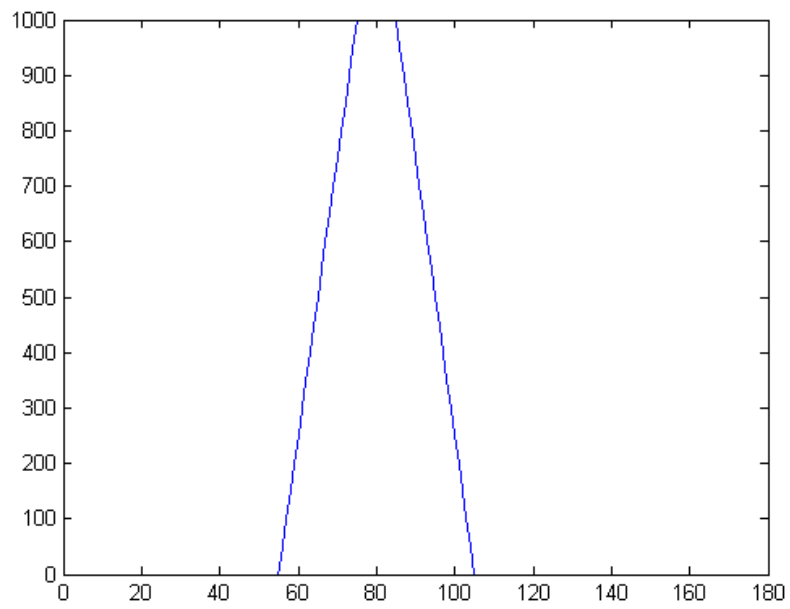


3.7 Треугольный импульсный сигнал

3.7.1 Код Matlab

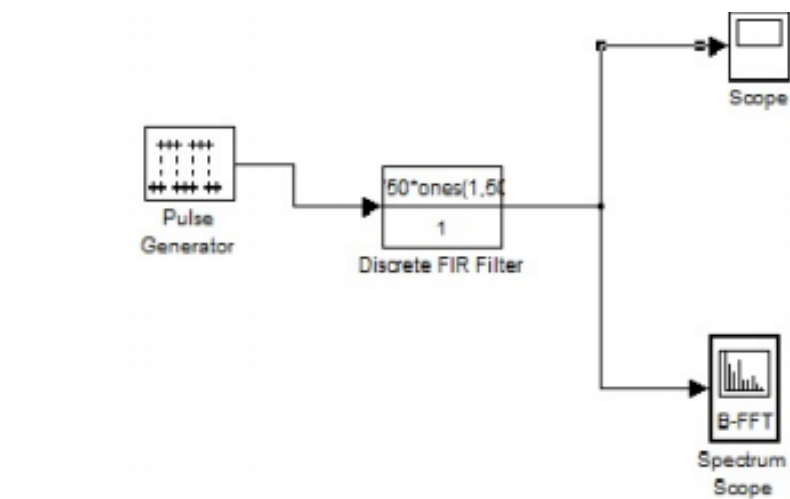
```
t=-0.04:1/1000:0.04;
y4=-5*rectpuls(t,0.02);
y5=-10*rectpuls(t,0.03);
y6= conv(y4,y5);
plot(y6);
figure;
spectrum = abs(fft(y6,1024))/1024;
plot(spectrum);
```

- Треугольный импульсный сигнал
- Спектр треугольного сигнала

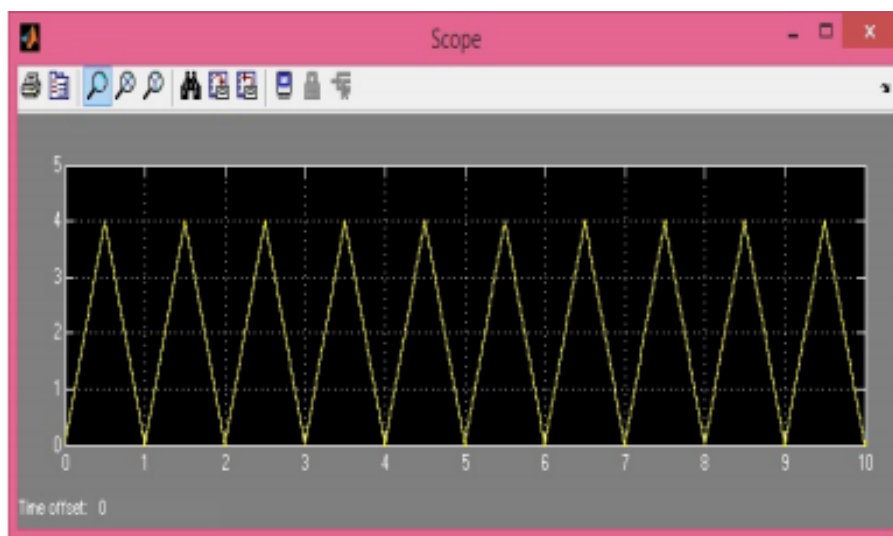


3.7.2 Модуляция в среде Simulink

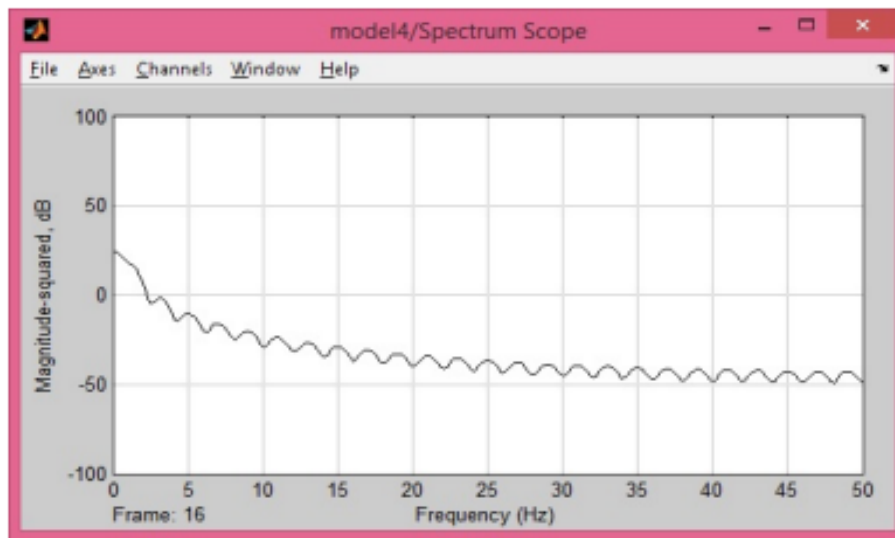
- модель



- сигнал



- спектр



3.8 Вывод

В лабораторной работе было проведено моделирование в среде MatLAB и Simulink полигармонического, прямоугольного и треугольного сигналов. После чего получены их спектры. Для получения сигналов использовались математические формулы данных функций. Треугольный сигнал был получен путем применения операции свертки для двух прямоугольных сигналов. Данная операция осуществляется с помощью специальной функции Matlab. В Simulink для получения генератора треугольного сигнала используется генератор прямоугольных импульсов каскадно с фильтром с прямоугольным окном.

Обоснованием получения треугольного сигнала из свертки двух прямоугольных служит тот факт, что интеграл от произведения двух констант есть линейная функция. График такого преобразования будет представлять две линейные функции, одна из которых имеет положительный коэффициент наклона, а другая отрицательный.

4 Линейная фильтрация

4.1 Цель работы

Изучить воздействия ФНЧ на тестовый сигнал с шумом. Сгенерировать гармонический сигнал с шумом и синтезировать ФНЧ. Получить сигнал во временной и частотной областях до и после фильтрации.

4.2 Теоретический материал

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, т.е.

это устройство целенаправленным образом изменяющее спектр сигнала. Фильтры бывают следующих видов: фильтры нижних частот(ФНЧ) - пропускают низкочастотные составляющие спектра и задерживают высокочастотные; фильтры верхних частот(ФВЧ) - пропускают только высокочастотные составляющие, фильтры полосно пропускающие(ФПП) - пропускают составляющие сигнала в определенной полосе частот и фильтры полосно-заграждающие(ФПЗ) - пропускают все составляющие спектра кроме заданной полосы.

Фильтры с конечной импульсной характеристикой (КИХ-фильтры) - их импульсная характеристика ограничена во времени. Фильтры с бесконечной импульсной характеристикой (БИХ-фильтры) - их импульсная характеристика, хоть и затухает во времени, теоретически неограничена во времени.

4.3 Ход работы

В данной работе в качестве ФНЧ используем фильтр Баттерворта, который синтезируем с помощью функции Matlab - BUTTER. Фильтр Баттерворта является БИХ фильтром. Порядок фильтра равен 16.

В командном окне Matlab сгенерируем гармонический сигнал без шума и с шумом, а также спектры сигнала с шумом и без шума. В результате код Matlab будет выглядеть так:

```
x = 0:0.01:4*pi;
f=100*(0:255)/512;
figure
noise=rand(size(x));
y = sin(2*pi*x);
y_noisy = y+0.3*noise;

%Построение сигнала без шума:
plot(x(1:200),y(1:200))
grid

%Построение сигнала с шумом:
plot(x(1:200),y_noisy(1:200))
grid

%синтез ФНЧ Баттерворта
[B,A] = butter(16,0.99);
B=B./sum(B);
A=A./sum(A);
%обработка сигнала ФНЧ
figure
y_filtered = conv(y_noisy,[B,A]);

%Построение сигнала после прохождения через фильтр:
plot(x(1:200),y_filtered(1:200))
grid
figure
%Построение спектра сигнала с шумом
```

```

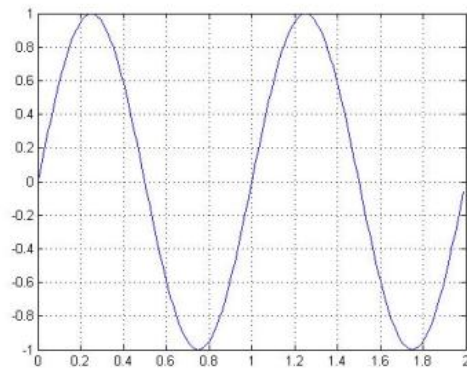
noisy_spectrum = fft(y_noisy,512);
norm_noisy_spectrum = noisy_spectrum.*conj(noisy_spectrum)/512;

%Построение нормированного спектра сигнала с шумом:
plot(f,norm_noisy_spectrum(1:256))
axis([0 max(f) 0 2])
grid
figure

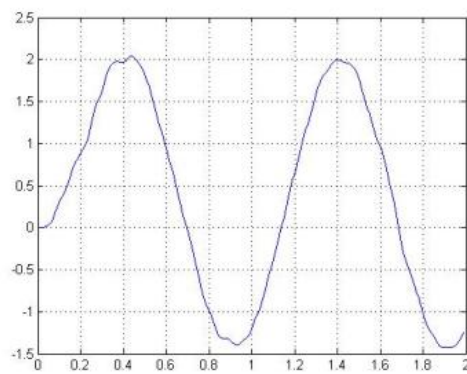
%Спектр отфильтрованного сигнала
spectrum = fft(y_filtered,512);
norm_filtered_spectrum=spectrum.*conj(spectrum)/512;
%Построение нормированного отфильтрованного спектра:
plot(f,norm_filtered_spectrum(1:256))
axis([0 max(f) 0 2])
grid

```

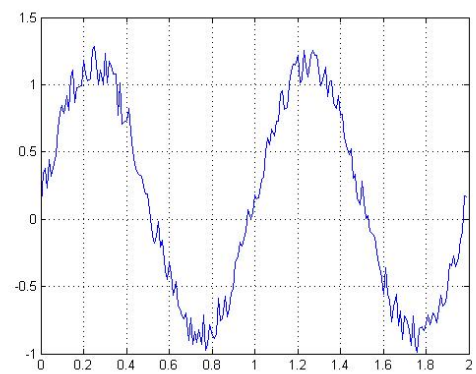
Результаты работы программы на рисунке.



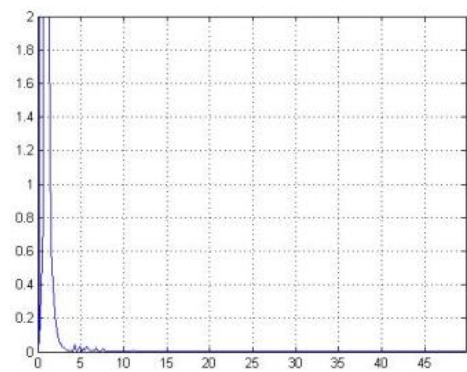
а)



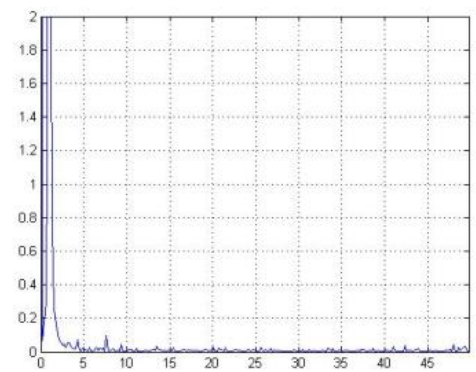
б)



в)



г)



д)

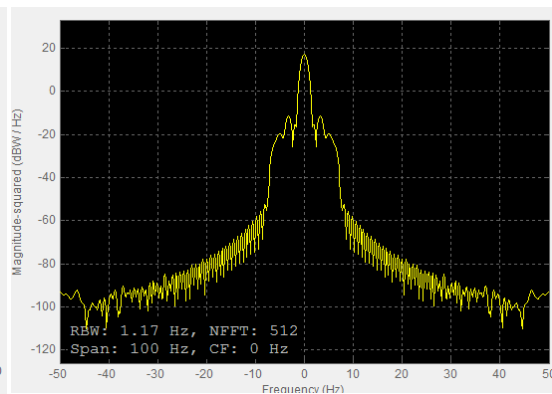
Рис. 1: Линейная фильтрация в Matlab: а) Сигнал без шума, б) Сигнал после фильтрации, в) Сигнал с шумом, г) Спектр сигнала после фильтрации, д) Спектр сигнала с шумом.

4.4 Моделирование в Simulink

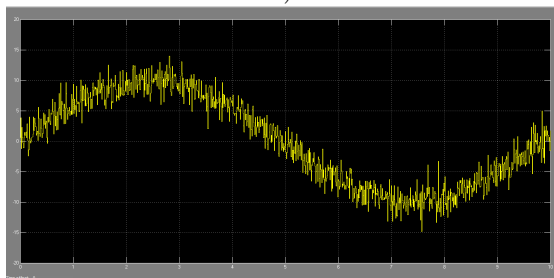
Для создания модели в Simulink используем блок Discrete FIR Filter раздела Discrete главной библиотеки и блок Digital Filter design из Signal Processing Blockset/Filtering/Filter Designs. Результаты моделирование в Simulink на рисунке.



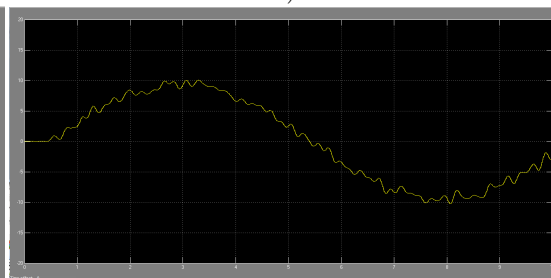
а)



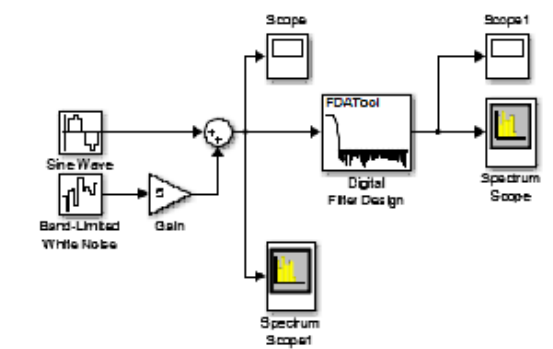
б)



в)



г)



д)

Рис. 2: Линейная фильтрация в Simulink: а) Спектр до фильтрации, б) Спектр после фильтрации, в) Сигнал до фильтрации, г) Сигнал после фильтрации, д) Схема модели.

4.5 Вывод

Фильтр нижних частот — это фильтр, который эффективно пропускает частотный спектр сигнала ниже некоторой частоты и подавляет частоты сигнала выше этой частоты. Степень подавления каждой частоты зависит от вида и порядка фильтра. В лабораторной работе было произведено удаление шума линейным фильтром. Отметим, что на практике фильтрация только подавляет шум, для полного удаления шума необходимо использовать идеальный фильтр с прямоугольным окном, которого на практике не существует. Также заметим, что линейный фильтр подавляет все сигналы, находящиеся в его полосе задержания и не изменяет сигналы из полосы пропускания, и поэтому не может удалить шум, который попал в его полосу пропускания.

5 Аналоговая модуляция

5.1 Цель работы

Изучить процесс амплитудной модуляции/демодуляции сигнала.

5.2 Алгоритм работы

- Сгенерировать однотональный сигнал низкой частоты.
- Выполнить амплитудную модуляцию (АМ) сигнала по закону

$$u(t) = (1 + M * U_m * \cos(\Omega * t)) * \cos(\omega_0 * t + \phi_0)$$

для различных значений глубины модуляции M .

- Получить спектр модулированного сигнала.
- Выполнить модуляцию с подавлением несущей:

$$u(t) = MU_m \cos(\omega t) \cos(\omega_0 t + \phi_0). \quad (1)$$

Получить спектр.

- Выполнить однополосную модуляцию:

$$u(t) = U_m \cos(\omega t) \cos(\omega_0 t + \phi_0) + \frac{U_m}{2} \sum_{n=1}^N M_n (\cos(\omega_0 + \Omega_n)t + \phi_0 + \Phi_n), \quad (2)$$

положив $n = 1$.

- Выполнить синхронное детектирование и получить исходный однополосный сигнал.
- Рассчитать КПД модуляции:

$$\eta_A M = \frac{U_m^2 M^2 / 4}{P_U} = \frac{M^2}{M^2 + 2}. \quad (3)$$

5.3 Теоретические сведения

Модуляция - это процесс, при котором высокочастотная волна используется для переноса низкочастотной волны. Другими словами, перенос спектра сигналов из низкочастотной области в выделенную для их передачи область высоких частот и выполняется операцией модуляции.

При амплитудной модуляции в соответствии с модулирующим сигналом изменяется амплитуда несущего колебания. АМ соответствует переносу информации $s(t) \Rightarrow U(t)$ при постоянных значениях параметров несущей частоты ω и ϕ (частоты и фазы). АМ – сигнал представляет собой произведение информационной огибающей $U(t)$ и гармонического колебания ее заполнения с более высокими частотами. Форма записи амплитудно-модулированного сигнала:

$$u(t) = (1 + M * U_m * \cos(\Omega * t)) * \cos(\omega_0 * t + \phi_0)$$

, где U – постоянная амплитуда несущего колебания при отсутствии входного (модулирующего) сигнала $s(t)$, M – коэффициент амплитудной модуляции. Значение M характеризует глубину амплитудной модуляции. В зависимости от значения M различают нормальную модуляцию ($M < 1$), глубокую модуляцию ($M = 1$) и перемодуляцию ($M > 1$).

Получим спектр модулированного сигнала:

$$u(t) = U_m \cos(\omega_0 t) + (U_m * M/2) \cos[(\omega_0 + \Omega)t] + (U_m * M/2) \cos[(\omega_0 - \Omega)t]$$

КПД амплитудной модуляции равен

$$\eta_A * M = \frac{M^2}{M^2 + 2}$$

Как видно, основная доля мощности АМ – сигнала приходится на несущую частоту. При балансной модуляции (с подавлением несущей) производится перемножение двух сигналов – модулирующего и несущего, при котором происходит подавление несущего колебания, соответственно, КПД модуляции становится равным 100

$$u(t) = M * \cos(\Omega * t)$$

имеем

$$u(t) = M * U_m * \cos(\Omega * t) * \cos(\omega_0 * t) = U_m * \frac{M}{2} * (\cos((\omega_0 + \Omega) * t) + \cos((\omega_0 - \Omega) * t))$$

, т.е. два одинаковых по амплитуде гармонических сигнала с верхней и нижней боковыми частотами. По существу, однотоновый модулирующий сигнал переносится на две высокие частоты.

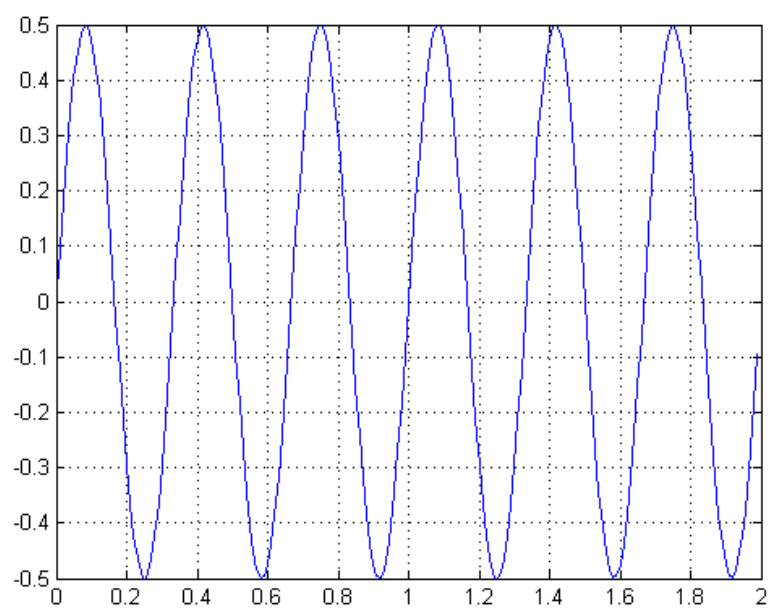
5.4 Код MATLAB для п.1-3

```
f0=3; частота сигнала
fd=150; частота дискретизации
fc=30; частота несущего колебания
x=0:0.01:4*pi;
y=0.5*sin(2*pi*f0*x);
```

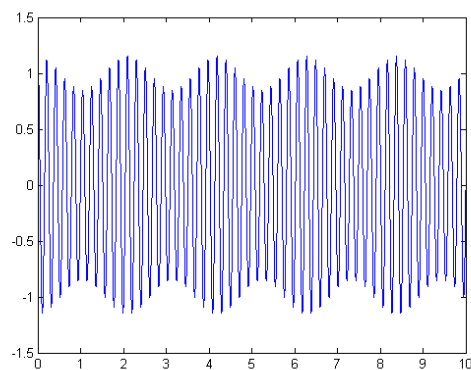
```

plot(x(1:200),y(1:200))
grid;
figure
M1=0.3;
M2=1;
M3=1.3;
t=0:0.001:10;
Um=0.5;
f=100*(0:255)/512;
u1=(1+Um*M1*cos(f0*t)).*cos(fc*t); модулированный сигнал
plot(t,u1)
figure
s1=fft(u1,512); спектр модулированного сигнала
ss1=s1.*conj(s1)/512;
plot(f,ss1(1:256));
figure
u2=Um*(1+M2*cos(f0*t)).*cos(fc*t);
plot(t,u2)
figure
s2=fft(u2,512);
ss2=s2.*conj(s2)/512;
plot(f,ss2(1:256));
figure
u3=Um*(1+M3*cos(f0*t)).*cos(fc*t);
plot(t,u3)
figure
s3=fft(u3,512);
ss3=s3.*conj(s3)/512;
plot(f,ss3(1:256));

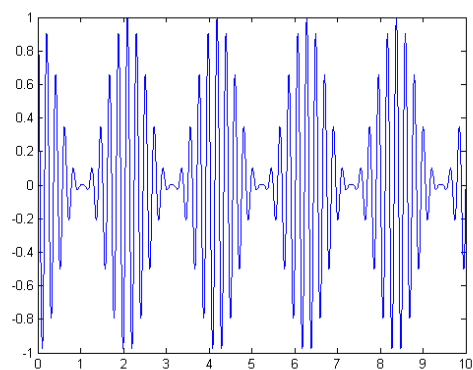
```



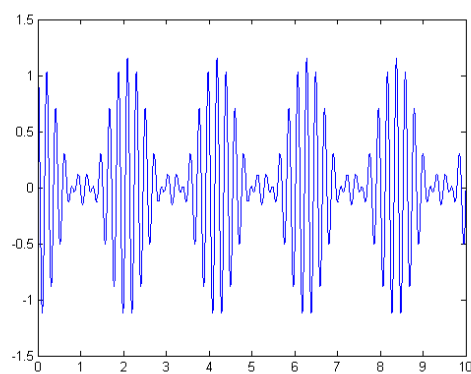
Исходный низкочастотный сигнал



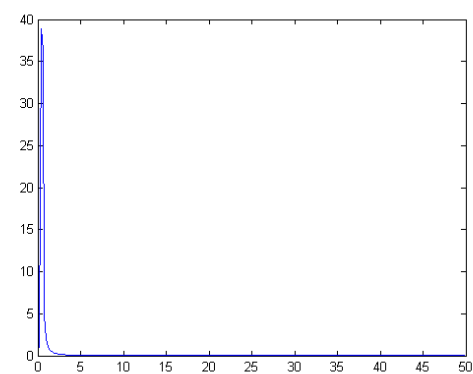
а)



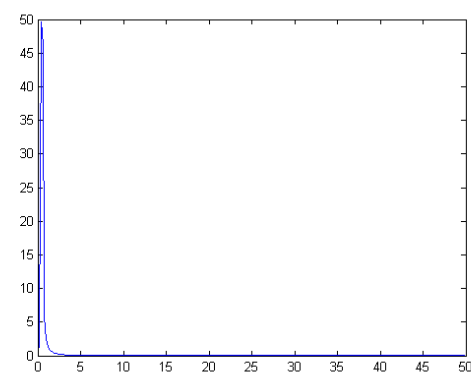
б)



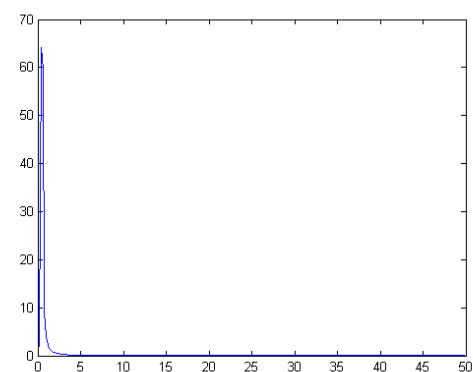
в)



г)



д)

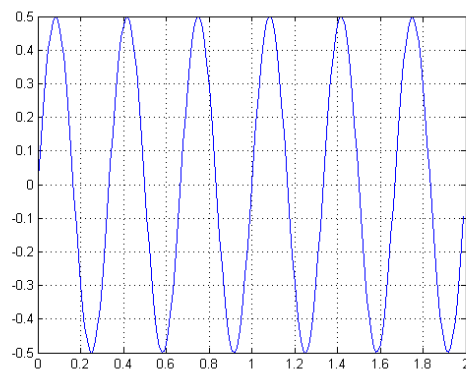


е)

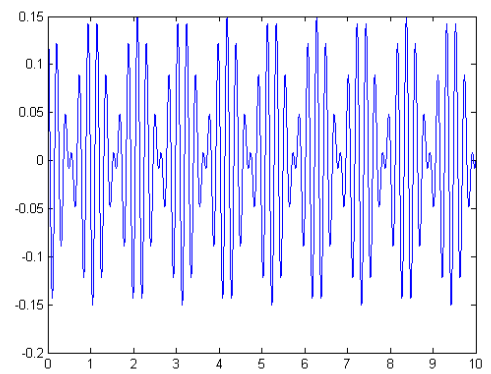
Рис. 3: Линейная фильтрация в Simulink: а) АМ при степени глубины=0.3, б) АМ при степени глубины=1, в) АМ при степени глубины=1.3, г) Спектр модулированного сигнала при степени глубины=0.3, д) Спектр модулированного сигнала при степени глубины=1, е) Спектр модулированного сигнала при степени глубины=1.3.

5.5 Код MATLAB для п.4

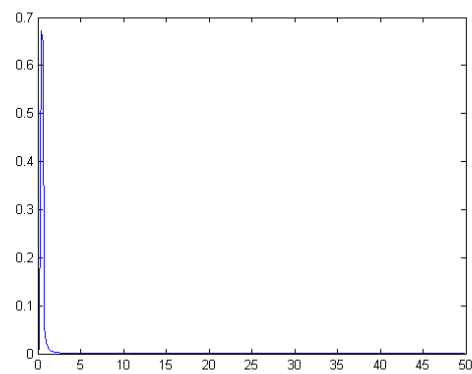
```
plot(x(1:200),y(1:200)) исходный сигнал
grid;
figure
u4=Um*M1*cos(f0*t).*cos(fc*t); модуляция с подавлением несущей
plot(t,u4)
figure
s4=fft(u4,512);
ss4=s4.*conj(s4)/512;
plot(f,ss4(1:256)); figure
plot(t,YY);
```

а)



б)

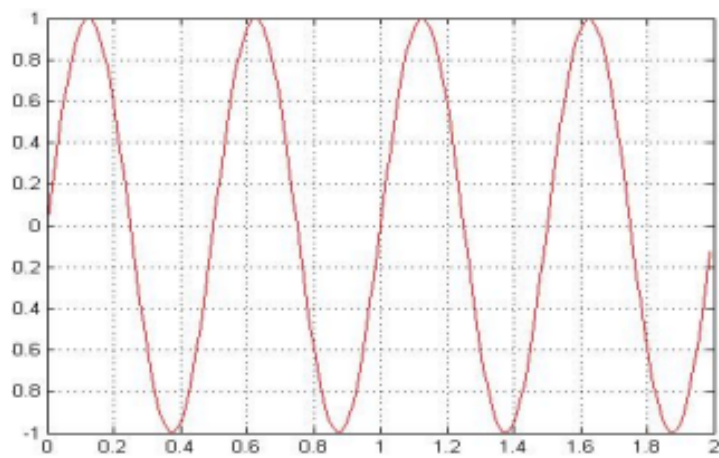


в)

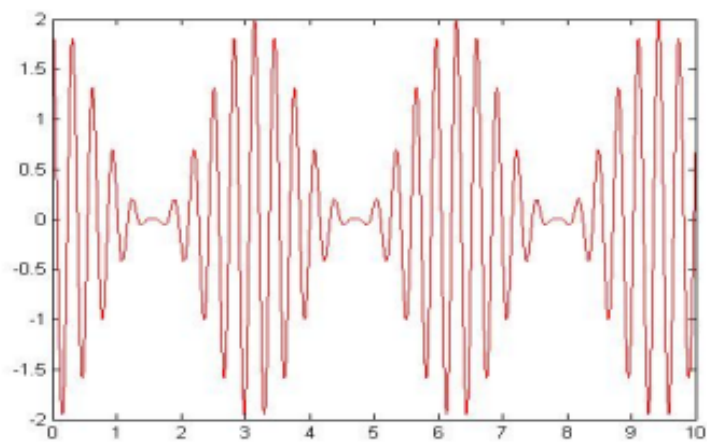
Рис. 4: Линейная фильтрация в Simulink: а) Исходный низкочастотный сигнал, б) Модулированный сигнал с подавлением несущей, в) Спектр модулированного сигнала с подавлением несущей.

5.6 Код MATLAB для п.5

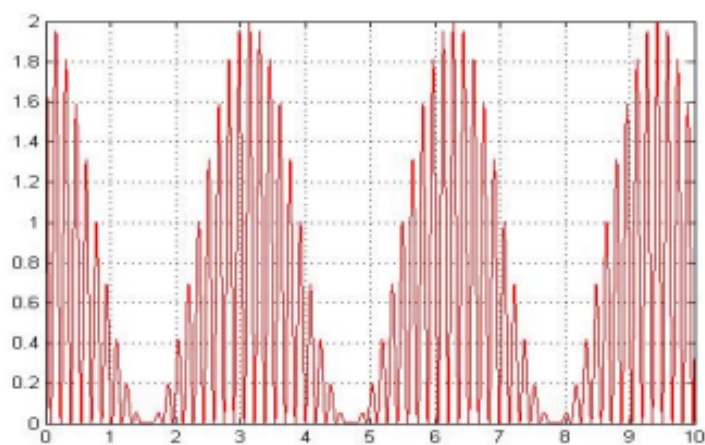
```
x=0:0.01:4*pi;  
y=sin(2*pi*2*x);  
plot(x(1:200),y(1:200),'r')  
t=0:0.001:10;  
u2=(1+1*cos(2*t)).*cos(20*t);  
s2=fft(u2,512);  
yy=u2.*cos(20*t);  
plot(t,yy,'r');  
ss=fft(yy,512);  
sss=ss.*conj(ss)/512;  
plot(f,sss(1:256),'r');
```



а)



б)



в)

Рис. 5: Линейная фильтрация в Simulink: а) Исходный сигнал, б) Модулированный сигнал, в) Сигнал после выполнения синхронного детектирования.

5.7 Код MATLAB для п.6

Найдем КПД модуляции по формуле

$$\eta_A * M = \frac{M^2}{M^2 + 2}$$

M=0.3; КПД=0.14

M=1; КПД=0.3(3)

M=1.3; КПД=0.35

5.8 Вывод

В результате выполнения лабораторной работы был сгенерирован однональный сигнал низкой частоты, выполнена АМ сигнала и получен спектр модулированного сигнала, выполнена БАМ и однопосная АМ и получены их спектры. Далее произведено синхронное детектирование и получен исходный однопосный сигнал. Рассчитан КПД модуляции.

АМ - сигнал представляет собой произведение информационной огибающей $U(t)$ и гармонического колебания ее заполнения с более высокими частотами. Значение M должно находиться в пределах от 0 до 1 для всех гармоник модулирующего сигнала. Малую глубину модуляции ($M < 1$) применять нецелесообразно, т.к. при этом мощность передаваемого информационного сигнала будет много меньше мощности несущего колебания, и мощность передатчика используется неэкономично.

В настоящее время амплитудная модуляция применяется редко (для радиовещания и для передачи изображения в телевизионном вещании) из-за низкого КПД использования энергии модулированных сигналов.

Для однотоновой модуляции начальная фаза модулирующего колебания для верхней боковой частоты складывается с начальной фазой несущей, для нижней – вычитаются из фазы несущей. Как в случае с обычной АМ, так и в случае АМ с подавленной несущей ширина спектра АМ-сигнала в два раза больше, чем у модулирующего сигнала. При балансной амплитудной модуляции производится перемножение двух сигналов – модулирующего и несущего, при котором происходит подавление несущего колебания, и тогда КПД модуляции становится равным 100.

Синхронное детектирование - это один из способов демодуляции АМ-сигнала. Его суть состоит в умножении частоты сигнала на опорное колебание с несущей частотой. В результате умножения получаем два слагаемых: искомая амплитуда и АМ-сигнал с несущей частотой $2\omega_0$, который удаляется в результате пропускания сигнала через ФНЧ.

6 Частотная и фазовая модуляция

6.1 Цель работы

Изучение частотной и фазовой модуляции/демодуляции сигнала.

6.2 Постановка задачи

1. Сгенерировать однотональный сигнал низкой частоты.
2. Выполнить фазовую модуляцию/демодуляцию сигнала по закону

$$u(t) = U_m \cos(\Omega t + k s(t)), \quad (4)$$

используя встроенную функцию MatLab `pmmod`, `pmdemod`.

3. Получить спектр модулированного сигнала.
4. Выполнить частотную модуляцию/демодуляцию по закону

$$u(t) = U_m \cos(\omega_0 t + k \int_0^t s(t) dt + \phi_0), \quad (5)$$

используя встроенные функции MatLab `fmmod`, `fmdemod`.

6.3 Справочные материалы

Н.В. Богач и др. Обработка сигналов в информационных системах, с. 118-125, 127-133.

6.4 Теоретический материал

Фазовая модуляция – процесс изменения мгновенной фазы несущего колебания пропорционально изменению непрерывного информационного сигнала, т.е. информационный сигнал управляет фазой несущего колебания (амплитуда и частота несущего колебания остаются постоянными). Фазомодулированный сигнал $s(t)$ имеет следующий вид:

$$s(t) = g(t) \sin[2\pi f_c t + \varphi(t)], \quad (6)$$

где $g(t)$ — огибающая сигнала; $\varphi(t)$ является модулирующим сигналом; f_c — частота несущего сигнала; t — время.

Частотная модуляция - это процесс изменения мгновенной частоты несущего колебания в соответствии и изменением информационного сигнала, т.е. информационный сигнал управляет частотой несущего колебания (амплитуда и фаза несущего колебания остаются постоянными).

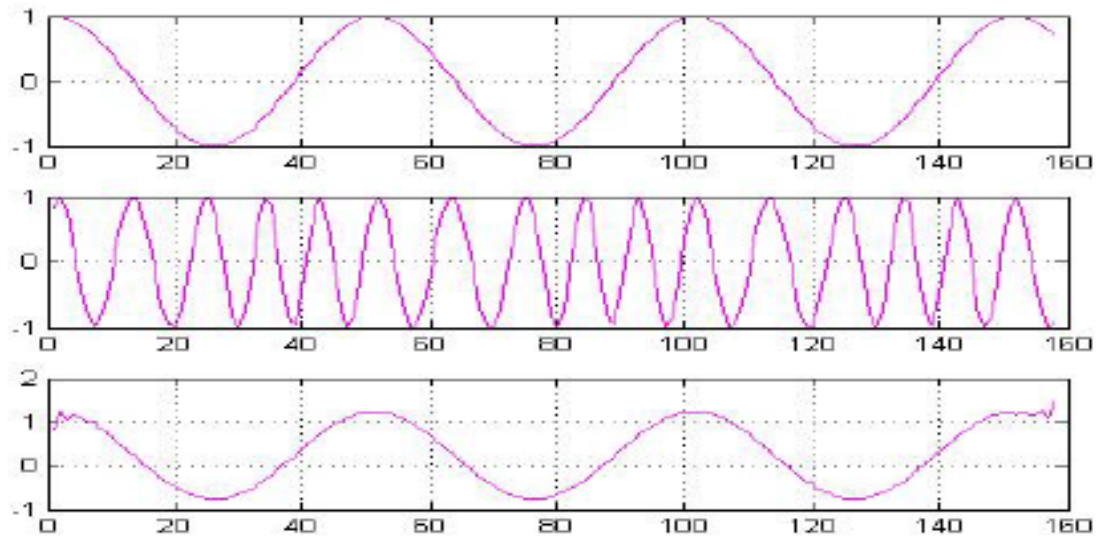
По характеристикам фазовая модуляция близка к частотной модуляции. В случае синусоидального модулирующего (информационного) сигнала, результаты частотной и фазовой модуляции совпадают.

6.5 Ход работы

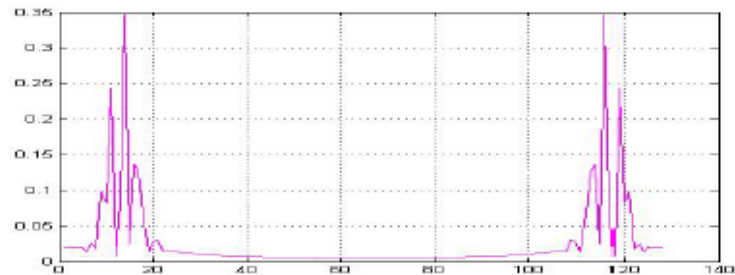
1. Сгенерирован однотональный сигнал низкой частоты. Выполнена фазовая модуляция/демодуляция сигнала. Получен спектр модулированного сигнала. Код в командном окне MATLAB:

```
x = 0:(1/(8*pi)):2*pi;  
y = sin(pi*x+pi/2);  
y1=pmmod(y, 5, 50, 1, 3*pi/2); Фазовая модуляция
```

$y_2 = \text{pmdemod}(y_1, 5, 50, 1, 3\pi/2)$; Фазовая демодуляция



Исходный, модулированный, демодулированный сигналы



Спектр

2. Выполнена частотная модуляция/демодуляция сигнала. Получен спектр модулированного сигнала.

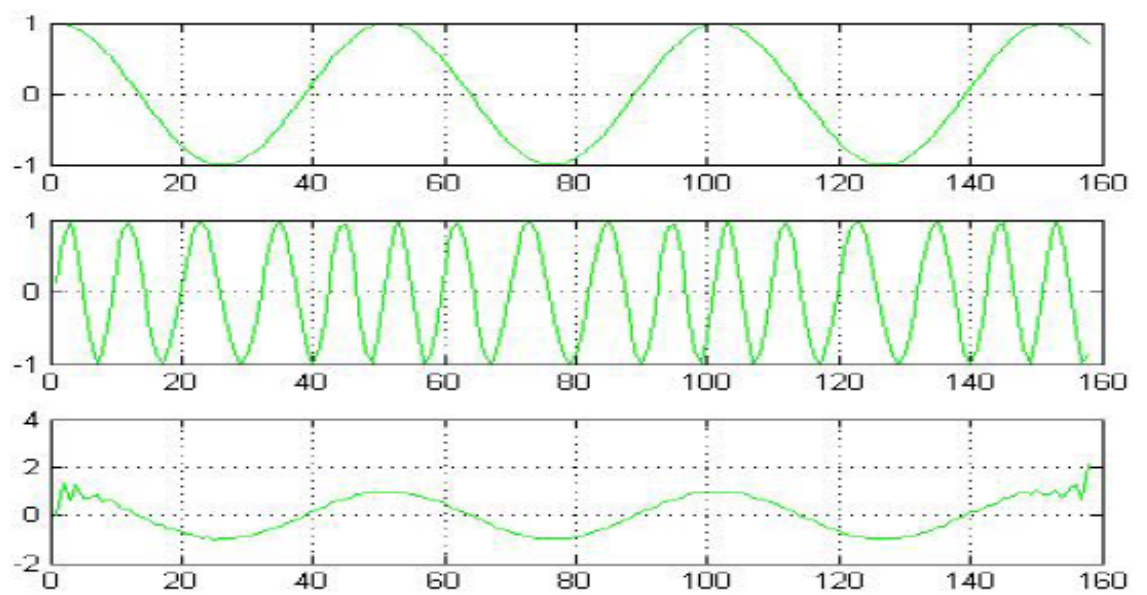
Код в командном окне MATLAB:

```
x = 0:(1/(8*pi)):2*pi;
```

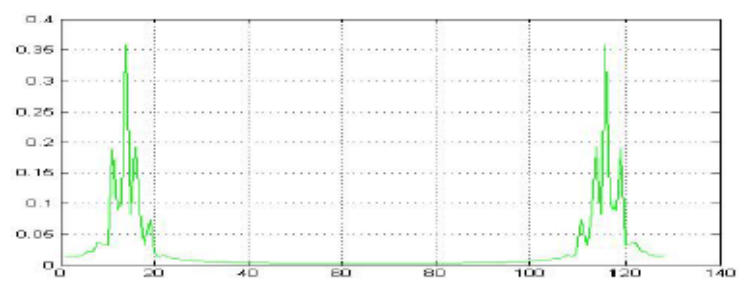
```
y = sin(pi*x+pi/2);
```

```
z1=fmmod(y, 5, 50, 1, 3*pi/2); Частотная модуляция
```

```
z2=fmdemod (z1, 5, 50, 1, 3*pi/2); Частотная демодуляция
```



Исходный, модулированный, демодулированный сигналы



Спектр

3. Фазовая модуляция в среде Multisim

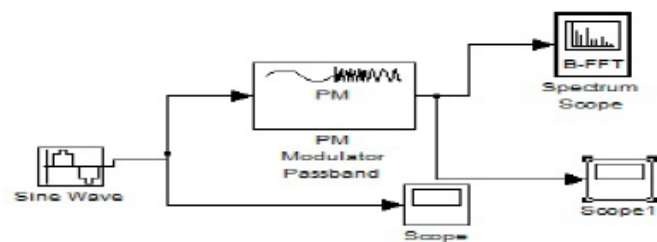
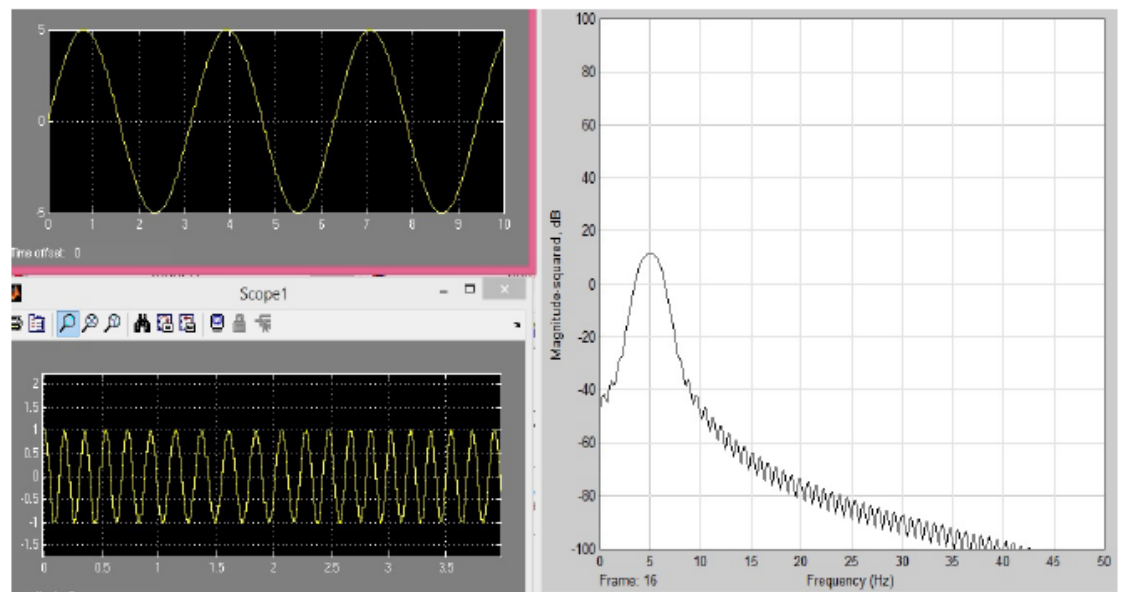


Схема simulink



Исходный сигнал, модулированный сигнал, спектр модулированного сигнала

4. Частотная модуляция в среде Multisim

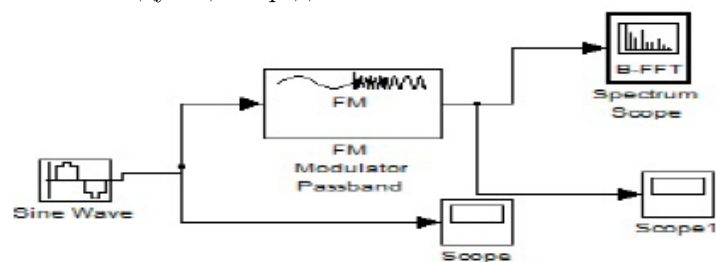
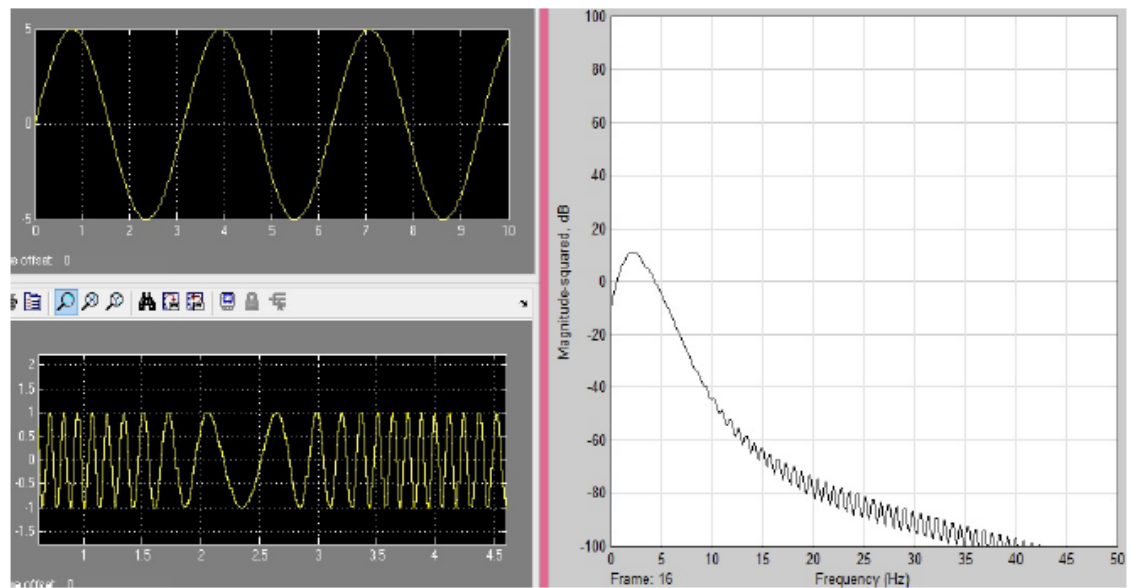


Схема simulink



Исходный сигнал, модулированный сигнал, спектр модулированного сигнала

5. Блок фазовой автоподстройки частоты(ФАПЧ)

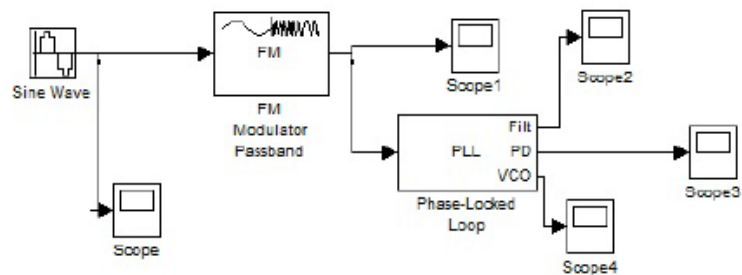
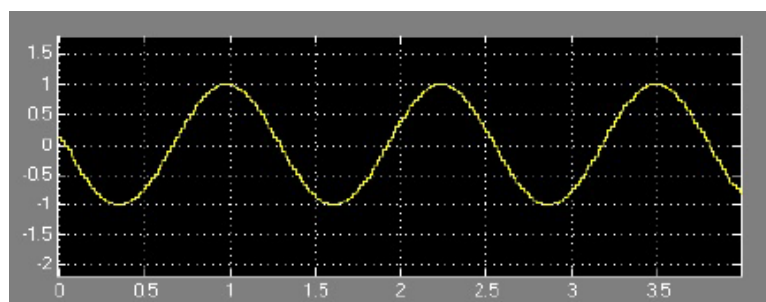
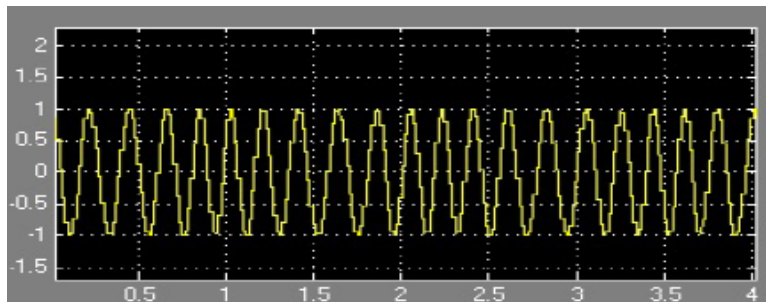


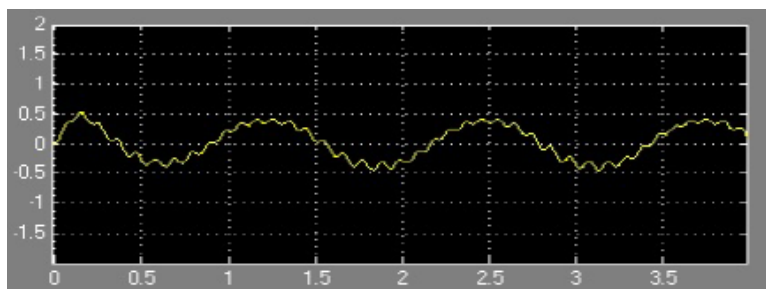
Схема simulink



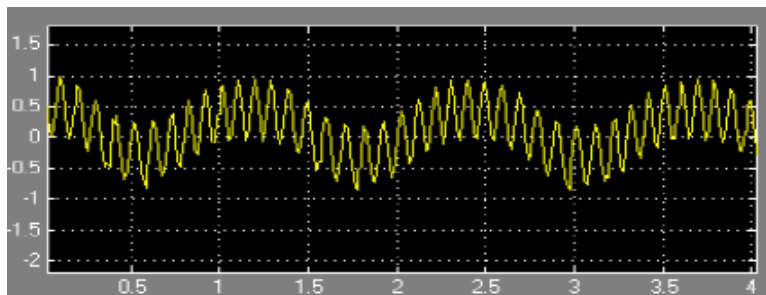
Исходный сигнал



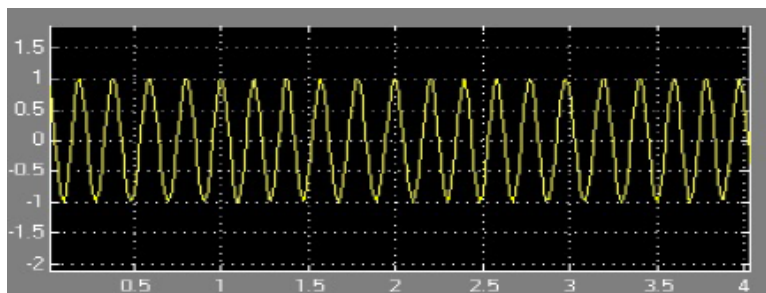
Модулированный сигнал



Сигнал на выходе ФНЧ



Сигнал на выходе фазового детектора



Сигнал на выходе ГУН(генератор, управляемый напряжением)

6.6 Выводы

В результате выполнения данной работы были выполнены частотная и фазовая модуляция/демодуляция, а также частотная демодуляция с помощью блока захвата фазы. Сделаем вывод, что частотная и фазовая модуляция очень тесно взаимосвязаны, поскольку обе они влияют на аргумент функции \cos . Поэтому эти два вида модуляции имеют общее название — угловая модуляция. Сигнал с угловой модуляцией имеет вид колебания, начальная фаза которого зависит от времени:

$$s(t) = A_0 \cos(\omega_0 t + j(t)). \quad (7)$$

В случае фазовой модуляции частота несущей зависит от производной модулируемого сигнала, а в случае частотной — от самого его значения.

Схема ФАПЧ позволяет обеспечить точную настройку, частотную селекцию и фильтрацию без использования громоздких элементов фильтров, используемых в схемах детектирования. ФАПЧ представляет собой систему управления с петлей обратной связи, в которой параметрами регулирования являются частота или фаза сигнала. При подаче на вход схемы сигнала с угловой модуляцией на выходе ФНЧ после захвата частоты и фазы формируется сигнал, пропорциональный модулирующему сигналу, что позволяет применять схему ФАПЧ для демодуляции. Для демодуляции использовалась петля ФАПЧ, состоящая из перемножителя (используемого в качестве фазового детектора), фильтра нижних частот и генератора, управляемого напряжением (ГУН). Если собственная частота ГУН f_0 достаточно близка к частоте внешнего опорного сигнала f_1 , то под действием обратной связи в схеме ФАПЧ ГУН синхронизируется, то есть захватывает внешний входной сигнал. Поэтому выходная частота ГУН — это сумма или разность его собственной частоты и разницы между внешней опорной частотой и собственной частотой ГУН.

7 Цифровая модуляция

7.1 Цель работы

Изучение методов модуляции цифровых сигналов и сравнение их свойств.

7.2 Алгоритм работы

- Получить сигналы BPSK, PSK, OQPSK, genQAM, MSK, M-FSK модуляторов
- Построить их сигнальные созвездия.
- Провести сравнение изученных методов модуляции цифровых сигналов.

7.3 Теоретическая часть

В настоящее время все большая часть информации, передаваемой по разнообразным каналам связи, существует в цифровом виде. Это означает,

что передаётся последовательность целых чисел, которые могут принимать значения из некоторого фиксированного конечного множества. Эти числа, называемые символами (symbol), поступают от источника информации с периодом T , а частота, соответствующая этому периоду, называется символьной скоростью (symbol rate): $fT = 1/T$. Последовательность передаваемых символов является дискретным сигналом. Поскольку символы принимают значения из конечного множества, этот сигнал является и квантованным, то есть его можно назвать цифровым сигналом. В данной работе рассматриваются вопросы, связанные с преобразованием этого цифрового сигнала в аналоговый модулированный сигнал. Типичный подход при осуществлении передачи дискретной последовательности символов состоит в следующем. Каждому из возможных значений символа сопоставляется некоторый набор параметров несущего колебания. Эти параметры поддерживаются постоянными в течение интервала T , то есть до прихода следующего символа. Фактически это означает преобразование последовательности чисел nk в ступенчатый сигнал $sn(t) : sn(t) = f(nk), kT \leq t < (k + 1)T$. Здесь f — некоторая функция преобразования. Полученный сигнал $sn(t)$ далее используется в качестве модулирующего сигнала обычным способом. Такой способ модуляции, когда параметры несущего колебания меняются скачкообразно, называется манипуляцией (keying). В зависимости от того, какие именно параметры изменяются, различают амплитудную (АМн), фазовую (ФМн), частотную (ЧМн) и квадратурную (КАМн) манипуляцию. Цифровая модуляция и демодуляция включают в себя две стадии. При модуляции цифровое сообщение сначала преобразуется в аналоговый модулирующий сигнал с помощью функции `modmap`, а затем осуществляется аналоговая модуляция. При демодуляции сначала получается аналоговый демодулированный сигнал, а затем он преобразуется в цифровое сообщение с помощью функции `demodmap`. Функция `ganderr` предназначена для формирования ошибок в цифровом сигнале. Она дает матрицу, в каждой строке которой имеется заданное число случайно расположенных ненулевых элементов. Для оценки помехоустойчивости системы связи необходимо произвести сравнение исходного (передаваемого) сообщения с сообщением, полученным в результате приема, и определить число ошибок, возникших в процессе передачи, а также вероятность ошибки. Эти действия выполняются функциями `sumerr` и `biterr`, первая из которых подсчитывает число несовпадающих символов в двух сообщениях, а вторая — число несовпадающих битов в двоичных представлениях этих символов. Кроме числа ошибок, обе функции могут возвращать долю ошибок в общем числе символов (битов) и индикаторы мест возникновения ошибок. Последние две функции данной группы предназначены для графического отображения сигналов с квадратурной манипуляцией. Функция `eyediagram` выводит так называемую глазковую диаграмму, а функция `scatterplot` — диаграмму рассеяния. Аналоговый несущий сигнал модулируется цифровым битовым потоком. Существуют три фундаментальных типа цифровой модуляции (или шифтинга) и один гибридный:

1. ASK – Amplitude shift keying (Амплитудная двоичная модуляция).
2. FSK – Frequency shift keying (Частотная двоичная модуляция).
3. PSK – Phase shift keying (Фазовая двоичная модуляция).
4. ASK/PSK.

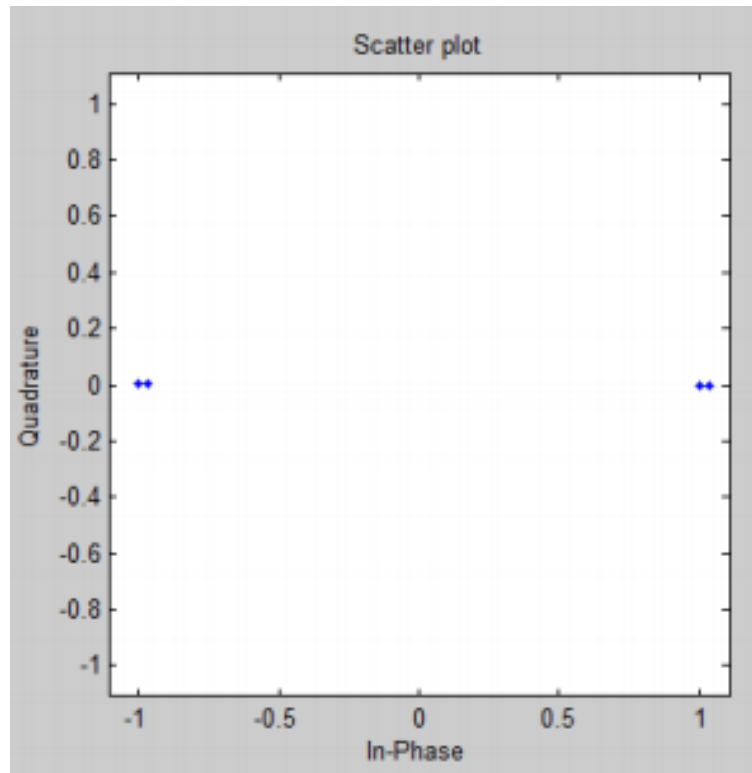
Одна из частных реализаций схемы ASK/PSK, которая называется QAM - Quadrature Amplitude Modulation (квадратурная амплитудная модуляция (КАМ)). Это метод объединения двух АМ-сигналов в одном канале. Он позволяет удвоить эффективную пропускную способность. В QAM используется две несущих с одинаковой частотой но с разницей в фазе на четверть периода (отсюда и возникает слово квадратура). Частотная модуляция представляет логическую единицу интервалом с большей частотой, чем ноль. Фазовый шифтинг представляет «0» как сигнал без сдвига, а «1» как сигнал со сдвигом.

BPSK : используется единственный сдвиг фазы между «0» и «1» — 180 градусов, половина периода. Существуют также QPSK: QPSK использует 4 различных сдвига фазы (по четверти периода) и может кодировать 2 бита в символе (01, 11, 00, 10).

7.4 Код MATLAB

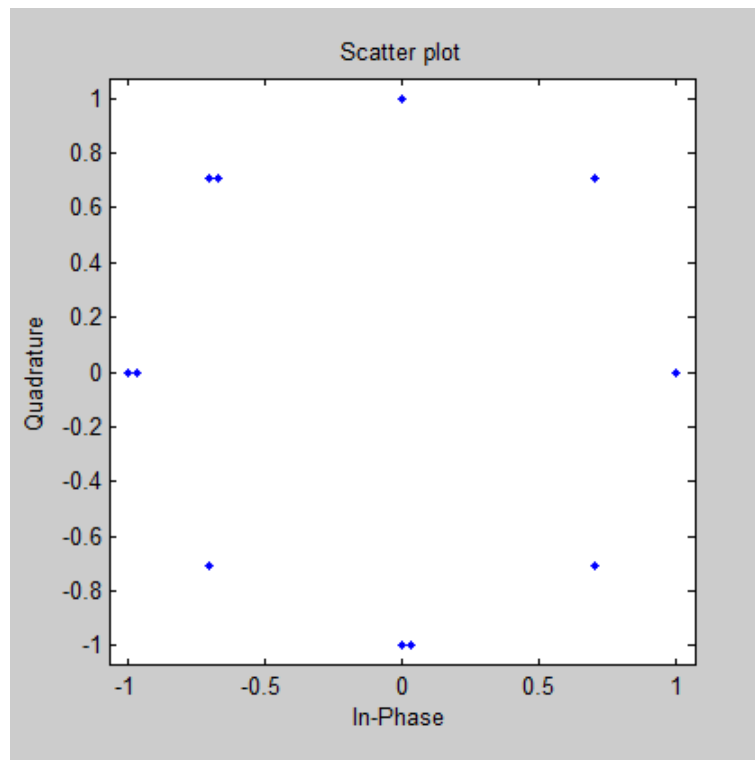
7.4.1 BPSK modulation

```
h = modem.pskmod('M', 4);  
g = modem.pskdemod('M', 4);  
msg = randint(10,1,2);  
modSignal = modulate(h,msg);  
errSignal = (randerr(1,10, 3) ./ 30)';  
modSignal = modSignal + errSignal;  
demodSignal = demodulate(g,modSignal);  
scatterplot(modSignal);
```



7.4.2 PSK modulation

```
h = modem.pskmod('M', 8);
g = modem.pskdemod('M', 8);
msg = randint(100,1,8);
modSignal = modulate(h,msg);
errSignal = (randerr(1,100, 3) ./ 30)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);
```

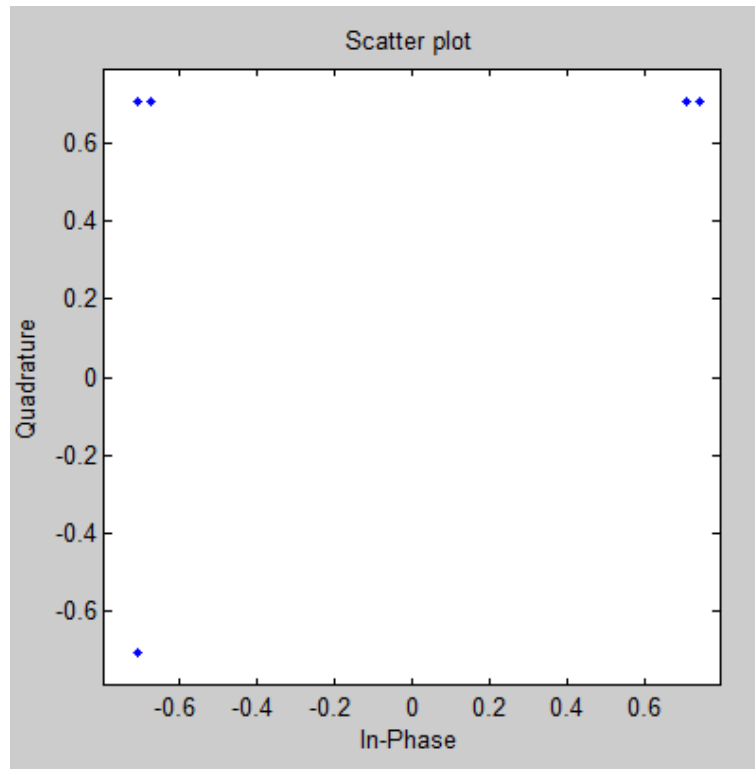


7.4.3 OQPSK modulation

```

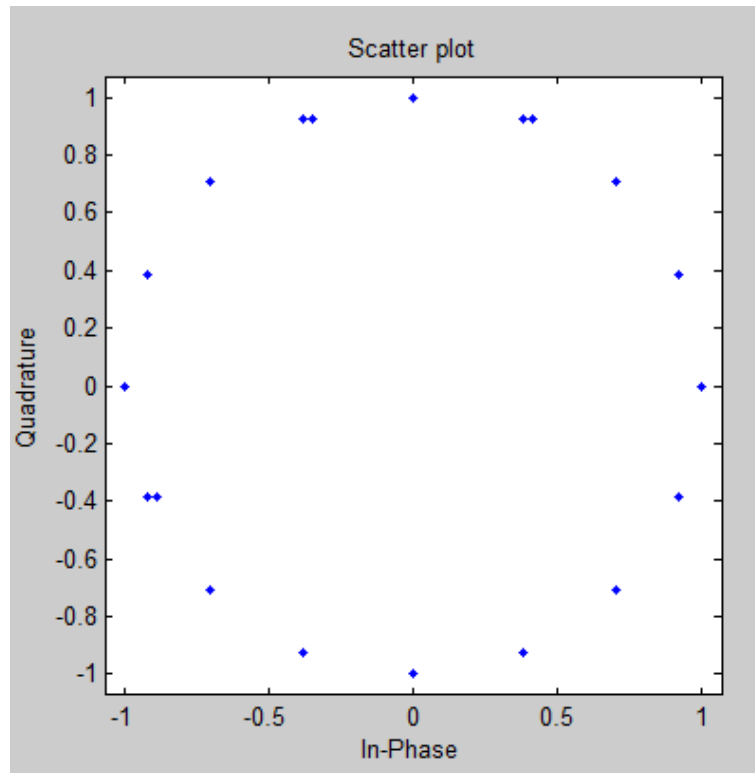
h = modem.oqpskmod;
g = modem.oqpskdemod;
msg = randint(200,1,2);
modSignal = modulate(h,msg);
errSignal = (randerr(1,400, 100) ./ 30)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);

```



7.4.4 GENQAM modulation

```
M = 16;
h = modem.genqammod('Constellation', exp(j*2*pi*[0:M-1]/M));
g = modem.genqamdemod('Constellation', exp(j*2*pi*[0:M-1]/M));
msg = randint(100,1,16);
modSignal = modulate(h,msg);
errSignal = (randerr(1,100, 3) ./ 30)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);
```

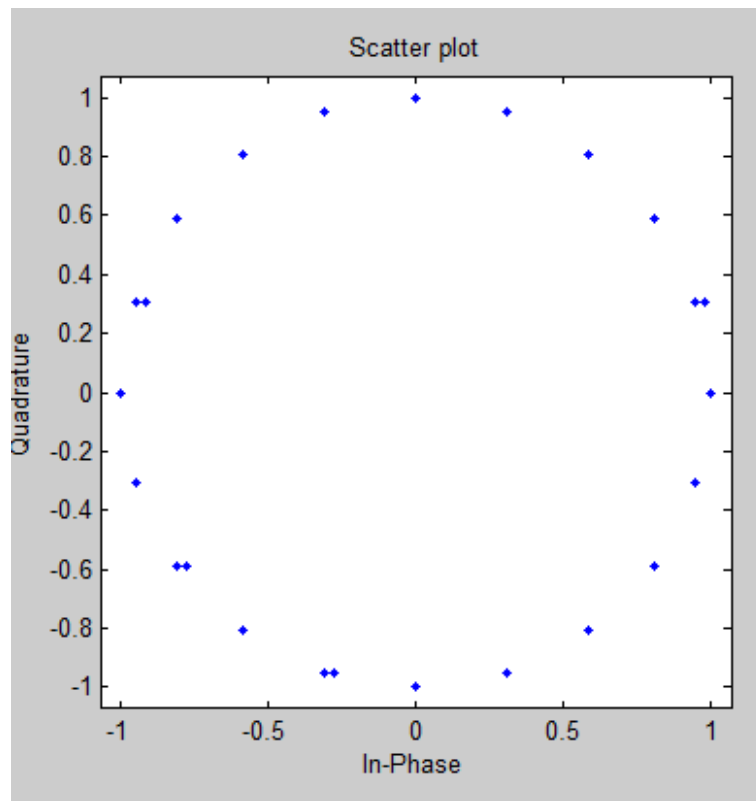



7.4.5 MSK modulation

```

h = modem.mskmod('SamplesPerSymbol', 5);
g = modem.msksdemod('SamplesPerSymbol', 5);
msg = randint(200,1,2);
modSignal = modulate(h, msg);
errSignal = (randerr(1,1000, 5) ./ 30)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g, modSignal);
scatterplot(modSignal);

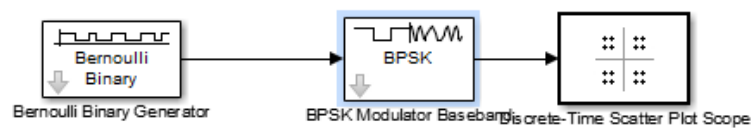
```



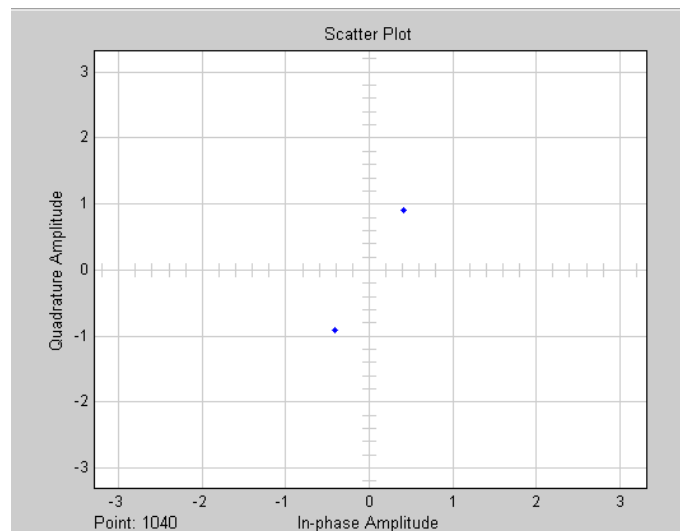
7.5 Работа в среде Simulink

7.5.1 BPSK modulation

- Получен сигнал BPSK и построено сигнальное созвездие в среде Simulink

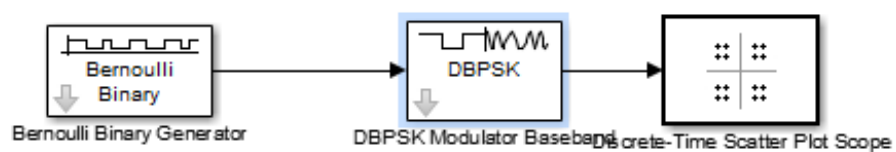


- Сигнальное созвездие

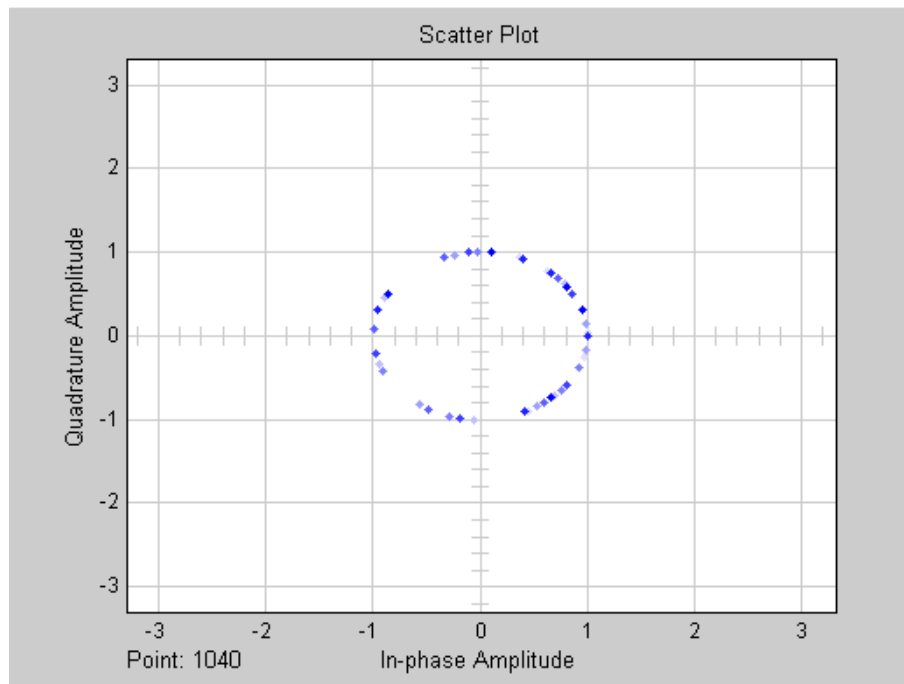


7.5.2 DBPSK modulation

- Получен сигнал DBPSK и построено сигнальное созвездие в среде Simulink

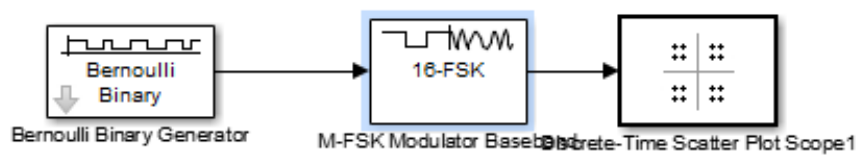


- Сигнальное созвездие

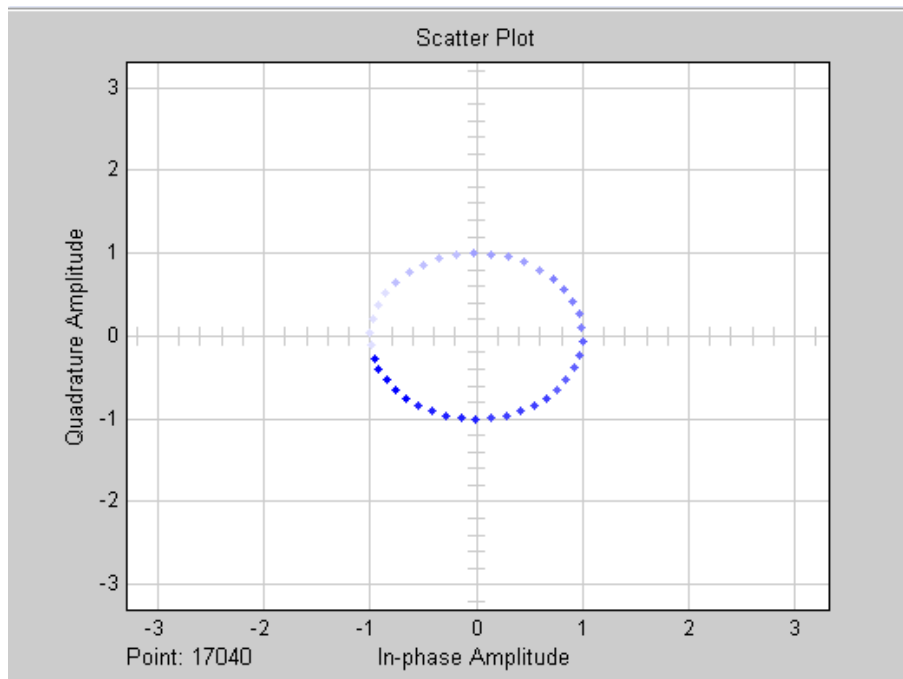


7.5.3 M-FSK modulation

- Получен сигнал M-FSK и построено сигнальное созвездие в среде Simulink



- Сигнальное созвездие



7.6 Вывод

Уровень модуляции определяет количество состояний несущей, используемых для передачи информации. Чем выше этот уровень, тем большими скоростными возможностями и меньшей помехоустойчивостью модуляция обладает. Число бит, передаваемых одним состоянием, определяется как $\log_2 N$, где N — уровень модуляции. Таким образом, чем выше уровень модуляции, тем больше данных мы можем передать (или потерять) за единицу времени.