

Отчет по лабораторной работе по дисциплине:
"Сети и системы передачи данных"

Киселев Антон и Кенть Никита

13.03.14

Глава 1

Визуализация сигналов во временной и частотной области

1.1 Цель работы

Познакомиться со средствами генерации сигналов и визуализации их спектров.

1.2 Постановка задачи

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

1.3 Теоретическая часть

В ходе данной лабораторной работы необходимо промоделировать чистый синусоидальный сигнал, а так же синусоидальный сигнал с шумом и получить их представления во временной и частотной областях. Синусоидальный сигнал задаётся по следующей формуле:

$$A(t) = A_0 * \sin(2 * \pi * f * t + u_0)$$

Для создания зашумленного синусоидального сигнала, к чистому синусоидальному сигналу прибавляется случайная составляющая, по формуле:

$$A(t) = A_0 * \sin(2 * \pi * f * t + u_0) + A_1 * rand()$$

Для выделения частот регулярных составляющих сигнала необходимо использовать преобразование Фурье, реализуемое следующей формулой:

$$X(k) = \sum_{j=1}^N x(j) * e^{2 * \pi * \frac{x}{N(j-1)(k-1)}}$$

1.4 Алгоритм работы. Построение сигналов

- Построение синусоидального сигнала без шумов
- Вывод временной характеристики сигнала
- Реализация преобразования Фурье
- Построение графика спектральной плотности
- Построение зашумленного синусоидального сигнала
- Вывод временной характеристики полученного сигнала
- Преобразования Фурье
- Построение графика спектральной плотности для зашумленного сигнала

1.5 Код MATLAB

```
function main()
x=0:0.01:4*pi;
t0 = 5;
%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,1024);
normspectrum = spectrum.*conj(spectrum)/1024;
f=100*(0:1023)/1024;
plot(f, normspectrum(1:1024))
axis([0 max(f) 0 10])
grid
%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,1024);
noizespectrum = spectrum.*conj(spectrum)/1024;
figure(4)
plot(f, noizespectrum())
axis([0 max(f) 0 10])
grid
```

1.6 Результаты работы

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

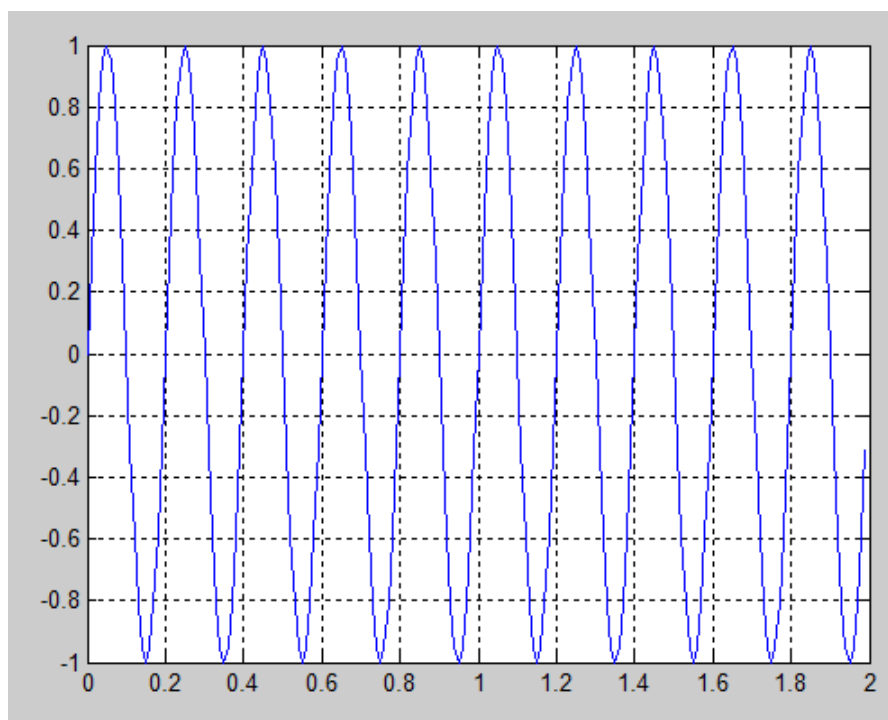


рис. 1 Исходный сигнал

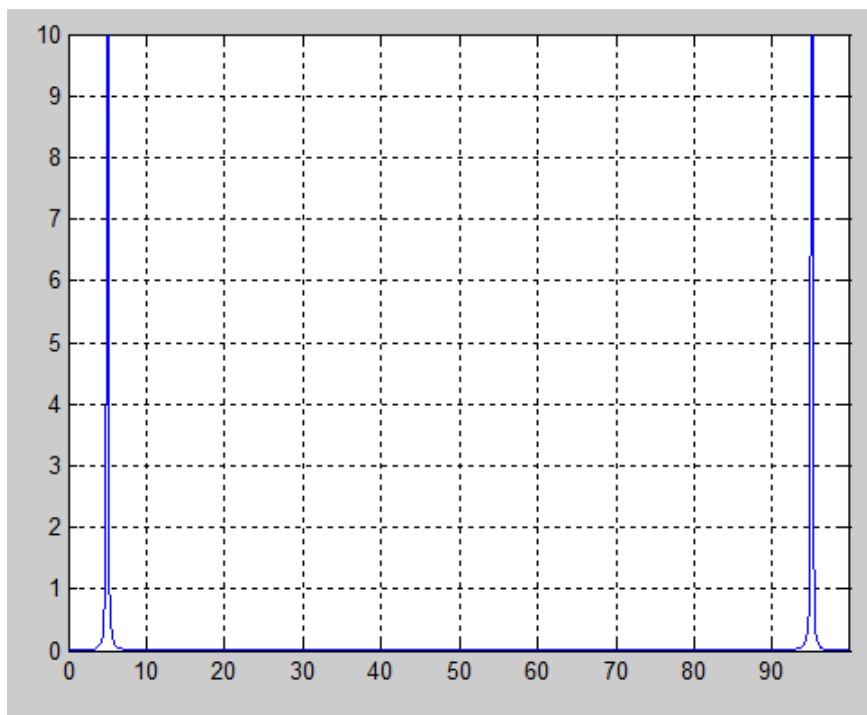


рис. 2. Спектр исходного сигнала

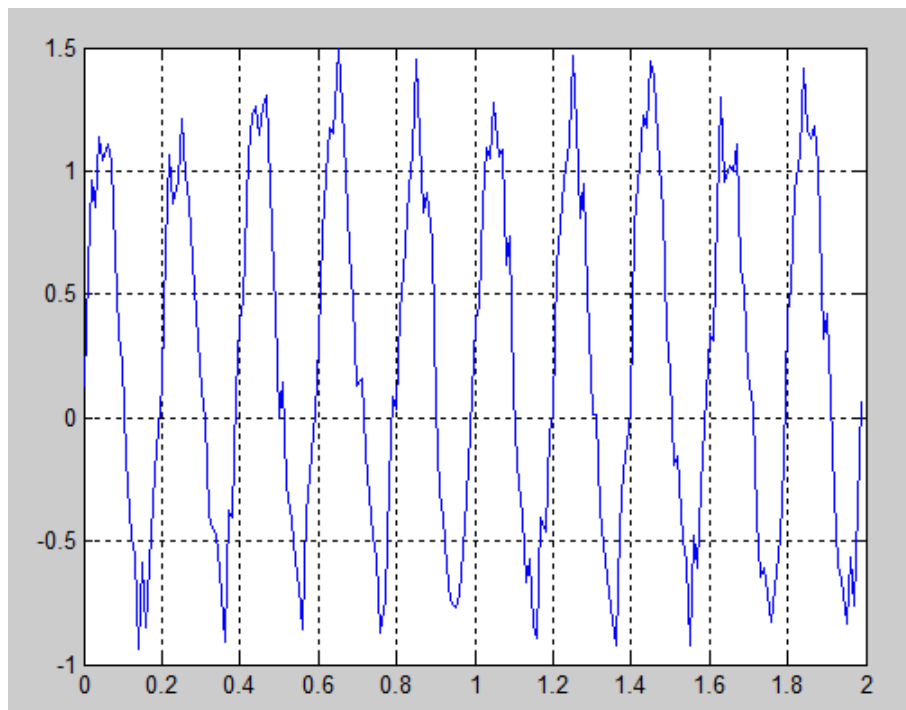


рис. 3. Зашумленный сигнал

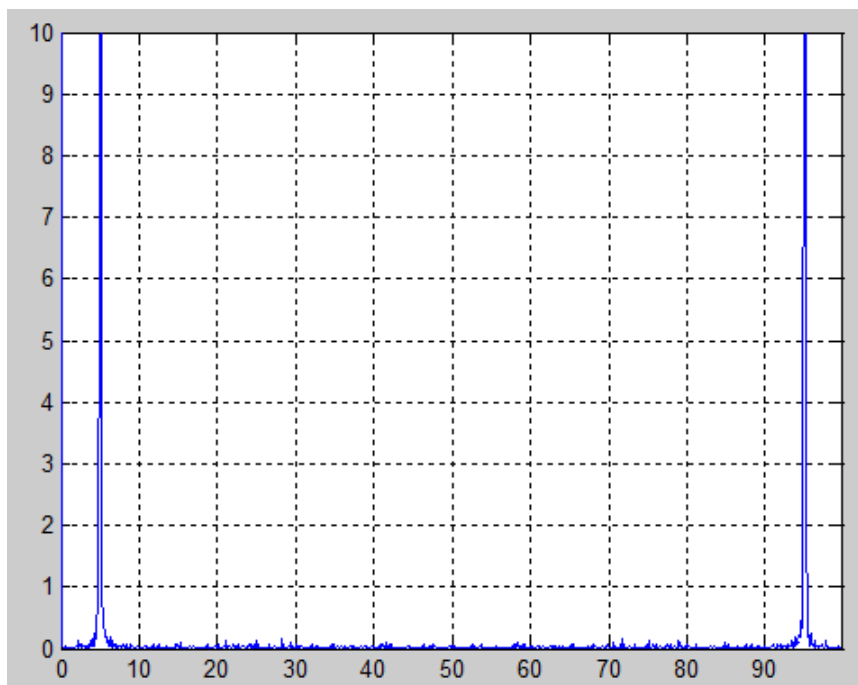


рис. 4. Спектр зашумленного сигнала

1.7 Вывод

В данной лабораторной работе было проведено исследование по получению и сравнению спектров чистого и зашумленного сигнала. Спектры сигналов вычислялись на ограниченном промежутке. Для этого длина промежутка и частота квантования выбирались так, чтобы результаты вычислений были достаточно точными. Полученные результаты при вычислении спектра оказались ожидаемыми. Теоретически ожидалось получить периодический спектр синусоидального сигнала как один импульс. Вследствие дискретности исходного сигнала, при выполнении операции по получению спектра, синусоидальный сигнал умножается на решетку дельта-импульсов. Происходит свертка исходного сигнала с решеткой дельта-импульсов. Поэтому при практическом опыте был получен периодический спектр исходного сигнала, представленный в виде нескольких импульсов на графике..

Глава 2

Построение спектров

2.1 Алгоритм работы

- Построение полигармонического сигнала
- Построение прямоугольного импульсного сигнала
- Построение треугольного импульсного сигнала
- Получение спектров этих сигналов
- Создание моделей в Simulink

2.2 Теоретическая часть

В данной работе мы рассматриваем три типа сигналов: полигармонический, прямоугольный импульсный и треугольный. Их формулы представлены ниже:

- полигармонический сигнал

$$y(t) = \sum_{n=0}^N -1 \cos(nt)$$

- прямоугольный импульсный сигнал

$$y(t) = \Pi(t, T_i)$$

- треугольный импульсный сигнал

$$y(t) = \Delta(t, T_i)$$

Для получения треугольного сигнала используется операция свертки двух прямоугольных сигналов, производимая по формуле:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(y)g(x-y)dy$$

2.3 Код MATLAB

```
function laba5()
x = 0:0.01:4*pi;
f0 = 5;
y = 0;
for i=1:1:100
y=y+cos(i*x);
end
plot(x(1:100),y(1:100));
figure(1)
spectrum=fft(y,512);
normspectrum=spectrum.*conj(spectrum)/512;
f=100*(0:255)/512;
figure(2)
plot(f,normspectrum(1:256))
axis([0 max(f) 0 10])
grid
figure(3)
y1=square(x,50)
plot(x(1:1000),y1(1:1000),'LineWidth',2);
ylim([-1.2,1.2]);
spectrum=fft(y1,512);
normspectrum=spectrum.*conj(spectrum)/512;
f1=100*(0:255)/512;
figure(4)
plot(f1,normspectrum(1:256))
axis([0 max(f1) 0 10])
grid
y2=conv(y1,y1);
figure(5)
plot(x(1:1000),y2(1:1000),'LineWidth',2);
grid
spectrum=fft(y2,512);
normspectrum=spectrum.*conj(spectrum)/512;
f2=100*(0:255)/512;
figure(6)
plot(f2,normspectrum(1:256)/1000)
axis([0 max(f2) 0 10])
grid
end
```

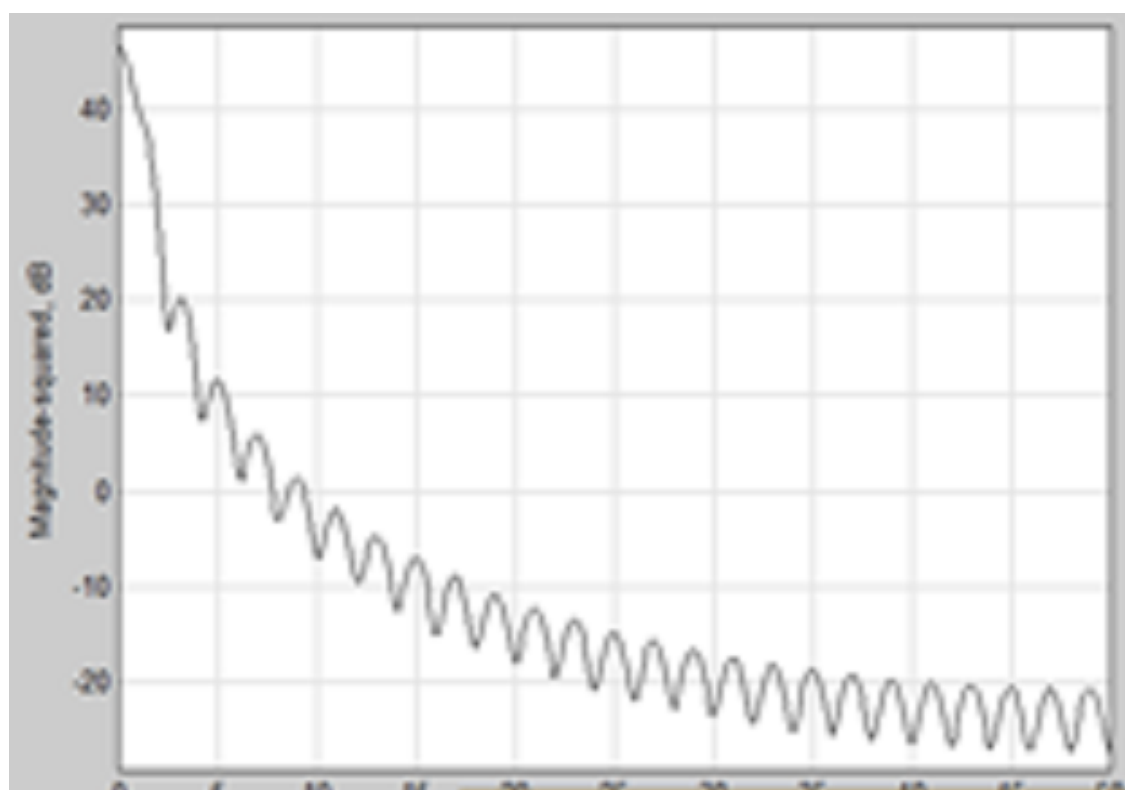


рис. 5. спектр прямоугольного сигнала

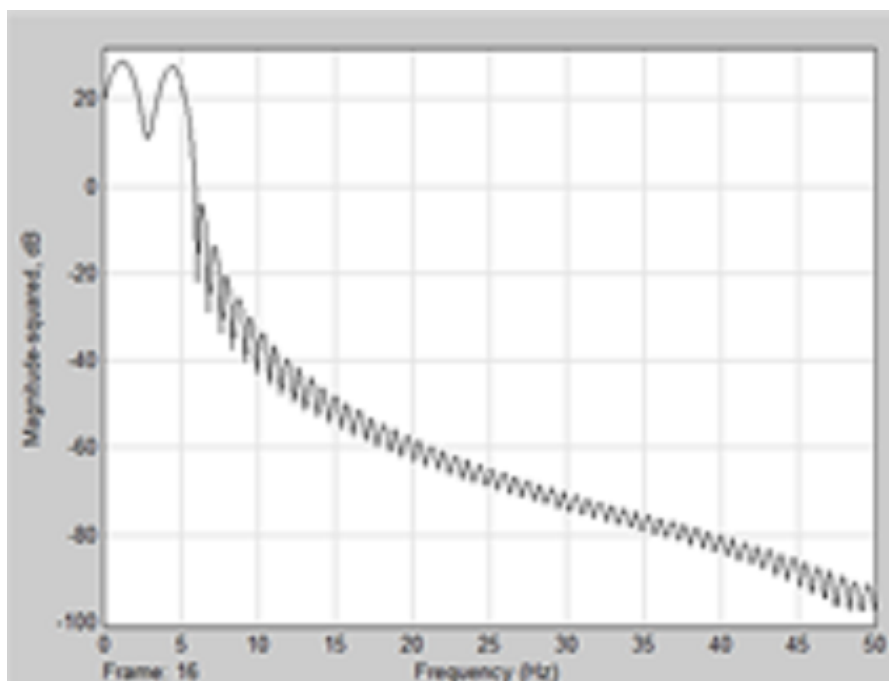


рис. 6. спектр полигармонического сигнала

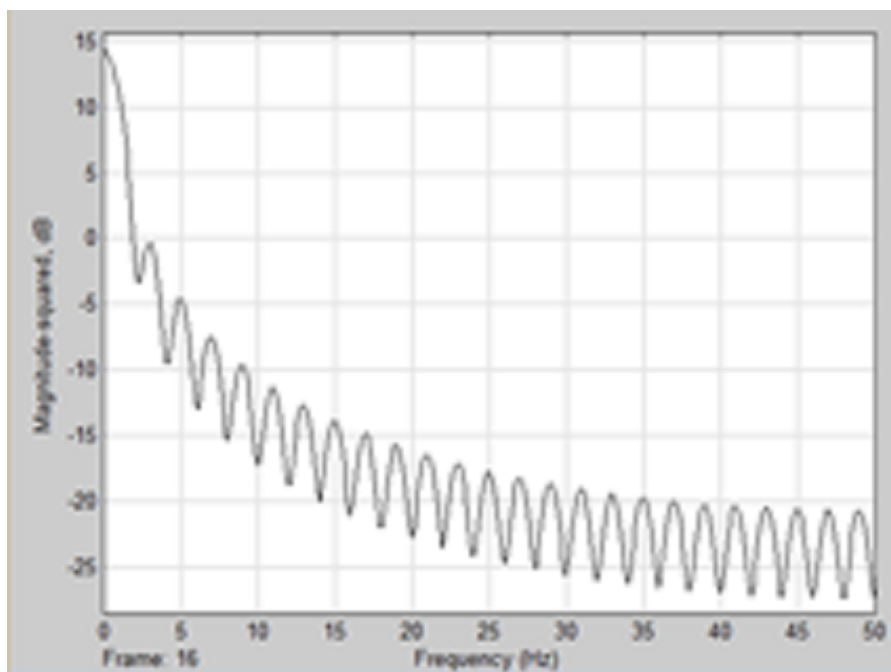


рис. 7. спектр треугольного сигнала

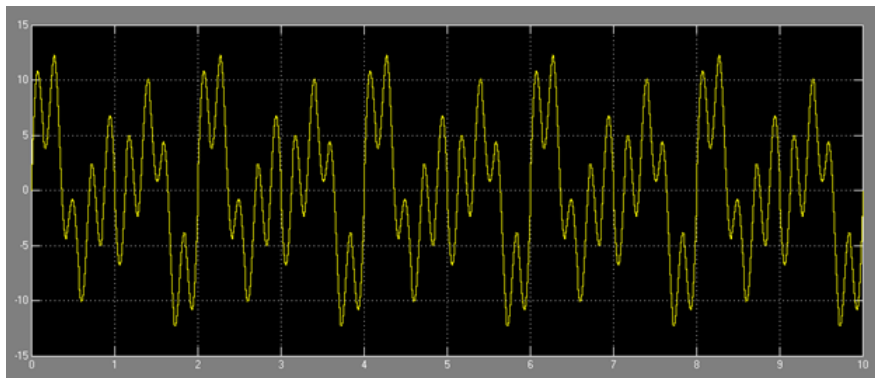


рис. 8 полигармонический сигнал

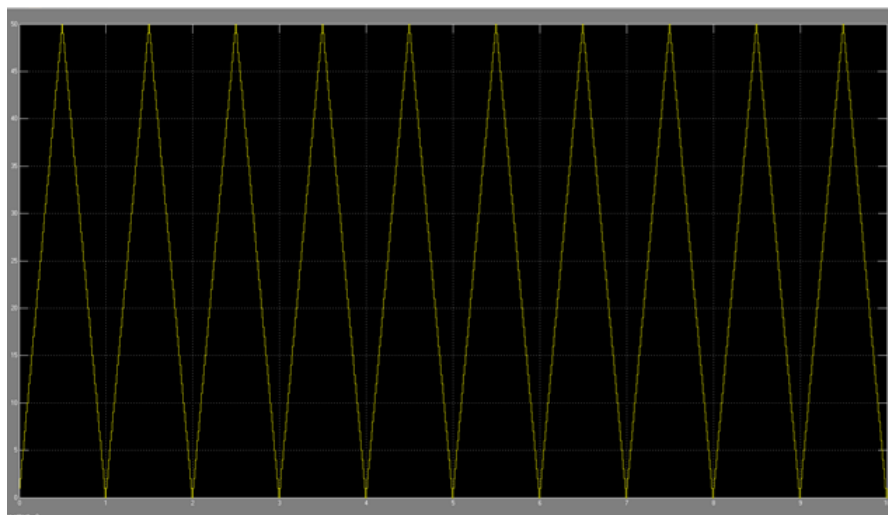


рис. 9 треугольный сигнал

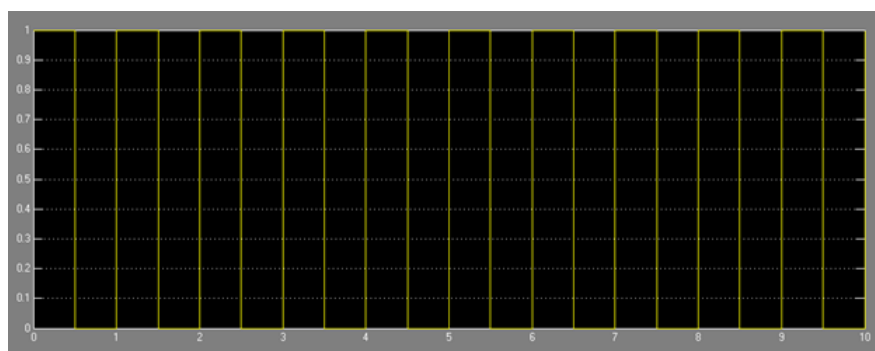


рис. 10 прямоугольный сигнал

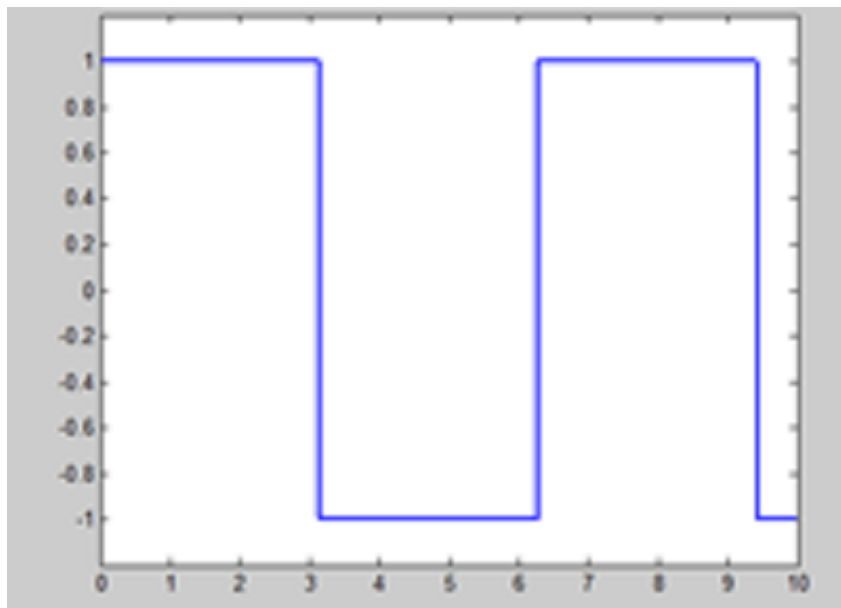


рис. 11 прямоугольный сигнал

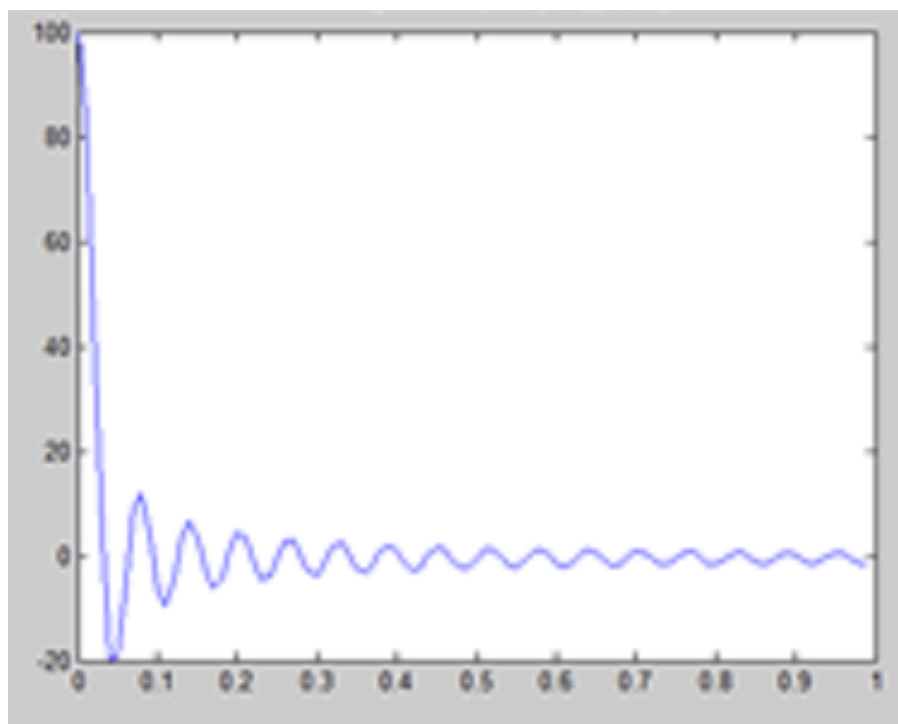


рис. 12 полигармонический сигнал

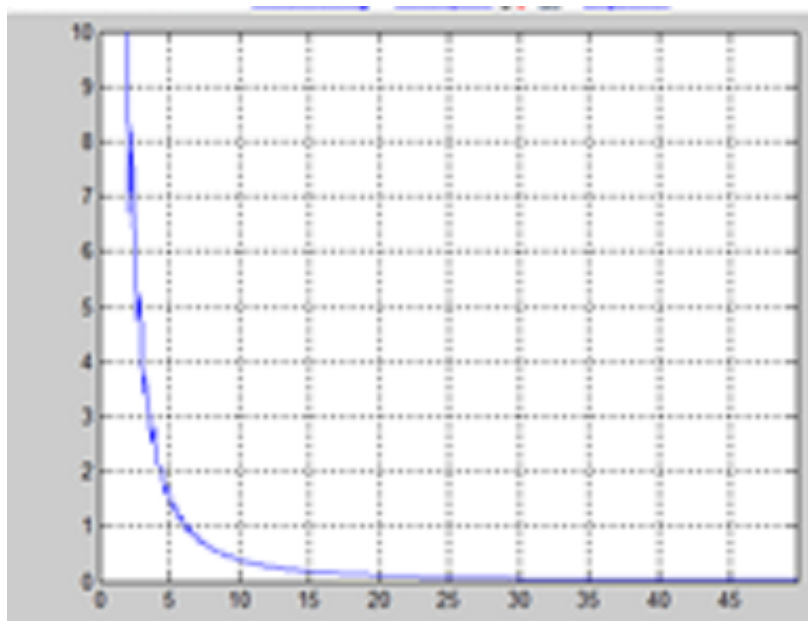


рис. 13 спектр треугольного сигнала

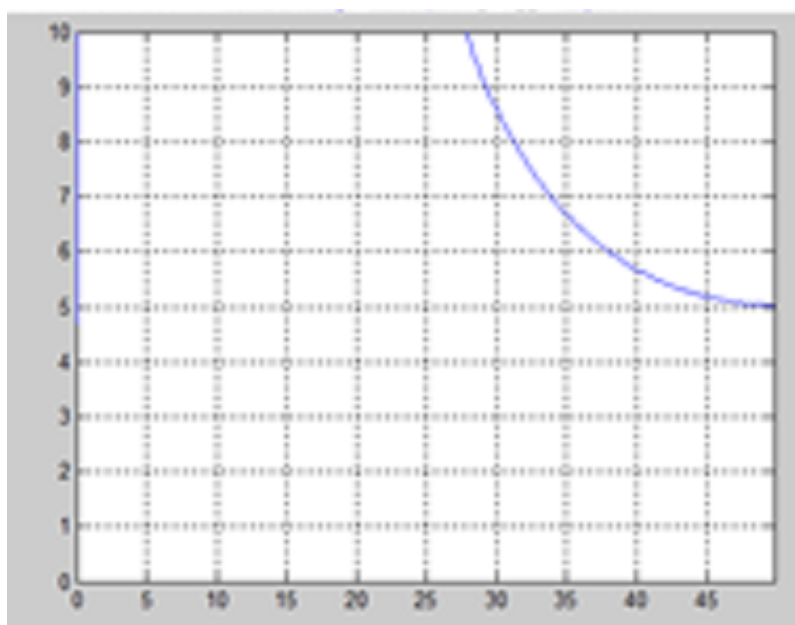


рис. 14 спектр прямоугольного сигнала

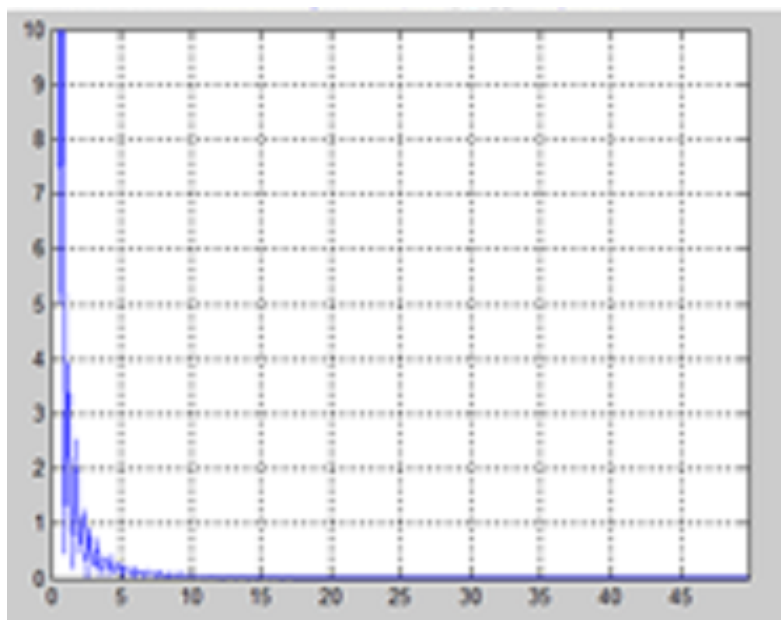


рис. 15 спектр полигармонического сигнала

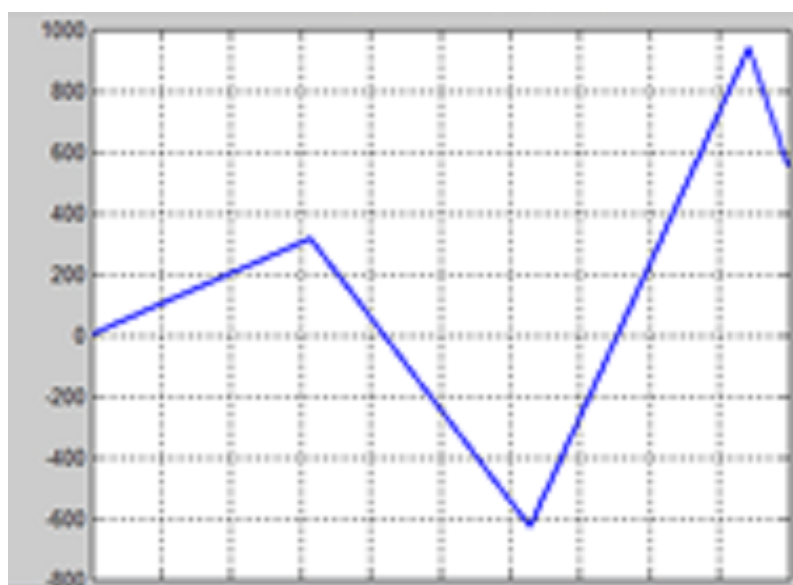


рис. 16 треугольный сигнал

2.4 Вывод

В данной лабораторной работе было проведено моделирование полигармонического сигнала, прямоугольного сигнала и треугольного сигнала. Были получены спектры данных сигналов. Для получения сигналов использовались математические формулы данных функций, а также средство моделирования Simulink. Обаими способами были получены спектры сигналов. Частным случаем в работе было получение треугольного сигнала через применение операции свертки над произведением двух прямоугольных сигналов. Данная операция осуществляется как математически, так и при моделировании в среде Simulink. 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 линейных функции, одна из которых имеет положительный коэффициент наклона, а другая отрицательный.

Глава 3

Лабораторная работа №6. Линейная фильтрация

3.1 Теоретическая часть

Фильтр низких частот - один из видов фильтров, эффективно пропускающий частотный спектр сигнала ниже определенной частоты (частоты среза) и уменьшающий частоты сигнала выше этой частоты. Для реализации фильтра в данной лабораторной работе был реализован фильтр Баттерворта. Фильтр Баттерворта проектируется так, чтобы его амплитудно-частотная характеристика была максимально гладкой на частотах полосы пропускания. Порядок фильтра прямопропорционален скорости подавления сигнала выше полосы пропускания. Так же замечено, что при более высоком порядке фильтра, частота среза выражена ярче. Амплитудно-частотная характеристика $G(w)$ фильтра Баттерворта n го порядка может быть получена из передаточной функции $H(s)$:

$$G^2(w) = |H^2(s)| = \frac{G_0^2}{1 + (\frac{w}{w_c})^2}$$

где:

- n — порядок фильтра
- w_c — частота среза (частота на которой амплитуда равна 3dB)
- G_0 — коэффициент усиления по постоянной составляющей (усиление на нулевой частоте)

3.2 Алгоритм работы

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.

3.3 Код MATLAB

```
function seti6()

    x = 0:0.01:4*pi;
    f=100*(0:255)/512;
    figure(1)
    noise=rand(size(x));
    y = sin(2*pi*x);
    ynoisy = y+0.3*noise;
    plot(x(1:200),y(1:200))
    title('Исходный сигнал')
    grid
    figure(2)
    plot(x(1:200),ynoisy(1:200))
    title('Исходный сигнала с шумом')
    grid
    [B,A] = butter(16,0.98); B=B./sum(B);
    A=A./sum(A);
    figure(3)
    yfiltered = conv(ynoisy,[B,A]);
    plot(x(1:200),yfiltered(1:200))
    title('Отфильтрованный зашумленный сигнал')
    grid
    figure(4)
    normalspectrum = fft(y,512);
    normspectrum = normalspectrum.*conj(normalspectrum)/512;
    plot(f,normspectrum(1:256))
    axis([0 max(f) 0 2])
    title('Спектр исходного сигнала')
    grid
    figure(5)
    noisyspectrum = fft(ynoisy,512);
    normnoisyspectrum = noisyspectrum.*conj(noisyspectrum)/512;
    plot(f,normnoisyspectrum(1:256))
    axis([0 max(f) 0 2])
    title('Спектр зашумленного сигнала')
    grid
    figure(6)
    spectrum = fft(yfiltered,512);
    normfilteredpectrum=spectrum.*conj(spectrum)/512;
    plot(f,normfilteredpectrum(1:256))
    axis([0 max(f) 0 2])
    title('Спектр отфильтрованного сигнала')
    grid
end
```

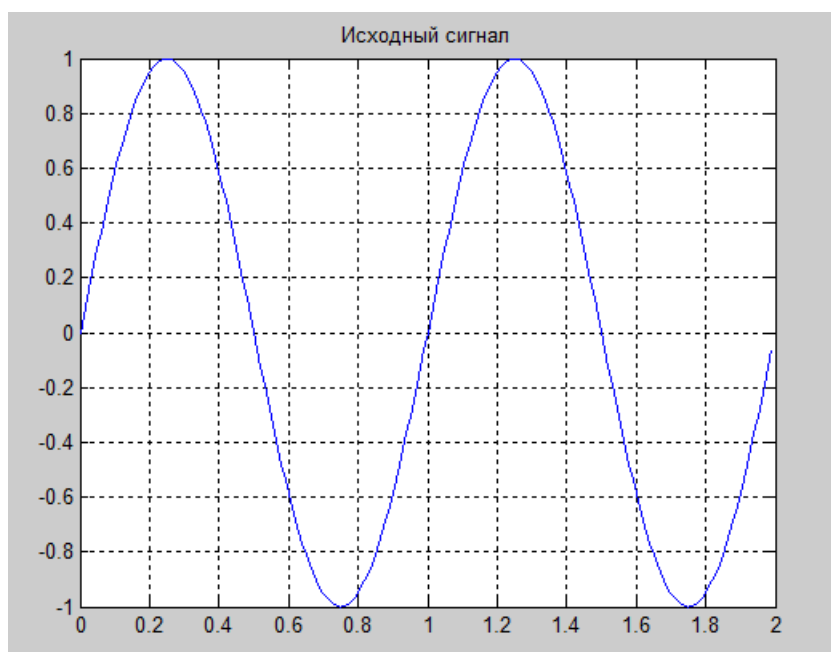


рис.17. Исходный сигнал

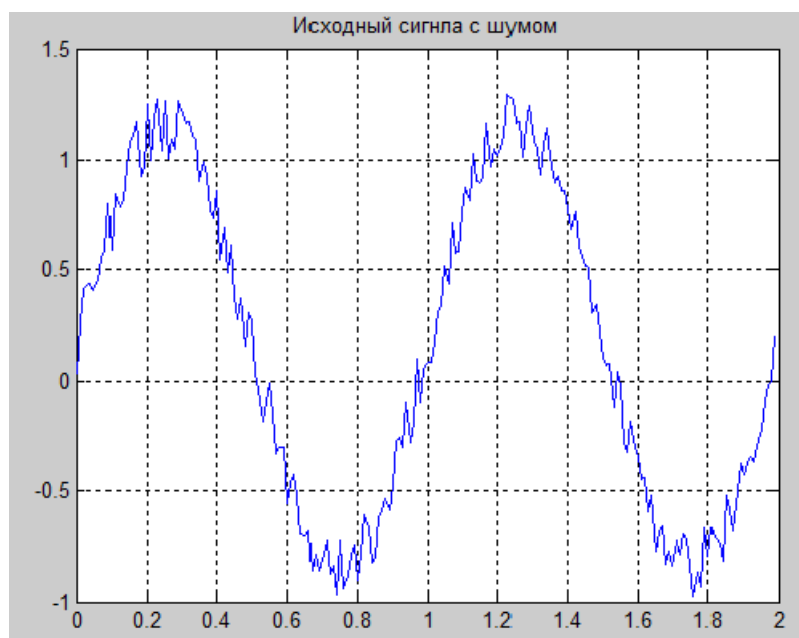


рис.18. Зашумленный сигнал

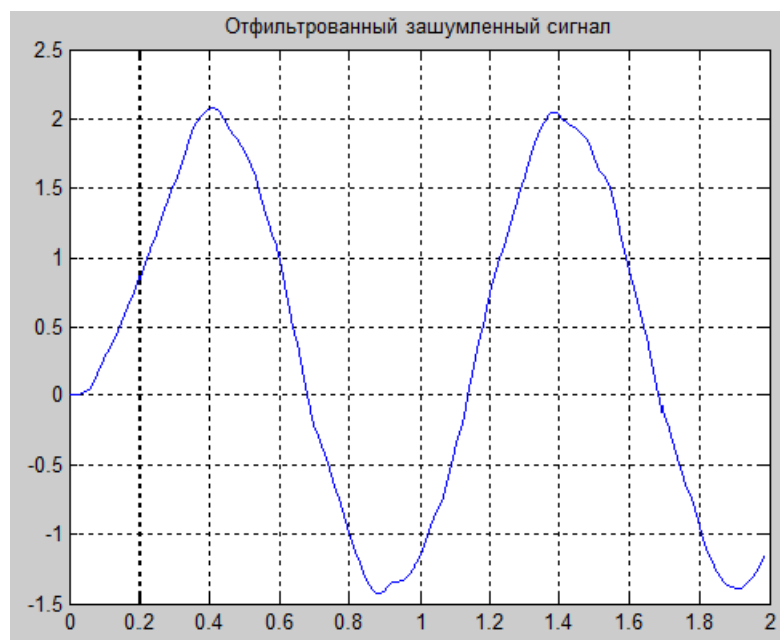


рис.19. Отфильтрованный сигнал

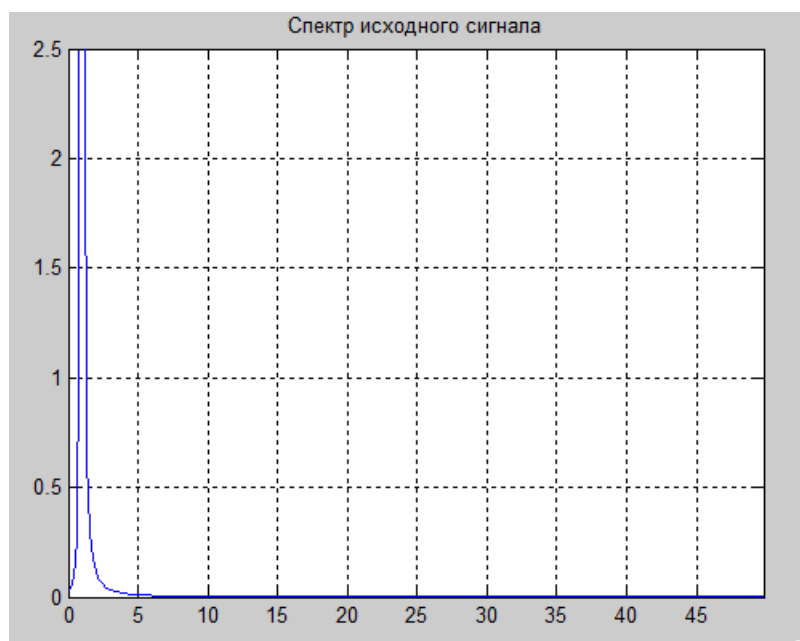


рис.20. Спектр исходного сигнала

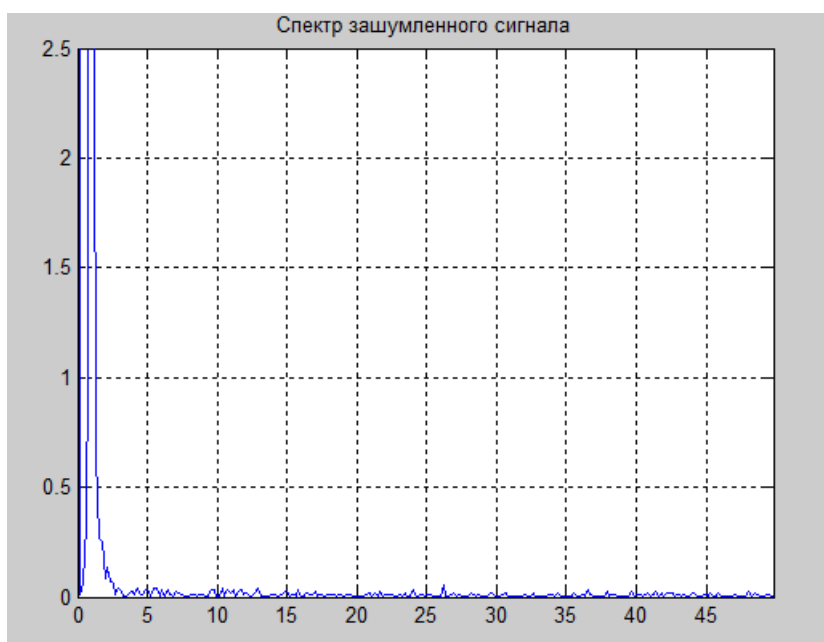


рис.21. Спектр зашумленного сигнала

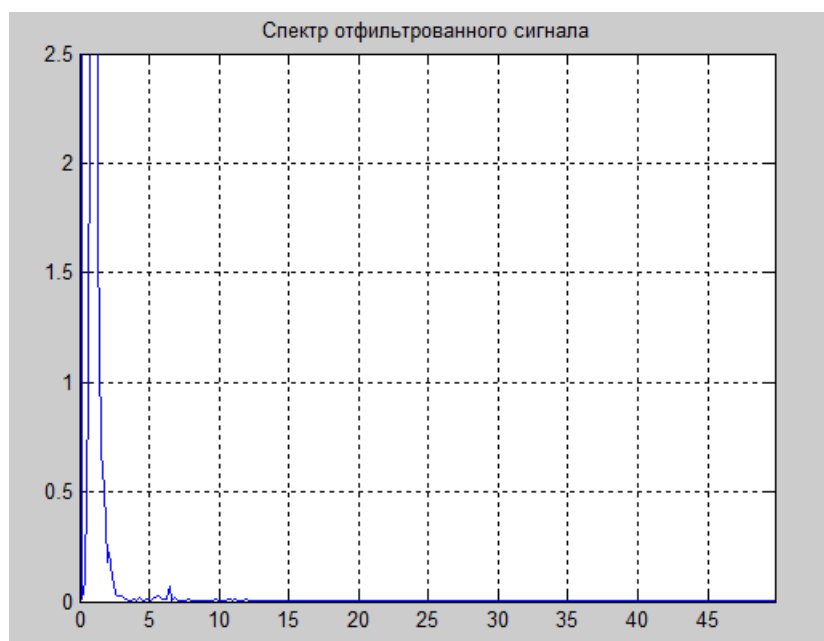


рис.21. Спектр отфильтрованного сигнала

3.4 Вывод

В данной лабораторной работе был синтезирован фильтра Баттерворта 16 порядка и была исследована фильтрация зашумленного сигнала с помощью данного фильтра. При фильтрации сигнала происходит подавление ненужных частот (частот шума), после чего сигнал приобретает форму, близкую к исходному. При этом фильтр полностью не игнорирует все ненужные частоты в сигнале, это связано с наличием переходной зоны между частотами подавления сигнала и частотами пропускания сигнала. Из-за наличия переходной зоны фильтра, он успевает захватить еще несколько частот. Для абсолютного подавления частот необходимо иметь идеальный фильтр, который не имеет переходной зоны между частотами подавления и пропускания. В результате использования такого фильтра будет происходить умножение исходного сигнала на прямоугольную функцию в частотной области.

Глава 4

Лабораторная работа №7. Аналоговая модуляция

4.1 Алгоритм работы

1. Сгенерировать однотоальный сигнал низкой частоты.
2. Выполнить амплитудную модуляцию сигнала по закону

$$u(t) = (1 + MU_m \cos(\Omega t)) \cos(\omega_0 t + \phi_0). \quad (4.1)$$

3. Получить спектр модулированного сигнала.
4. Выполнить модуляцию с подавлением несущей

$$u(t) = MU_m \cos(\omega t) \cos(\omega_0 t + \phi_0). \quad (4.2)$$

Получить спектр.

5. Выполнить однополосную модуляцию:

$$u(t) = U_m \cos(\omega t) \cos(\omega_0 t + \phi_0) + \frac{U_m}{2} \sum_{n=1}^N M_n (\cos(\omega_0 + \Omega_n)t + \phi_0 + \Phi_n), \quad (4.3)$$

положив $n = 1$.

6. Выполнить синхронное детектирование и получить исходный однополосный сигнал.
7. Рассчитать КПД модуляции

$$\eta_A M = \frac{U_m^2 M^2 / 4}{P_U} = \frac{M^2}{M^2 + 2}. \quad (4.4)$$

4.2 Код MATLAB

```
function seti7()
close all;
x = 0:0.01:4*pi;
f0 = 0.5;

    y = sin(2*pi*f0*x);
figure(1)
plot(x(1:1000),y(1:1000), 'LineWidth', 1)
title('Однотональный сигнал низкой частоты')
grid

    spectrum = fft(y, 1024);
normspectrum = spectrum.*conj(spectrum)/1024;
f = 100*(0:255)/1024;
figure(2)
plot(f, normspectrum(1:256), 'LineWidth', 1)
axis([0 max(f) 0 150])
title('Спектр исходного сигнала')
grid

    Fc = 10*f0;
Fs = 100*f0;
U = ammod(y, Fc, Fs, 0, 1);
figure(3)
plot(x(1:1000), U(1:1000), 'LineWidth', 1)
title('Модулированный сигнал')
grid

    uspectrum = fft(U, 1024);
normuspectrum = uspectrum.*conj(uspectrum)/1024;
figure(4)
plot(f, normuspectrum(1:256))
axis([0 max(f) 0 70])
title('Спектр модулированного сигнала')
grid

    Fc = 10*f0;
Fs = 100*f0;
U = ammod(y, Fc, Fs);
figure(5)
plot(x(1:1000), U(1:1000), 'LineWidth', 1)
title('Модуляция с подавлением несущей')
grid

    uspectrum = fft(U, 1024);
normuspectrum = uspectrum.*conj(uspectrum)/1024;
figure(6)
plot(f, normuspectrum(1:256))
```



```

axis([0 max(f) 0 70])
title('Спектр модулированного сигнала с подвалением несущей')
grid

Fc = 10*f0;
Fs = 100*f0;
U = ssbmod(y, Fc, Fs, [], 'upper');
figure(7)
plot(x(1:500), U(1:500), 'LineWidth', 1)
title('Однополосая модуляция')
grid

    uspectrum = fft(U, 1024);
    normuspectrum = uspectrum.*conj(uspectrum)/1024;
    figure(8)
    plot(f, normuspectrum(1:256))
    axis([0 max(f) 0 70])
    title('Спектр однополосной модуляции')
    grid

    [b, a] = butter(10, Fc*2/Fs);
    z = ssbdemod(U, Fc, Fs, 0, b, a);
    figure(9)
    plot(x(1:1000), z(1:1000), 'LineWidth', 1)
    title('Синхронное детектирование')
    grid

    duspectrum = fft(U, 1024);
    normduspectrum = duspectrum.*conj(duspectrum)/1024;
    figure(10)
    plot(f, normduspectrum(1:256))
    axis([0 max(f) 0 70])
    title('Спектр детектированного сигнала')
    grid

    M = 0.5;
    n = M*M/(M*M+2)
end

```

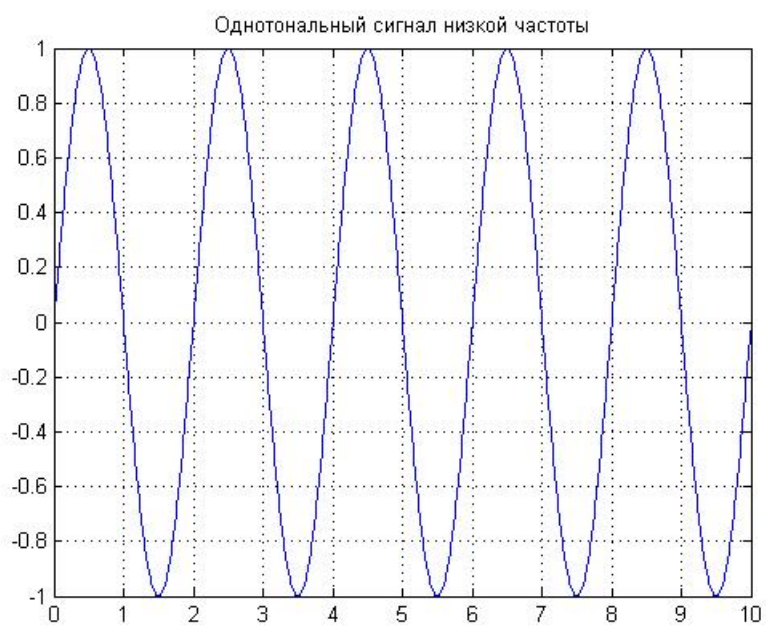


рис. Исходный сигнал

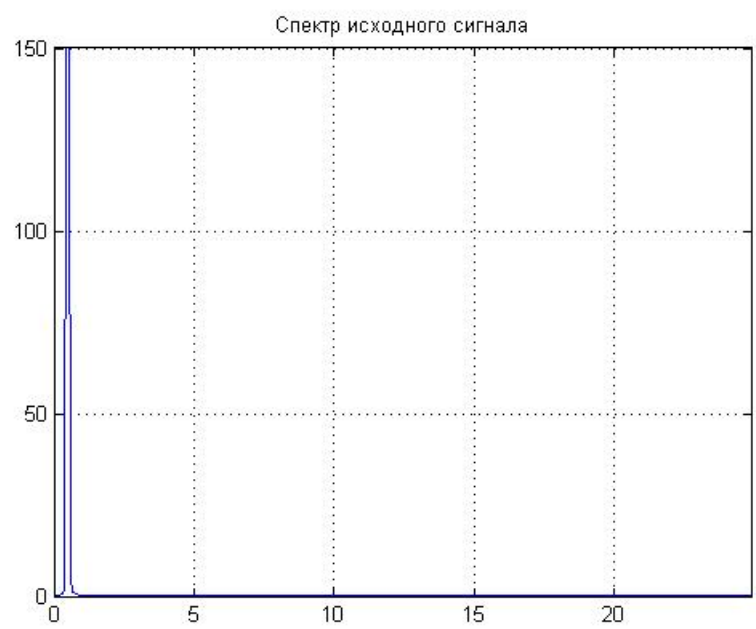


рис. спектр исходного сигнала

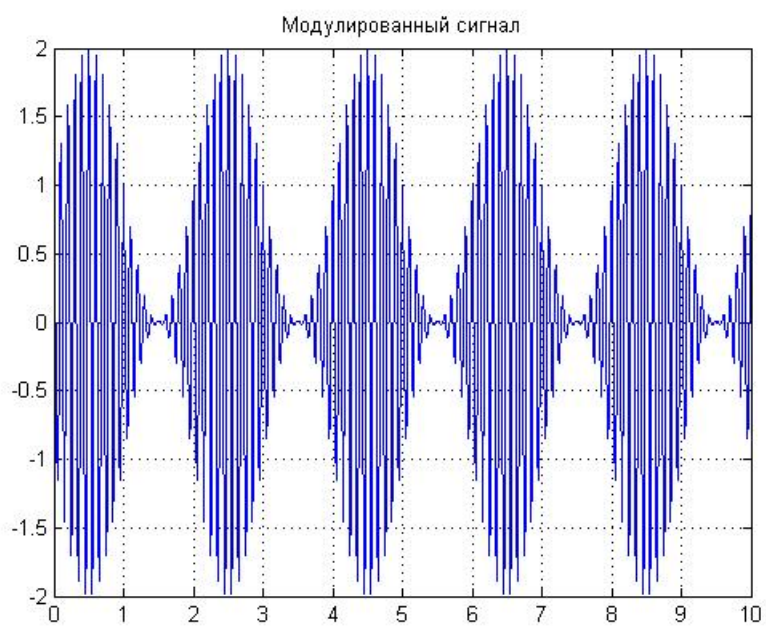


рис. Модулированный сигнал

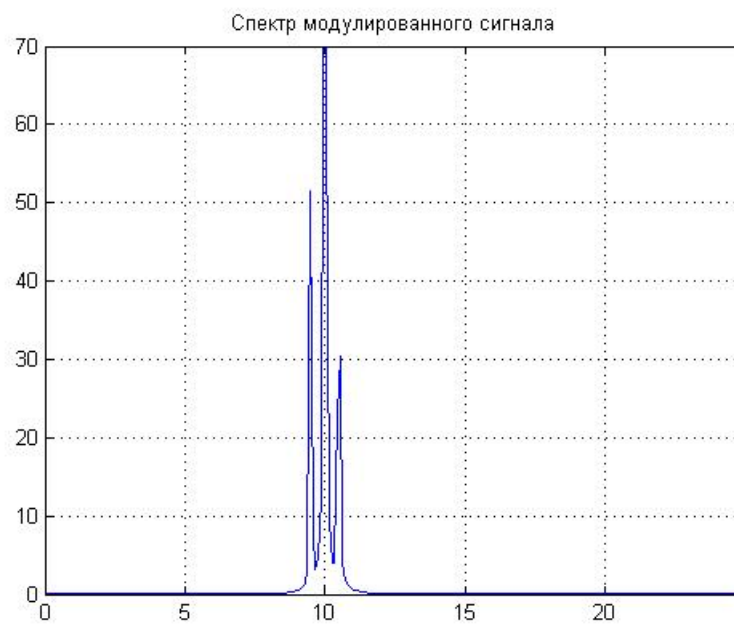


рис. Спектр модулированного сигнала

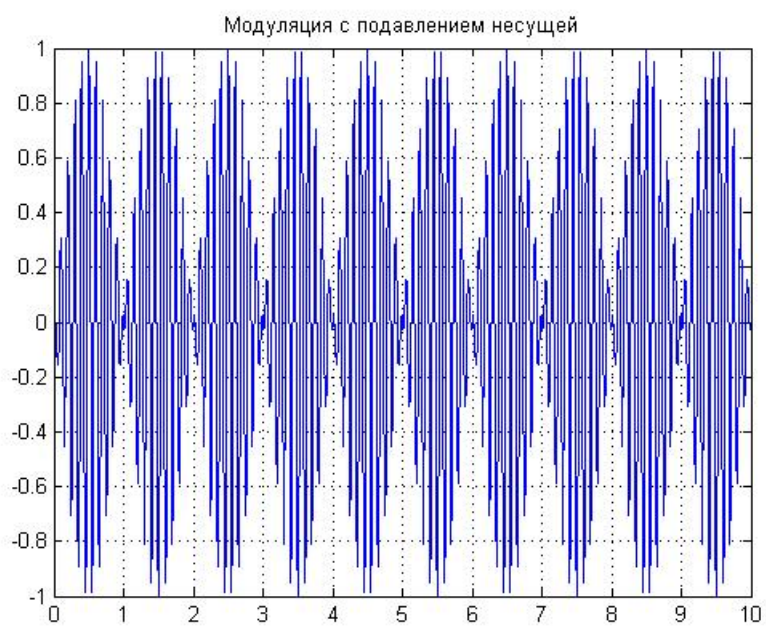


рис. Сигнал с подавлением несущей



рис. Спектр сигнала с подавлением несущей

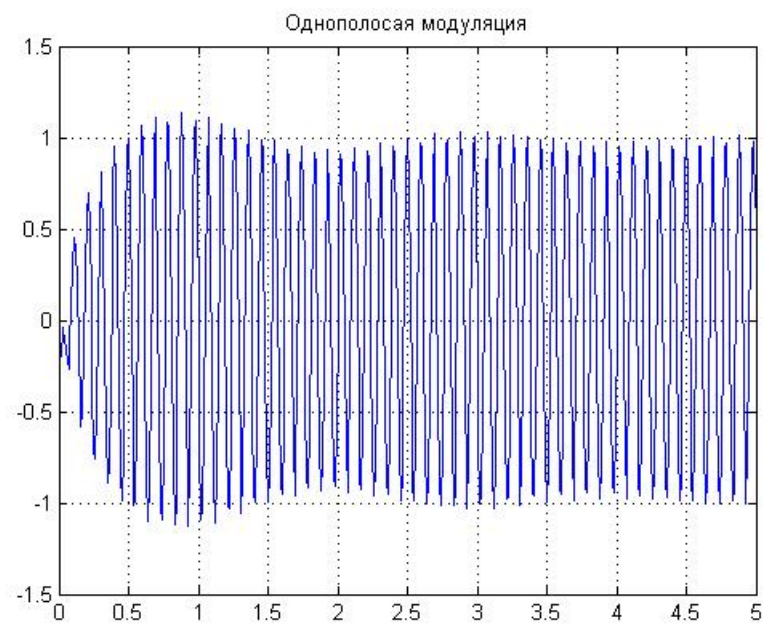


рис. Однополосный сигнал

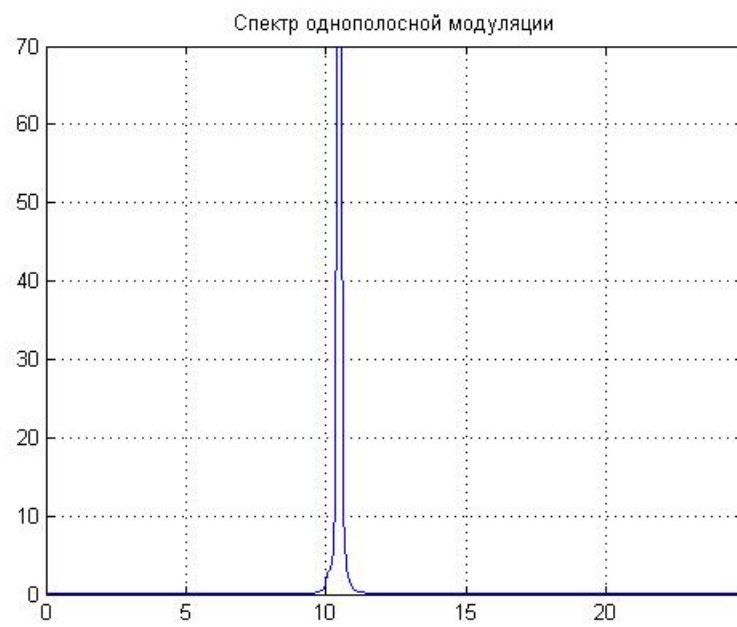


рис. Спектр однополосного сигнала

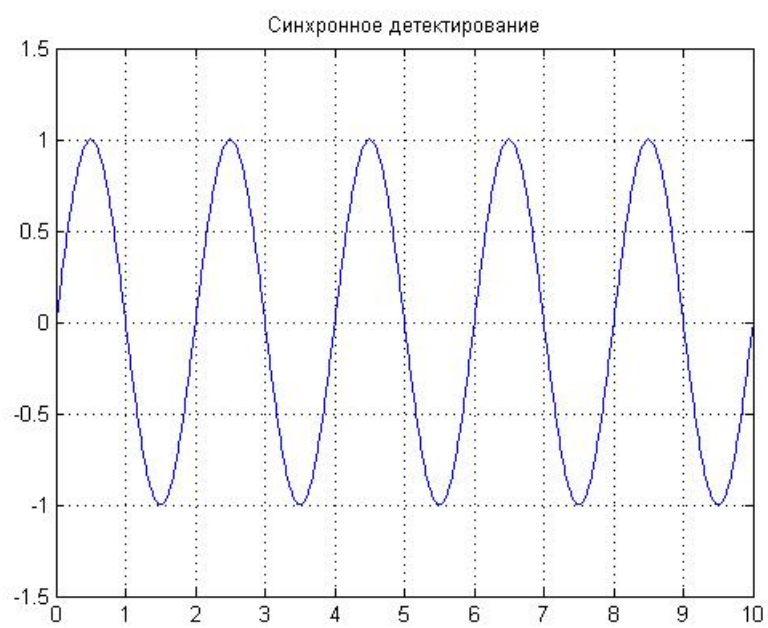


рис.детектированный сигнал

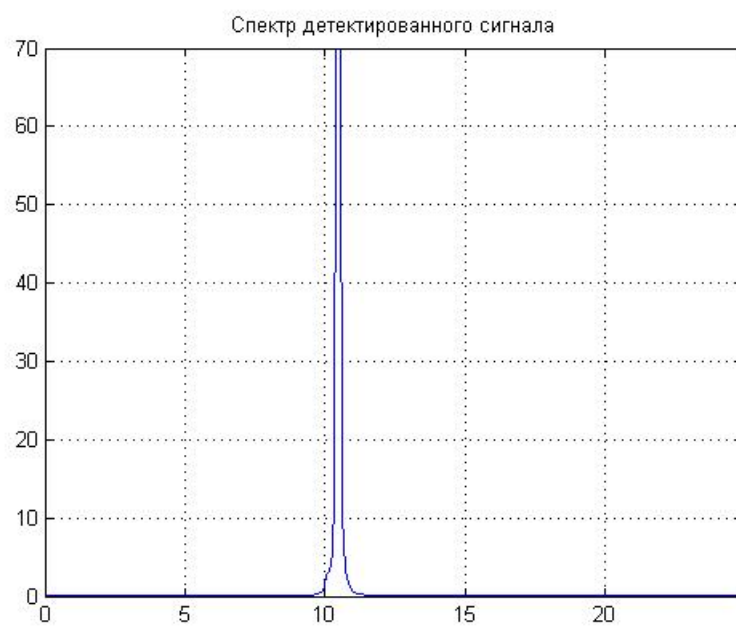


рис. Спектр детектированного сигнала

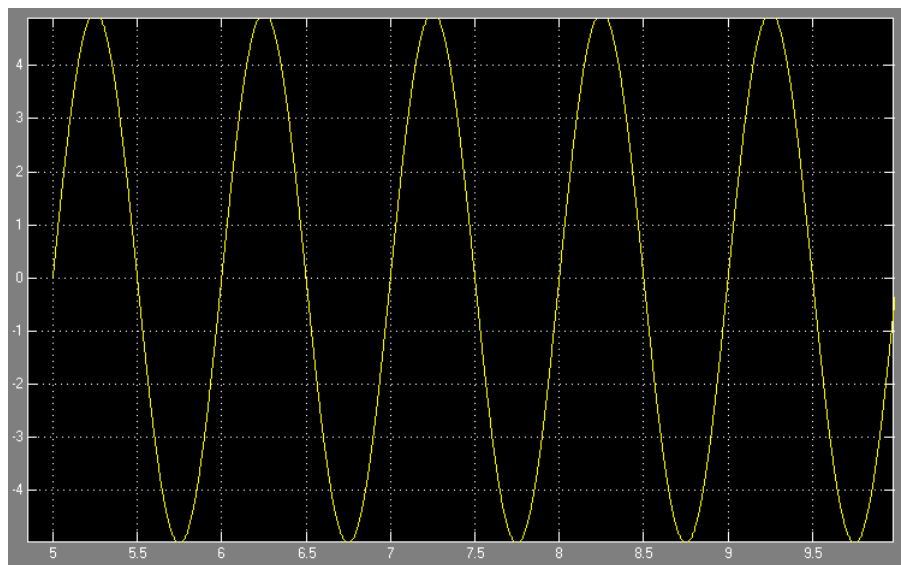


рис. Исходный сигнал. Simulink

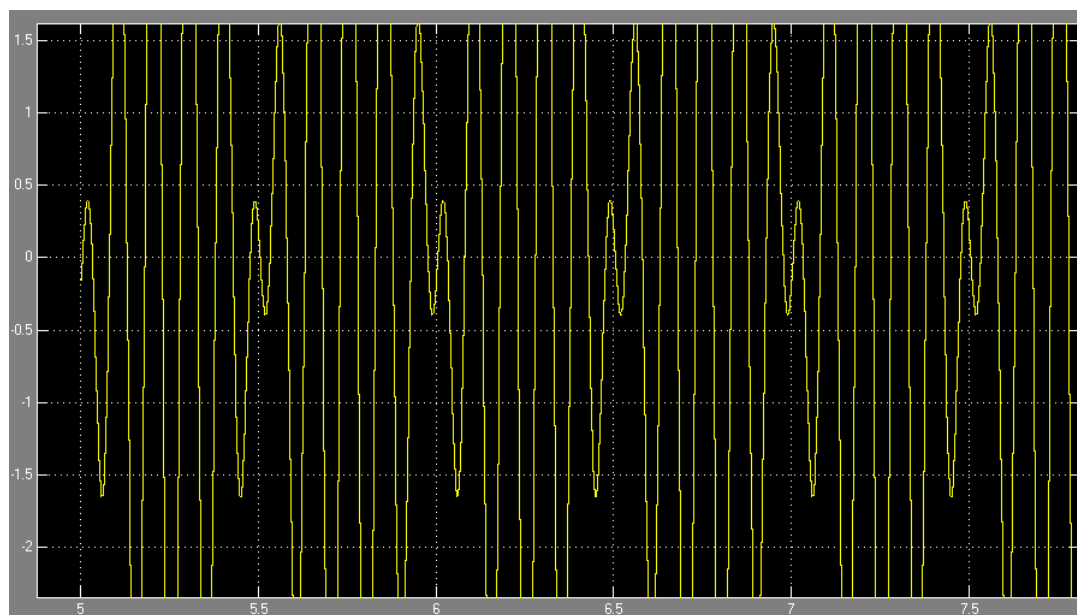


рис. Модуляция Simulink

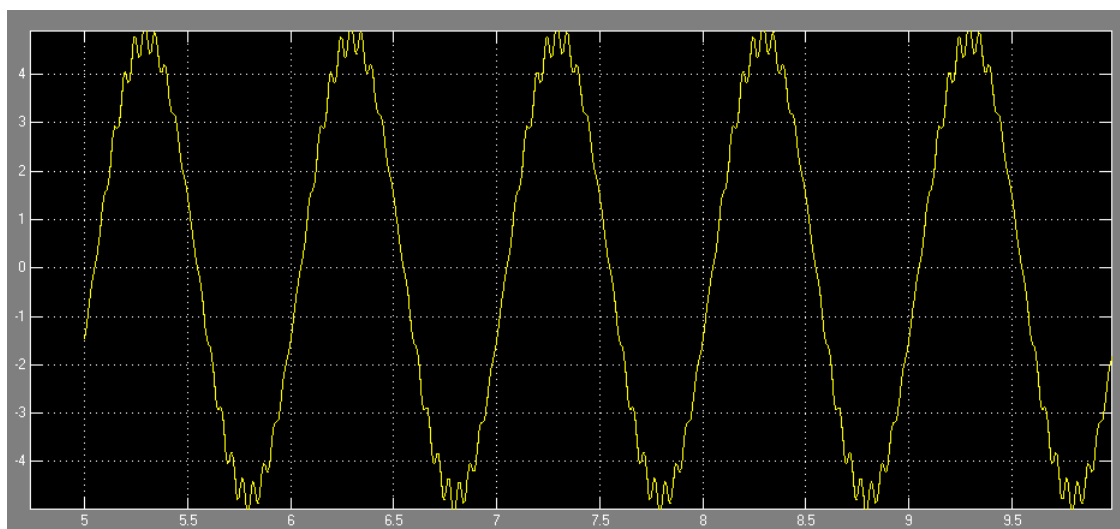


рис. Демодуляция. Simulink

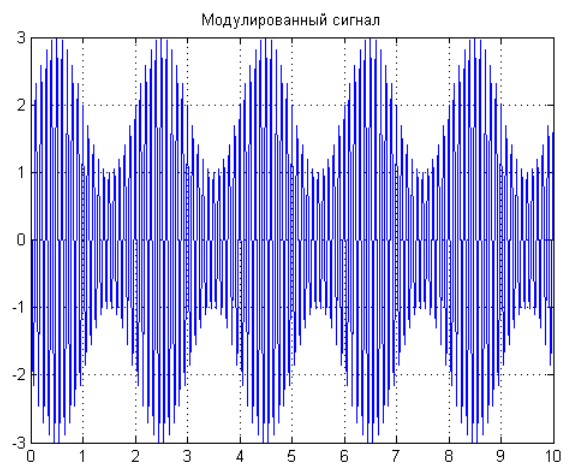


рис. Модулированный сигнал с коэффициентом 2

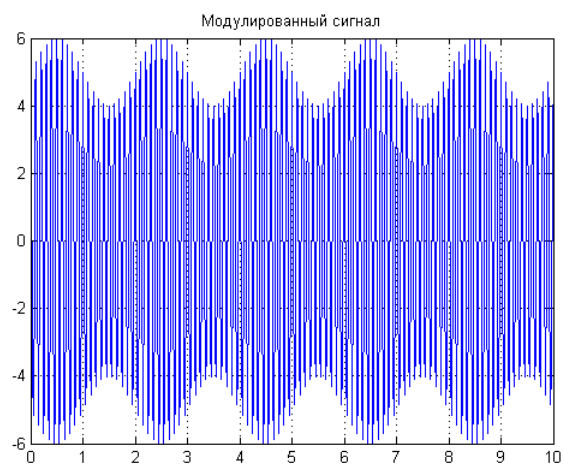


рис. Модулированный сигнал с коэффициентом 5

4.3 Вывод

В данной лабораторной работе была исследована аналоговая модуляция. Был проведен эксперимент по изменению амплитуды сигнала. Амплитудная модуляция применяется на низких частотах, что обусловлено низким КПД использования энергии модулированных сигналов. Основным применением амплитудной модуляции является передача сигналов звуковой частоты. Для получения колебаний боковых частот без несущей применяют так называемую балансную модуляцию, основанную на том, что складывают два АМ колебания, у которых колебания боковых частот находятся в фазе, а несущих - в противофазе (или берут разность двух АМ колебаний, у которых боковые частоты в противофазе, а несущие - в фазе). КПД такой модуляции становится равно 100 процентам. При этом спектр модулированного сигнала представляет собой три импульса, две из которых симметрично расположены относительно средней. Средний импульс обозначает спектр несущего сигнала. В ходе работы была произведена модуляция с подавлением несущей. При этом происходит игнорирование несущего сигнала, а источником информации являются два боковых сигнала. Спектр такого сигнала представляет собой два вертикальных симметричных импульса, что говорит об отсутствии несущего сигнала. В работе была проведена модуляция однополосного сигнала. Данный сигнал используется для уменьшения мощности потребления энергии для переноса информации и для осуществления переноса информации на более дальние расстояния. Спектры двух боковых полос АМ-сигнала являются зеркальным отражением друг друга, то есть они несут одну и ту же информацию. Поэтому одну из боковых полос можно удалить. Спектр такого сигнала представляет собой один импульс, так как в нем исключены несущий сигнал и половина исходного. Также было рассмотрено синхронное детектирование сигнала (демодуляция сигнала), в результате которого из модулированного сигнала был получен исходный сигнал. Суть детектирования состоит в умножении частоты сигнала на опорное колебание с несущей частотой. Результат умножения содержит два слагаемых: искомая амплитуда и АМ-сигнал с несущей частотой $2\omega_0$, который легко удаляется путем пропускания сигнала через фильтр низких частот. Спектр демодулированного сигнала отличается от первоначального.

Глава 5

Лабораторная работа №8

5.1 Постановка задачи

1. Сгенерировать однотоновый сигнал низкой частоты.
2. Выполнить фазовую модуляцию/демодуляцию сигнала по закону $u(t) = (U_m \cos(\omega t + ks(t)))$, используя встроенную функцию MatLab `pmmod`, `pmdemod`.
3. Получить спектр модулированного сигнала.
4. Выполнить частотную модуляцию/демодуляцию по закону

$$u(t) = U_m \cos(\omega_0 t + k \int_0^t s(t) dt + \phi_0) \quad (5.1)$$

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

5.2 Теоретическая часть

Частотная модуляция — вид аналоговой модуляции, при котором информационный сигнал управляет частотой несущего колебания. По сравнению с амплитудной модуляцией здесь амплитуда остаётся постоянной.

Фазовая модуляция — один из видов модуляции колебаний, при которой фаза несущего колебания управляется информационным сигналом. Фазомодулированный сигнал $s(t)$ имеет следующий вид:

$$s(t) = g(t) \sin[2\pi f_c t + \phi(t)], \quad (5.2)$$

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

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

5.3 Код matlab

```
x = 0:0.01:2;
u = cos(2*pi*x);

figure;
subplot(3,1,1);
plot(x,u);
grid;
am = fmmmod(u,2,30,1);
subplot(3,1,2);
plot(x,am);
grid;
modulatedSpectr = fft(am,512);
normSpectrum = modulatedSpectr.*conj(modulatedSpectr)/512;
f = 100*(-256:255)/512;
subplot(3,1,3);
plot(f,normSpectrum);
grid;
axis([min(f) max(f) 0 max(normSpectrum)]);

figure;
subplot(3,1,1);
plot(x,u);
grid;
am = pmmmod(u,5,1000,pi/2);
subplot(3,1,2);
plot(x,am);
grid;
modulatedSpectr = fft(am,512);
normSpectrum = modulatedSpectr.*conj(modulatedSpectr)/512;
f = 100*(-256:255)/512;
subplot(3,1,3);
plot(f,normSpectrum);
grid;
axis([min(f) max(f) 0 max(normSpectrum)]);
```

5.4 Результаты работы

Выполним демодуляцию, в том числе с помощью блока захвата фазы (фазовой автоподстройки частоты) Phase-Locked Loop.

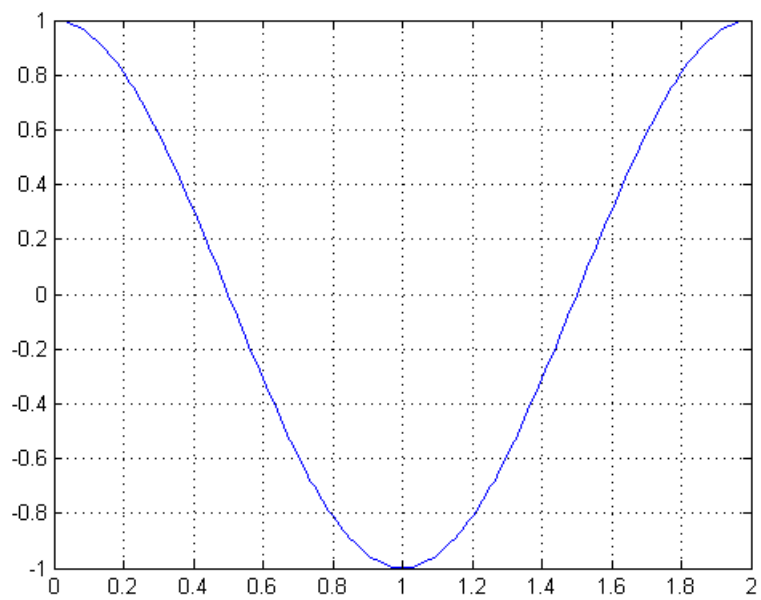


рис. исходный сигнал

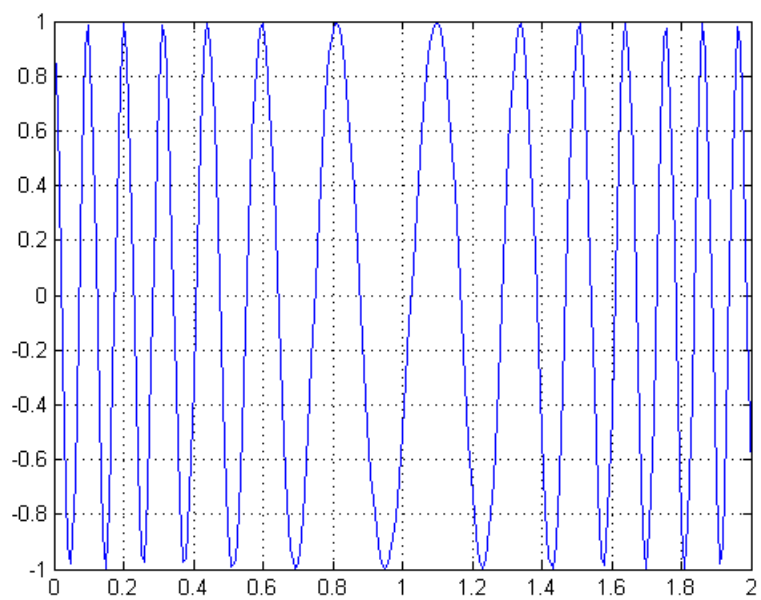


рис. Частотная модуляция исходного сигнала

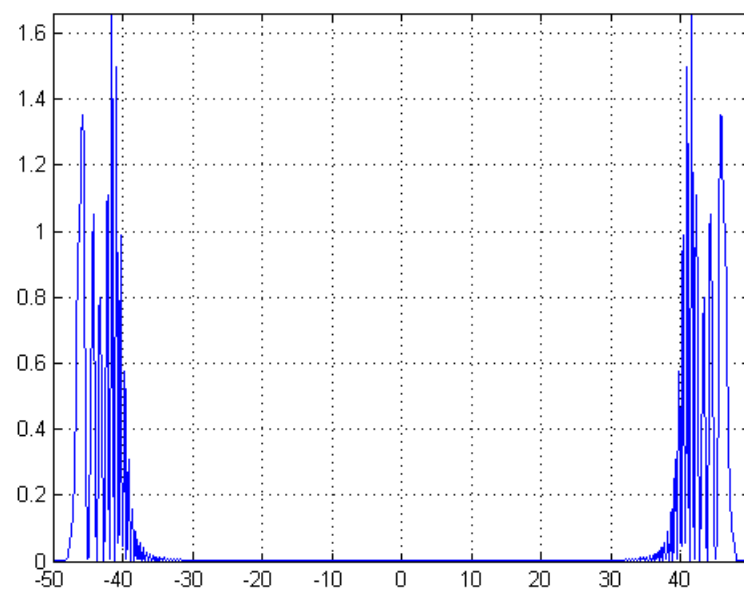


рис. Спектр частотной модуляции

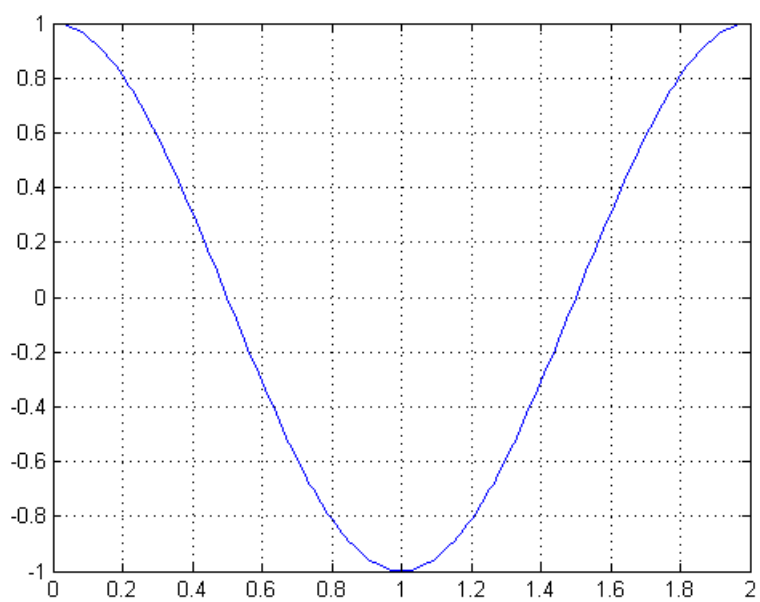


рис. Исходный сигнал

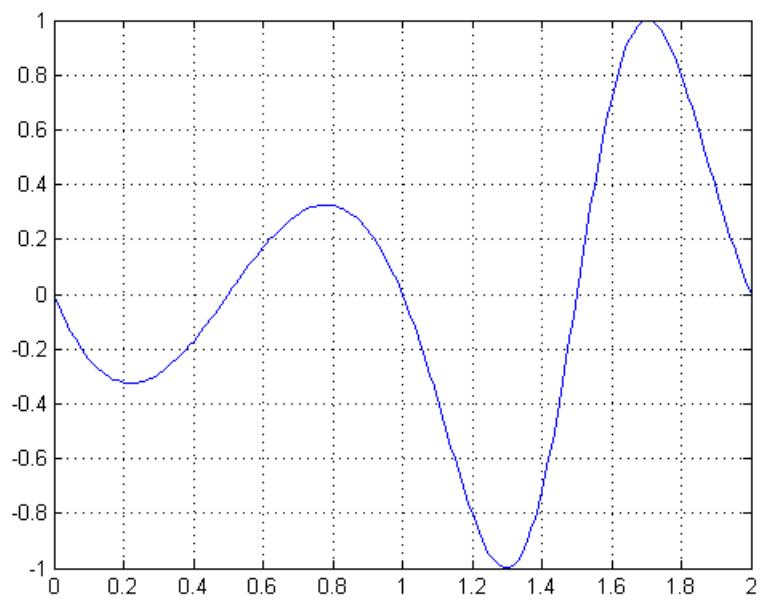


рис. Фазовая модуляция

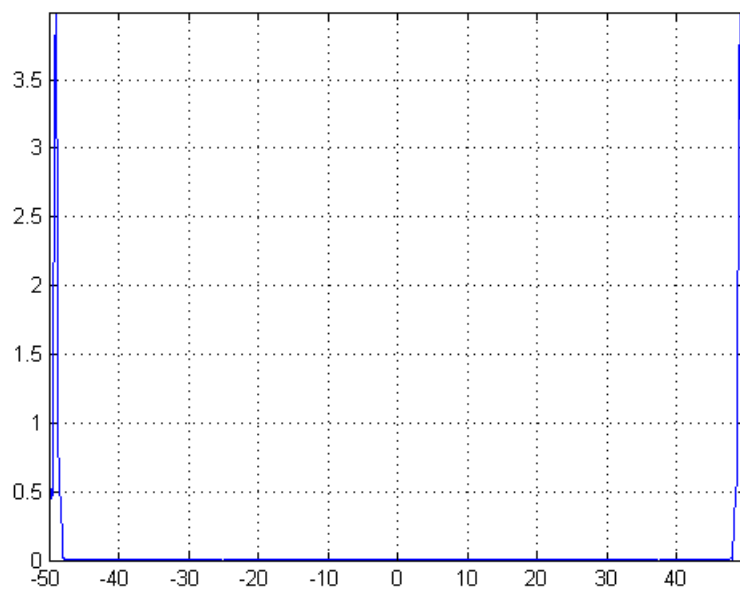


рис. Спектр фазовой модуляции

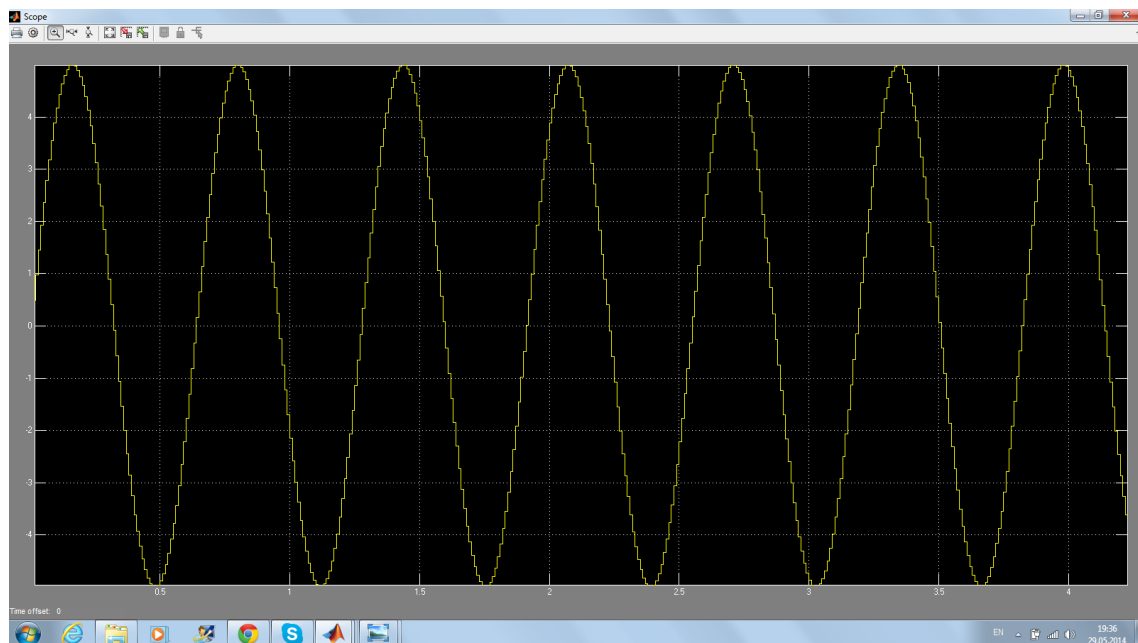


рис. Симулинк исходный сигнал

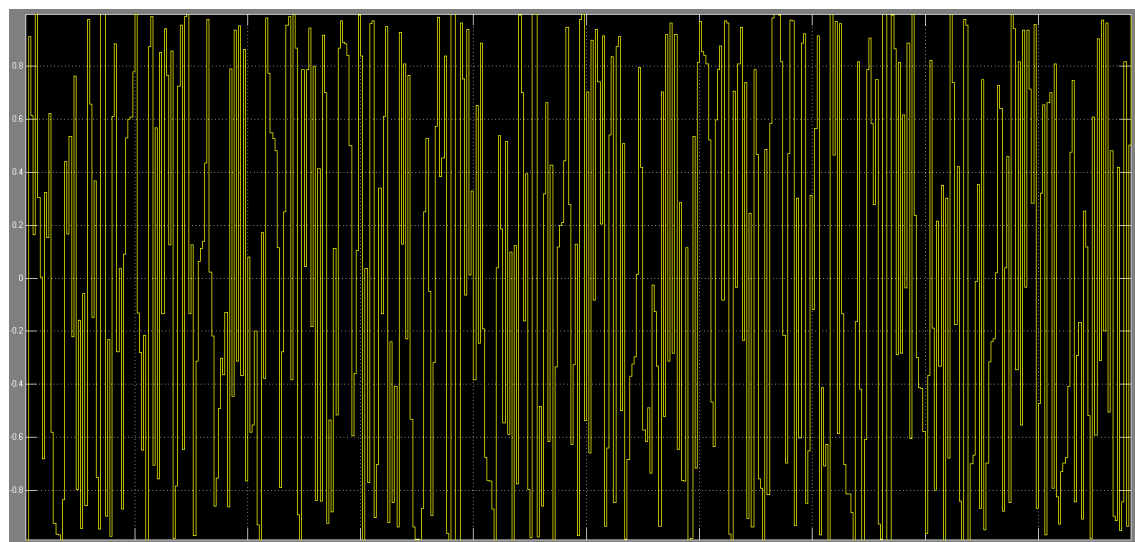


рис .Симулинк частотная модуляция

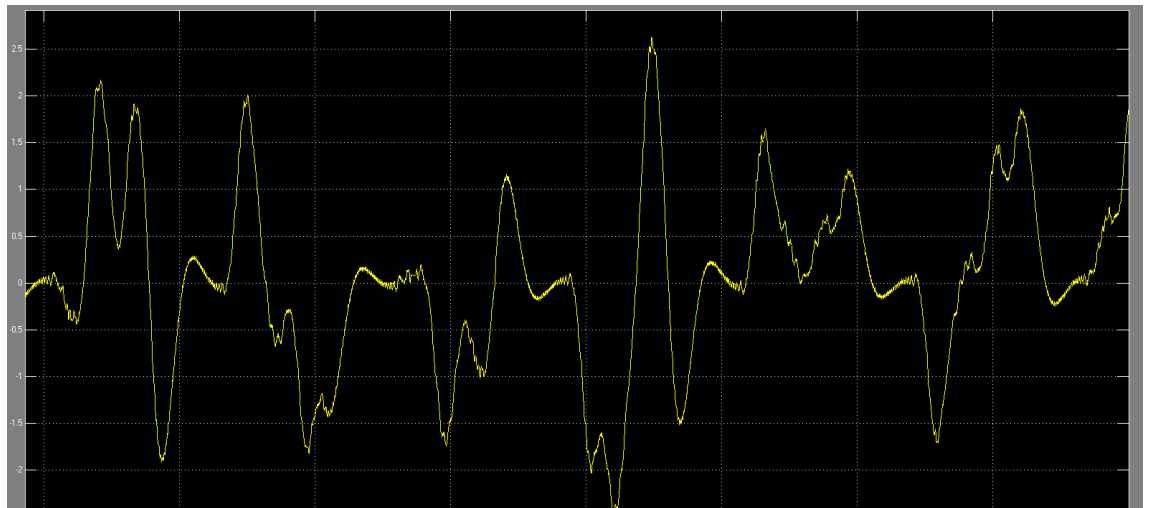


рис. Симулирк сигнал на выходе ФНЧ

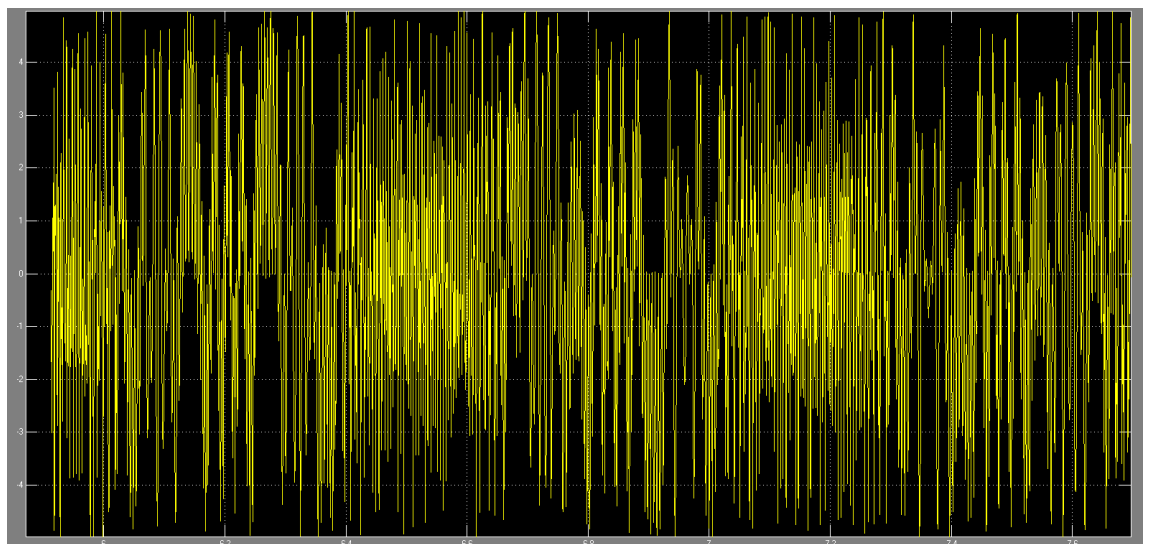


рис. Частотная модуляция PLL

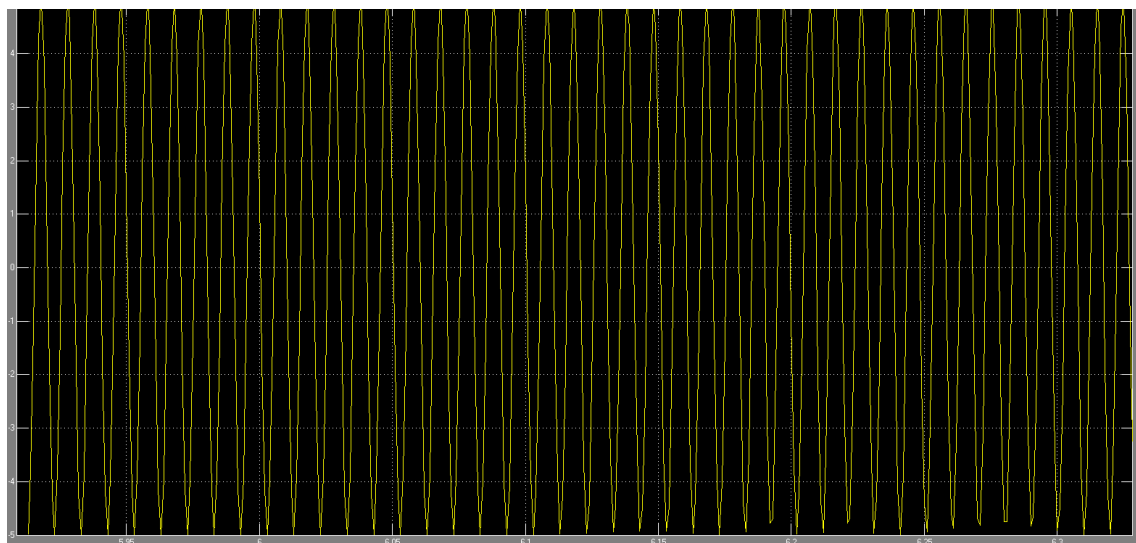


рис. Сигнал на выходе фазового детектора

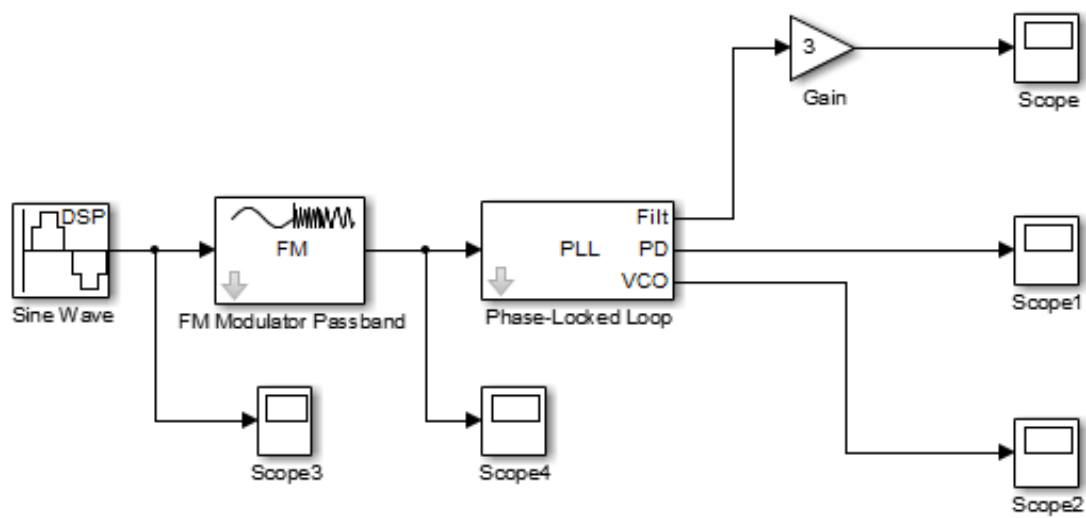


рис. Схема симулинк

5.5 Вывод

В данной лабораторной работе было проведено исследование частотной и фазовой модуляций/демодуляций. Частотная модуляция представляет собой изменение частоты исходного сигнала при постоянной фазе и амплитуде. При этом происходит изменением длины волн сигнала на разных временных промежутках. Фазовая модуляция представляет собой изменение исходного сигнала по фазе при постоянной амплитуде и постоянной частоте. При фазовой модуляции происходит изменение формы сигнала на отдельных промежутках. Фазовая и частотная модуляции позволяют производить передачу высокочастотного звукового сигнала. Спектр частотной модуляции представляет собой несколько вертикальных импульсов расположенных относительно центра спектральной прямой, это говорит о наличии нескольких гармоник в сигнале. Спектр фазовой модуляции представляет собой также несколько импульсов расположенных симметрично относительно центра спектральной прямой. Различие между фазовой и частотной модуляцией обнаруживается при модуляции спектром частот. При фазовой модуляции индекс модуляции (отношение девиации фазы к фаземодулирующего сигнала) не зависит от частоты модуляции, а девиация частоты (амплитуда изменения частоты) пропорциональна частоте модуляции. При частотной модуляции девиация частоты не зависит от частоты модуляции, а индекс модуляции (отношение девиации частоты к частоте модулирующего сигнала) обратно пропорционален частоте модуляции. Частотную и фазовую модуляцию объединяют под общим названием угловой модуляции (УМ). По форме колебаний с угловой модуляцией невозможно определить, к какому виду модуляции относится данное колебание, к ФМ или ЧМ, а при достаточно гладких функциях формы сигналов ФМ и ЧМ вообще практически не отличаются. Также в лабораторной работе произведена частотная модуляция с использованием устройства фазовой автоподстройки частоты. Данное устройство представляет собой систему автоматического регулирования, подстраивающая фазу управляемого генератора так, чтобы она была равна фазе опорного сигнала, либо отличалась на известную функцию от времени. Регулировка осуществляется благодаря наличию отрицательной обратной связи. Данный прибор используется для частотной модуляции и демодуляции, преобразования частоты и частотной фильтрации.

Глава 6

Лабораторная работа №9

6.1 Постановка задачи

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

6.2 Теоретическая часть

Сущность цифровой модуляции заключается в том, что передаваемый непрерывный сигнал дискретизируется во времени, квантуется по уровню и полученные отчеты, следующие в дискретные моменты времени, преобразуются в кодовые комбинации. Полученной последовательностью кодовых видеосигналов модулируется высокочастотный сигнал-переносчик. Существует 3 основных вида манипуляции сигналами: амплитудная (Amplitude-shift keying (ASK)), частотная (Frequency-shift keying (FSK)) и фазовая (Phase-shift keying (PSK)). Этот набор манипуляций определяется основными характеристиками, которыми обладает любой сигнал. Для моделирования модуляции цифрового сигнала в ходе лабораторной работы предлагается использовать следующий набор функций:

1. Функция `randerr` предназначена для формирования ошибок в цифровом сигнале. Она дает матрицу, в каждой строке которой имеется заданное число случайно расположенных ненулевых элементов.
2. 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.

3. 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.
1. Амплитудная манипуляция (ASK) — изменение сигнала, при котором скачкообразно меняется амплитуда несущего колебания. АМн можно рассматривать как частный случай квадратурной манипуляции.
2. При частотной манипуляции (FSK) значениям «0» и «1» информационной последовательности соответствуют определённые частоты синусоидального сигнала при неизменной амплитуде. Частотная манипуляция весьма помехоустойчива. Однако при частотной манипуляции неэкономно расходуется ресурс полосы частот телефонного канала. Поэтому этот вид модуляции применяется в низкоскоростных протоколах, позволяющих осуществлять связь по каналам с низким отношением сигнал/шум.
3. Минимальная частотная модуляция (MSK) представляет собой способ модуляции, при котором не происходит скачков фазы и изменение частоты происходит в моменты пересечения несущей нулевого уровня. MSK уникальна потому, что значение частот соответствующих логическим «0» и «1» отличаются на величину равную половине скорости передачи данных.
4. Фазовая манипуляция (PSK) — один из видов фазовой модуляции, при которой фаза несущего колебания меняется скачкообразно в зависимости от информационного сообщения.
5. Квадратурной амплитудной манипуляцией (QASK) называется манипуляция, при которой изменяется как фаза, так и амплитуда сигнала, что позволяет увеличить количество информации, передаваемой одним состоянием (отсчётом) сигнала.

6.3 Код matlab

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

```

scatterplot(demodSignal);
eyediagram(demodSignal,10);

%PSK modulation
h = modem.pskmod('M', 8);
g = modem.pskdemod('M', 8);
msg = randint(10,1,8);
modSignal = modulate(h,msg);
errSignal = (randerr(1,10, 3) ./ 30)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);
eyediagram(modSignal,10);
scatterplot(demodSignal);
eyediagram(demodSignal,10);

%QPSK modulation
h = modem.oqpskmod;
g = modem.oqpskdemod;
msg = randint(200,1,4);
modSignal = modulate(h,msg);
errSignal = (randerr(1,400, 100) ./ 30)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);
eyediagram(modSignal,10);
scatterplot(demodSignal);
eyediagram(demodSignal,10);

%GENQAM modulation
M = 5;
h = modem.genqammod('Constellation', exp(1i*2*pi*[0:M-1]/M));
g = modem.genqamdmod('Constellation', exp(1i*2*pi*[0:M-1]/M));
msg = randint(10,1,8);
modSignal = modulate(h,msg);
errSignal = (randerr(1,10, 3) ./ 30)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);
eyediagram(modSignal,10);
scatterplot(demodSignal);
eyediagram(demodSignal,10);

%MSK modulation
h = modem.mskmod('SamplesPerSymbol', 10);
g = modem.msksdemod('SamplesPerSymbol', 10);
msg = randint(10,1,2);
modSignal = modulate(h, msg);
errSignal = (randerr(1,100, 3) ./ 30)';
modSignal = modSignal + errSignal;

```

```
demodSignal = demodulate(g, modSignal);  
scatterplot(modSignal);  
eyediagram(modSignal,10);  
scatterplot(demodSignal);  
eyediagram(demodSignal,10);
```

6.4 Результаты работы

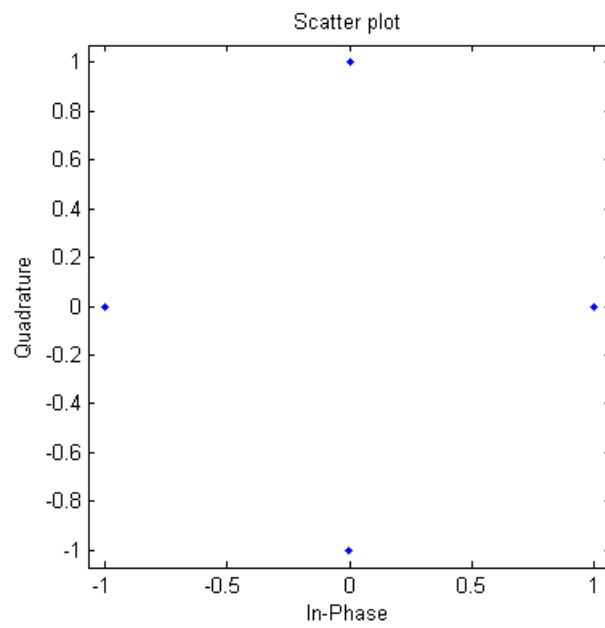


рис. Сигнальное созвездие BPSK-модуляции

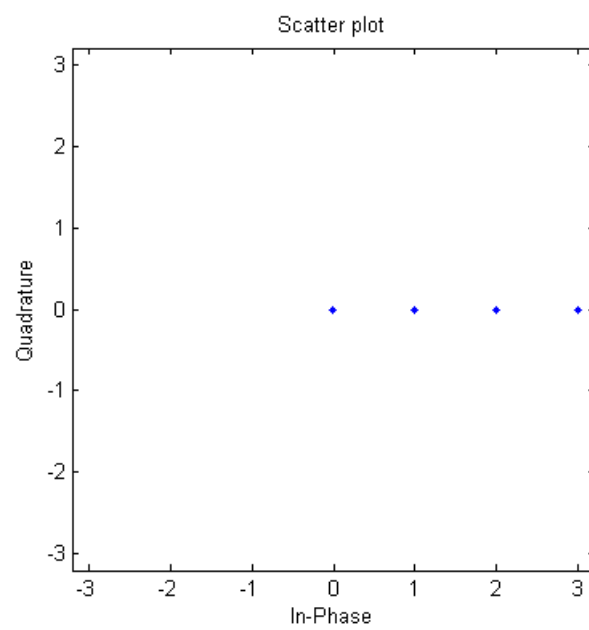


рис. Сигнальное созвездие BPSK-демодуляции

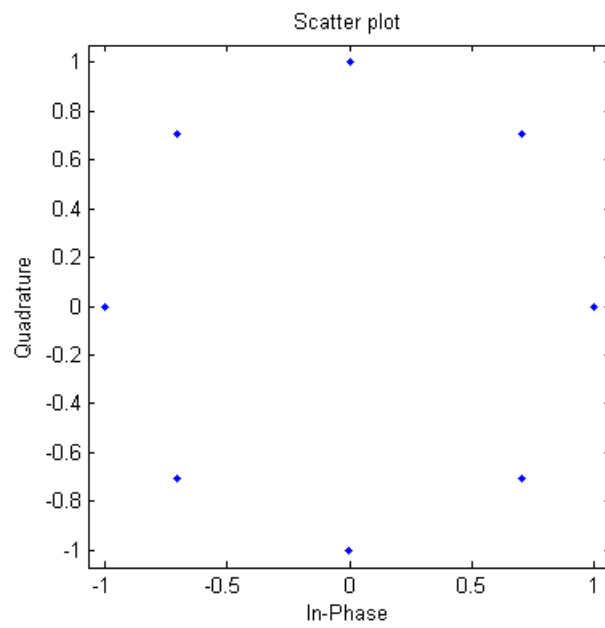


рис. Сигнальное созвездие PSK-модуляции

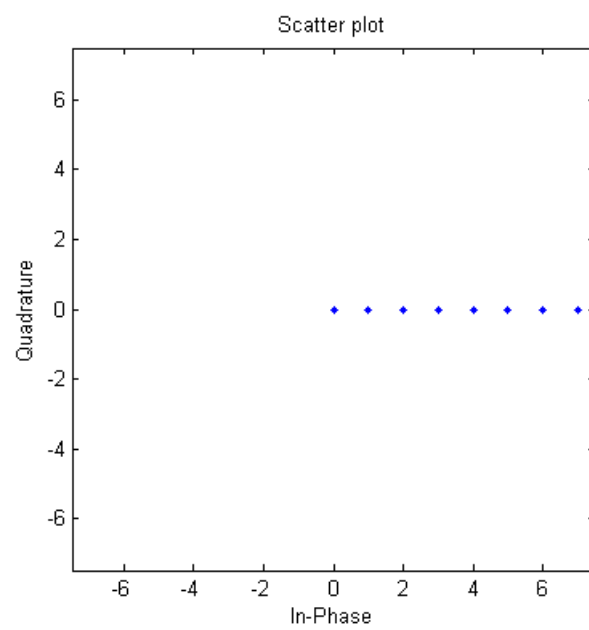


рис. Сигнальное созвездие PSK-демодуляции

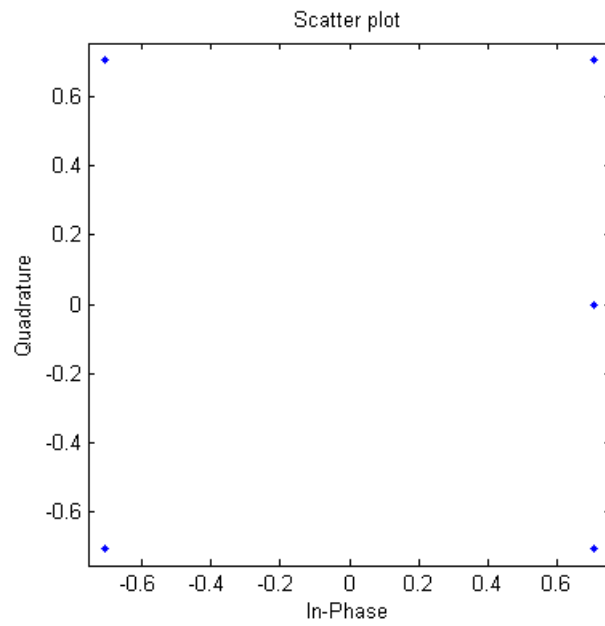


рис. Сигнальное созвездие OQPSK-модуляции

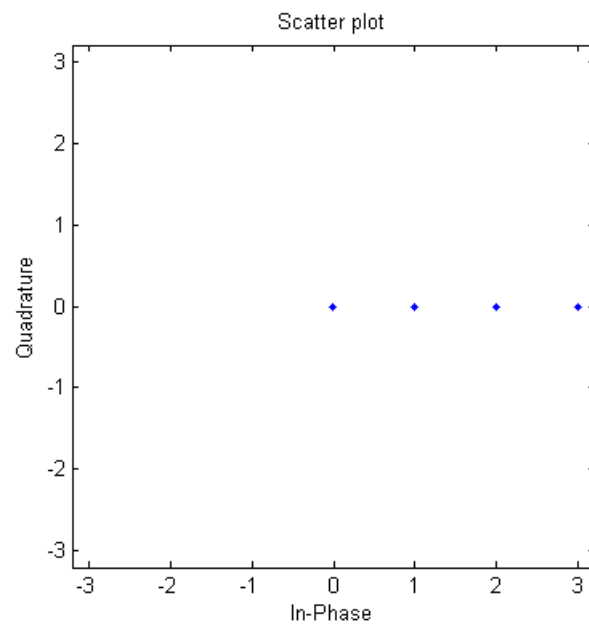


рис. Сигнальное созвездие OQPSK-демодуляции

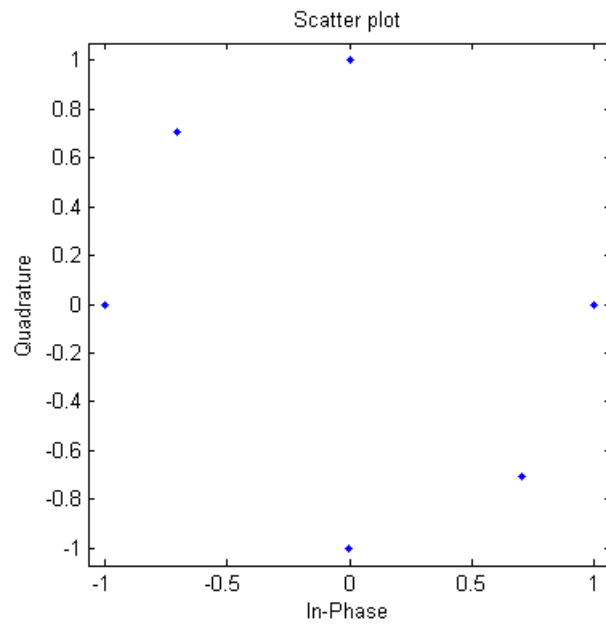


рис. Сигнальное созвездие GENQAM-модуляции

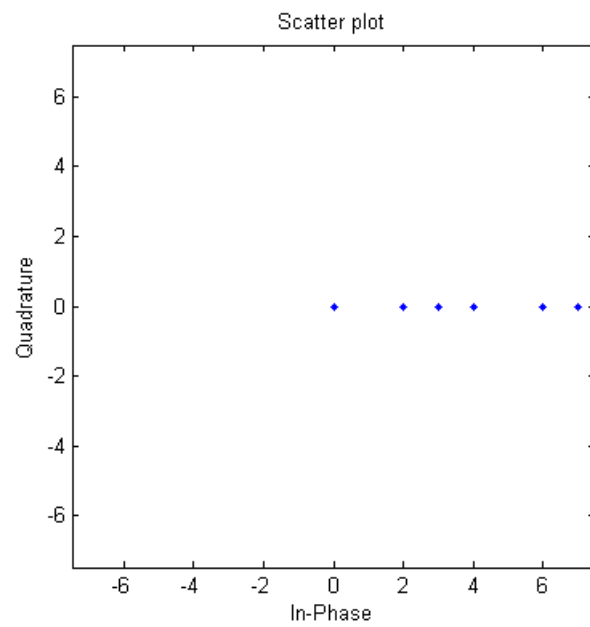


рис. Сигнальное созвездие GENQAM-демодуляции

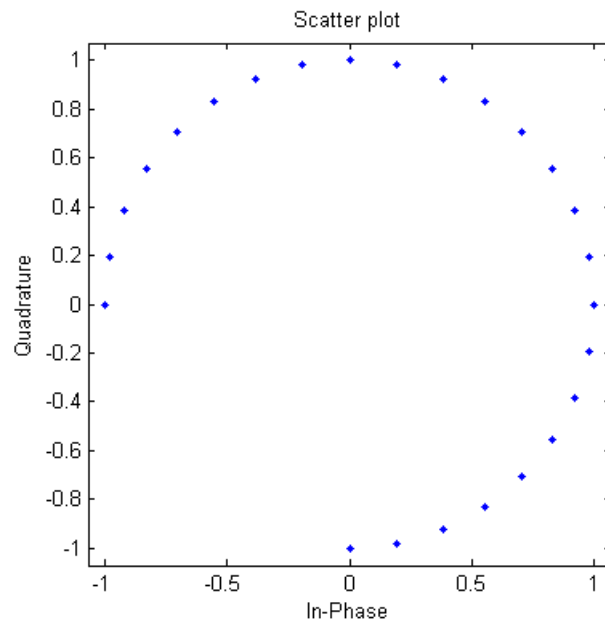


рис. Сигнальное созвездие MSK-модуляции

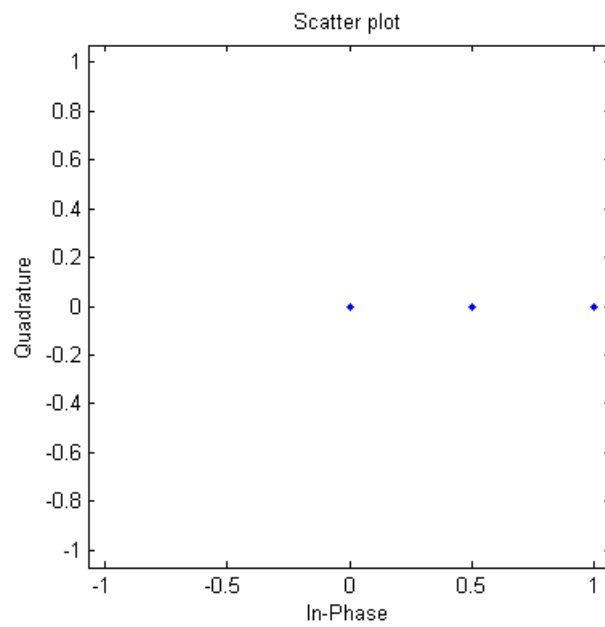


рис. Сигнальное созвездие MSK-демодуляции

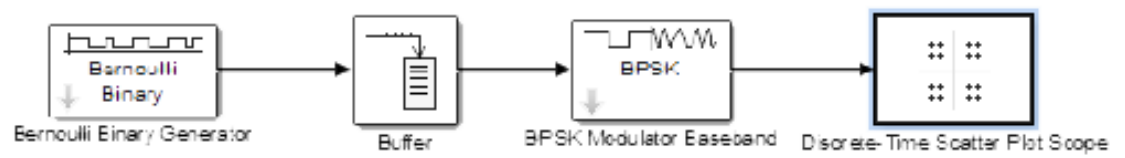


рис. 85. Модель BPSK-модуляции

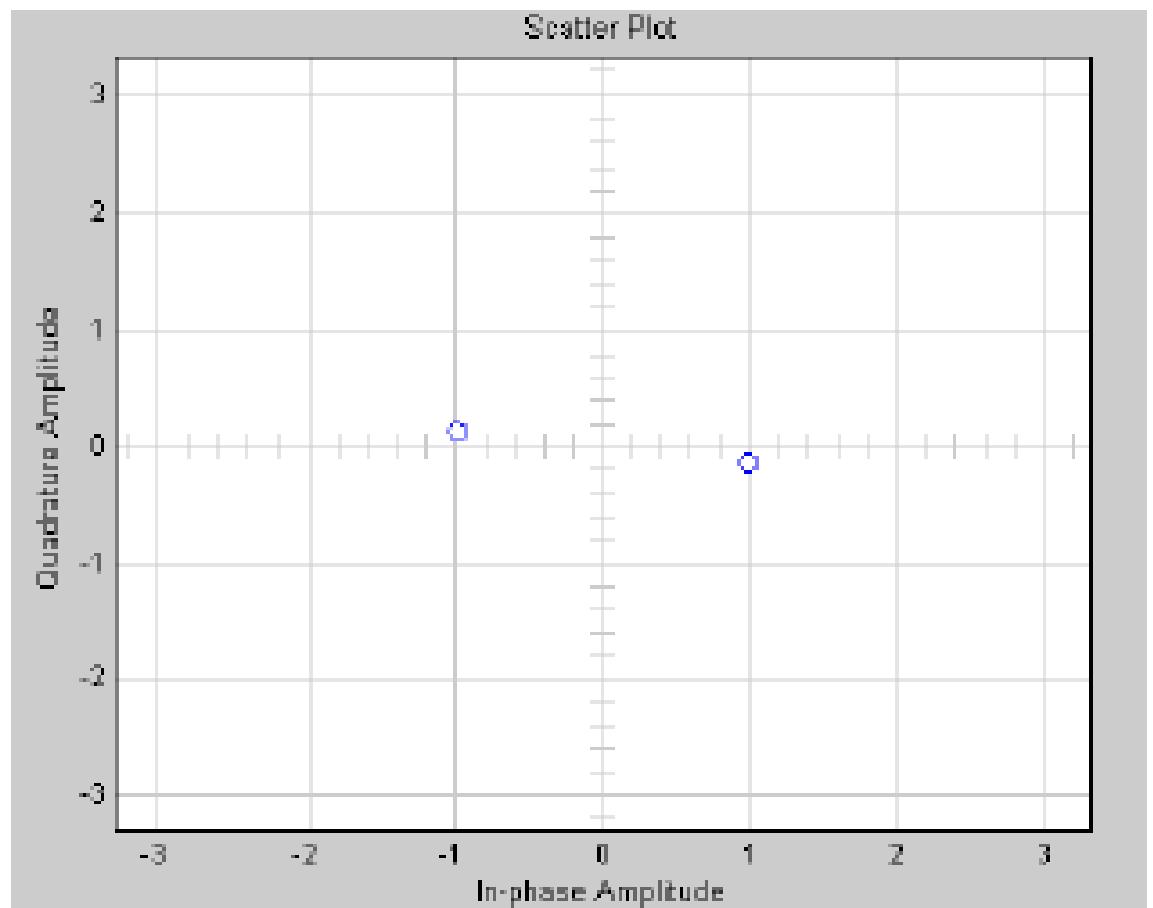


рис. 86. Сигнальное созвездие BPSK-модуляции

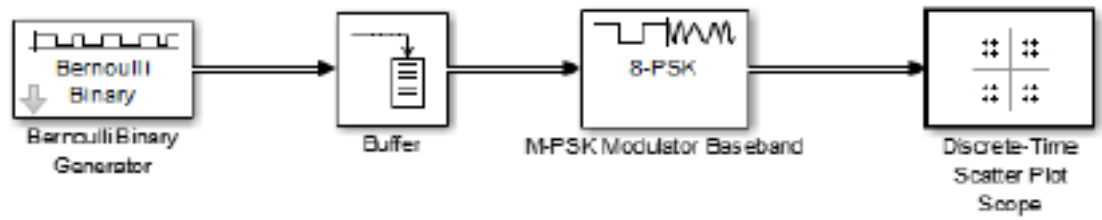


рис. 87. Модель PSK-модуляции

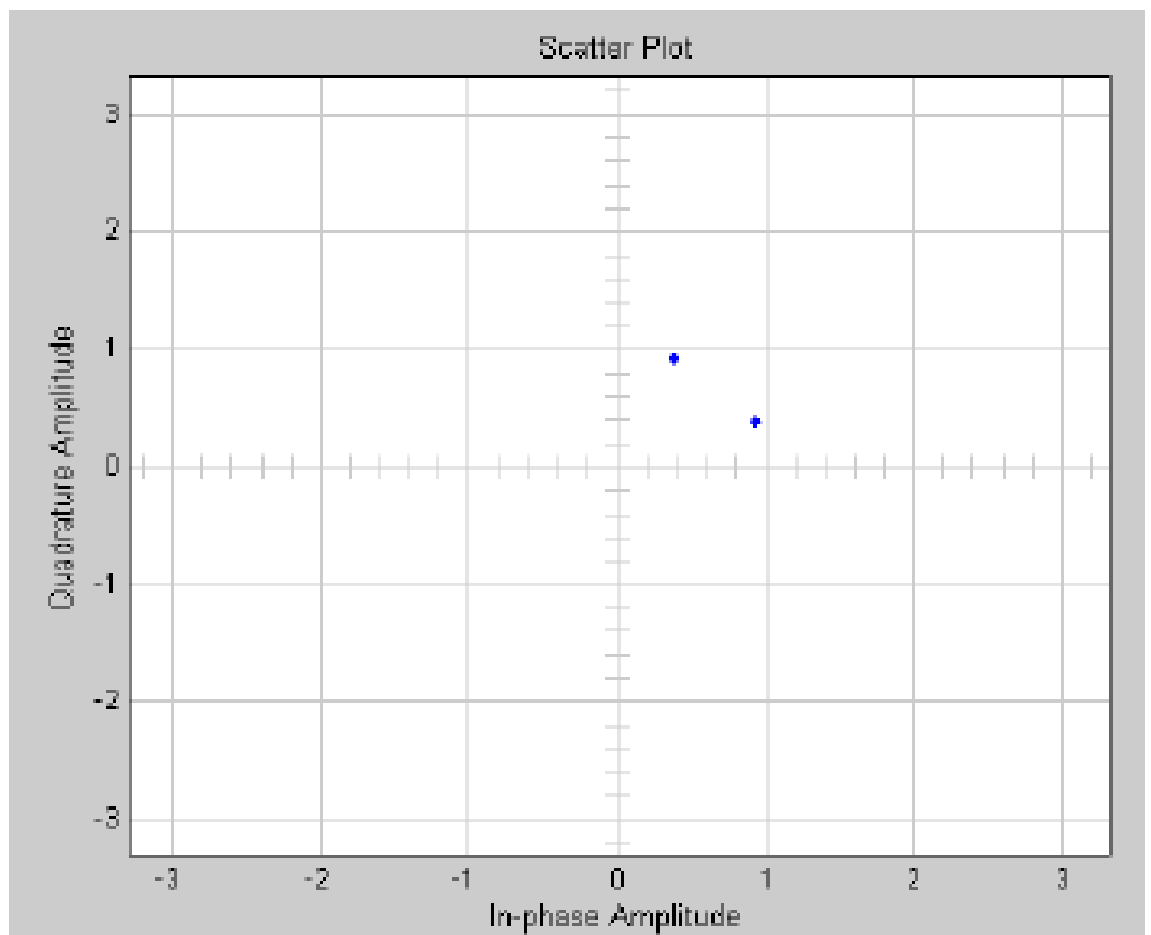


рис. 88. Сигнальное созвездие PSK-модуляции.

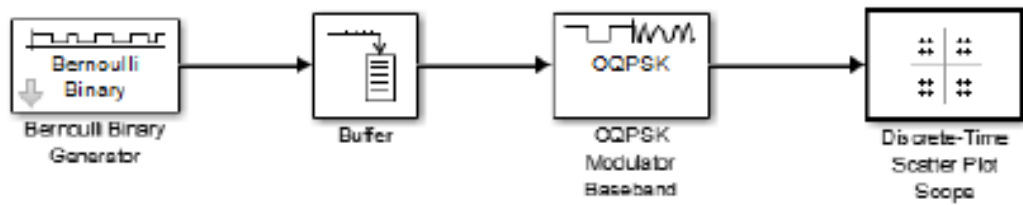


рис. 89. Модель OQPSK-модуляции

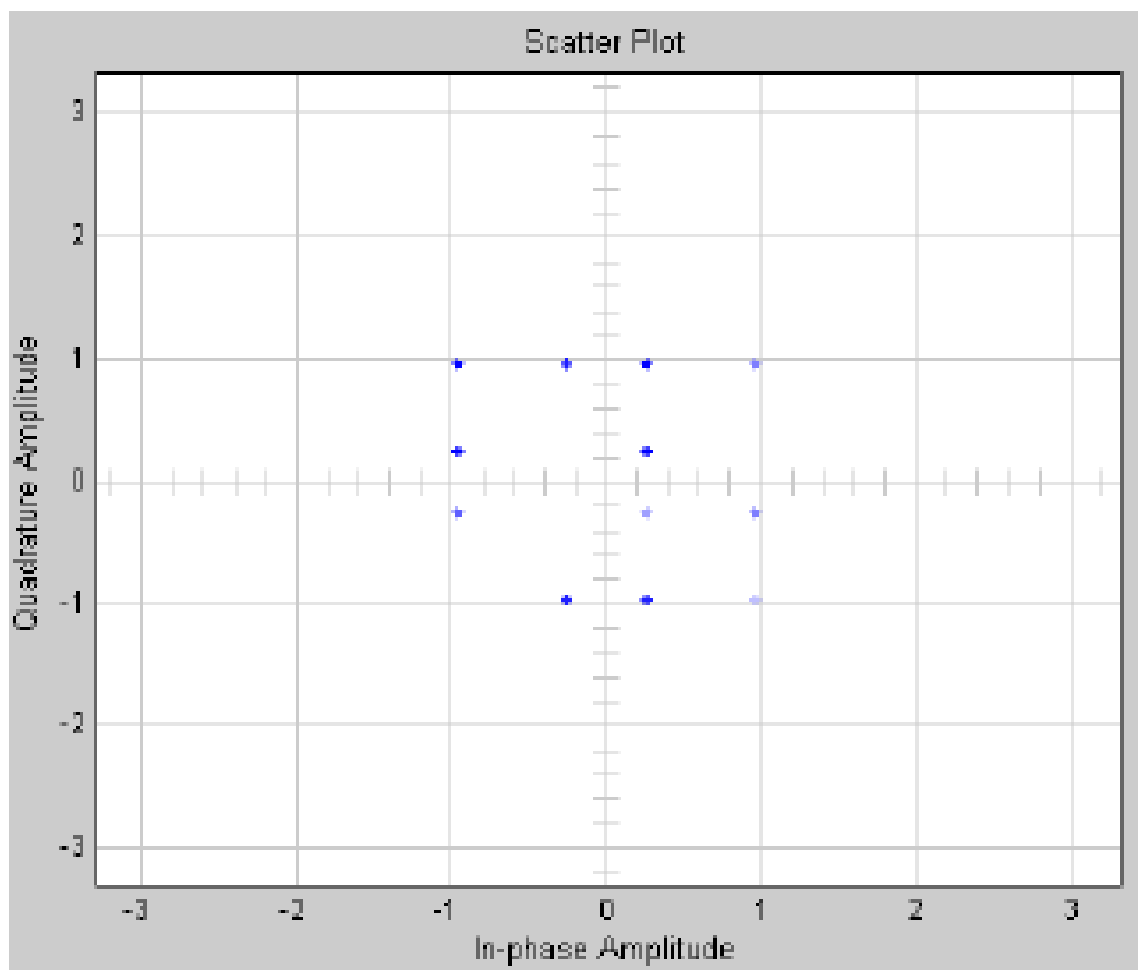


рис. 90. Сигнальное созвездие OQPSK-модуляции.

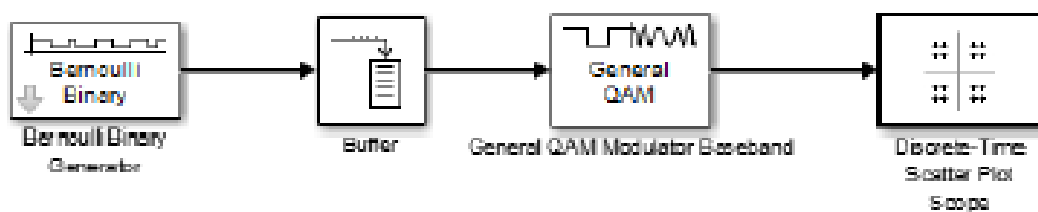


рис. 91. Модель genQAM-модуляции

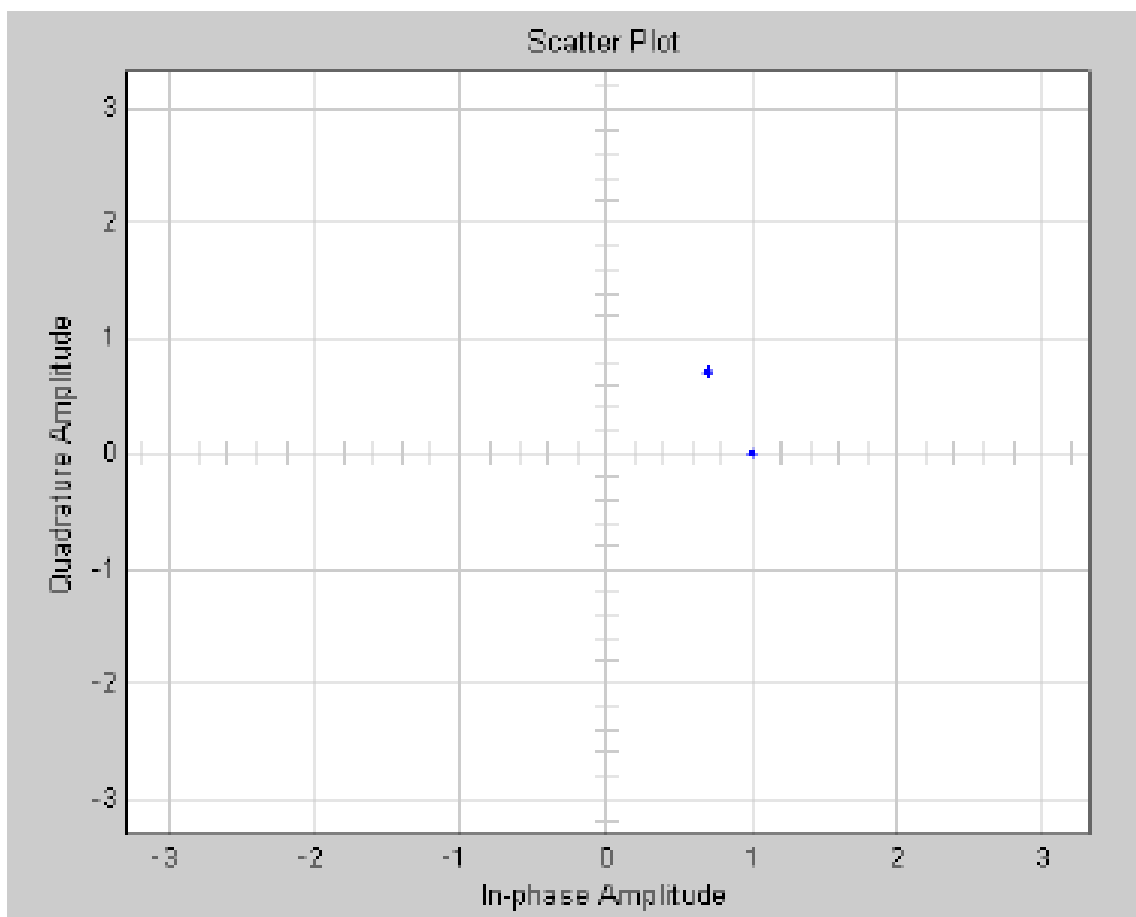


рис. Сигнальное созвездие genQAM-модуляции

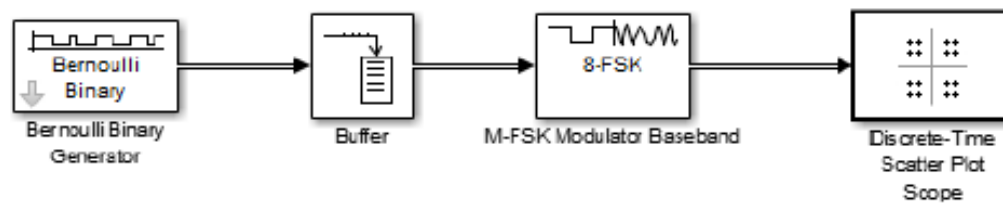


рис. Модель MSK-модуляции

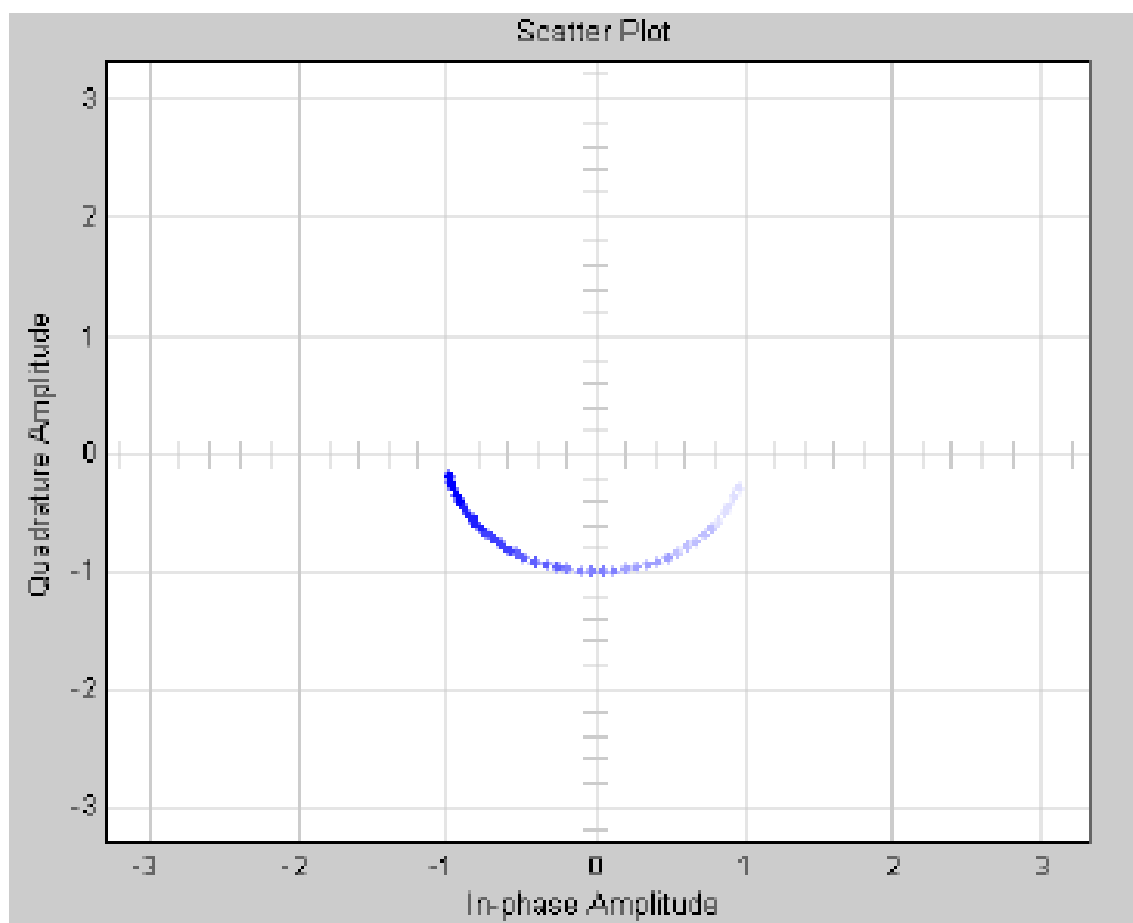


рис. Сигнальное созвездие MSK-модуляции

6.5 Вывод

В данной лабораторной работе было проведено исследование цифровой модуляции для различных модуляторов: BPSK, PSK, OQPSK, GENQAM, MSK. Для каждого способа цифровой модуляции были получены сигнальные созвездия модуляции и демодуляции. Радиосигнал представляется в виде двумерной точечной диаграммы на комплексной плоскости, точками на которой являются все возможные символы, представленные в геометрической форме. Более абстрактно, на диаграмме отмечены все значения, которые могут быть выбраны данной схемой манипуляции, как точки на комплексной плоскости. Сигнальные созвездия, полученные в результате измерения радиосигнала, могут использоваться для определения типа манипуляции и уровня искажений. При представлении передаваемого символа в виде комплексного числа и при модуляции косинусного и синусного сигналов несущей частоты, соответственно действительной и мнимой частями, символ можно передать двумя несущими с одной частотой. Часто такие несущие называются квадратурными. Если символы представлены в виде комплексных чисел, их можно представить в виде точек на комплексной плоскости. Действительная и мнимая оси часто называют in phase (синфазной) или I-осью и quadrature (квадратурной) или Q-осью. При нанесении на диаграмму точек от нескольких символов можно получить сигнальное созвездие. Точки созвездия представляют множество модулирующих символов, то есть модулирующий алфавит. При сравнении методов модуляции видно, что все созвездия, кроме случая с OQPSK модулятором, наносятся на круг с центром в начале координат и определенного радиуса. При осуществлении модуляции каждого типа количество точек в каждом типе модуляции увеличивается. При демодуляции точки налагают на ось абсцисс координат. Результат сравнения ошибок всех типов модуляции: BPSK - 0; PSK - 0; OQPSK - 74; MSK - 0; M-FSK - 50 - Число несовпадающих символов в двух сообщениях (разница между символами исходного сигнала и демодулированного). 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 $\log N$, где N — уровень модуляции. Таким образом, чем выше уровень модуляции, тем больше данных мы можем передать (или потерять) за единицу времени. Количество точек созвездия говорит об уровне модуляции, его помехоустойчивости скоростных возможностях.