

Отчет по лабораторной работе по дисциплине  
"Сети и системы передачи данных"  
на тему "Визуализация сигналов во  
временной и частотной области"

Скрипаль Борис, Баратынский Александр

23.02.14

# Глава 1

## Лабораторная работа 4

### 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 Введение

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:

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

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

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

### 1.4 Алгоритм работы

- Построение чистого синусоидального сигнала с регулярной составляющей 10 Гц и с нулевой начальной фазой

- Вывод временной характеристики сигнала
- Реализация одномерного преобразования Фурье на основе 512 точек
- Построение графика спектральной плотности для чистого синусоидального сигнала
- Построение зашумленного синусоидального сигнала путем добавления к чистому синусоидальному сигналу случайной аддитивной компоненты с нулевым средним
- Вывод временной характеристики полученного сигнала
- Реализация одномерного преобразования Фурье на основе 512 точек
- Построение графика спектральной плотности для зашумленного синусоидального сигнала

## 1.5 Код MATLAB

```
x = 0:0.01:4*pi;
t0 = 10;
%исходный сигнал
ynorm = sin(2*pi*t0*x);
plot(x(1:200),ynorm(1:200))
grid
%спектр исходного сигнала
figure
spectrum = fft(ynorm,512);
normspectrum = spectrum.*conj(spectrum)/512;
f=100*(0:255)/512;
plot(f, normspectrum(1:256))
axis([0 max(f) 0 10])
grid
%зашумленный сигнал
ynoise = ynorm+ 0.8*rand(size(x));
figure
plot(x(1:200),ynoise(1:200));
grid
%спектр зашумленного сигнала
spectrum = fft(ynoise,512);
noizespectrum = spectrum.*conj(spectrum)/512;
figure
plot(f, noizespectrum(1:256))
axis([0 max(f) 0 10])
grid
```

## 1.6 Результаты работы

В результате выполнения программы мы получили четыре графика: временная и частотная характеристика для чистого и зашумленного синусоидального сигнала. Графики представлены ниже.

Как видно из графиков, регулярная составляющая, как у чистого синусоидального сигнала, так и у зашумленного равна 10 Гц. Временная же характеристика чистого и зашумленного сигнала отличается достаточно сильно, хотя в обоих случаях заметны общие черты.

## 1.7 Вывод

Данная лабораторная работа заключалась в получении и сравнении 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 значения совпали с ожидаемыми. Стоит отметить, что спектр получается периодическим по частоте.

Теоретически, спектр синуса должен выглядеть как один импульс очень малой ширины. Но, из-за того, что синус конечный, то есть умноженный на прямоугольное окно, то и его спектр сворачивается со спектром прямоугольного окна. Так как наш сигнал дискретизирован, он умножается на решетку дельта-импульсов. Спектр решетки дельта-импульсов - также решетка дельта-импульсов, и спектр нашего сигнала сворачивается также с решеткой дельта-импульсов. Из-за чего получается спектр в виде нескольких импульсов.

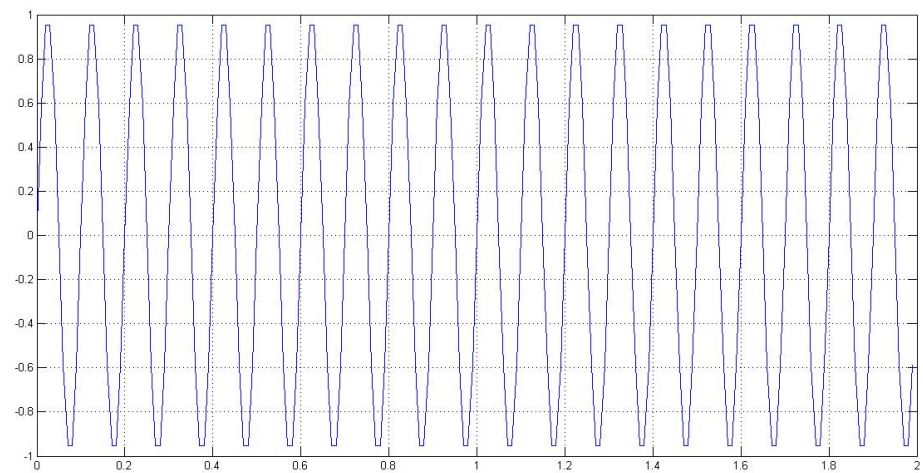


рис. 1 Временная характеристика чистого синусоидального сигнала

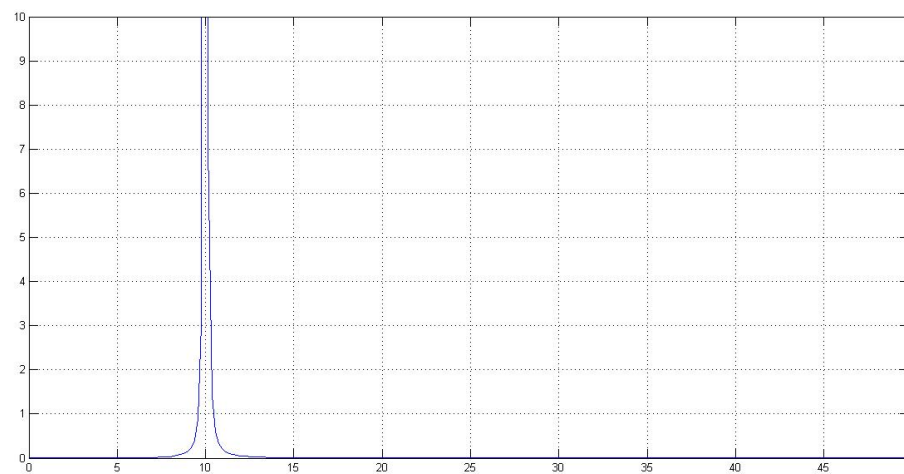


рис. 2. Частотная характеристика чистого синусоидального сигнала

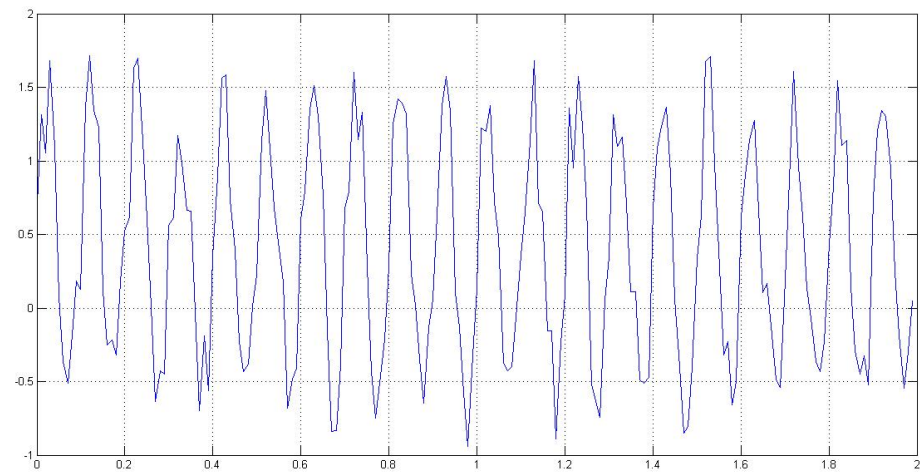


рис. 3. Временная характеристика зашумленного синусоидального сигнала

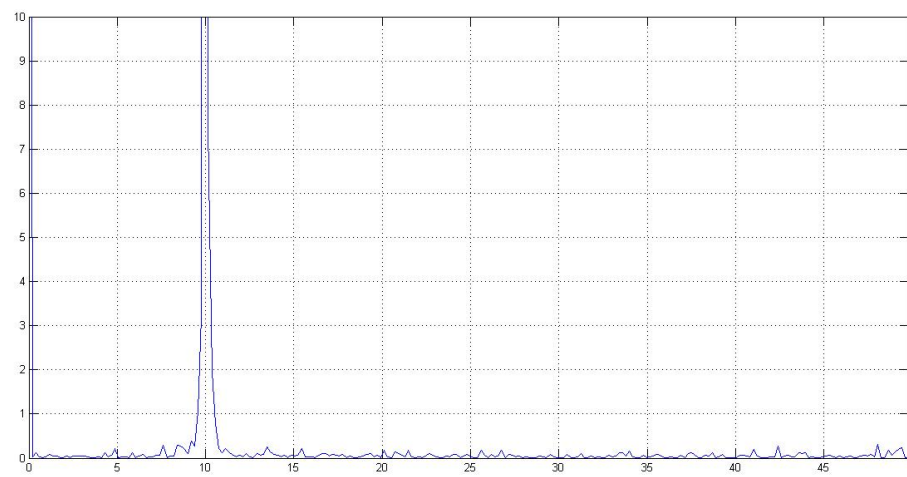


рис. 4. Частотная характеристика зашумленного синусоидального сигнала

## Глава 2

# Лабораторная работа 5

### 2.1 Цель работы

Целью данной работы является получение представлений о свойствах спектров, а именно:

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

### 2.2 Теоритическая 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

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

## 2.4 Результаты работы

В результате, были получены 6 графиков: графики полигармонического, прямоугольного и треугольного сигналов, а так же графики их спектров соответственно. Графики представлены на рисунках ниже. Ниже представлены результаты работы в Simulink

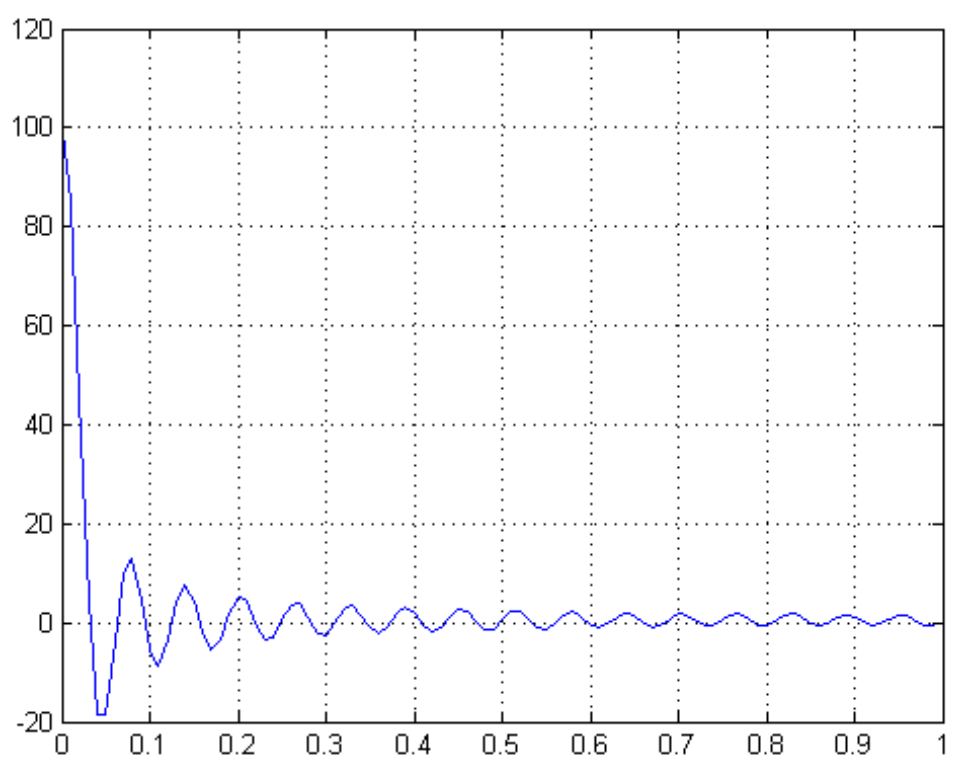


рис. 5. Полигармонический сигнал

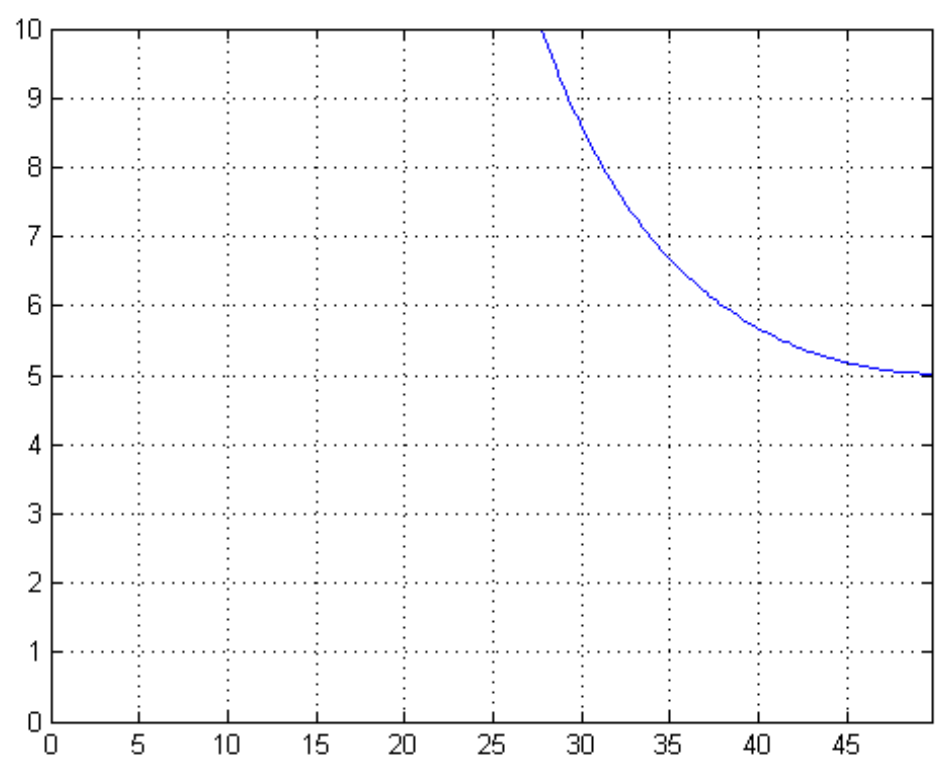


рис. 6. Спектр полигармонического сигнала

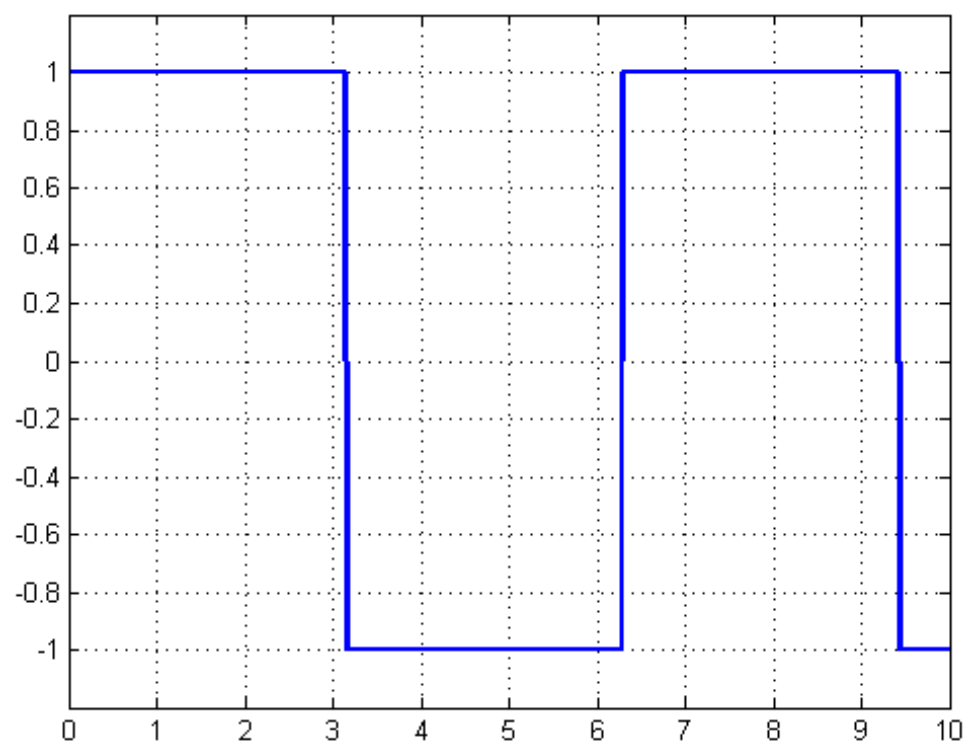


рис. 7. Прямоугольный сигнал

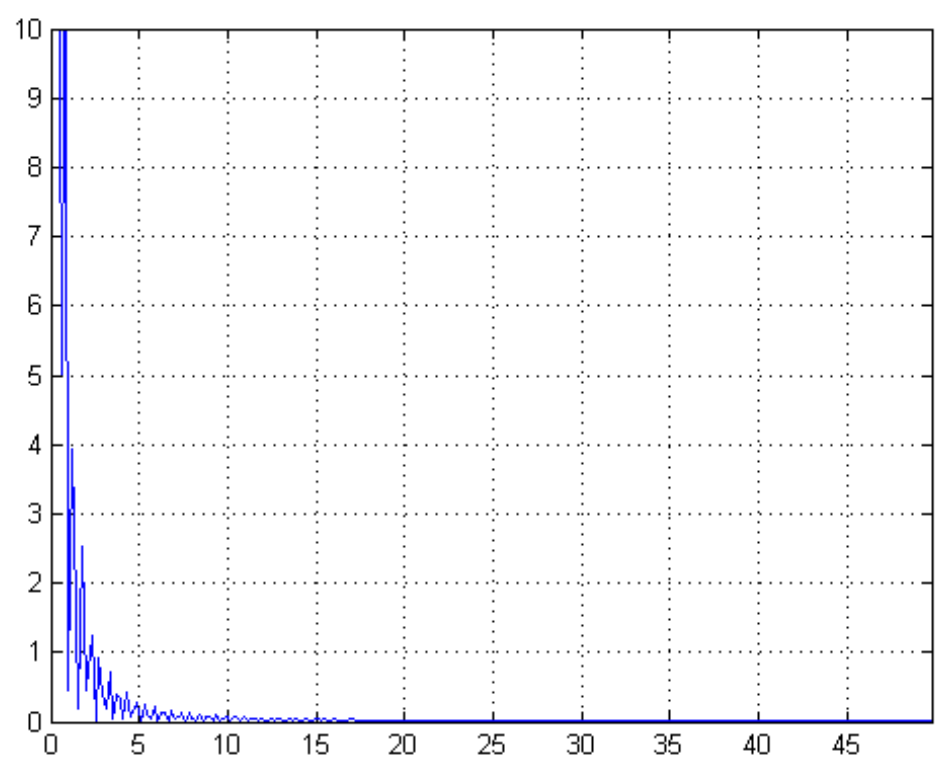


рис. 8. Спектр прямоугольного сигнала

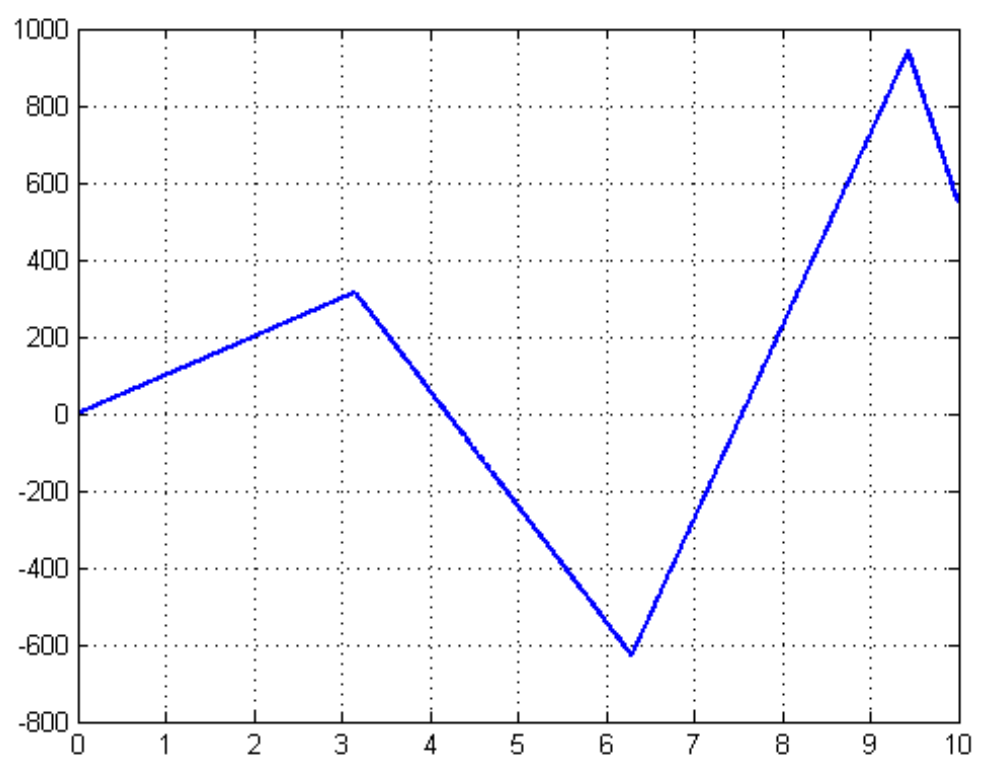


рис. 9. Треугольный сигнал

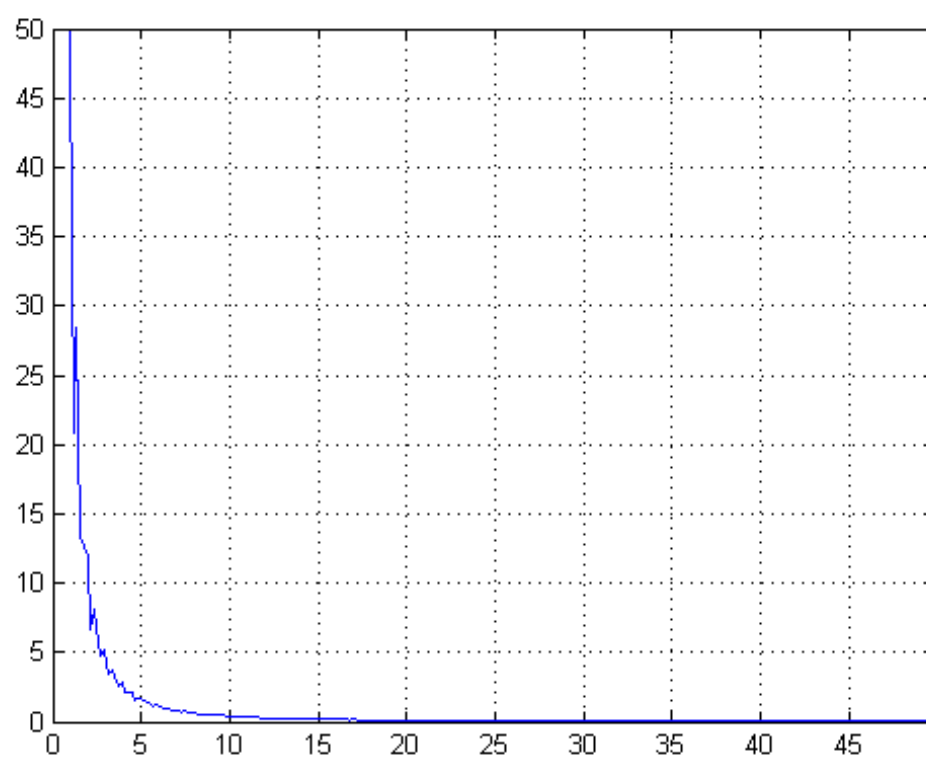


рис. 10. Спектр треугольного сигнала

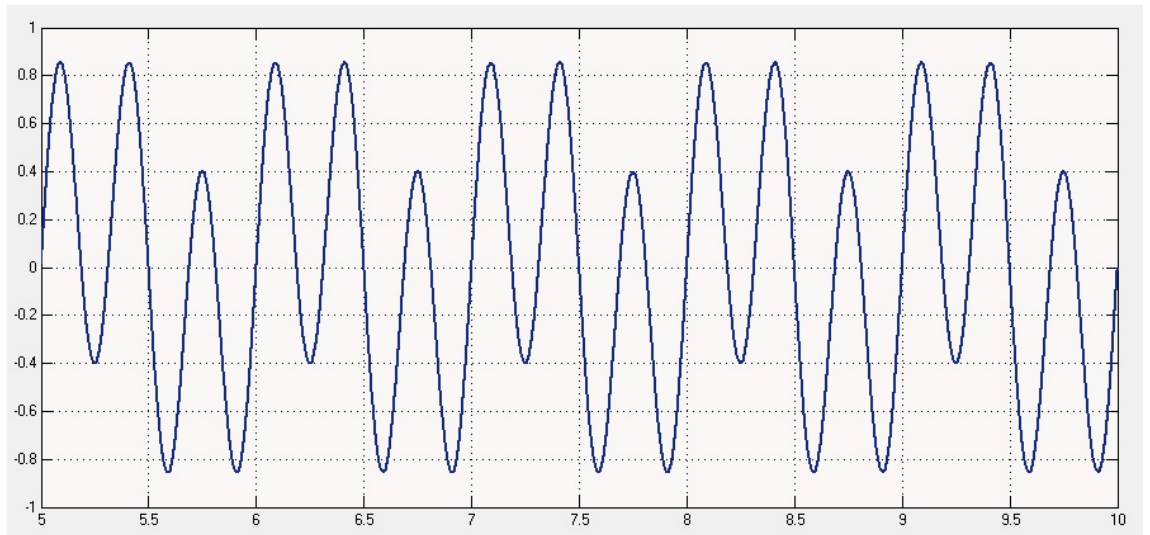


рис. 11. Полигармонический сигнал

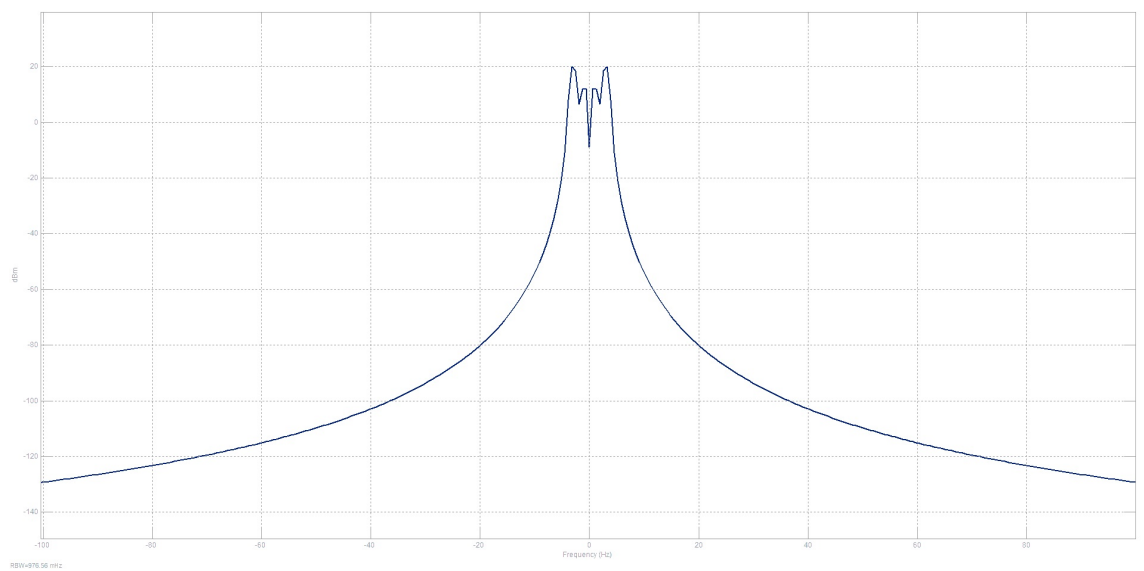


рис. 12. Спектр полигармонического сигнала



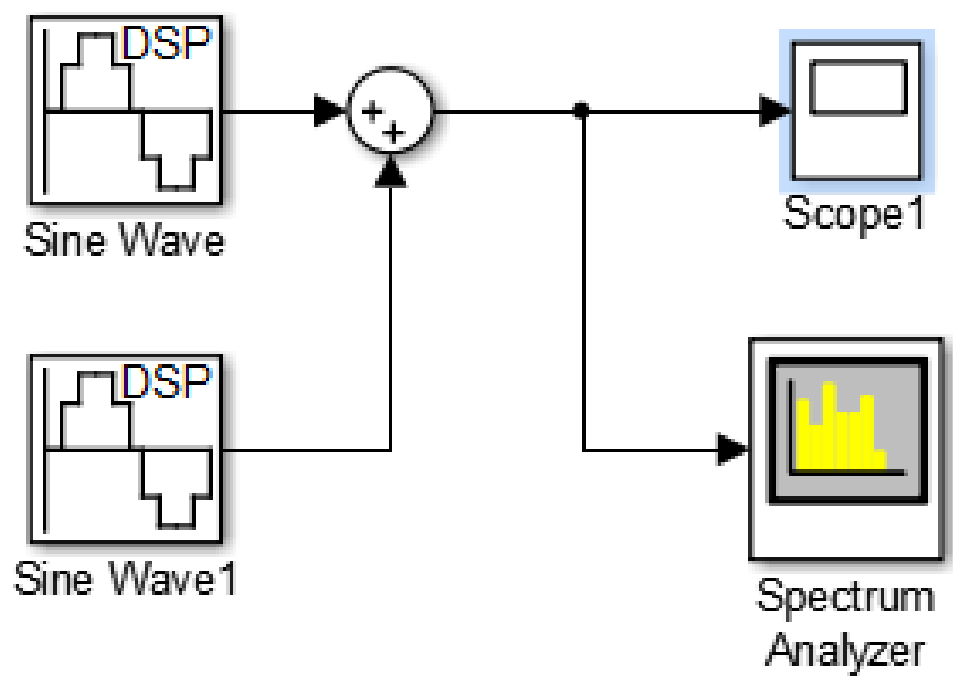


рис. 13. Модель

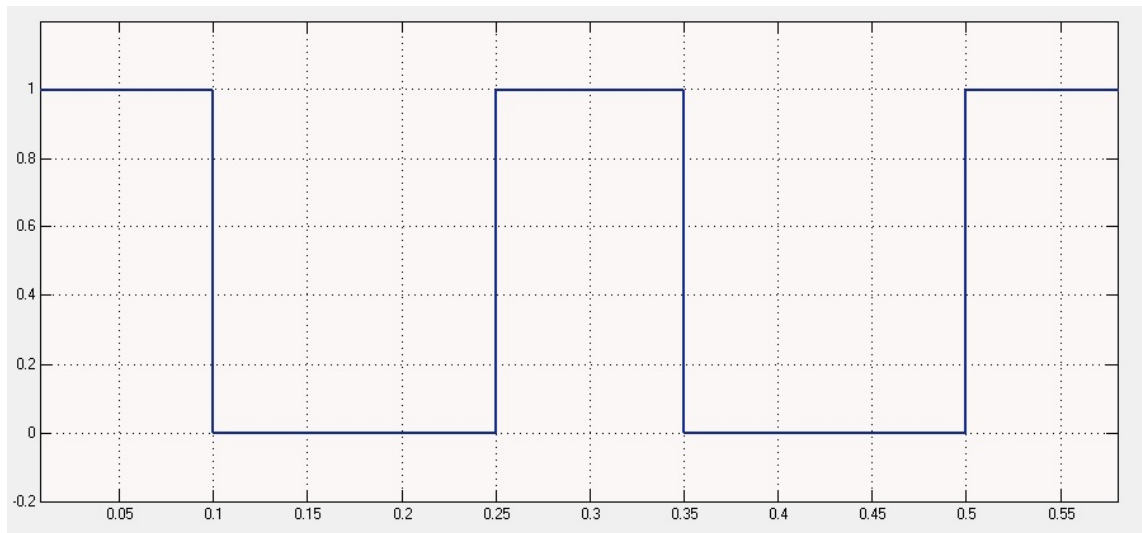


рис. 14. Прямоугольный сигнал

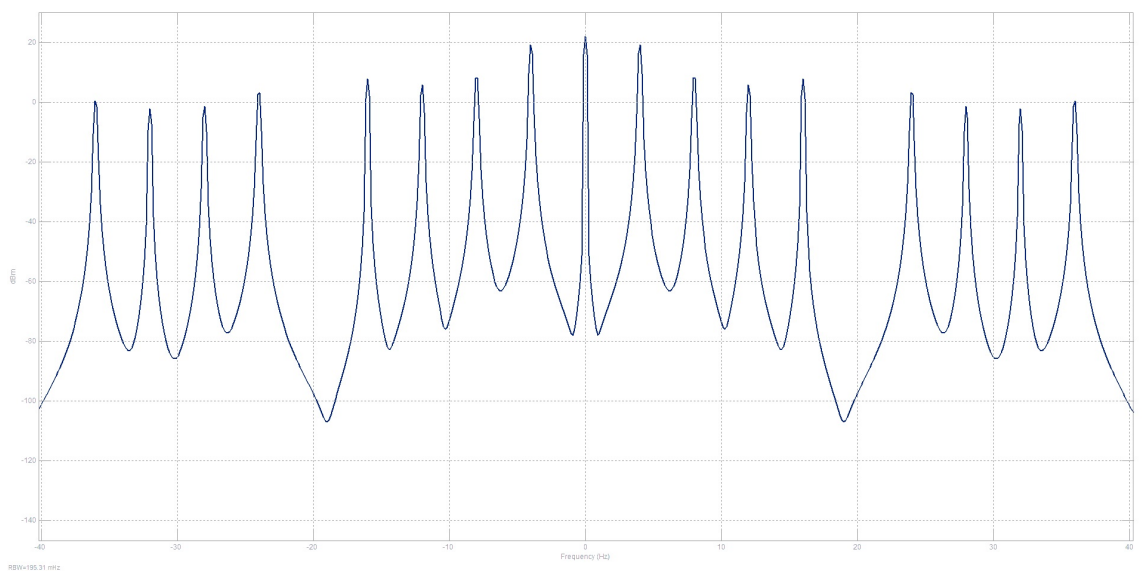


рис. 15. Спектр прямоугольного сигнала

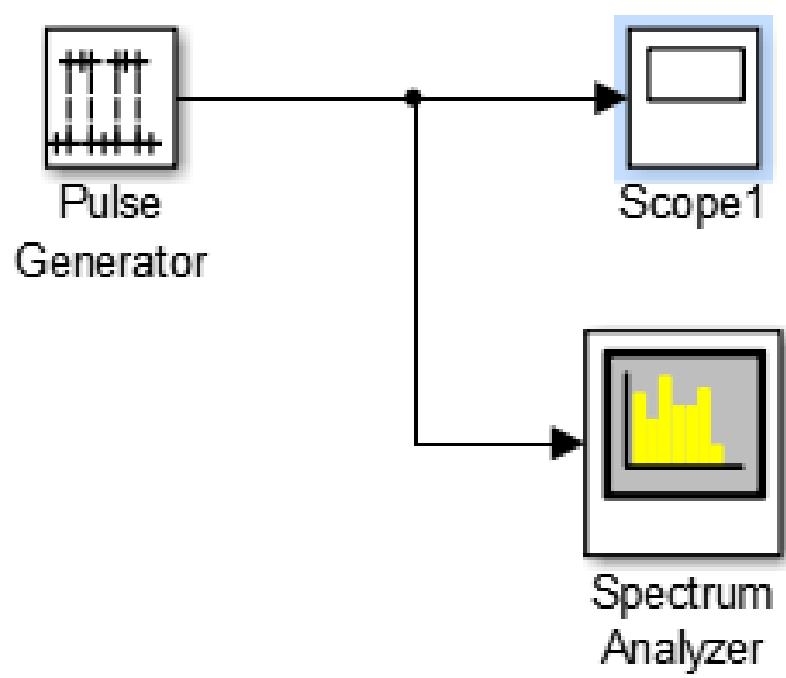


рис. 16. Модель

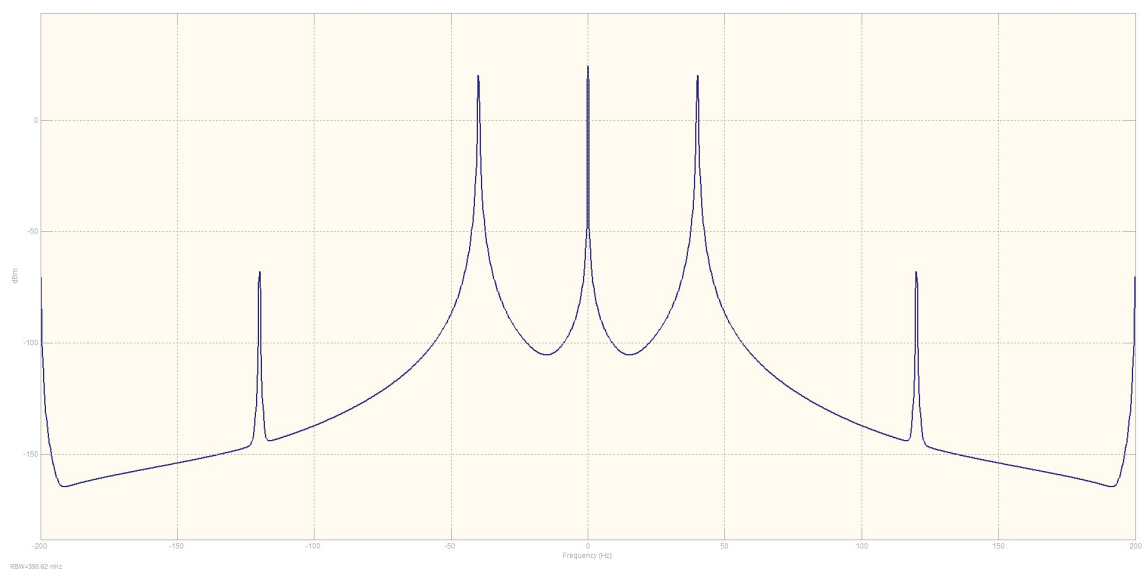


рис. 18. Спектр треугольного сигнала

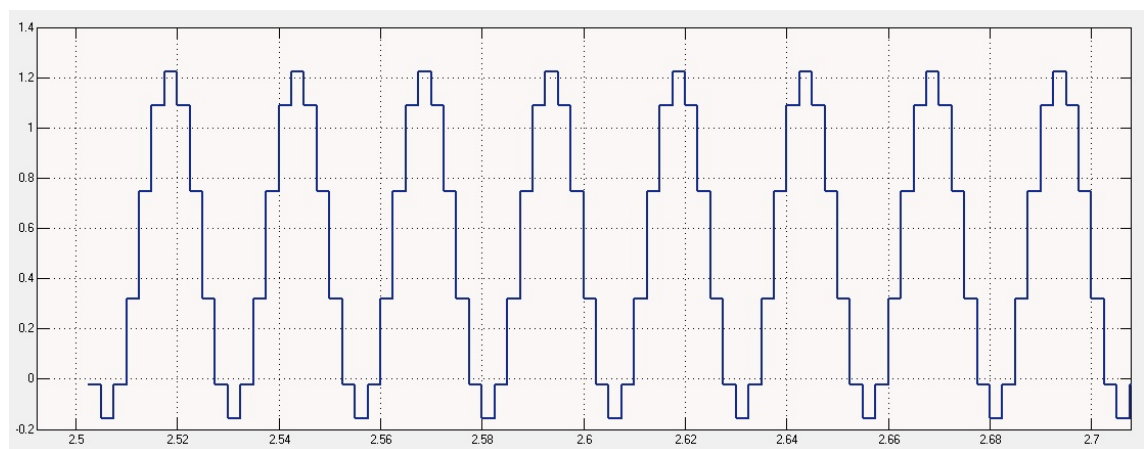


рис. 17. Треугольный сигнал

## 2.5 Вывод

В данной лабораторной работе было проведено моделирование основных видов сигналов: полигармонического, прямоугольного и треугольного, а так же были получены их спектры. Сигналы были получены как при помощи формул данных функций, так и при помощи среды моделирования Simulink. Спектры сигналов так же были получены обоими вышеприведенными способами. Стоит отметить получение треугольного сигнала, т.к. он был по-

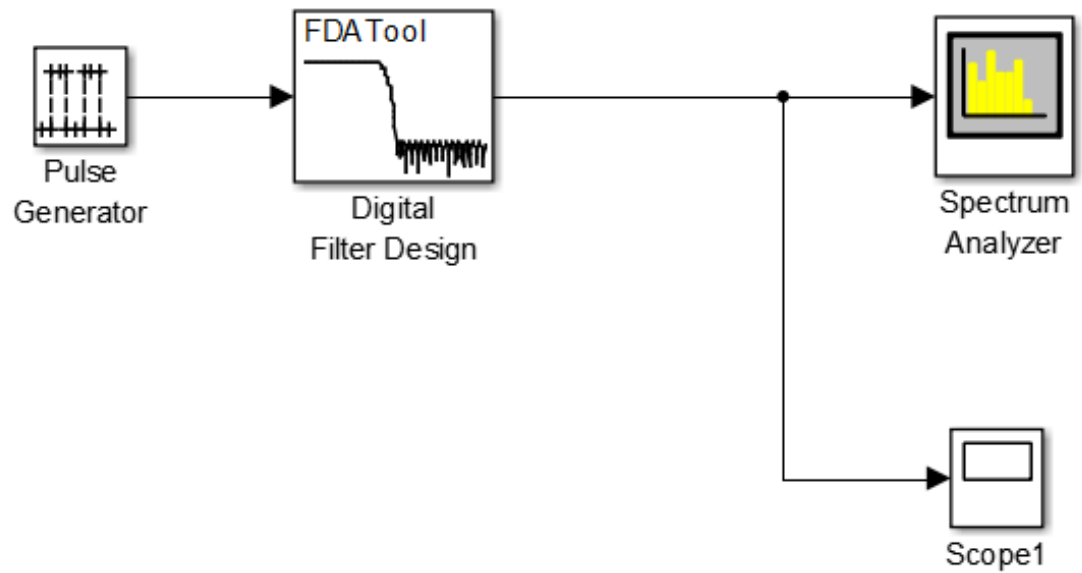


рис. 19. Модель

лучен не при помощи конкретной формулы, а путем свертки двух прямоугольных сигналов. Это возможно из-за того, что линейная функция может быть получена как интеграл от двух констант. Таким образом, мы получаем две линейные функции, с отличием в коэффициенте наклона (у одной он положителен, а у другой отрицателен). Так же стоит отметить, что все полученные результаты соответствуют теоретическим ожиданиям.

## Глава 3

# Лабораторная работа №6

### 3.1 Цель работы

Изучить воздействие фильтра нижних частот на тестовый сигнал с шумом. Для этого предлагается сгенерировать гармонический сигнал с шумом, затем смоделировать фильтр нижних частот (ФНЧ) и сравнить сигналы до и после фильтрации во временной и частотной областях.

### 3.2 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  $H(s)$ :

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

где:

- $n$  — порядок фильтра
- $w_c$  (3dB)  $G_0$  (0dB) . Легко заметить, что для бесконечных значений  $n$  АЧХ становится прямоугольной функцией, и частоты ниже частоты среза будут пропускаться с коэффициентом усиления  $G_0$ , а частоты выше частоты среза будут подавляться.

### 3.3 Код matlab

- `x = 0:0.01:4*pi;\newline`

```

f=100*(0:255)/512;\newline
figure\newline
noise=rand(size(x));\newline
y = sin(2*pi*x);\newline
y_noisy = y + 0.3*noise;\newline
plot(x(1:200),y(1:200))\newline
grid\newline
figure\newline
plot(x(1:200),y_noisy(1:200))\newline
grid\newline
[B,A] = butter(16,0.99); синтез ФНЧ Баттерворта\newline
B=B./sum(B);\newline
A=A./sum(A);\newline
обработка сигнала ФНЧ\newline
figure\newline
y_filtered = conv(y_noisy,[B,A]);\newline
plot(x(1:200),y_filtered(1:200))\newline
grid\newline
выделение спектра\newline
figure\newline
normal_spectrum = fft(y,512);\newline
norm_spectrum = normal_spectrum.*conj(normal_spectrum)/512;\newline
plot(f,norm_spectrum(1:256))\newline
axis([0 max(f) 0 2])\newline
grid\newline
figure\newline
noisy_spectrum = fft(y_noisy,512);\newline
norm_noisy_spectrum = noisy_spectrum.*conj(noisy_spectrum)/512;\newline
plot(f,norm_noisy_spectrum(1:256))\newline
axis([0 max(f) 0 2])\newline
grid\newline
figure\newline
spectrum = fft(y_filtered,512);\newline
norm_filtered_spectrum=spectrum.*conj(spectrum)/512;\newline
plot(f,norm_filtered_spectrum(1:256))\newline
axis([0 max(f) 0 2])\newline
grid\newline

```

### 3.4 Результаты работы

результаты работы в Simulink beginfigure

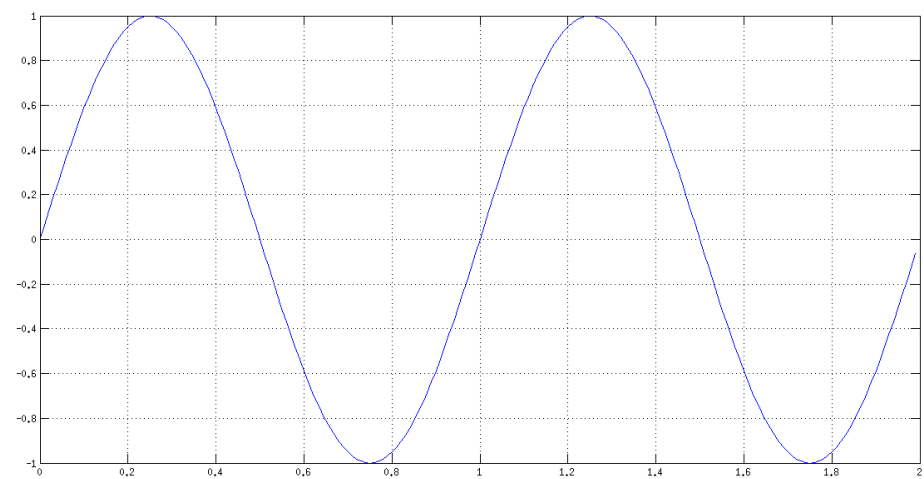


рис. 20. Полигармонический сигнал

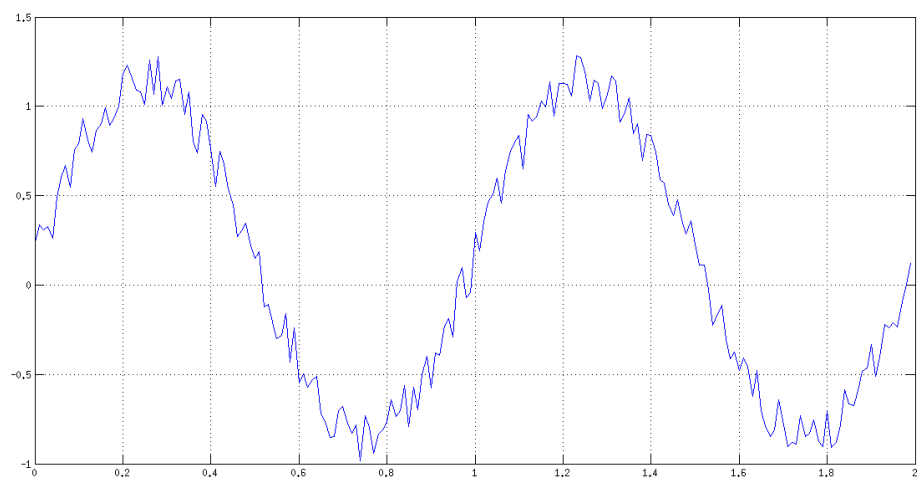


рис. 21. Зашумленный полигармонический сигнал



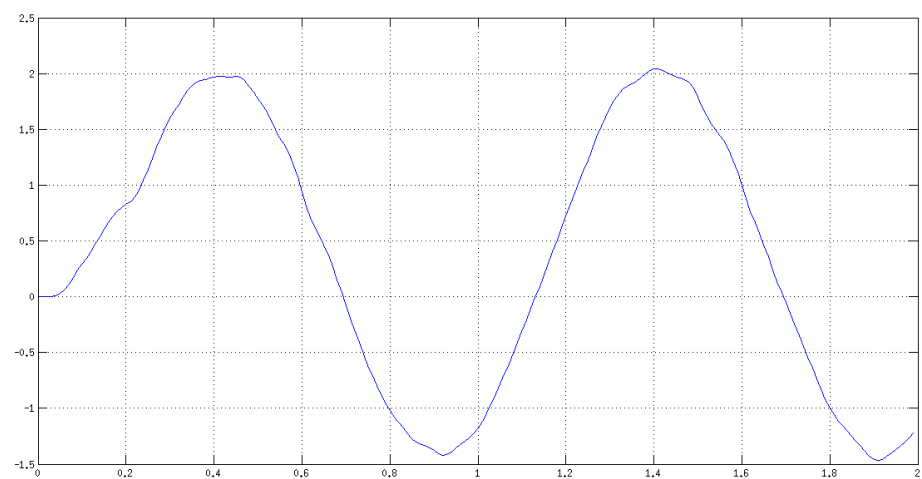


рис. 22. Сигнал после фильтрации

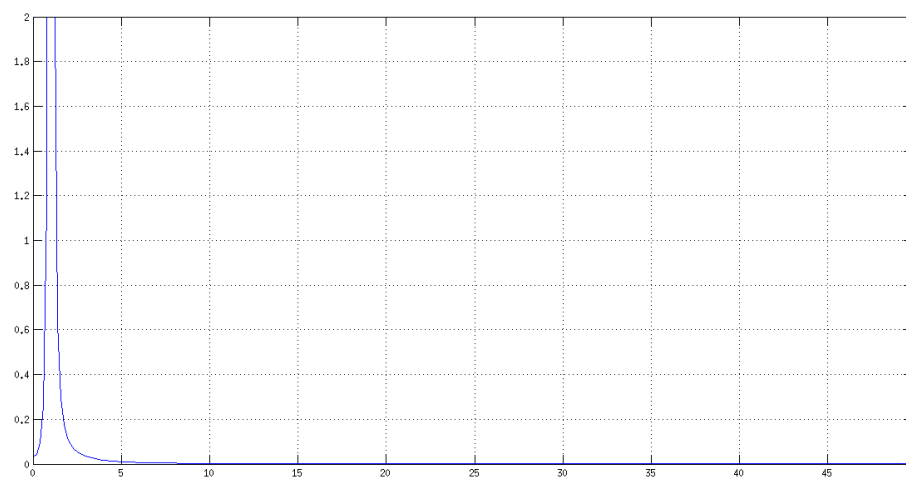


рис. 23. Спектр полигармонического сигнала

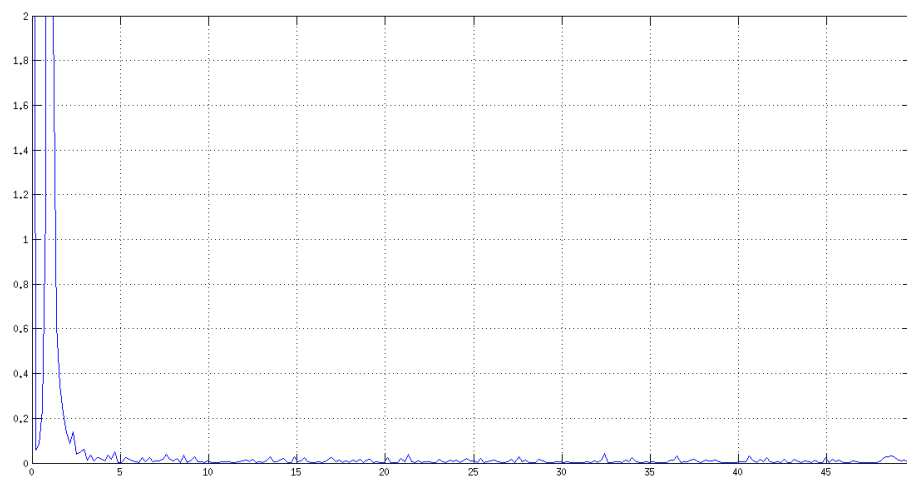


рис. 24. Спектр зашумленного сигнала

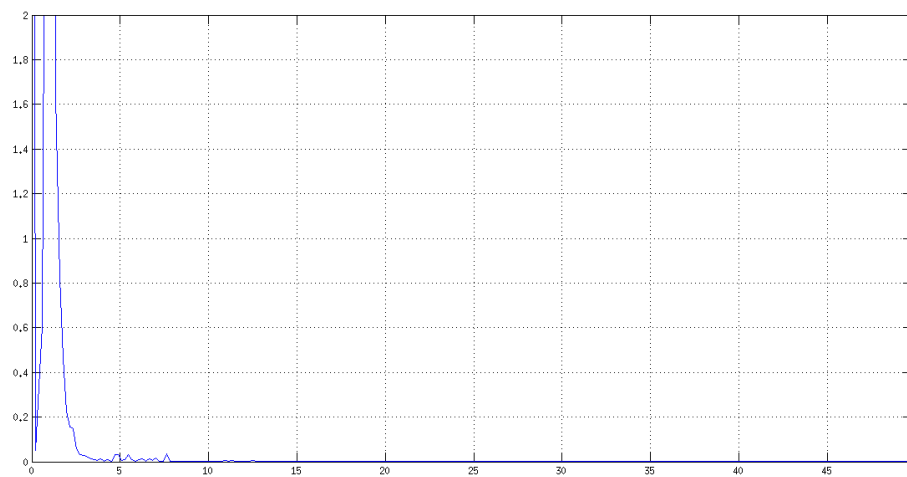


рис. 25. Спектр сигнала, прошедшего фильтрацию сигнала

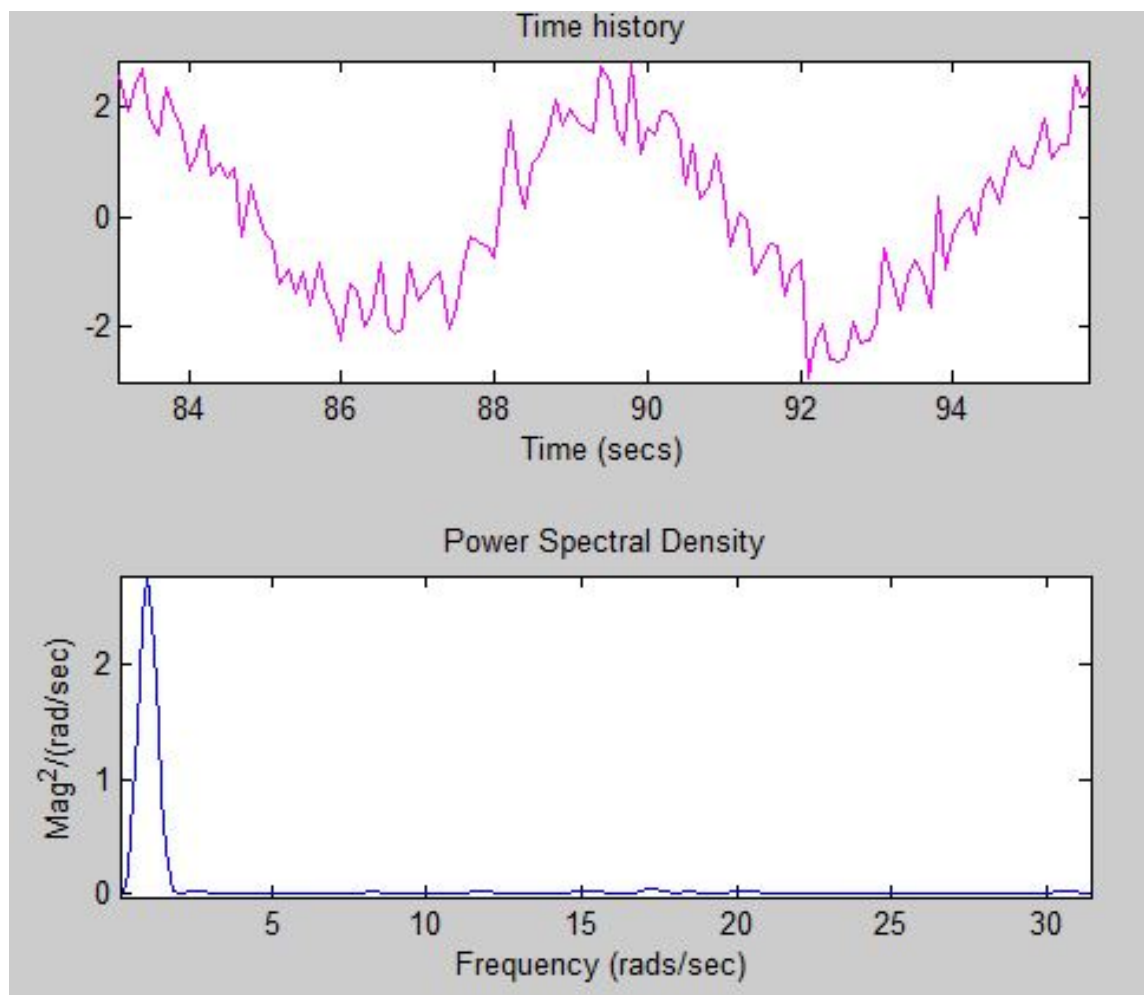


рис. 26. Спектр зашумленного сигнала

### 3.5 Вывод

Временная характеристика сигнала после фильтрации фильтром Баттерворта 16 порядка, практически совпадает с исходным сигналом. Незначительные отличия объясняются во-первых, погрешностью вычислений, а во-вторых - погрешностью алгоритма. Для получения идеального сигнала, порядок фильтра должен стремиться к бесконечности, что нереализуемо на практике. Частотная характеристика у исходного, отфильтрованного и зашумленного сигнала очень схожи, однако стоит отметить, что характеристика отфильтрованного сигнала более плавная и ближе к частотной характеристике исходного сигнала, чем частотная характеристика зашумленного сигнала.

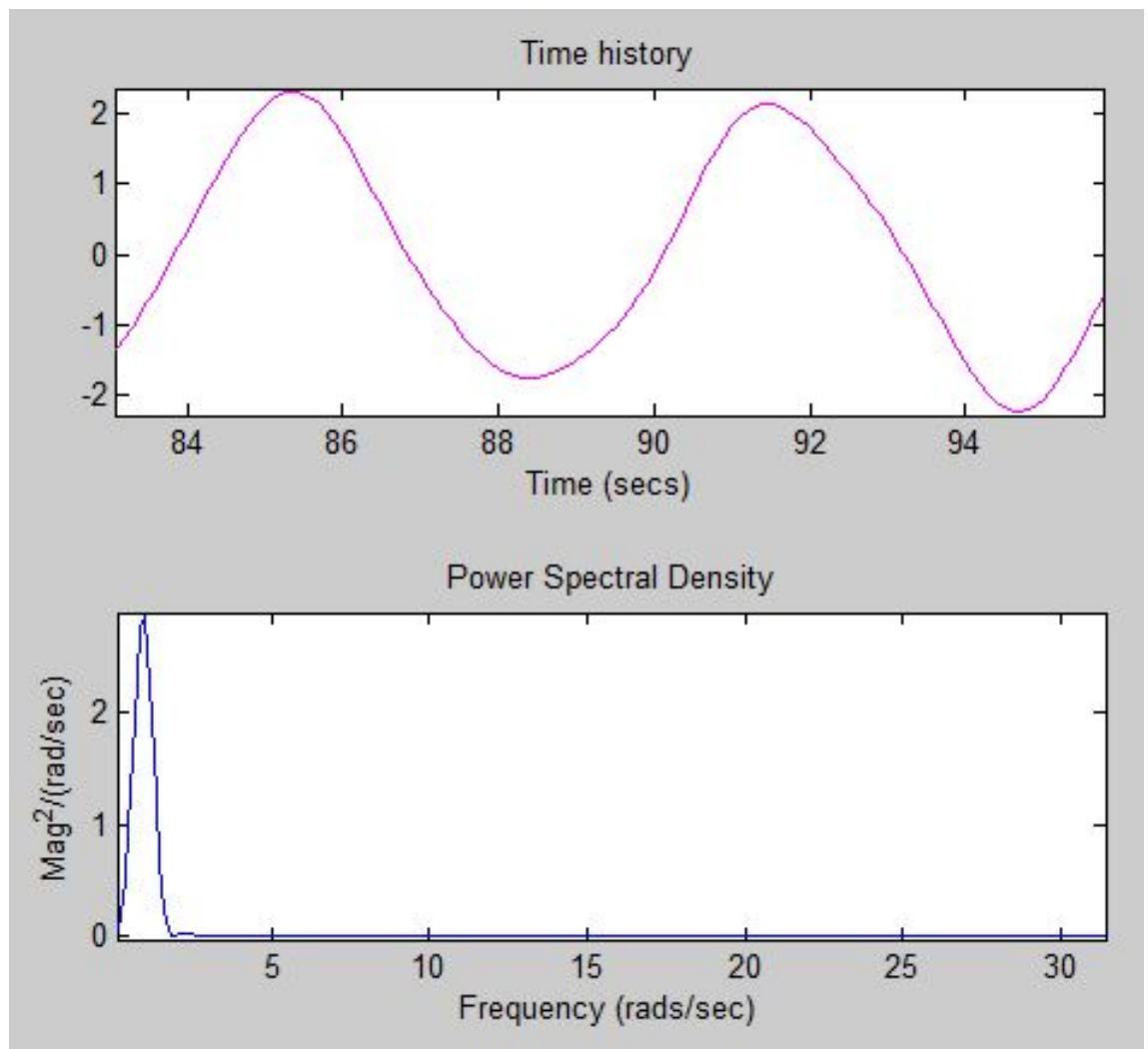


рис. 27. Спектр отфильтрованного сигнала

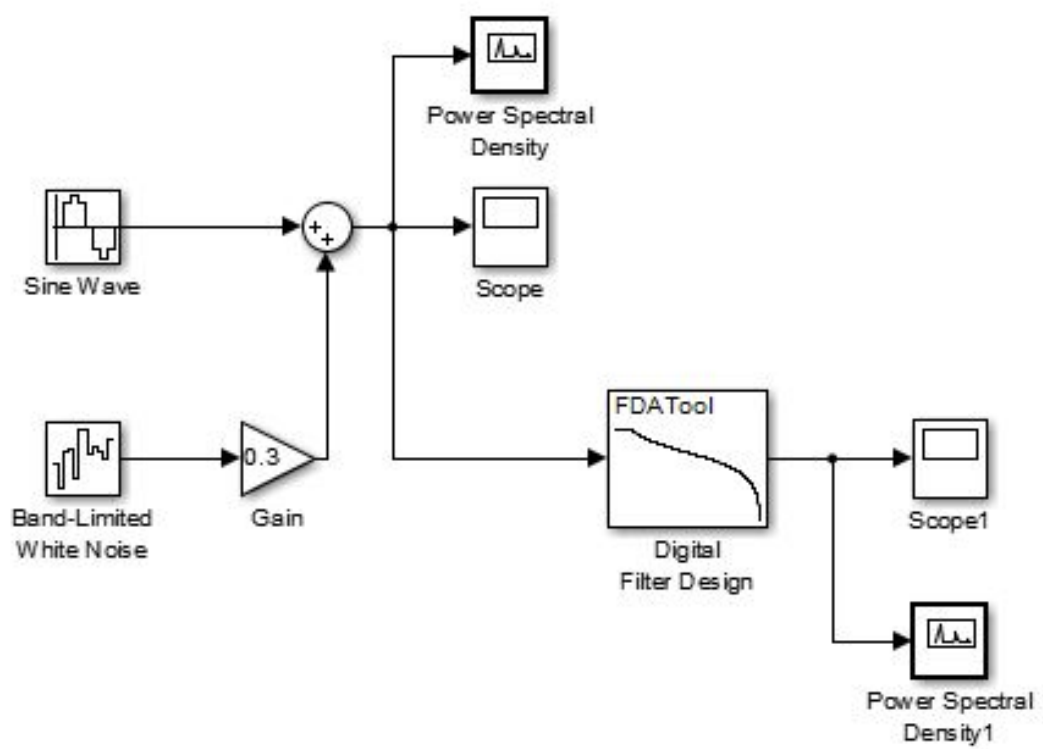


рис. 28. Модель

## Глава 4

# Лабораторная работа №7

### 4.1 Постановка задачи

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 Теоретическая часть

Амплитудная модуляция - модуляция, при которой сигнал представляет собой произведение информационной огибающей  $u(t)$  и гармонического колебания ее заполнения с более высокими частотами. Простейшая форма

модулированного сигнала создается при однотоновой амплитудной модуляции, при которой несущий сигнал модулируется гармоническим колебанием с одной частотой  $\Omega$ :

$$u(t) = (1 + MU_m \cos(\Omega t)) \cos(\omega_0 t + \phi_0). \quad (4.5)$$

При модуляции с подавлением несущей частоты производится перемножение двух сигналов - модулирующего и несущего, при котором происходит подавление несущего колебания. КПД модуляции равен 100%. Физическая сущность подавления несущей заключается в том, что при переходе огибающей биений  $U(t)$  через 0 фаза несущей частоты высокочастотного заполнения изменяется на  $180^\circ$ . Синхронное детектирование является одним из способов демодуляции АМ-сигнала. Его суть состоит в умножении частоты сигнала на опорное колебание с несущей частотой. Результат умножения содержит два слагаемых: искомая амплитуда и АМ-сигнал с несущей частотой  $2\omega_0$ . Эта частота может быть убрана, если пропустить данный сигнал, через ФНЧ.

### 4.3 Код matlab

```
x = 0:0.01:8*pi;\newline
f0 = 0.1;\newline
\newline
y = sin(2*pi*f0*x);\newline
figure\newline
plot(x(1:1000),y(1:1000), 'LineWidth', 2)\newline
grid\newline
\newline
spectrum = fft(y, 512);\newline
norm_spectrum = spectrum.*conj(spectrum)/512;\newline
f = 100*(0:255)/512;\newline
figure\newline
plot(f, norm_spectrum(1:256), 'LineWidth', 2)\newline
axis([0 max(f) 0 70])\newline
grid\newline
\newline
Fc = 10*f0;\newline
Fs = 100*f0;\newline
U = ammod(y, Fc, Fs, 0, 1);\newline
figure \newline
plot(x(1:1000), U(1:1000), 'LineWidth', 2)\newline
grid\newline
\newline
u_spectrum = fft(U, 512);\newline
norm_u_spectrum = u_spectrum.*conj(u_spectrum)/512;\newline
figure\newline
plot(f, norm_u_spectrum(1:256))\newline
axis([0 max(f) 0 70])\newline
grid\newline
\newline
```

```

Fc = 10*f0;\newline
Fs = 100*f0;\newline
U = ammod(y, Fc, Fs);\newline
figure \newline
plot(x(1:1000), U(1:1000), 'LineWidth', 2)\newline
grid\newline
\newline
u_spectrum = fft(U, 512);\newline
norm_u_spectrum = u_spectrum.*conj(u_spectrum)/512;\newline
figure\newline
plot(f, norm_u_spectrum(1:256))\newline
axis([0 max(f) 0 70])\newline
grid\newline
\newline
Fc = 10*f0;\newline
Fs = 100*f0;\newline
U = ssbmod(y, Fc, Fs, [], 'upper');\newline
figure \newline
plot(x(1:500), U(1:500), 'LineWidth', 2)\newline
grid\newline
\newline
u_spectrum = fft(U, 512);\newline
norm_u_spectrum = u_spectrum.*conj(u_spectrum)/512;\newline
figure\newline
plot(f, norm_u_spectrum(1:256))\newline
axis([0 max(f) 0 70])\newline
grid\newline
\newline
[b, a] = butter(10, Fc*2/Fs);\newline
z = ssbdemod(U, Fc, Fs, 0, b, a);\newline
figure\newline
plot(x(1:1000), z(1:1000), 'LineWidth', 2)\newline
grid\newline
\newline
du_spectrum = fft(U, 512);\newline
norm_du_spectrum = du_spectrum.*conj(du_spectrum)/512;\newline
figure\newline
plot(f, norm_du_spectrum(1:256))\newline
axis([0 max(f) 0 70])\newline
grid\newline
\newline
M = 0.5;\newline
n = M^2/(M^2+2)\newline

```

## 4.4 Результаты работы

beginfigure



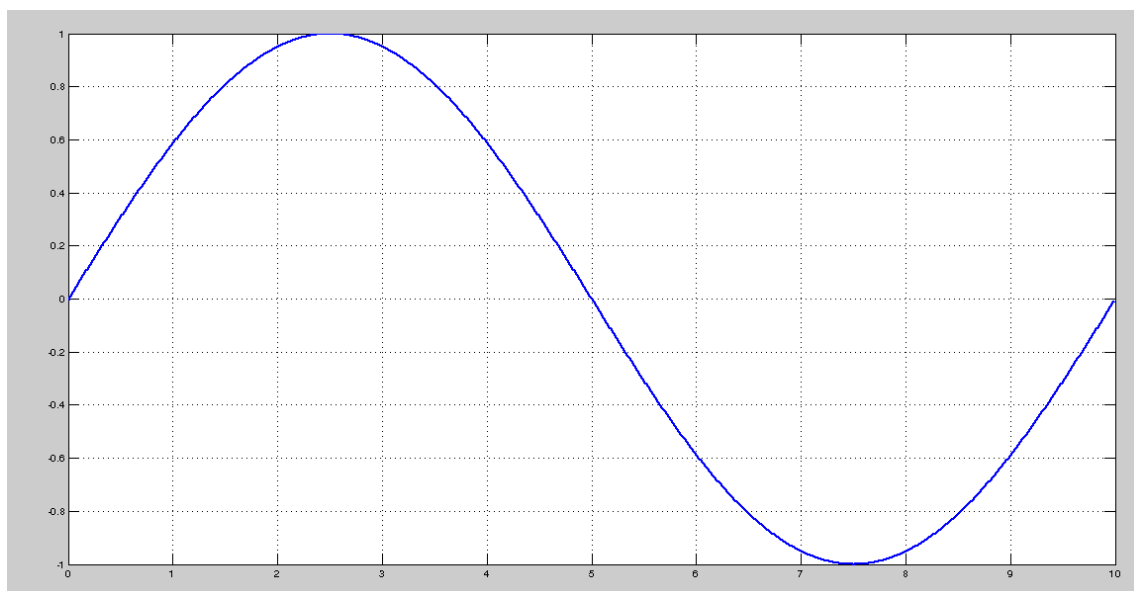


рис. 29. Исходный сигнал

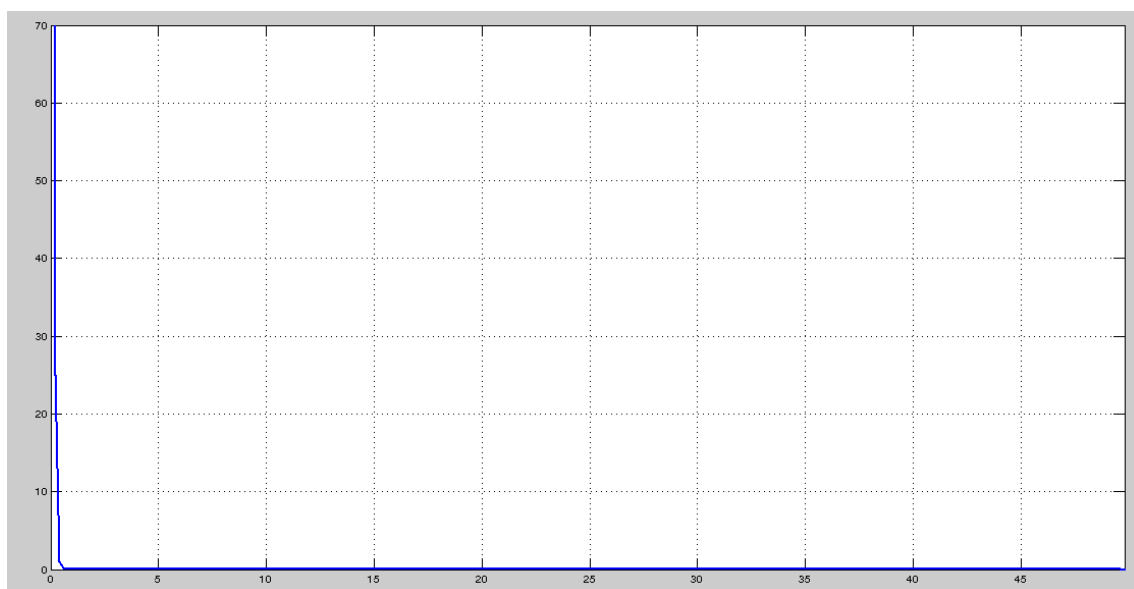


рис. 30. Спектр исходного сигнала

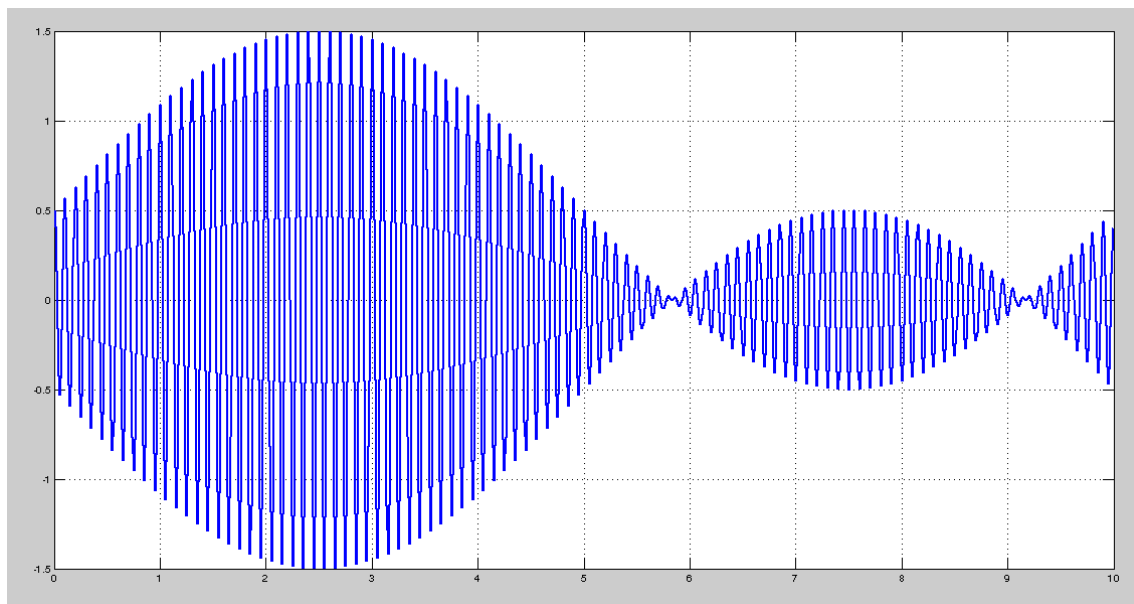


рис. 31. Модуляция с  $M = 0.5$

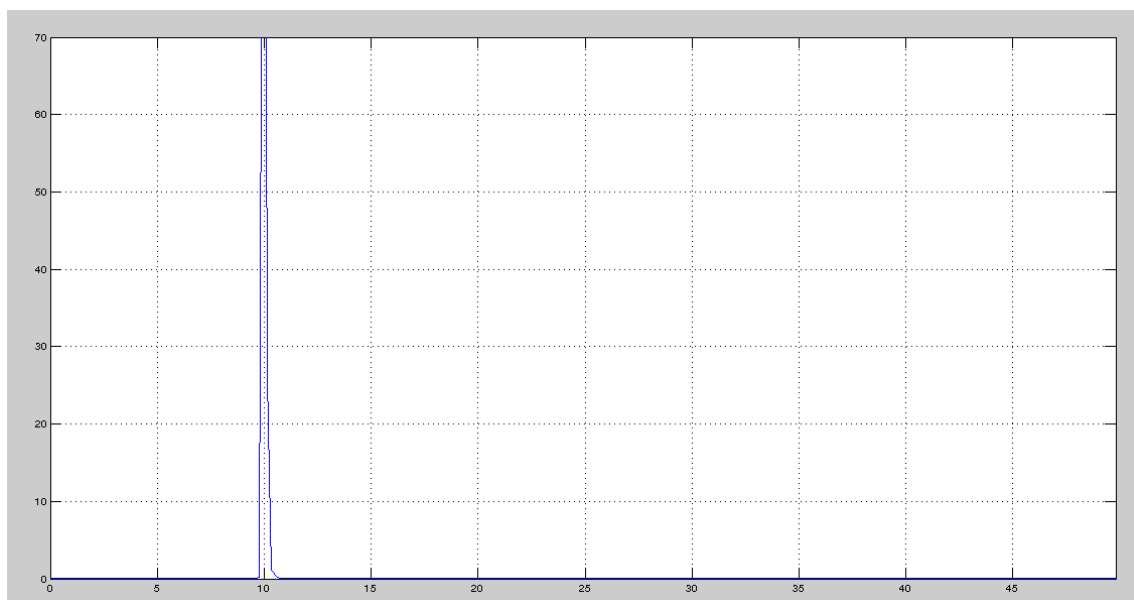


рис. 32. Спектр сигнала с  $M=0.5$

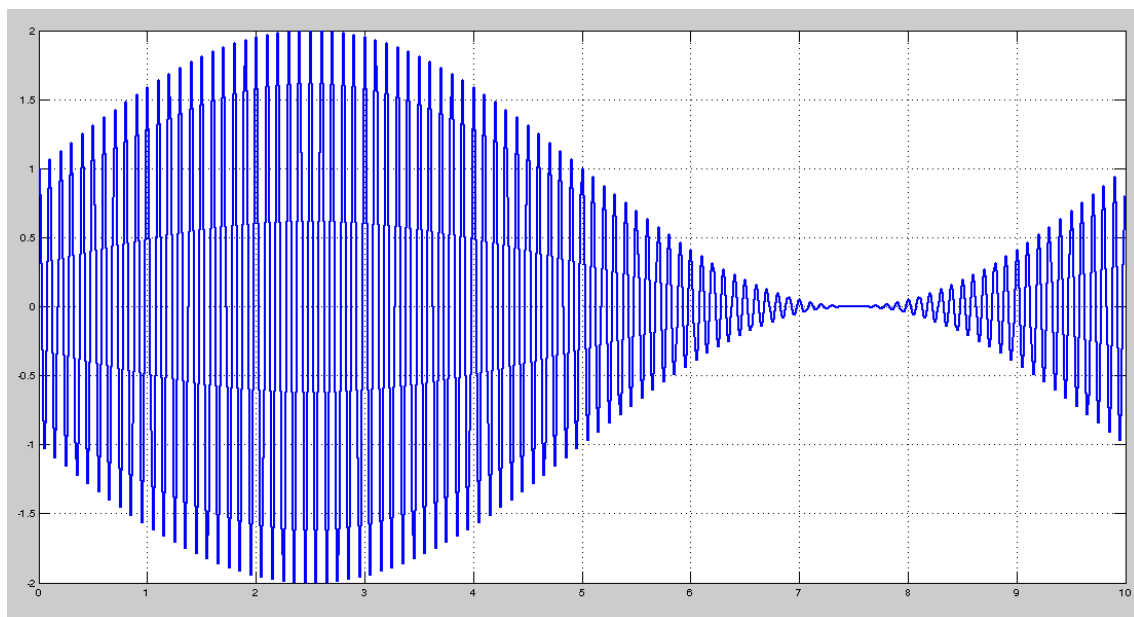


рис. 33. Модуляция с  $M=1$

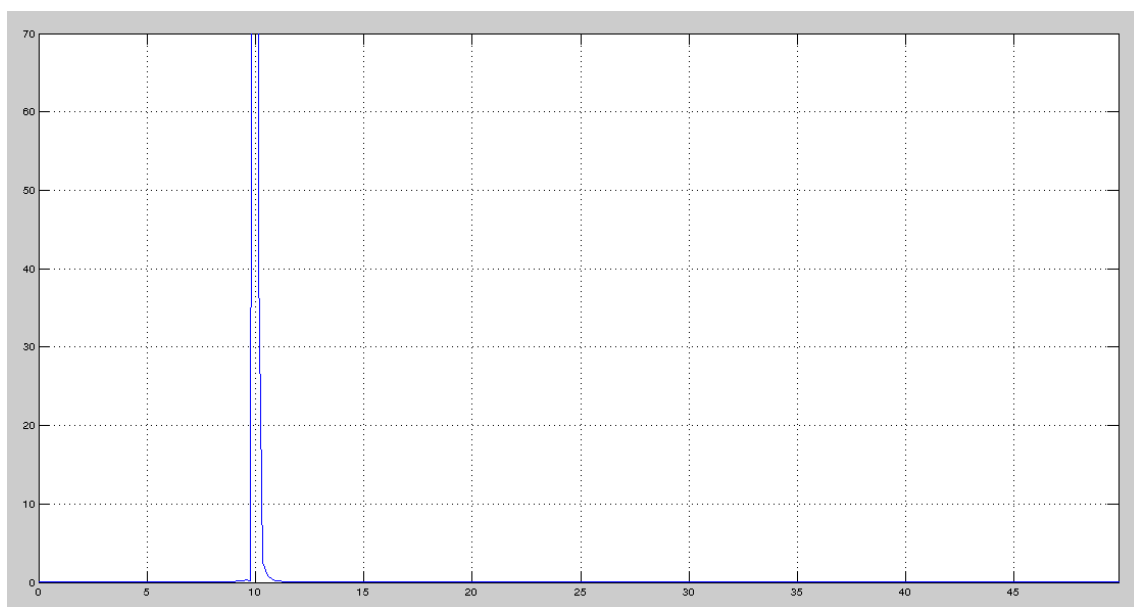


рис. 34. Спектр сигнала с  $M = 1$

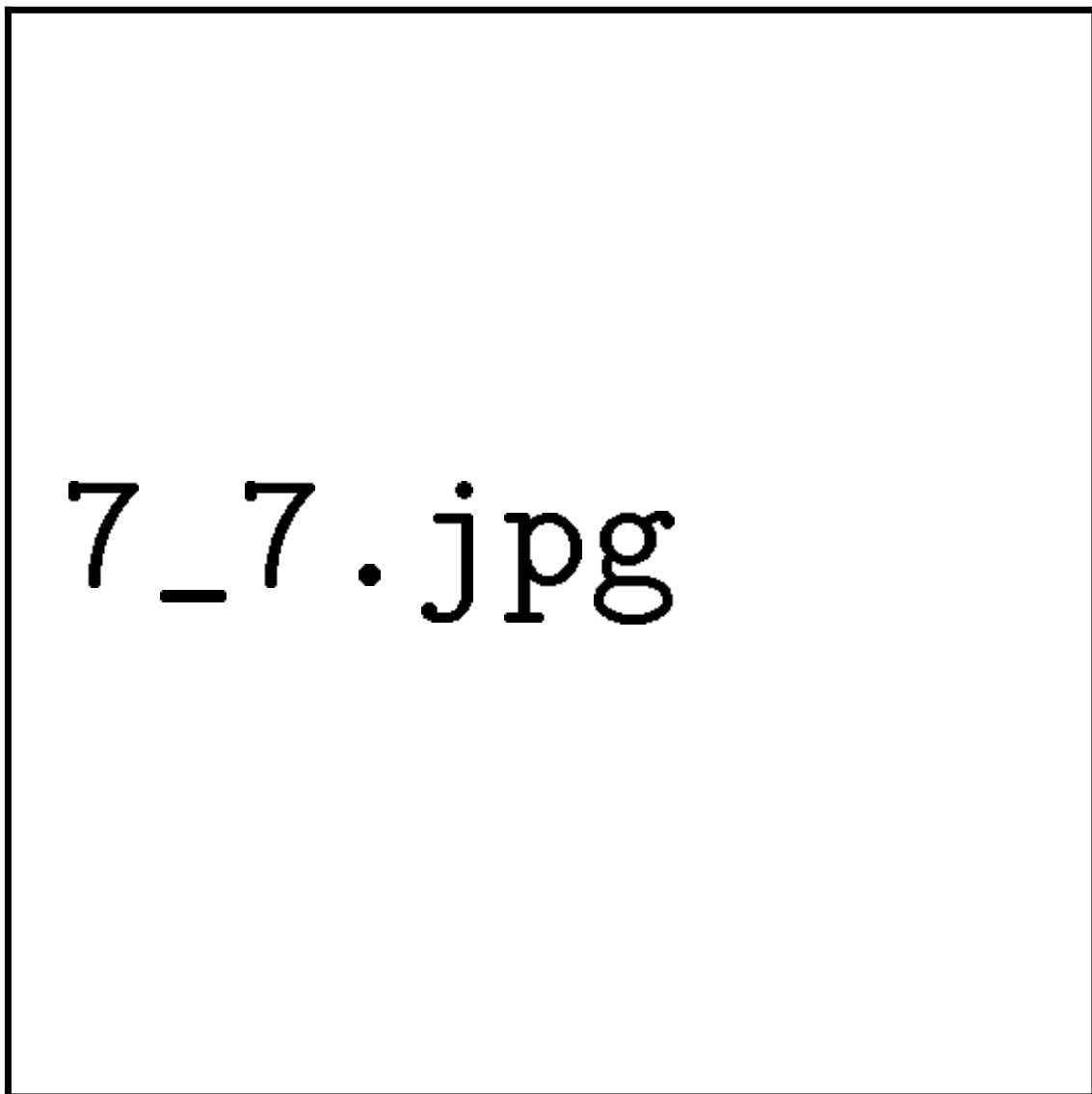


рис. 35. Сигнал с  $M = 5$

Для глубины модулирования 1 КПД равен 0.1111 Моделирование в среде Simulink

#### 4.5 Вывод

В данной лабораторной работе был сгенерирован низкочастотный гармонический сигнал и проведена его амплитудная модуляция, балансная амплитудная модуляция, однополосная амплитудная модуляция, а так же последующая демодуляция. Все результаты соответствуют ожиданиям. Амплитудная модуляция применяется на низких частотах (не выше коротких

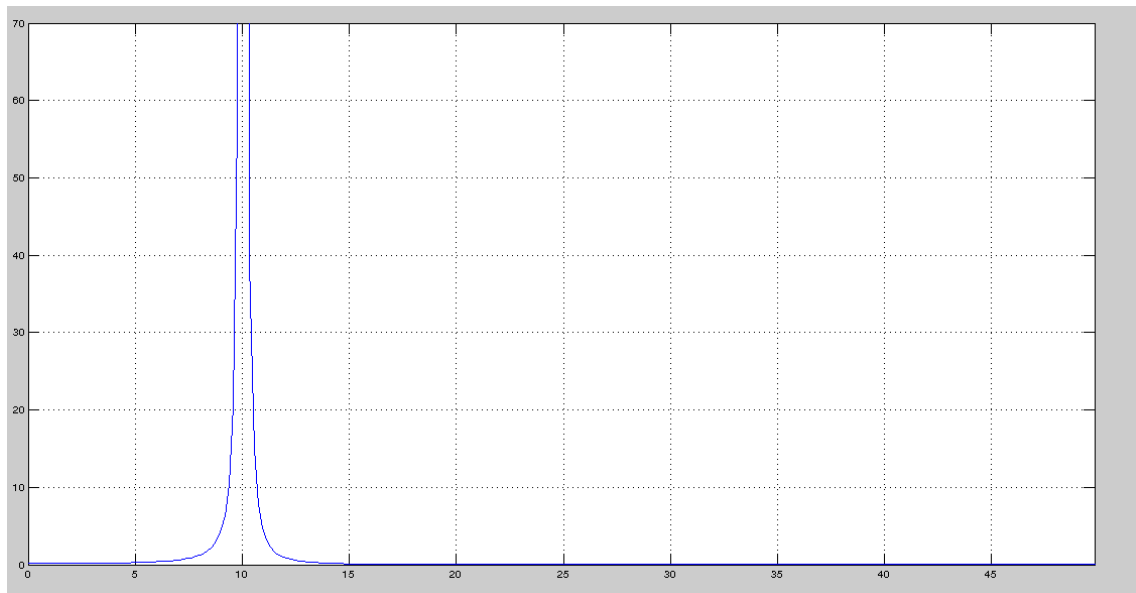


рис. 36. Спектр сигнала с  $M = 5$

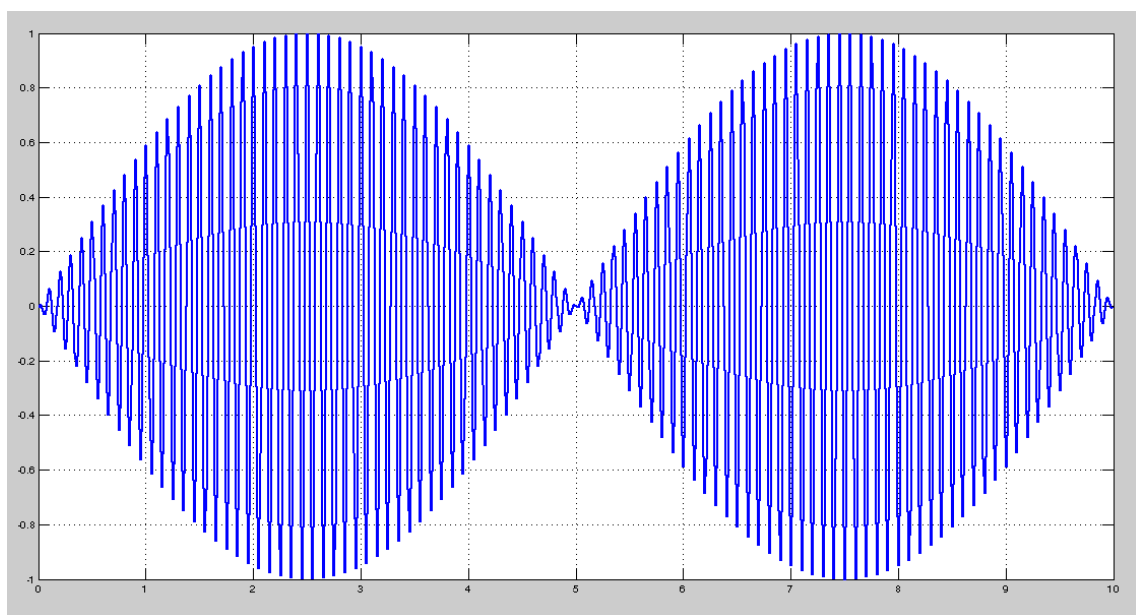


рис. 37. Модуляция с подавлением несущей

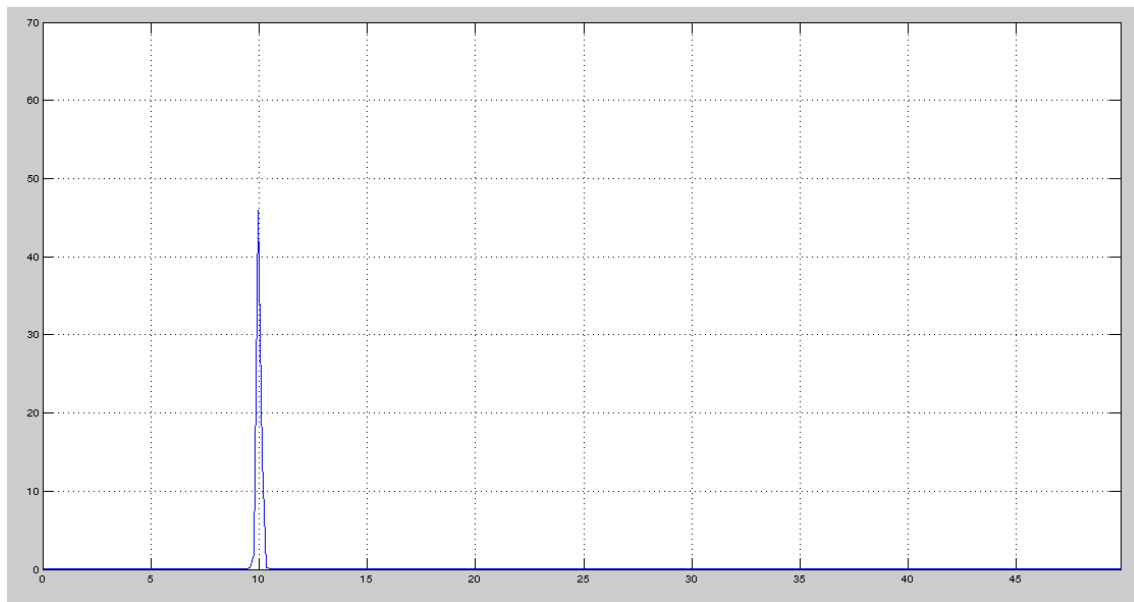


рис. 38. Спектр сигнала с подавлением несущей

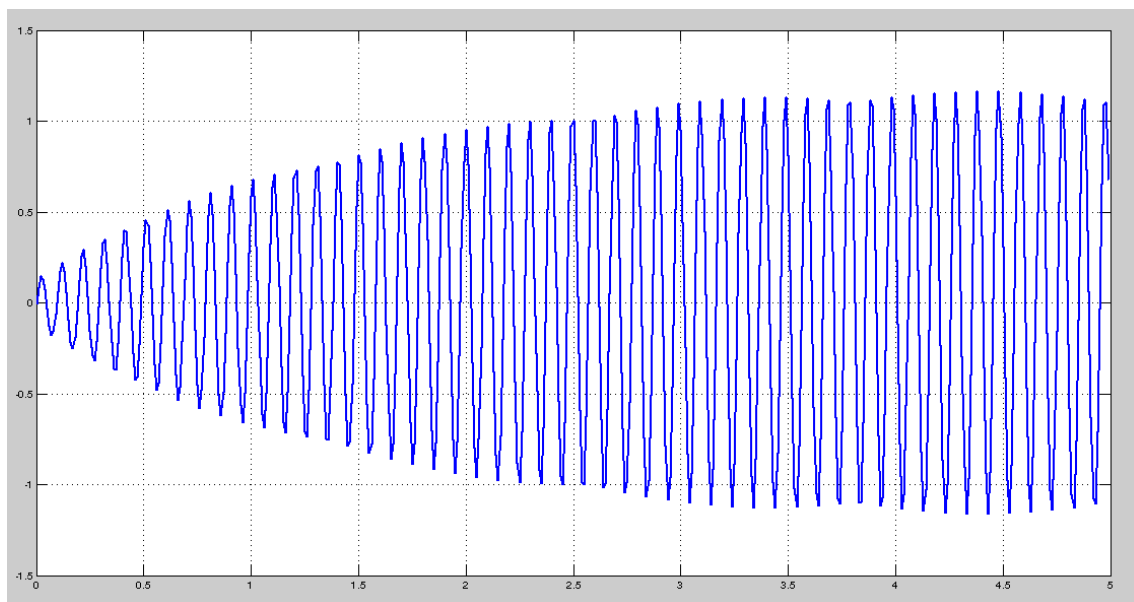


рис. 39. Однополосный АМ-сигнал

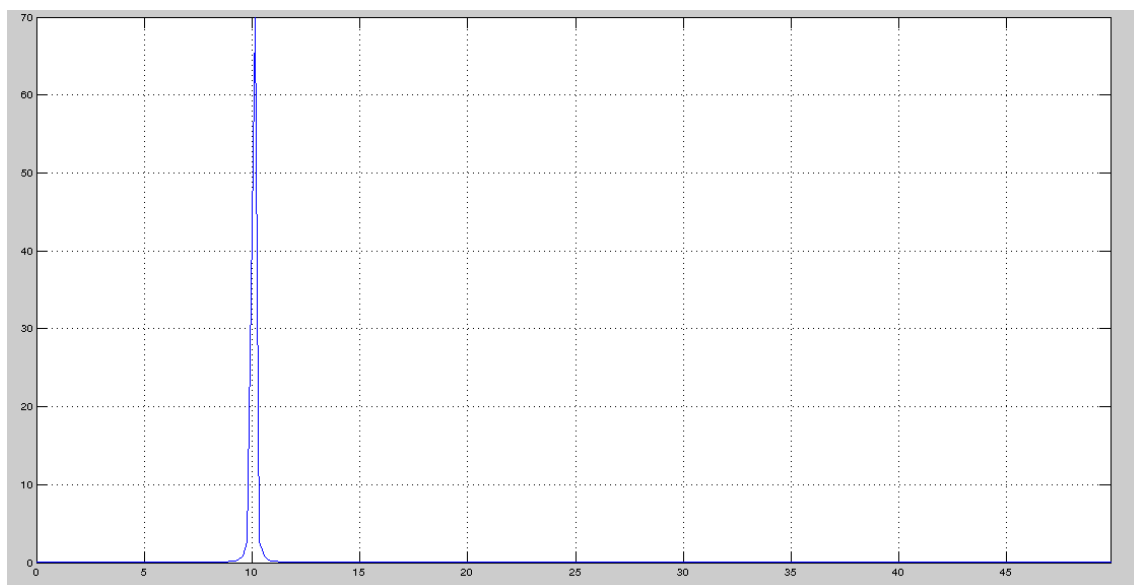


рис. 40. Спектр однополосного модулированного сигнала

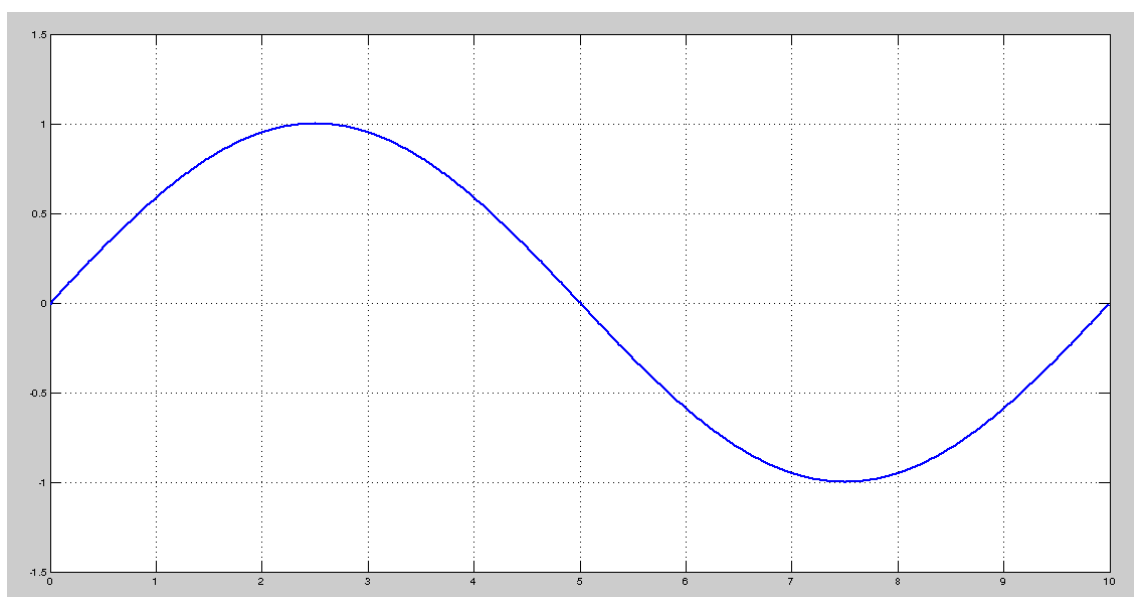


рис. 41. Демодулированный сигнал

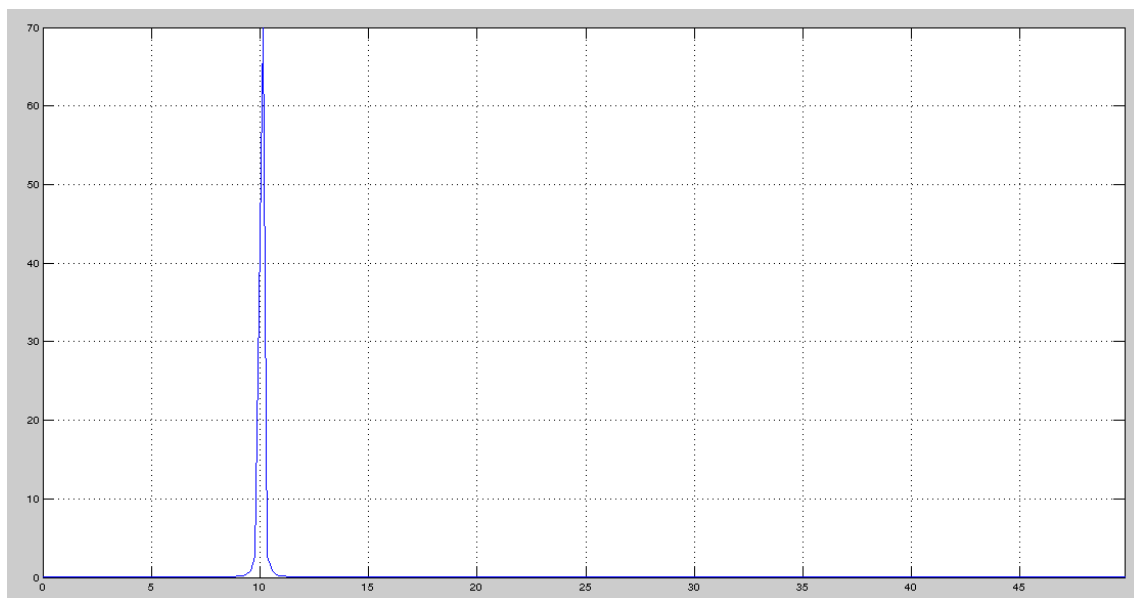


рис. 42. Спектр демодулированного сигнала

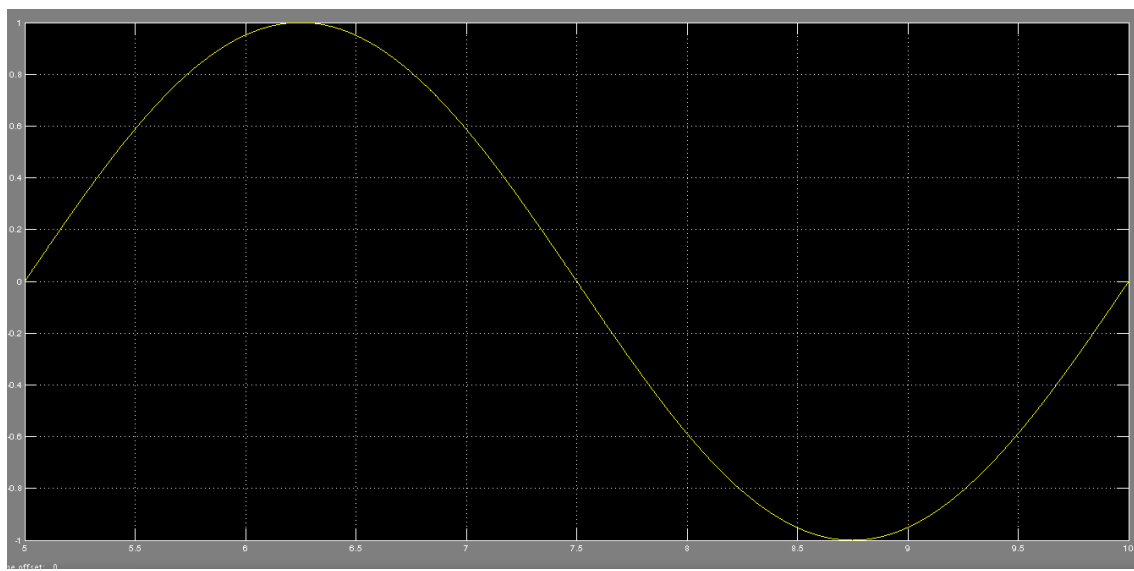


рис. 43. Исходный сигнал



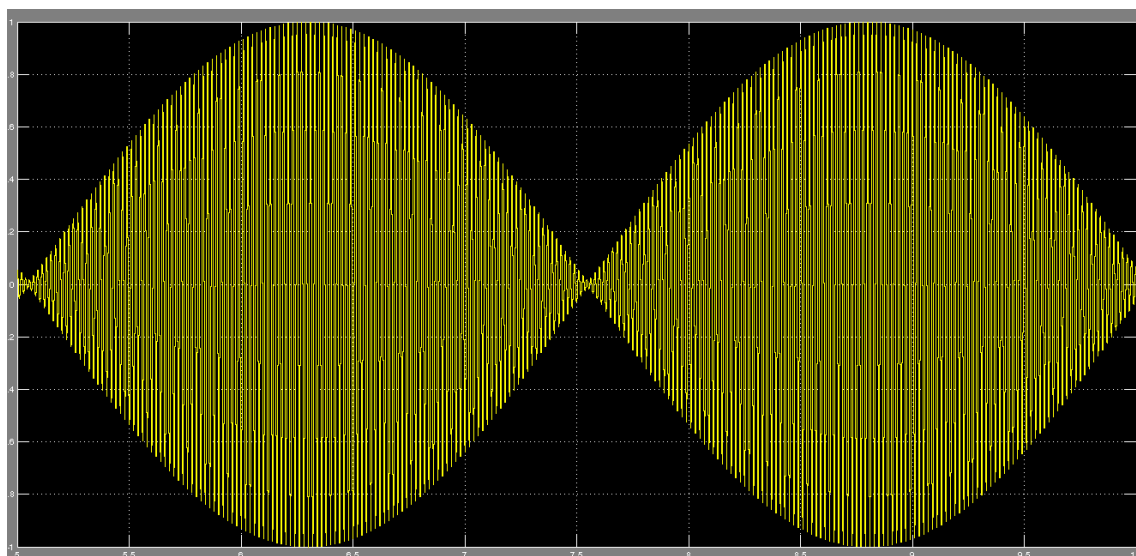


рис. 44. Модулированный сигнал

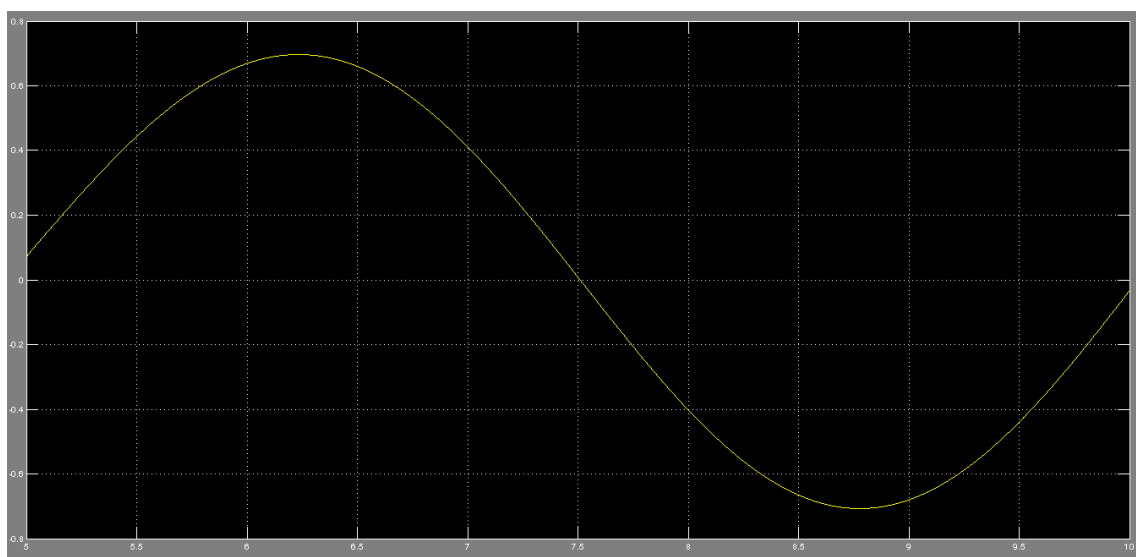


рис. 45. Демодулированный сигнал

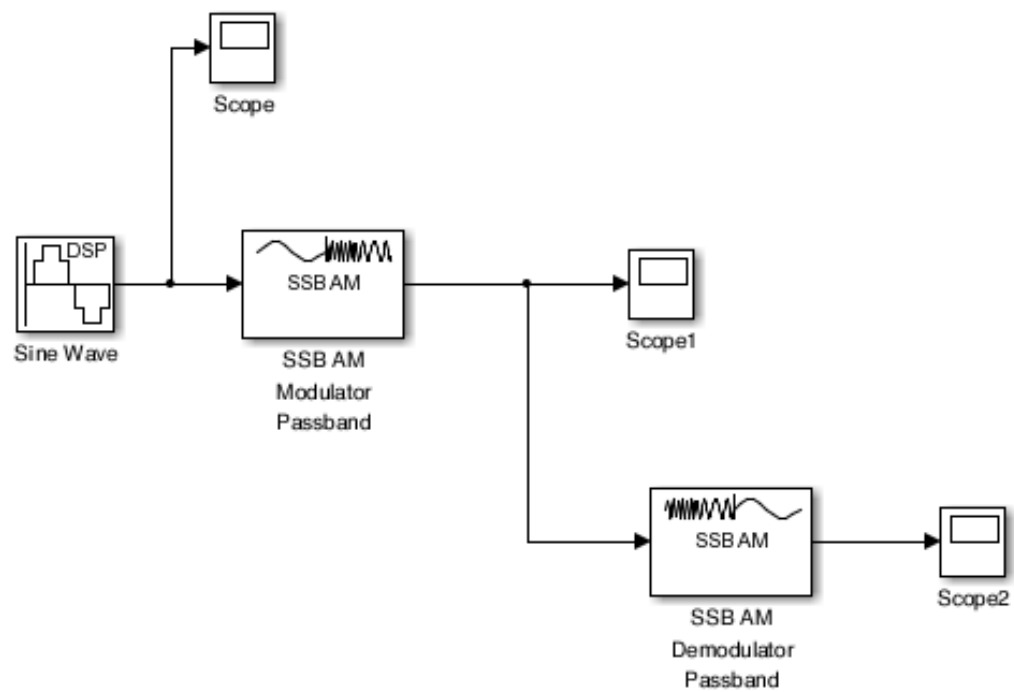


рис. 46. Модель

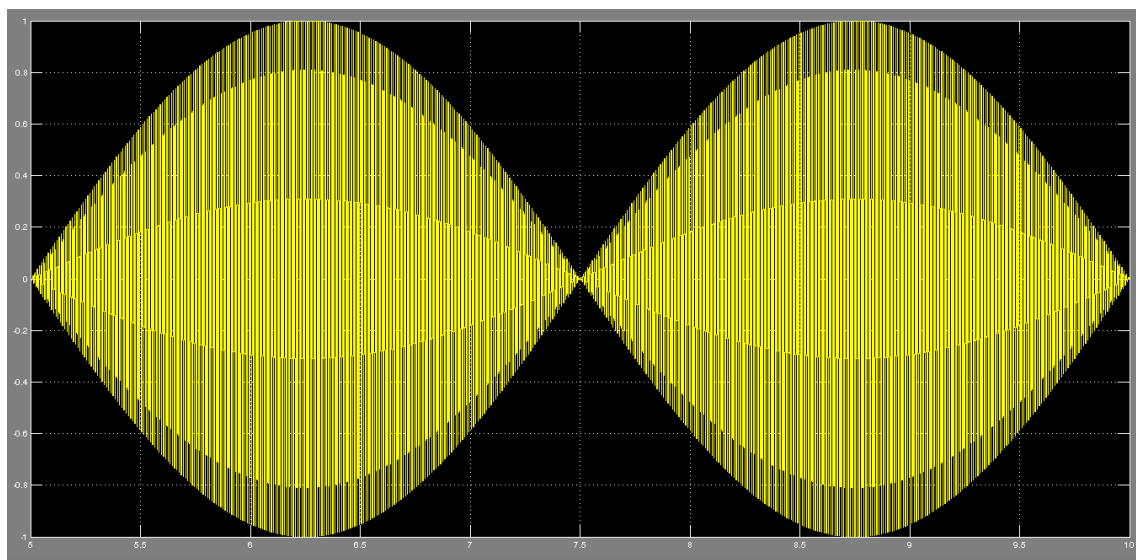


рис. 47. Сигнал БАМ

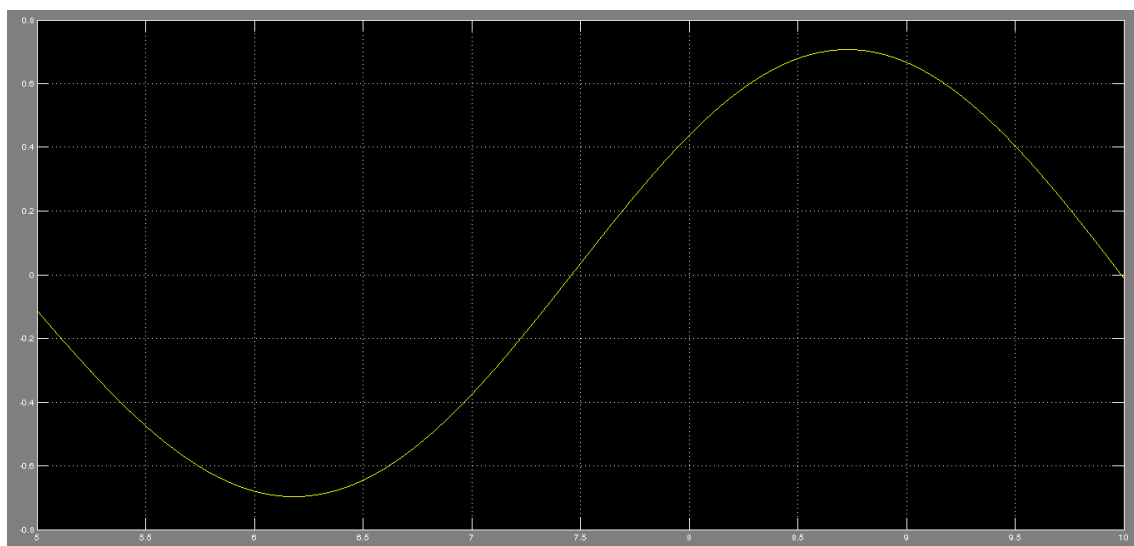


рис. 48. Демодулированный БАП сигнал

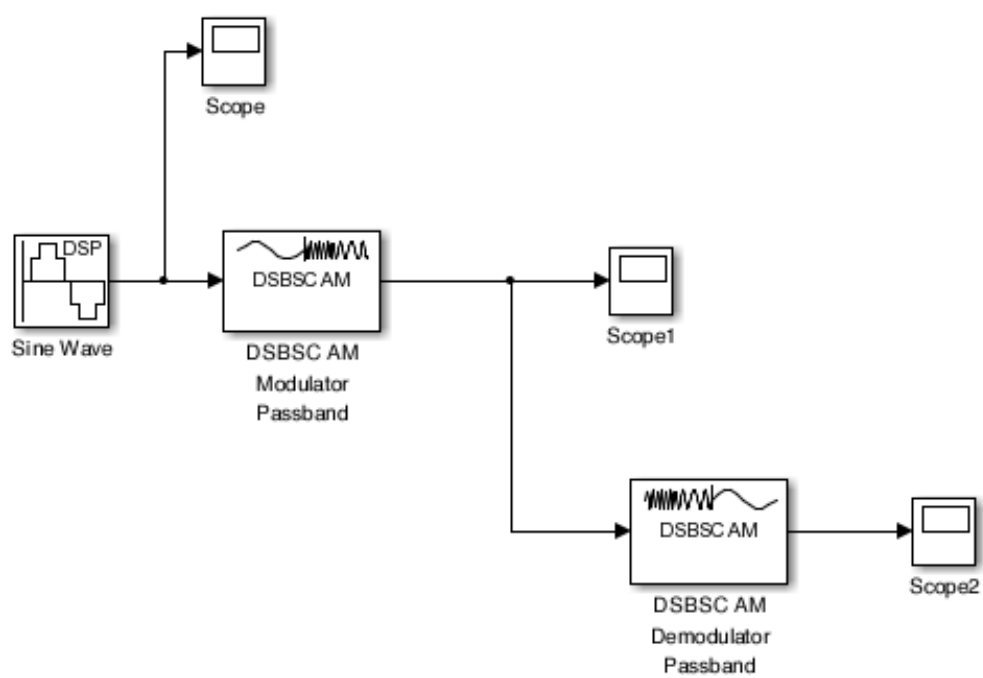


рис. 49. Модель

волн), 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. Рассмотренная в предыдущем разделе двухполосная АМ с подавленной несущей имеет преимущества перед обычной АМ только в энергетическом плане - за счет устранения несущего колебания. Ширина спектра при этом по-прежнему вдвое больше, чем у модулирующего сигнала. Однако 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. Получающаяся модуляция называется однополосной (английский термин - Single Side Band, SSB). В зависимости от того, какая боковая полоса сохраняется, говорят об однополосной модуляции с использованием верхней или нижней боковой полосы. Итак, однополосный сигнал можно представить как сумму двух АМ-сигналов, несущие колебания которых имеют одну и ту же частоту, но сдвинуты по фазе друг относительно друга на 90. Амплитудными функциями этих АМ-сигналов являются модулирующий сигнал и его квадратурное дополнение. В зависимости от того, складываются эти два АМ-сигнала или вычитаются (а точнее, какая из двух несущих опережает другую по фазе), формируется однополосный сигнал с верхней или нижней боковой полосой. Синхронное детектирование является одним из способов демодуляции АМ-сигнала. Его суть состоит в умножении частоты сигнала на опорное колебание с несущей частотой. Результат умножения содержит два слагаемых: искомая амплитуда и АМ-сигнал с несущей частотой  $2\omega$ , который легко удаляется путем пропускания сигнала через ФНЧ (например через фильтр Баттерворта).

## 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 `pmmmod`, `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 `fmmmod`, `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

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

### 5.5 Вывод

Различие между фазовой и частотной модуляцией обнаруживается при модуляции спектром частот. При фазовой модуляции индекс модуляции не зависит от частоты модуляции ( $\beta = \text{const}$ ), а девиация частоты пропорциональна частоте модуляции ( $DW = mW$ ). При частотной модуляции девиация

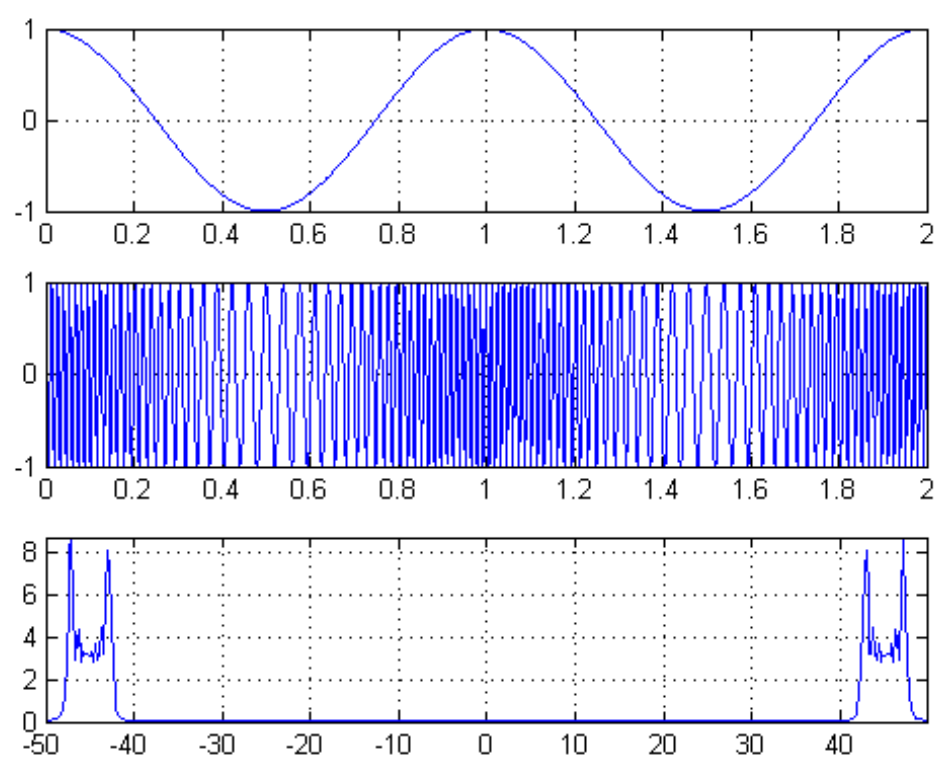


рис. 50. Частотная модуляция сигнала

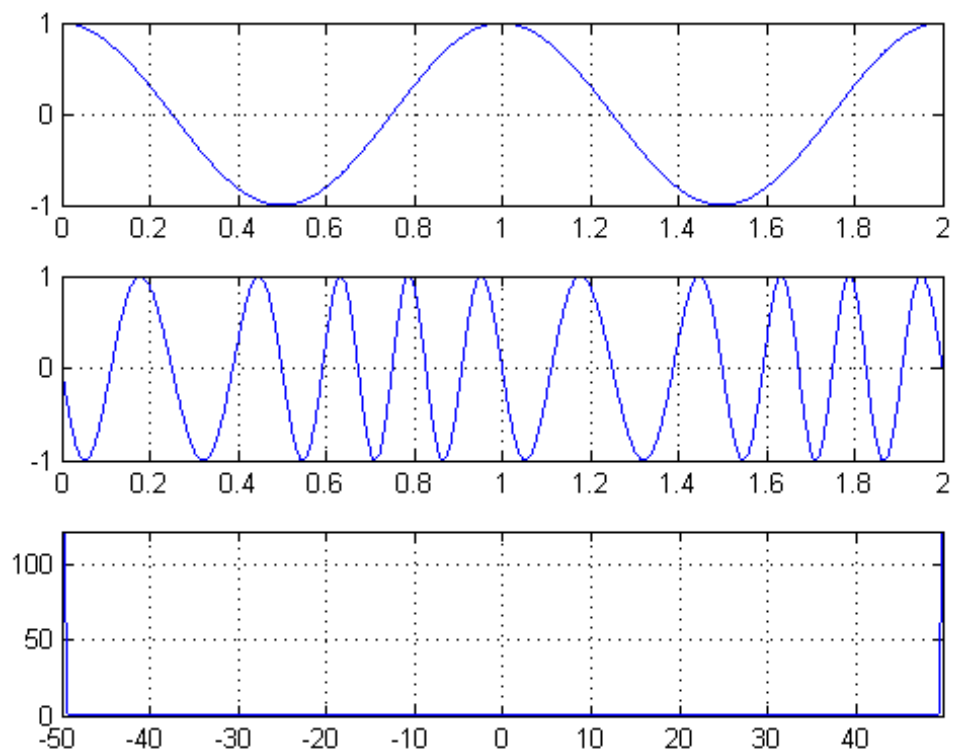


рис. 51. Фазовая модуляция сигнала

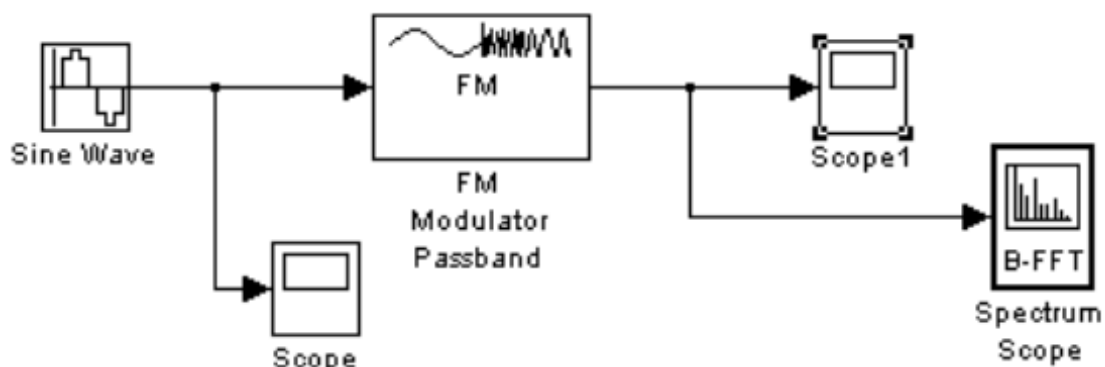


рис. 52. Модель для частотной модуляции



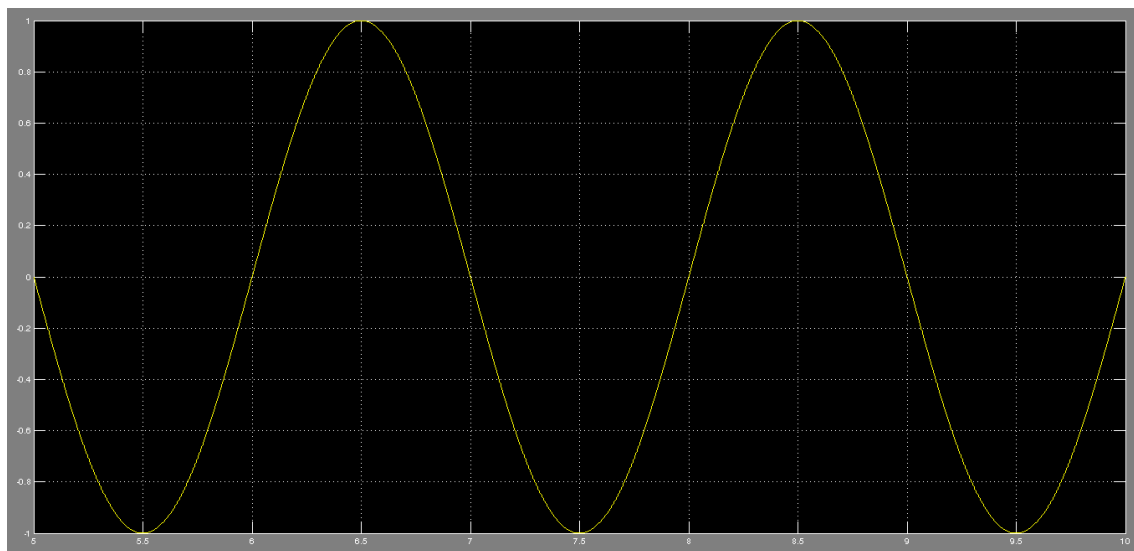


рис. 53. Исходный сигнал

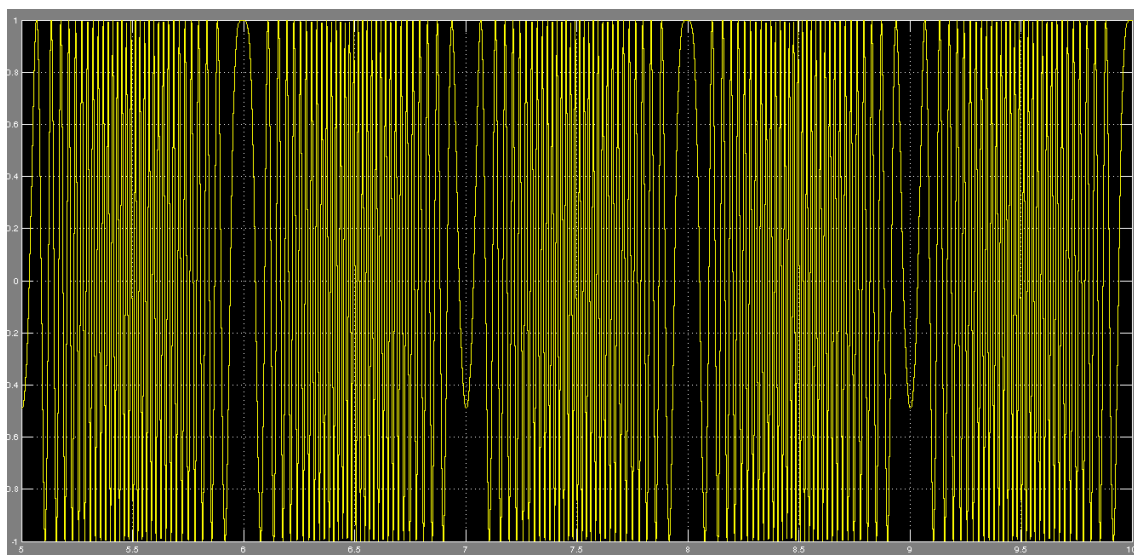


рис. 54. Моделированный сигнал

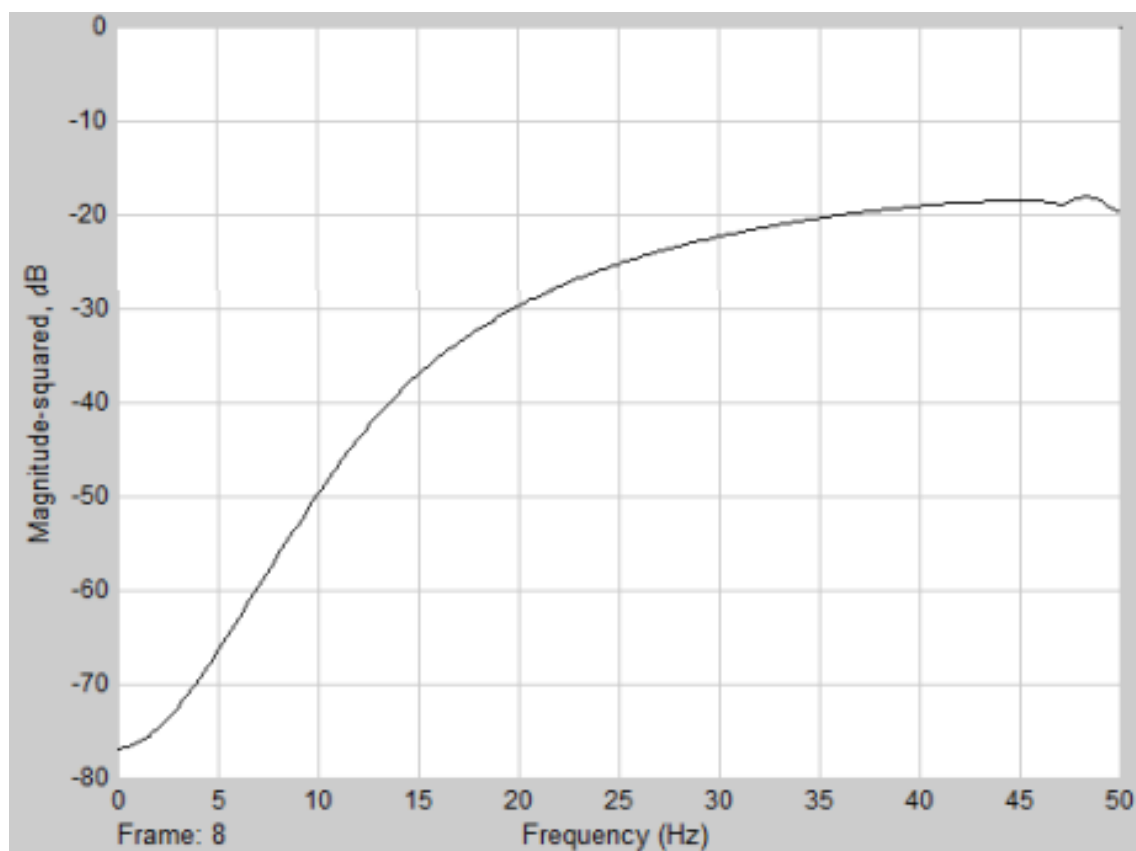


рис. 55. Спектр моделированного сигнала

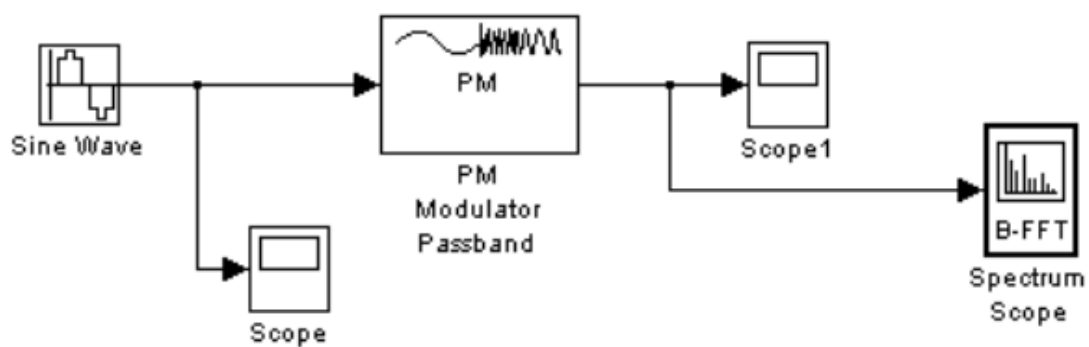


рис. 56. Модель для фазовой модуляции

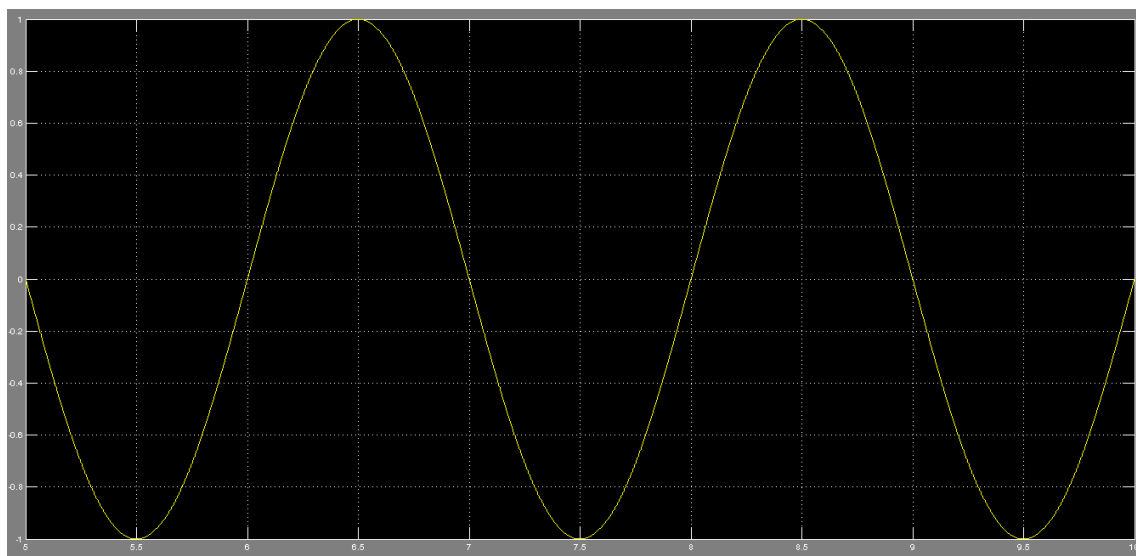


рис. 57. Исходный сигнал

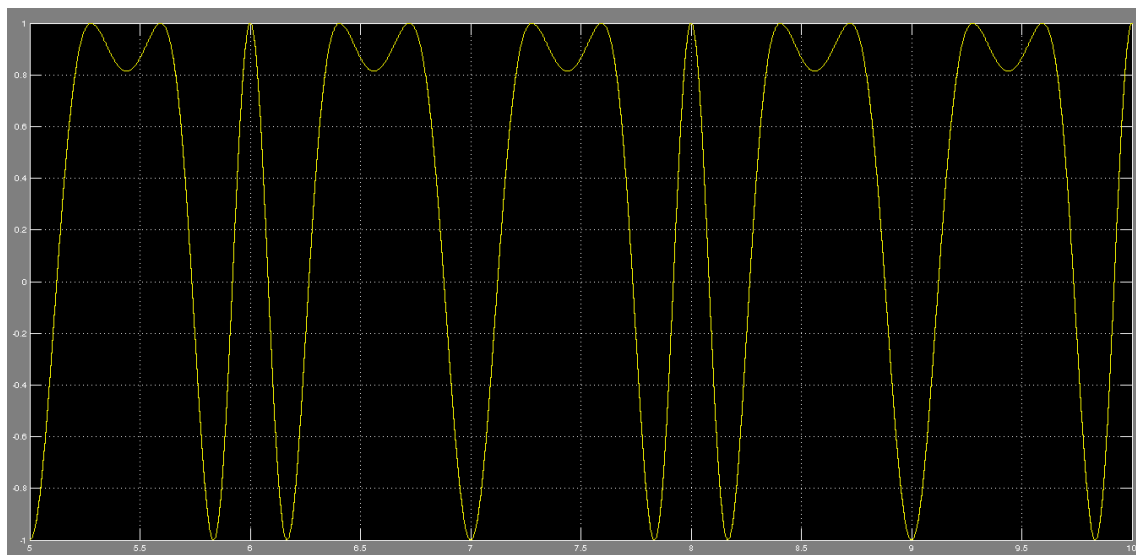


рис. 58. Моделированный сигнал

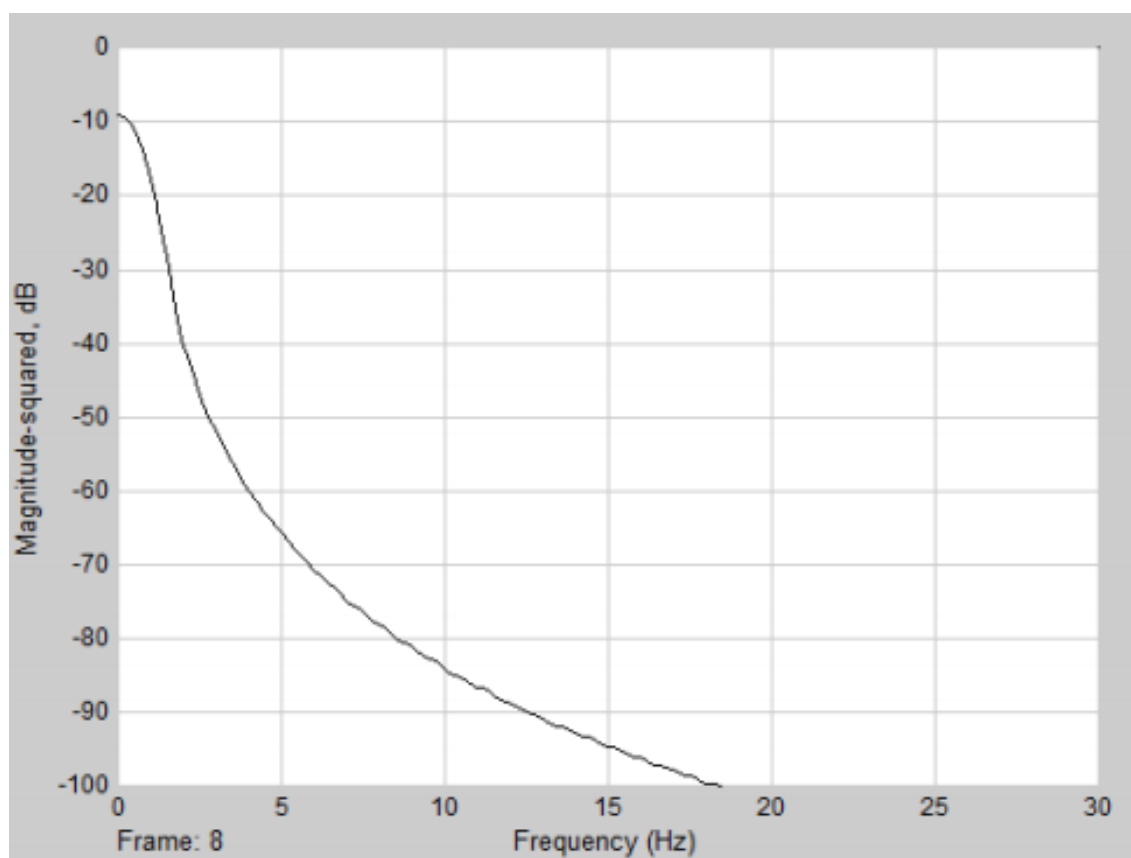


рис. 59. Спектр моделированного сигнала

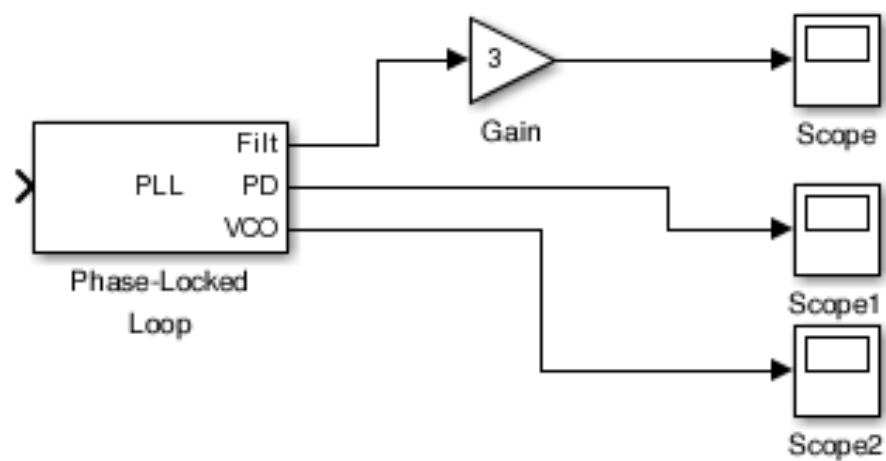


рис. 60. Модель фазовой автоподстройки частоты в режиме слежения

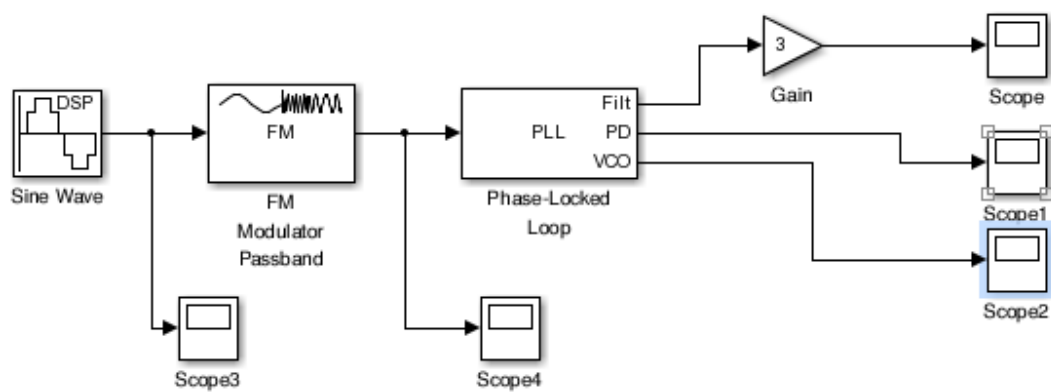


рис. 61. Модель фазовой автоподстройки частоты в режиме захвата и удержания сигнала

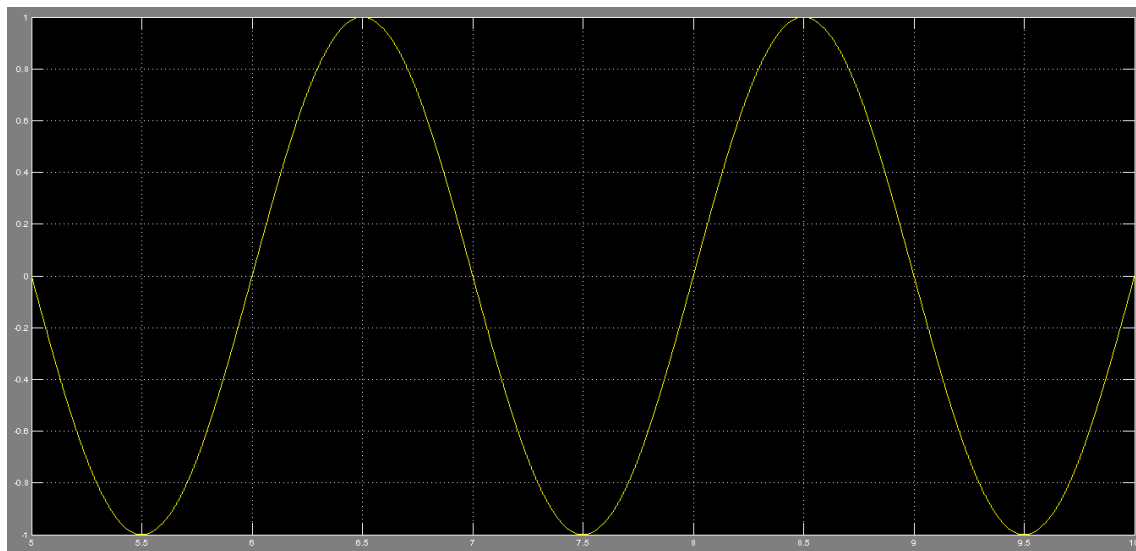


рис. 62. Исходный сигнал

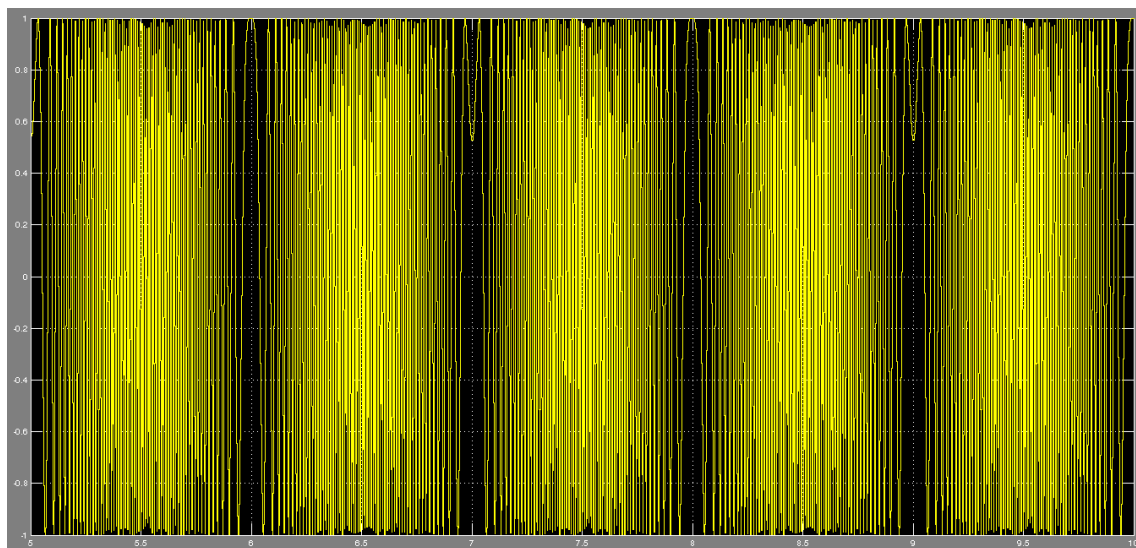


рис. 63. Моделированный сигнал



рис. 64. Сигнал на выходе ФНЧ

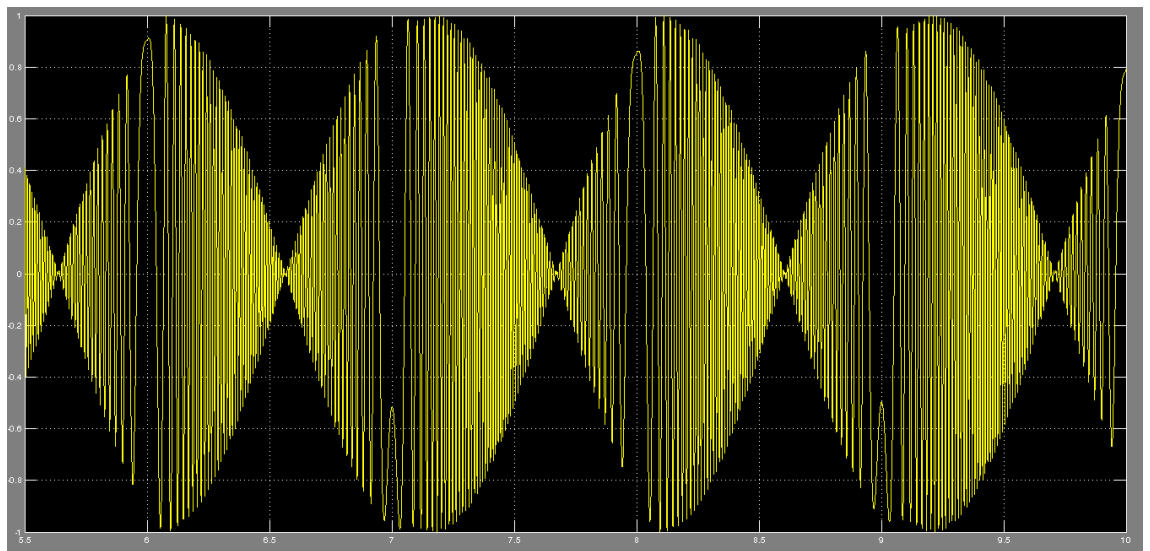


рис. 66. Сигнал на выходе фазового детектора

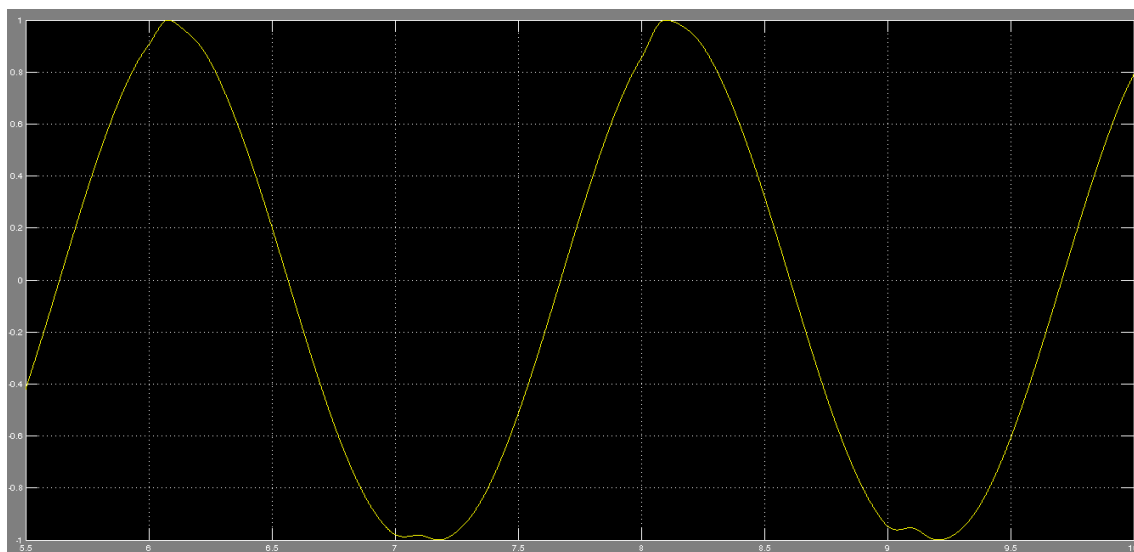


рис. 67. Сигнал на выходе ГУН

частоты не зависит от частоты модуляции ( $D_w = \text{const}$ ), а индекс модуляции обратно пропорционален частоте модуляции  $m = D_w / W$ . 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  $s(t)$  формы сигналов ФМ и ЧМ вообще практически не отличаются. Угловая модуляция, а особенно ЧМ (частотная модуляция), имеет преимущество перед АМ в отношении помехозащищенности (т.к. изменение амплитуды из-за шумов можно легко исключить, ограничив амплитуду перед демодуляцией).



## 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 значения 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. Четырёхпозиционная фазовая модуляция со сдвигом квадратур (OQPSK). В OQPSK биты в потоках, подаваемых на модуляторы квадратур I и Q, переключаются в один и тот же момент времени. Сигнал символьной синхронизации, и синхросигналы битовых потоков, подаваемых на модуляторы I и Q, синхронизированы. Это приводит к тому, что в диаграмме переходов сигнала четырёхпозиционной фазовой модуляции встречаются переходы через точку с нулевой амплитудой сигнала. Битовые потоки, подаваемые на модуляторы квадратур I и Q, сдвинуты друг относительно друга на длительность одного бита.
6. Квадратурная амплитудная модуляция (QAM) — разновидность амплитудной модуляции сигнала, которая представляет собой сумму двух несущих колебаний одной частоты, но сдвинутых по фазе относительно друг друга на 90, каждое из которых модулировано по амплитуде своим модулирующим сигналом.

## 6.3 Код matlab

```
noise_persent = 20;
noise_persent = noise_persent/100;

%BPSK modulation
M=2;
N=10;
h = modem.pskmod('M', M);
g = modem.pskdemod('M', M);
msg = randint(N,1,M);
modSignal = modulate(h,msg);
% errSignal = (randerr(1,N, N*noise_persent) ./ noise_level)';
% modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);
scatterplot(demodSignal);
errSignal = (randerr(1,N, N*noise_persent) ./ noise_level)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
symerr(msg,demodSignal)/length(msg)*100

M=4;
N=10;
%PSK modulation
h = modem.pskmod('M', M);
g = modem.pskdemod('M', M);
msg = randint(N,1,M);
modSignal = modulate(h,msg);
% errSignal = (randerr(1,N, N*noise_persent) ./ noise_level)';
% modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);
scatterplot(demodSignal);
errSignal = (randerr(1,N, N*noise_persent) ./ noise_level)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
symerr(msg,demodSignal)/length(msg)*100

%QPSK modulation
N=200;
M=4;
h = modem.oqpskmod('M',M);
g = modem.oqpskdemod(h);
msg = randint(N,1,M);
modSignal = modulate(h,msg);
% errSignal = (randerr(1,2*N, N*noise_persent) ./ noise_level)';
% modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);
```

```

scatterplot(demodSignal);
errSignal = (randerr(1,2*N, N*noise_persent) ./ noise_level)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
symerr(msg,demodSignal)/length(msg)*100

%GENQAM modulation
M = 10;
N = 10;
h = modem.genqammod('Constellation', exp(1i*2*pi*[0:M-1]/M));
g = modem.genqamdemod('Constellation', exp(1i*2*pi*[0:M-1]/M));
msg = randint(N,1,4);
modSignal = modulate(h,msg);
% errSignal = (randerr(1,N, N*noise_persent) ./ noise_level)';
% modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
scatterplot(modSignal);
scatterplot(demodSignal);
errSignal = (randerr(1,N, N*noise_persent) ./ noise_level)';
modSignal = modSignal + errSignal;
demodSignal = demodulate(g,modSignal);
symerr(msg,demodSignal)/length(msg)*100

N=20;
%MSK modulation
h = modem.mskmod('SamplesPerSymbol', N);
g = modem.msksdemod('SamplesPerSymbol', N);
msg = randint(N,1,2);
modSignal = modulate(h, msg);
% errSignal = (randerr(1,N*N, N*noise_persent) ./ noise_level)';
% modSignal = modSignal + errSignal;
demodSignal = demodulate(g, modSignal);
scatterplot(modSignal);
scatterplot(demodSignal);
demodSignal = demodulate(g,modSignal);
errSignal = (randerr(1,N*N, N*noise_persent) ./ noise_level)';
modSignal = modSignal + errSignal;
symerr(msg,demodSignal)/length(msg)*100

```

рис. 87. Сигналы

## 6.4 Результаты работы

Смоделируем проделанные опыты при помощи simulink

## 6.5 Вывод

Уровень модуляции определяет количество состояний несущей, используемых для передачи информации. Чем выше этот уровень, тем большими скоростными возможностями и меньшей помехоустойчивостью модуляция

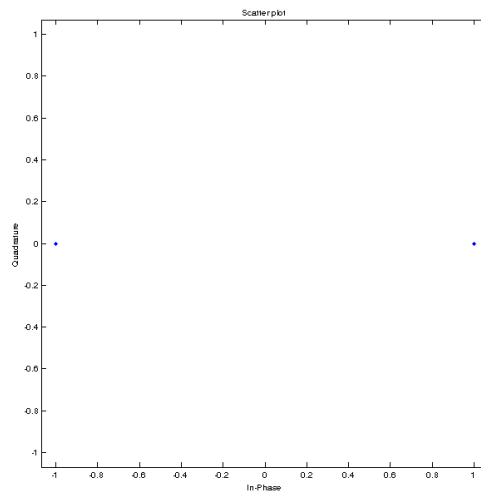


рис. 68. Сигнальное созвездие BPSK-модуляции

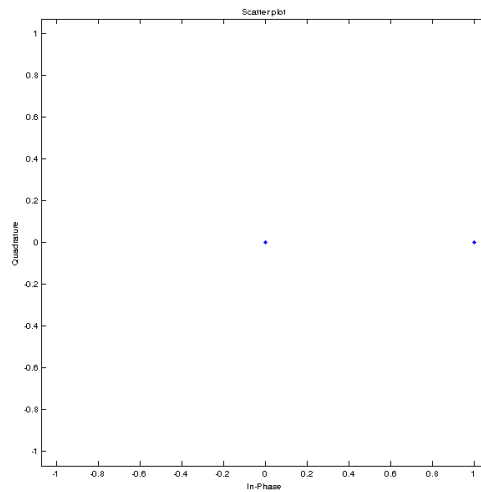


рис. 69. Сигнальное созвездие BPSK-демодуляции

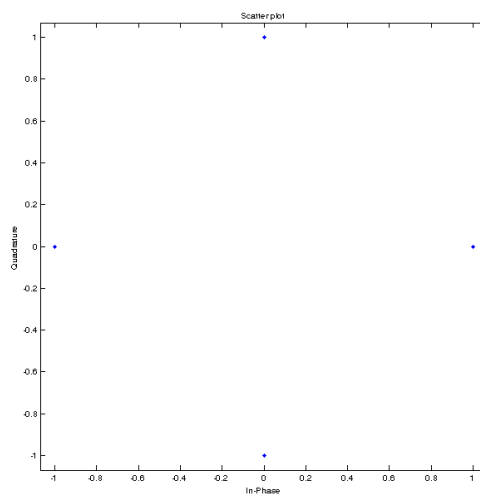


рис. 70. Сигнальное созвездие PSK-модуляции

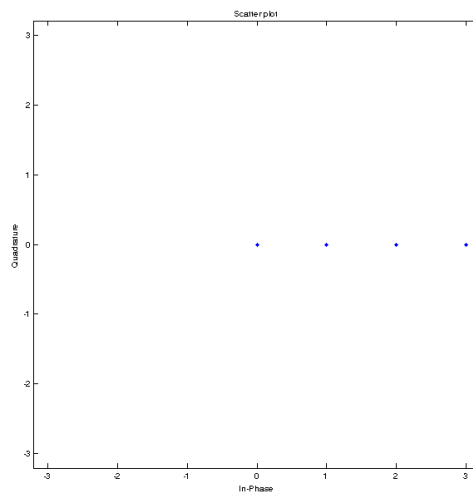


рис. 71. Сигнальное созвездие PSK-демодуляции

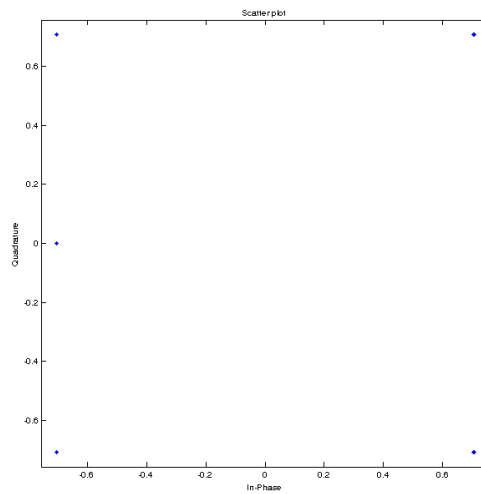


рис. 72. Сигнальное созвездие OQPSK-модуляции

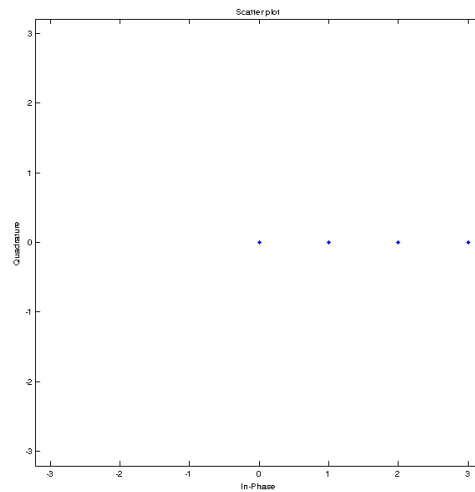


рис. 73. Сигнальное созвездие OQPSK-демодуляции

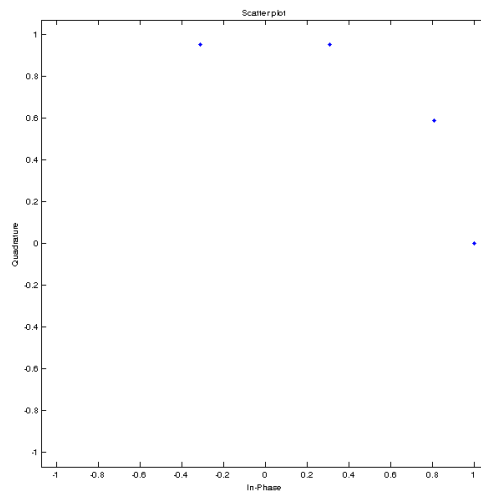


рис. 74. Сигнальное созвездие genQAM-модуляции

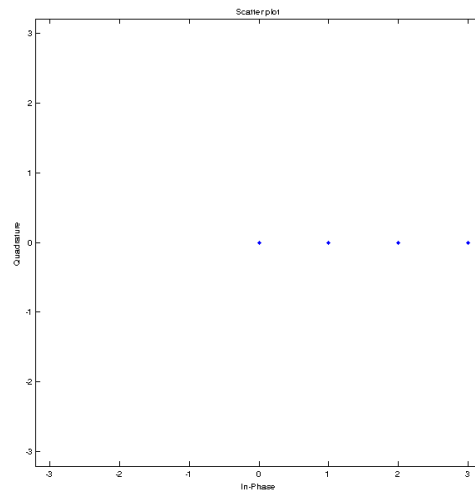


рис. 75. Сигнальное созвездие genQAM-демодуляции



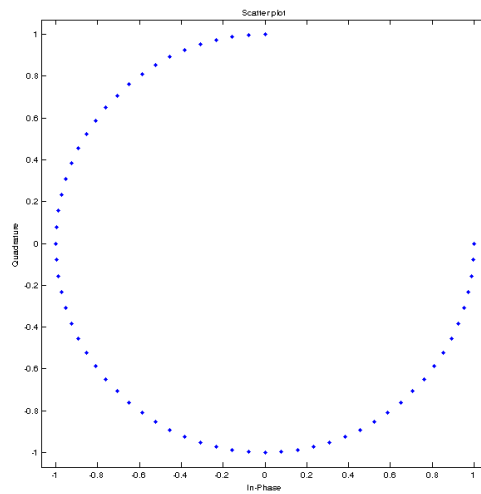


рис. 76. Сигнальное созвездие MSK-модуляции

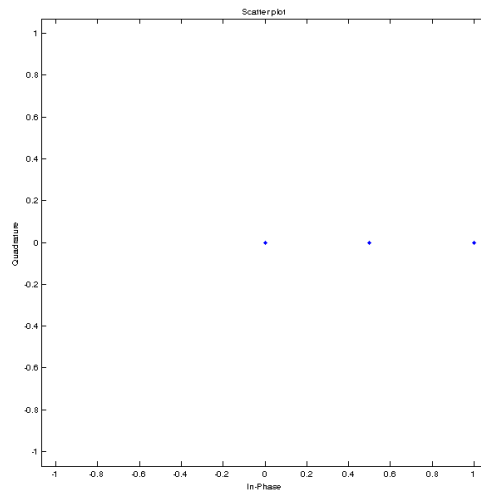


рис. 77. Сигнальное созвездие MSK-демодуляции

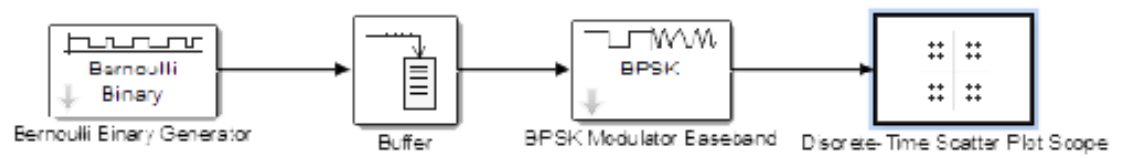


рис. 78. Модель BPSK-модуляции

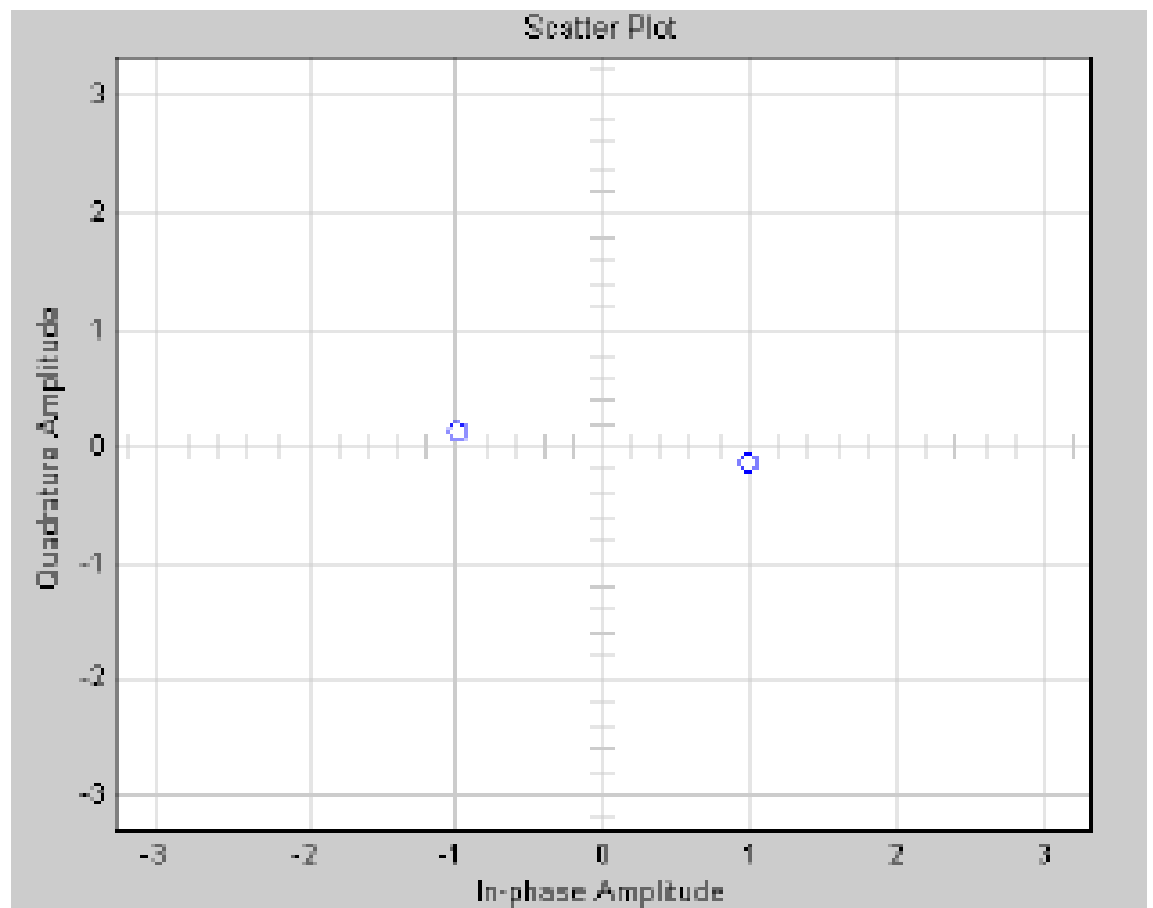


рис. 79. Сигнальное созвездие BPSK-модуляции

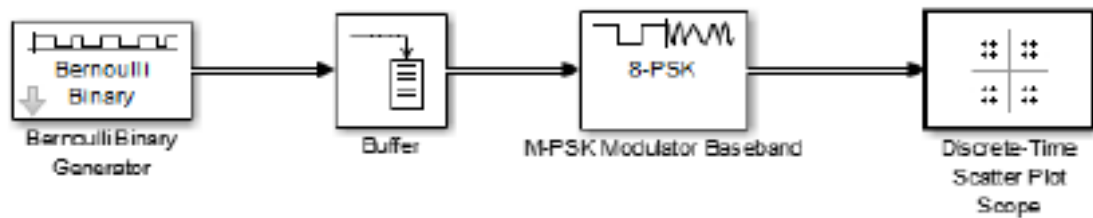


рис. 80. Модель PSK-модуляции

рис. 81. Сигнальное созвездие PSK-модуляции.

рис. 82. Модель OQPSK-модуляции

рис. 83. Сигнальное созвездие OQPSK-модуляции.

рис. 84. Модель genQAM-модуляции

рис. 85. Сигнальное созвездие genQAM-модуляции

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.