

Санкт-Петербургский Политехнический Университет Петра
Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Телекоммуникационные технологии

Отчет по лабораторной работе №1,2

Сигналы телекоммуникационных систем

Ряд Фурье. Преобразование Фурье. Корреляция

Работу выполнил:

Вашуров А., гр 33501.4

Преподаватель:

Богач Н.В.

Санкт-Петербург
2017

Содержание

1. Цель работы.....	3
2. Постановка задачи.....	3
3. Теоретический раздел.....	3
4. Ход работы.....	6
4.1 Генерация сигналов.....	6
4.2 Сравнение методов корреляции.....	23
5. Вывод.....	23
6. Приложение.....	24

1. Цель работы

Познакомиться со средствами генерации и визуализации простых сигналов.

Получить представление о спектрах телекоммуникационных сигналов

2. Постановка задачи

Промоделировать сигналы в командном окне MATLAB

Для данных сигналов выполнить расчет преобразования Фурье, получить спектры.

С помощью функции корреляции найти позицию синхропосылки [101] в сигнале [0001010111000010]. Получить пакет данных, если известно, что его длина составляет 8 бит без учета синхропосылки. Вычислить корреляцию прямым методом, воспользоваться алгоритмом быстрой корреляции, сравнить время работы обоих алгоритмов.

3. Теоретический раздел

3.1 Аналоговый сигнал

Аналоговый сигнал с математической точки зрения представляет собой функцию (как правило – времени), и при его дискретизации мы получаем отсчеты, являющиеся значениями этой функции, вычисленными в дискретные моменты времени. Поэтому для расчета дискретизированного сигнала необходимо сформировать вектор дискретных значений времени. Для этого удобно задать частоту дискретизации и использовать обратную величину в качестве шага временного ряда. Сформировав вектор опорных значений времени, можно вычислять значения сигнала, используя этот вектор в соответствующих формулах.

3.2 Необходимые формулы

Гармонический сигнал

$s1=A*\cos(2*\pi*f0*t+\phi)$, где A – амплитуда, $f0$ – частота, ϕ – начальная фаза

Затухающая синусоида

$s2=\exp(-\alpha*t)*s1$, где α – скорость затухания

Многоканальный сигнал

$s = \cos(2 \cdot \pi \cdot t \cdot f)$, где t – время, f – частота

Кусочные зависимости:

Односторонний экспоненциальный импульс

$s = A \cdot \exp(-\alpha \cdot t)$, при $t \geq 0$

Прямоугольный импульс, центрированный относительно начала отсчета времени

$s = A \cdot |t|$, при $|t| \leq T/2$

Несимметричный треугольный импульс

$s = A \cdot t/T$, при $0 \leq t \leq T$

Одиночный прямоугольный импульс

$$y = \begin{cases} 1, & -\frac{\text{width}}{2} \leq t < \frac{\text{width}}{2}, \\ 0, & t < -\frac{\text{width}}{2}, \quad t \geq \frac{\text{width}}{2}. \end{cases}$$

, где width – ширина импульса

Одиночный треугольный импульс

$$y = \begin{cases} \frac{2t + \text{width}}{\text{width}(\text{skew} + 1)}, & -\frac{\text{width}}{2} \leq t < \frac{\text{width} \cdot \text{skew}}{2}, \\ \frac{2t - \text{width}}{\text{width}(\text{skew} - 1)}, & \frac{\text{width} \cdot \text{skew}}{2} \leq t < \frac{\text{width}}{2}, \\ 0, & |t| > \frac{\text{width}}{2}. \end{cases}$$

, где width – ширина импульса, skew – коэффициент асимметрии импульса, определяющий положение его вершины (от -1 до 1)

Импульс с ограниченной полосой частот

$y = \sin(\pi \cdot x) / (\pi \cdot x)$

Гауссов радиоимпульс

$y = \exp(-a \cdot t^2) \cos(2 \cdot \pi \cdot f_c \cdot t)$, где f_c – несущая частота в Герцах

Функция Дирихле

$\text{diric}_n(x) = \sin(nx/2) / (n \cdot \sin(x/2))$

Также будем моделировать последовательности импульсов с помощью специальных функций MATLAB: `pulstran`, `square` и `sawtooth`.

3.3 Преобразования Фурье

Для построения спектра сигнала применяется преобразование Фурье.

Формула прямого преобразования Фурье:

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt$$

Формула обратного преобразования:

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{j\omega t} d\omega$$

3.4 Взаимная корреляция

Для нахождения синхропосылки в сигнале используется метод взаимной корреляции. Значение корреляции двух векторов x и y строится по формуле:

$$R = \frac{1}{N} \sum_{i=1}^N x_i * y_i$$

Если искомая посылка y короче передаваемого вектора x , то она дополняется нулями до необходимой длины. Если вектор x это сигнал, вектор y – искомая посылка, а N – длина сигнала, то для поиска позиции посылки в сигнале N раз повторяется вычисление

корреляции между x и y и сдвиг y . Максимальные значения в полученном векторе корреляций соответствуют сдвигу, при котором была найдена часть сигнала x максимально похожая на искомую посылку.

Для ускорения вычисления корреляции, особенно на больших посылках, применяется метод быстрой корреляции:

$$R = \frac{1}{N} F_D^{-1} [X^* * Y]$$

где X^* - комплексно-сопряженный вектор от вектора преобразования Фурье от посылки x , Y – результат преобразования Фурье от вектора искомой посылки, F_D^{-1} – обратное преобразование Фурье.

Данная формула позволяет найти вектор значений взаимной корреляции двух векторов быстрее, чем обычный алгоритм.

4. Ход работы

Листинг программы представлен в Приложении.

4.1 Генерация сигналов

Генерация затухающего синусоидального сигнала:

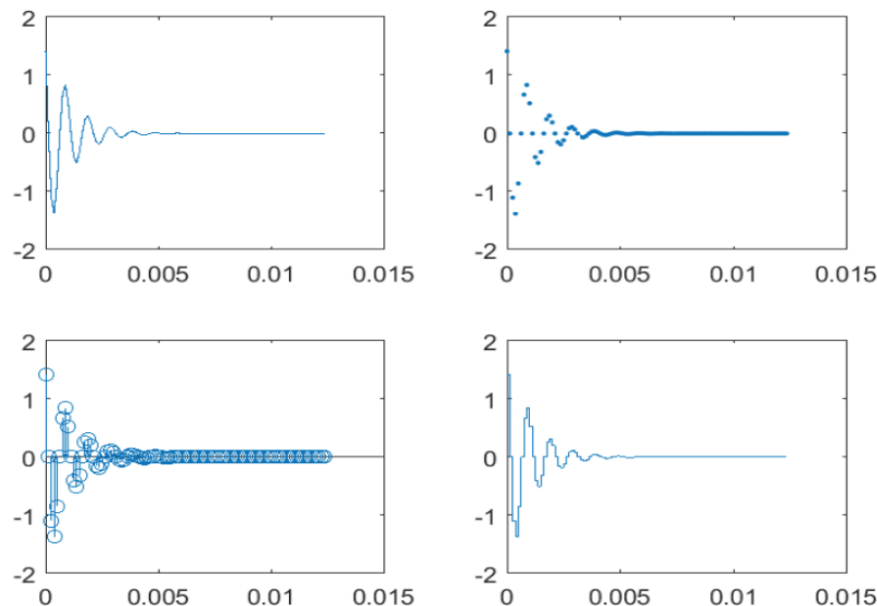


Рисунок 1.1

На рисунке 1.1 представлены результаты моделирования затухающего синусоидального сигнала в различных формах – обычный вид, точечное представление, в виде «стебельков» с помощью функции `stem`, а также в ступенчатом виде с помощью функции `stairs`. На рисунке 1.2 – спектр сигнала.

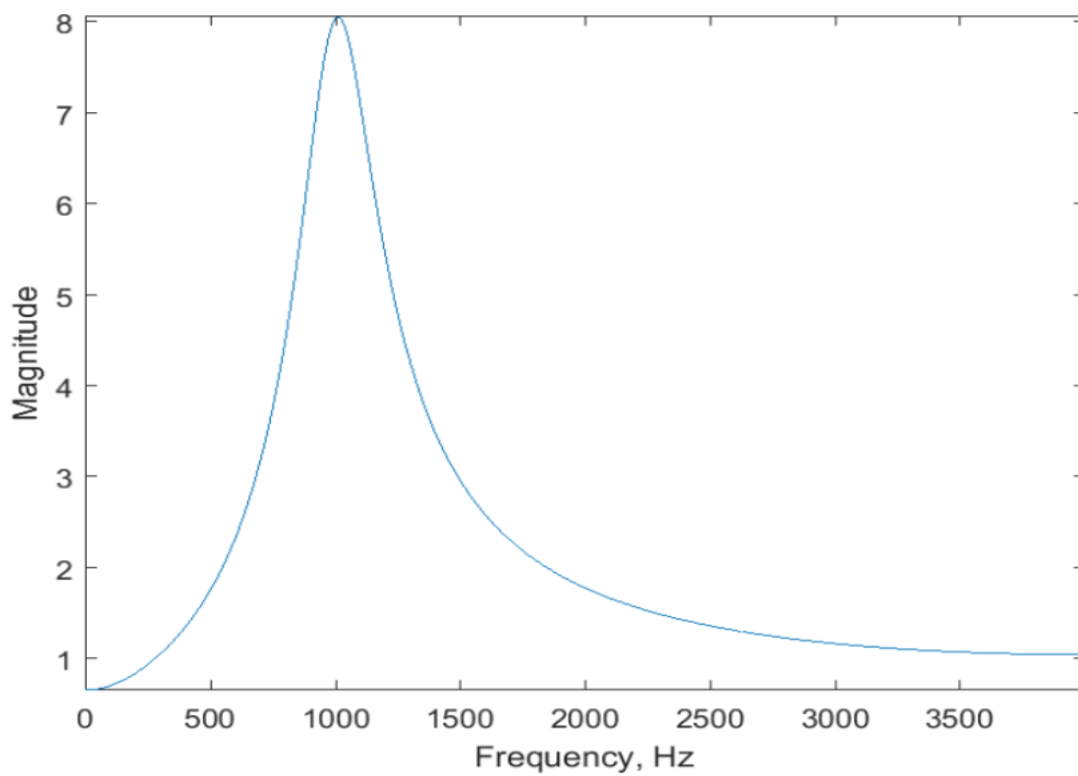


Рисунок 1.2

Многоканальный сигнал и его спектр:

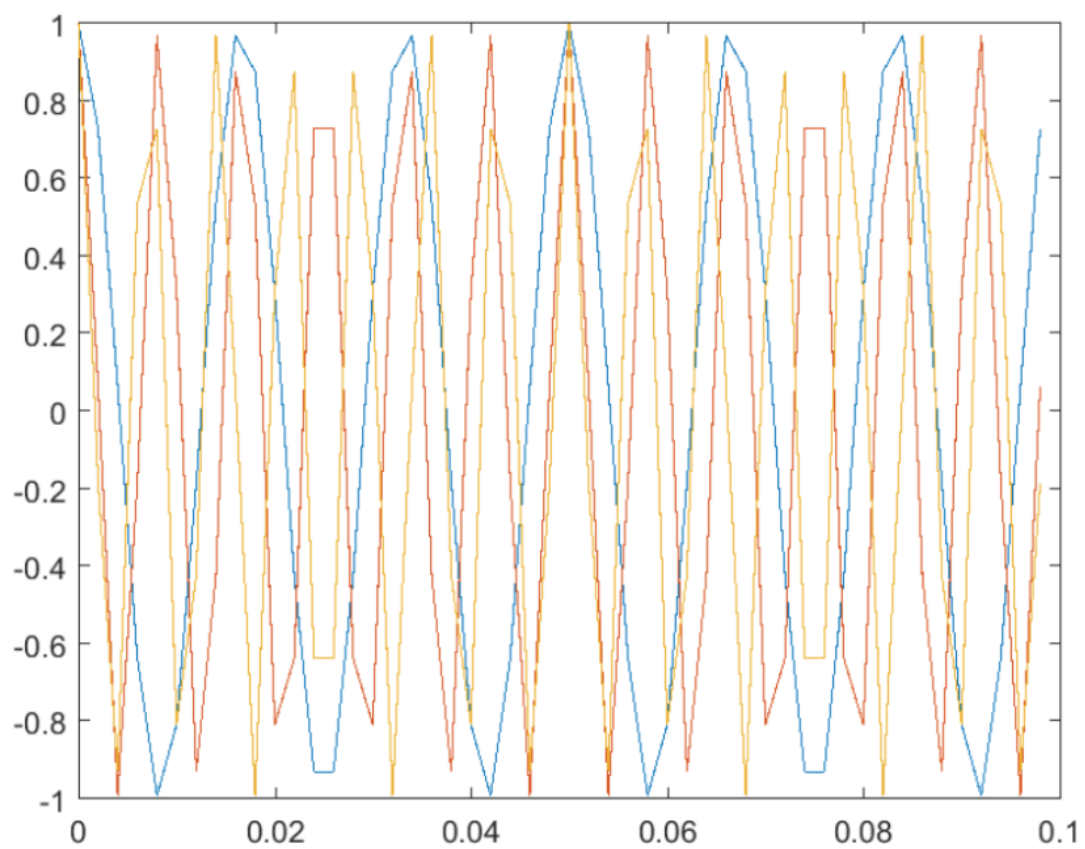


Рисунок 2.1

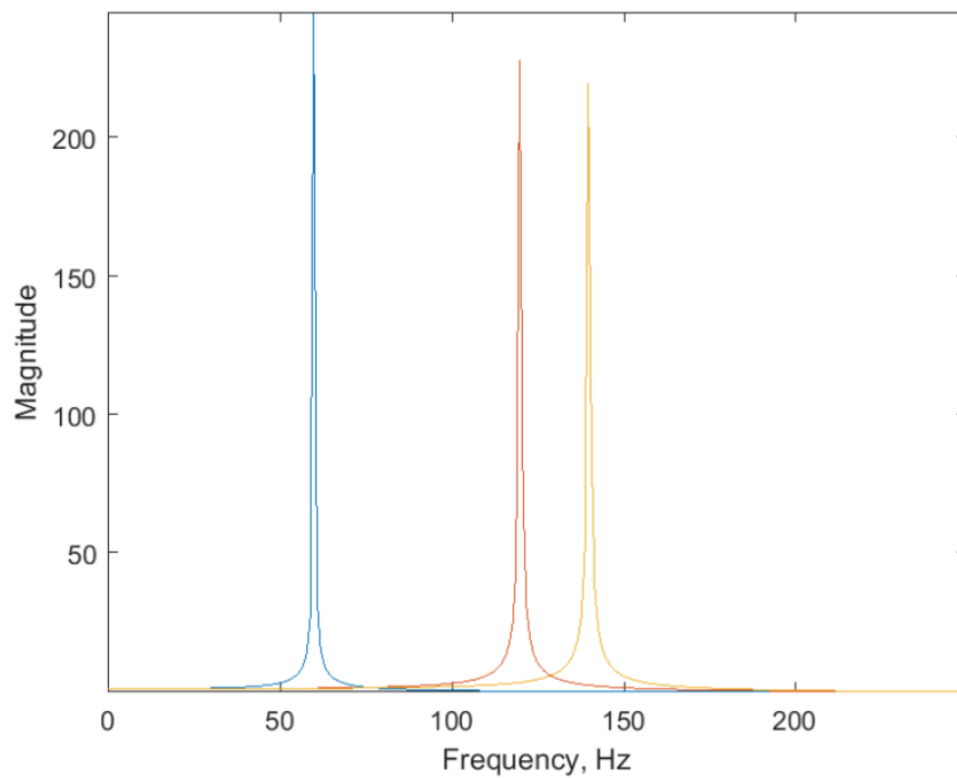


Рисунок 2.2

На рисунке 2.1 представлено несколько многоканальных сигналов разных частот, на рисунке 2.2 – их спектры

Односторонний экспоненциальный импульс и его спектр:

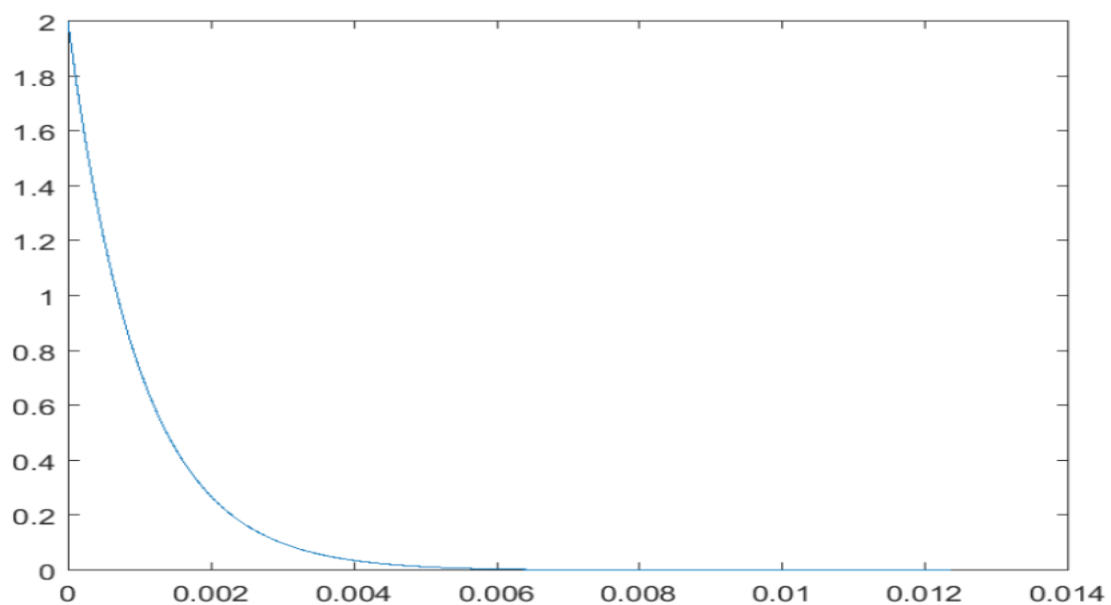


Рисунок 3.1

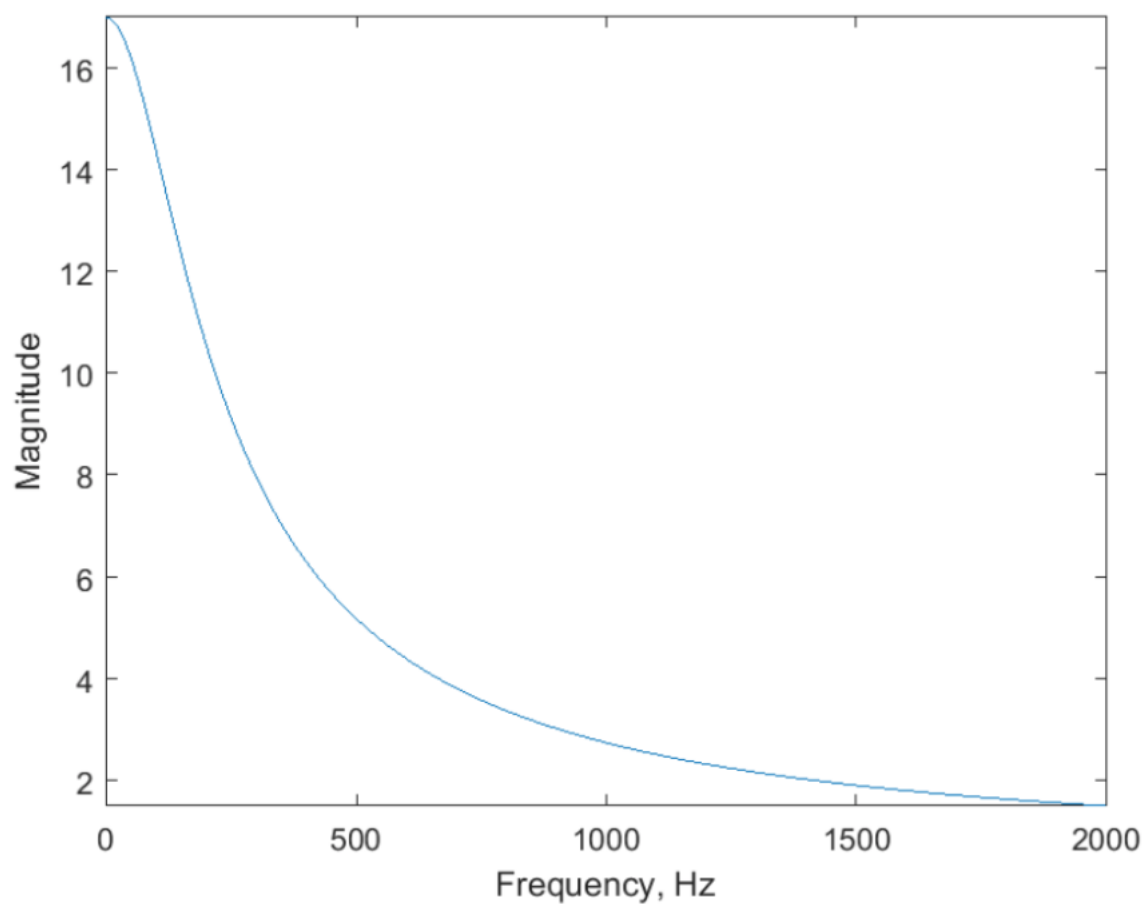


Рисунок 3.2

Прямоугольный импульс, центрированный относительно начала координат и его спектр:

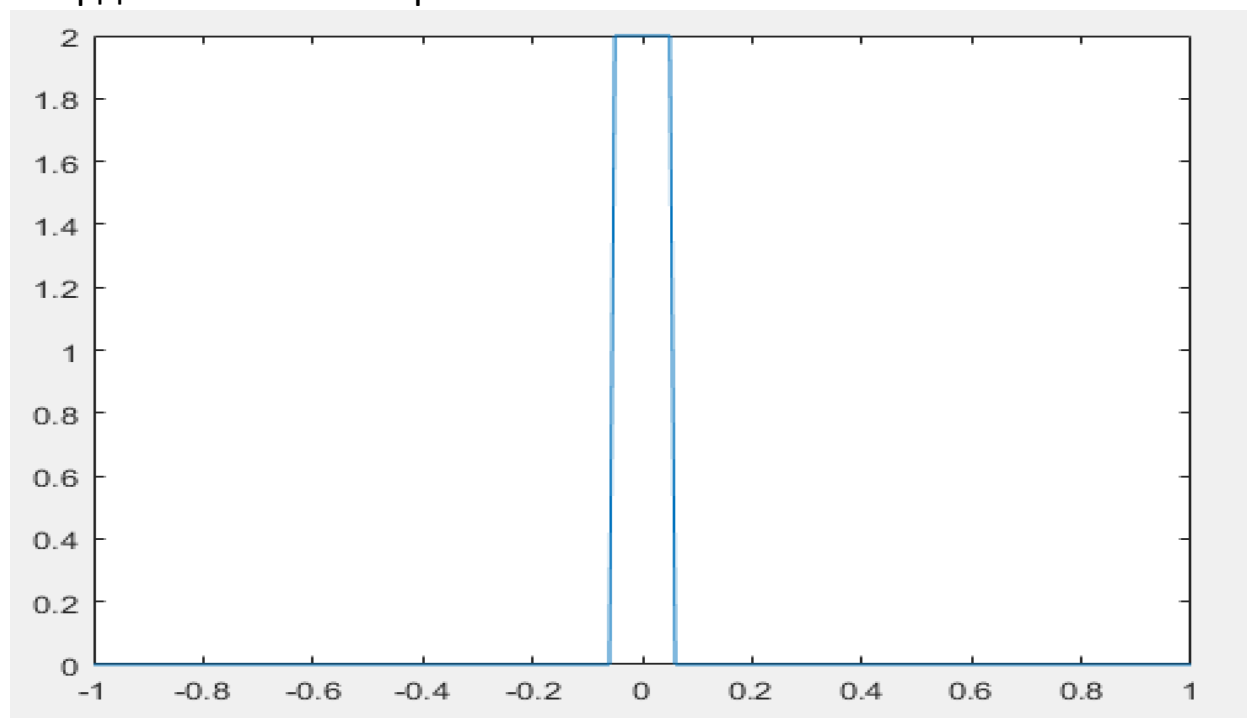


Рисунок 4.1

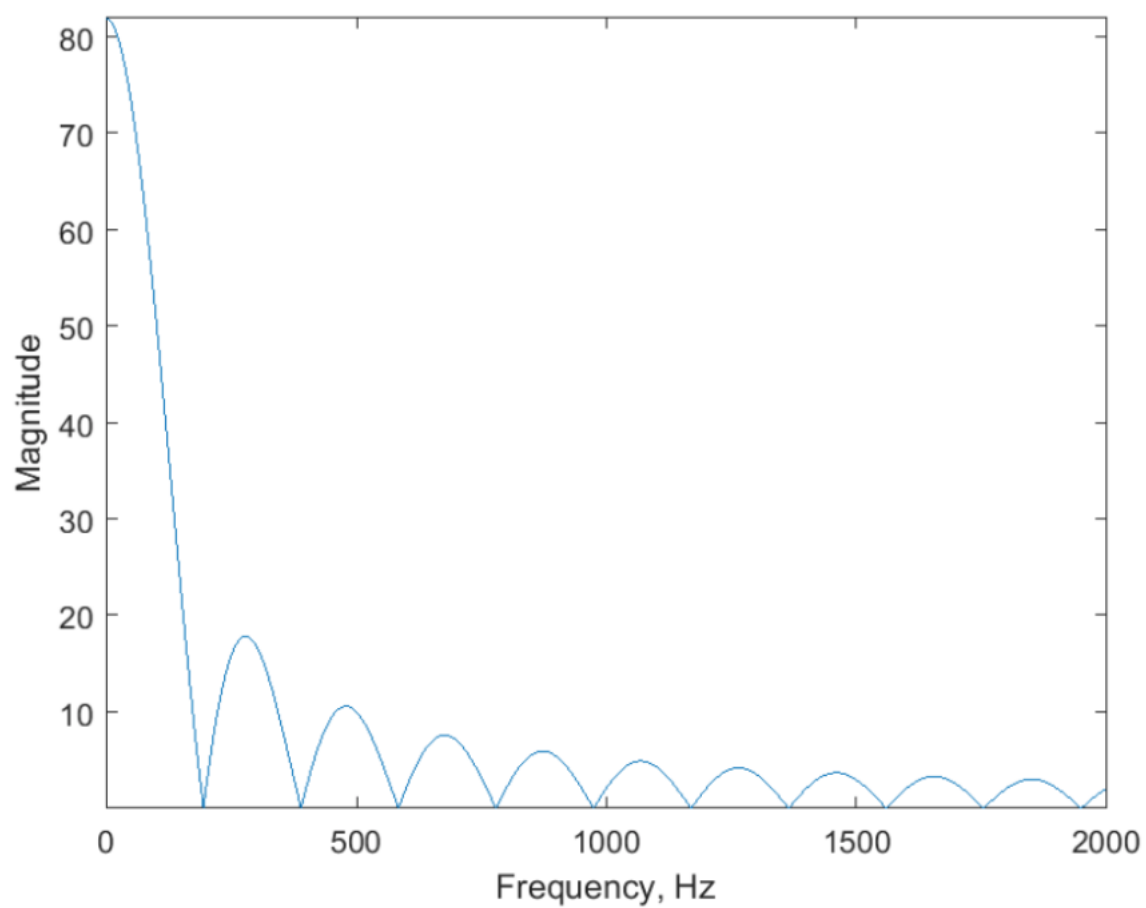


Рисунок 4.2

Треугольный импульс и его спектр:

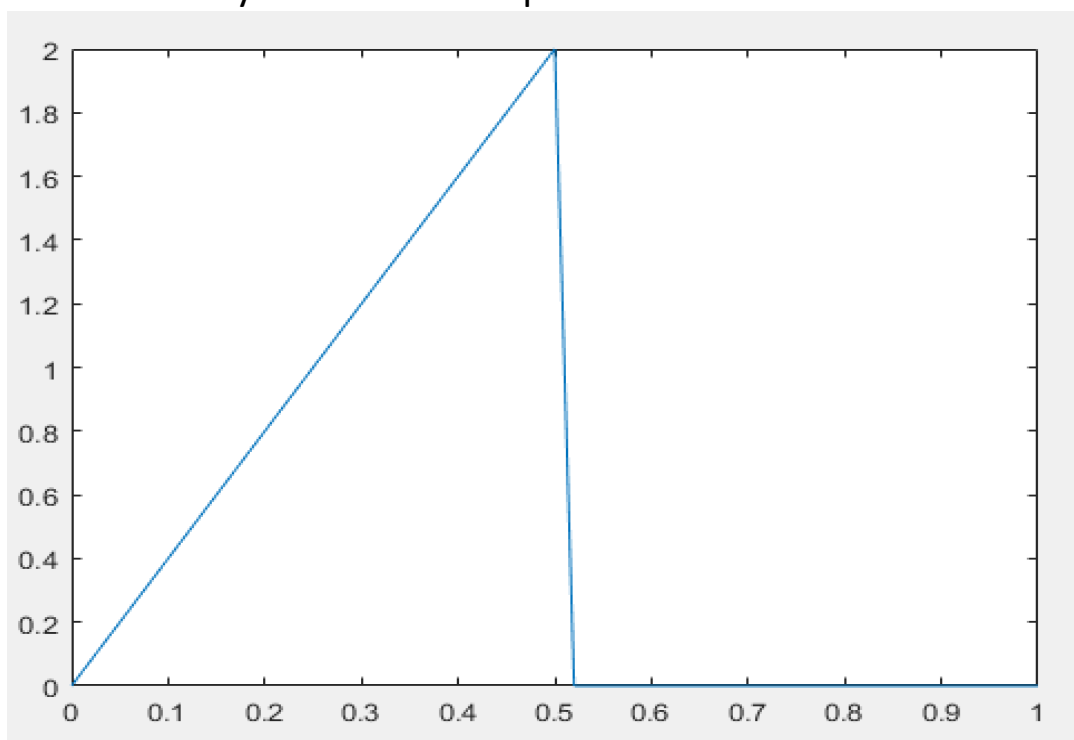


Рисунок 5.1

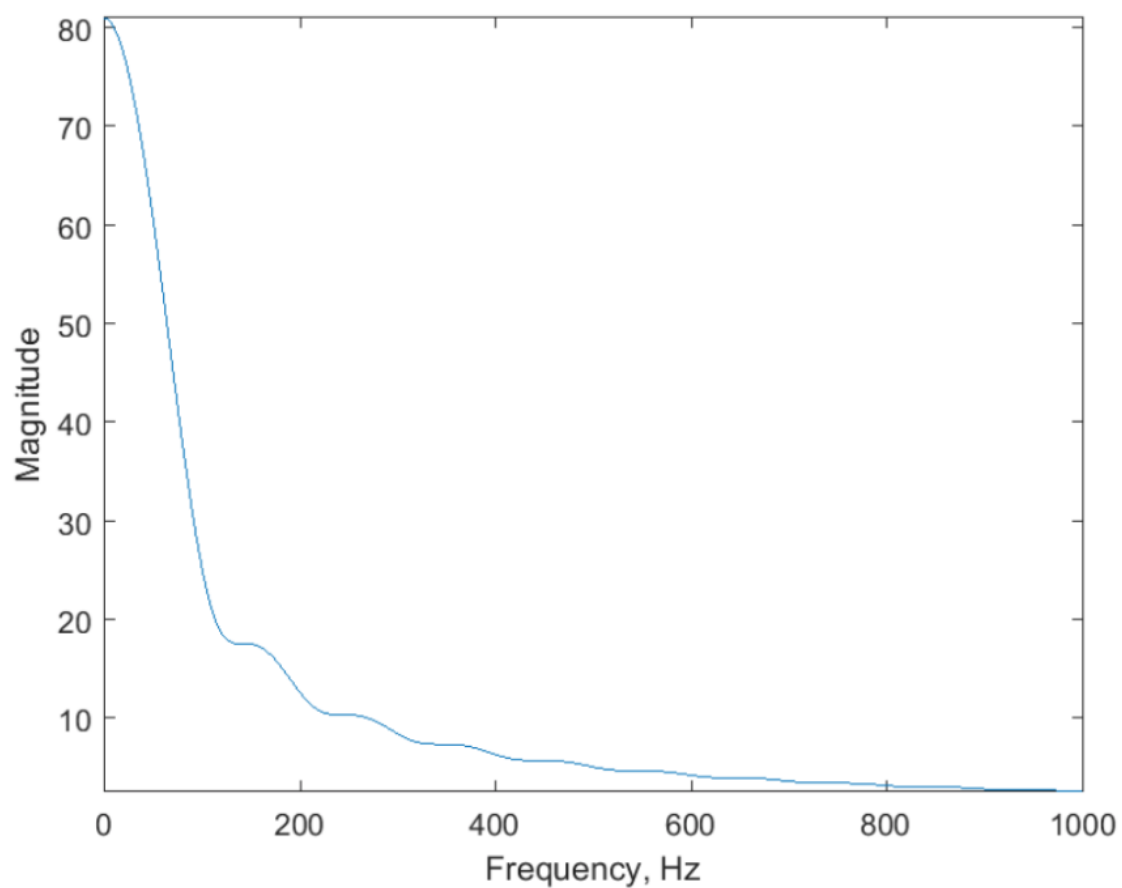


Рисунок 5.2

Прямоугольный импульс и его спектр:

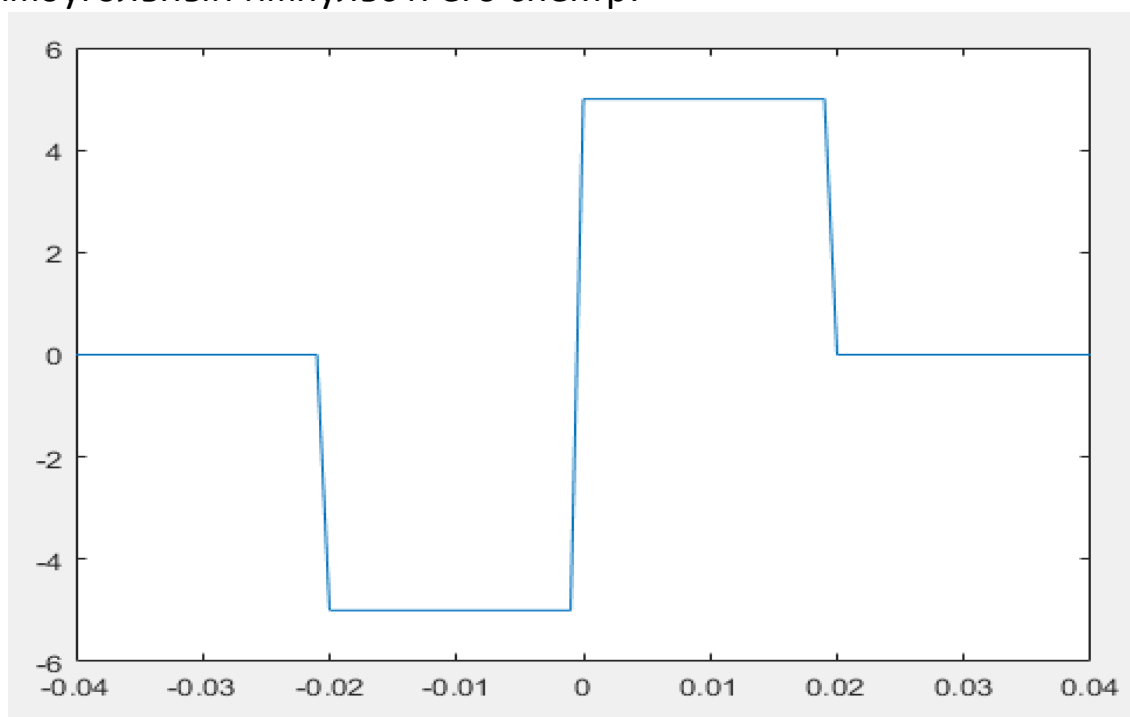


Рисунок 6.1

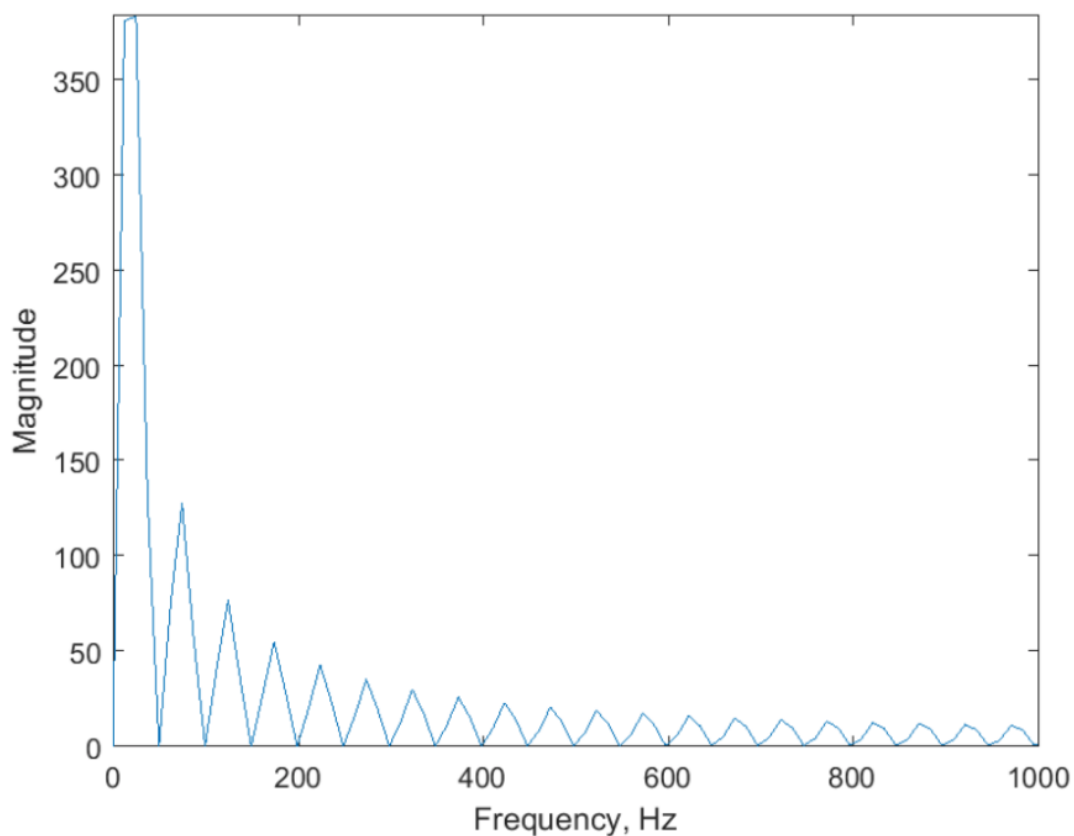


Рисунок 6.2

Сигнал на рисунке 6.1 получен с помощью соединения двух прямоугольных импульсов с использованием функции `rectpuls`.

Трапецевидный импульс и его спектр:

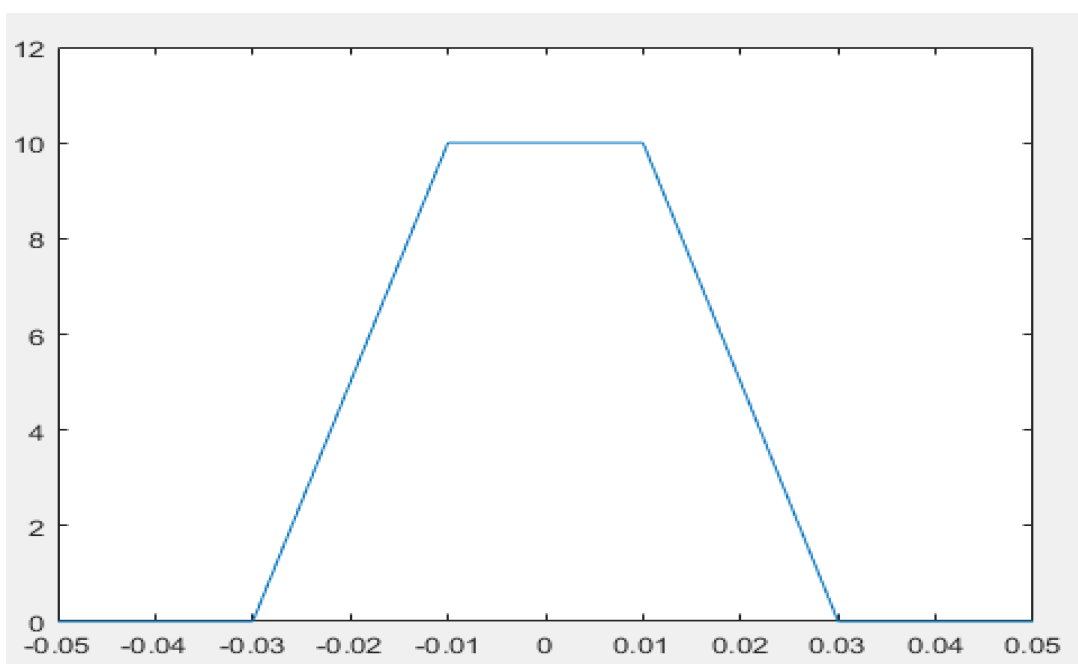


Рисунок 7.1

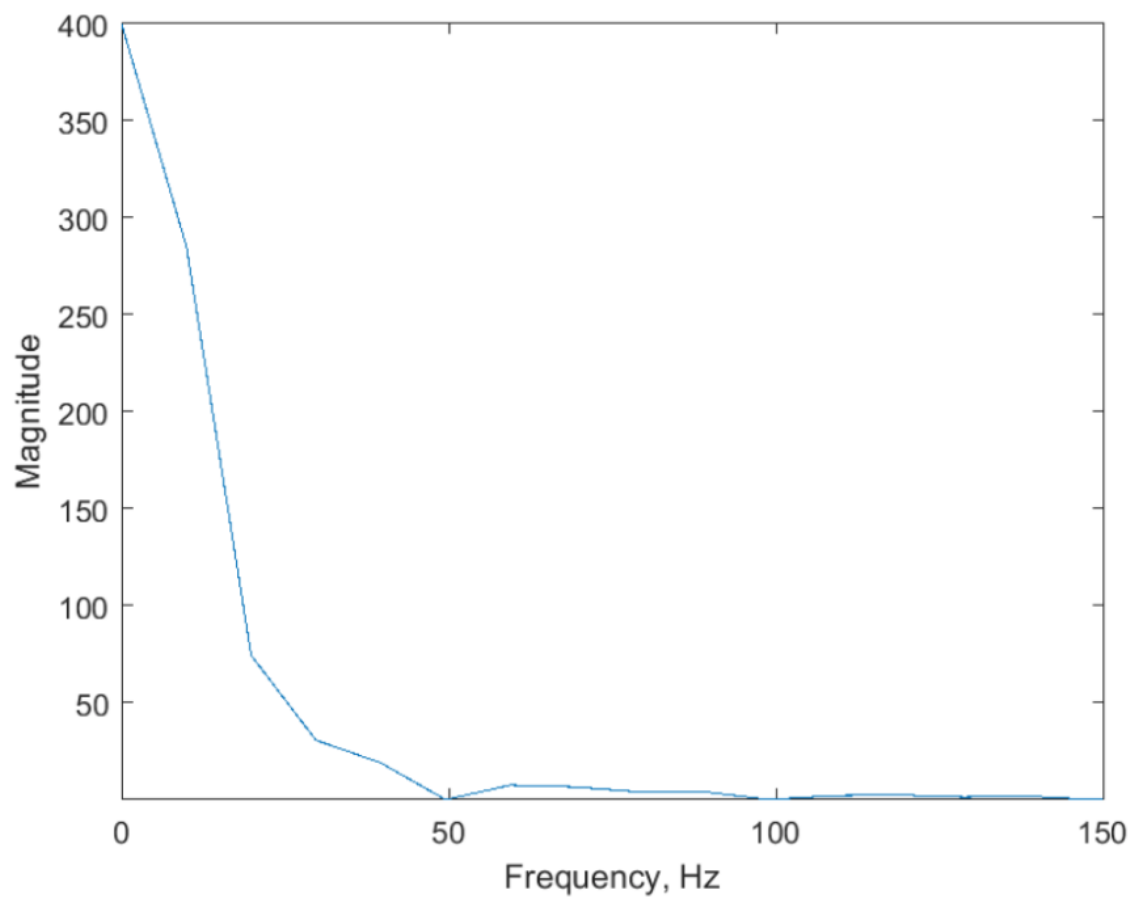


Рисунок 7.2

Импульс с ограниченной полосой частот и его спектр:

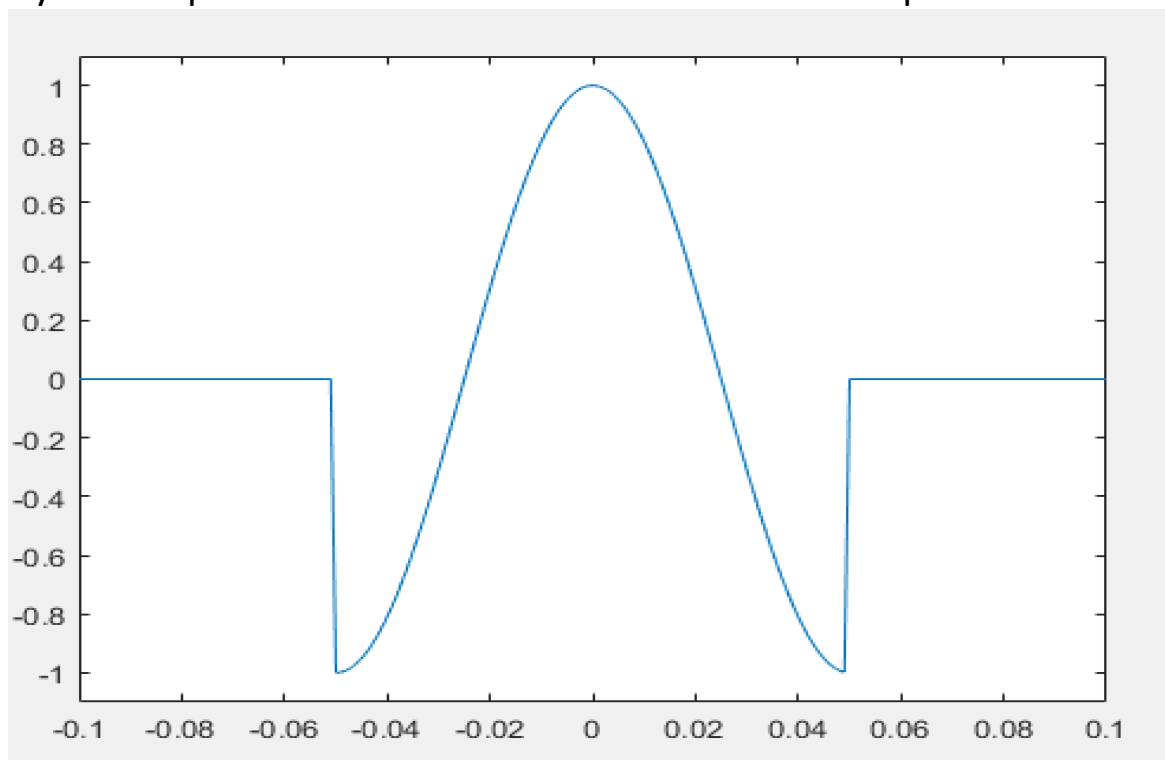


Рисунок 8.1

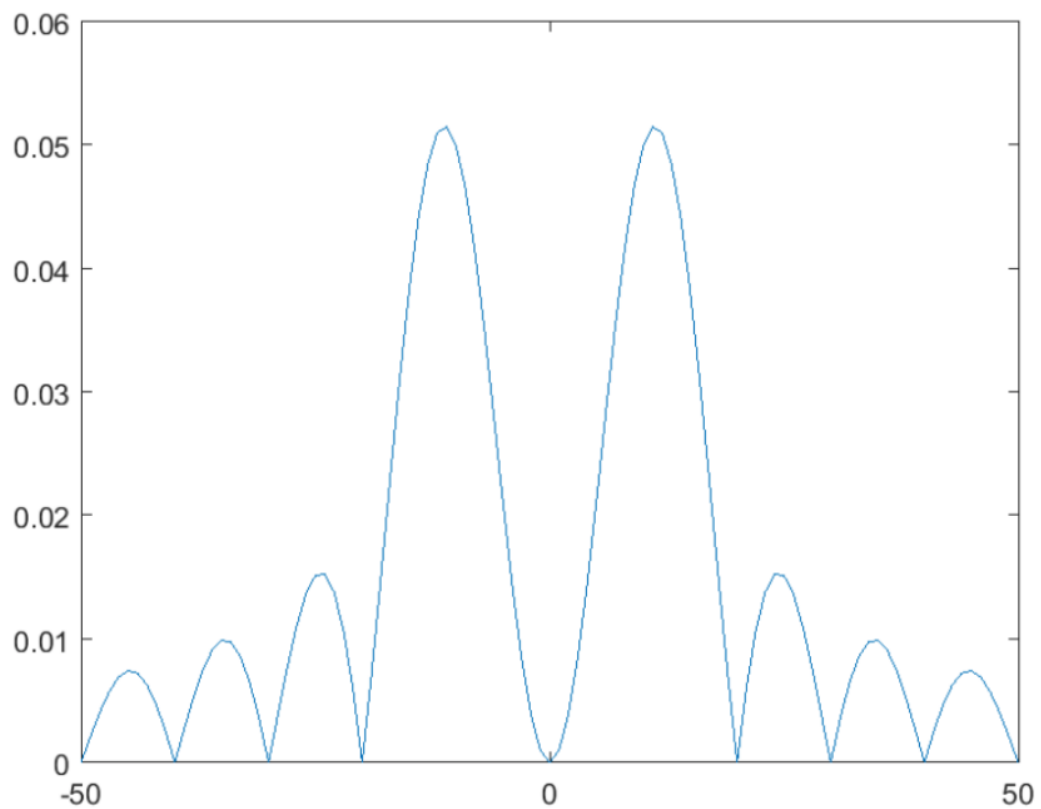


Рисунок 8.2

Гауссов радиоимпульс и его спектр:

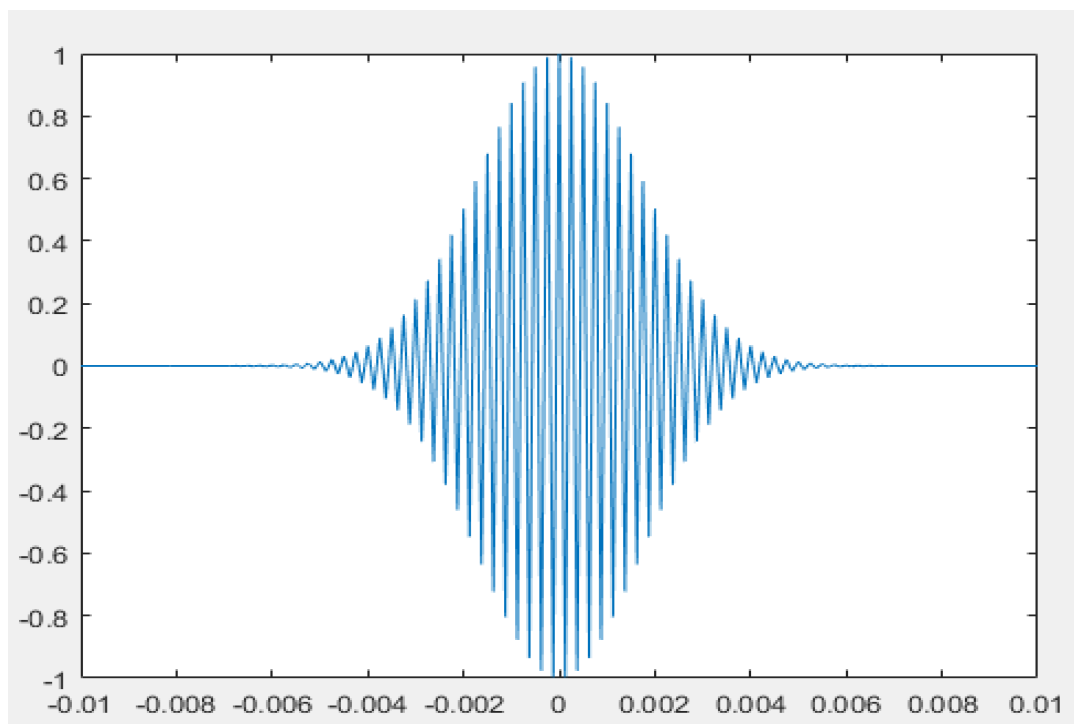


Рисунок 9.1

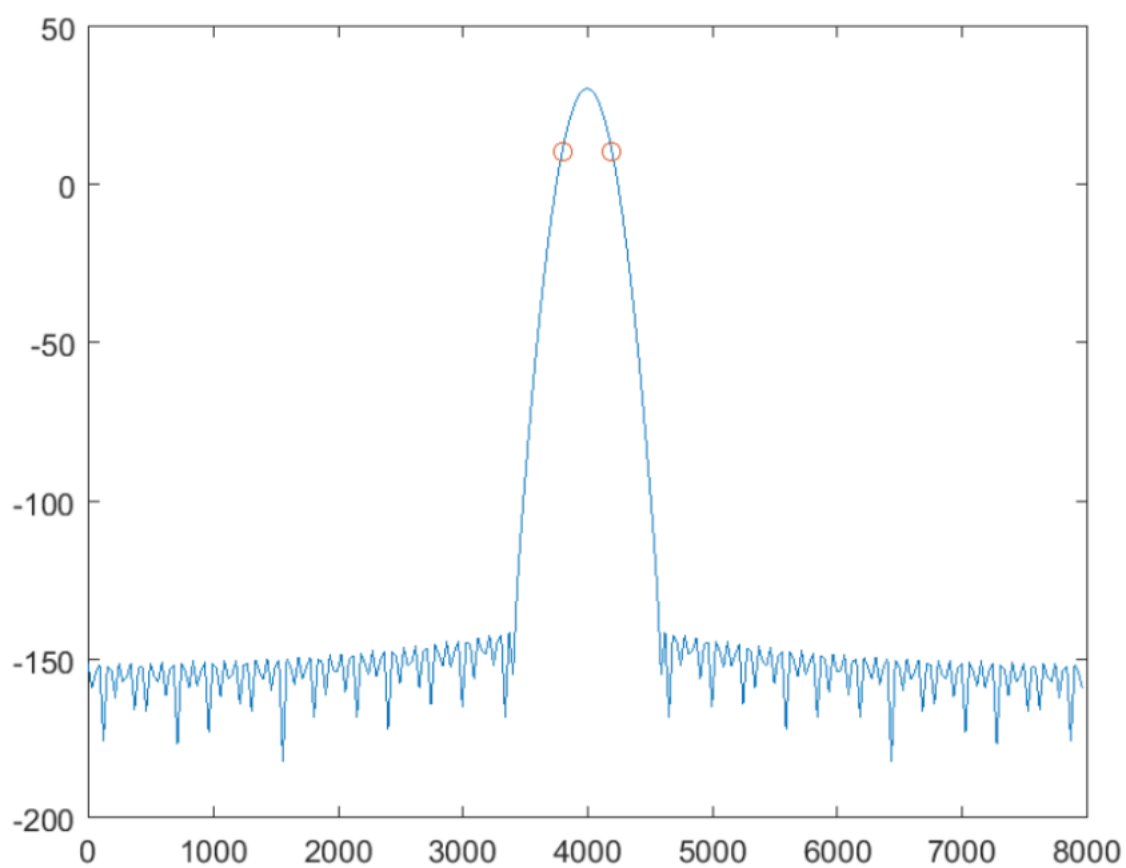


Рисунок 9.2

Генерация последовательностей импульсов:
Треугольные импульсы и их спектр:

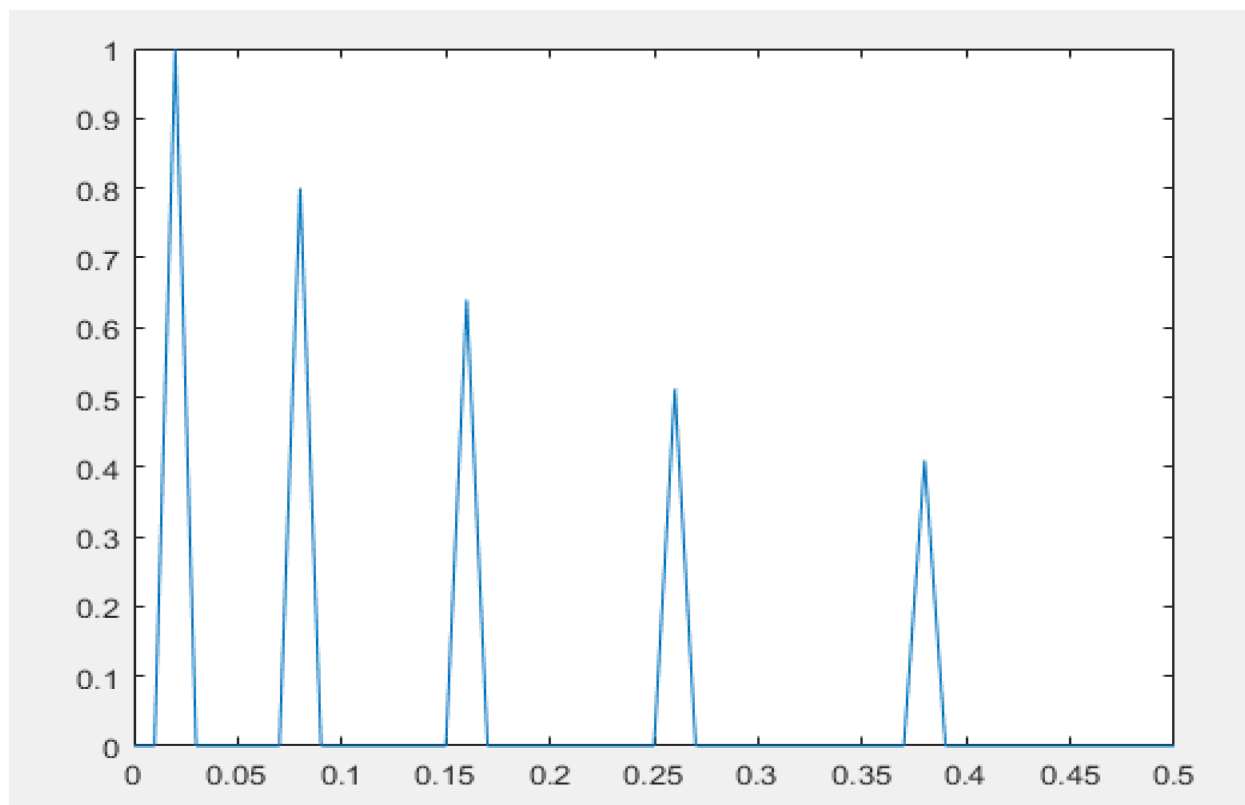


Рисунок 10.1

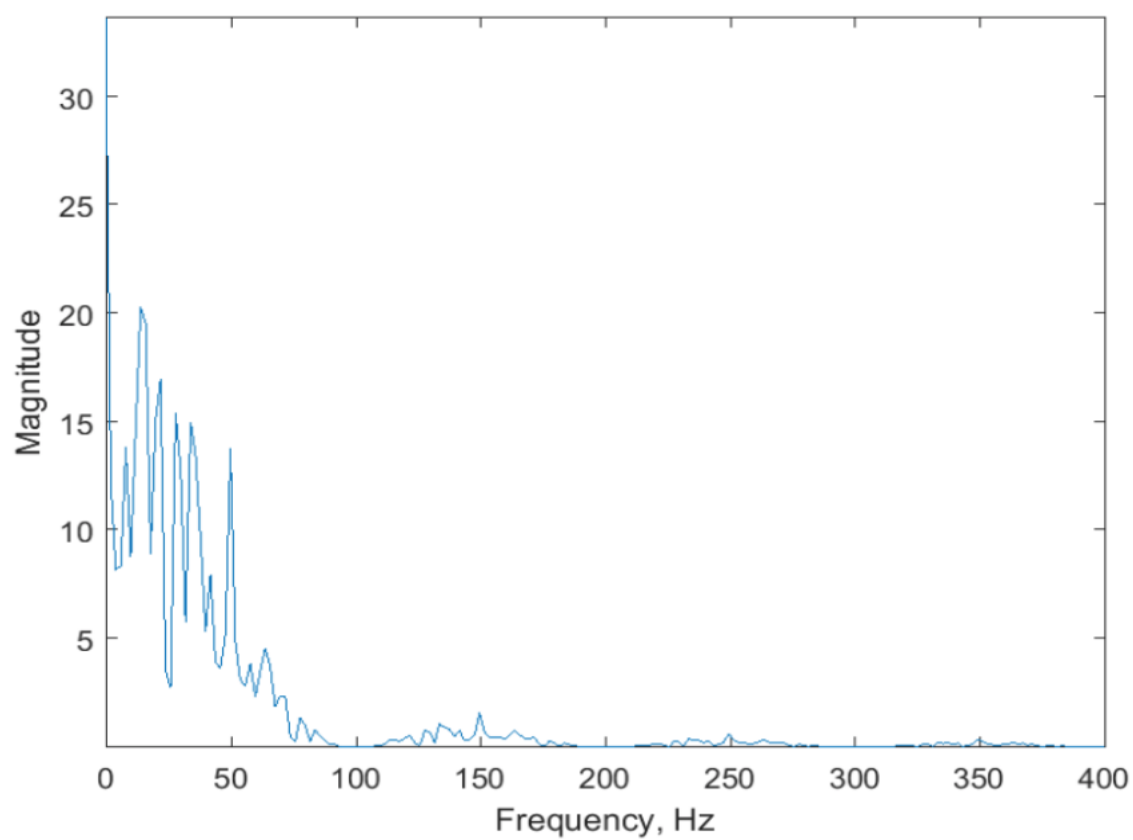


Рисунок 10.2

Гармонические импульсы и их спектр:

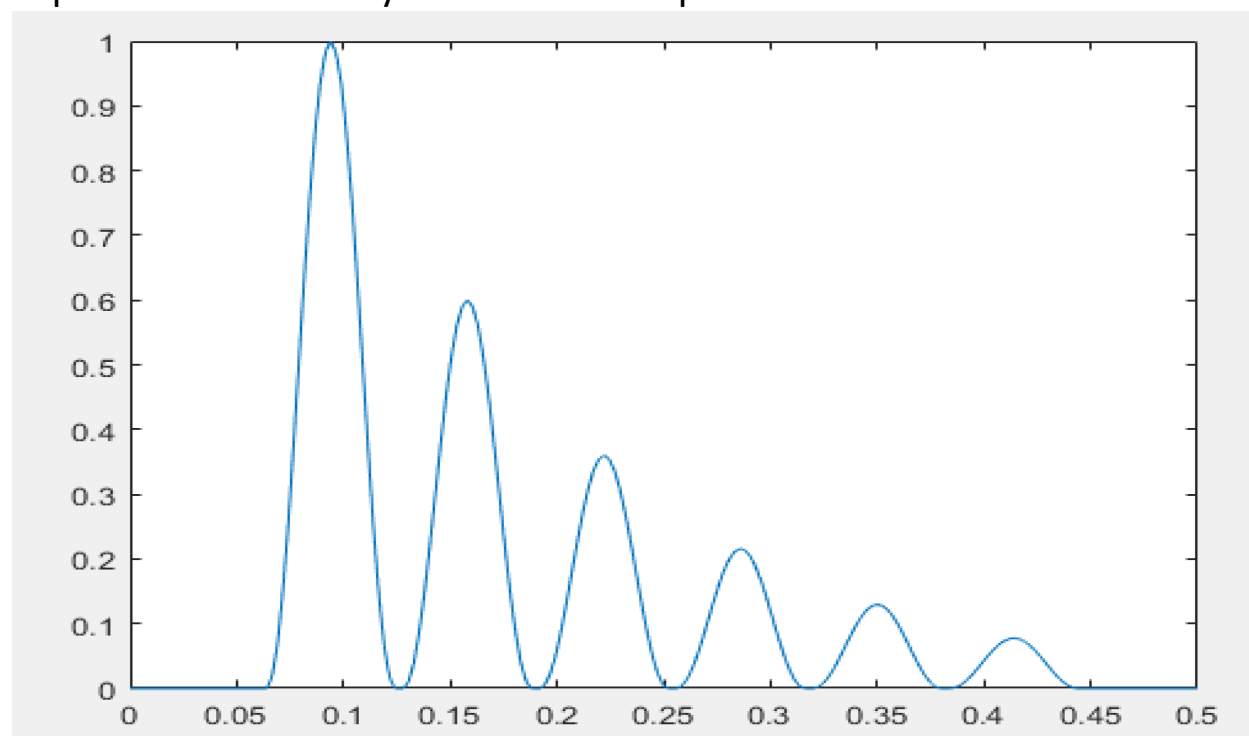


Рисунок 11.1

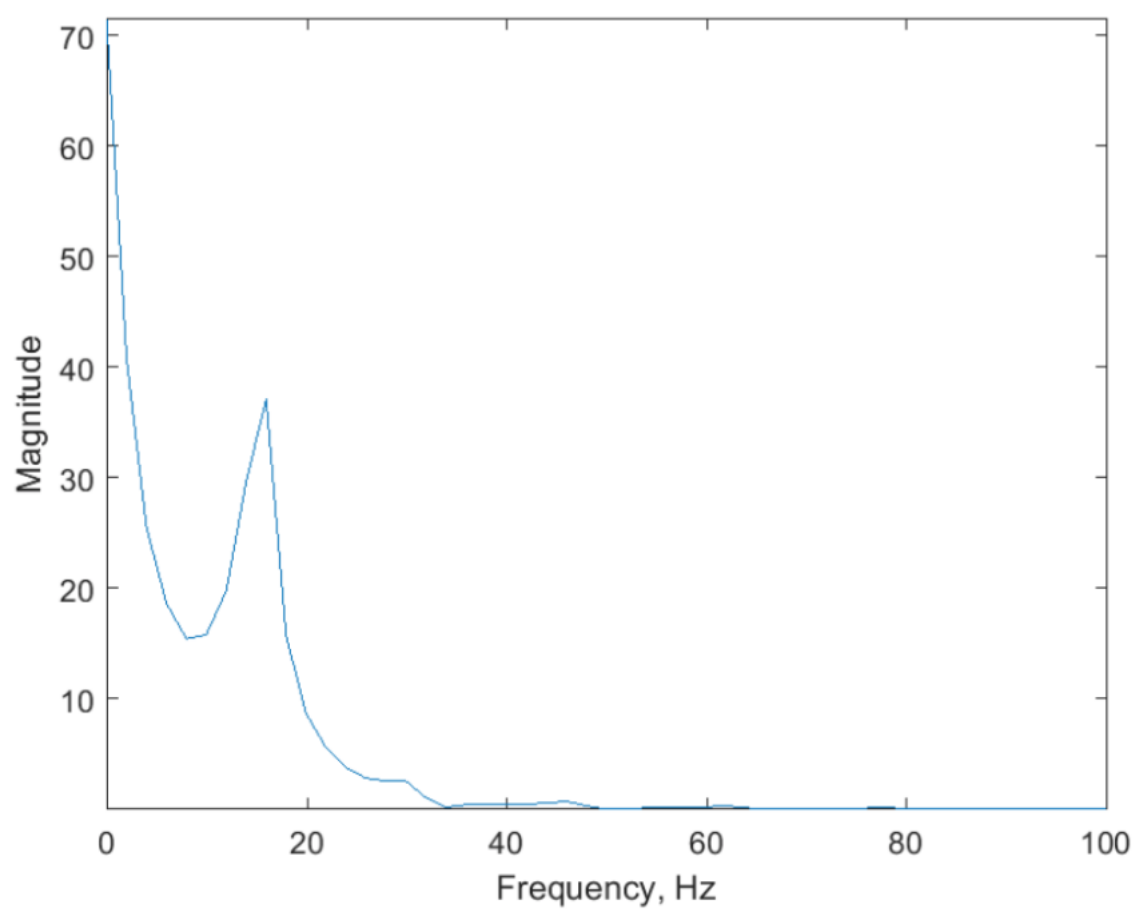


Рисунок 11.2

Генерация периодических сигналов:
Прямоугольные импульсы и их спектр:

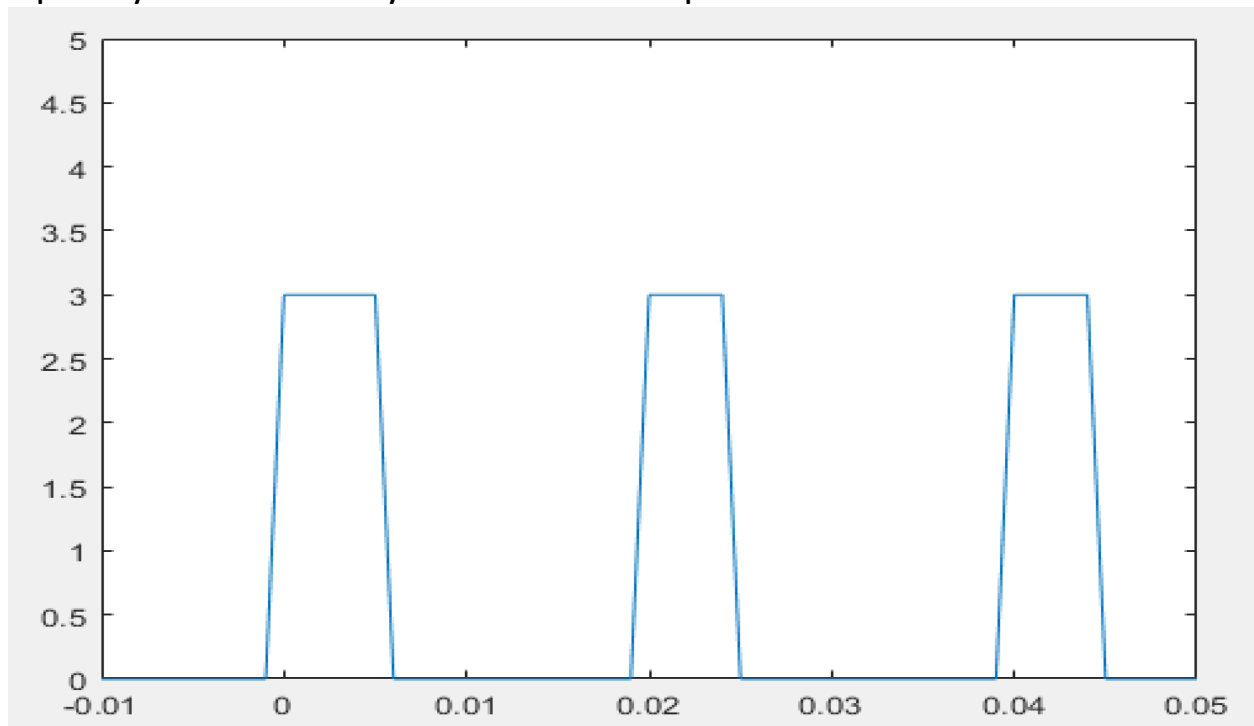


Рисунок 12.1

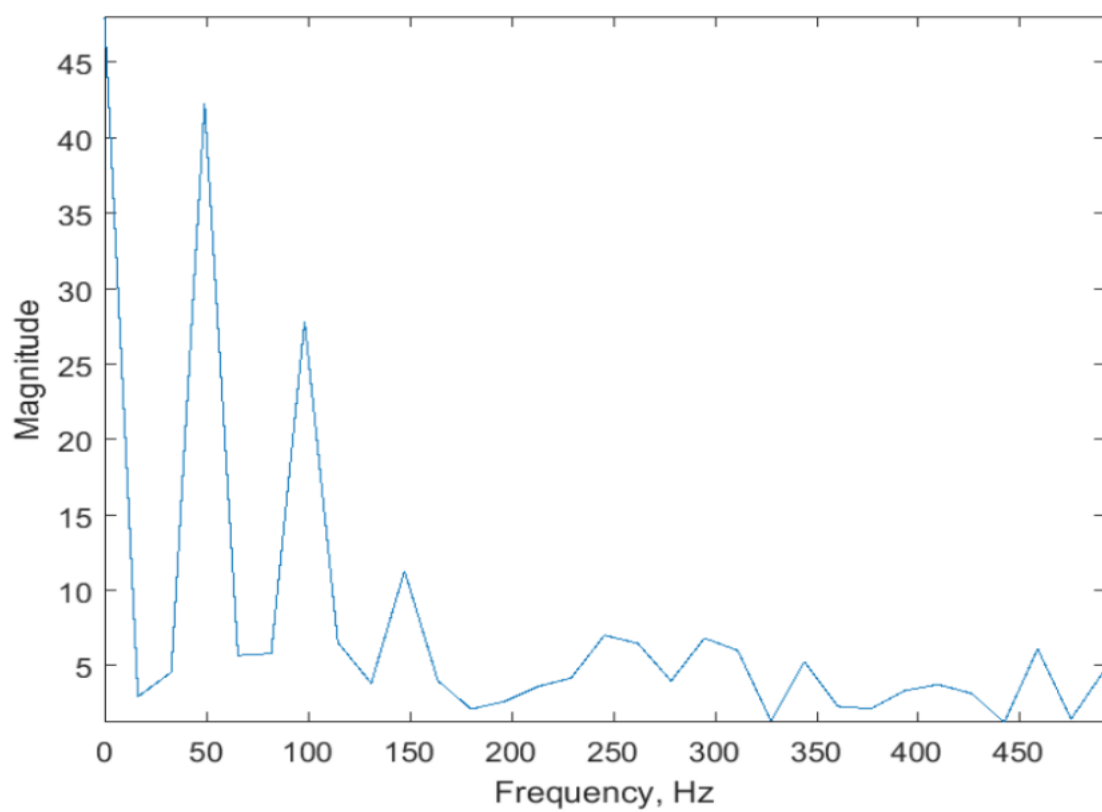


Рисунок 12.2

Треугольные импульсы и их спектр:

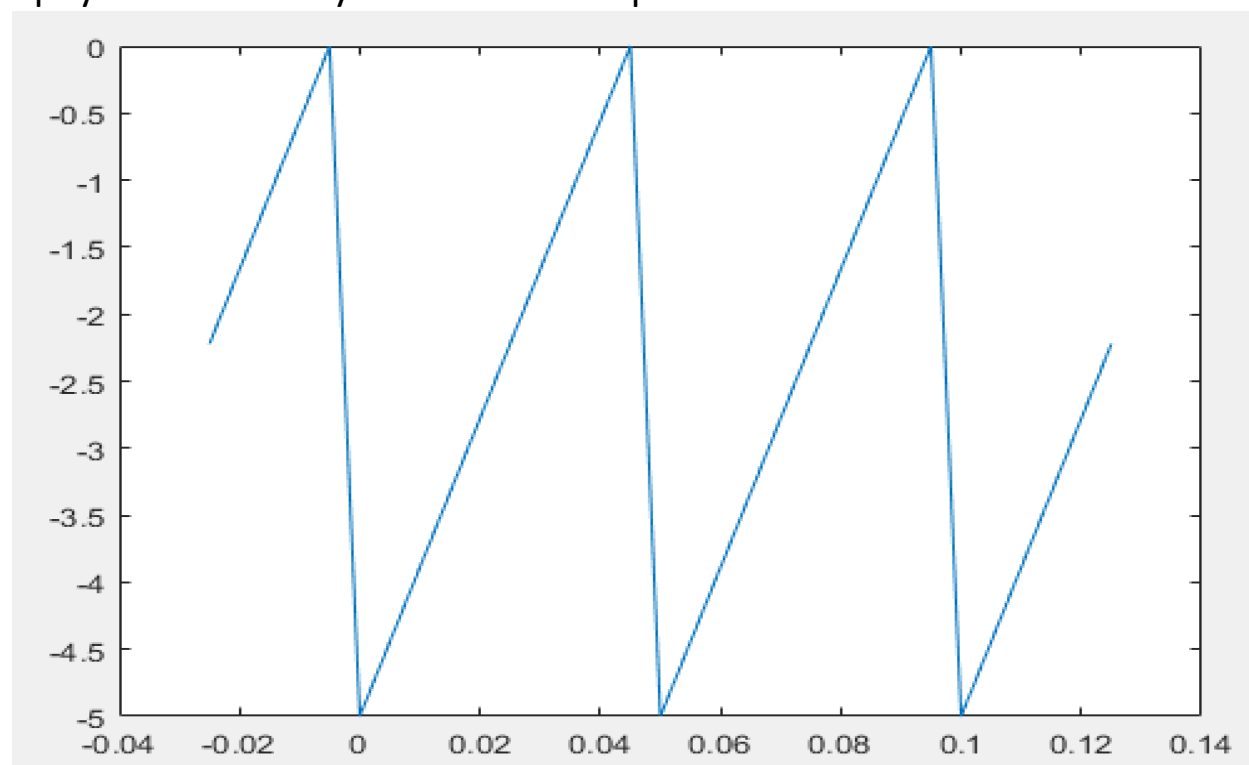


Рисунок 13.1

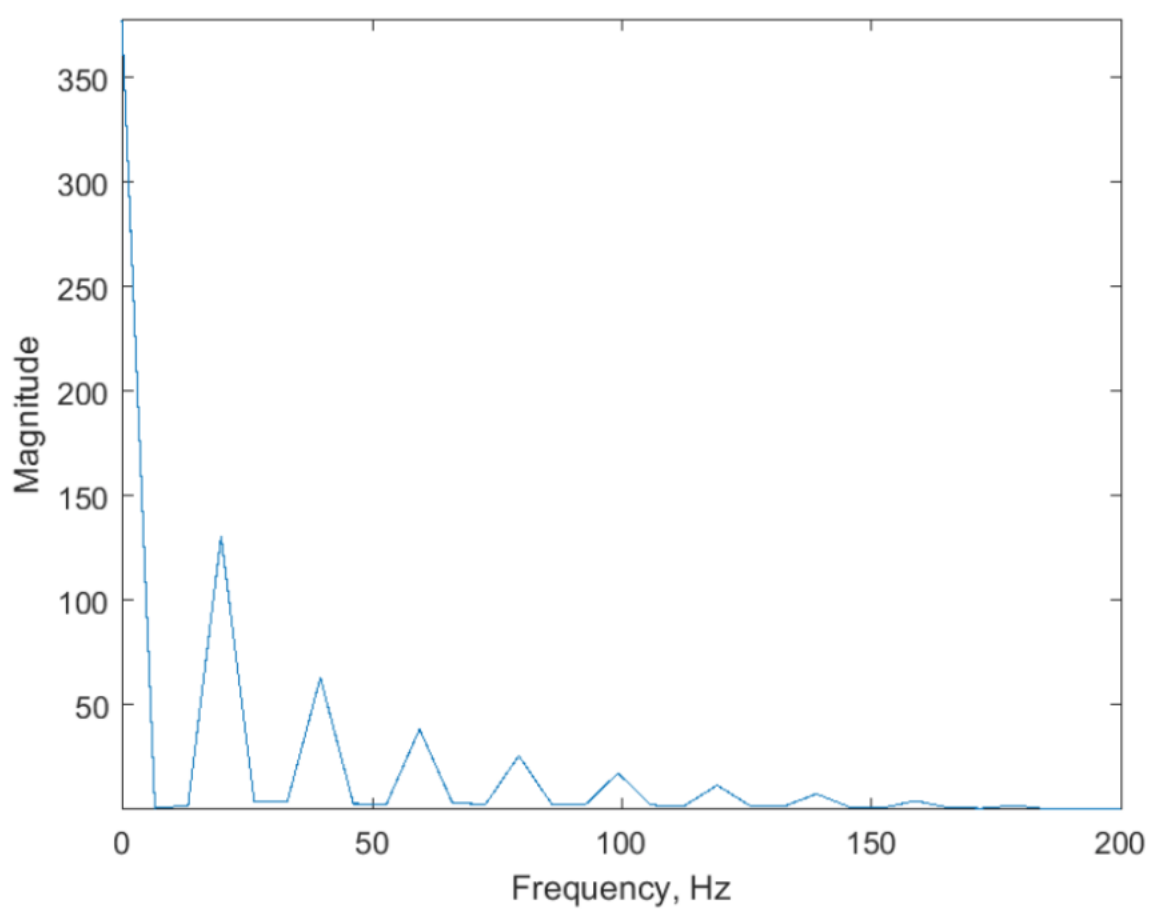


Рисунок 13.2

Функция Дирихле при $n = 7$ и ее спектр:

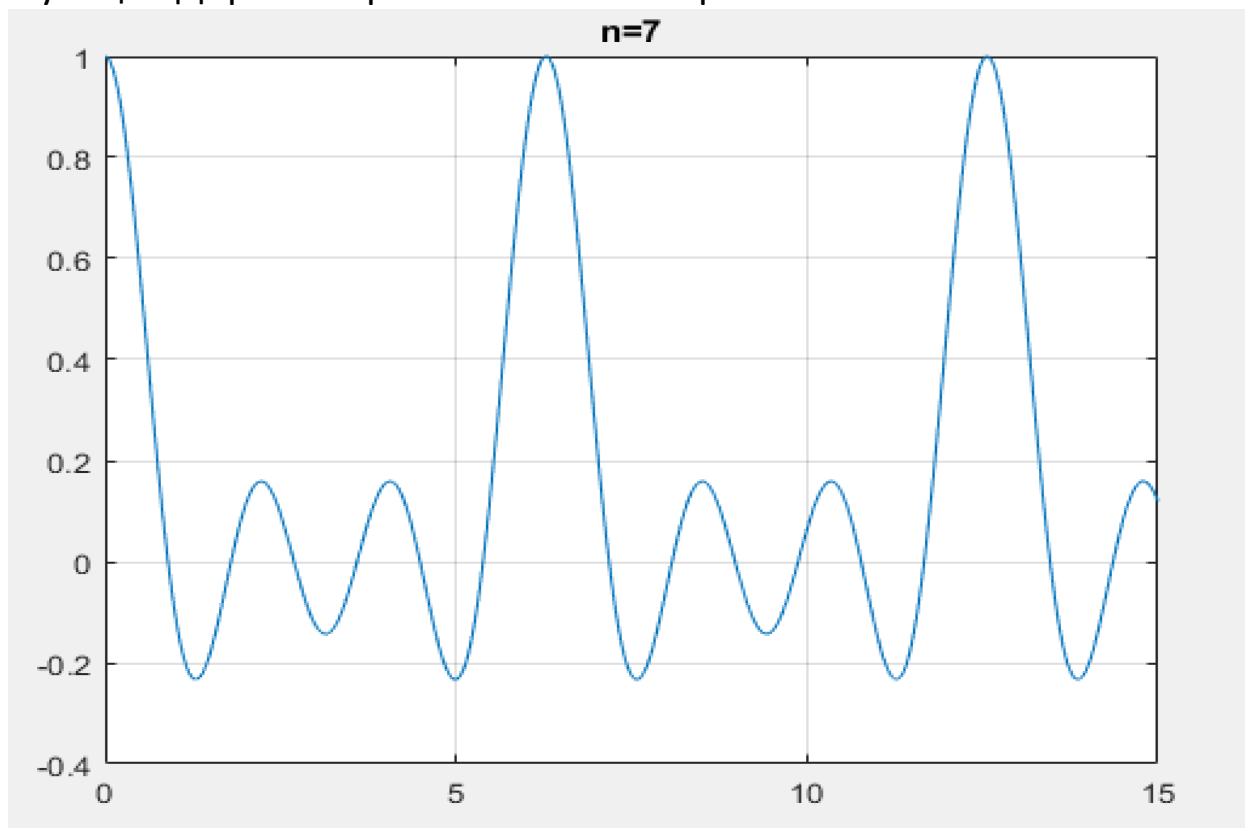


Рисунок 14.1

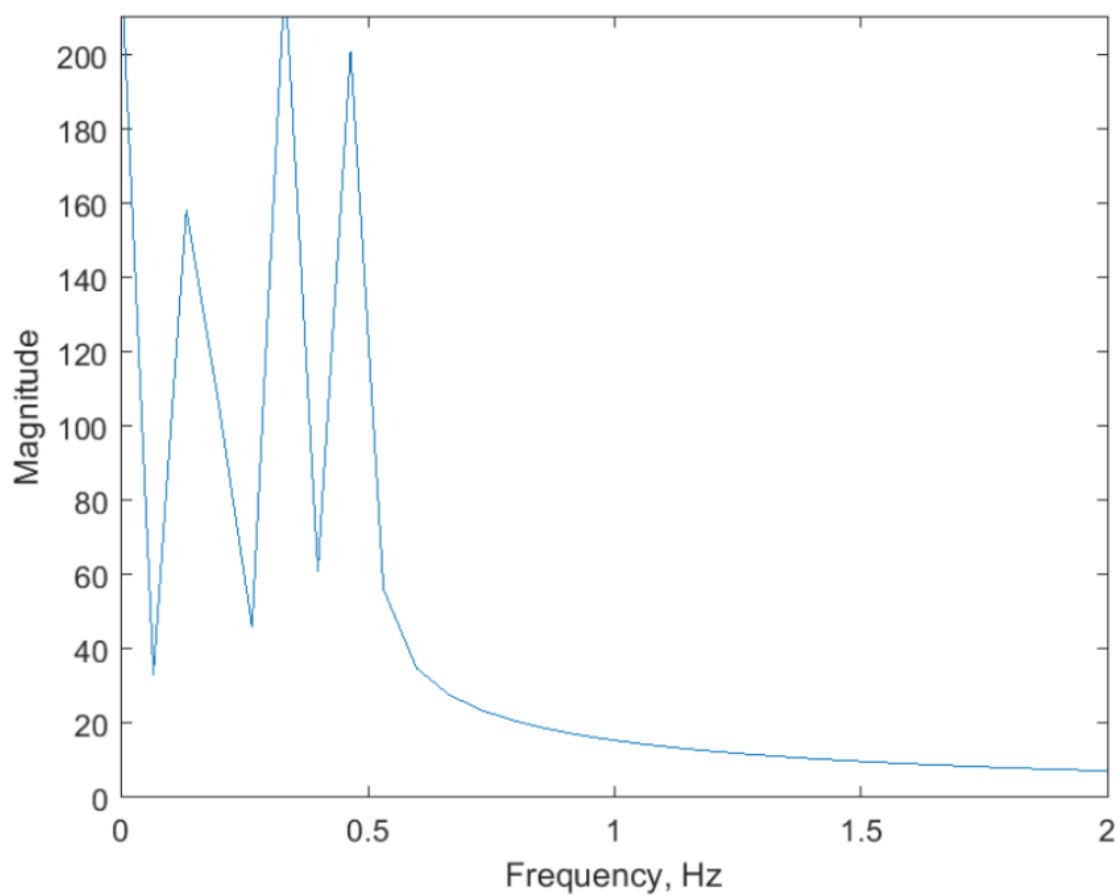


Рисунок 14.2

Функция Дирихле при $n = 8$ и ее спектр:

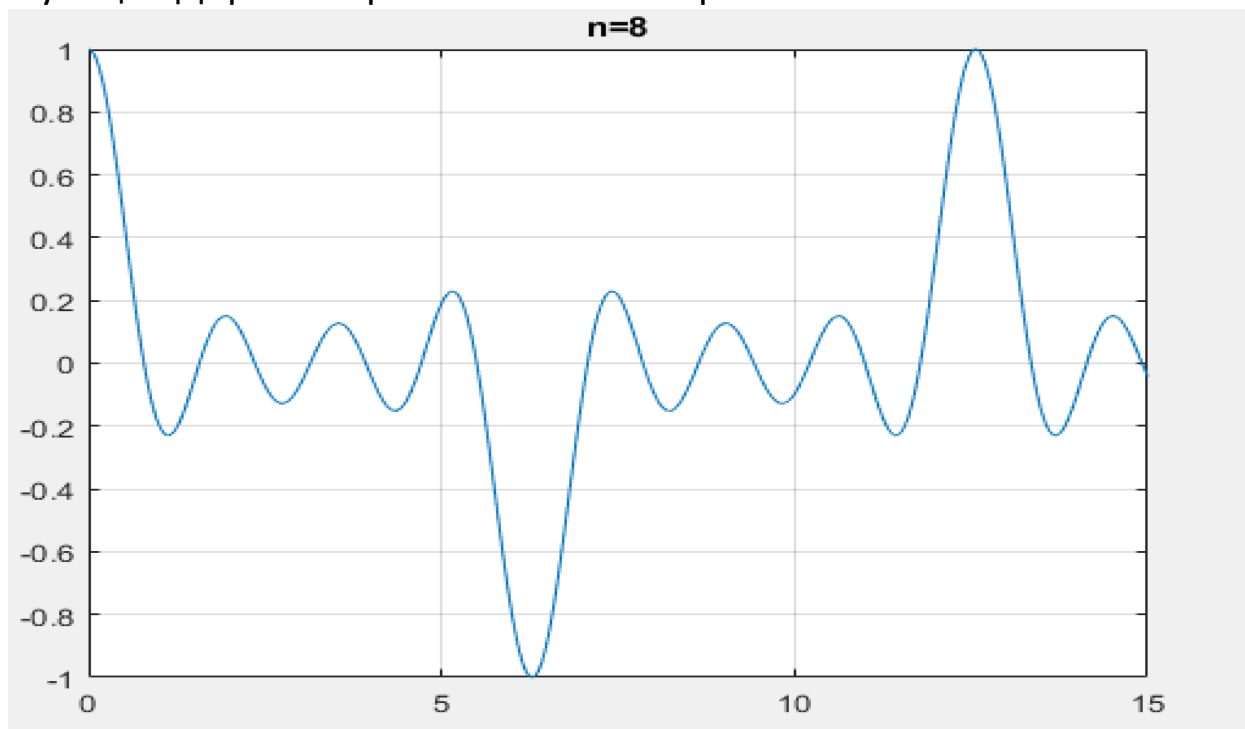


Рисунок 15.1

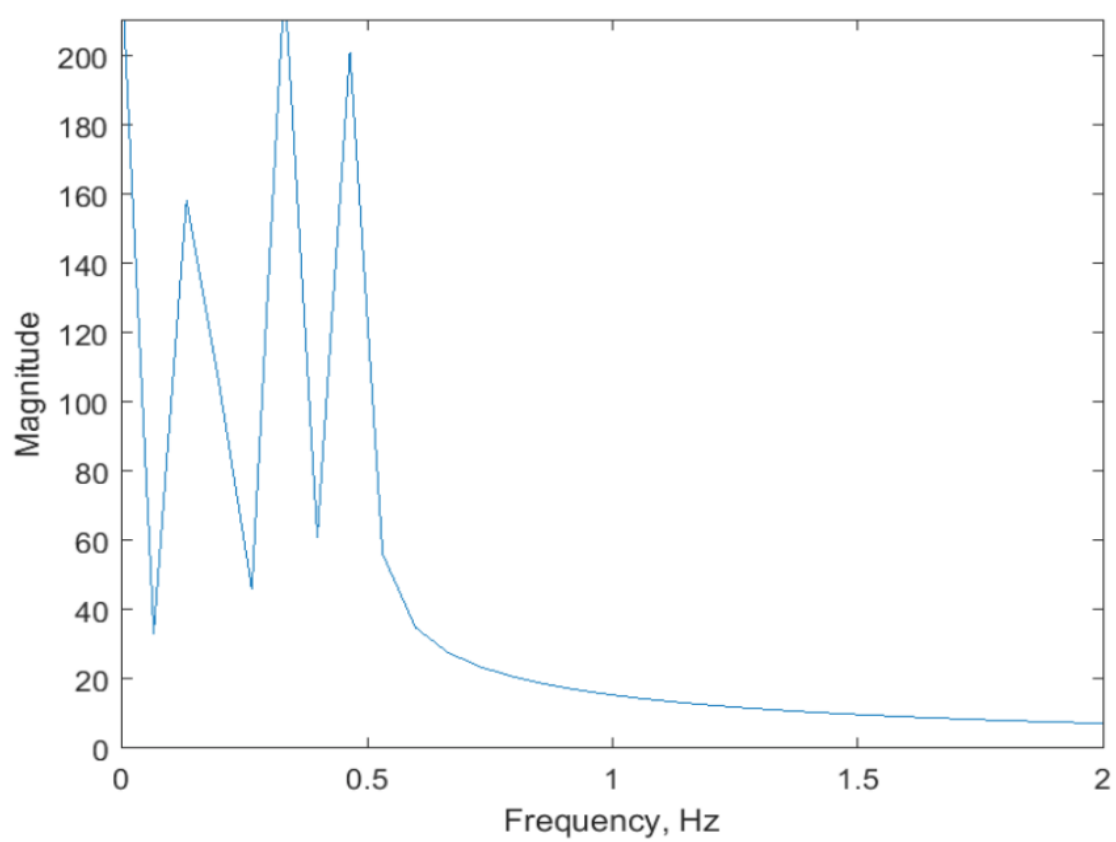


Рисунок 15.2

Как видно из рисунков 14.1 и 15.1, нечетный параметр в функции Дирихле обеспечивает однонаправленность импульсов.

Генерация сигналов с меняющейся частотой:

С помощью функции `chirp` можно генерировать колебания, мгновенная частота которых изменяется согласно выбранной функции. Далее рассмотрены 3 таких функции — линейная, квадратичная и логарифмическая. На рисунках ниже представлены спектрограммы этих сигналов — зависимость мгновенного амплитудного спектра от времени.

Линейная функция:

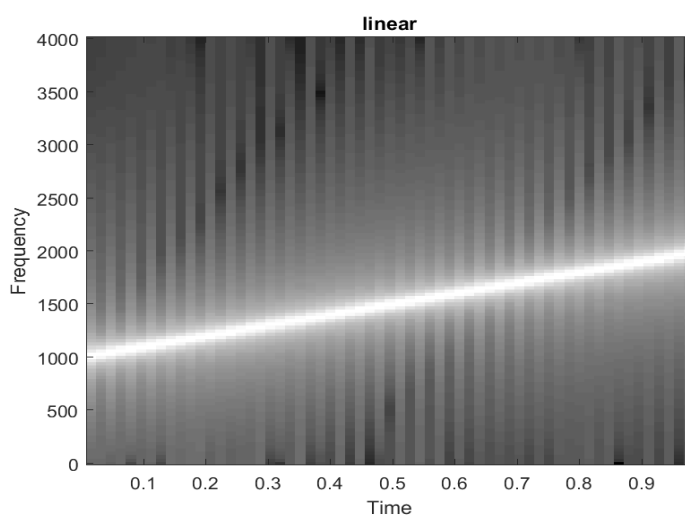


Рисунок 16.1

Квадратичная функция:

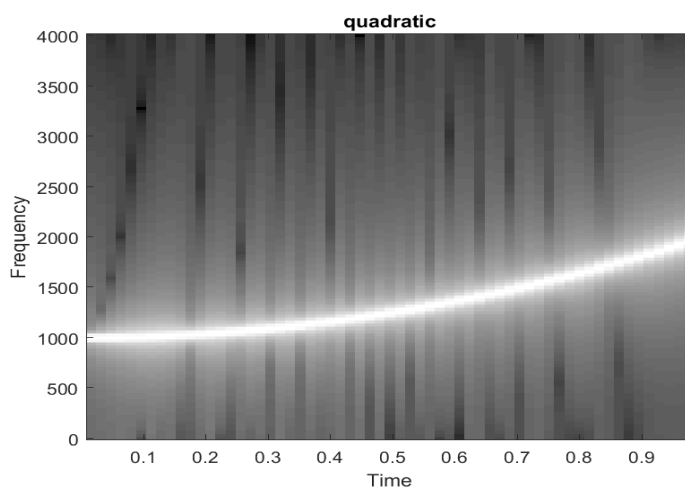


Рисунок 16.2

Логарифмическая функция:

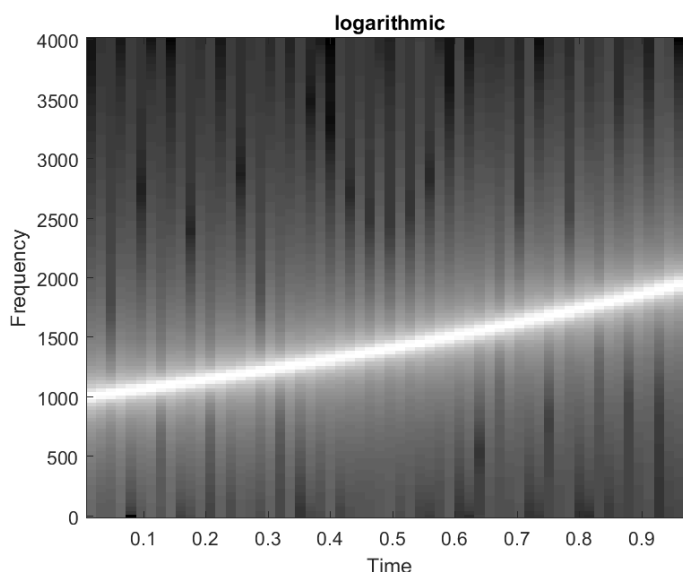


Рисунок 16.3

4.2 Сравнение методов корреляции

В качестве примера было взято нахождение синхропосылки [101] в сигнале [0001010111000010].

Оба алгоритма показали, что искомая посылка была найдена в сигнале 2 раза.

Время работы алгоритма прямой корреляции: 0,118 мс

Время работы алгоритма быстрой корреляции: 0,025 мс

5. Выводы

В ходе работы были получены навыки генерации и визуализации сигналов различных типов в среде Matlab. Были рассмотрены периодические и непериодические сигналы, одиночные импульсы и последовательности импульсов, сигналы различных форм: треугольные, синусоидальные, прямоугольные, а также сигналы с меняющейся частотой.

Также с помощью преобразований Фурье были получены спектры исследуемых сигналов. Преобразования Фурье используются в телекоммуникационных технологиях для обработки звука и изображений.

Также были использованы 2 способа подсчета корреляции. Быстрая корреляция показала результат в 4.72 раза лучший метода прямой корреляции.

6. Приложение

```
Fs =  
8e3;  
    t = 0:1/Fs:1;  
    t = t';  
    A = 2;  
    f0 = 1e3;  
    phi = pi/4;  
    s1 = A * cos(2*pi*f0*t + phi);  
    alpha = 1e3;  
    s2 = exp(-alpha*t) .* s1;  
    g = figure();  
    subplot (2,2,1); plot(t(1:100), s2(1:100))  
    subplot (2,2,2); plot(t(1:100), s2(1:100), 'l')  
    subplot (2,2,3); stem(t(1:100), s2(1:100))  
    subplot (2,2,4); stairs(t(1:100), s2(1:100))  
    s = figure();  
    spectrum(s2, Fs);
```

```
Fs =  
500;  
    t = 0:1/Fs:1;  
    t = t';  
    f = [60 120 140];  
    s3 = cos(2*pi*t*f);  
    FFT = fft(s3);  
    g = figure();  
    plot(t(1:50), s3(1:50,:))  
    s = figure();  
    spectrum(s3,Fs);  
    Fs =  
    8e3;  
        t = 0:1/Fs:1;  
        t = t';  
        A = 2;  
        f0 = 1e3;  
        phi = pi/4;  
        alpha = 1e3;  
        T = 0.01;  
        s1 = A * exp (-alpha * t) .* (t >= 0);
```



```

s2 = A * (abs(t) <= T/2);
s3 = A * t / T .* (T >= 0) .* (t <= T);
g1 = figure();
plot(t(1:100), s1(1:100))
g2 = figure();
plot(t(1:100), s2(1:100))
g3 = figure();
plot(t(1:100), s3(1:100))
s1_f = figure();
spectrum(s1,Fs);
xlim([0,2000]);
s2_f = figure();
spectrum(s2,Fs);
xlim([0,2000]);
s3_f = figure();
spectrum(s3,Fs);
xlim([0,1000]);

```

```

Fs =
3e3;

```

```

T = 20e-3;
A = 5;
t = -2*T:1/Fs:2*T;
s = -A * rectpuls(t + T/2, T) + A * rectpuls(t - T/2, T);
g = figure();
plot(t, s)
ylim([-6 6]);
s_f = figure();
spectrum(s',Fs);
xlim([0,1000]);

```

```

Fs =
1e3;

```

```

t = -50e-3:1/Fs:50e-3;
A = 10;
T1 = 20e-3;
T2 = 60e-3;
s = A * (T2*tripuls(t,T2) - T1*tripuls(t,T1))/(T2-T1);
g = figure();
plot(t, s);
s_f = figure();
spectrum(s',Fs);
xlim([0,150]);

```

```

Fs = 1e3;

```

```

t = -0.1:1/Fs:0.1;
f0 = 10;
T = 1/f0;

```

```

s = rectpuls(t,T) .* cos (2*pi*f0*t);
f = -50:50;
sp = T/2 * (sinc((f-f0)*T) + sinc((f+f0)*T));
g1 = figure();
plot(t,s)
ylim([-1.1 1.1]);
g2 = figure();
plot(f, abs(sp))

```

```

Fs = 16e3;

```

```

t = -10e-3:1/Fs:10e-3;
Fc = 4e3;
bw = 0.1;
bwr = -20;
s = gauspuls(t, Fc, bw, bwr);
Nfft = 2^nextpow2(length(s));
sp = fft(s, Nfft);
sp_dB = 20*log10(abs(sp));
f = (0:Nfft-1)/Nfft*Fs;
g1 = figure();
plot(t,s);
g2 = figure();
plot(f(1:Nfft/2), sp_dB(1:Nfft/2))
sp_max_db = 20*log10(max(abs(sp)));
edges = Fc * [1-bw/2 1+bw/2];
hold on;
plot(edges, sp_max_db([1 1])+bwr, 'o');
hold off;

```

```

x =
[0 0
0 1
0 1
0 1
1 1
0 0
0 0
1 0];

```

```

y = [1 0 1];

```

```

xx = x;
yy = zeros(1,length(x));
yy(y == 1) = 1;
yy(y == 0) = -1;

```

```

R = zeros(1,length(xx));

```

```
tic
for i = 1:length(xx)
    R(i) = sum(xx .* circshift(yy, i-1, 2)) / length(xx);
end
toc
```

```
tic
xx = fft(xx);
yy = fft(yy);
xx = conj(xx);
BR = ifft(xx .* yy)/length(xx);
toc
```