

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

Телекоммуникационные технологии

Отчет по лабораторной работе №1
Сигналы телекоммуникационных систем

Работу
выполнил:
Волкова М.Д.
Группа: 33501/3
Преподаватель:
Богач Н.В.

Санкт-Петербург
2017

Содержание

1. Цель работы	2
2. Постановка задачи	2
3. Теоретическая информация	2
3.1. Сигналы	2
3.2. Затухающие сигналы	2
3.3. Одиночные импульсы	2
3.4. Ограниченная полоса частот	3
3.5. Гауссов радиоимпульс	3
3.6. Функция Дирихле	3
3.7. Математические законы изменения мгновенной частоты	3
3.8. Преобразование Фурье	4
3.9. Корреляция	4
4. Ход работы	5
4.1. Генерация затухающего гармонического сигнала	5
4.2. Многоканальный сигнал	7
4.3. Кусочные зависимости	8
4.4. Прямоугольный импульс	12
4.5. Трапецевидный импульс	14
4.6. Импульс с ограниченной полосой частот	15
4.7. Гауссов радиоимпульс	17
4.8. Последовательности импульсов	18
4.9. Генерация периодических сигналов	21
4.10. Функция Дирихле	24
4.11. Сигнал с меняющейся частотой	27
4.12. Сравнение методов корреляции	29
5. Выводы	29

1. Цель работы

Познакомиться со средствами генерации и визуализации простых сигналов.

2. Постановка задачи

В командном окне MATLAB и в среде Simulink промоделировать сигналы из Главы 3, сс. 150–170.

3. Теоретическая информация

3.1. Сигналы

В MatLab аналоговый сигнал можно представить как вектор дискретных отсчетов функции. Вторым вектором удобно формировать как возрастающую последовательность чисел, шаг между которыми есть величина, обратная частоте дискретизации. Определив вектор отсчетов времени и константы: амплитуда, частота колебаний и тому подобное, мы можем задать вектор значений функции в известных нам моментах времени.

3.2. Затухающие сигналы

$$s2 = \exp^{-\alpha t} s1 \quad (1)$$

где $s1$ - гармонический сигнал

3.3. Одиночные импульсы

Встроенная функция `rectpuls` работает по следующему принципу:

$$y = \begin{cases} 1, & -\frac{width}{2} \leq t \leq \frac{width}{2} \\ 0, & t < -\frac{width}{2}, t > \frac{width}{2} \end{cases} \quad (2)$$

где y -возвращаемое значение, t -вектор значений времени, сгенерированный заранее, $width$ -ширина импульса.

Встроенная функция `tripuls` работает по следующему принципу:

$$y = \begin{cases} \frac{2t+width}{width(skew+1)}, & -\frac{width}{2} \leq t < \frac{width*skew}{2} \\ \frac{2t-width}{width(skew-1)}, & \frac{width*skew}{2} \leq t < \frac{width}{2} \\ 0, & |t| > \frac{width}{2} \end{cases} \quad (3)$$

где параметр $skew$ - коэффициент асимметрии импульса, а другие параметры имеют те же значения.

3.4. Ограниченная полоса частот

Для формирования сигнала, имеющего ограниченный спектр, используется функция:

$$y = \frac{\sin(\pi x)}{\pi x} \quad (4)$$

Спектральная функция сигнала в этом случае имеет прямоугольный вид:

$$y = \begin{cases} 1, & |\omega| < \pi \\ 0, & |\omega| > \pi \end{cases} \quad (5)$$

3.5. Гауссов радиоимпульс

Функция для получения отсчетов радиоимпульса имеет внутри себя следующую математическую формулу:

$$y = \exp^{-\alpha t^2} \cos(2\pi f_c t) \quad (6)$$

А спектр такого сигнала можно получить путем преобразования Фурье, формула которого представлена ниже:

$$S(\omega) = \frac{1}{2} \sqrt{\frac{\pi}{\alpha}} \left(\exp^{-\frac{(\omega+2\pi f_c)^2}{4\alpha}} + \exp^{-\frac{(\omega-2\pi f_c)^2}{4\alpha}} \right) \quad (7)$$

3.6. Функция Дирихле

Функция Дирихле описывается формулой:

$$\text{diric}_n(x) = \frac{\sin(n\frac{x}{2})}{n\sin(\frac{x}{2})} \quad (8)$$

где n - целое положительное число.

Функцию Дирихле еще называют периодической sinc функцией. При четном (нечетном) значении параметра n функция приобретает вид:

$$\text{diric}_n(x) = \sum_{k=-\infty}^{\infty} \text{sinc}\left(n\left(\frac{t}{2\pi} - k\right)\right) \quad (9)$$

$$\text{diric}_n(x) = \sum_{k=-\infty}^{\infty} (-1)^k \text{sinc}\left(n\left(\frac{t}{2\pi} - k\right)\right) \quad (10)$$

3.7. Математические законы изменения мгновенной частоты

Формула линейного закона:

$$f(t) = f_0 + \beta t, \text{ где } \beta = \frac{f_1 - f_0}{t_1}, \quad (11)$$

квадратичного:

$$f(t) = f_0 + \beta t^2, \text{ где } \beta = \frac{f_1 - f_0}{t_1^2}, \quad (12)$$

логарифмического:

$$f(t) = f_0 + e^{\beta t}, \text{ где } \beta = \frac{\ln(f_1 - f_0)}{t_1}. \quad (13)$$

3.8. Преобразование Фурье

Формула прямого преобразования Фурье:

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt, \quad (14)$$

обратного преобразования Фурье:

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{j\omega t} d\omega. \quad (15)$$

3.9. Корреляция

Метод взаимной корреляции используют для нахождения синхропосылки в сигнале. Значение корреляции двух векторов x и y строится по формуле:

$$R = \frac{1}{N} \sum_{i=1}^N x_i * y_i \quad (16)$$

где N - длина векторов x и y .

Если искомая посылка y короче передаваемого вектора x , то она дополняется нулями до необходимой длины. Для определения позиции синхропосылки в передаче необходимо сдвигать вектор y пошагово, на каждом шаге высчитывая значение корреляции и, таким образом, получая вектор значений корреляции. Максимальное значение этого вектора будет соответствовать сдвигу, при котором была найдена искомая посылка или же максимально похожая на нее часть вектора.

Для ускорения вычисления корреляции, особенно в больших посылках, применим метод быстрой корреляции:

$$R = \frac{1}{N} F_D^{-1} [X^* * Y] \quad (17)$$

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

Данная формула позволяет найти вектор значений взаимной корреляции двух векторов быстрее.

4. Ход работы

4.1. Генерация затухающего гармонического сигнала

Листинг 1: Код Matlab

```
1 t = (0:1/8e3:1)';  
2  
3 s2 = exp(-1e3*t) .* (4 * cos(2*pi*1e3*t + pi/2));  
4 FFT = fft(s2);  
5  
6 subplot(2,2,1);  
7 plot(t(1:100), s2(1:100))  
8  
9 subplot(2,2,2);  
10 plot(t(1:100), s2(1:100), 'r')  
11  
12 subplot(2,2,3);  
13 stem(t(1:100), s2(1:100))  
14  
15 subplot(2,2,4);  
16 stairs(t(1:100), s2(1:100))  
17 figure;  
18 plot(t(1:8001), FFT(1:8001))
```

Здесь представлен код программы, генерирующей затухающий сигнал.

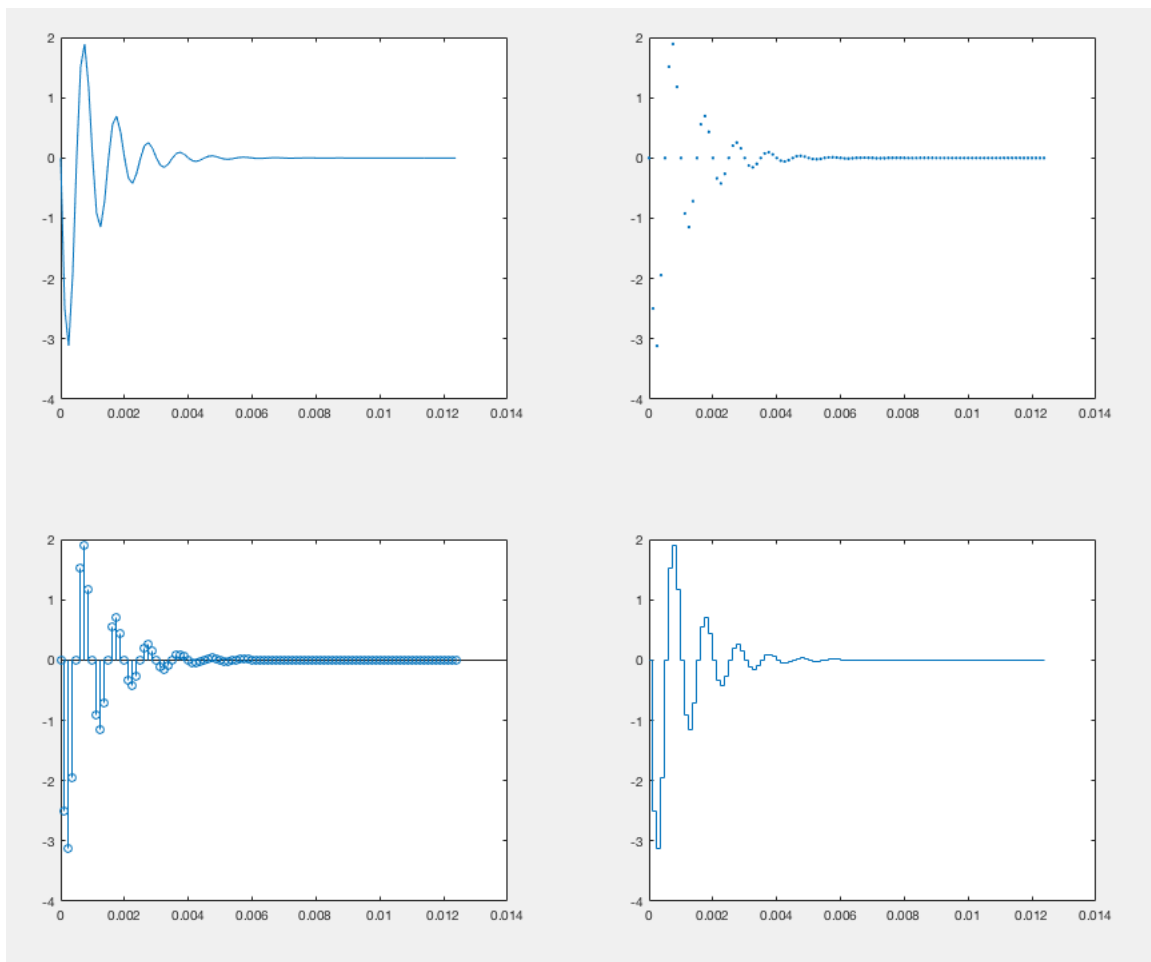


Рис. 4.1.1. Графики сигнала

На первом графике виден обычный вид затухающего гармонического сигнала, построенный средой Matlab по дискретным отсчетам. Вторым графиком представляется из себя точки того же сигнала, соответствующие дискретным отсчетам. Третий график представляет собой те же точки, как некоторые значения, отклоненные от нулевого. Четвертый график — ступенчатый график.

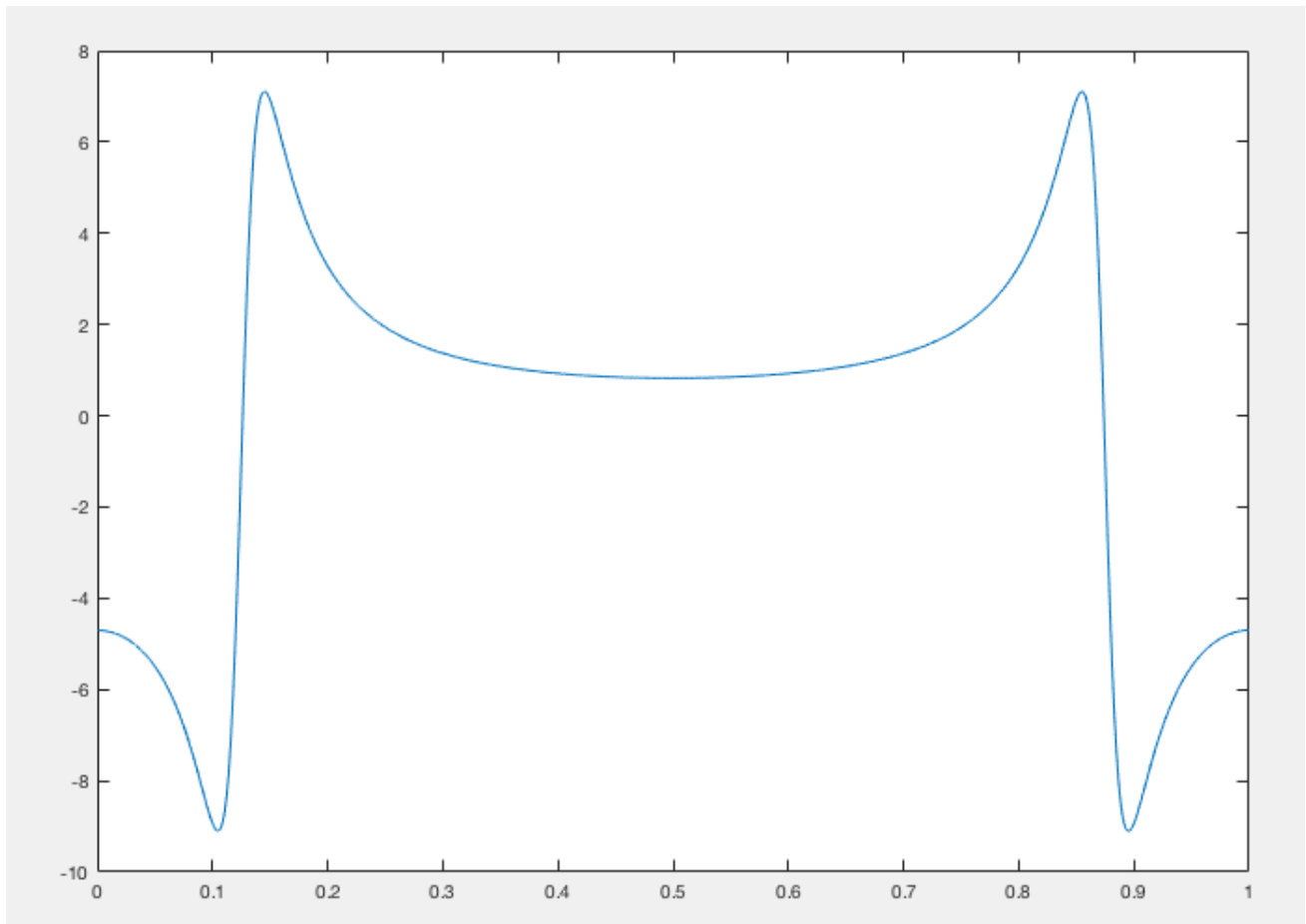


Рис. 4.1.2. Спектр сигнала

Спектр представленного выше сигнала получен с помощью разложение в ряд Фурье.

4.2. Многоканальный сигнал

Листинг 2: Код в МатЛаб

```

1 t = (0:1/8e3:1)';
2 s3 = cos(2*pi*t*[400_1000_1200]);
3 FFT = fft(s3);
4
5 plot(t(1:70), s3(1:70,:))
6 figure;
7
8 plot(t(1:8000), FFT(1:8000,:))

```

Данный код генерирует сразу несколько сигналов, записываемых в одну матрицу, различающихся по частоте.

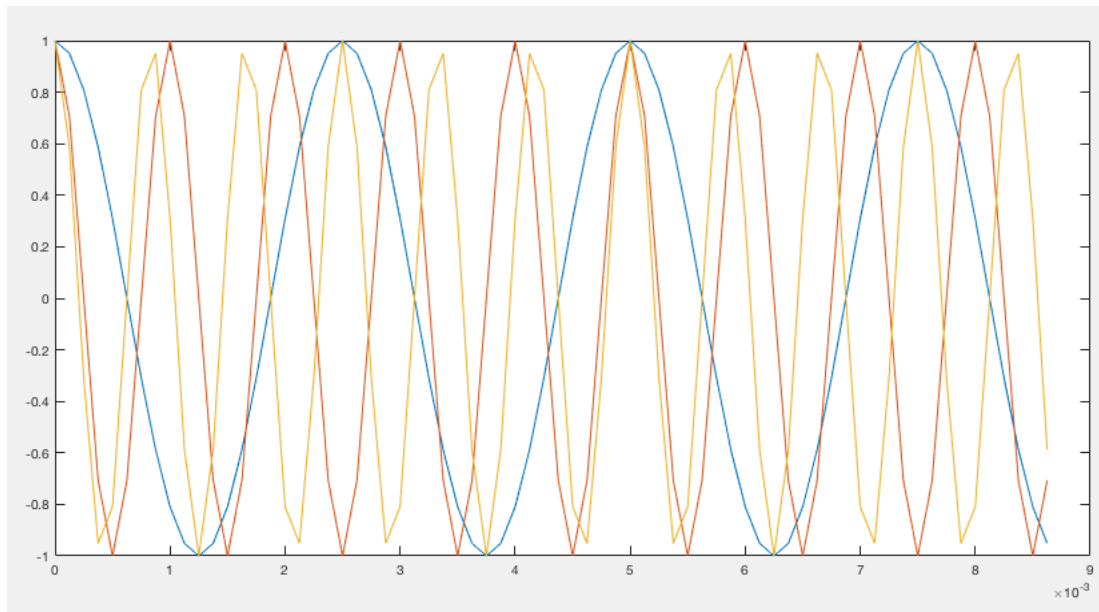


Рис. 4.2.1. График сигналов

На данном графике видно несколько гармонических сигналов, различающихся по частоте.

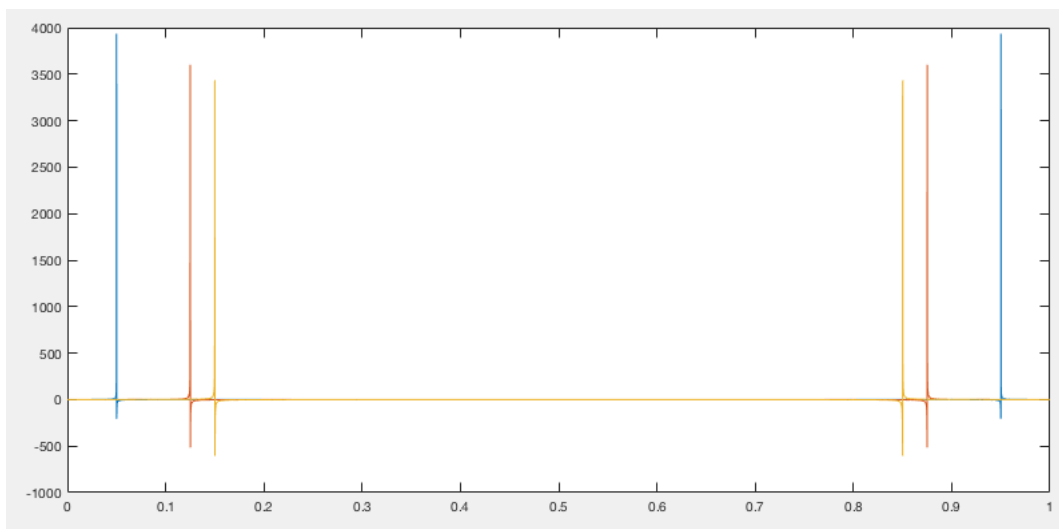


Рис. 4.2.2. Спектр сигналов

На этом рисунке видны спектры данных синусоид. Линии спектра сигнала с более высокой частотой располагаются ближе к нулю.

4.3. Кусочные зависимости

Листинг 3: Код в МатЛаб

```

1 t = (0:1/8e3:1)';
2 T = 0.009;
3
4 s1 = 4 * exp(-1e3 * t) .* (t >= 0);
5 FFT1 = fft(s1);

```

```

6
7 s2=_4*_(_abs(t)_<=T/2);
8 FFT2=_fft(s2);
9
10 s3=_4*_t/_T*_(_T>=0)*_(_t<=T);
11 FFT3=_fft(s3);
12
13 figure(1);
14 plot(t(1:100),_s1(1:100))
15
16 figure(2);
17 plot(t(1:100),_s2(1:100))
18
19 figure(3);
20 plot(t(1:100),_s3(1:100))
21
22 figure(4);
23 plot(t(1:8001),_FFT1(1:8001))
24
25 figure(5);
26 plot(t(1:8001),_FFT2(1:8001))
27
28 figure(6);
29 plot(t(1:8001),_FFT3(1:8001))

```

Данный участок кода генерирует и выводит на экран односторонний экспоненциальный импульс, прямоугольный импульс и несимметричный треугольный импульс согласно заданным уравнениям.

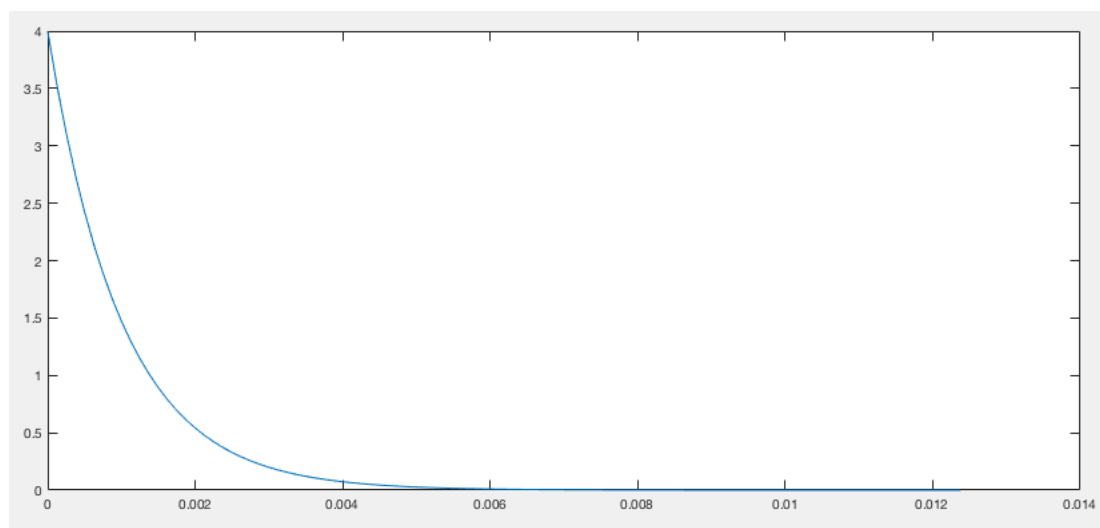


Рис. 4.3.1. Экспоненциальный импульс

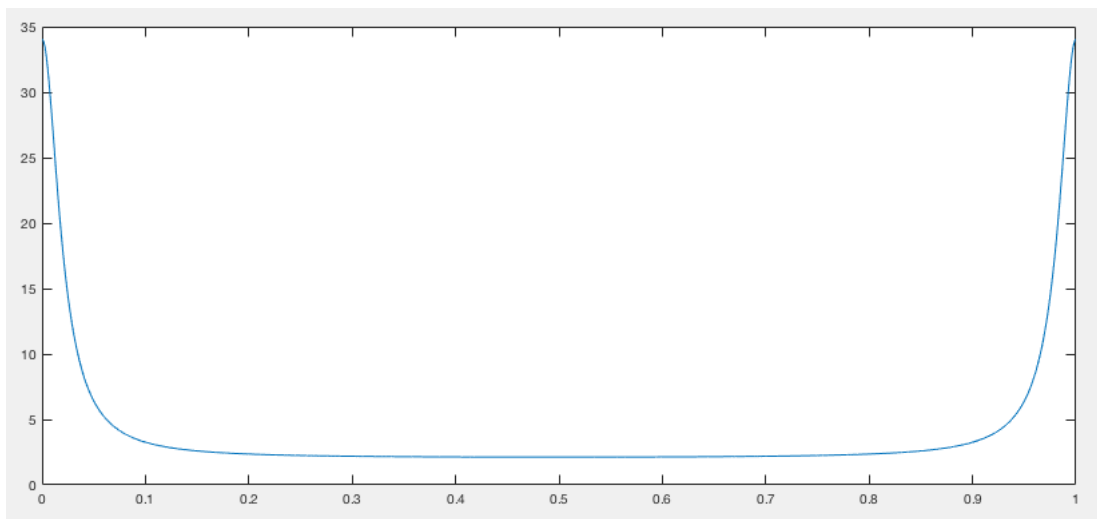


Рис. 4.3.2. Спектр экспоненциального импульса

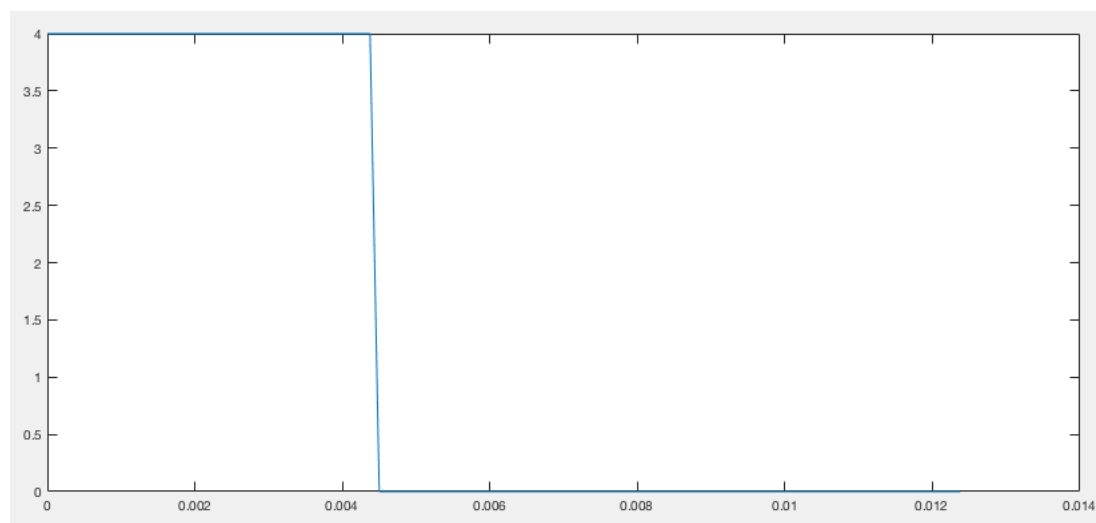


Рис. 4.3.3. Прямоугольный импульс

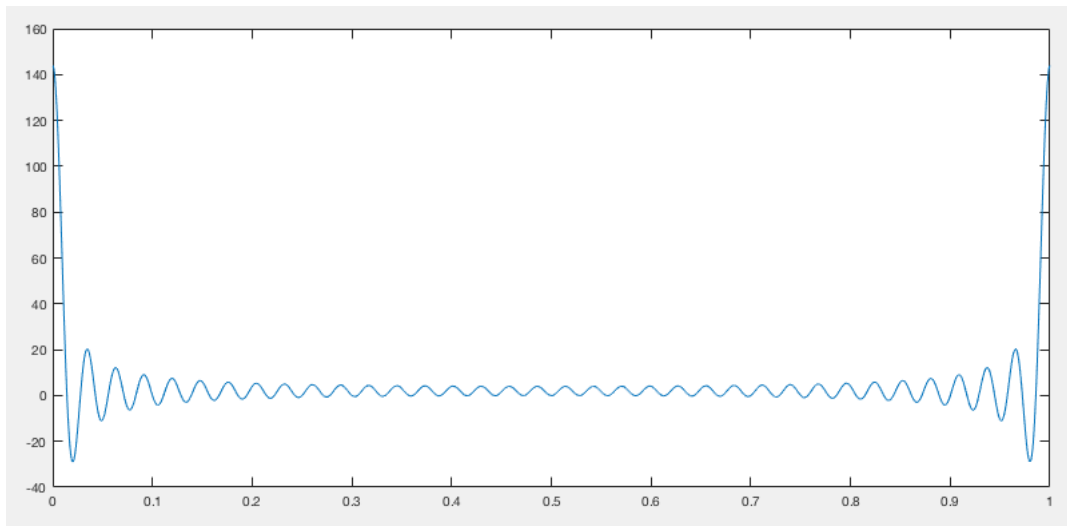


Рис. 4.3.4. Спектр прямоугольного импульса

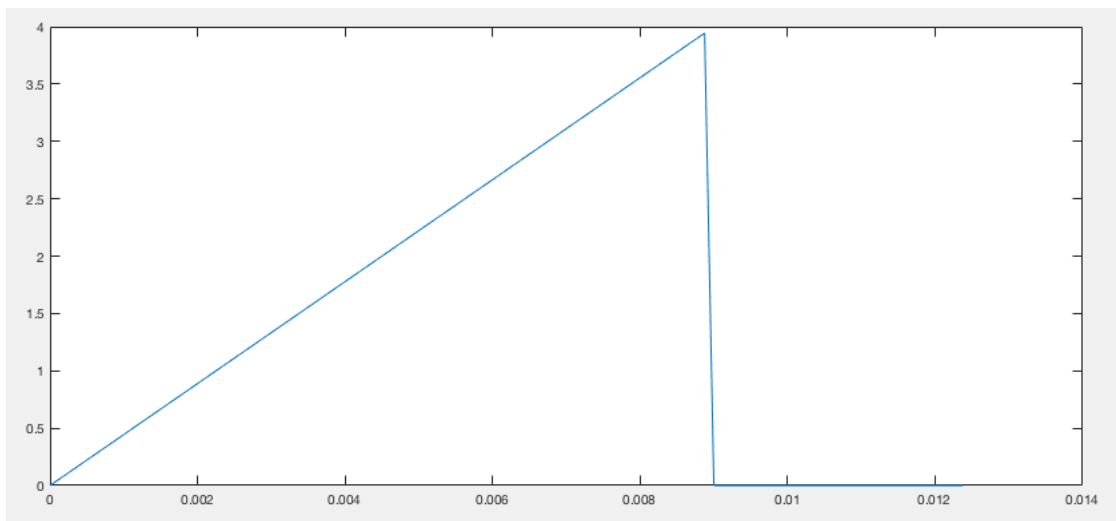


Рис. 4.3.5. Несимметричный треугольный импульс

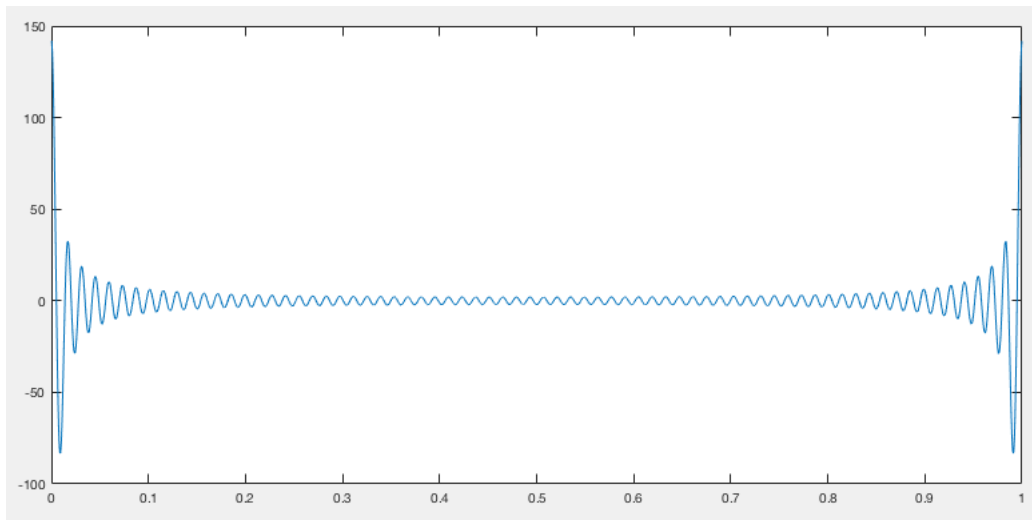


Рис. 4.3.6. спектр несимметричного треугольного импульса

На рисунках 4.3.1 — 4.3.6 представлены графики сгенерированных сигналов, выведенных с помощью стандартной функции построения графиков в MatLab и спектры сигналов.

4.4. Прямоугольный импульс

Листинг 4: Код в МатЛаб

```

1 t = -40e-3:1/1e3:40e-3;
2 T = 20e-3;
3 s = -5 * rectpuls(t + T/2, T) + 5 * rectpuls(t-T/2, T);
4
5 plot(t, s)
6 ylim([-6 6]);
7 FFT = fft(s);
8 figure;
9
10 plot(t(1:81), FFT(1:81))

```

Данный сигнал генерируется путем соединения двух прямоугольных импульсов, с использованием встроенных функций.

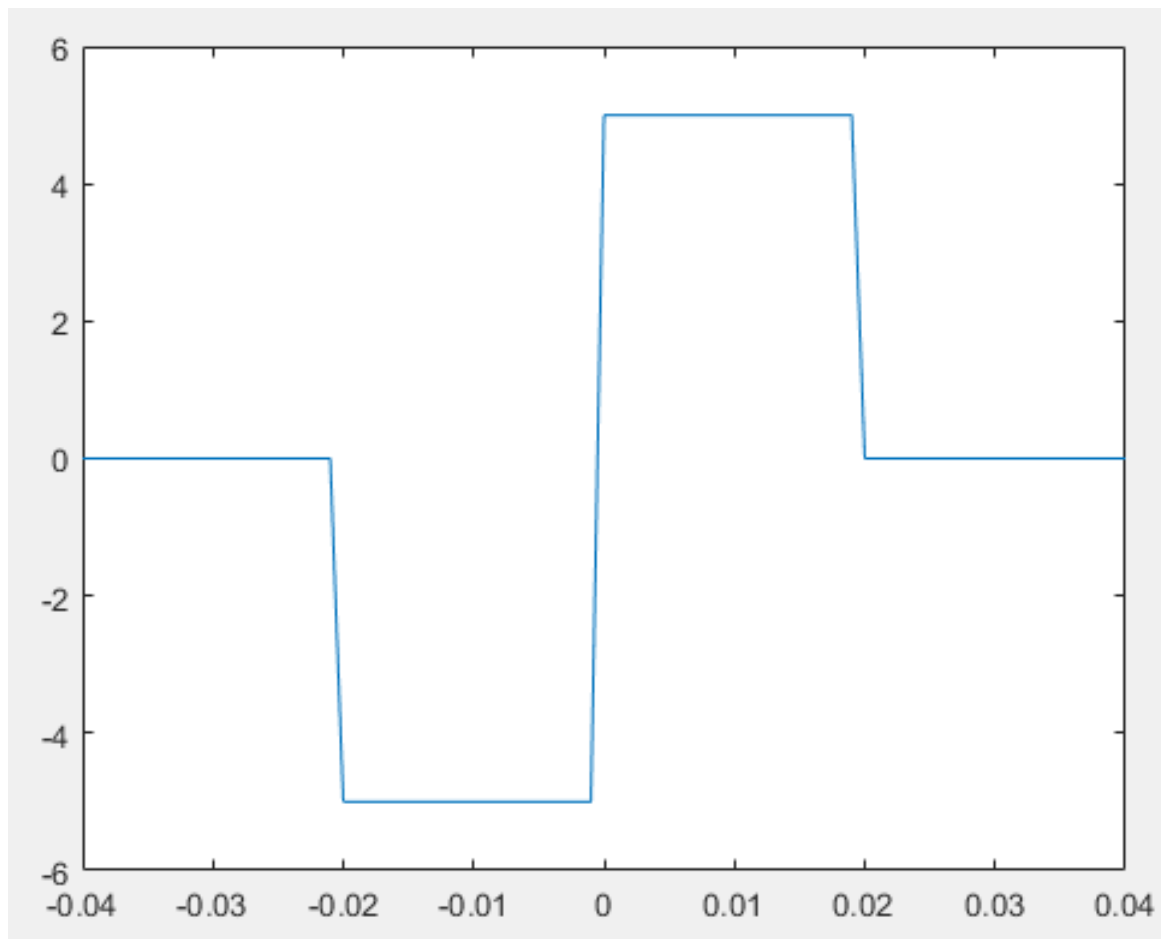


Рис. 4.4.1. Прямоугольные импульсы

На данном рисунке представлен график прямоугольных импульсов.

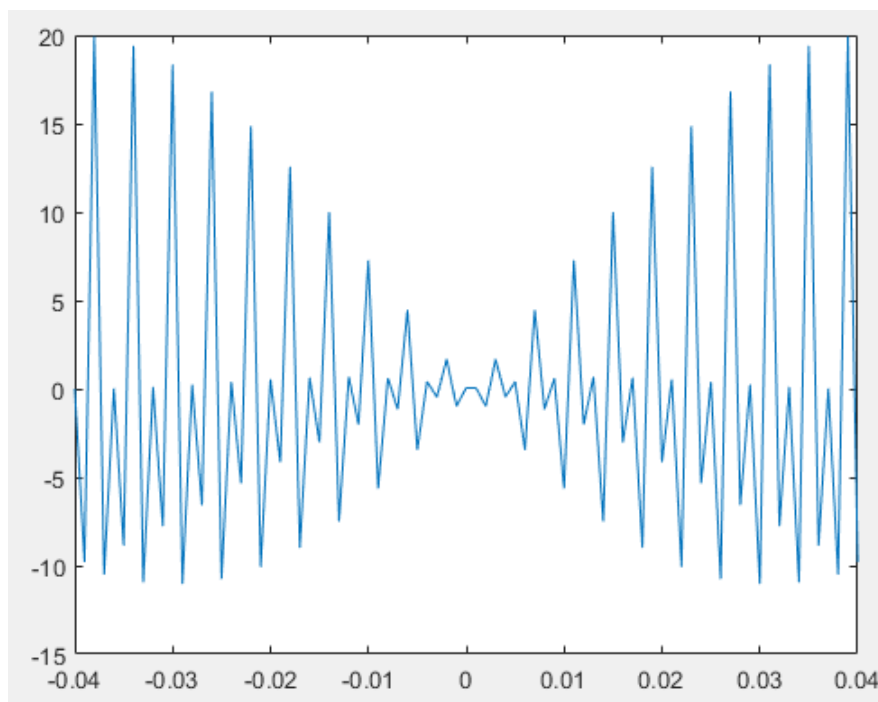


Рис. 4.4.2. Спектр прямоугольных импульсов

Спектр прямоугольного импульса получен как разложение сигнала в ряд Фурье.

4.5. Трапецевидный импульс

Листинг 5: Код в MatLab

```
1 t = -50e-3:1/1e3:50e-3;  
2 T1 = 20e-3;  
3 T2 = 60e-3;  
4 s = 10 * (T2*tripuls(t,T2) - T1*tripuls(t,T1))/(T2-T1);  
5  
6 plot(t, s);  
7 FFT = fft(s);  
8 figure;  
9  
10 plot(t(1:101), FFT(1:101))
```

Данный сигнал генерируется разностью двух треугольных импульсов, с использованием встроенной функции `tripuls`.

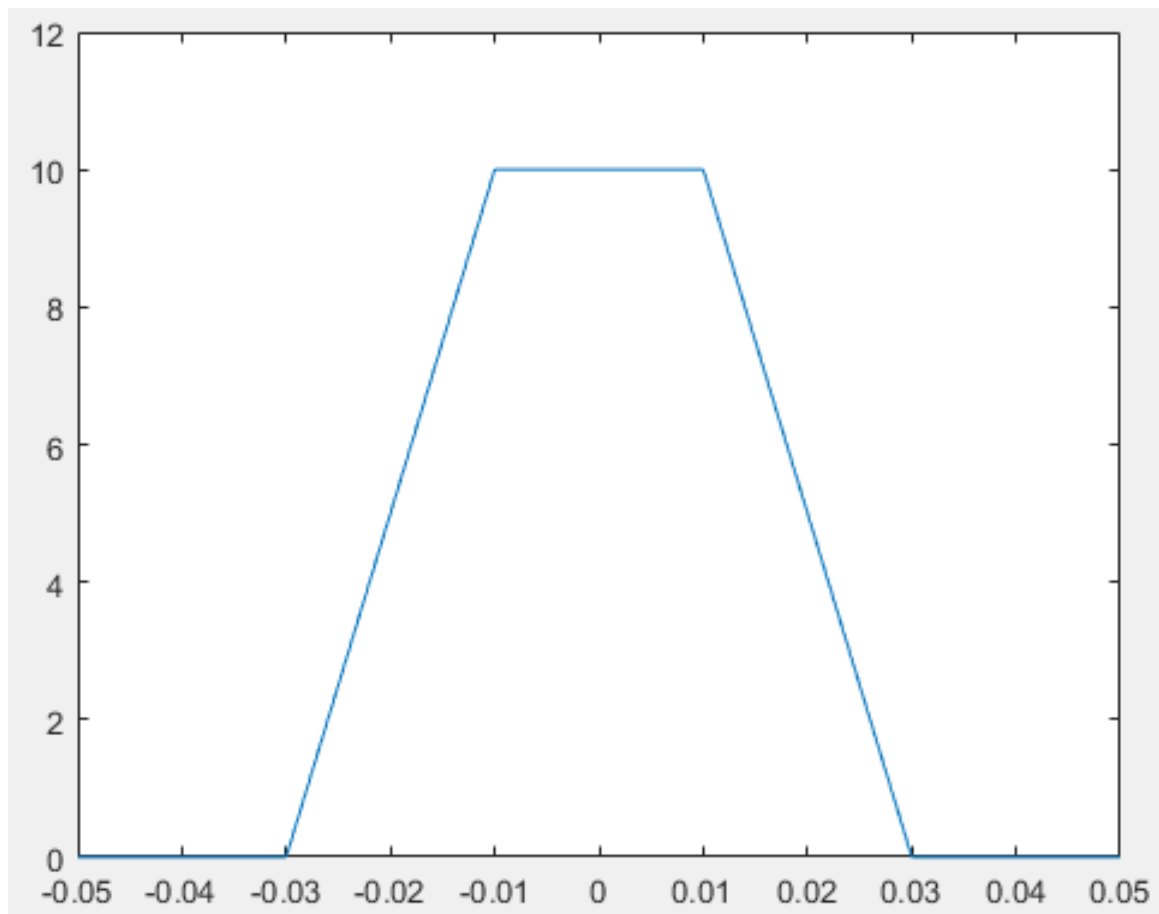


Рис. 4.5.1. Трапецевидный импульс

На данном рисунке представлен вид трапецевидного импульса в среде MatLab.

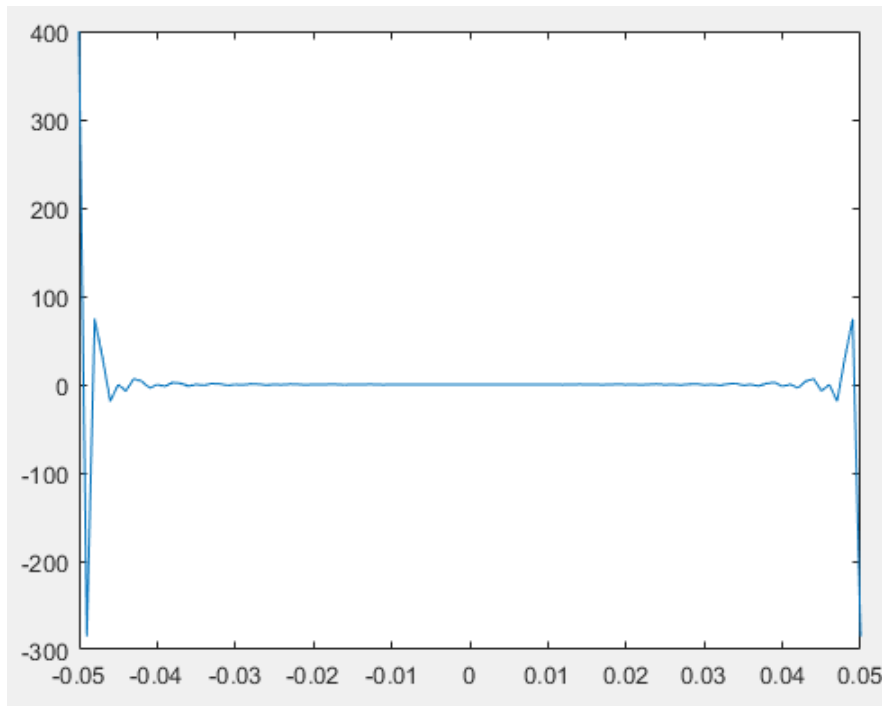


Рис. 4.5.2. Спектр трапецевидного импульса

На рисунке представлен спектр трапецевидного импульса.

4.6. Импульс с ограниченной полосой частот

Листинг 6: Код в MatLab

```

1 t = -0.1:1/1e3:0.1;
2 T = 1/5;
3 f = -50:50;
4
5 plot(t,rectpuls(t,T) .* cos (2*pi*5*t))
6 ylim([-1.1 1.1]);
7 figure;
8
9 plot(f, abs(T/2 * (sinc((f-5)*T) + sinc((f+5)*T))))

```

Данный код генерирует сигнал, у которого спектр ограничен по частоте. Затем выводится и сам спектр данного сигнала, что можно увидеть на рисунках ниже:

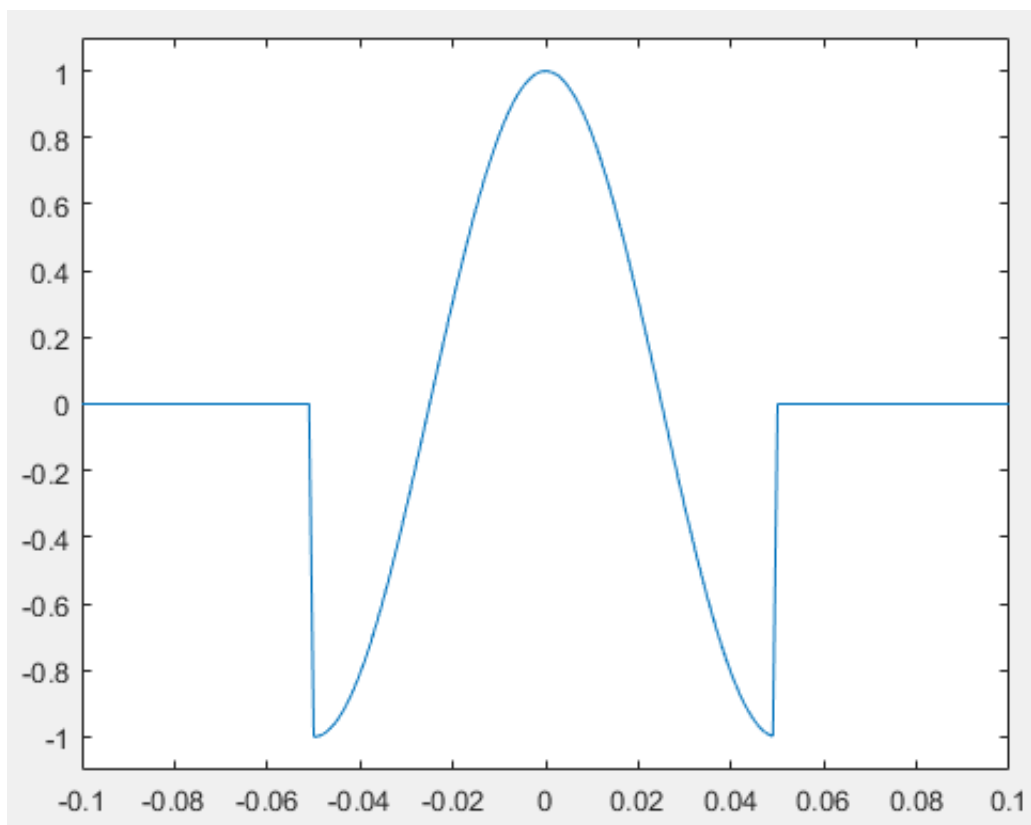


Рис. 4.6.1. Сигнал с ограниченным спектром

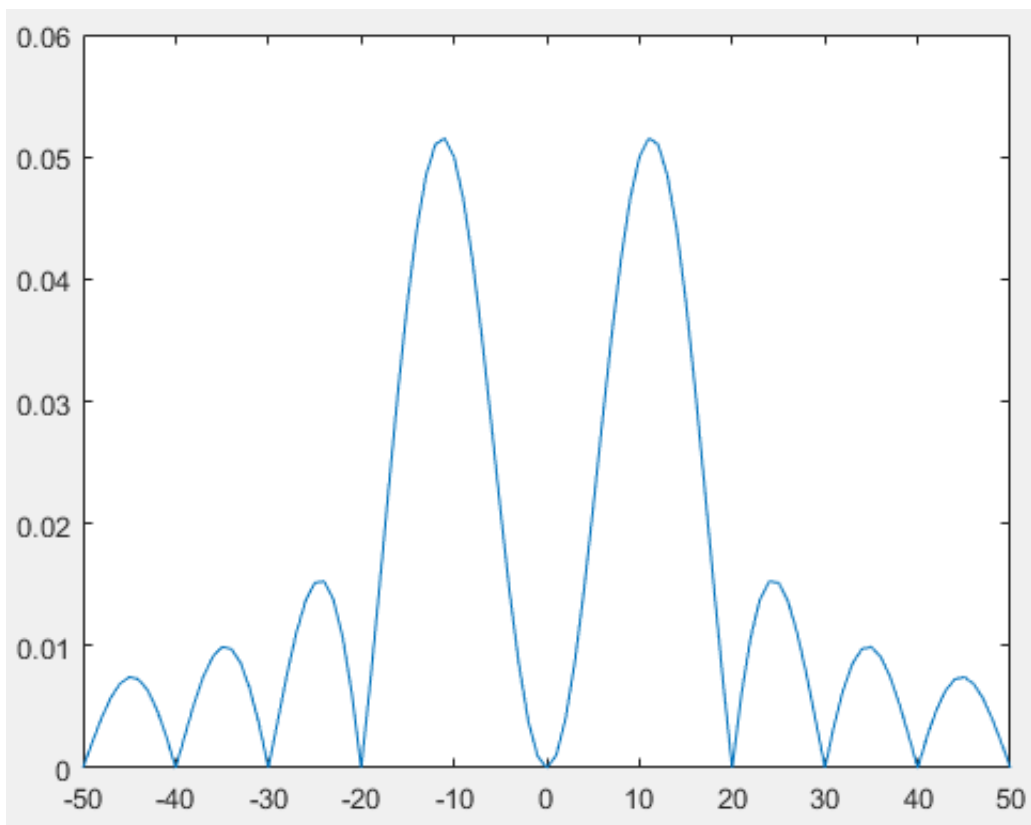


Рис. 4.6.2. Ограниченный спектр сигнала

Спектр сигнала получен с помощью функции sinc.

4.7. Гауссов радиоимпульс

Листинг 7: Код в MatLab

```
1 t = -10e-3:1/16e3:10e-3;  
2 s = gausspuls(t, 4e3, 0.1, -20);  
3 Nfft = 2^nextpow2(length(s));  
4 f = (0:Nfft-1)/Nfft*16e3;  
5  
6 sp = fft(s, Nfft);  
7 sp_dB = 20*log10(abs(sp));  
8 sp_max_db = 20*log10(max(abs(sp)));  
9  
10 plot(t,s);  
11 figure;  
12 plot(f(1:Nfft/2), sp_dB(1:Nfft/2))  
13  
14 hold on;  
15 plot(4e3 * [1-0.1/2 1+0.1/2], sp_max_db([1 1])+ -20, 'o');  
16 hold off;
```

Данный код генерирует Гауссов радиоимпульс с помощью встроенной функции gausspuls, а затем находит спектр этого сигнала, выражая его в дБ.

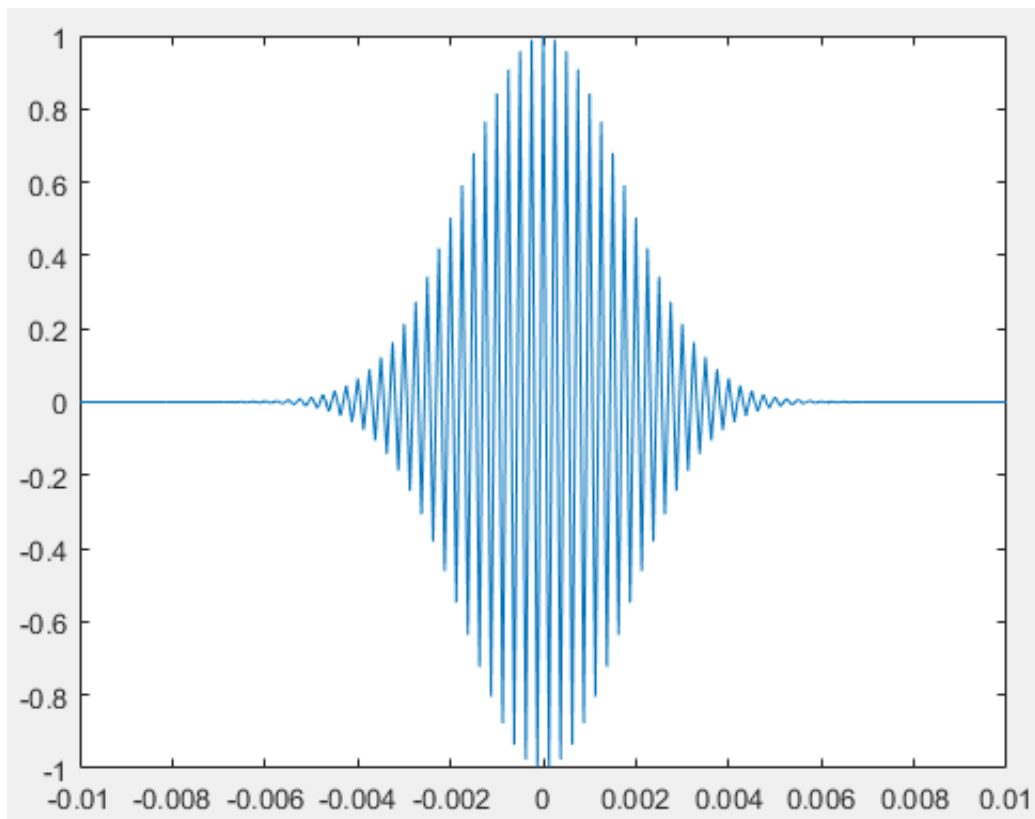


Рис. 4.7.1. Гауссов радиоимпульс

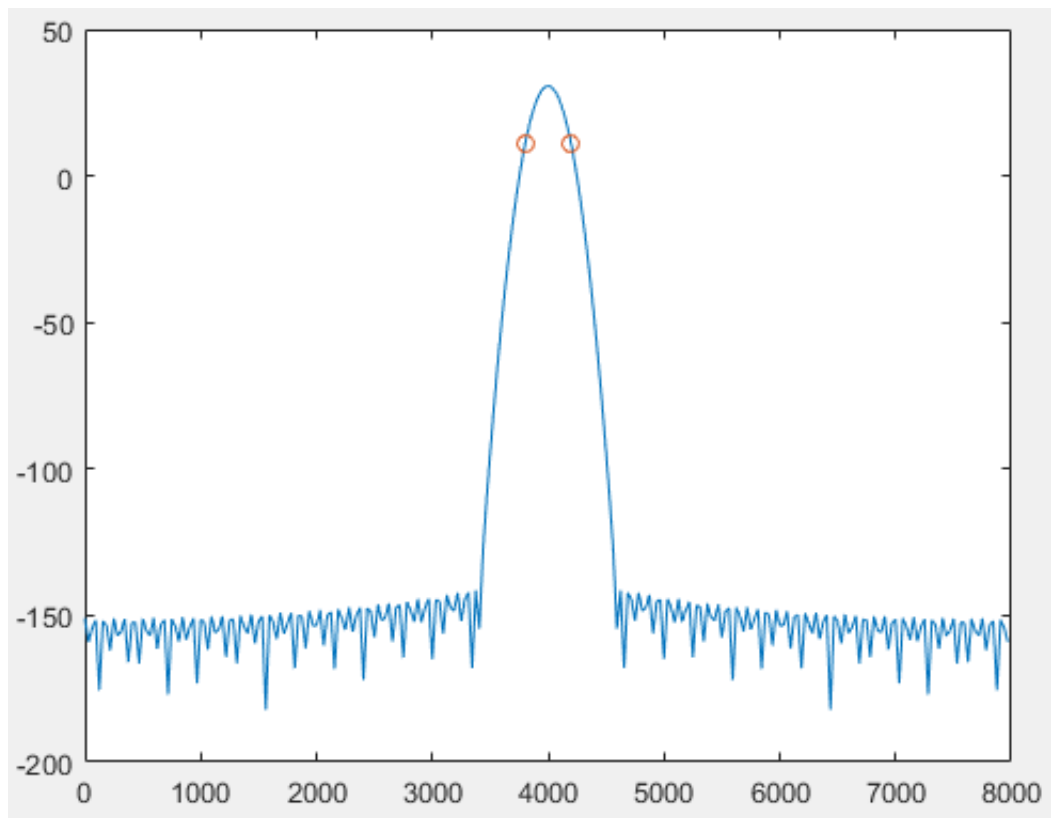


Рис. 4.7.2. Амплитудный спектр радиоимпульса

На графике спектра также отмечены расчетные границы этого спектра.

4.8. Последовательности импульсов

Листинг 8: Код в MatLab

```

1 t = 0:1/1e3:0.5;
2
3 p = [20 80 160 260 380]'*1e-3;
4 p(:,2) = 0.8.^(0:4)';
5 y = pulstran(t, p, 'tripuls', 20e-3);
6
7 plot(t,y)
8 FFT = fft(y);
9 figure;
10 plot(t(1:501), FFT(1:501))

```

Данный код генерирует треугольные импульсы с заданными амплитудами, через заданные промежутки времени.

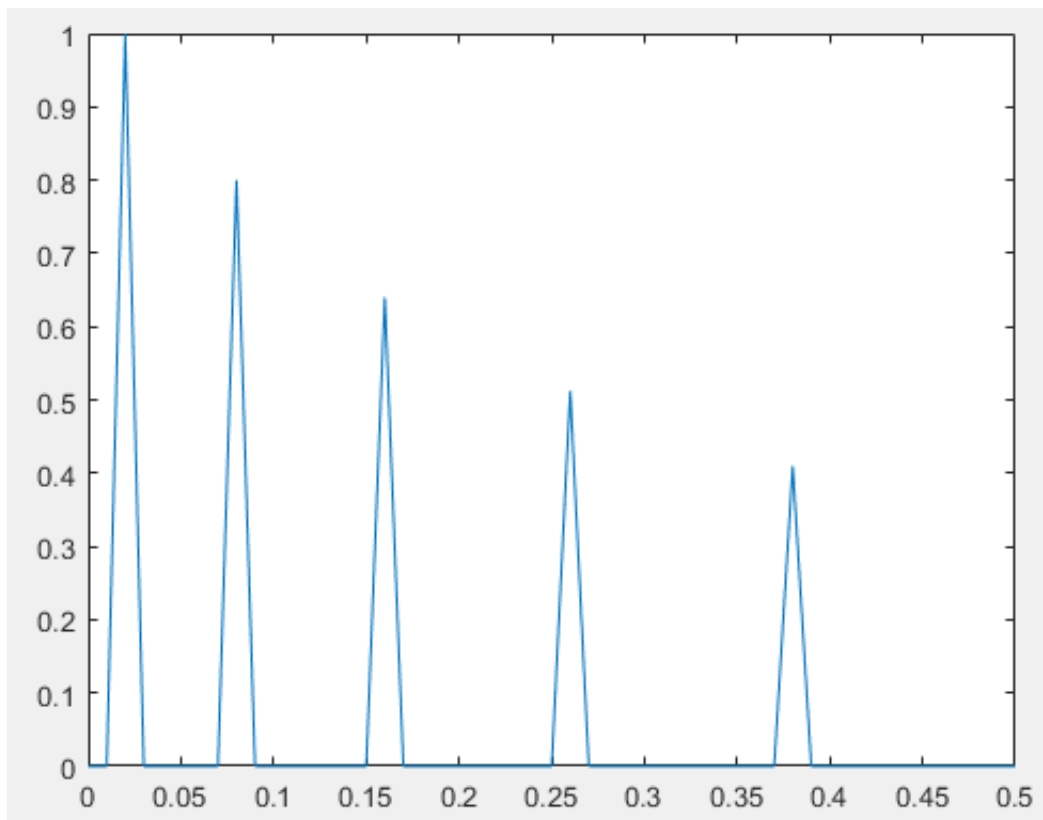


Рис. 4.8.1. Треугольные импульсы

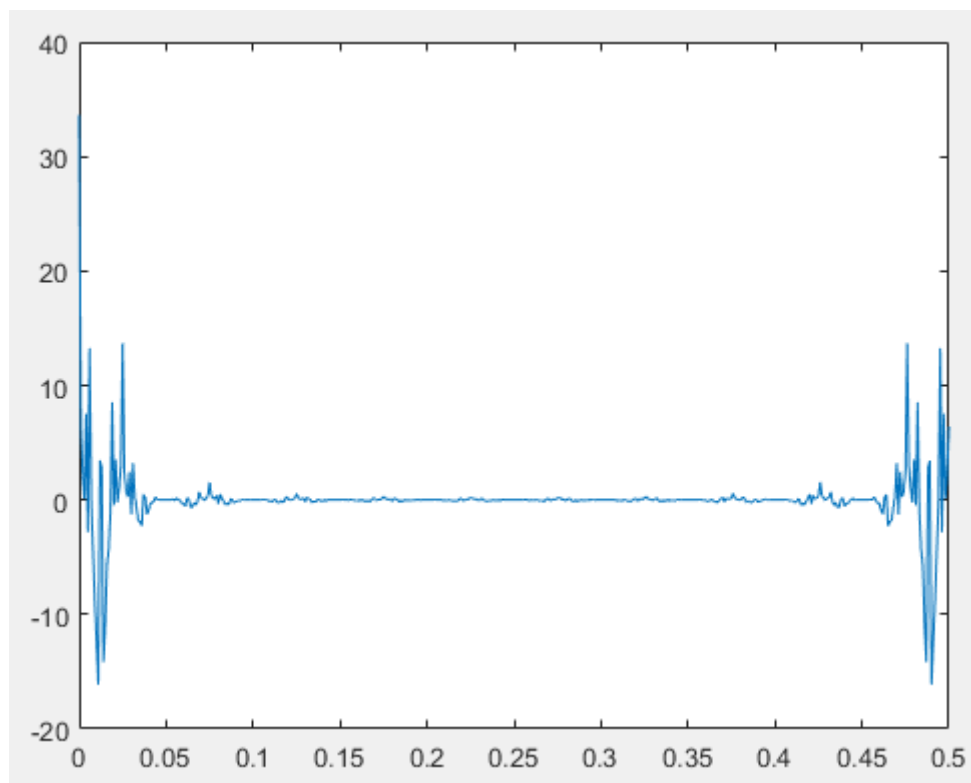


Рис. 4.8.2. Спектр импульсов

На рисунках представлены - треугольные импульсы, сгенерированные с помощью

встроенной функции, (4.8.1) и спектр этого сигнала (4.8.2).

Листинг 9: Код в MatLab

```
1 t0 = 0:1/400:60e-3;  
2  
3 t = 0:1/1e3:0.5;  
4 p = (1:6)' * 64e-3;  
5 p(:,2) = 0.6.^(0:5)';  
6 y = pulstran(t, p, sin(pi*t0/60e-3).^2, 400);  
7  
8 plot(t,y)  
9 FFT = fft(y);  
10 figure;  
11  
12 plot(t(1:501), FFT(1:501))
```

Данный код генерирует и выводит гармонические импульсы.

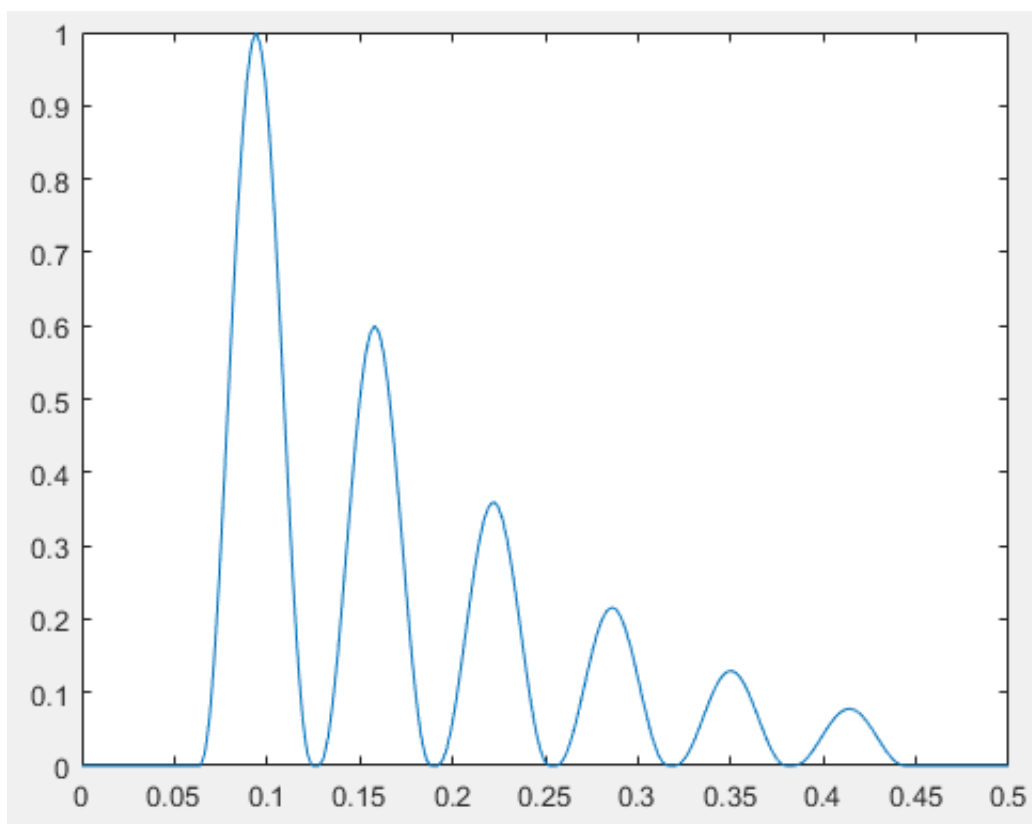


Рис. 4.8.3. Гармонические импульсы

Данные импульсы сгенерированы функцией `pulstran` из вектора отсчетов одиночного импульса.

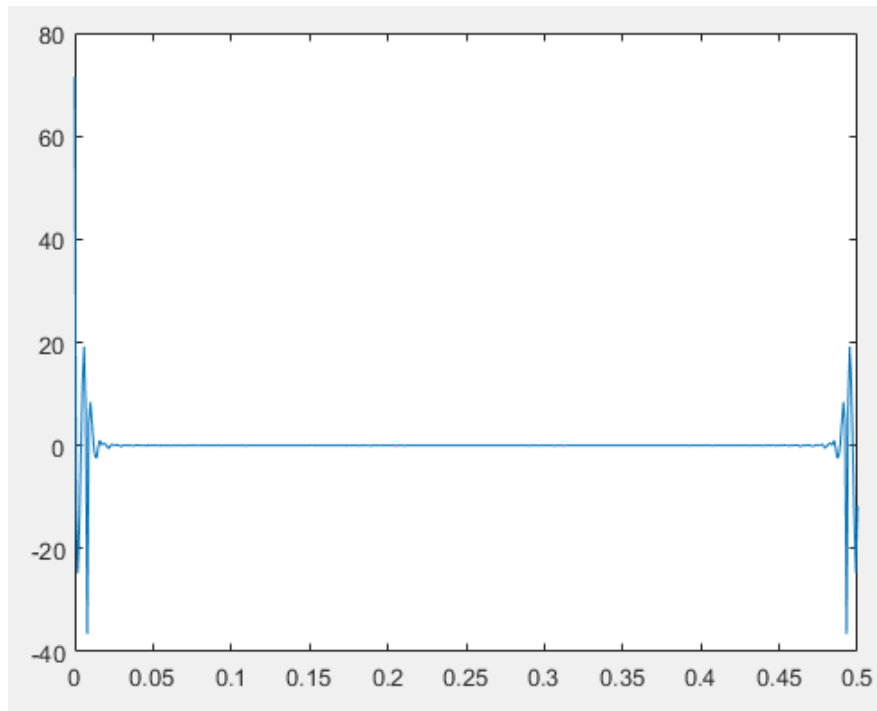


Рис. 4.8.4. Спектр гармонических импульсов

На рисунке представлен спектр гармонических импульсов.

4.9. Генерация периодических сигналов

Листинг 10: Код в MatLab

```

1 t = -10e-3:1/1e3:50e-3;
2
3 s = (square(2*pi*t*50, 50*5e-3*100) + 1) * 3/2;
4
5 plot(t, s)
6 ylim([0 5]);
7 FFT = fft(s);
8 figure;
9
10 plot(t(1:61), FFT(1:61))

```

Данная программа создает и выводит на экран периодически повторяющиеся прямоугольные сигналы, создаваемые с помощью функции `square`.

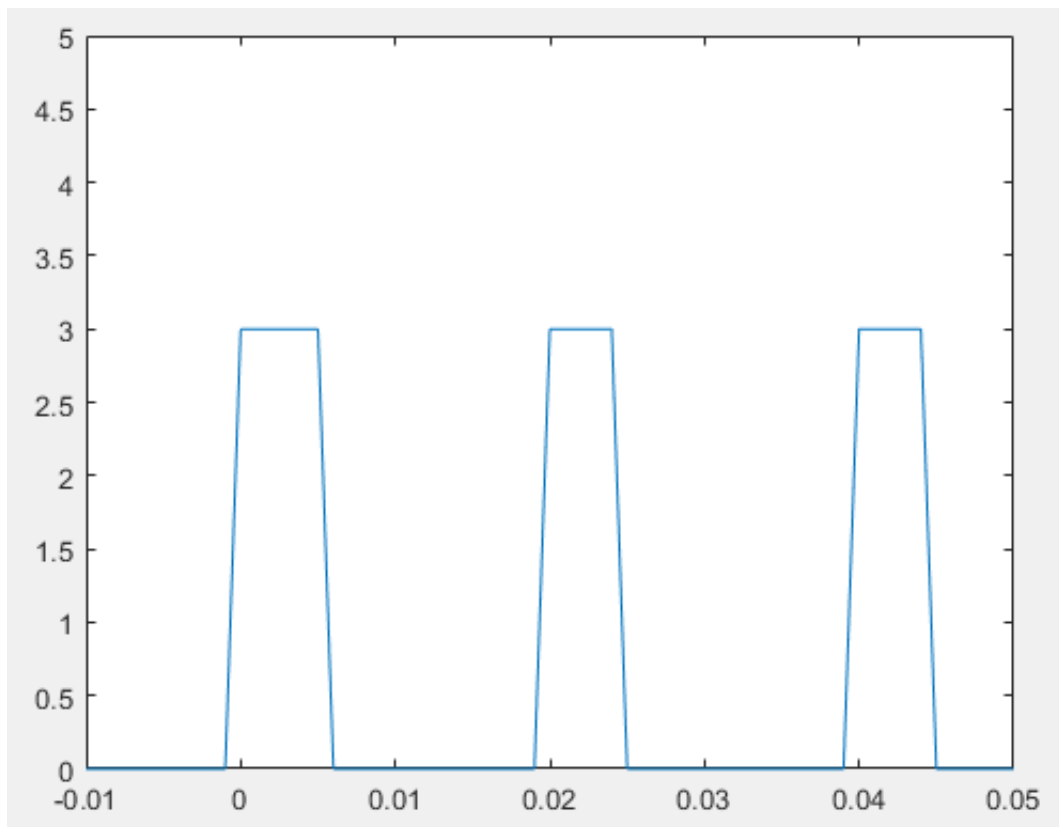


Рис. 4.9.1. Периодические прямоугольные импульсы

Импульсы обладают одинаковой длительностью и временем паузы между ними, что можно увидеть более отчетливо, если увеличить частоту дискретизации.

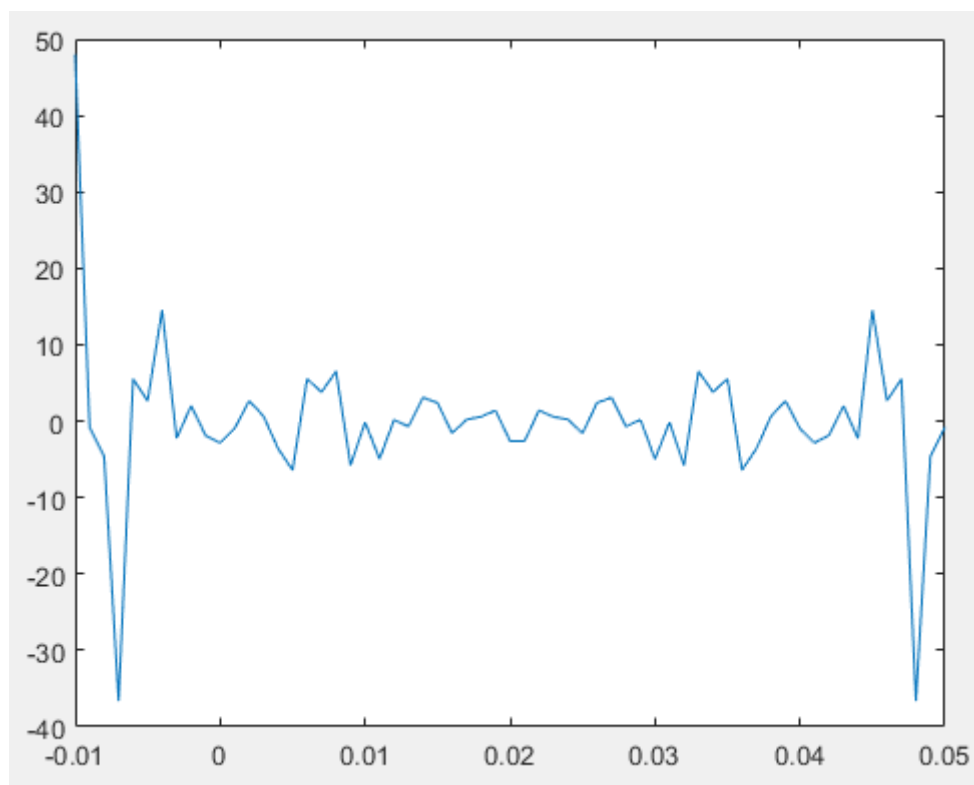


Рис. 4.9.2. Спектр прямоугольных импульсов

Листинг 11: Код в MatLab

```
1 t = -25e-3:1/1e3:125e-3;  
2 s = (sawtooth(2*pi*t/50e-3, 1-5e-3/50e-3) - 1) * 5/2;  
3  
4 plot(t,s);  
5 FFT = fft(s);  
6 figure;  
7  
8 plot(t, FFT)
```

Эта программа, используя функцию `sawtooth`, создает импульсы треугольной формы с заданными параметрами.

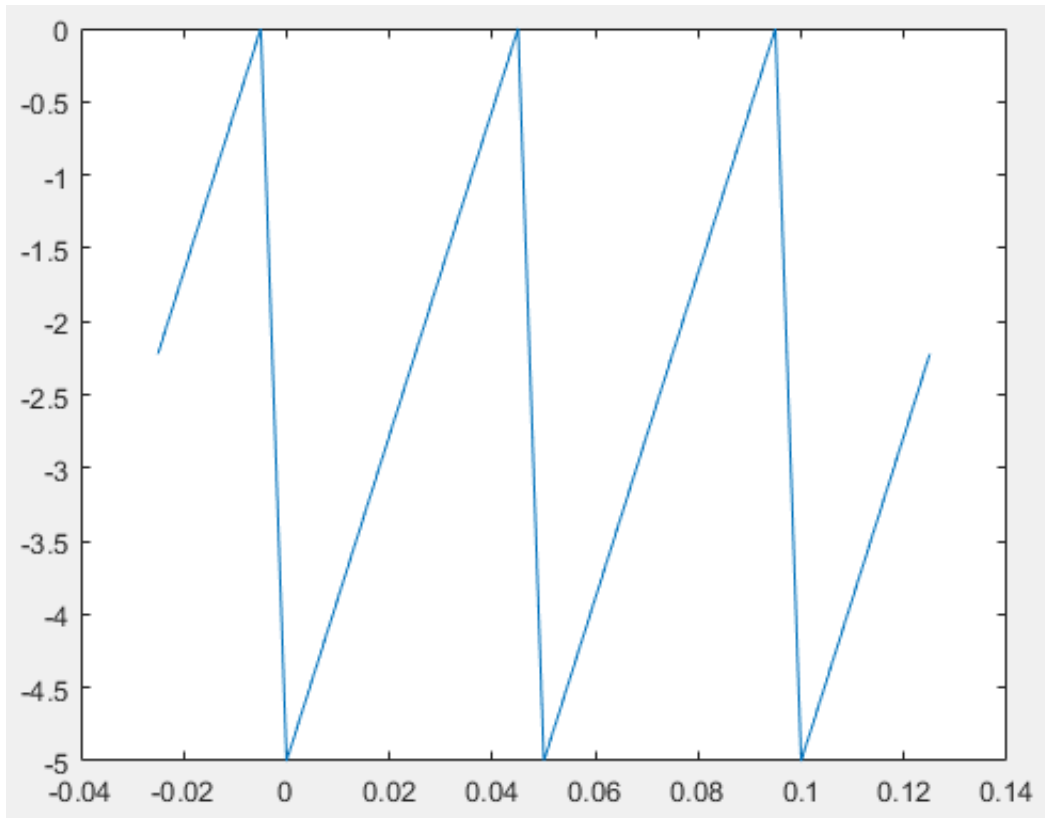


Рис. 4.9.3. Треугольные импульсы `sawtooth`

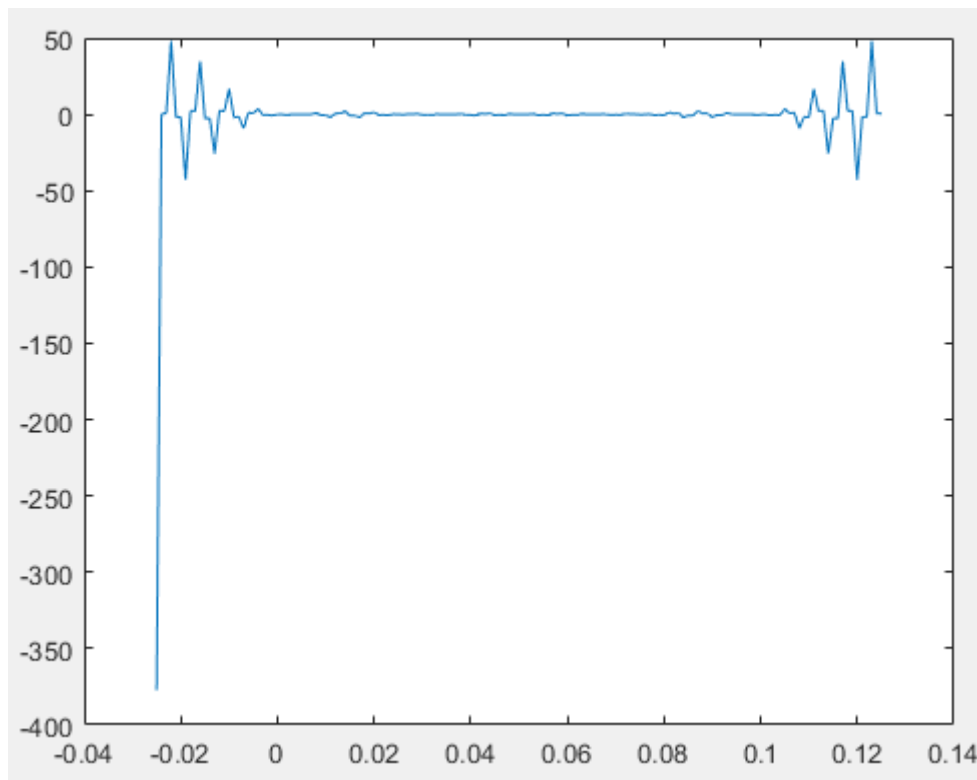


Рис. 4.9.4. Спектр прямоугольных импульсов

4.10. Функция Дирихле

Листинг 12: Код в MatLab

```

1 x = 0:0.01:15;
2 FFT1 = fft(diric(x,7));
3 FFT2 = fft(diric(x,8));
4
5 plot(x, diric(x,7));
6 grid on
7 title('n=7');
8
9 figure;
10 plot(x, diric(x,8));
11 grid on
12 title('n=8');
13
14 figure;
15 plot(x, FFT1)
16
17 figure;
18 plot(x, FFT2)

```

Программа использует встроенную функцию `diric` для создания выборки из функции Дирихле с четным и нечетным значением параметра.

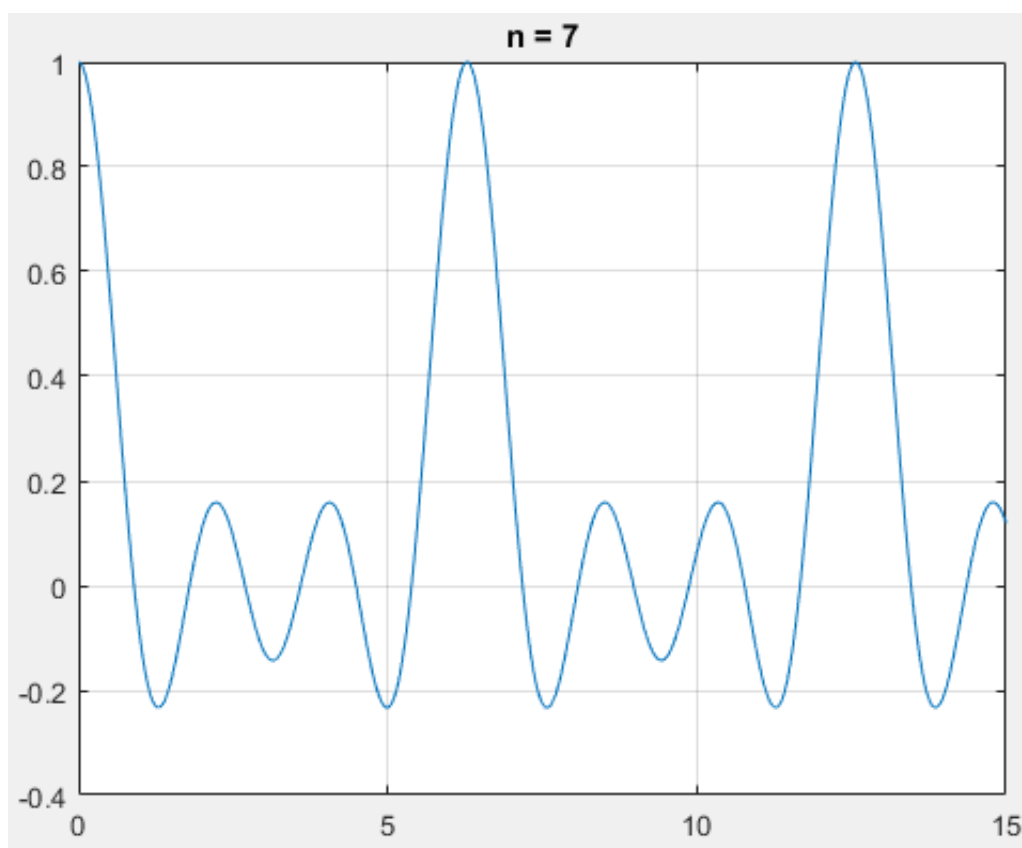


Рис. 4.10.1. Функция Дирихле с параметром равным 7

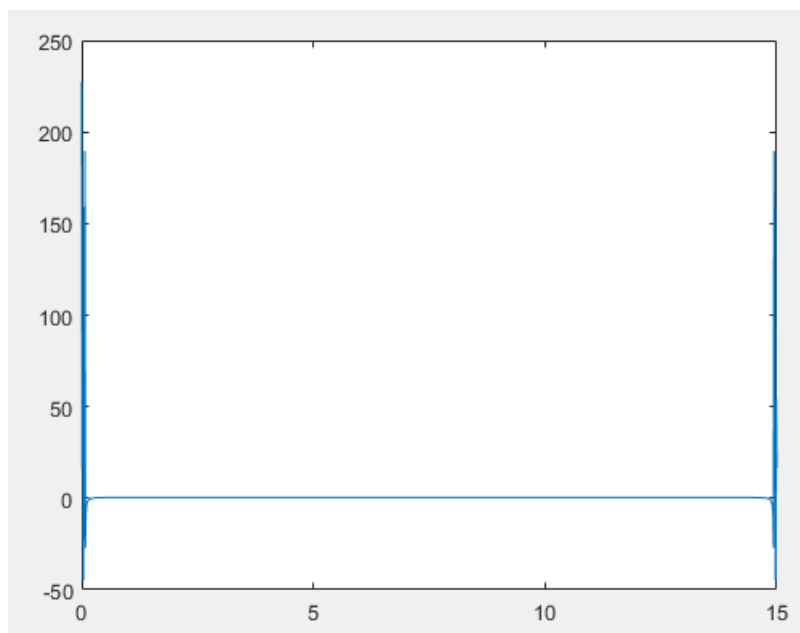


Рис. 4.10.2. Функция Дирихле с параметром равным 8

Видно, что нечетный параметр обеспечивает однонаправленные импульсы, а большее значение параметра увеличивает частоту колебаний.

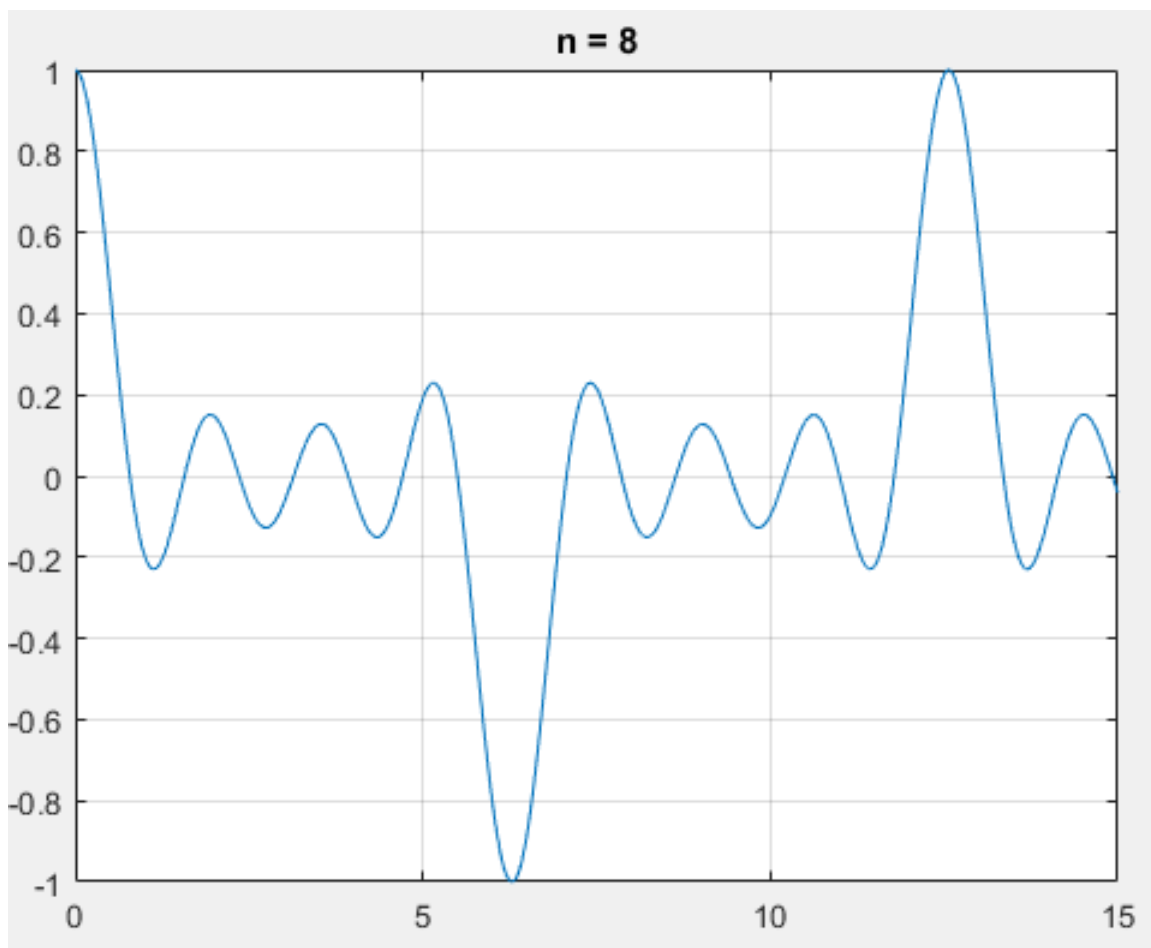


Рис. 4.10.3. Спектр функции Дирихле с параметром равным 7

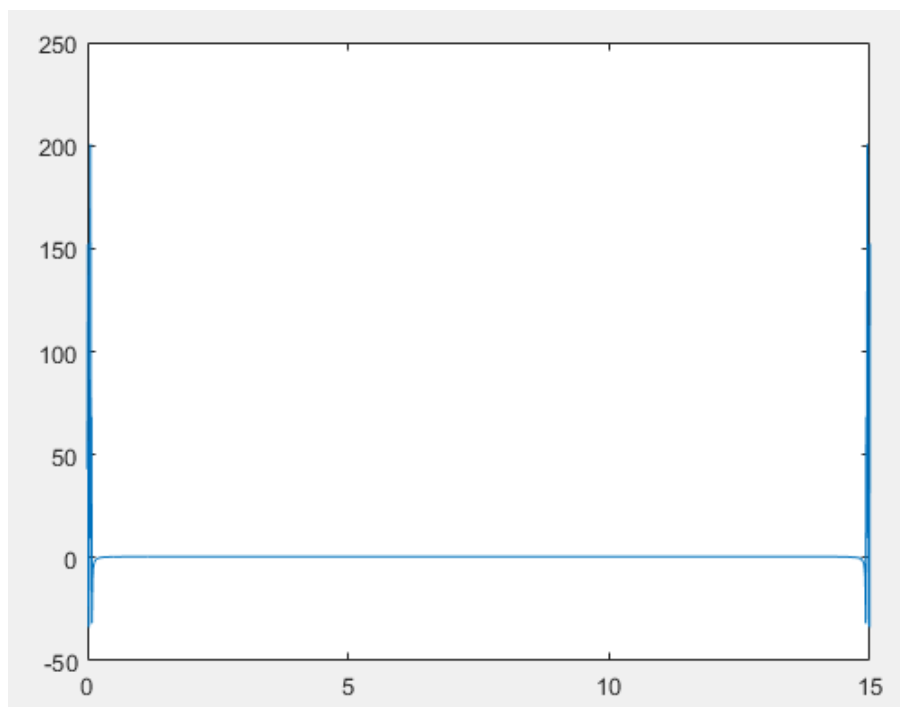


Рис. 4.10.4. Спектр функция Дирихле с параметром равным 8

4.11. Сигнал с меняющейся частотой

Листинг 13: Код в MatLab

```
1 t = 0:1/8e3:1;
2
3 s1 = chirp(t,1e3, 1,2e3, 'linear');
4 s2 = chirp(t, 1e3, 1, 2e3, 'quadratic');
5 s3 = chirp(t, 1e3, 1, 2e3, 'logarithmic');
6
7 specgram(s1, [], 8e3);
8 title('linear');
9 colormap gray
10
11 figure
12 specgram(s2, [], 8e3);
13 title('quadratic');
14 colormap gray
15
16 figure
17 specgram(s3, [], 8e3);
18 title('logarithmic');
19 colormap gray
```

Эта программа с помощью функции `chirp` генерирует колебания, мгновенная частота которых изменяется согласно выбранной функции. В данном примере рассмотрены 3 таких функции — линейная, квадратичная и логарифмическая. На экран выводятся спектрограммы этих сигналов — зависимость мгновенного амплитудного спектра от времени.

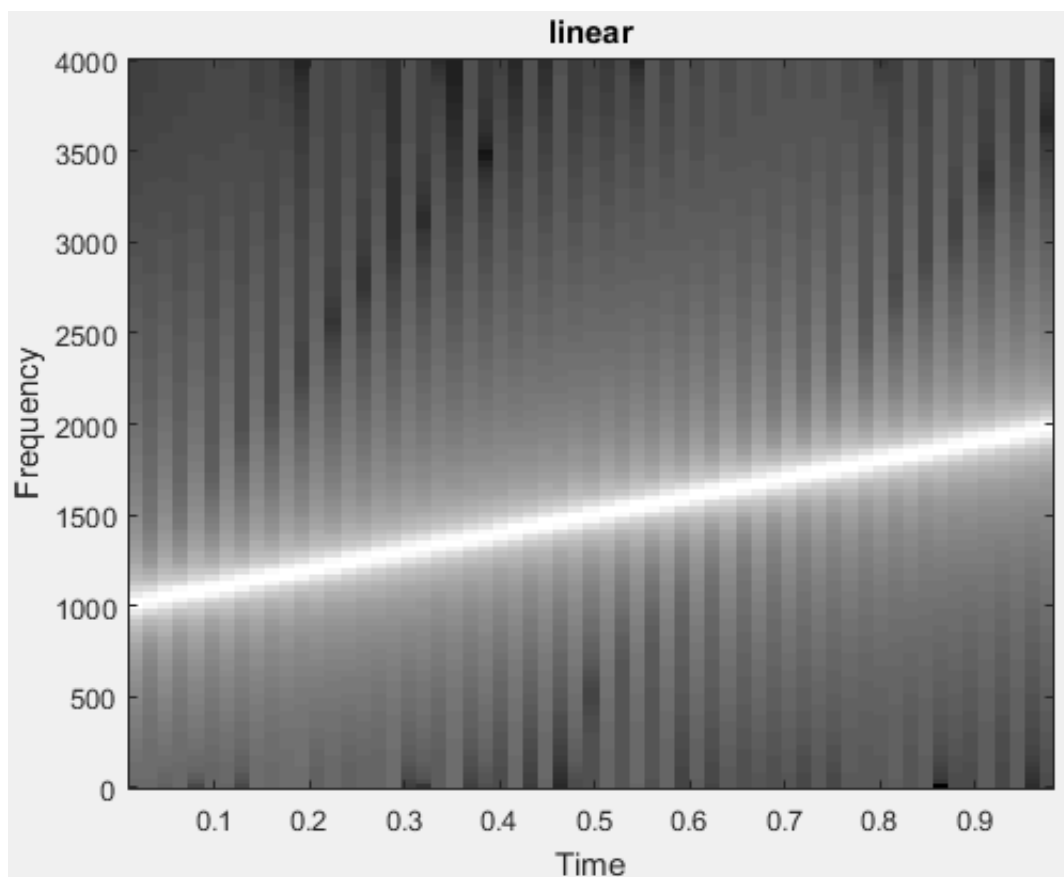


Рис. 4.11.1. Спектрограмма линейной функции `chirp`

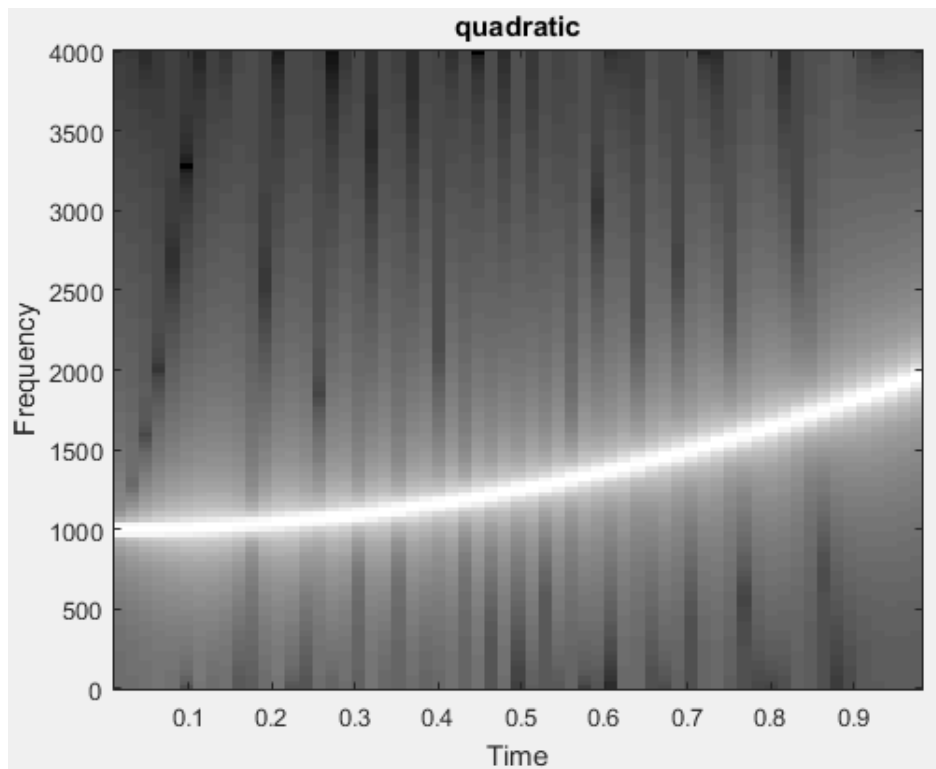


Рис. 4.11.2. Спектрограмма квадратичной функции chirp

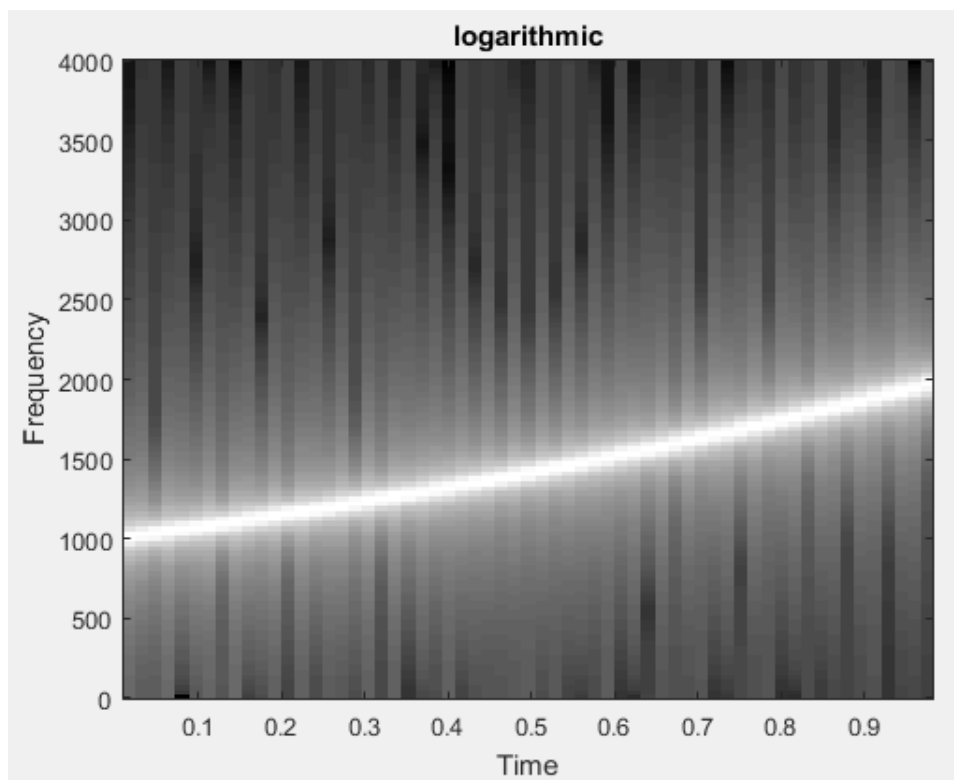


Рис. 4.11.3. Спектрограмма логарифмической функции chirp

На рисунках 4.11.1, 4.11.2 и 4.11.3 показаны спектрограммы, наглядно демонстрирующие характер изменения мгновенной частоты сигнала.

4.12. Сравнение методов корреляции

В качестве исходного примера была взята задача нахождения синхросылки 0 0 1 в сигнале 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0.

Листинг 14: Код в МатЛаб

```
1 x = [1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0];
2 y = [0 0 1];
3 y1 = [];
4
5 for i = 1:length(y)
6     if (y(i) == 1)
7         y1(i) = 1;
8     else
9         y1(i) = -1;
10    end
11 end
12
13 for i = 1:length(x)
14     if(i <= length(y))
15         y1(i) = y1(i);
16     else
17         y1(i) = 0;
18     end
19 end
20
21 R = [];
22 BR = [];
23
24 tic
25 for i = 1:length(x)
26     R(i) = sum(x .* circshift(y1, i-1, 2)) / length(x);
27 end
28 toc
29
30 tic
31 x = fft(x);
32 y1 = fft(y1);
33 x = conj(x);
34 BR = ifft(x .* y1)/length(x);
35 toc
```

Перед началом вычисления корреляции синхросылка была изменена - 0 0 1 на -1 -1 1 для улучшения качества этой посылки с целью ее более надежного нахождения в посылке, а затем она была дополнена нулями для совпадения длин двух векторов.

Далее производились 2 расчета корреляции - обычным алгоритмом и быстрым с контролем времени на каждую операцию. Оба алгоритма показали, что синхросылка была найдена в сигнале 2 раза - по смещению +3 и +5.

Первый алгоритм показал время выполнения - 0.001628 с, в то время как второй - 0.004082 с. Значит алгоритм быстрой корреляции на самом деле много быстрее стандартного.

5. Выводы

В этой работе исследованы методы генерации и визуализации различных сигналов в среде MatLab.

Рассмотрены различные виды сигналов :

- детерминированные
- периодические
- конечные
- бесконечные
- гармонические
- представляющие из себя единичные импульсы различной формы

Были быстро и легко построены спектры сигналов благодаря преобразованию Фурье.

Попробовали два метода подсчета колляции: обычный и быстрый с контролем времени на каждую операцию. Второй метод оказался быстрее.