

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:
Болдырев А.В.
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. Понятие сигналов как векторов отсчетов функций	2
3.2. Затухающие сигналы	2
3.3. Одиночные импульсы	2
3.4. Ограниченная полоса частот	3
3.5. Гауссов радиоимпульс	3
3.6. Дирихле	4
3.7. Математические законы изменения мгновенной частоты	4
3.8. Преобразование Фурье	5
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	13
4.6. Сигнал с ограниченной полосой частот	15
4.7. Гауссов радиоимпульс	16
4.8. Последовательности импульсов	18
4.9. Генерация периодических сигналов	21
4.10. Функция Дирихле	24
4.11. Сигнал с меняющейся частотой	26
5. Выводы	28

1. Цель работы

Целью данной работы является приобретение навыков генерации и визуализации простых сигналов в среде MatLab, а также разложение этих сигналов в ряд Фурье для построения спектра сигналов.

2. Постановка задачи

Задачей работы является промоделировать сигналы в командном окне MATLAB и в среде Simulink из Главы 3, сс. 150–170 справочного пособия и получить их разложение в ряд Фурье.

3. Теоретическая информация

3.1. Понятие сигналов как векторов отсчетов функций

Аналоговый сигнал, с математической точки зрения, представляет собой функцию. В среде MatLab может быть удобно представлен как вектор дискретных отсчетов этой функции. Графически это представляется в виде графика зависимости значений этого вектора от значений вектора отсчетов времени. Второй вектор удобно формировать как возрастающую последовательность чисел, шаг между которыми есть величина, равная периоду дискретизации.

Таким образом, определив вектор отсчетов времени и некоторые константы, необходимые для представления вида сигнала в математической формуле, такие как амплитуда колебаний, частота колебаний и так далее, мы можем задать вектор значений функции в известных нам моментах времени для дальнейшего построения графика. Делается это путем использования известных математических законов и встроенных в MatLab функций генерации специальных сигналов.

3.2. Затухающие сигналы

Затухание обычного гармонического сигнала получается путем его домножения на убывающую экспоненциальную функцию, как показано ниже на 3.2.1:

```
s2 = exp(-alpha*t) .* s1;
```

Рис. 3.2.1. Домножение сигнала на экспоненту в MatLab

3.3. Одиночные импульсы

Встроенная функция `rectpuls` имеет следующий вид:

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

Рис. 3.3.1. Генерация прямоугольного импульса

где y -возвращаемое значение, t -вектор значений времени, сгенерированный заранее, $width$ -ширин (длительность) импульса.

Встроенная функция `tripuls` имеет следующий вид:

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

Рис. 3.3.2. Генерация треугольного импульса

где параметр `skew` - коэффициент асимметрии импульса (по-умолчанию равен 0), а другие параметры имеют те же значения.

3.4. Ограниченная полоса частот

Для формирования сигнала, имеющего ограниченный спектр, используется функция `sinc`:

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

Рис. 3.4.1. Sinc

Спектр сигнала в этом случае имеет прямоугольный вид:

$$Y(\omega) = \begin{cases} 1, & |\omega| < \pi, \\ 0, & |\omega| > \pi. \end{cases}$$

Рис. 3.4.2. Спектр Sinc

3.5. Гауссов радиоимпульс

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

$$y = \exp(-a t^2) \cos(2\pi f_c t).$$

Рис. 3.5.1. Гауссов радиоимпульс

А спектр такого сигнала можно получить путем преобразования Фурье, формула которого представлена на 3.5.2 ниже:

$$\dot{S}(\omega) = \frac{1}{2} \sqrt{\frac{\pi}{a}} \left(\exp \left(-\frac{(\omega + 2\pi f_c)^2}{4a} \right) + \exp \left(-\frac{(\omega - 2\pi f_c)^2}{4a} \right) \right).$$

Рис. 3.5.2. Формула для разложения радиоимпульса

3.6. Дирихле

Функция Дирихле описывается формулой:

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

Рис. 3.6.1. Функция Дирихле

где n - целое положительное число.

Функцию Дирихле еще называют периодический sinc функцией. При нечетном / четном значении параметра n функция приобретает вид:

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

Рис. 3.6.2. Функция Дирихле при нечетном n

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

Рис. 3.6.3. Функция Дирихле при четном n

3.7. Математические законы изменения мгновенной частоты

В данной работе рассматриваются 3 закона - линейный, квадратичный и логарифмический. Формулы этих законов представлены на ?? - ??:

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

Рис. 3.7.1. Линейный закон

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

Рис. 3.7.2. Квадратичный закон

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

Рис. 3.7.3. Логарифмический закон

Стоит отметить, что логарифмический закон противоречит своему названию, т.к. зависимость частоты от времени в нем экспоненциальная, а не логарифмическая.

3.8. Преобразование Фурье

Для нахождения спектра сигнала чаще всего применяют разложение функции в ряд Фурье, или же преобразование Фурье. Формула прямого преобразования Фурье (нормированный логарифмический закон) выглядит следующим образом:

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

Рис. 3.8.1. Преобразование Фурье

4. Ход работы

4.1. Генерация затухающего гармонического сигнала

Листинг 1: Код в MatLab

```

1 Fs = 8e3;
2 t = 0:1/Fs:1;
3 t = t';
4 A = 2;
5 f0 = 1e3;
6 phi = pi/4;
7 s1 = A*cos(2*pi*f0*t + phi);
8 alpha = 1e3;
9 s2 = exp(-alpha*t) .* s1;
10 FFT = fft(s2);
11 subplot(2,2,1); plot(t(1:100), s2(1:100))
12 subplot(2,2,2); plot(t(1:100), s2(1:100), 'r')
13 subplot(2,2,3); stem(t(1:100), s2(1:100))
14 subplot(2,2,4); stairs(t(1:100), s2(1:100))
15 figure;
16 plot(t(1:8001), FFT(1:8001))

```

Здесь представлен код программы, генерирующей затухающий сигнал и выводящий на экран 4 различных графика этого сигнала.

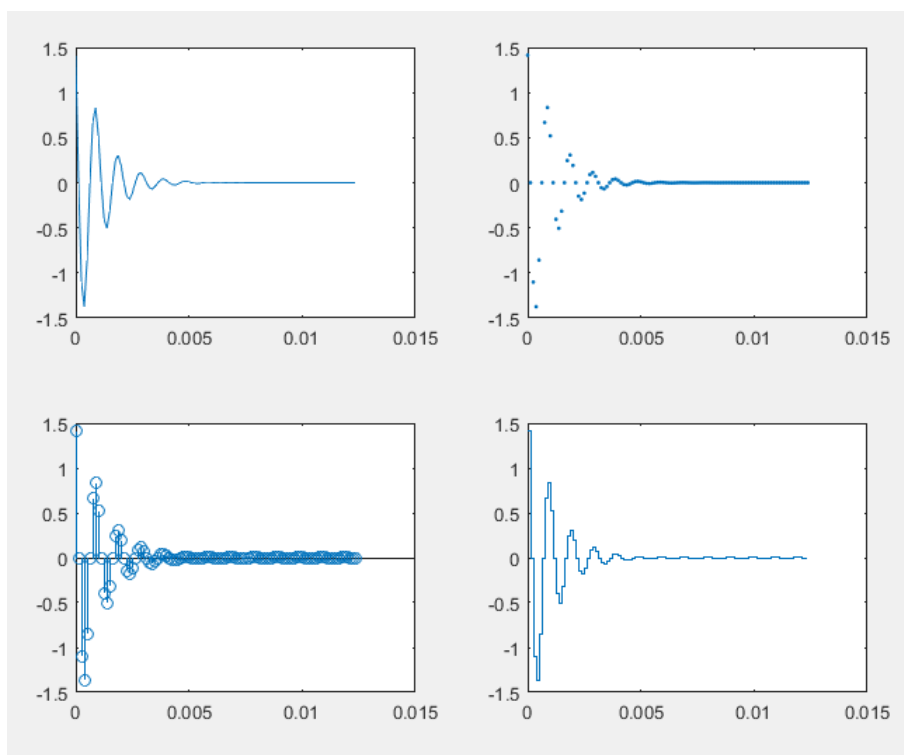


Рис. 4.1.1. Графики сигнала

На первом графике виден обычный вид затухающего гармонического сигнала, построенный средой MatLab по дискретным отсчетам. Второй график представляет из себя точки того же сигнала, соответствующие дискретным отсчетам. Третий график (stem) представляет собой те же точки, но в виде «лепестков» - как некоторые значения, отклоненные от нулевого. Четвертый график (stairs) — ступенчатый график.

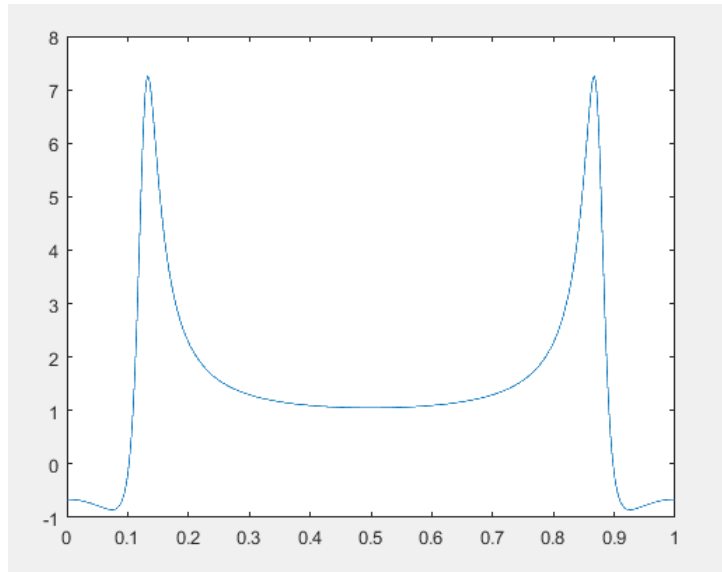


Рис. 4.1.2. Спектр сигнала

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

4.2. Многоканальный сигнал

Листинг 2: Код в MatLab

```

1 Fs = 8e3;
2 t = 0:1/Fs:1;
3 t = t';
4 f=[60_120_140];
5 s3=cos(2*pi*t*f);
6 FFT=fft(s3);
7 plot(t(1:200),s3(1:200,:))
8 figure;
9 plot(t(1:8001),FFT(1:8001,:))

```

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

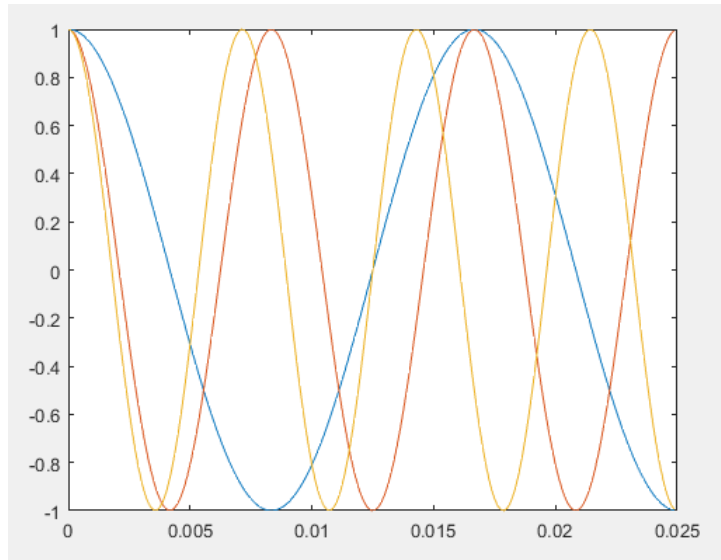


Рис. 4.2.1. График сигналов

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

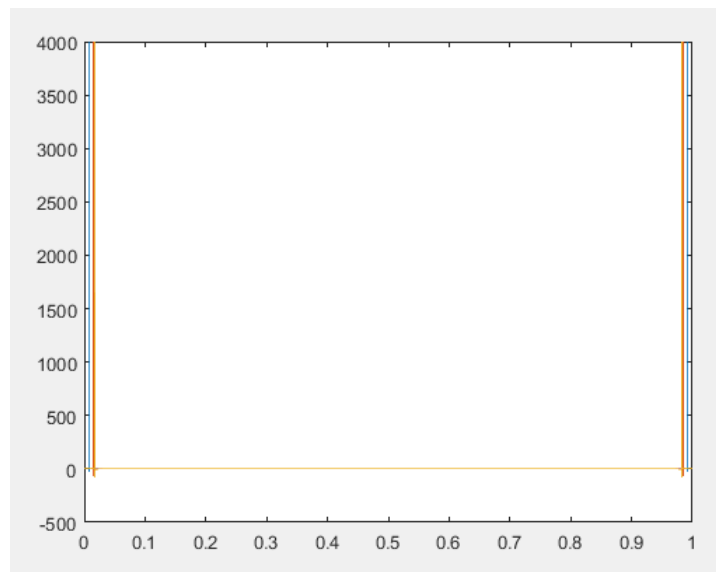


Рис. 4.2.2. Спектр сигналов

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 MatLab

```
1 Fs = 8e3;
2 t = 0:1/Fs:1;
3 t = t';
```

```

4 A=2;
5 f0=1e3;
6 phi=pi/4;
7 alpha=1e3;
8 T=0.01;
9 s1=A*exp(-alpha*t).* (t >= 0);
10 FFT1=fft(s1);
11 s2=A*(abs(t) <= T/2);
12 FFT2=fft(s2);
13 s3=A*t./T.*(T >= 0).* (t <= T);
14 FFT3=fft(s3);
15 figure(1);
16 plot(t(1:100),s1(1:100))
17 figure(2);
18 plot(t(1:100),s2(1:100))
19 figure(3);
20 plot(t(1:100),s3(1:100))
21 figure(4);
22 plot(t(1:8001),FFT1(1:8001))
23 figure(5);
24 plot(t(1:8001),FFT2(1:8001))
25 figure(6);
26 plot(t(1:8001),FFT3(1:8001))

```

Данный код генерирует и выводит на экран односторонний экспоненциальный импульс, прямоугольный импульс и несимметричный треугольный импульс согласно заданным параметрам.

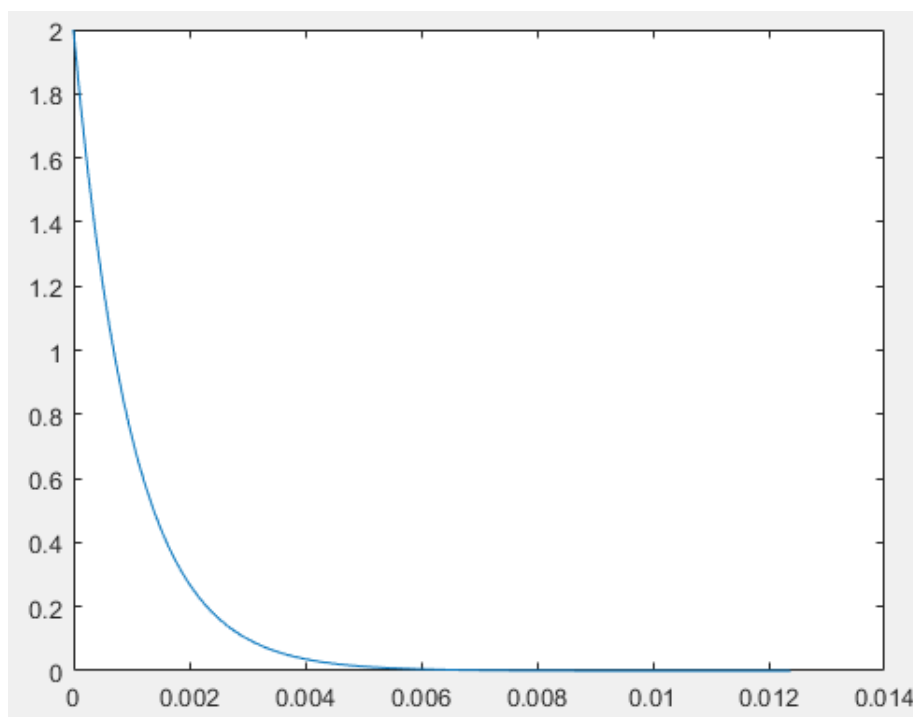


Рис. 4.3.1. Экспоненциальный импульс

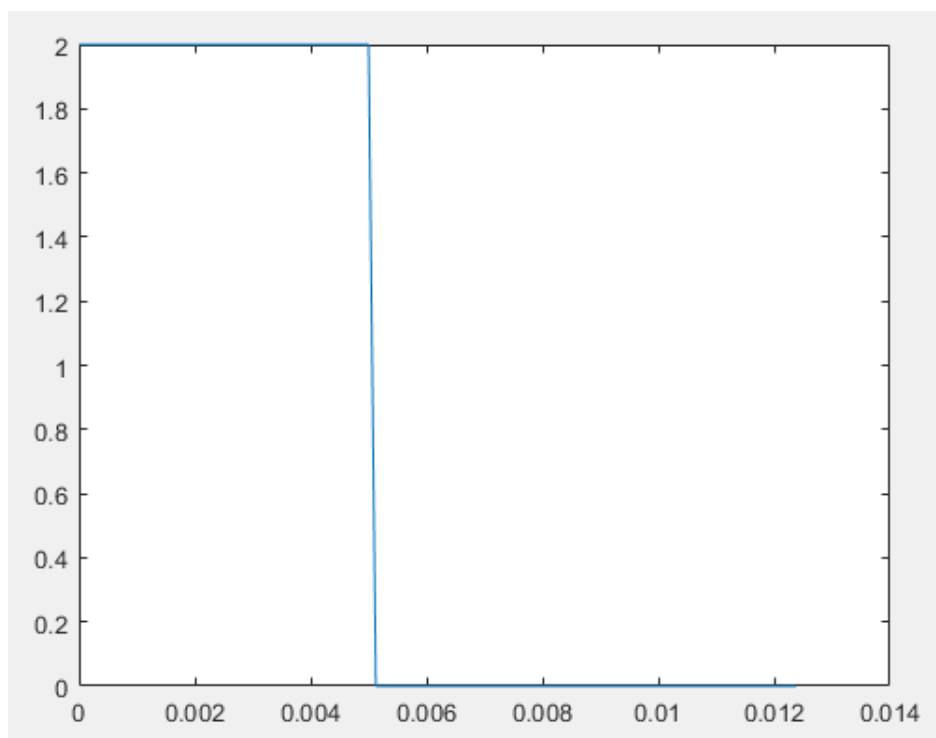


Рис. 4.3.2. Прямоугольный импульс

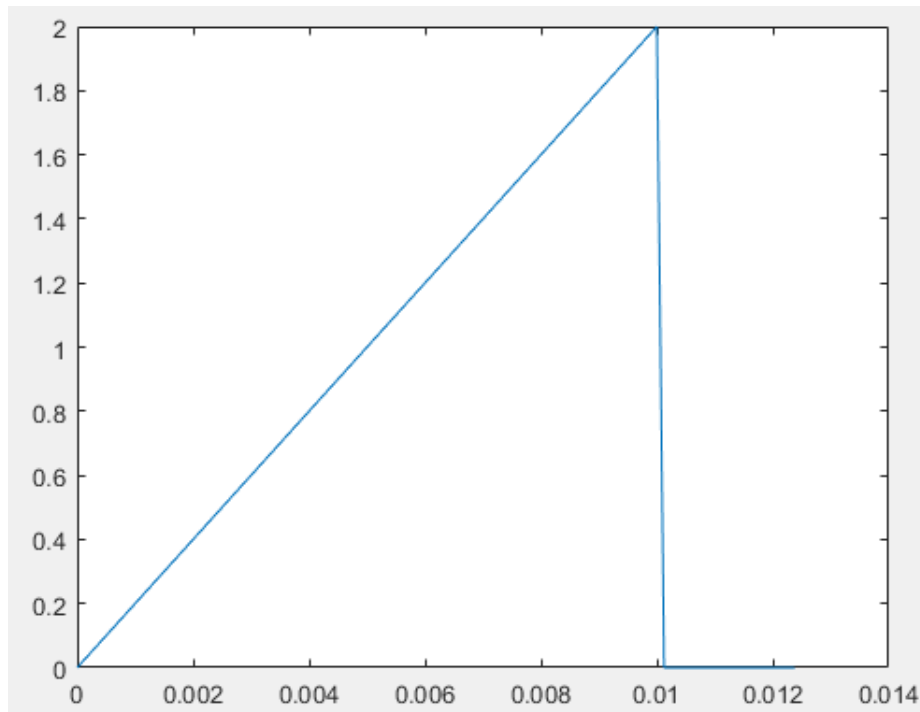


Рис. 4.3.3. Несимметричный треугольный импульс

На рисунках 4.3.1 — 4.3.3 представлены графики сгенерированных сигналов, выведенных с помощью стандартной функции построения графиков в MatLab.

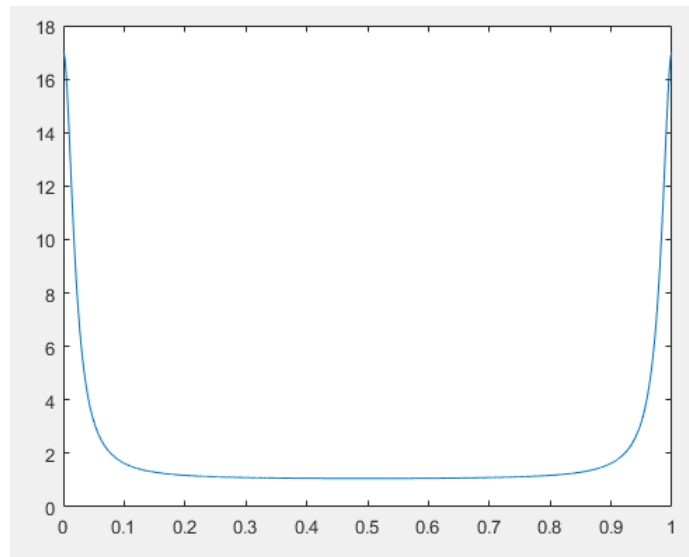


Рис. 4.3.4. Спектр экспоненциального импульса

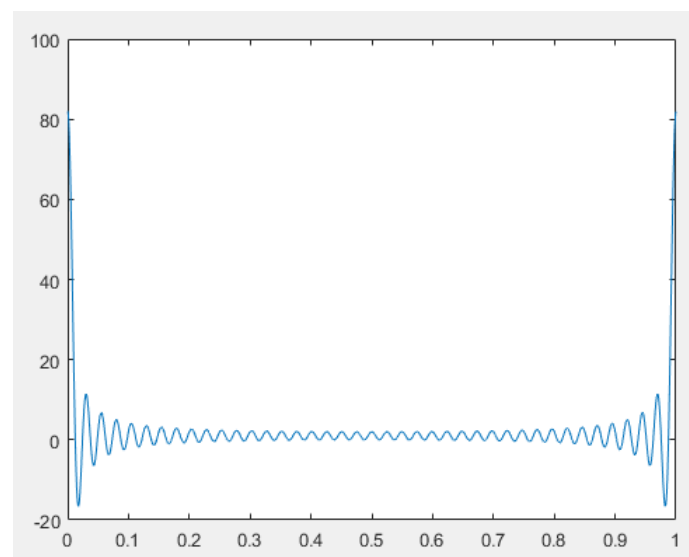


Рис. 4.3.5. Спектр прямоугольного импульса

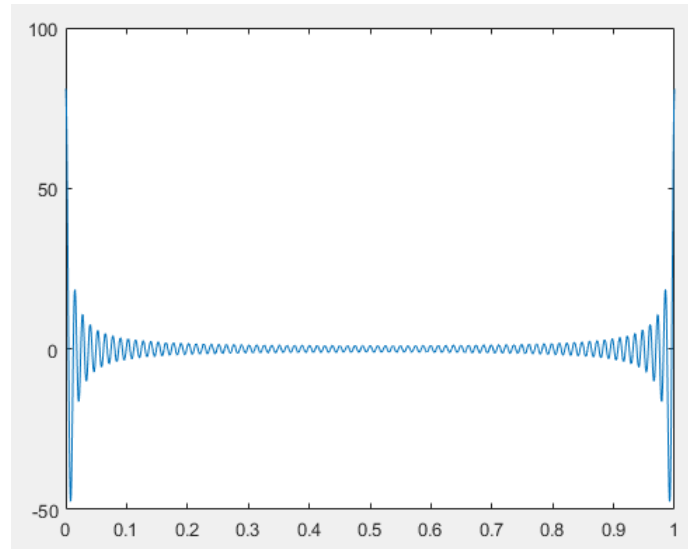


Рис. 4.3.6. спектр несимметричного треугольного импульса

На рисунках 4.3.4 — 4.3.6 представлены спектры сигналов 4.3.1 — 4.3.3.

4.4. Прямоугольный импульс

Листинг 4: Код в MatLab

```

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

```

Данный сигнал представлен конкатенацией двух разнополярных прямоугольных импульсов, с использованием встроенных функций.

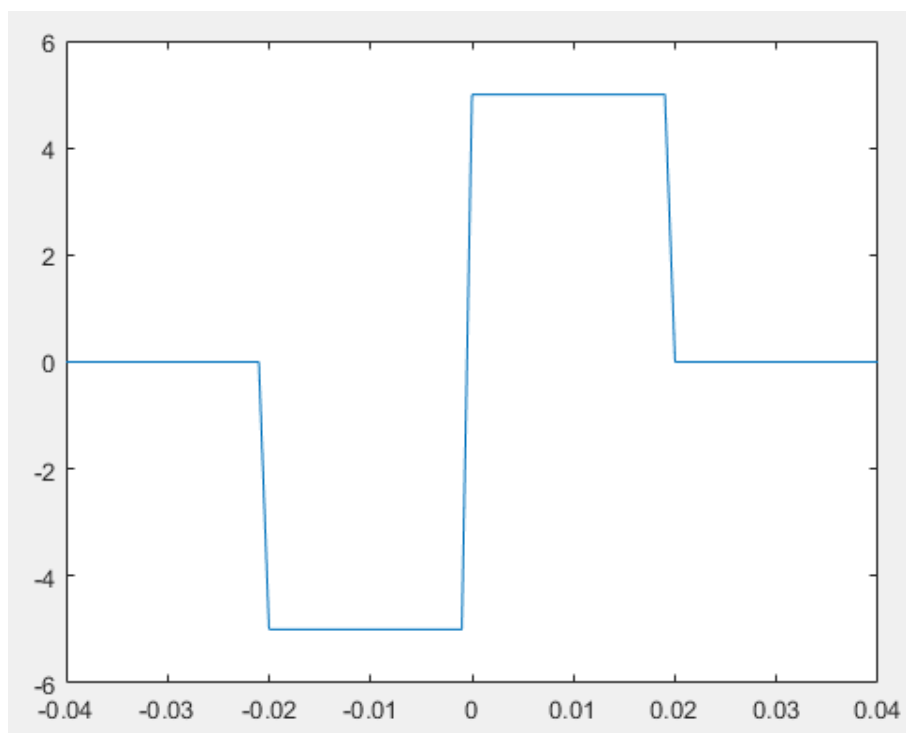


Рис. 4.4.1. Прямоугольные импульсы

На данном рисунке представлен график прямоугольных импульсов.

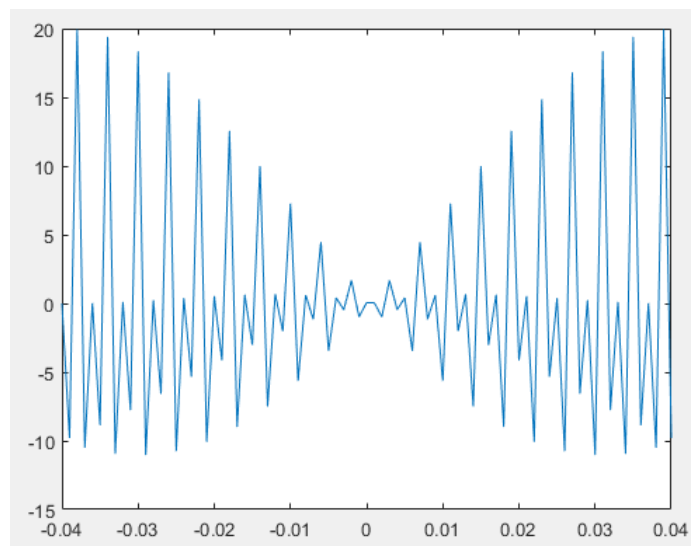


Рис. 4.4.2. Спектр прямоугольных импульсов

Спектр прямоугольного импульса получен с помощью преобразования Фурье.

4.5. Трапецевидный импульс

Листинг 5: Код в MatLab

```

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

```

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

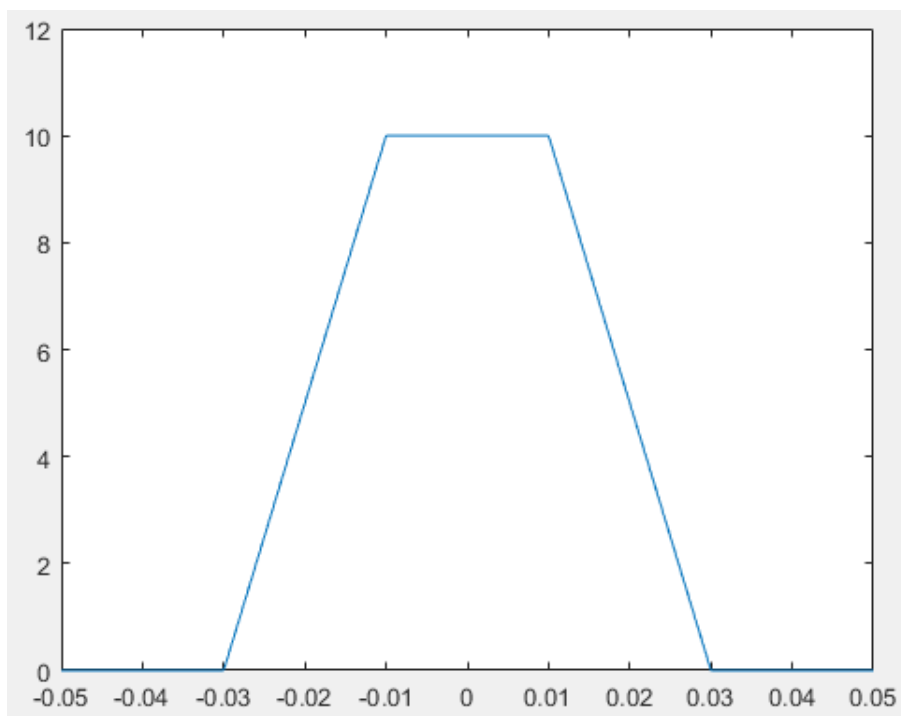


Рис. 4.5.1. Трапецевидный импульс

На данном рисунке представлен вид трапецевидного импульса в среде MatLab.

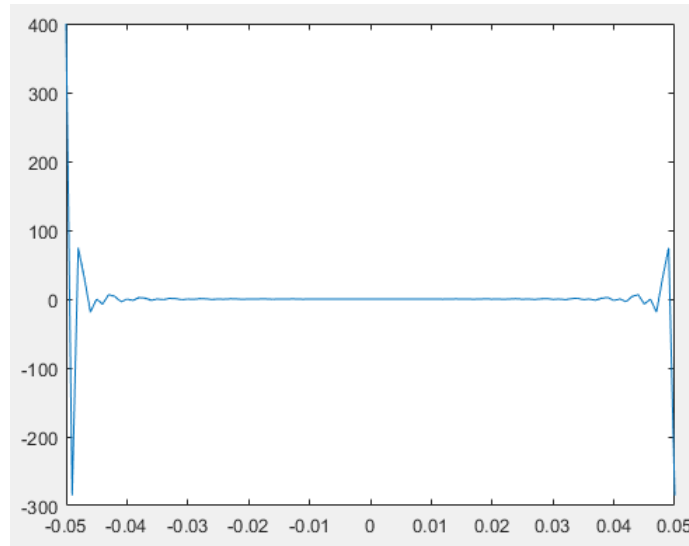


Рис. 4.5.2. Спектр трапецевидного импульса

На рисунке представлен спектр трапецевидного импульса.

4.6. Сигнал с ограниченной полосой частот

Листинг 6: Код в MatLab

```

1 Fs = 1e3;
2 t = -0.1:1/Fs:0.1;
3 f0 = 10;
4 T = 1/f0;
5 s = rectpuls(t,T) .* cos(2*pi*f0*t);
6 f = -100:100;
7 sp = T/2 * (sinc((f-f0)*T) + sinc((f+f0)*T));
8 plot(t,s)
9 ylim([-1.1 1.1]);
10 figure;
11 plot(f, abs(sp))

```

Данный код генерирует сигнал, у которого спектр ограничен по частоте. Затем выводится и сам спектр данного сигнала:

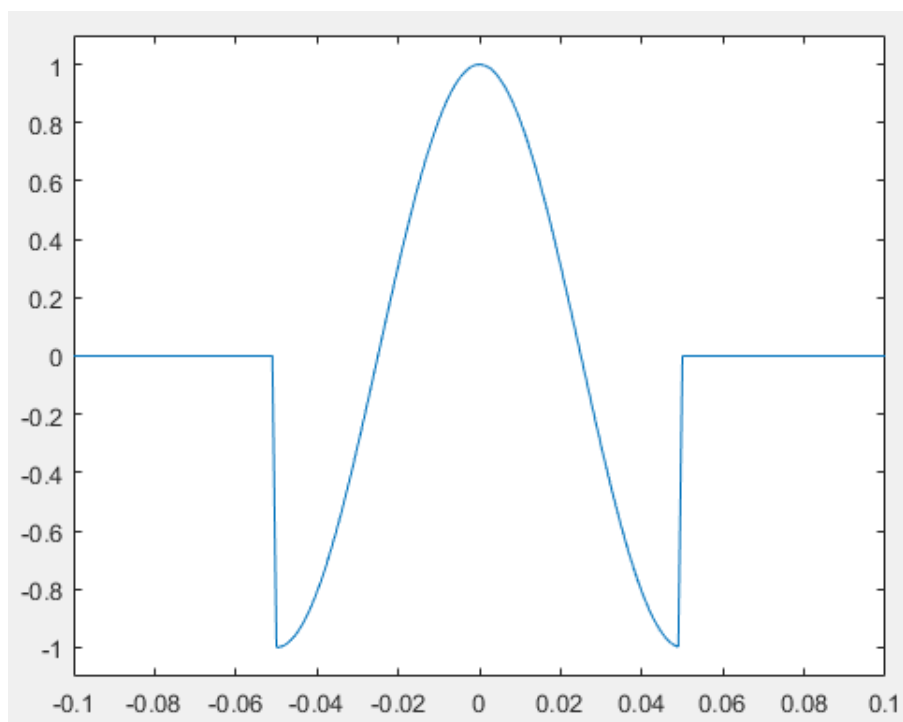


Рис. 4.6.1. Сигнал с ограниченным спектром

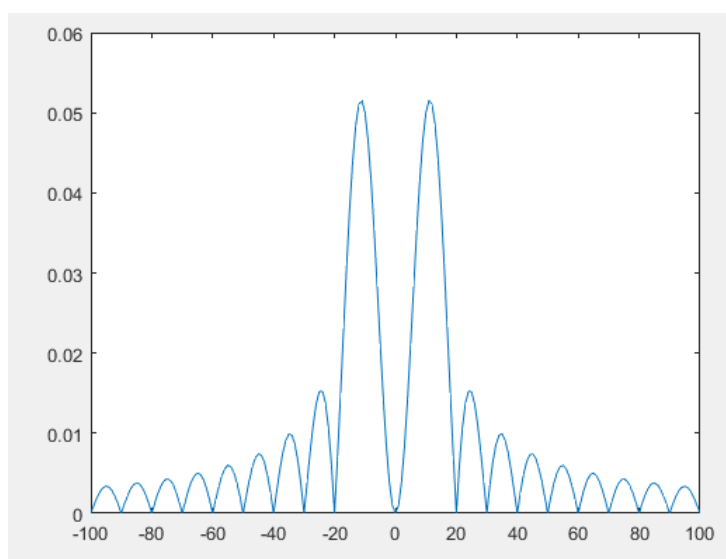


Рис. 4.6.2. Ограниченный спектр ограниченного сигнала

Спектр сигнала получен с помощью функции `sinc`.

4.7. Гауссов радиоимпульс

Листинг 7: Код в MatLab

```
1 Fs = 16e3;
2 t = -10e-3:1/Fs:10e-3;
```

```

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

```

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

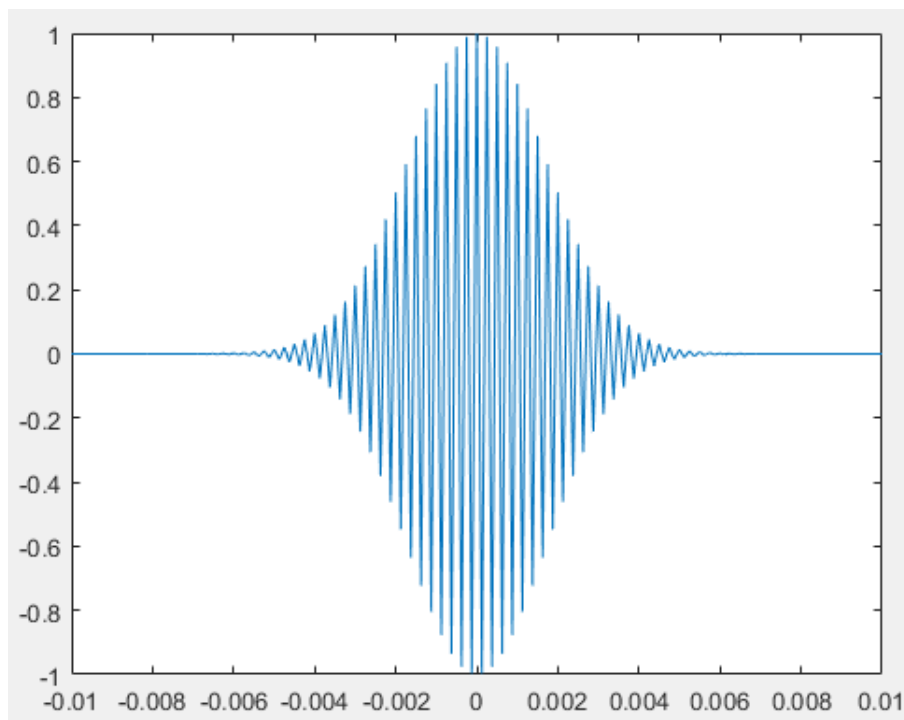


Рис. 4.7.1. Гауссов радиоимпульс

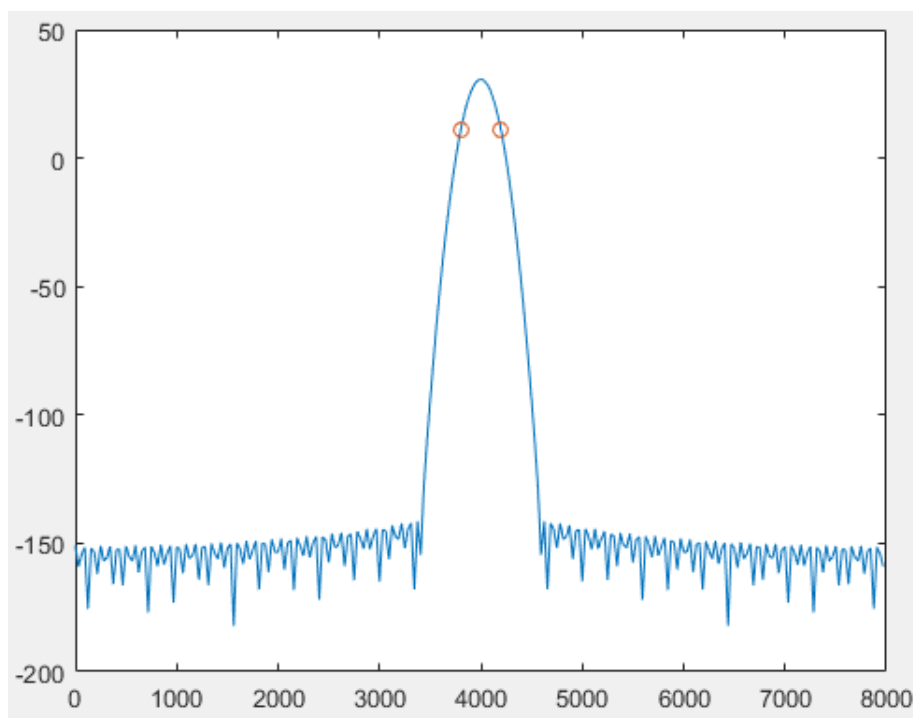


Рис. 4.7.2. Амплитудный спектр радиоимпульса

На графике спектра также отмечены расчетные границы этого спектра.

4.8. Последовательности импульсов

Листинг 8: Код в MatLab

```

1 Fs = 1e3;
2 t = 0:1/Fs:0.5;
3 tau = 20e-3;
4 d = [20 80 160 260 380]'*1e-3;
5 d(:,2) = 0.8.^(0:4)';
6 y = pulstran(t, d, 'tripuls', tau);
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.

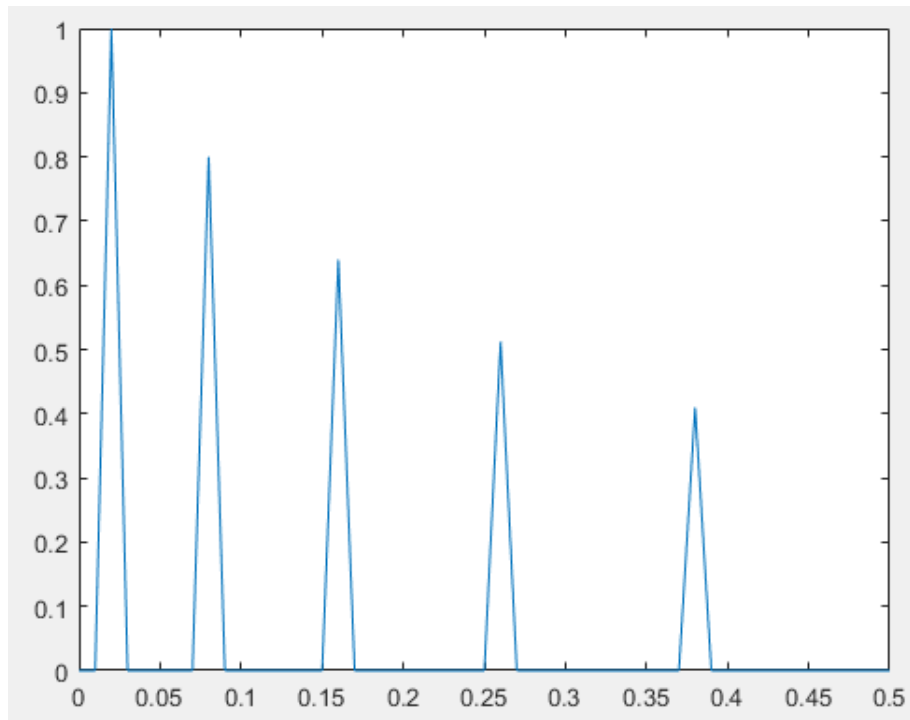


Рис. 4.8.1. Треугольные импульсы

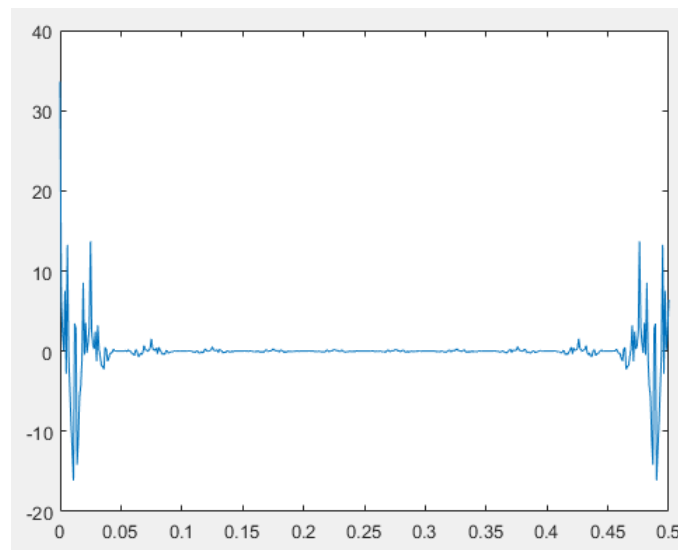


Рис. 4.8.2. Спектр импульсов

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 Fs0 = 400;
2 tau = 60e-3;
3 t0 = 0:1/Fs0:tau;
4 s0 = sin(pi*t0/tau).^2;
5 Fs = 1e3;
```

```

6| t = 0:1/Fs:0.5;
7| d = (1:6)'*_64e-3;
8| d(:,2) = 0.6.^(0:5)';
9| y = pulstran(t, d, s0, Fs0);
10| plot(t,y)
11| FFT = fft(y);
12| figure;
13| plot(t(1:501), FFT(1:501))

```

Данный код генерирует и выводит гармонические импульсы.

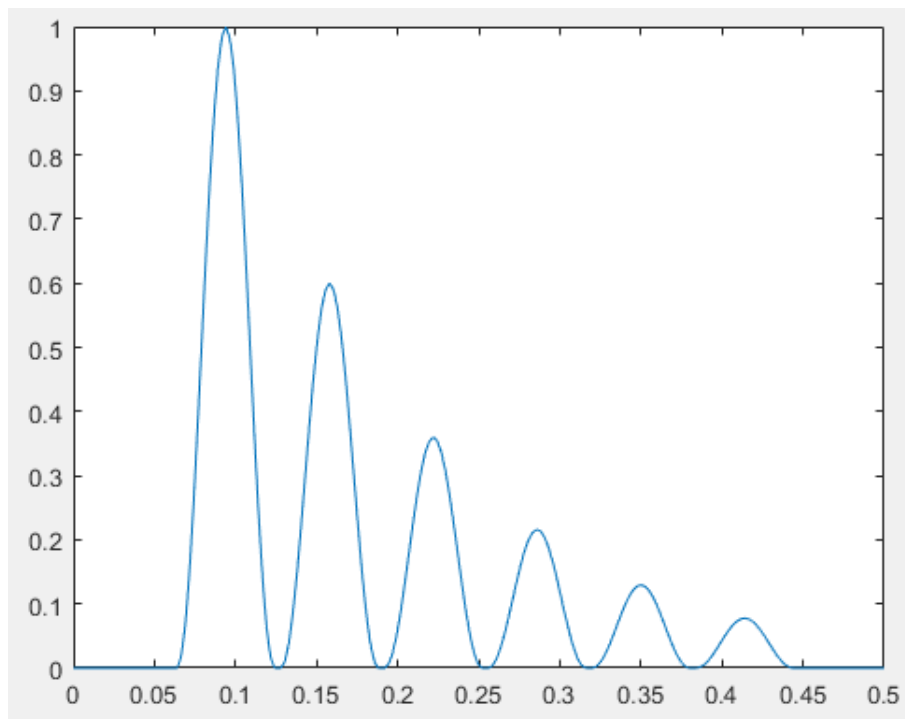


Рис. 4.8.3. Гармонические импульсы

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

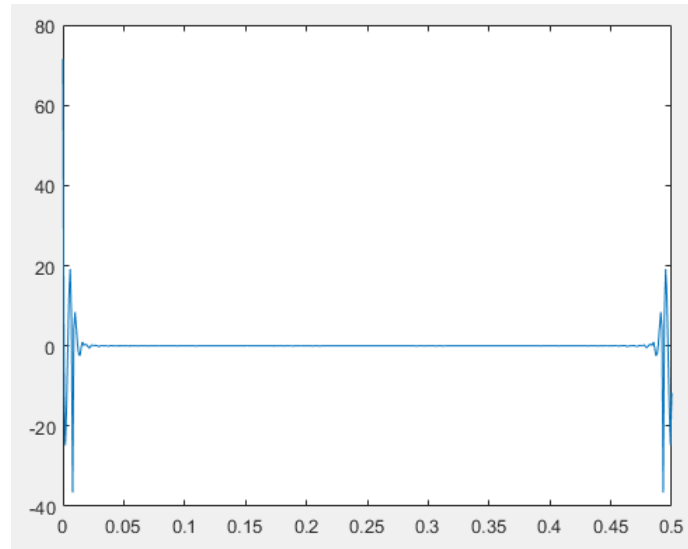


Рис. 4.8.4. Спектр гармонических импульсов

На рисунке представлен спектр гармонических импульсов.

4.9. Генерация периодических сигналов

Листинг 10: Код в MatLab

```

1 Fs = 1e3;
2 t = -10e-3:1/Fs:50e-3;
3 A = 3;
4 f0 = 50;
5 tau = 5e-3;
6 s = (square(2*pi*t*f0, f0*tau*100) + 1) * A/2;
7 plot(t, s)
8 ylim([0 5]);
9 FFT = fft(s);
10 figure;
11 plot(t(1:61), FFT(1:61))

```

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

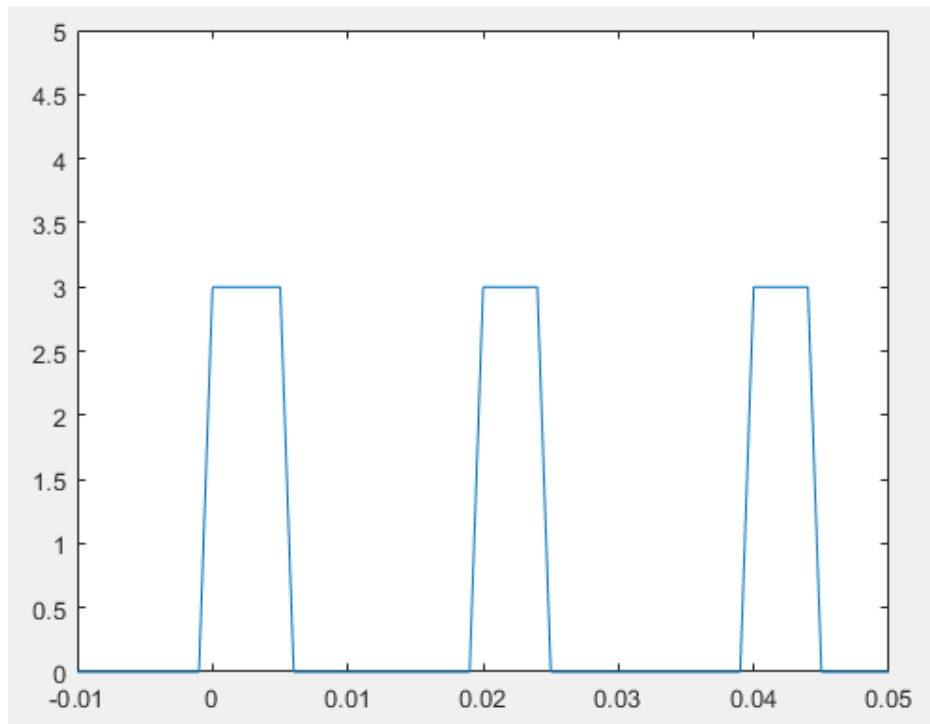


Рис. 4.9.1. Периодические прямоугольные импульсы

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

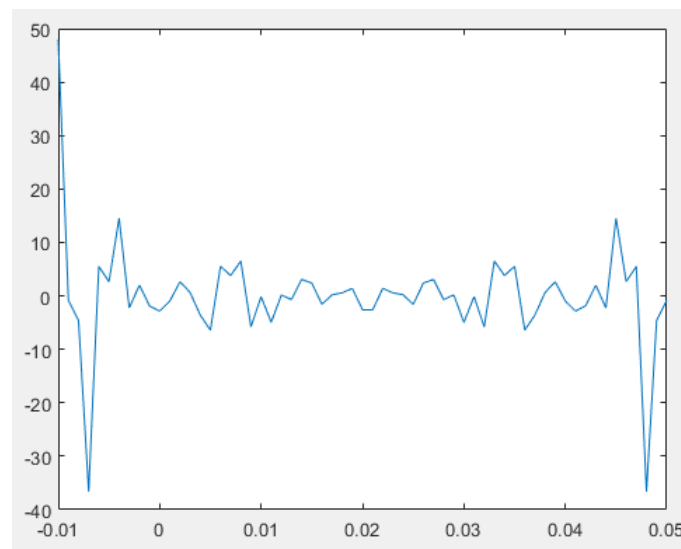


Рис. 4.9.2. Спектр прямоугольных импульсов

Листинг 11: Код в MatLab

```
1 Fs = 1e3;
2 t = -25e-3:1/Fs:125e-3;
3 A = 5;
4 T = 50e-3;
```

```

5 | t1 = 5e-3;
6 | s = (sawtooth(2*pi*t/T, 1-t1/T) - 1) * A/2;
7 | plot(t,s);
8 | FFT = fft(s);
9 | figure;
10 | plot(t, FFT)

```

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

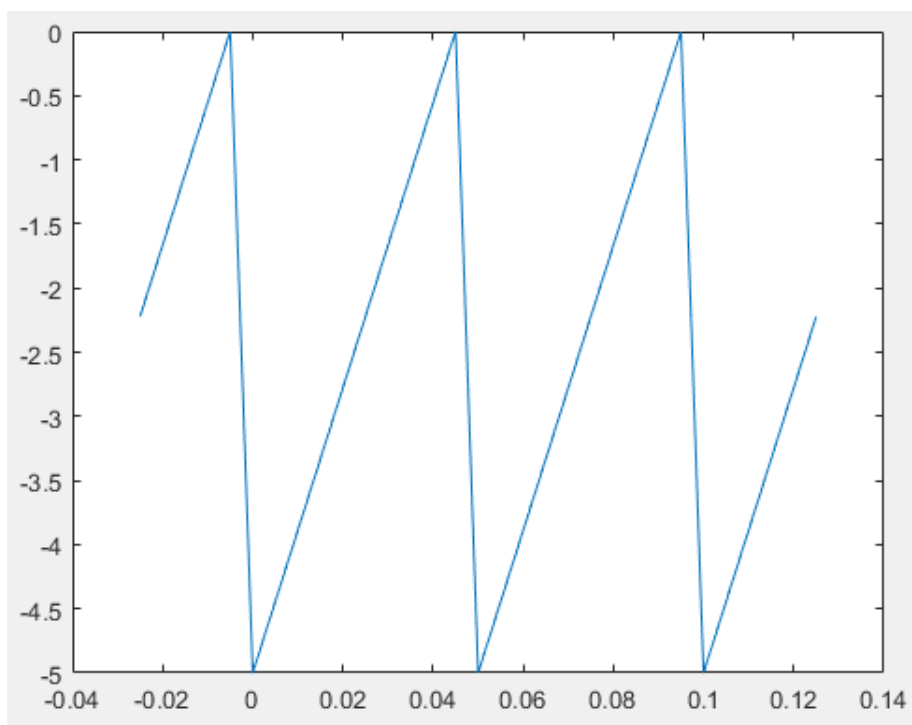


Рис. 4.9.3. Треугольные импульсы `sawtooth`

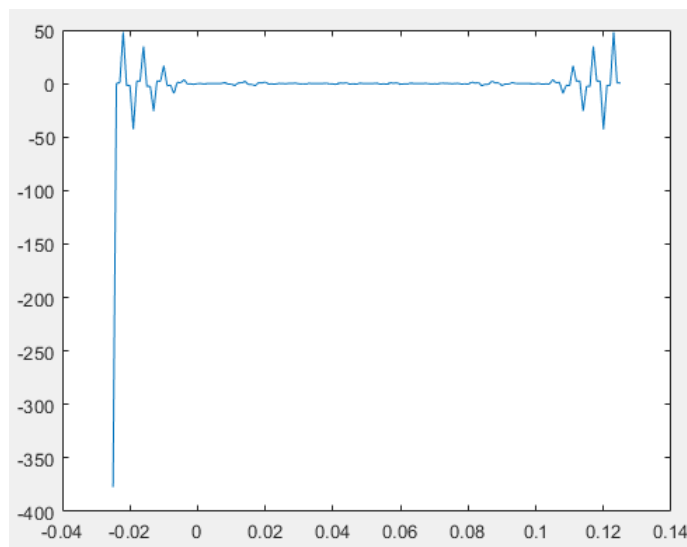


Рис. 4.9.4. Спектр прямоугольных импульсов

4.10. Функция Дирихле

Листинг 12: Код в MatLab

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

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

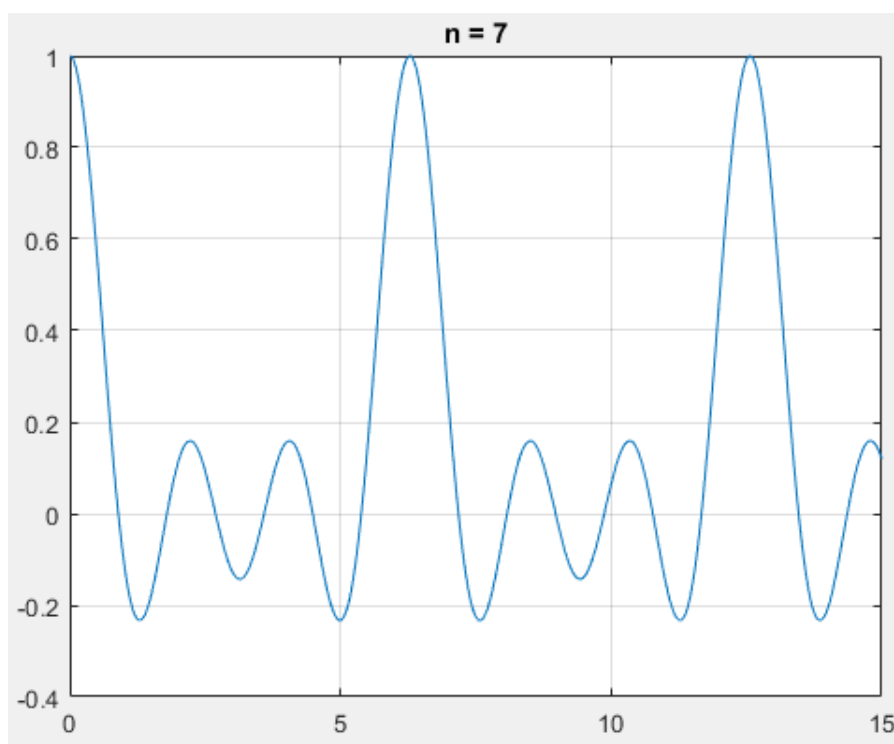


Рис. 4.10.1. Функция Дирихле с параметром равным 7

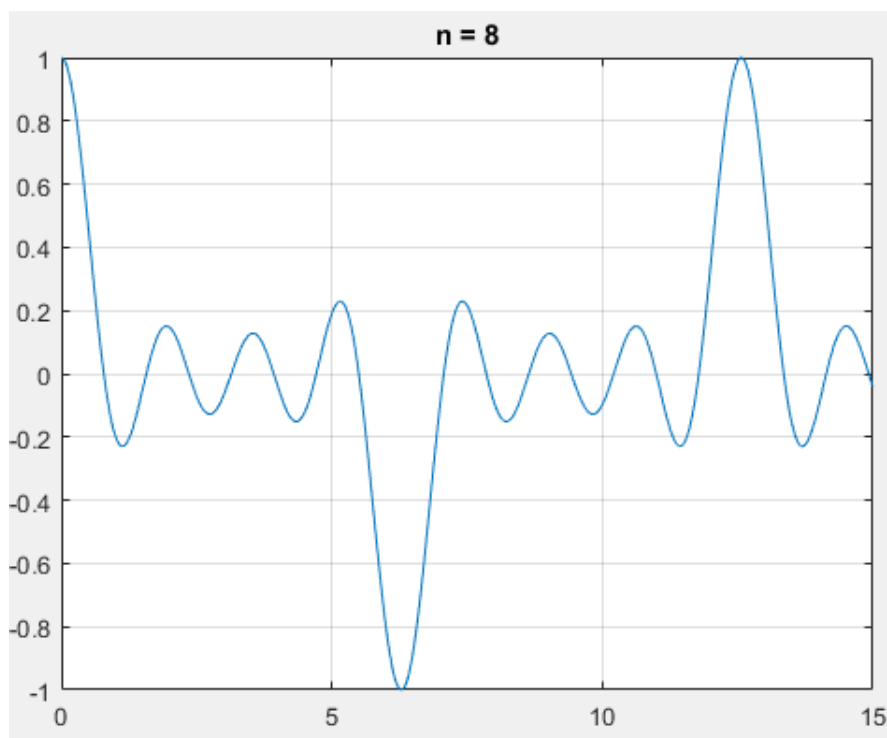


Рис. 4.10.2. Функция Дирихле с параметром равным 8

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

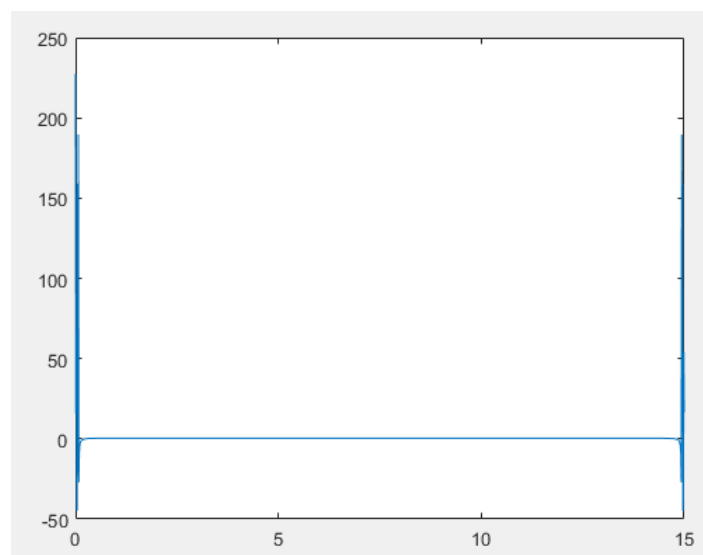


Рис. 4.10.3. Спектр функции Дирихле с параметром равным 7

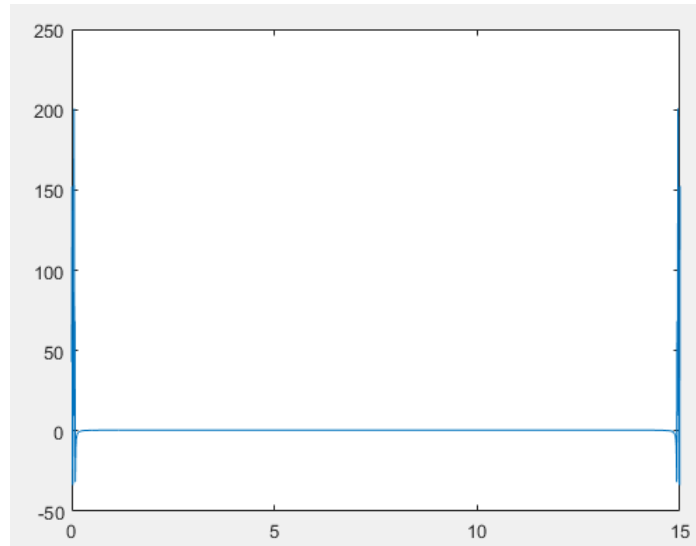


Рис. 4.10.4. Спектр функция Дирихле с параметром равным 8

4.11. Сигнал с меняющейся частотой

Листинг 13: Код в MatLab

```

1 Fs = 8e3;
2 t = 0:1/Fs:3;
3 f0 = 1e3;
4 t1 = 1;
5 f1 = 2e3;
6 s1 = chirp(t,f0, t1,f1, 'linear');
7 s2 = chirp(t, f0, t1, f1, 'quadratic');
8 s3 = chirp(t, f0, t1, f1, 'logarithmic');
9 specgram(s1, [], Fs);
10 title('linear');
11 colormap gray
12 figure
13 specgram(s2, [], Fs);
14 title('quadratic');
15 colormap gray
16 figure
17 specgram(s3, [], Fs);
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.

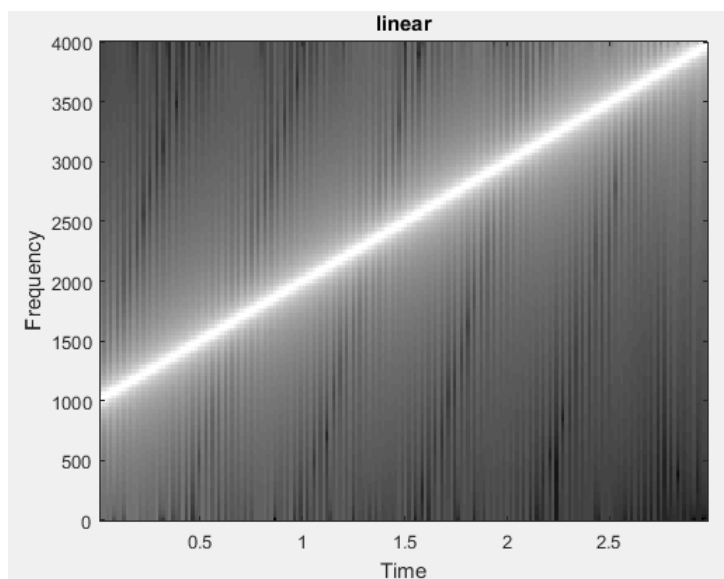


Рис. 4.11.1. Спектрограмма линейной функции chirp

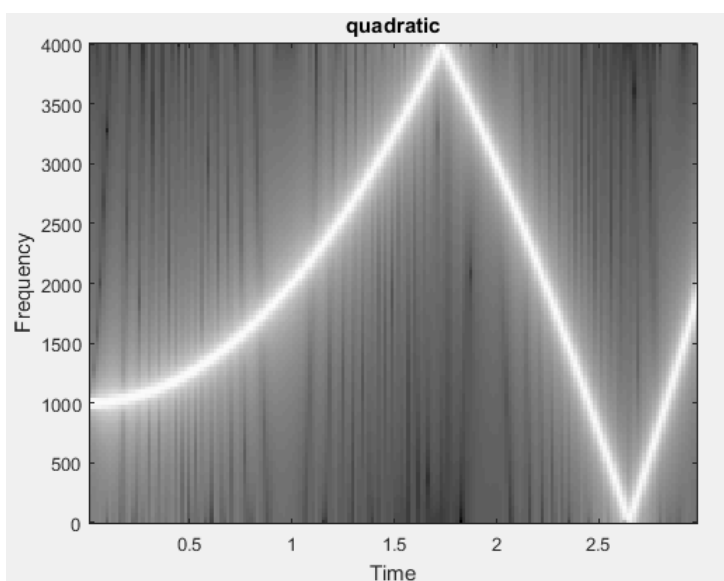


Рис. 4.11.2. Спектрограмма квадратичной функции chirp

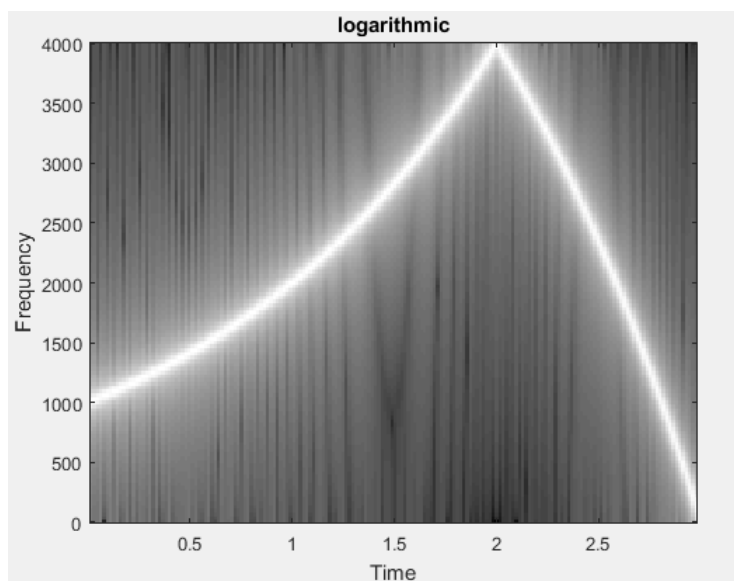


Рис. 4.11.3. Спектрограмма логарифмической функции chirp

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

5. Выводы

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

Рассмотрены различные виды сигналов - детерминированные сигналы, периодические колебания и сигналы, полученные на их основе, сигналы, представляющие из себя единичные импульсы различной формы, конечные и бесконечные сигналы.

Получены и построены спектры сигналов с помощью преобразования Фурье, имеющего реализацию в среде MatLab.

В качестве применений преобразования Фурье можно отметить следующие - фильтрация сигнала от шумов и помех, что представляется возможным, исходя из наличия паразитных гармоник в спектре; определение наличия полезного сигнала в случайном шуме; возможность иметь представление о присутствующих в сигнале частотах, равно как и определение факта аperiodичности, конечности, дискретности искомого сигнала по его спектру.