

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:**

Городничева Л.В.

Группа: 33501/3

**Преподаватель:**

Богач Н.В.

Санкт-Петербург  
2017

# Содержание

<b>1. Цель работы</b>	<b>2</b>
<b>2. Постановка задачи</b>	<b>2</b>
<b>3. Теоретическая информация об основах анализа сигнала, о ряде Фурье, преобразовании Фурье и корреляции</b>	<b>2</b>
<b>4. Ход выполнения работы</b>	<b>3</b>
4.1. Расчет временных функций . . . . .	3
4.2. Многоканальный сигнал . . . . .	5
4.3. Кусочные зависимости . . . . .	5
4.4. Функции генерации одиночных импульсов. Прямоугольный импульс . . . . .	7
4.5. Функции генерации одиночных импульсов. Треугольный импульс . . . . .	8
4.6. Функции генерации одиночных импульсов. Импульс с ограниченной полосой частот . . . . .	9
4.7. Функции генерации одиночных импульсов. Гауссов радиоимпульс . . . . .	11
4.8. Генерация последовательности импульсов . . . . .	12
4.9. Функции генерации периодических сигналов. Последовательность прямоугольных импульсов . . . . .	14
4.10. Функции генерации периодических сигналов. Последовательность треугольных импульсов . . . . .	15
4.11. Функции генерации периодических сигналов. Функция Дирихле . . . . .	16
4.12. Генерация сигнала с меняющейся частотой . . . . .	17
4.13. Сравнение методов корреляции . . . . .	20
<b>5. Выводы</b>	<b>20</b>

## 1. Цель работы

Познакомиться со средствами генерации и визуализации простых сигналов. Получить представление о спектрах телекоммуникационных сигналов.

## 2. Постановка задачи

В командном окне MATLAB и в среде Simulink промоделировать сигналы из Главы 3, сс. 150–170. 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 и воспользоваться алгоритмом быстрой корреляции, сравнить время работы обоих алгоритмов.

## 3. Теоретическая информация об основах анализа сигнала, о ряде Фурье, преобразовании Фурье и корреляции

Сигнал - это зависимость одной величины от другой (с математической точки зрения сигнал является функцией).

Аналоговый сигнал представляет собой функцию (функцию времени), и при его дискретизации мы получаем отсчеты, являющиеся значениями этой функции, вычисленными в дискретные моменты времени. Поэтому для расчета дискретизированного сигнала необходимо прежде всего сформулировать вектор дискретных значений времени. Для этого удобно задать значение частоты 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.

Для формирования затухающей синусоиды зададим синус с соответствующей начальной фазой:

$$s1 = A * \cos(2 * \pi * f0 * t + phi) \quad (1)$$

Домножим синусоиду (1) на экспоненту:

$$s2 = \exp(-alpha * t) * s1 \quad (2)$$

Для того, чтобы задать экспоненциальный импульс, воспользуемся следующим выражением:

$$s = A * \exp(-alpha * t), t \geq 0 \quad (3)$$

Прямоугольный импульс зададим следующим образом:

$$s = A * |t|, |t| \leq \frac{T}{2} \quad (4)$$

Несимметричный треугольный импульс можно рассчитать по следующей формуле:

$$s = A * \frac{t}{T}, 0 \leq t \leq T \quad (5)$$

Треугольный импульс, в общем случае, имеет следующий вид:

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

Здесь,  $width$  - ширина импульса, а  $skew$  - коэффициент асимметрии импульса. Соответственно меняя данные параметры можем получать импульсы нужной нам длительности и с нужным положением вершины.

Гауссов радиоимпульс, вектор рассчитанных значений которого определяется по следующей формуле:

$$y = \exp(-at^2) * \cos(2\pi f_c t) \quad (6)$$

Функция Дирихле, зависящая не только от параметра  $x$ , но и от  $n$ , которое является целым положительным числом:

$$diric_n(x) = \frac{\sin(nx/2)}{n\sin(x/2)} \quad (7)$$

В данной работе будем рассматривать спектры простейших сигналов. Для построения спектра в осях амплитуда-частота, необходимо воспользоваться разложением в ряд Фурье. Ряд Фурье может быть применен не только для периодических сигналов, но и для сигналов конечной длительности. При этом оговаривается промежуток, для которого этот ряд строится, во все остальное время сигнал считается равным нулю. Комплексная форма записи ряда Фурье:

$$s(t) = \sum_{k=-\infty}^{\infty} C_k \exp(-jk\omega t) \quad (8)$$

Амплитуды  $A_k$  и фазы  $\varphi_k$  связаны с коэффициентами ряда  $C_k$  следующим образом:

$$C_k = \frac{1}{2} A_k \exp(j\varphi_k) \quad (9)$$

$$A_k = 2|C_k| \quad (10)$$

Для нахождения посылки в сигнале можно использовать алгоритм взаимной корреляции, где  $N$  - длина векторов  $x$  и  $y$ . Для нахождения посылки будем сдвигать один вектор относительно другого, каждый раз находя значение корреляции. И максимальная корреляция будет соответствовать месту искомой посылки:

$$R = \frac{1}{N} \sum_{i=1}^N x_i * y_i \quad (11)$$

Можно ускорить алгоритм нахождения посылки, используя быструю корреляцию:

$$R = \frac{1}{N} F_D^{-1}[X^* * Y] \quad (12)$$

## 4. Ход выполнения работы

### 4.1. Расчет временных функций

### Листинг 1: Код в программе MatLab

```

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

```

Сверху представлен код программы, который генерирует 4 различных формы представления графиков дискретного сигнала (рис 4.1.1).

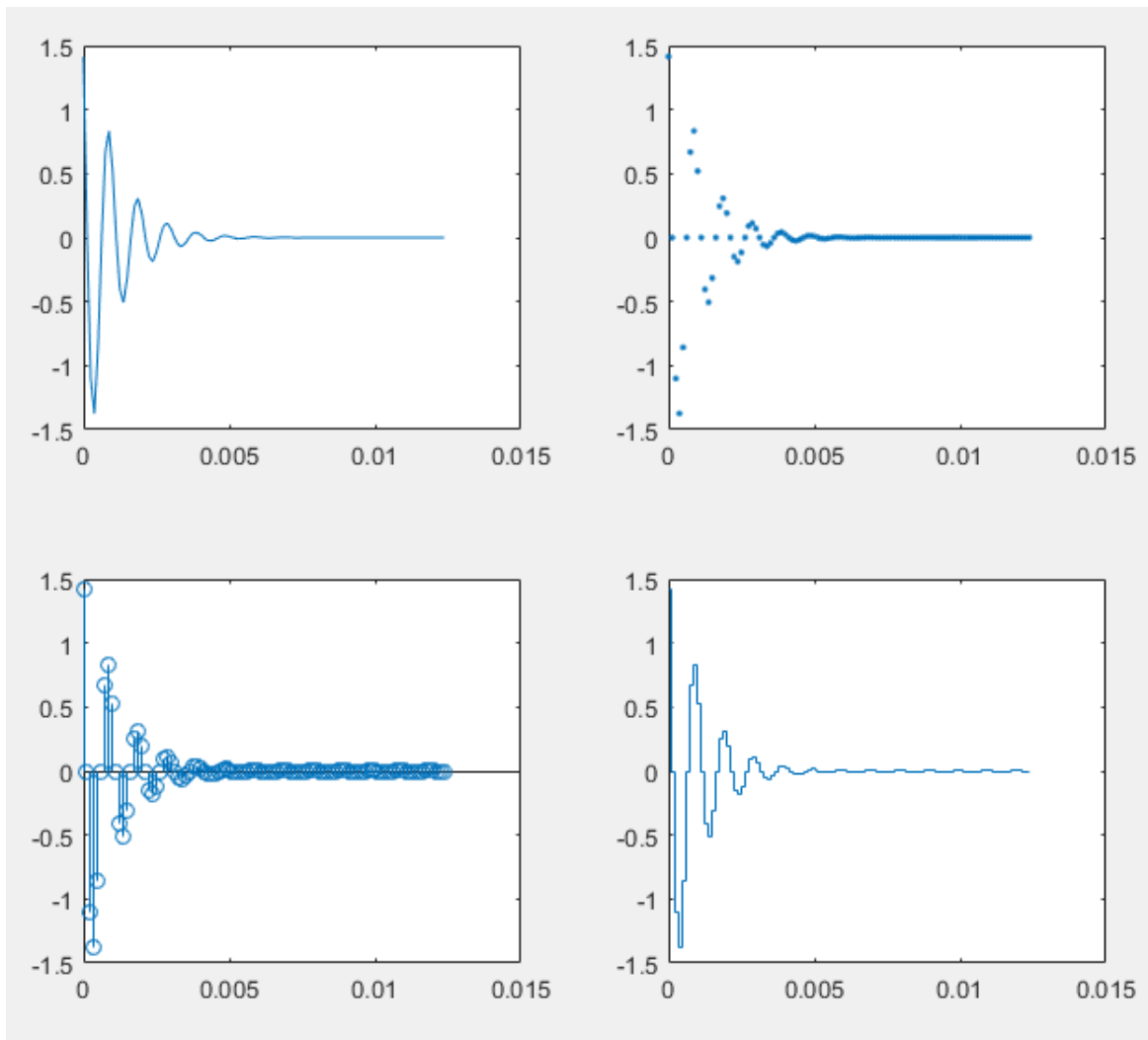


рис. 4.1.1. Различные формы представления графиков дискретного сигнала

Горизонтальная ось на приведенных графиках проградуирована в номерах отсчетов.

## 4.2. Многоканальный сигнал

Для генерации многоканального сигнала, каналы которого описываются одной и той же формулой, но с разными чистовыми значениями параметров, можно использовать средства матричных операций MatLab.

В качестве примера - генерация набора синусоид с заданными частотами 2.

Листинг 2: Код в программе MatLab

```
19 f = [600 800 1000 1200 1400];  
20 s3 = cos(2*pi*t*f);  
21 figure;  
22 plot(t(1:100), s3(1:100,:));  
24  
24 FFT = fft(s3);  
25 subplot(2, 1, 1); plot(t(1:100), s3(1:100,:));  
26 subplot(2, 1, 2); plot(t(1:8001), FFT(1:8001,:))
```

Результат вычислений представлен на рисунке 4.2.1.

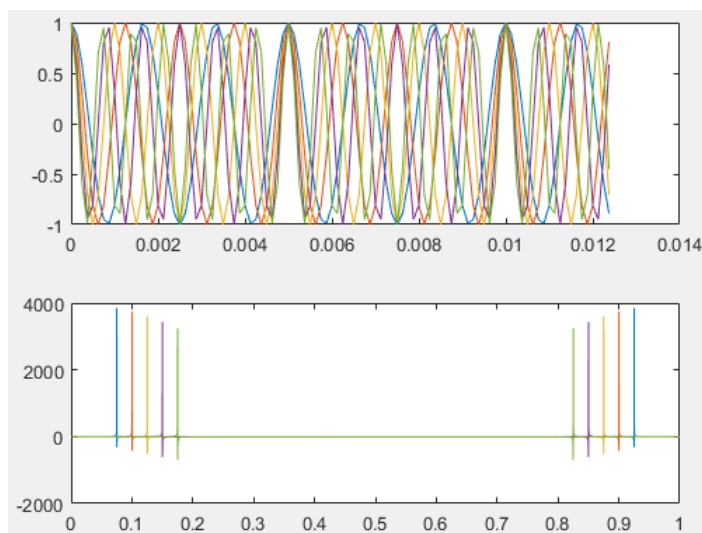


рис. 4.2.1. Многоканальный сигнал и его спектр

## 4.3. Кусочные зависимости

Кусочные зависимости - это моделирование отсчетов сигнала, который для разных интервалов времени описывается разными формулами. Простейший случай - моделирование импульсов конечной длительности, значения которых описываются одной или несколькими формулами в пределах заданного интервала, а в остальное время равны нулю.

В качестве примеров - односторонний экспоненциальный импульс, прямоугольный импульс и несимметричный треугольный импульс 3.

Листинг 3: Код в программе MatLab

```
29 T = 0.01;  
30 s1 = A * exp(-alpha * t) .* (t >= 0);  
31 figure;  
32 FFT1 = fft(s1);  
33 subplot(2, 1, 1); plot(t(1:100), s1(1:100,:));  
34 subplot(2, 1, 2); plot(t(1:8001), FFT1(1:8001,:))  
36
```

```

36 s2 = A *(abs(t) <= T/2);
37 figure;
38 FFT2 = fft(s2);
39 subplot(2, 1, 1); plot(t(1:100), s2(1:100,:));
40 subplot(2, 1, 2); plot(t(1:8001), FFT2(1:8001,:))
43
43
43 s3 = A * t/T .* (t >= 0) .* (t <= T);
44 figure;
45 FFT3 = fft(s3);
46 subplot(2, 1, 1); plot(t(1:100), s3(1:100,:));
47 subplot(2, 1, 2); plot(t(1:8001), FFT3(1:8001,:))

```

Результат представлен на рисунках 4.3.1, 4.3.2, 4.3.3.

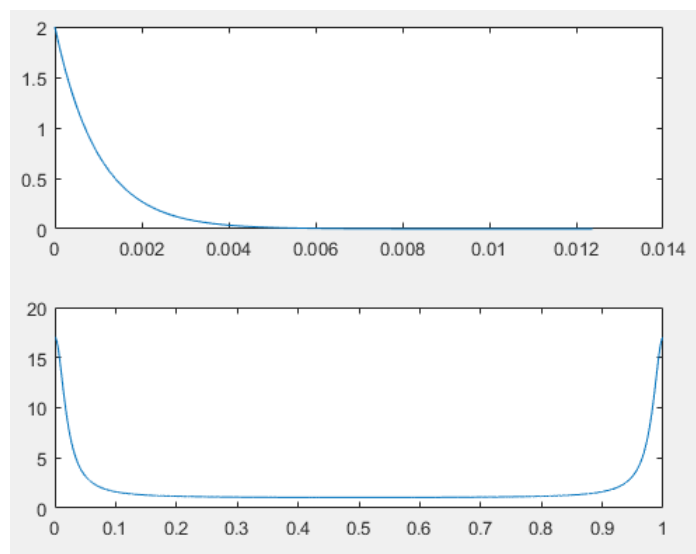


рис. 4.3.1. Односторонний экспоненциальный импульс и его спектр

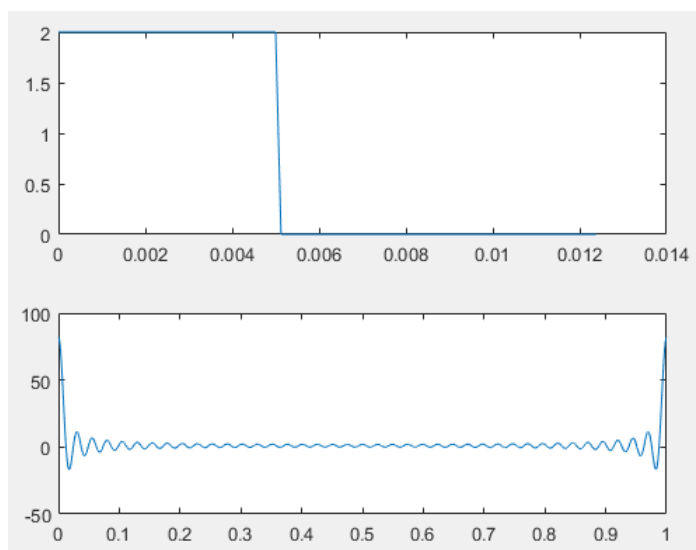


рис. 4.3.2. Прямоугольный импульс и его спектр

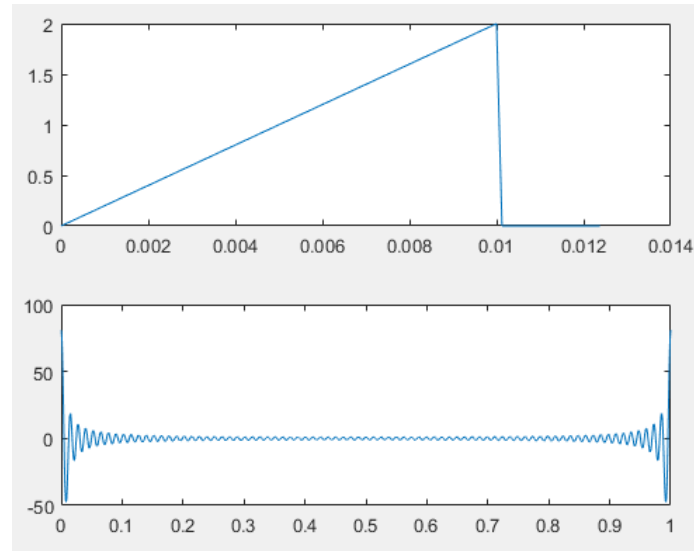


рис. 4.3.3. Несимметричный треугольный импульс и его спектр

#### 4.4. Функции генерации одиночных импульсов. Прямоугольный импульс

В пакете Signal Processing присутствует ряд функций, генерирующих часто встречающиеся на практике непериодические сигналы.

Для формирования одиночного прямоугольного импульса с единичной амплитудой будем использовать функцию *rectpuls*.

В качестве примеров сформируем пару разнополярных прямоугольных импульсов с амплитудой 5 В и длительностью 20 мс каждый, расположенных справа и слева от начала отсчета времени. Частота дискретизации - 1 кГц (листинг4).

Листинг 4: Код в программе MatLab

```

52 Fs = 1e3;
53 t = -40e-3:1/Fs:40e-3;
54 T = 20e-3;
55 A = 5;
56 s = -A*rectpuls(t + T/2, T) + A*rectpuls(t - T/2, T);
57 FFT3 = fft(s);
58 figure;
59 subplot(2, 1, 1); plot(t, s);
60 subplot(2, 1, 2); plot(t, FFT3)

```

Результат представлен на рисунке 4.4.1.



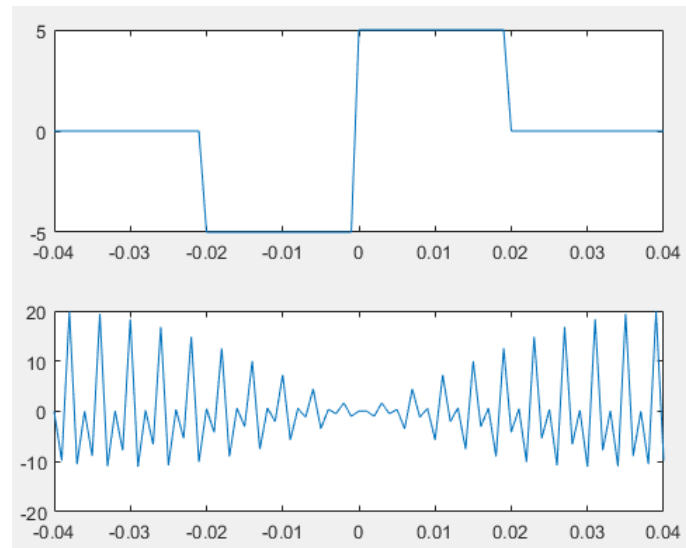


рис. 4.4.1. Сигнал, сформированный с использованием функции `rectpuls` и его спектр

#### 4.5. Функции генерации одиночных импульсов. Треугольный импульс

Для формирования одиночного треугольного импульса с единичной амплитудой будем использовать функцию `tripuls`.

В качестве примера сформируем симметричный трапецевидный импульс с амплитудой 10 В и размерами верхнего и нижнего оснований 20 и 60 мс соответственно. Частота дискретизации - 1 кГц (листинг5).

Листинг 5: Код в программе MatLab

```

64 Fs = 1e3;
65 t = -50e-3:1/Fs:50e-3;
66 A = 10;
67 T1 = 20e-3;
68 T2 = 60e-3;
69 s = A*(T2*tripuls(t, T2) - T1*tripuls(t, T1))/(T2-T1);
70 FFT3 = fft(s);
71 figure;
72 subplot(2, 1, 1); plot(t, s);
73 subplot(2, 1, 2); plot(t, FFT3)

```

Результат представлен на рисунке 4.5.1.

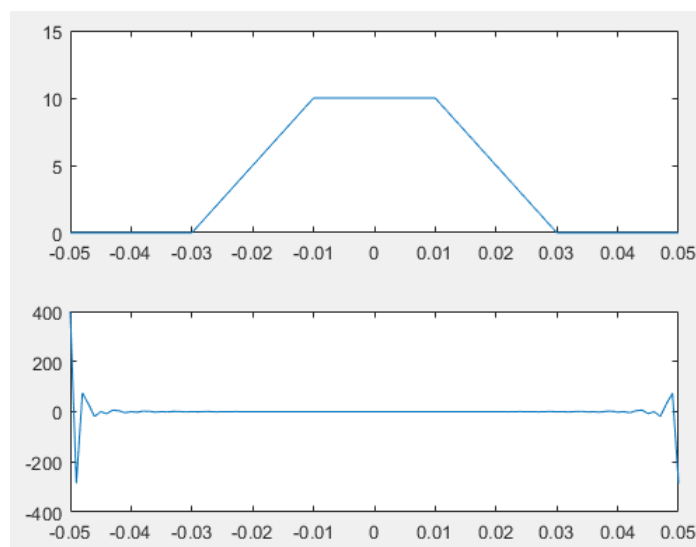


рис. 4.5.1. Сигнал, сформированный с использованием функции `tripuls` и его спектр

## 4.6. Функции генерации одиночных импульсов. Импульс с ограниченной полосой частот

Для формирования сигнала, имеющего прямоугольный, то есть ограниченный по частоте спектр, служит функция `sinc`.

В качестве примера посмотрим с помощью функции `sinc` график амплитудного спектра очень короткого радиоимпульса, на длительности которого укладывается лишь один период синусоидального заполнения. (листинг6).

Листинг 6: Код в программе MatLab

```

76 Fs = 1e3;
77 t = -0.1:1/Fs:0.1;
78 f0 = 10;
79 T = 1/f0;
80 s = rectpuls(t, T) .* cos(2*pi*f0*t);
81 f = -50:50;
82 sp = T/2 * (sinc((f-f0)*T) + sinc((f+f0)*T));
83 figure;
84 plot(t, s);
85 ylim([-1.1 1.1]);
86 figure;
87 plot(f, abs(sp));

```

Результат представлен на рисунках 4.6.1 , 4.6.2.

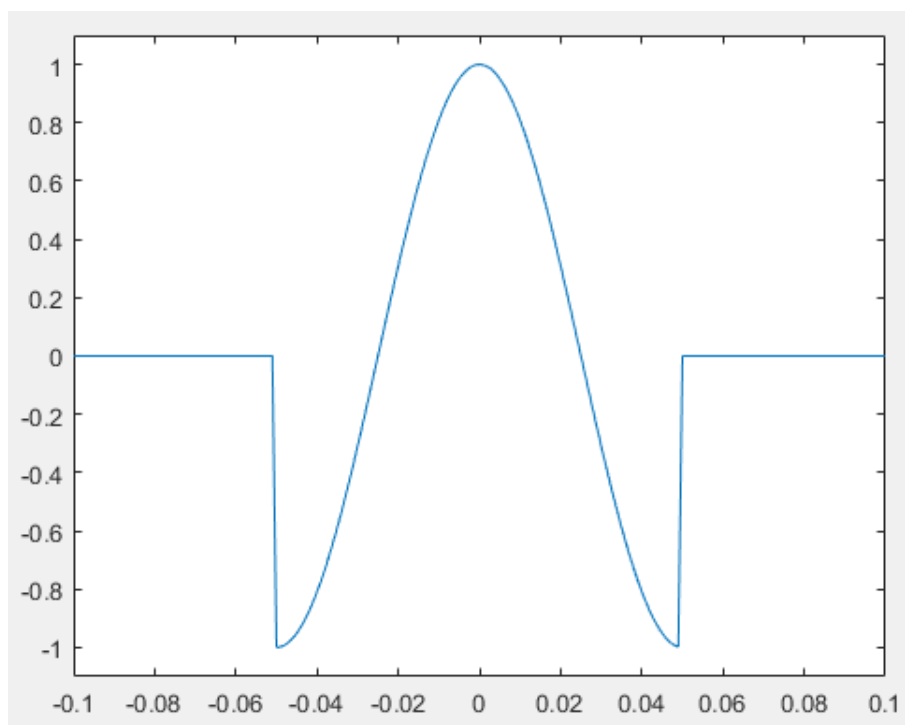


рис. 4.6.1. Короткий радиоимпульс

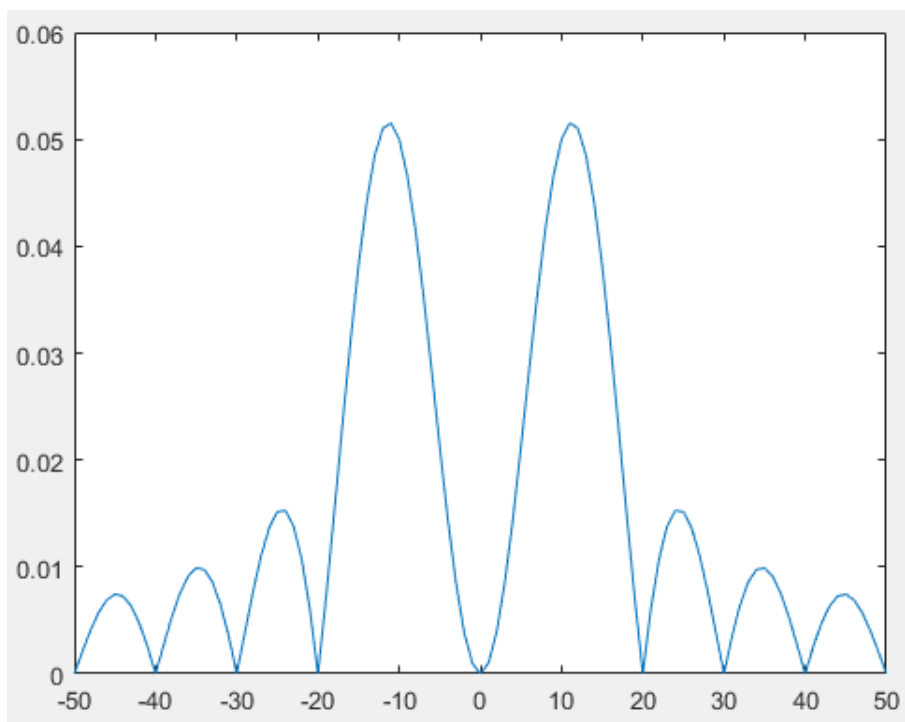


рис. 4.6.2. Амплитудный спектр радиоимпульса, построенный с помощью функции sinc

## 4.7. Функции генерации одиночных импульсов. Гауссов радиопульс

Для формирования одиночного радиопульса с гауссовой огибающей и единичной амплитудой служит функция *gauspuls*.

В качестве примера сформируем гауссов радиопульс с несущей частотой 4 кГц и относительной шириной спектра 10%, измеренной по уровню -20 дБ, а затем построим график его спектра, чтобы убедиться в правильности расчетов. Частота дискретизации - 16 кГц. (листинг7).

Листинг 7: Код в программе MatLab

```
90 Fs = 16e3;
91 t = -10e-3:1/Fs:10e-3;
92 Fc = 4e3;
93 bw = 0.1;
94 bwr = -20;
95 s = gauspuls(t, Fc, bw, bwr);
96 Nfft = 2^nextpow2(length(s));
97 sp = fft(s, Nfft);
98 sp_dB = 20*log10(abs(sp));
99 f = (0:Nfft - 1)/Nfft*Fs;
100 figure;
101 plot(t, s);
102 figure;
103 plot(f(1:Nfft/2), sp_dB(1:Nfft/2));
104 sp_max_db = 20*log10(max(abs(sp)));
105 edges = Fc * [1-bw/2 1+bw/2];
106 hold on
107 plot(edges, sp_max_db([1 1]) +bwr, 'o');
108 hold off
109 gauspuls('cutoff', Fc, bw, bwr, -6);
```

Результат представлен на рисунках 4.7.1 , 4.7.2.

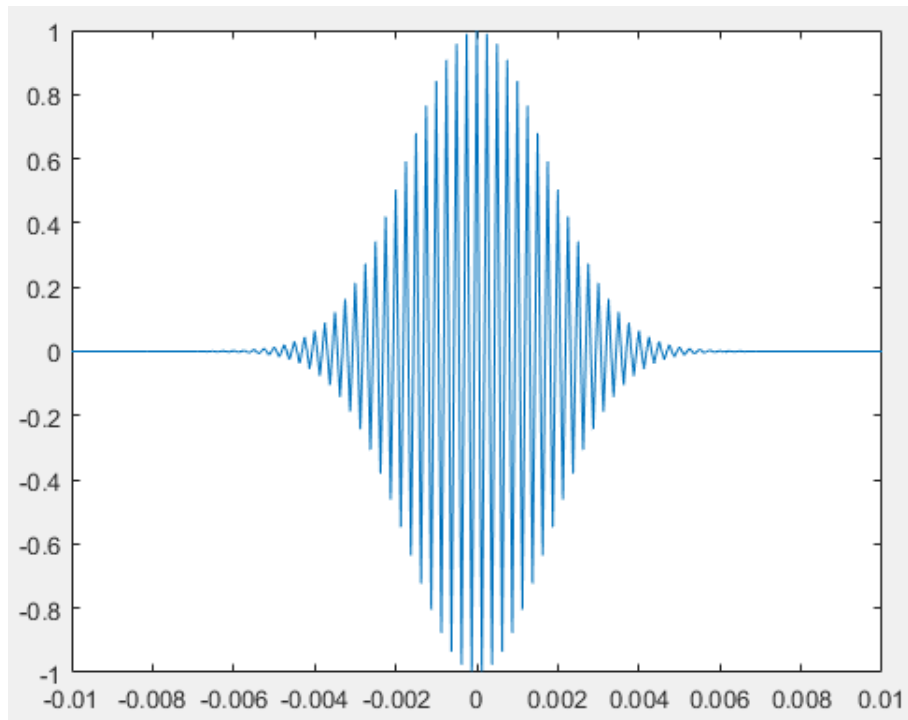


рис. 4.7.1. Гауссов радиоимпульс, сформированный функцией `rauspuls`

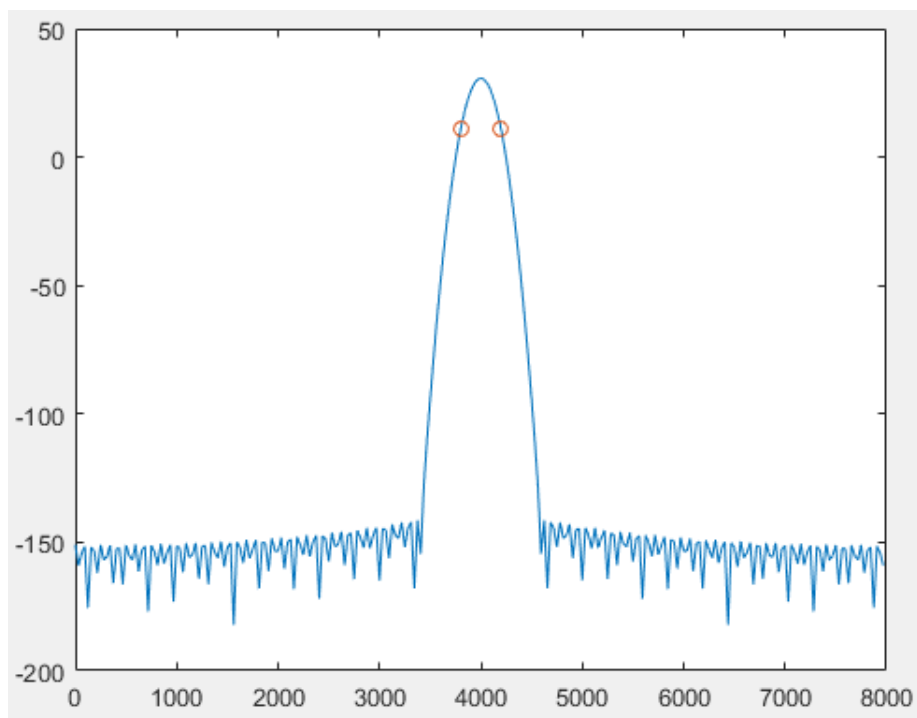


рис. 4.7.2. Амплитудный спектр гауссова радиоимпульса

## 4.8. Генерация последовательности импульсов

Функция *pulstran* служит для генерации конечной последовательности импульсов (pulse train) одинаковой формы с произвольно задаваемыми задержками и уровнями.

В качестве примера сформируем последовательность из пяти симметричных треугольных импульсов, интервалы между которыми линейной увеличиваются, а амплитуды экспоненциально уменьшаются. Частота дискретизации - 1 кГц. Длительность импульса - 20 мс. (листинг8).

Листинг 8: Код в программе MatLab

```
112 Fs = 1e3;
113 t = 0:1/Fs:0.5;
114 tau = 20e-3;
115 d = [20 80 160 260 380] ' * 1e-3;
116 d(:,2) = 0.8.^(0:4)';
117 y = pulstran(t, d, 'tripuls', tau);
118 FFT3 = fft(y);
119 figure;
120 subplot(2, 1, 1); plot(t, y);
121 subplot(2, 1, 2); plot(t, FFT3)
```

Результат представлен на рисунке 4.8.1.

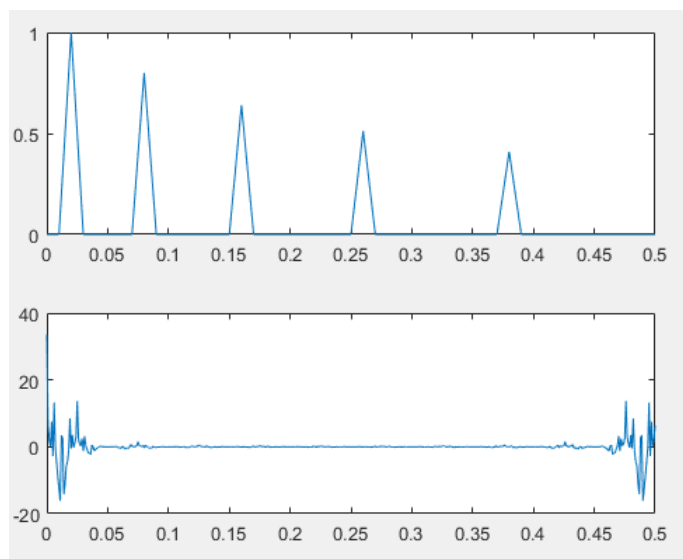


рис. 4.8.1. Последовательность треугольных импульсов, сформированная с помощью функции pulstran и их спектр

Поскольку частота дискретизации может не совпадать с шагом значений вектора и задержки из вектора тоже не всегда кратны этому шагу, для пересчета задержанных импульсов к сетке моментов необходимо использовать интерполяцию.

В качестве примера сформируем последовательность из шести импульсов, имеющих форму одного периода функции  $\sin \cdot \sin$ . Пусть длительность импульса равна 60 мс, а частота его дискретизации - 400 Гц. Расстояние между центрами импульсов будет одинаковым и равным 64 мс, а частота дискретизации выходного сигнала - 1 кГц. (листинг9).

### Листинг 9: Код в программе MatLab

```

124 Fs0 = 400;
125 tau = 60e-3;
126 t0 = 0:1/Fs0:tau;
127 s0 = sin(pi*t0/tau).^2;
128 Fs = 1e3;
129 t = 0:1/Fs:0.5;
130 d = (1:6)'*64e-3;
131 d(:,2) = 0.6.^(0:5)';
132 y = pulstran(t, d, s0, Fs0);
133 FFT3 = fft(y);
134 figure;
135 subplot(2, 1, 1); plot(t, y);
136 subplot(2, 1, 2); plot(t, FFT3)

```

Результат представлен на рисунке 4.8.2.

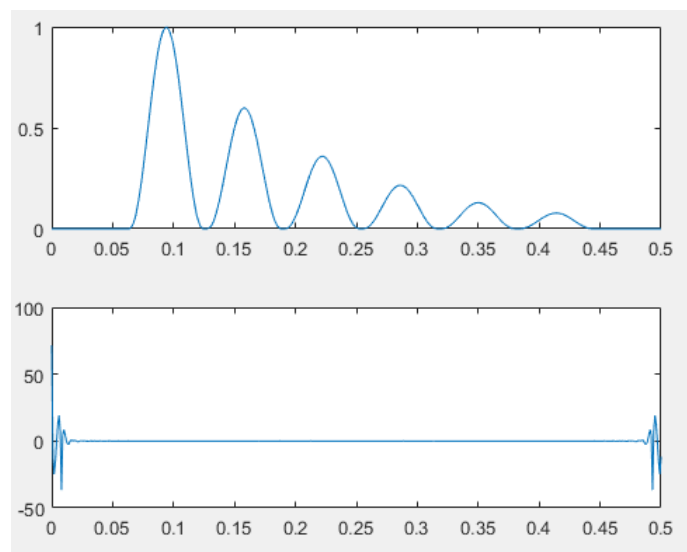


рис. 4.8.2. Последовательность импульсов, сформированная функцией `pulstran` из вектора отсчетов одиночного импульса и их спектр

## 4.9. Функции генерации периодических сигналов. Последовательность прямоугольных импульсов

Функция `square` служит для формирования последовательности прямоугольных импульсов.

В качестве примера сформируем последовательность однополярных прямоугольных импульсов с амплитудой 3 В, частотой следования 50 Гц и длительностью 5 мс. Будем использовать частоту дискретизации 1 кГц и временной интервал -10..50 мс. (листинг10).

### Листинг 10: Код в программе MatLab

```

140 Fs = 1e3;
141 t = -10e-3:1/Fs:50e-3;
142 A = 3;
143 f0 = 50;
144 tau = 5e-3;
145 s = (square(2*pi*t*f0, f0*tau*100) + 1) * A/2;
146 FFT3 = fft(s);

```

```

147 figure;
148 subplot(2, 1, 1); plot(t, s);
149 subplot(2, 1, 2); plot(t, FFT3)

```

Результат представлен на рисунке 4.9.1.

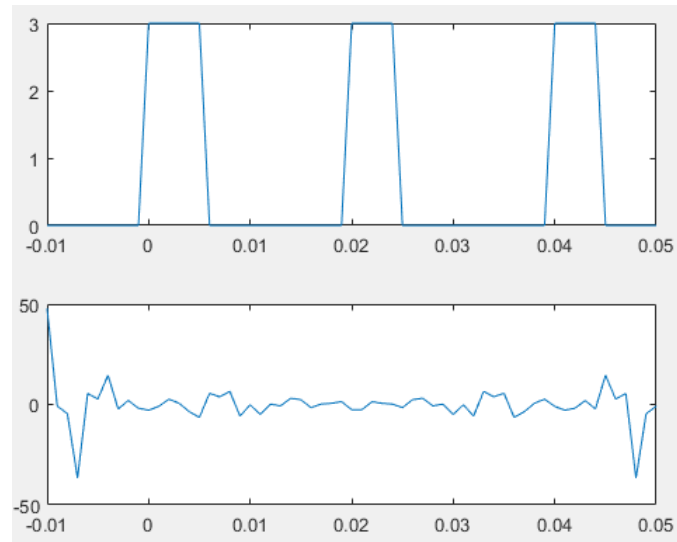


рис. 4.9.1. Последовательность прямоугольных импульсов, полученная с помощью функции `square` и их спектр

#### 4.10. Функции генерации периодических сигналов. Последовательность треугольных импульсов

Функция `sawtooth` служит для формирования последовательности треугольных импульсов.

В качестве примера сформируем последовательность треугольных импульсов отрицательной полярности с амплитудой 5 В, периодом 50 мс и длительностью падающего участка 5 мс. Частота дискретизации - 1 кГц и временной интервал -25.. 125 мс. (листинг11).

Листинг 11: Код в программе MatLab

```

152 Fs = 1e3;
153 t = -25e-3:1/Fs:125e-3;
154 A = 5;
155 T = 50e-3;
156 t1 = 5e-3;
157 s = (sawtooth(2*pi*t/T, 1 - t1/T) - 1) * A/2;
158 FFT3 = fft(s);
159 figure;
160 subplot(2, 1, 1); plot(t, s);
161 subplot(2, 1, 2); plot(t, FFT3)

```

Результат представлен на рисунке 4.10.1.



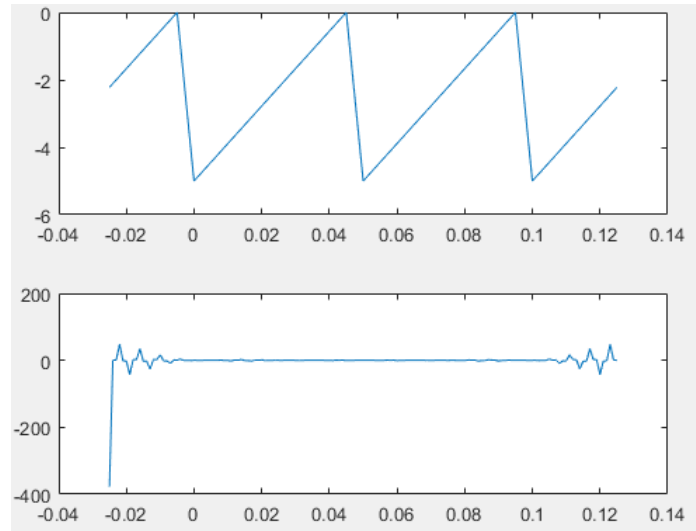


рис. 4.10.1. Последовательность треугольных импульсов, полученная с помощью функции `sawtooth` и их спектр

#### 4.11. Функции генерации периодических сигналов. Функция Дирихле

Функция *diric* служит для формирования функции Дирихле.

Функцию Дирихле называют еще периодической sinc-функцией.

В качестве примера построим графики функции Дирихле при нечетном и четном значениях  $n$  (листинг12).

Листинг 12: Код в программе MatLab

```

163 x = 0:0.01:15;
164 FFT3 = fft(diric(x,7));
165 figure;
166 subplot(2, 1, 1); plot(x, diric(x, 7));
167 subplot(2, 1, 2); plot(x, FFT3);
168 grid on;
169 title('n=7');
170 FFT4 = fft(diric(x,8));
171 figure;
172 subplot(2, 1, 1); plot(x, diric(x, 8));
173 subplot(2, 1, 2); plot(x, FFT4);
174 grid on;
175 title('n=8');

```

Результат представлен на рисунках 4.11.1 , 4.11.2.

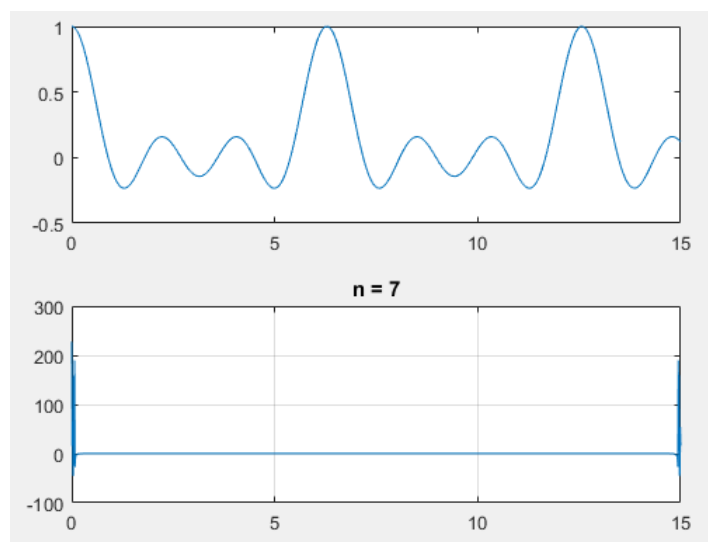


рис. 4.11.1. Функция Дирихле нечетного порядка и ее спектр

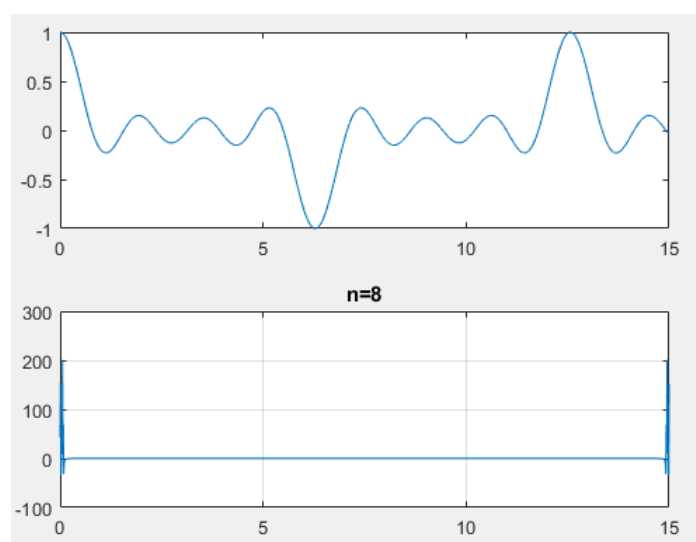


рис. 4.11.2. Функция Дирихле четного порядка и ее спектр

## 4.12. Генерация сигнала с меняющейся частотой

Функция *chirp* предназначена для генерации колебаний с единичной амплитудой, мгновенная частота которых меняется по заданному закону.

В качестве примера сформируем три сигнала, определенных на промежутке 0...1 с и имеющих разные законы изменения мгновенной частоты. В нулевой момент времени все сигналы имеют мгновенную частоту 1 кГц, а в момент времени 1 с - 2 кГц. Частоту дискретизации выберем равной 8 кГц. (листинг13).

Листинг 13: Код в программе MatLab

```
178 Fs = 8e3;
179 t = 0:1/Fs:1;
180 f0 = 1e3;
```

```

181 t1 = 1;
182 f1 = 2e3;
183 s1 = chirp(t, f0, t1, f1, 'linear');
184 s2 = chirp(t, f0, t1, f1, 'quadratic');
185 s3 = chirp(t, f0, t1, f1, 'logarithmic');
186 specgram(s1, [], Fs);
187 title('linear')
188 colormap gray
189 figure
190 specgram(s2, [], Fs)
191 title('quadratic')
192 colormap gray
193 figure
194 specgram(s3, [], Fs)
195 title('logarithmic')
196 colormap gray

```

Результат представлен на рисунках 4.12.1 , 4.12.2 , 4.12.3

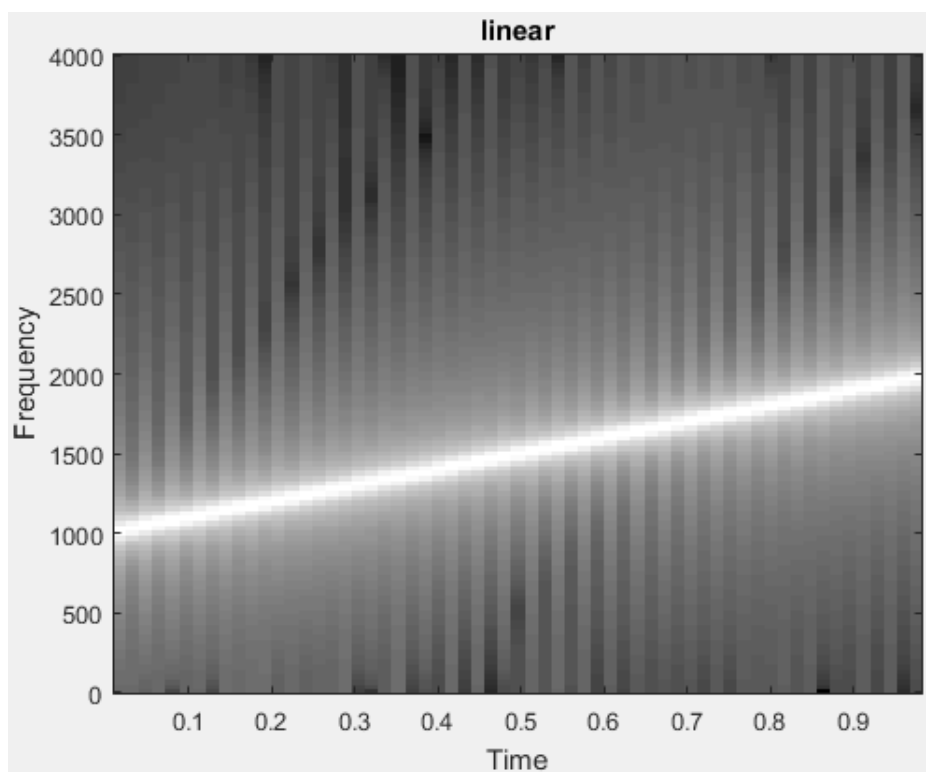


рис. 4.12.1. Спектограмма сигнала, сформированного функцией `chirp` при линейном законе изменения мгновенной частоты

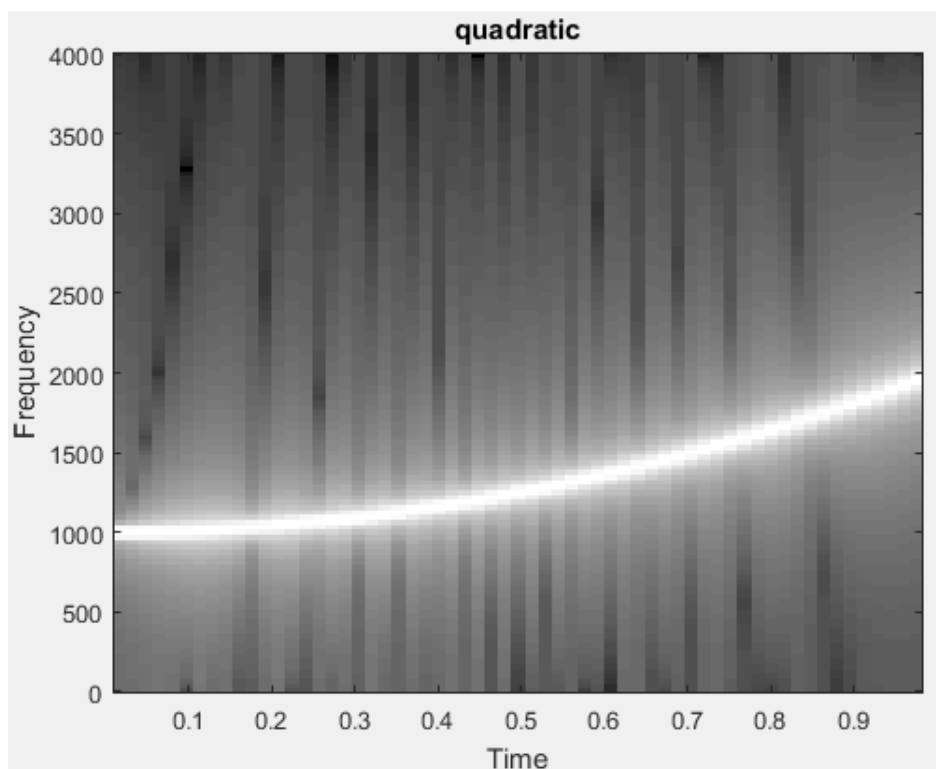


рис. 4.12.2. Спектограмма сигнала, сформированного функцией chirp при квадратичном законе изменения мгновенной частоты

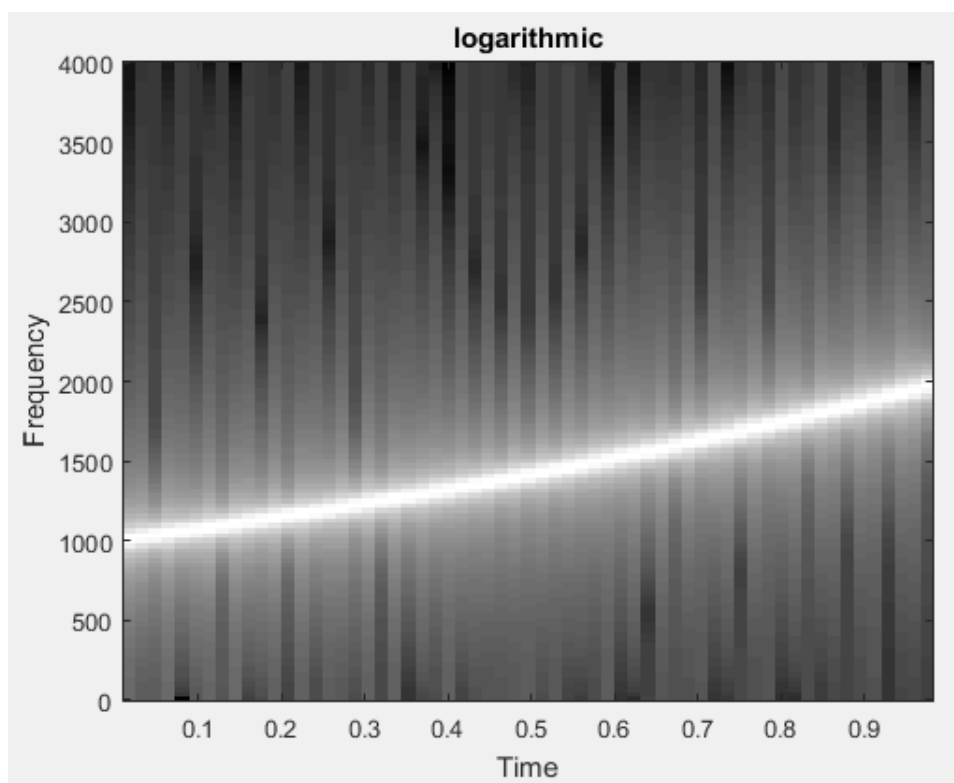


рис. 4.12.3. Спектограмма сигнала, сформированного функцией chirp при экспоненциальном законе изменения мгновенной частоты

### 4.13. Сравнение методов корреляции

В качестве исходного примера было взято нахождение синхропосылки [101] в сигнале [0001010111000010]. Листинг исходного кода 14.

Листинг 14: Код в программе MatLab

```
24 tic
25 for i = 1:length(xx)
26     R(i) = sum(xx .* circshift(yy, i-1, 2)) / length(xx);
27 end
28 toc
30
30 tic
31 xx = fft(xx);
32 yy = fft(yy);
33 xx = conj(xx);
34 BR = ifft(xx .* yy)/length(xx);
35 toc
```

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

Время выполнения прямой корреляции: 0,124 мс.

Время выполнения быстрой корреляции: 0,029 мс.

## 5. Выводы

В данной лабораторной работе мы ознакомились с различными средствами генерации и визуализации простых сигналов в пакете MatLab. Были рассмотрены многоканальные сигналы, кусочные зависимости, одиночные импульсы, такие как: прямоугольный, треугольный, с ограниченной полосой частот и Гауссов радиоимпульс. Были сгенерированы последовательности различных импульсов. Также рассмотрены периодические сигналы и сигналы с меняющейся частотой. На основе проделанной работы можно сделать вывод о том, что сигналов различных видов существует очень много, соответственно классификация сигналов достаточно обширна.

Основными признаками классификации сигналов являются:

- Характер измерения информативного и временного параметров (аналоговый, дискретный, цифровой).
- Характер изменения во времени (постоянные и переменные).
- По степени наличия априорной информации (детерминированные, квазидетерминированные и случайные).
- Периодичность сигналов

В качестве примеров применения преобразования Фурье в телекоммуникационных технологиях можно привести обработку звука и изображений (их сжатие и кодировка, восстановление и улучшение, обработка массивов отсчетов), модуляция и демодуляция данных для передачи по каналам связи, фильтрация сигналов.