

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:**  
Ерниязов Т.Е.  
Группа: 33501/3  
**Преподаватель:**  
Богач Н.В.

Санкт-Петербург  
2017

# Содержание

<b>1. Цель работы</b>	<b>2</b>
<b>2. Программа работы</b>	<b>2</b>
<b>3. Теоретическая информация</b>	<b>2</b>
<b>4. Ход выполнения работы</b>	<b>3</b>
4.1. Расчет временных функций . . . . .	3
4.2. Многоканальный сигнал . . . . .	4
4.3. Кусочные зависимости . . . . .	5
4.4. Генерация одиночных импульсов . . . . .	7
4.5. Генерация последовательности импульсов . . . . .	10
4.6. Функции генерации периодических сигналов . . . . .	12
4.7. Генерация сигналов с меняющейся частотой . . . . .	15
4.8. Нахождение посылки в сигнале с помощью методов корреляции . . . . .	17
<b>5. Выводы</b>	<b>18</b>

# 1. Цель работы

Познакомиться со средствами генерации и визуализации простых сигналов. Получить навыки построения спектров телекоммуникационных сигналов.

# 2. Программа работы

В командном окне MATLAB и в среде Simulink промоделировать сигналы из Главы 3, сс. 150-170 А.Б. Сергиенко Цифровая обработка сигналов. 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

Сигнал - это зависимость одной величины от другой.

## Классификация сигналов:

Различают детерминированные и случайные сигналы. Детерминированный сигнал полностью известен - его значение в любой момент времени можно определить точно. Случайный сигнал в любой момент времени представляет собой случайную величину, которая принимает конкретные значения с некоторой вероятностью.

Другой класс сигналов - сигналы с интегрируемым квадратом (с ограниченной энергией). Для таких сигналов  $s(t)$  выполняется соотношение  $\int_{-\infty}^{\infty} s(t)dt$

Еще один признак классификации - периодичность. Для периодических сигналов  $s(t) = s(t + nT)$  для любого  $t$ .

Также выделяются сигналы конечной длительности - отличные от нуля на ограниченном промежутке времени.

Для генерации одиночных импульсов в MATLAB используются функции:

- `rectpuls` — прямоугольный импульс
- `tripuls` — треугольный импульс
- `sinc` — импульс  $\sin(\pi t)/(\pi t)$
- `gauspuls` — радиоимпульс с гауссовой огибающей
- `pulstran` — последовательность из конечного числа импульсов произвольной формы

Следующие функции позволяют формировать периодические последовательности импульсов: `square` — прямоугольные; `sawtooth` — треугольные; `diric` — функция Дирихле.

Функция Дирихле имеет следующий вид:  $diric(x) = \frac{\sin(nx/2)}{n\sin(x/2)}$

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

Для построения амплитудно-частотного спектра сигнала его раскладывают в ряд Фурье. Запись ряда Фурье в комплексной форме:

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

Коэффициент ряда  $k$  связан с амплитудой  $A_k$  и фазой  $\phi_k$ :

$$C_k = \frac{1}{2} A_k \exp(-j\phi_k)$$

Для нахождения послания в сигнале используется метод взаимной корреляции. Его суть заключается в сдвиге векторов  $x$  и  $y$  относительно друг друга, каждый раз рассчитывая значение корреляции. Место максимальной корреляции соответствует месту посылки.

$$R = \frac{1}{N} \sum_{i=1}^N x_k \cdot y_i$$

$N$  в этой формуле это длина векторов  $x$  и  $y$ .

## 4. Ход выполнения работы

### 4.1. Расчет временных функций

Визуализация затухающих колебаний с частотой дискретизации 8КГц:

Листинг 1: Расчет временных функций

```

1 % 1
2 Fs = 8e3;
3 t = 0:1/Fs:1;
4 t = t';
5
6 T = 50;
7 A = 2;
8 f0 = 1e3;
9 phi = pi/4;
10 s1 = A * cos(2*pi*f0*t+phi);
11 alpha = 1e3;
12 s2 = exp(-alpha*t).*s1;
13
14
15 subplot(2,2,1):plot(t(1:100), s2(1:100))
16 subplot(2,2,2):plot(t(1:100), s2(1:100), ' . ')
17 subplot(2,2,3):stem(t(1:100), s2(1:100))
18 subplot(2,2,4):stairs(t(1:100), s2(1:100))

```

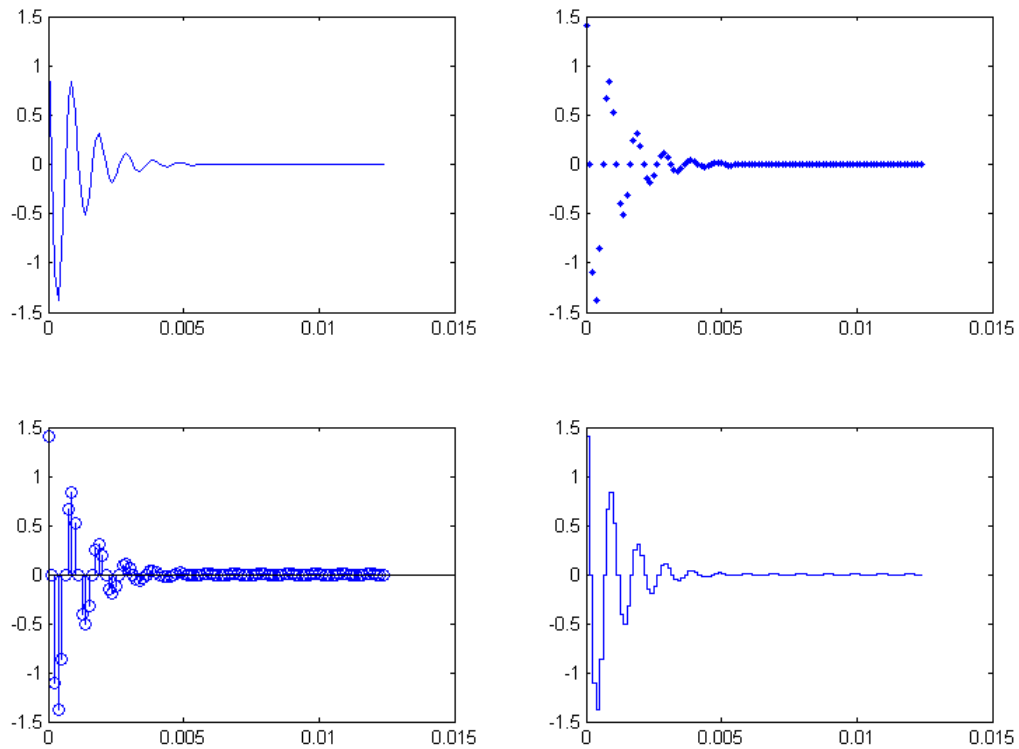


Рисунок 4.1. Расчет временных функций

## 4.2. Многоканальный сигнал

Генерация набора синусоид с разными частотами:

Листинг 2: Многоканальный сигнал и его спектр

```

1 f = [600 800 1000 1200 1400];
2 s3 = cos(2*pi*t*f);
3 figure;
4 subplot(1, 1, 1)
5 plot(t(1:100), s3(1:100,:))
6
7 FFT = fft(s3);
8 subplot(2, 1, 1); plot(t(1:100), s3(1:100,:));
9 subplot(2, 1, 2); plot(t(1:8001), FFT(1:8001,:));

```

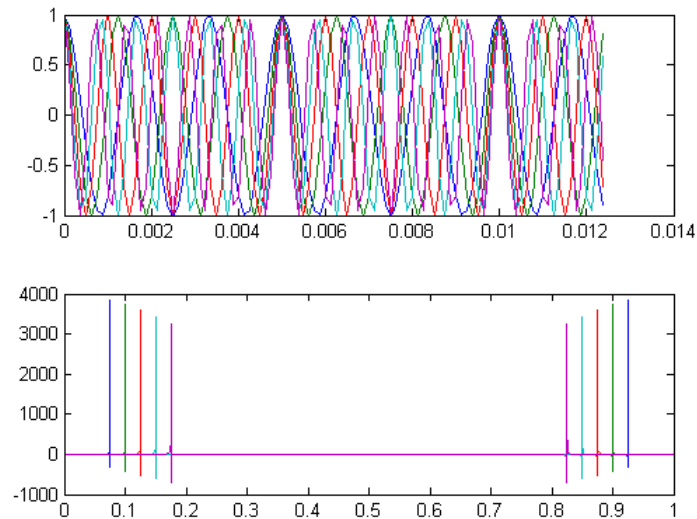


Рисунок 4.2. Многокальный сигнал и его спектр

### 4.3. Кусочные зависимости

Листинг 3: Односторонний экспоненциальный импульс и его спектр

```

1 T = 0.01;
2 s1 = A * exp(-alpha * t).*(t>=0);
3 figure;
4 FFT1 = fft(s1);
5 subplot(2, 1, 1); plot(t(1:100), s1(1:100,:));
6 subplot(2, 1, 2); plot(t(1:8001), FFT1(1:8001,:))

```

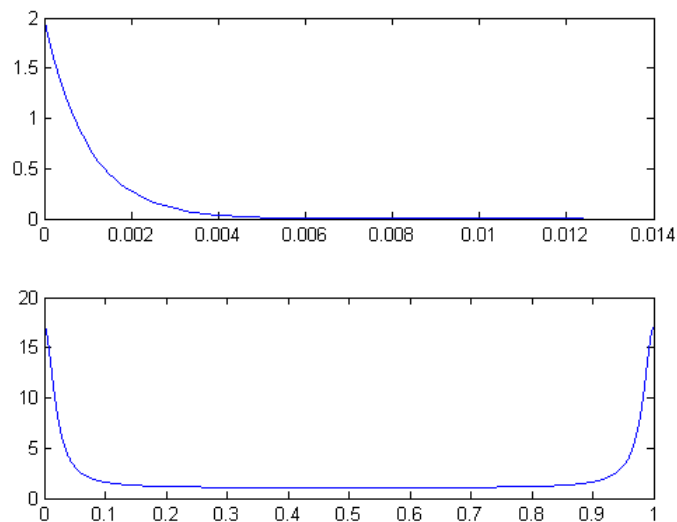


Рисунок 4.3. Односторонний экспоненциальный импульс и его спектр

Листинг 4: Прямоугольный импульс и его спектр

```

1 s2 = A * (abs(t)<=T/2);
2 figure;
3 FFT2 = fft(s2);
4 subplot(2, 1, 1); plot(t(1:100), s2(1:100,:));
5 subplot(2, 1, 2); plot(t(1:8001), FFT2(1:8001,:));

```

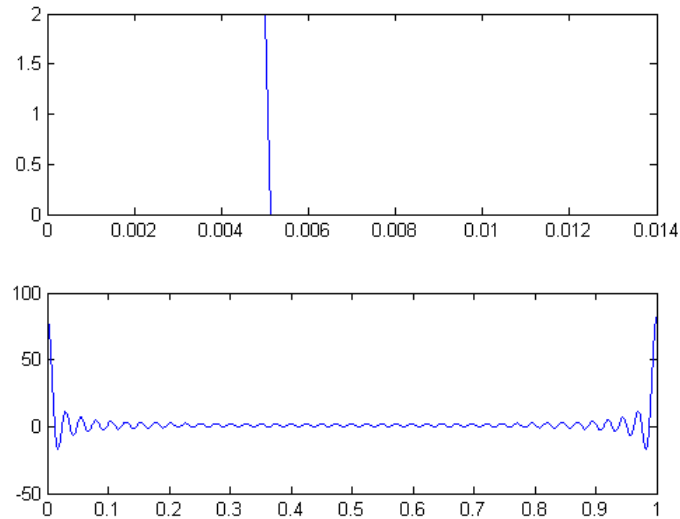


Рисунок 4.4. Прямоугольный импульс и его спектр

Листинг 5: Треугольный импульс и его спектр

```

1 s3 = A * t/T.*(t>=0).*(t<=T);
2 figure;
3 FFT3 = fft(s3);
4 subplot(2, 1, 1); plot(t(1:100), s3(1:100,:));
5 subplot(2, 1, 2); plot(t(1:8001), FFT3(1:8001,:));

```

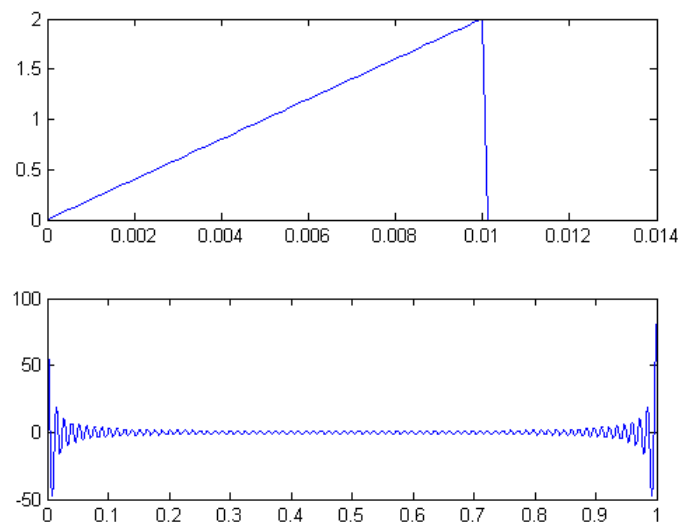


Рисунок 4.5. Треугольный импульс и его спектр

## 4.4. Генерация одиночных импульсов

В пакете Signal Processing имеется ряд специальных функций, генерирующих одиночные сигналы.

Прямоугольный импульс

Листинг 6: Прямоугольный импульс и его спектр

```
1 width = 1;
2 y = rectpuls(t, width)
3 Fs = 1e3;
4 t = -40e-3:1/Fs:40e-3;
5 T = 20e-3;
6 A = 5;
7 s = -A*rectpuls(t+T/2,T)+A*rectpuls(t-T/2,T);
8 FFT3 = fft(s);
9 figure;
10 subplot(2, 1, 1); plot(t, s);
11 subplot(2, 1, 2); plot(t, FFT3);
```

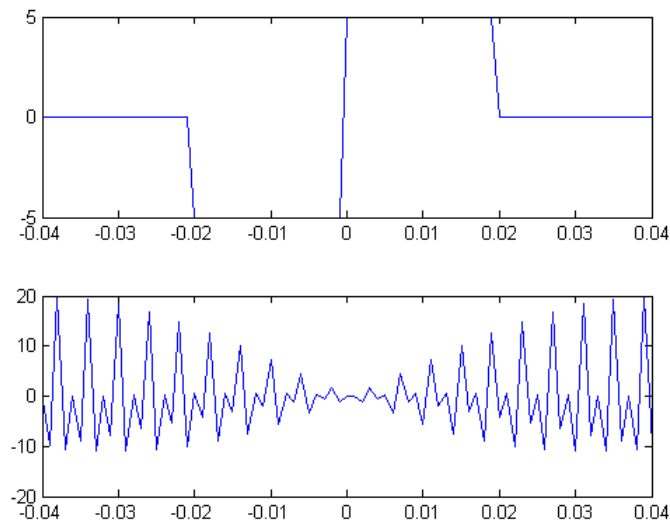


Рисунок 4.6. Прямоугольный импульс и его спектр

Треугольный импульс

Листинг 7: Треугольный импульс и его спектр

```
1 skew = 0;
2 y = tripuls(t,width, skew)
3 Fs = 1e3;
4 t = -50e-3:1/Fs:50e-3;
5 A = 10;
6 T1 = 1e-3;
7 T2 = 60e-3;
8 s = A*(T2*tripuls(t,T2) - T1*tripuls(t, T1))/(T2-T1);
9 FFT3 = fft(s);
10 figure;
11 subplot(2, 1, 1); plot(t, s);
12 subplot(2, 1, 2); plot(t, FFT3);
```

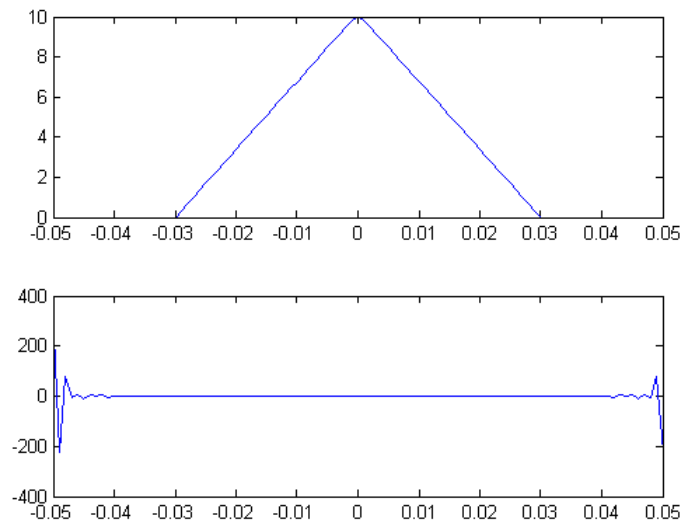


Рисунок 4.7. Треугольный импульс и его спектр

#### Импульс с ограниченной полосой частот

```

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 = -50:50;
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))

```

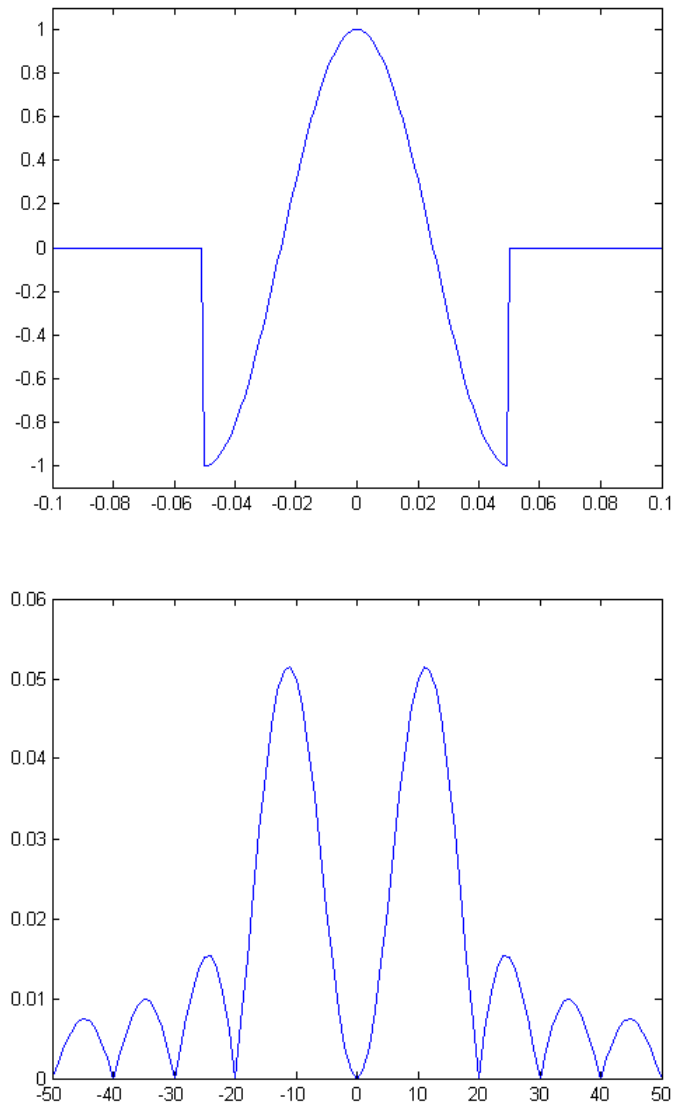


Рисунок 4.8. Короткий радиоимпульс и его спектр

Гауссов радиоимпульс

Листинг 8: Гауссов радиоимпульс

```

1 Fs = 16e3;
2 t = -10e-3:1/Fs:10e-3;
3 Fc = 4e3;
4 bw = 0.1;
5 bwr = -20;
6 s = gauspuls(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
14 plot(f(1:Nfft/2), sp_dB(1:Nfft/2))
15

```

```

16 sp_max_db = 20*log10(max(abs(sp)));
17 edges = Fc * [1-bw/2 1+bw/2];
18 hold on
19 plot(edges, sp_max_db([1 1])+bwr, 'o')
20 hold off

```

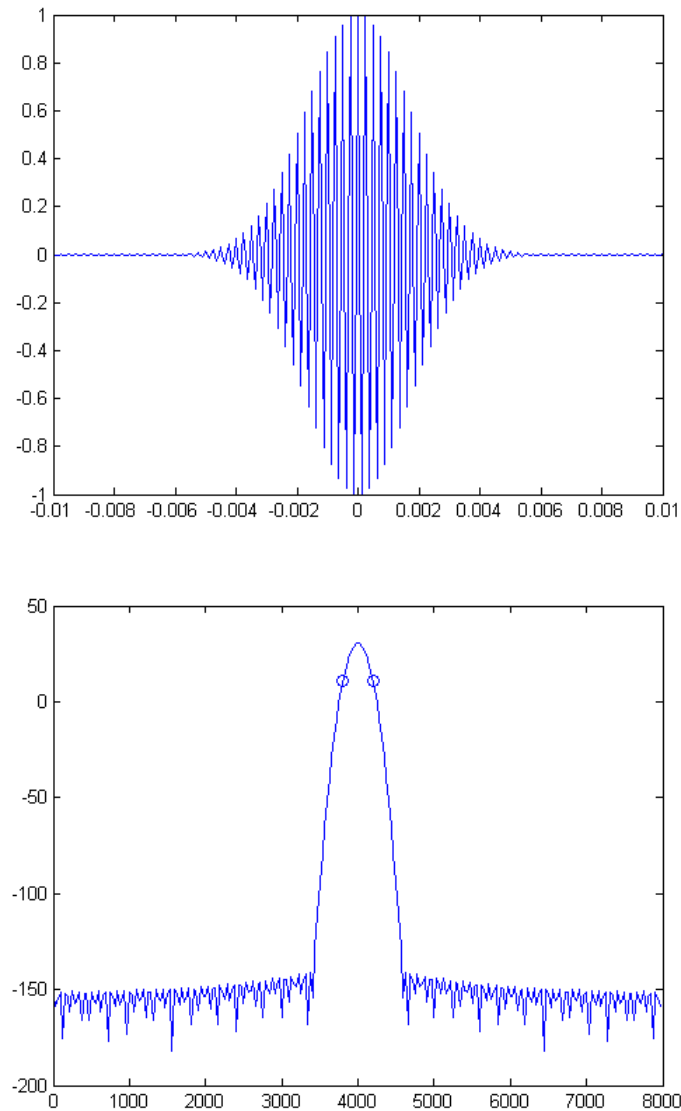


Рисунок 4.9. Гауссов радиоимпульс и его спектр

## 4.5. Генерация последовательности импульсов

Функция `pulstran` служит для генерации конечной последовательности импульсов одинаковой формы с произвольно задаваемыми задержками и уровнями. Импульсы могут задаваться как именем функции, так и уже рассчитанным вектором отсчетов. Сформируем последовательность из пяти симметричных треугольных импульсов, интервалы между которыми линейно увеличиваются, а амплитуды экспоненциально уменьшаются. Частота дискретизации 1 кГц, а длительность - 20 мс.

Листинг 9: Последовательности импульсов

```

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 FFT3 = fft(y);
8 figure;
9 subplot(2, 1, 1); plot(t, y);
10 subplot(2, 1, 2); plot(t, FFT3);
11 %
12 Fs0 = 400;
13 tau = 60e-3;
14 t0 = 0:1/Fs0:tau;
15 s0 = sin(pi*t0/tau).^2;
16 Fs = 1e3;
17 t = 0:1/Fs:0.5;
18 d = (1:6)'*64e-3;
19 d(:,2) = 0.6.^(0:5)';
20 y = pulstran(t, d, s0, Fs0);
21 FFT3 = fft(y);
22 figure;
23 subplot(2, 1, 1); plot(t, y);
24 subplot(2, 1, 2); plot(t, FFT3);

```

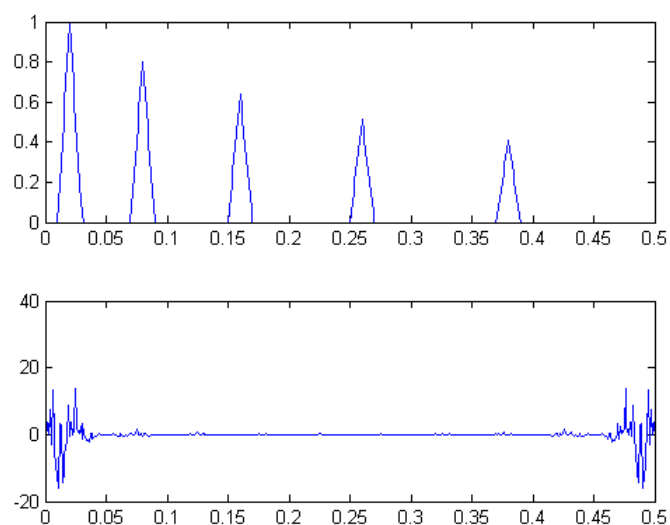


Рисунок 4.10. Последовательность треугольных импульсов и их спектр

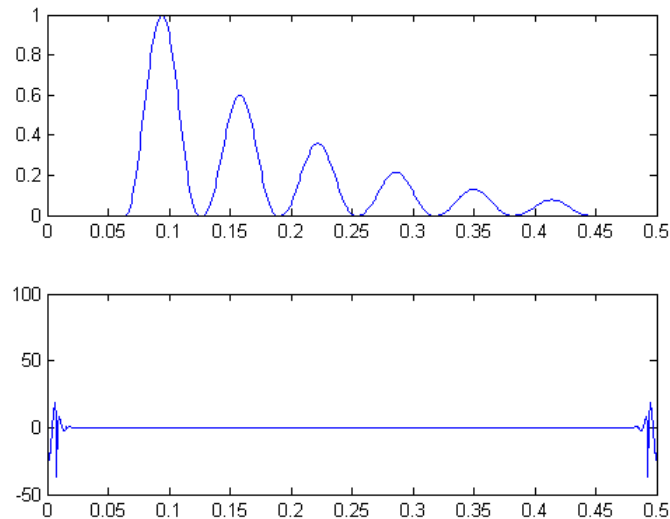


Рисунок 4.11. Последовательность импульсов  $\sin^2$

## 4.6. Функции генерации периодических сигналов

Для формирования периодических сигналов различной формы используются функции: `square` - последовательность прямоугольных импульсов. `sawtooth` - последовательность треугольных импульсов. `diric` - функция Дирихле.

Функция `square`

Листинг 10: Периодический прямоугольный сигнал

```

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 FFT3 = fft(s);
8 figure;
9 subplot(2, 1, 1); plot(t, s);
10 ylim([0, 4])
11 subplot(2, 1, 2); plot(t, FFT3);

```

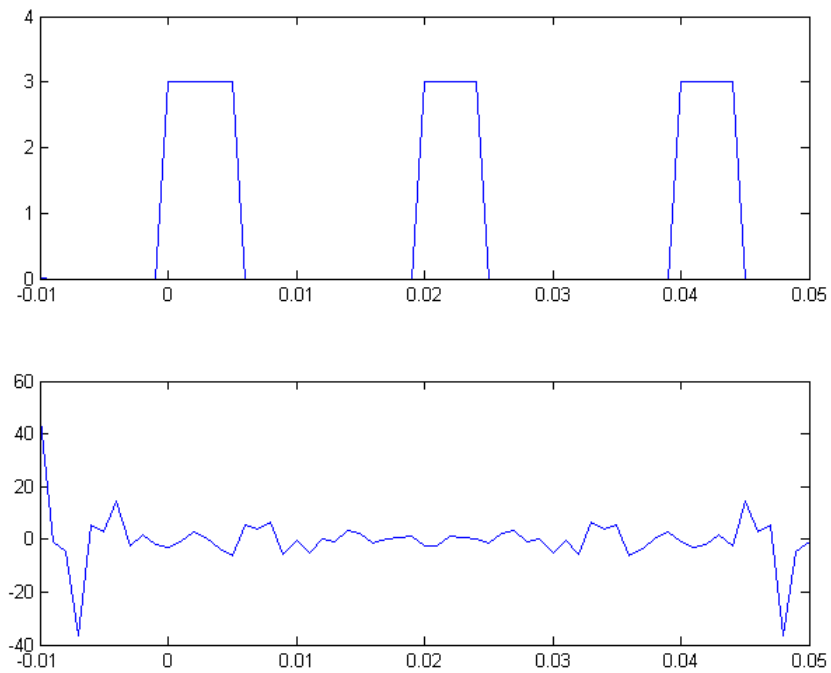


Рисунок 4.12. Периодический прямоугольный сигнал и его спектр

Функция `sawtooth`

Листинг 11: Периодический треугольный сигнал

```

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 FFT3 = fft(s);
8 figure;
9 subplot(2, 1, 1); plot(t, s);
10 %dim([0, 4])
11 subplot(2, 1, 2); plot(t, FFT3);

```

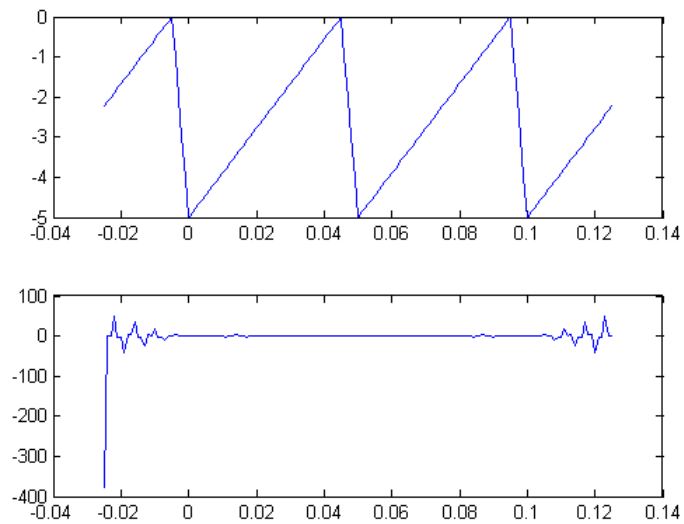


Рисунок 4.13. Периодический треугольный сигнал и его спектр

Функция `diric`

Листинг 12: функция Дирихле

```

1 x = 0:0.01:15;
2 FFT3 = fft(diric(x, 7));
3 figure;
4 subplot(2, 1, 1); plot(x, diric(x, 7));
5 subplot(2, 1, 2); plot(x, FFT3);
6 grid on
7 title('n=7')
8 FFT4 = fft(diric(x, 8));
9 figure;
10 subplot(2, 1, 1); plot(x, diric(x, 8));
11 subplot(2, 1, 2); plot(x, FFT4);
12 grid on
13 title('n=8')

```

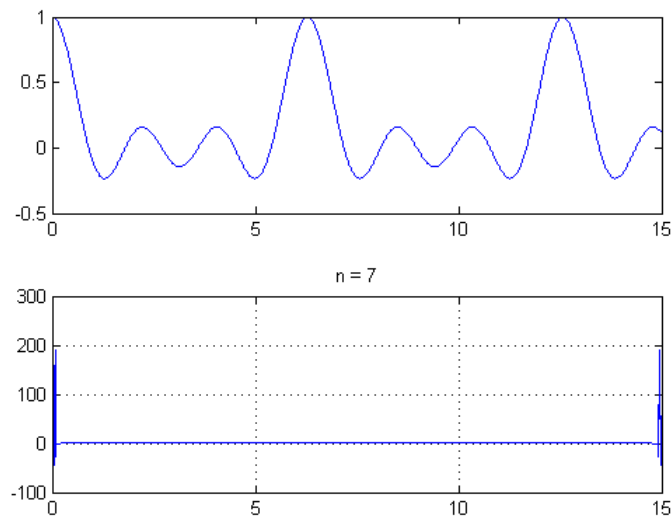


Рисунок 4.14. Функция Дирихле при  $n = 7$  и ее спектр

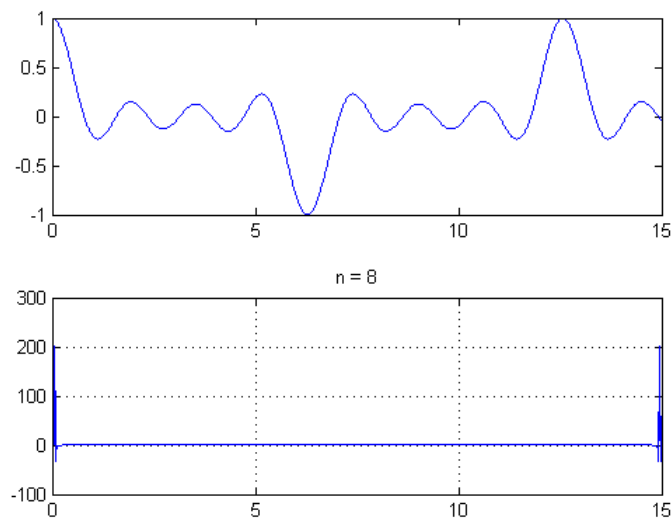


Рисунок 4.15. Функция Дирихле при  $n = 8$  и ее спектр

## 4.7. Генерация сигналов с меняющейся частотой

Для генерации колебаний с единичной амплитудой с разными законами изменения мгновенной частоты используем функцию `chirp`.

Листинг 13: Генерация сигналов с меняющейся частотой

```

1 Fs = 8e3;
2 t = 0:1/Fs:1;
3 t1 = 1;
4 f0 = 1e3;
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

```

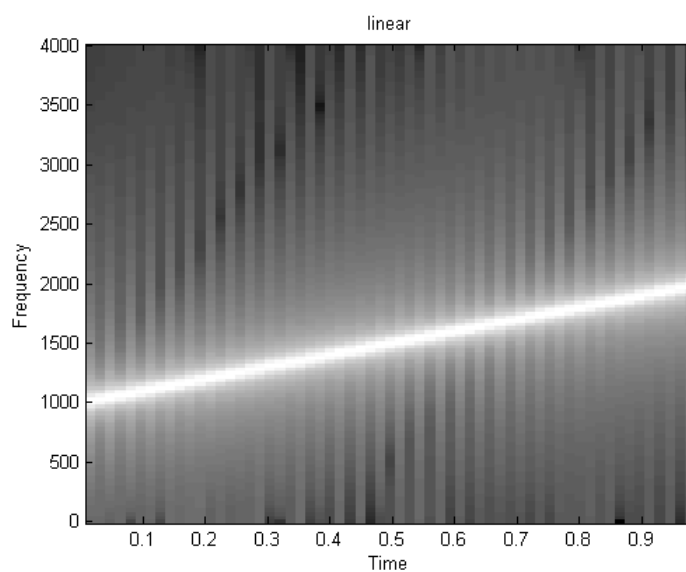


Рисунок 4.16. Линейный закон изменения мгновенной частоты

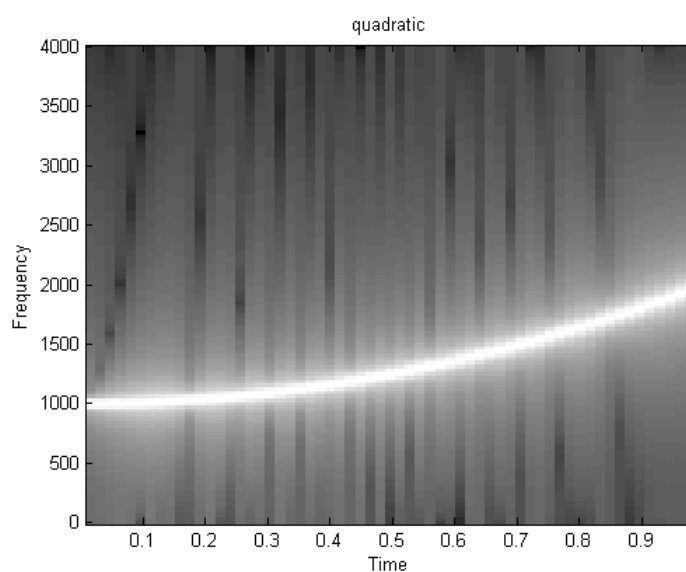


Рисунок 4.17. Квадратичный закон изменения мгновенной частоты

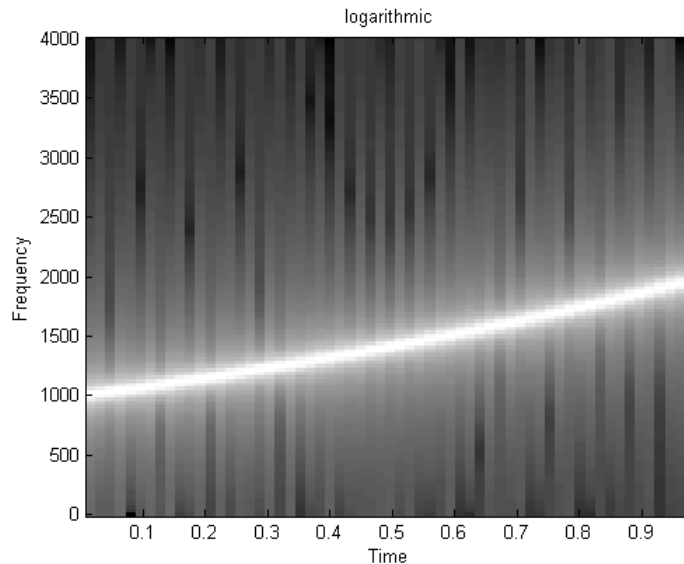


Рисунок 4.18. Экспоненциальный закон изменения мгновенной частоты

#### 4.8. Нахождение посылки в сигнале с помощью методов корреляции

Для сравнения двух методов корреляции, решим следующее задание: найдем посылку [101] в сигнале [0001010111000010].

Листинг 14: Генерация сигналов с меняющейся частотой

```

1 a = [0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0];
2 b = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1];
3 tic
4 for i = 1:length(b)
5     R(i) = sum(a .* b) / length(b);
6     for j = 1:length(yy)-1
7         b(j) = b(j+1);
8     end
9     b(length(b)) = 0;
10 end
11 toc
12
13
14 tic
15 a = fft(a);
16 b = fft(b);
17 a = conj(a);
18 BR = ifft(a .* b)/length(a);
19 toc

```

Оба метода выдали одинаковый ответ, что посылка встречается 2 раза в сигнале. Максимальное значение корреляции (0,1250) соответствует посылке.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0.0625	0	-3.7335e-18	0.0625	0.0625	0.1250	0.0625	0.1250	3.4694e-18	0.1250	-1.0144e-17	0.0625	0	

Рисунок 4.19. Полученный вектор

Временные затраты: Метод быстрой корреляции - 0.000061 секунды Метод прямой корреляции - 0.000457 секунды

## 5. Выводы

В данной работе были получены навыки по визуализации простых сигналов и получению их спектров. В зависимости от того, известен ли сигнал, различают детерминированные и случайные сигналы. Другой класс сигналов - сигналы с интегрируемым квадратом (с ограниченной энергией) и с неограниченной энергией. Другие признаки классификации - периодичность и конечность. Для получения спектра сигнала было произведено разложение в ряд Фурье. Оно используется для модуляции, демодуляции и фильтрации сигналов, с его помощью производится цифровая обработка изображений и звука.