

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

Телекоммуникационные технологии

Отчет по лабораторным работам №1-2

Сигналы телекоммуникационных систем, преобразование Фурье,
корреляция

Работу

выполнил:

Соболь В.О.

Группа: 33501/4

Преподаватель:

Богач Н.В.

Санкт-Петербург
2017

Содержание

1. Цель работы	2
2. Постановка задачи	2
3. Теоретическая информация	2
3.1. Аналоговый сигнал	2
3.2. Затухающие сигналы	2
3.3. Одиночные импульсы	2
3.4. Ограниченная полоса частот	2
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. Взаимная корреляция	4
4. Ход работы	4
4.1. Генерация затухающего гармонического сигнала	4
4.2. Многоканальный сигнал	5
4.3. Кусочные зависимости	6
4.4. Прямоугольный импульс	9
4.5. Трапециевидный импульс	10
4.6. Импульс с ограниченной полосой частот	11
4.7. Гауссов радиоимпульс	12
4.8. Последовательности импульсов	13
4.9. Генерация периодических сигналов	15
4.10. Функция Дирихле	17
4.11. Сигнал с меняющейся частотой	19
4.12. Сравнение методов корреляции	20
5. Выводы	20
6. Листинг	22

1. Цель работы

Целью данной работы является приобретение навыков генерации и визуализации сигналов, построение спектра сигналов разложением этих сигналов в ряд Фурье и определение синхропосылки методом взаимной корреляции.

2. Постановка задачи

Задачей работы является:

1. промоделировать сигналы из Главы 3, сс. 150–170 справочного пособия в Matlab
2. получить их спектр с помощью разложения в ряд Фурье
3. с помощью функции корреляции найти позицию синхропосылки

3. Теоретическая информация

3.1. Аналоговый сигнал

Аналоговый сигнал, с математической точки зрения, представляет собой функцию времени. Для визуализации сигнал необходимо дискретизировать, то есть подставить в функцию вектор дискретных значений времени (временные отсчёты), и получить вектор значений функции в этих отсчётах.

3.2. Затухающие сигналы

Затухание гармонического сигнала получается умножением на убывающую экспоненциальную функцию:

$$S = \exp^{-\alpha t} * s \quad (1)$$

где s — гармонический сигнал

3.3. Одиночные импульсы

Для получения прямоугольных импульсов используется функция

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

t —вектор временных отсчётов, $width$ —ширина (длительность) импульса. В Matlab эта функция называется `rectpuls`.

Для создания треугольных импульсов используется функция

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

$skew$ - коэффициент асимметрии импульса, t —вектор временных отсчётов, $width$ —ширина импульса. В Matlab эта функция называется `tripuls`.

3.4. Ограниченная полоса частот

Для формирования сигнала, с ограниченной полосой частот, используется функция вида:

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

Функция спектра такого сигнала имеет прямоугольный вид:

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

3.5. Гауссов радиоимпульс

Функция для получения радиоимпульса имеет вид:

$$y = \exp(-\alpha t^2) \cos(2\pi f t) \quad \alpha = -\frac{5(2\pi f * bw)^2}{bwr * \log 10} \quad (6)$$

где t — временные отсчёты, f — несущая частота, bw — относительная ширина спектра, bwr — уровень в децибелах, по которому производится измерение ширины спектра.

Спектр такого сигнала можно получить с помощью преобразования Фурье.

3.6. Функция Дирихле

Функция Дирихле описывается формулой:

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

В этой функции n — целое положительное число и определяет порядок.

Функцию Дирихле это периодическая sinc функция, которая имеет вид:

при чётном значении n

$$diric_n(x) = \sum_{k=-\infty}^{\infty} sinc\left(n\left(\frac{t}{2\pi} - k\right)\right) \quad (8)$$

при нечётном значении n

$$diric_n(x) = \sum_{k=-\infty}^{\infty} (-1)^k sinc\left(n\left(\frac{t}{2\pi} - k\right)\right) \quad (9)$$

3.7. Математические законы изменения мгновенной частоты

В данной работе рассматриваются 3 закона:

линейный

$$f(t) = f_0 + \beta t \quad \beta = \frac{f_1 - f_0}{t_1} \quad (10)$$

квадратичный

$$f(t) = f_0 + \beta t^2 \quad \beta = \frac{f_1 - f_0}{t_1^2} \quad (11)$$

логарифмический

$$f(t) = f_0 + \exp(\beta t) \quad \beta = \frac{\ln(f_1 - f_0)}{t_1} \quad (12)$$

Логарифмический закон противоречит своему названию, т.к. зависимость частоты от времени в нем экспоненциальная.

3.8. Преобразование Фурье

Для нахождения спектра сигнала чаще всего применяют преобразование Фурье. Формула прямого преобразования Фурье выглядит следующим образом:

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt \quad (13)$$

Обратное преобразование Фурье строится по следующей формуле:

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{j\omega t} d\omega \quad (14)$$

3.9. Взаимная корреляция

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

$$R = \frac{1}{N} \sum_{i=1}^N x_i * y_i \quad (15)$$

Если искомая посылка у 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 для больших посылок, применяется метод быстрой корреляции:

$$R = \frac{1}{N} F_D^{-1} [X^* * Y] \quad (16)$$

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

4.1. Генерация затухающего гармонического сигнала

Затухающий гармонический сигнал и его спектр получены с помощью кода из листинга 1.

На графике 4.1.1 показаны различные формы представления затухающего гармонического сигнала, построенного по дискретным отсчетам. На первом графике обычный вид, на втором — точечное представление, на третьем точки представлены как отклонения от нуля и на четвёртом — ступенчатый график.

Спектр представленного выше сигнала (рис. 4.1.2) получен с помощью разложения в ряд Фурье.

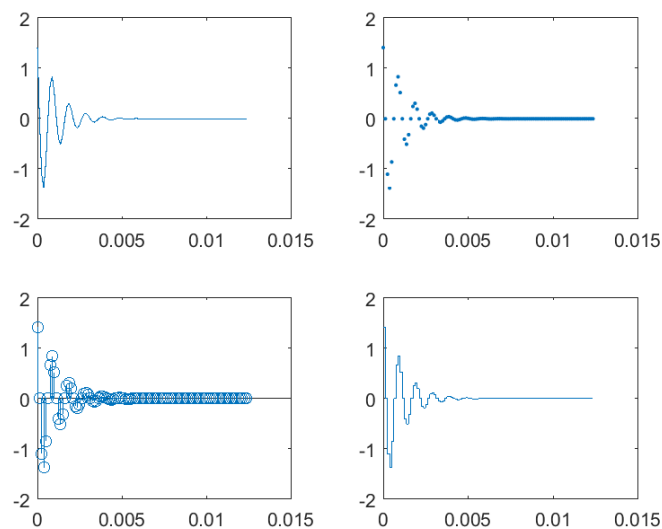


Рис. 4.1.1. Графики сигнала

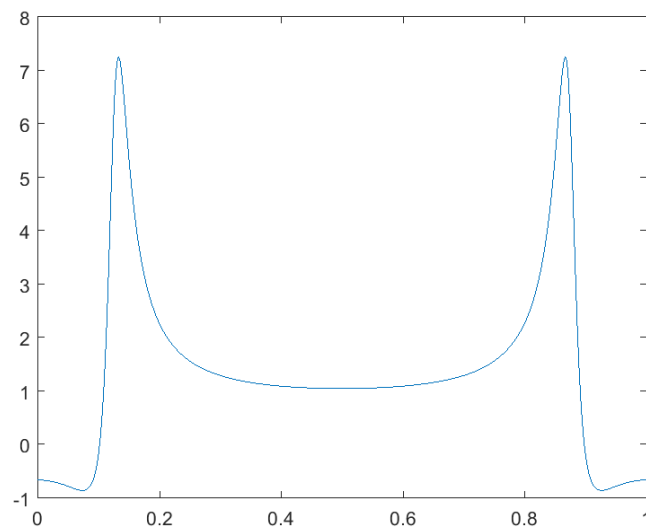


Рис. 4.1.2. Спектр сигнала

4.2. Многоканальный сигнал

Несколько сигналов, различающихся по частоте (рис. 4.2.1) получены с помощью кода из листинга 2. Их спектр приведен на рис.4.2.2.

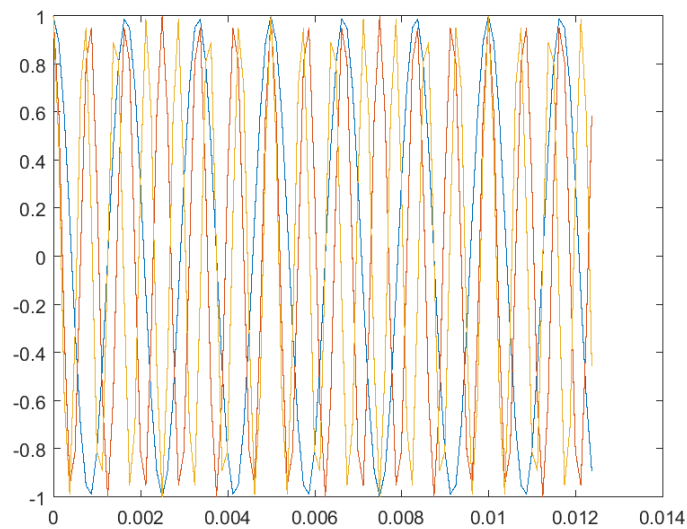


Рис. 4.2.1. График сигналов

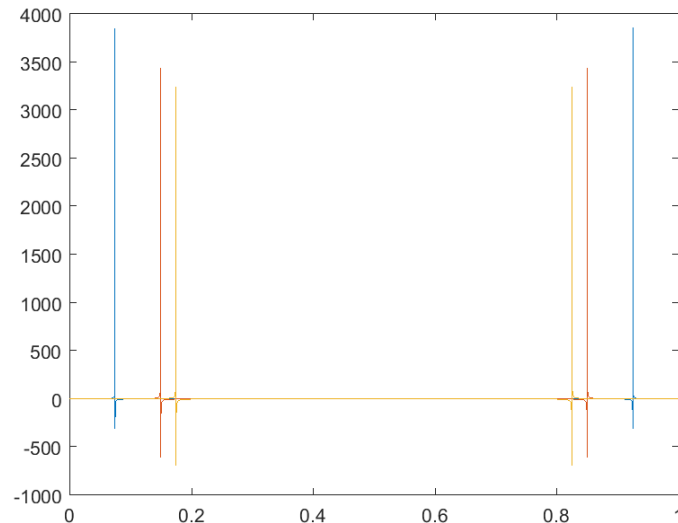


Рис. 4.2.2. Спектр сигналов

4.3. Кусочные зависимости

Односторонний экспоненциальный импульс (рис. 4.3.1), прямоугольный импульс (рис. 4.3.2) и несимметричный треугольный импульс (рис. 4.3.3) получены с помощью кода из листинга 3 согласно заданным уравнениям. Спектры сигналов представлены на рисунках 4.3.1 — 4.3.3.

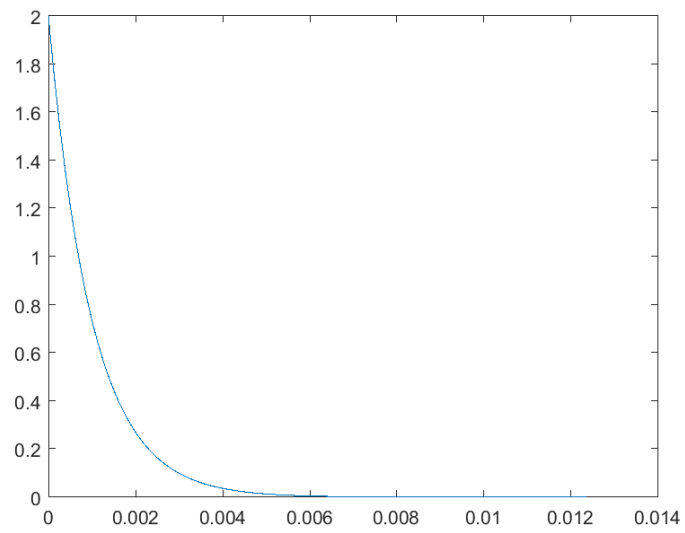


Рис. 4.3.1. Экспоненциальный импульс

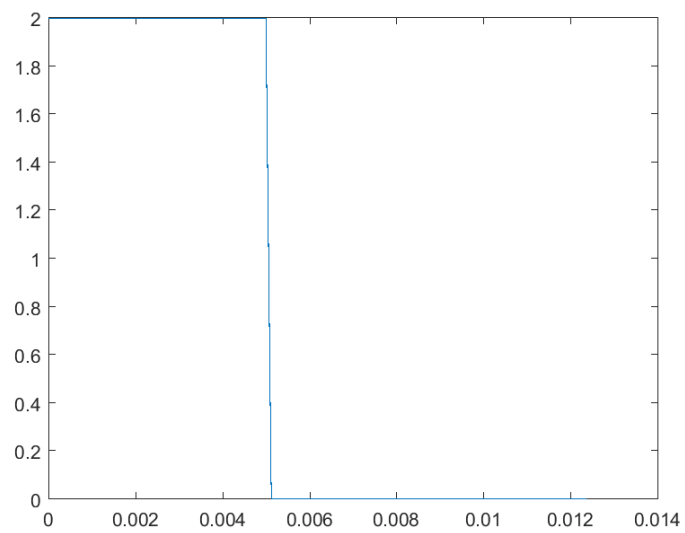


Рис. 4.3.2. Прямоугольный импульс

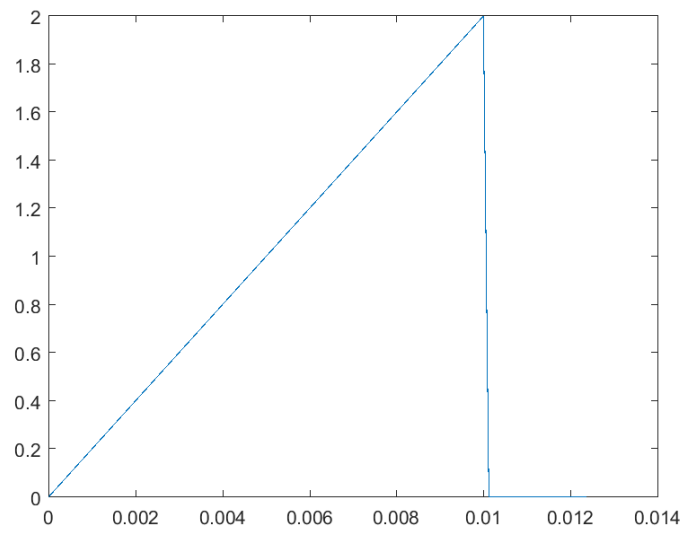


Рис. 4.3.3. Несимметричный треугольный импульс

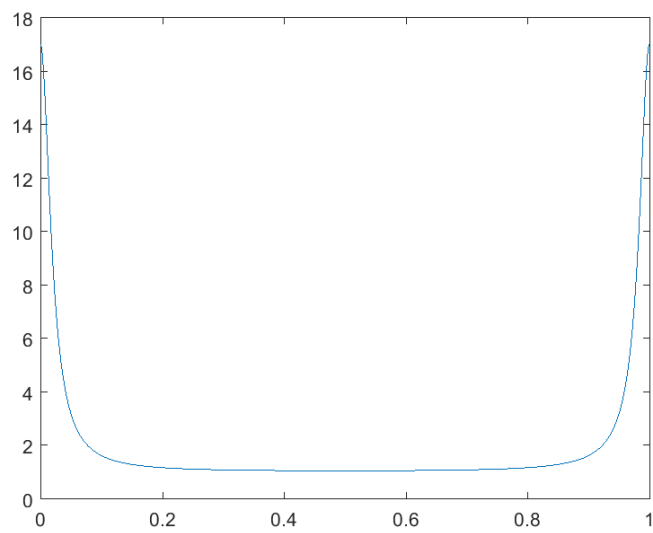


Рис. 4.3.4. Спектр экспоненциального импульса

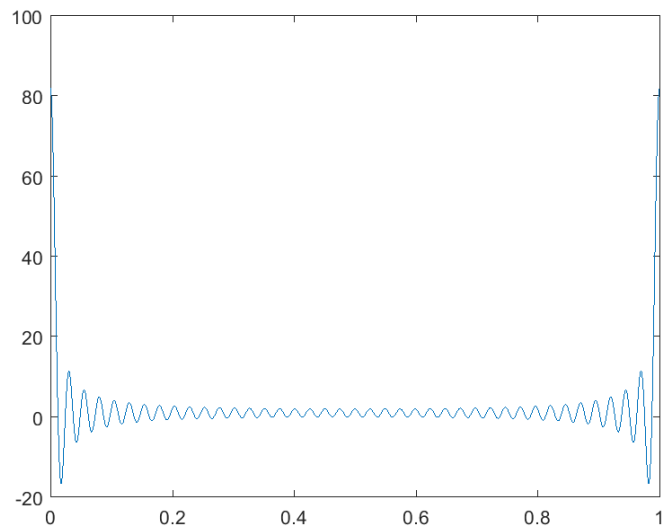


Рис. 4.3.5. Спектр прямоугольного импульса

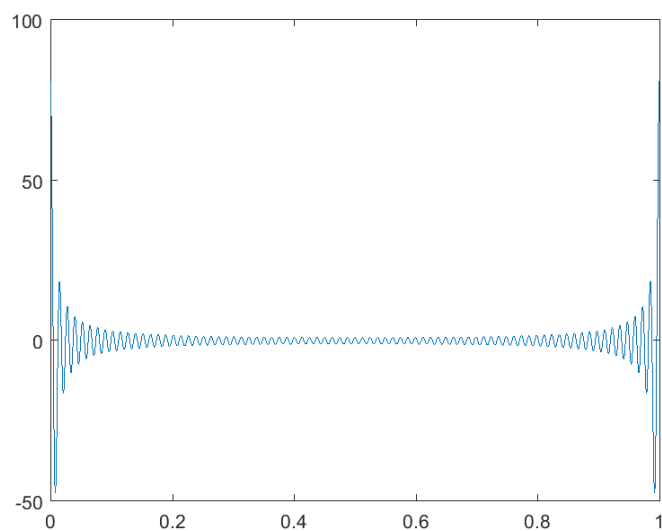


Рис. 4.3.6. Спектр несимметричного треугольного импульса

4.4. Прямоугольный импульс

Сигнал (рис. 4.4.1) генерируется путем соединения двух прямоугольных импульсов, с использованием встроенной функции `rectpuls` (листинг 4). Спектр (рис. 4.4.2) прямоугольного импульса получен разложением сигнала в ряд Фурье.

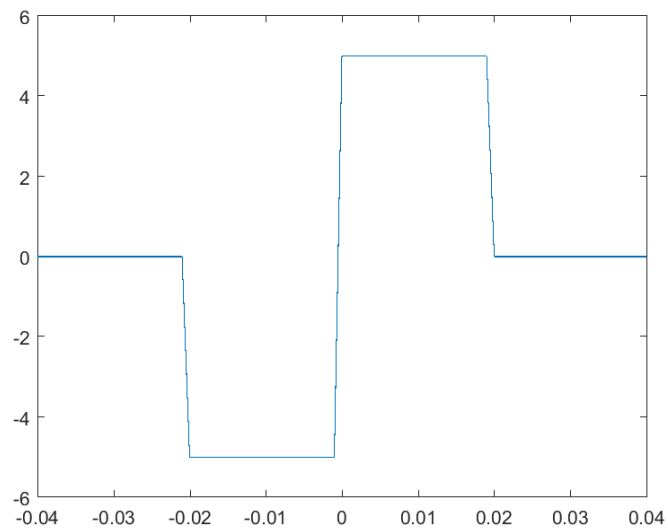


Рис. 4.4.1. Прямоугольные импульсы

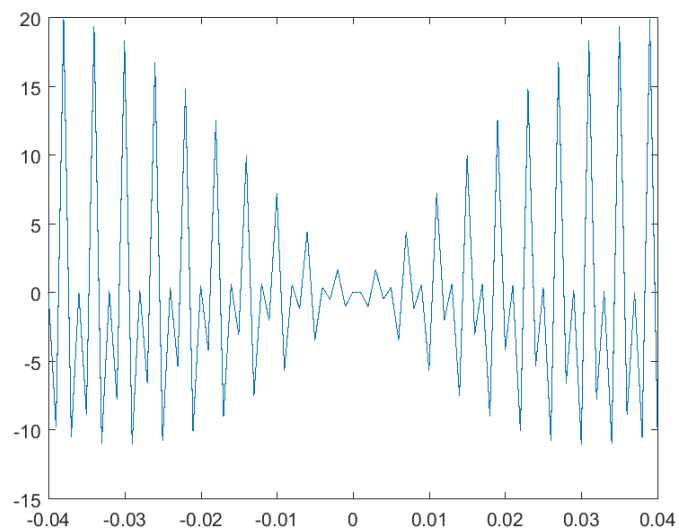


Рис. 4.4.2. Спектр прямоугольных импульсов

4.5. Трапециевидный импульс

Трапециевидный сигнал (рис. 4.5.1) генерируется разностью двух треугольных импульсов, с использованием встроенной функции `tripuls` (листинг 5).

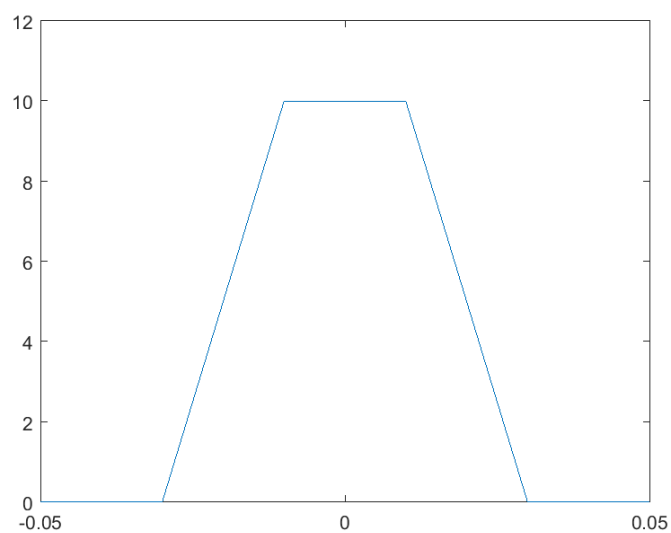


Рис. 4.5.1. Трапециевидный импульс

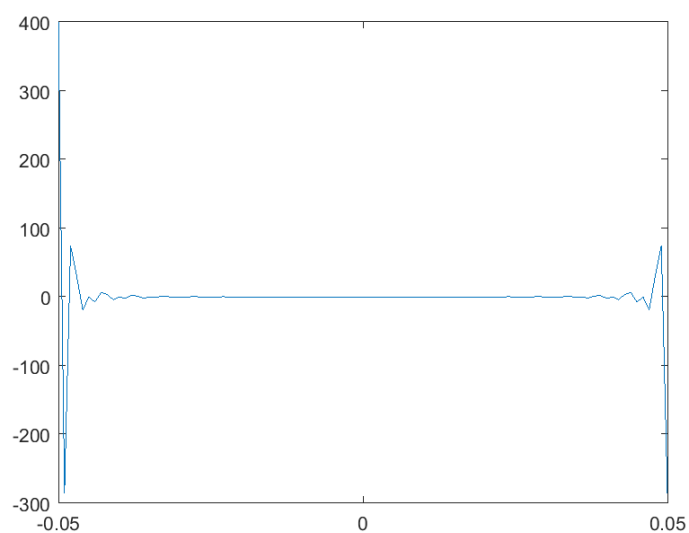


Рис. 4.5.2. Спектр трапециевидного импульса

4.6. Импульс с ограниченной полосой частот

Сигнал, у которого спектр ограничен по частоте (рис. 4.6.1), получен с помощью программы из листинга 6. Спектр сигнала (рис. 4.6.2) получен с помощью функции `sinc`.

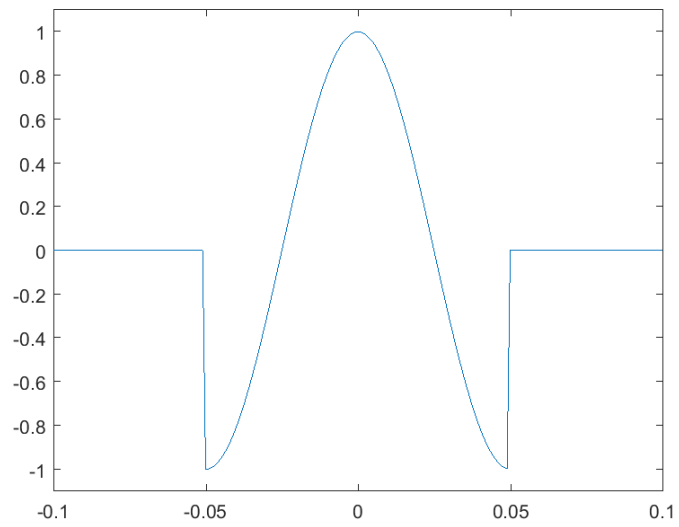


Рис. 4.6.1. Сигнал с ограниченным спектром

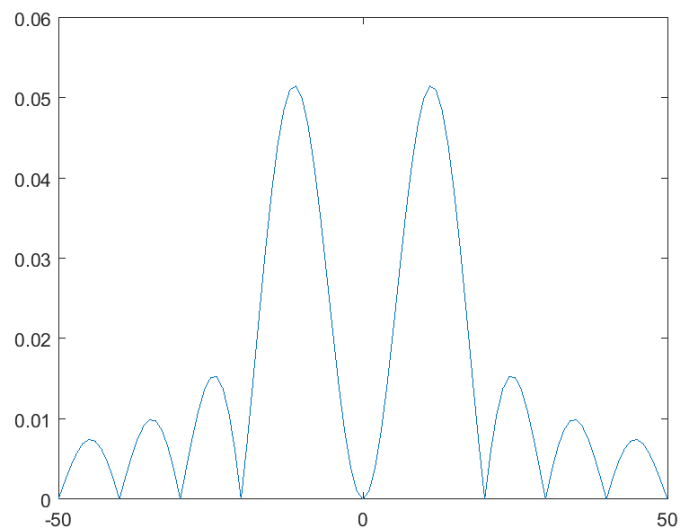


Рис. 4.6.2. Спектр сигнала с ограниченным спектром

4.7. Гауссов радиопульс

Гауссов радиопульс получен с помощью встроенной функции `gauspuls` (7).

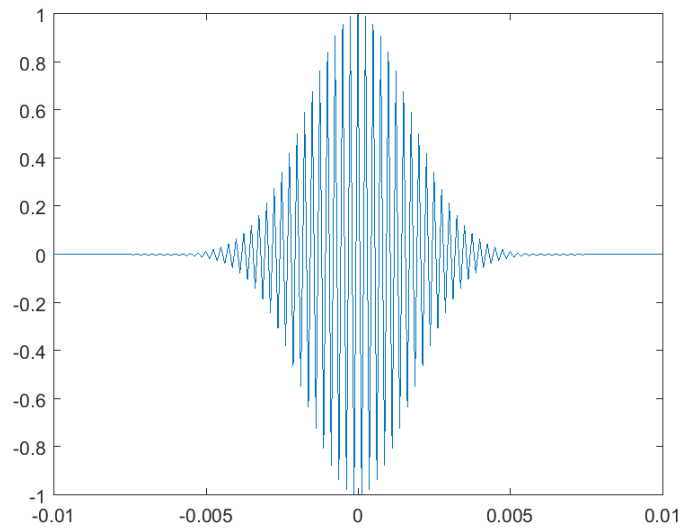


Рис. 4.7.1. Гауссов радиоимпульс

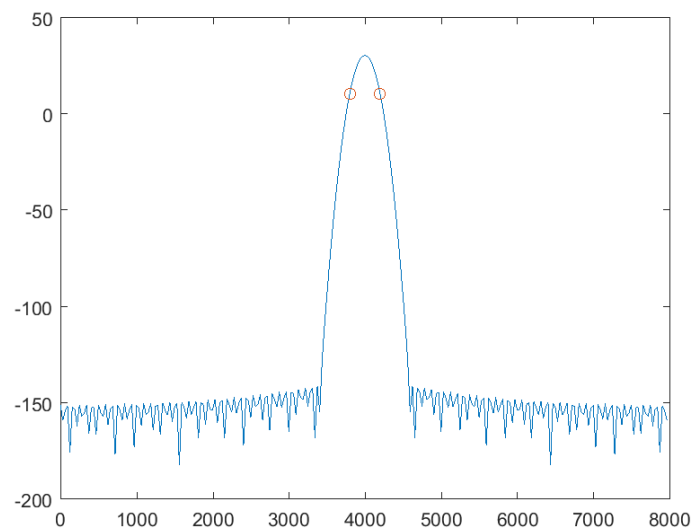


Рис. 4.7.2. Амплитудный спектр радиоимпульса

На графике спектра (рис. 4.7.2) также отмечены расчетные границы этого спектра.

4.8. Последовательности импульсов

Треугольные импульсы (рис. 4.8.1), получены функцией `pulstran` (листинг 8), которая генерирует треугольные импульсы с заданными амплитудами и через заданные промежутки времени. Спектр этого сигнала показан на рис. 4.8.2.

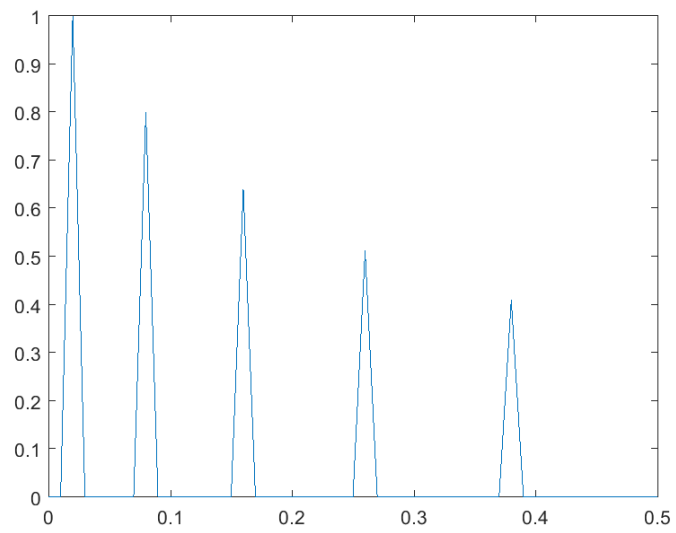


Рис. 4.8.1. Треугольные импульсы

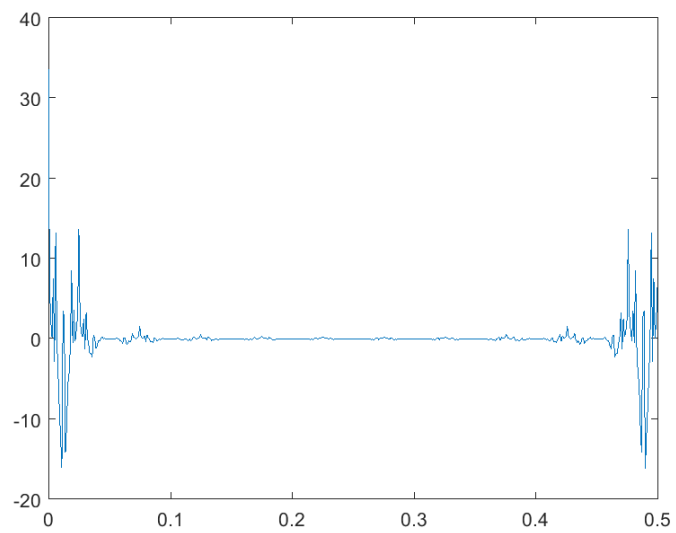


Рис. 4.8.2. Спектр треугольных импульсов

Гармонические импульсы получены функцией `pulstran` (листинг 9) из вектора отсчетов одиночного импульса.

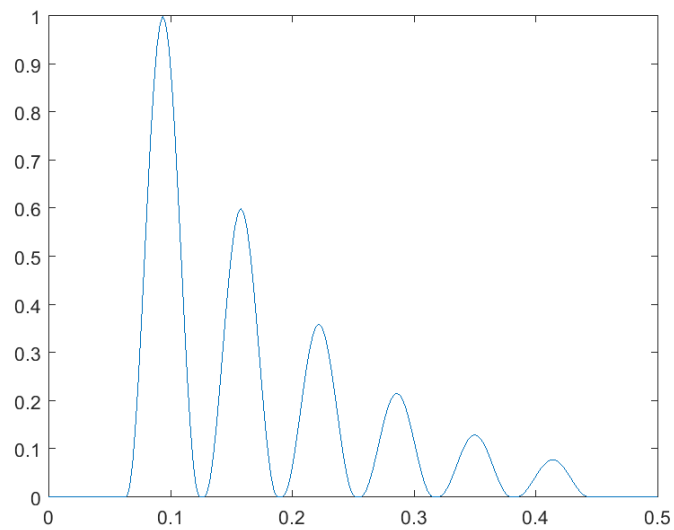


Рис. 4.8.3. Гармонические импульсы

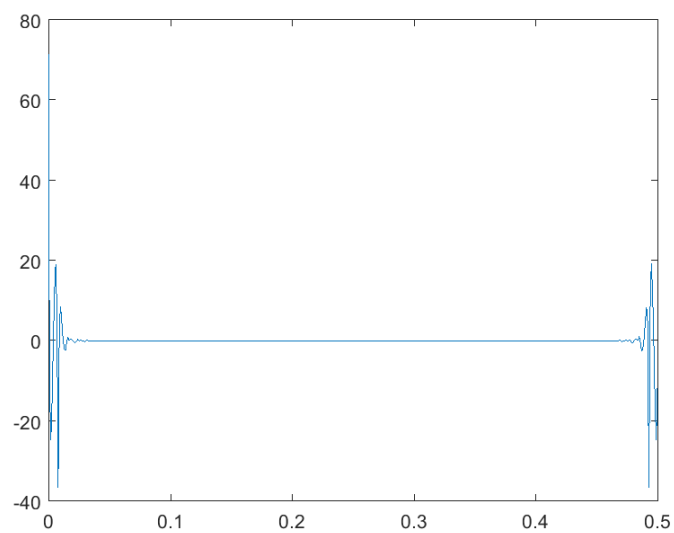


Рис. 4.8.4. Спектр гармонических импульсов

4.9. Генерация периодических сигналов

Периодически повторяющиеся прямоугольные сигналы, созданы с помощью функции `square` (листинг 10).

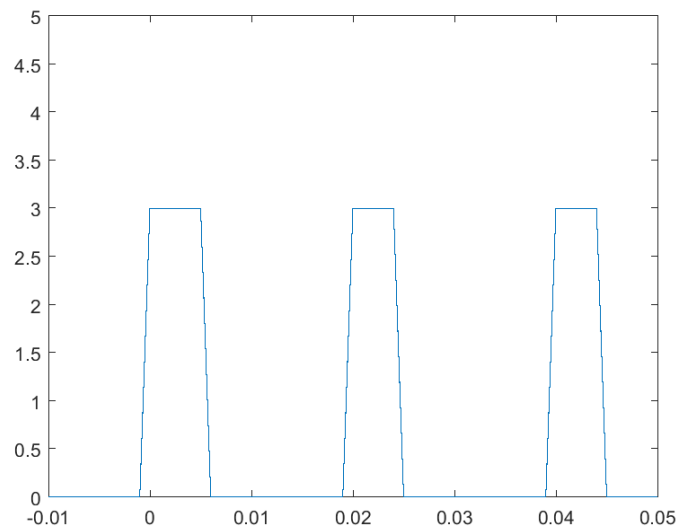


Рис. 4.9.1. Периодические прямоугольные импульсы

Импульсы обладают одинаковой длительностью и временем паузы между ними.

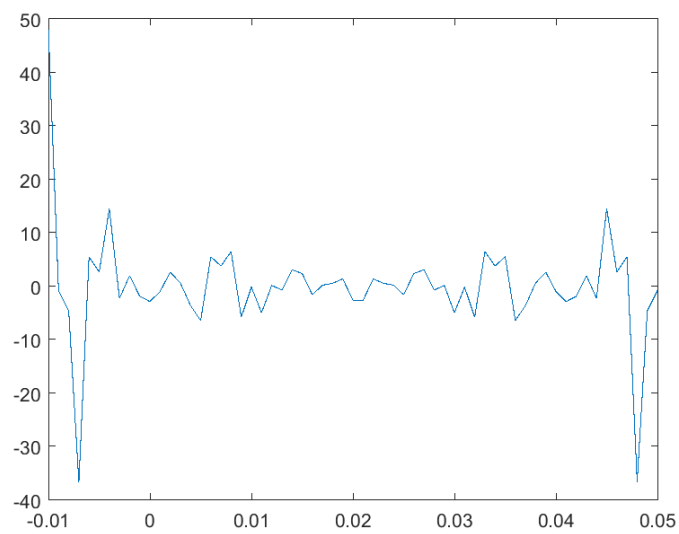


Рис. 4.9.2. Спектр прямоугольных импульсов

Импульсы треугольной формы с заданными параметрами получены с помощью функции `sawtooth` (листинг 11).

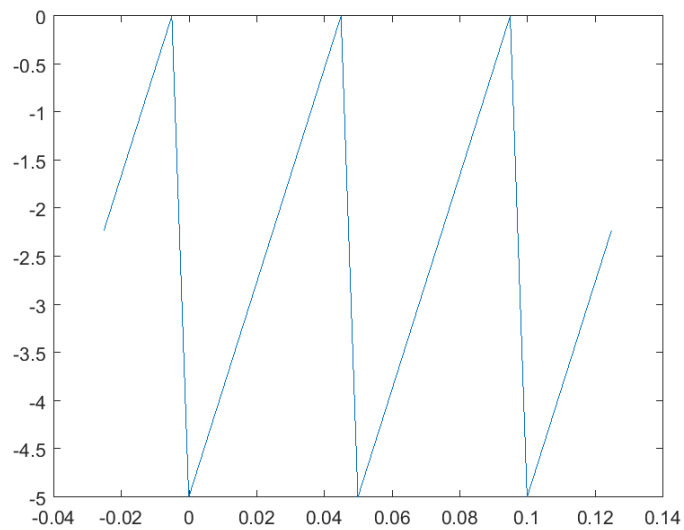


Рис. 4.9.3. Треугольные импульсы

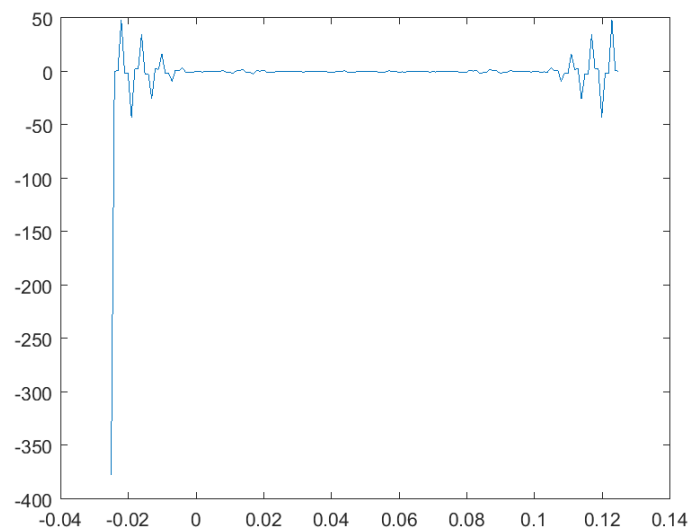


Рис. 4.9.4. Спектр треугольных импульсов

4.10. Функция Дирихле

Для создания выборки из функции Дирихле с четным и нечетным значением параметра использована функция `diric` (листинг 12).

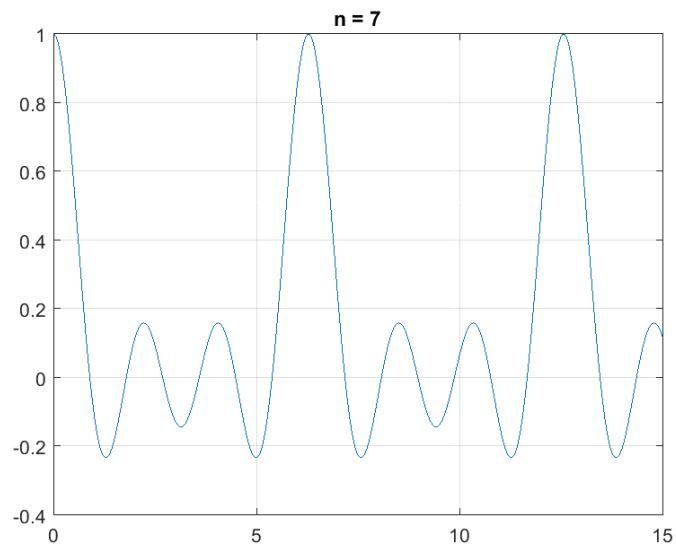


Рис. 4.10.1. Функция Дирихле с параметром равным 7

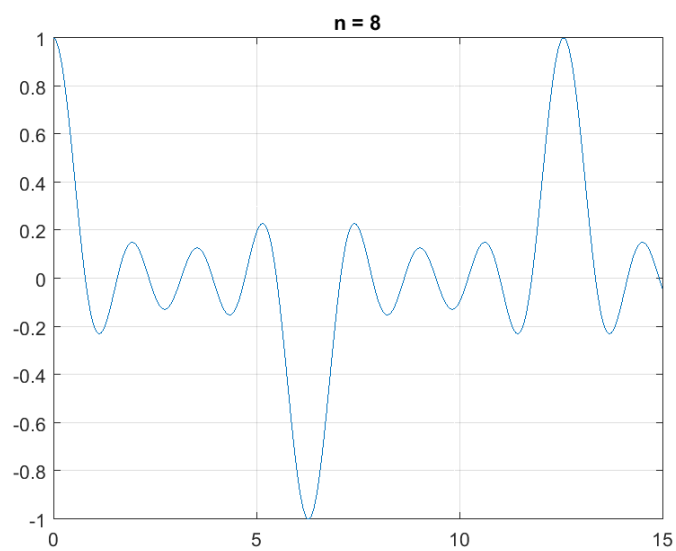


Рис. 4.10.2. Функция Дирихле с параметром равным 8

Видно, что нечетный параметр обеспечивает однонаправленные импульсы, а частота колебаний растёт с увеличением значения параметра.

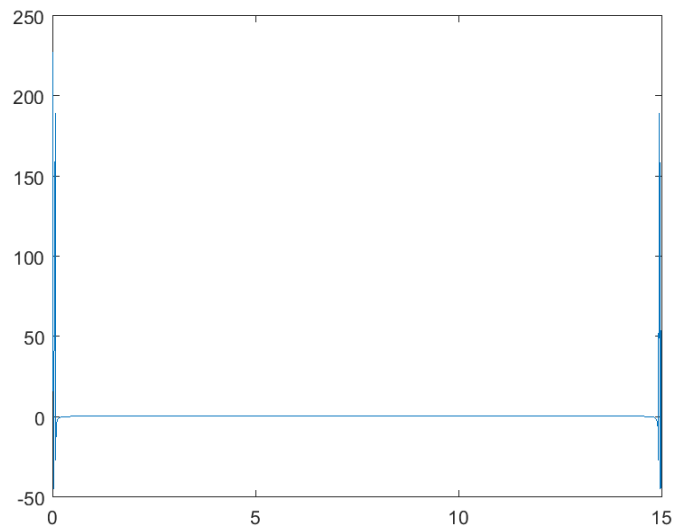


Рис. 4.10.3. Спектр функции Дирихле с параметром равным 7

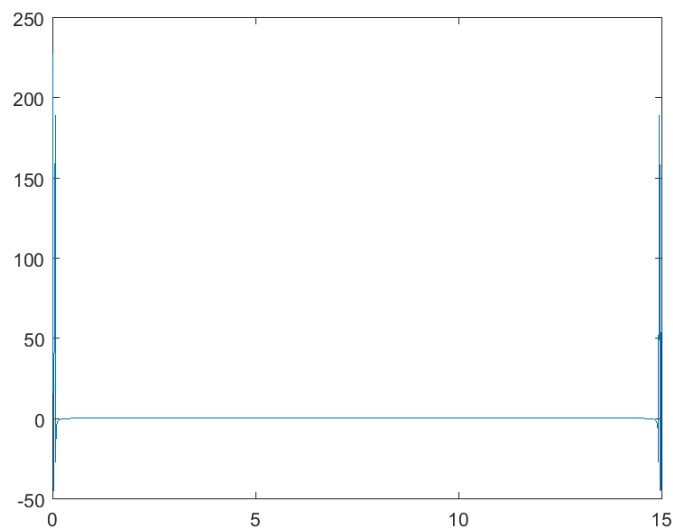


Рис. 4.10.4. Спектр функции Дирихле с параметром равным 8

4.11. Сигнал с меняющейся частотой

Программа, написанная для исследования (листинг 13), 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 рисунках 4.11.2, 4.11.2 и 4.11.3 показаны спектрограммы этих сигналов — зависимость мгновенного амплитудного спектра от времени.

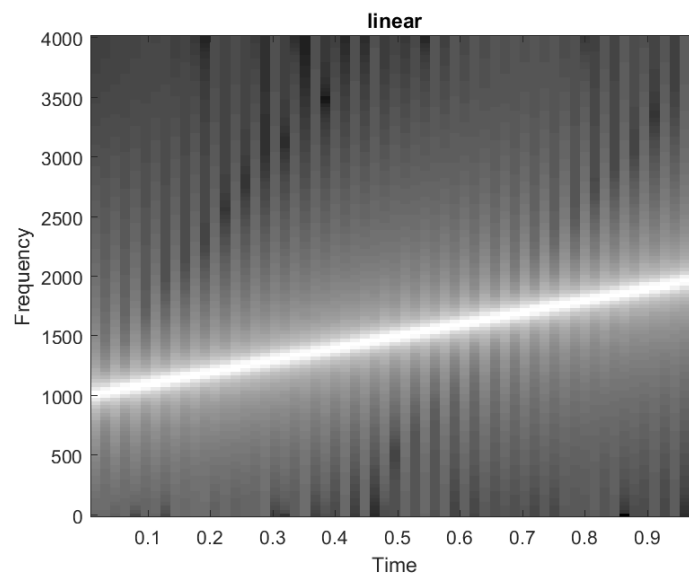


Рис. 4.11.1. Спектрограмма линейной функции chirp

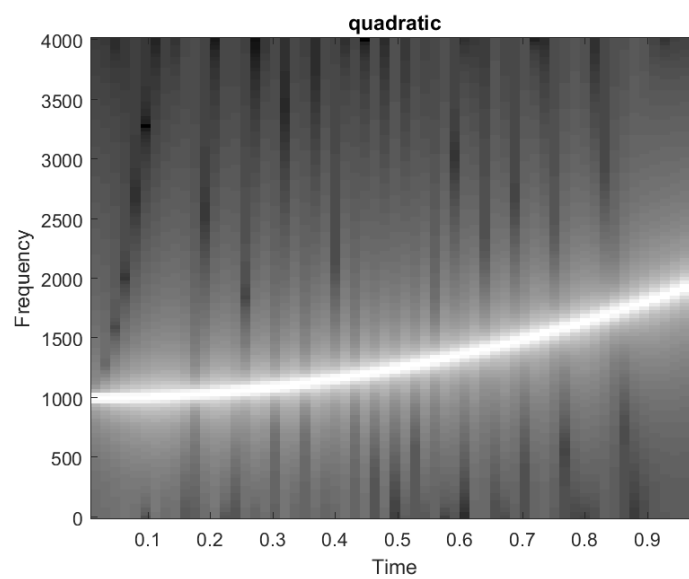


Рис. 4.11.2. Спектрограмма квадратичной функции chirp

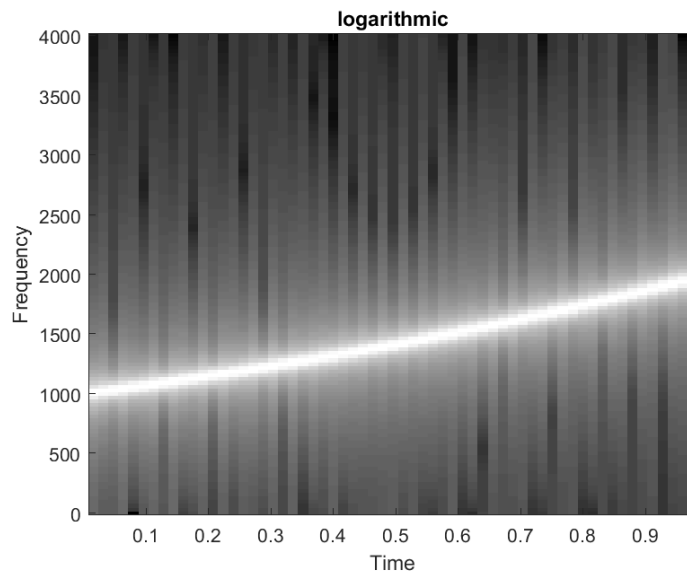


Рис. 4.11.3. Спектрограмма логарифмической функции chirp

На приведенных спектрограммах видно, что характер изменения мгновенной частоты сигнала совпадает с выбранной функцией.

4.12. Сравнение методов корреляции

Код использованный при сравнении алгоритмов находится в листинге 14. В качестве исходного примера была взята задача нахождения синхропосылки 101 в сигнале 0001010111000010. Перед началом вычисления корреляции синхропосылка была изменена - 1 0 1 на 1 -1 1 для ее более надежного нахождения в посылке, и 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 синхропосылка была найдена в сигнале 2 раза - по смещению +3 и +5. Первый алгоритм показал время выполнения - 0,035 мс, в то время как второй - 0,017 мс. Можно сделать вывод, что алгоритм быстрой корреляции эффективней обычного. Разница не значительна из-за малой длины посылки.

5. Выводы

В данной работе исследованы методы получения и визуализации различных сигналов в среде Matlab.

Рассмотрены различные виды сигналов - детерминированные, периодические, конечные (финитные) и бесконечные, гармонические колебания и сигналы, полученные на их основе, единичные импульсы различной формы.

Получены спектры сигналов с помощью преобразования Фурье. Преобразование Фурье - одна из фундаментальных операций в телекоммуникационных технологиях, т.к. с его помощью можно быстро получать спектры сигналов для их анализа и модификации.

Были опробованы 2 метода подсчета корреляции на простом примере. Стоит отметить, на таком коротком примере алгоритм быстрой корреляции оказался всего в 2 раза быстрее обычного алгоритма. Его эффективность вырастет во много раз, по сравнению с обычным алгоритмом на больших посылках.

6. Листинг

Листинг 1: Затухающий гармонический сигнал

```
1 close all
2 clc
3 Fs = 8e3;
4 t = 0:1/Fs:1;
5 t = t';
6 A_ = 2;
7 f0_ = 1e3;
8 phi_ = pi/4;
9 s1_ = A_ * cos(2*pi*f0_*t_ + phi_);
10 alpha_ = 1e3;
11 s2_ = exp(-alpha_*t_) .* s1_;
12 FFT_ = fft(s2_);
13 g_ = figure();
14 subplot(2,2,1); plot(t(1:100), s2(1:100))
15 subplot(2,2,2); plot(t(1:100), s2(1:100), 'r')
16 subplot(2,2,3); stem(t(1:100), s2(1:100))
17 subplot(2,2,4); stairs(t(1:100), s2(1:100))
18 s_ = figure();
19 plot(t(1:Fs+1), FFT(1:Fs+1))
20
21 saveas(g, '../fig/graph1', 'png');
22 saveas(s, '../fig/spec1', 'png');
```

Листинг 2: Сигналы, различающиеся по частоте

```
1 close all
2 clc
3 Fs = 8e3;
4 t = 0:1/Fs:1;
5 t = t';
6 f_ = [600_1200_1400];
7 s3_ = cos(2*pi*t*f_);
8 FFT_ = fft(s3_);
9 g_ = figure();
10 plot(t(1:100), s3(1:100,:))
11 s_ = figure();
12 plot(t(1:Fs+1), FFT(1:Fs+1,:))
13
14 saveas(g, '../fig/graph2', 'png');
15 saveas(s, '../fig/spec2', 'png');
```

Листинг 3: Кусочные сигналы

```
1 close all
2 clc
3 Fs = 8e3;
4 t = 0:1/Fs:1;
5 t = t';
6 A_ = 2;
7 f0_ = 1e3;
8 phi_ = pi/4;
9 alpha_ = 1e3;
10 T_ = 0.01;
11 s1_ = A_ * exp(-alpha_*t_) .* (t_ >= 0);
12 FFT1_ = fft(s1_);
13 s2_ = A_ * (abs(t) <= T/2);
```

```

14 FFT2 = fft(s2);
15 s3 = A * t ./ T .* (T >= 0) .* (t <= T);
16 FFT3 = fft(s3);
17 g1 = figure();
18 plot(t(1:100), s1(1:100))
19 g2 = figure();
20 plot(t(1:100), s2(1:100))
21 g3 = figure();
22 plot(t(1:100), s3(1:100))
23 s1 = figure();
24 plot(t(1:Fs+1), FFT1(1:Fs+1))
25 s2 = figure();
26 plot(t(1:Fs+1), FFT2(1:Fs+1))
27 s3 = figure();
28 plot(t(1:Fs+1), FFT3(1:Fs+1))
29
30 saveas(g1, '.. / fig/graph3_1', 'png');
31 saveas(g2, '.. / fig/graph3_2', 'png');
32 saveas(g3, '.. / fig/graph3_3', 'png');
33 saveas(s1, '.. / fig/spec3_1', 'png');
34 saveas(s2, '.. / fig/spec3_2', 'png');
35 saveas(s3, '.. / fig/spec3_3', 'png');

```

Листинг 4: Прямоугольный сигнал

```

1 close all
2 clc
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 g = figure();
9 plot(t, s)
10 ylim([-6 6]);
11 FFT = fft(s);
12 s = figure();
13 plot(t(1:81), FFT(1:81))
14
15 saveas(g, '.. / fig/graph4', 'png');
16 saveas(s, '.. / fig/spec4', 'png');

```

Листинг 5: Трапециевидный сигнал

```

1 close all
2 clc
3 Fs = 1e3;
4 t = -50e-3:1/Fs:50e-3;
5 A = 10;
6 T1 = 20e-3;
7 T2 = 60e-3;
8 s = A * (T2 * tripuls(t, T2) - T1 * tripuls(t, T1)) / (T2 - T1);
9 g = figure();
10 plot(t, s);
11 FFT = fft(s);
12 s = figure();
13 plot(t(1:101), FFT(1:101))
14
15 saveas(g, '.. / fig/graph5', 'png');
16 saveas(s, '.. / fig/spec5', 'png');

```


Листинг 6: Импульс с ограниченной полосой частот

```

1 close all
2 clc
3
4 Fs = 1e3;
5 t = -0.1:1/Fs:0.1;
6 f0 = 10;
7 T = 1/f0;
8 s = rectpuls(t,T) .* cos (2*pi*f0*t);
9 f = -50:50;
10 sp = T/2 * (sinc((f-f0)*T) + sinc((f+f0)*T));
11 g1 = figure();
12 plot(t,s)
13 ylim([-1.1 1.1]);
14 g2 = figure();
15 plot(f, abs(sp))
16
17 saveas(g1, '..\fig/graph6_1', 'png');
18 saveas(g2, '..\fig/graph6_2', 'png');

```

Листинг 7: 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 = 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 g1 = figure();
12 plot(t,s);
13 g2 = figure();
14 plot(f(1:Nfft/2), sp_dB(1:Nfft/2))
15 sp_max_db = 20*log10(max(abs(sp)));
16 edges = Fc * [1-bw/2 1+bw/2];
17 hold on;
18 plot(edges, sp_max_db([1 1])+bwr, 'o');
19 hold off;
20
21 saveas(g1, '..\fig/graph7_1', 'png');
22 saveas(g2, '..\fig/graph7_2', 'png');

```

Листинг 8: Треугольные импульсы

```

1 close all
2 clc
3 Fs = 1e3;
4 t = 0:1/Fs:0.5;
5 tau = 20e-3;
6 d = [20 80 160 260 380]' * 1e-3;
7 d(:,2) = 0.8.^(0:4)';
8 y = pulstran(t, d, 'tripuls', tau);
9 g = figure();
10 plot(t,y)
11 FFT = fft(y);
12 s = figure();
13 plot(t(1:501), FFT(1:501))

```

```

14
15 saveas(g, '../fig/graph8_1', 'png');
16 saveas(s, '../fig/spec8_1', 'png');

```

Листинг 9: Гармонические импульсы

```

1 close all
2 clc
3 Fs0 = 400;
4 tau = 60e-3;
5 t0 = 0:1/Fs0:tau;
6 s0 = sin(pi*t0/tau).^2;
7 Fs = 1e3;
8 t = 0:1/Fs:0.5;
9 d = (1:6) * 0.64e-3;
10 d(:,2) = 0.6.^(0:5);
11 y = pulstran(t, d, s0, Fs0);
12 g = figure();
13 plot(t,y)
14 FFT = fft(y);
15 s = figure();
16 plot(t(1:501), FFT(1:501))
17
18 saveas(g, '../fig/graph8_2', 'png');
19 saveas(s, '../fig/spec8_2', 'png');

```

Листинг 10: Код для генерации периодического прямоугольного сигнала

```

1 close all
2 clc
3 Fs = 1e3;
4 t = -10e-3:1/Fs:50e-3;
5 A = 3;
6 f0 = 50;
7 tau = 5e-3;
8 s = (square(2*pi*t*f0, f0*tau*100) + 1) * A/2;
9 g = figure();
10 plot(t, s)
11 ylim([0 5]);
12 FFT = fft(s);
13 s = figure();
14 plot(t(1:61), FFT(1:61))
15
16 saveas(g, '../fig/graph9_1', 'png');
17 saveas(s, '../fig/spec9_1', 'png');

```

Листинг 11: Код для генерации треугольных импульсов

```

1 close all
2 clc
3 Fs = 1e3;
4 t = -25e-3:1/Fs:125e-3;
5 A = 5;
6 T = 50e-3;
7 t1 = 5e-3;
8 s = (sawtooth(2*pi*t/T, 1-t1/T) - 1) * A/2;
9 g = figure();
10 plot(t,s);
11 FFT = fft(s);
12 s = figure();
13 plot(t, FFT)

```

```

14
15 saveas(g, '../fig/graph9_2','png');
16 saveas(s, '../fig/spec9_2','png');

```

Листинг 12: Функция Дирихле

```

1 close all
2 clc
3 x = 0:0.01:15;
4 g1 = figure();
5 plot(x, diric(x,7));
6 grid on
7 title('n=7');
8 g2 = figure();
9 plot(x, diric(x,8));
10 grid on
11 title('n=8');
12 FFT1 = fft(diric(x,7));
13 FFT2 = fft(diric(x,8));
14 s1 = figure();
15 plot(x, FFT1)
16 s2 = figure();
17 plot(x, FFT2)
18
19 saveas(g1, '../fig/graph10_1','png');
20 saveas(g2, '../fig/graph10_2','png');
21 saveas(s1, '../fig/spec10_1','png');
22 saveas(s1, '../fig/spec10_2','png');

```

Листинг 13: Код генерации сигнала с меняющейся частотой

```

1 close all
2 clc
3 Fs = 8e3;
4 t = 0:1/Fs:1;
5 f0 = 1e3;
6 t1 = 1;
7 f1 = 2e3;
8 s1 = chirp(t, f0, t1, f1, 'linear');
9 s2 = chirp(t, f0, t1, f1, 'quadratic');
10 s3 = chirp(t, f0, t1, f1, 'logarithmic');
11 g1 = figure();
12 specgram(s1, [], Fs);
13 title('linear');
14 colormap gray
15 g2 = figure();
16 specgram(s2, [], Fs);
17 title('quadratic');
18 colormap gray
19 g3 = figure();
20 specgram(s3, [], Fs);
21 title('logarithmic');
22 colormap gray
23
24 saveas(g1, '../fig/graph11_1','png');
25 saveas(g2, '../fig/graph11_2','png');
26 saveas(g3, '../fig/graph11_3','png');

```

Листинг 14: Код сравнения алгоритмов определения корреляции

```

1 close all

```

```

2| clc
3|
4| x = [0 0 0 1 0 1 0 1 1 1 0 0 0 0 1 0];
5| y = [1 0 1];
6|
7| xx = x;
8| yy = zeros(1,length(x));
9| yy(y == 1) = 1;
10| yy(y == 0) = -1;
11|
12| R = zeros(1,length(xx));
13|
14| tic
15| for i = 1:length(xx)
16|     R(i) = sum(xx .* circshift(yy, i-1, 2)) / length(xx);
17| end
18| toc
19|
20| tic
21| xx = fft(xx);
22| yy = fft(yy);
23| xx = conj(xx);
24| BR = ifft(xx .* yy)/length(xx);
25| toc

```