

Санкт-Петербургский государственный политехнический университет Петра
Великого

Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе №1

Выполнила студентка гр.33501/3:Ивашкевич О.А.

Преподаватель: Богач Н.В.

Санкт-Петербург
2017

1 Сигналы телекоммуникационных систем

1.1 Цель работы

Познакомиться со средствами генерации и визуализации простых сигналов.

1.2 Постановка задачи

В командном окне MATLAB и в среде Simulink промоделировать сигналы из Главы 3, сс. 150–170 (см. Справочные материалы).

1.3 Справочные материалы

А.Б. Сергиенко Цифровая обработка сигналов. Глава 1, сс.18–25, Глава 3, сс. 150–170.

1.4 Ход работы

Необходимо провести генерацию сигналов в MATLAB, используя программный код, предложенный на страницах методического пособия. Для генерации сигнала необходимо сначала создать вектор дискретных значений времени. Рассчитаем обратную величину частоты дискретизации F_s , вычисленное значение будет использоваться для временного шага при моделировании частоты.

```
Fs = 8e3;           % частота дискретизации 8 кГц
t = 0:1/Fs:1;       % одна секунда дискретных значений времени
t = t';             % преобразуем строку в столбец
```

Ниже приведена функция для вычисления значения сигнала в любой момент времени.

```
A = 2;              % амплитуда - два вольта
f0 = 1e3;           % частота - 1 кГц
phii = pi/4;        % начальная фаза 45
s1 = A * cos(2*pi*f0*t + pi); % гармонический сигнал
alpha = 1e3;         % скорость затуханий
s2 = exp(-alpha*t).*s1; % затухающая синусоида
```

Для получения графиков, иллюстрирующих дискретный сигнал, используются нижеприведенные функции.

- Рис. 1, а – график зависимости $U(t)$, где U - сигнал t -время, построенный с помощью `plot`, при этом отсчеты соединяются линиями;
- Рис. 1, b – тот же график, но отображены только отсчеты;
- Рис. 1, c – функция `stem` изображает сигнал в виде «стебельков»;
- Рис. 1, d – функция `stairs` выводит график в ступенчатом виде.

Ниже приведена последовательность команд, для приведенных выше случаев:

```
subplot(2, 2, 1); plot(s2(1:100));      % с линиями
subplot(2, 2, 2); plot(s2(1:100), 'o'); % только точки
subplot(2, 2, 3); stem(s2(1:100));      % <<стебельки>>
subplot(2, 2, 4); stairs(s2(1:100));    % ступеньки
```

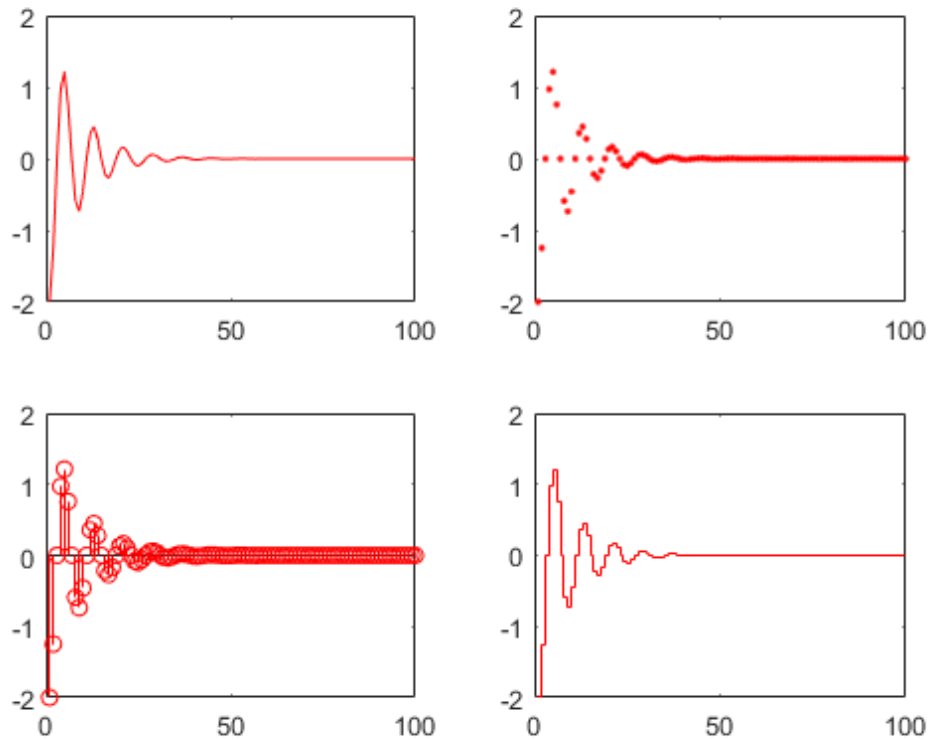


Рис. 1: Различные формы представления графиков дискретного сигнала.

Ось абсцисс, на графиках Рис. 1, проградуирована в номерах отсчетов. Чтобы показать на этой оси значения времени, можно подставить первым аргументом функций `plot`, `stem` или `stairs` вектор со значениями времени, например, так: `plot(t(1:100), s2(1:100))` (Рис. 2).

В MATLAB предоставляет возможность для генерации многоканального сигнала, каналы которого описываются одной и той же формулой, но с разными числовыми значениями параметров:

```
f = [600 800 1000 1200 1400]; % вектор частот (строка!)
s3 = cos(2*pi*t*f);           % пятиканальный сигнал
figure(3)
plot(t(1:100), s3(1:100, :));
```

В результате `s3` – матрица, содержащая значения произведения времени и частоты. Итог вычислений показан на Рис. 3:

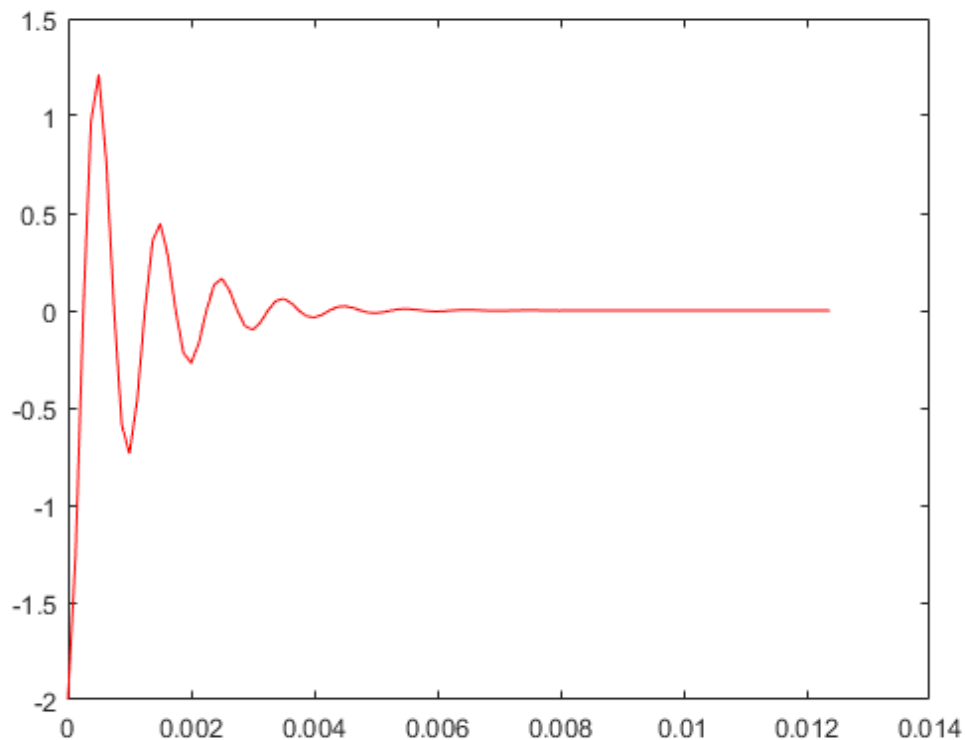


Рис. 2: График сигнала с корректно проградуированной временной осью.

1.4.1 Кусочные зависимости

В MATLAB создание кусочных функций возможно провести несколькими способами.

Первый вариант – использовать операцию сравнения, возвращающую значение 1, если условие выполняется, и 0 в противном случае. Этим можно воспользоваться для создания графиков, изображенных на Рис. 4:

- односторонний экспоненциальный импульс:

$$s4 = A * \exp(-\alpha * t) .* (t \geq 0);$$

- прямоугольный импульс, центрированный относительно начала отсчета времени:

```
Fs = 100;
t = -1:1/Fs:1; % 2 секунды дискретных значений времени
T = 0.5;
s5 = A*(abs(t) <= T/2);
```

- несимметричный треугольный импульс:

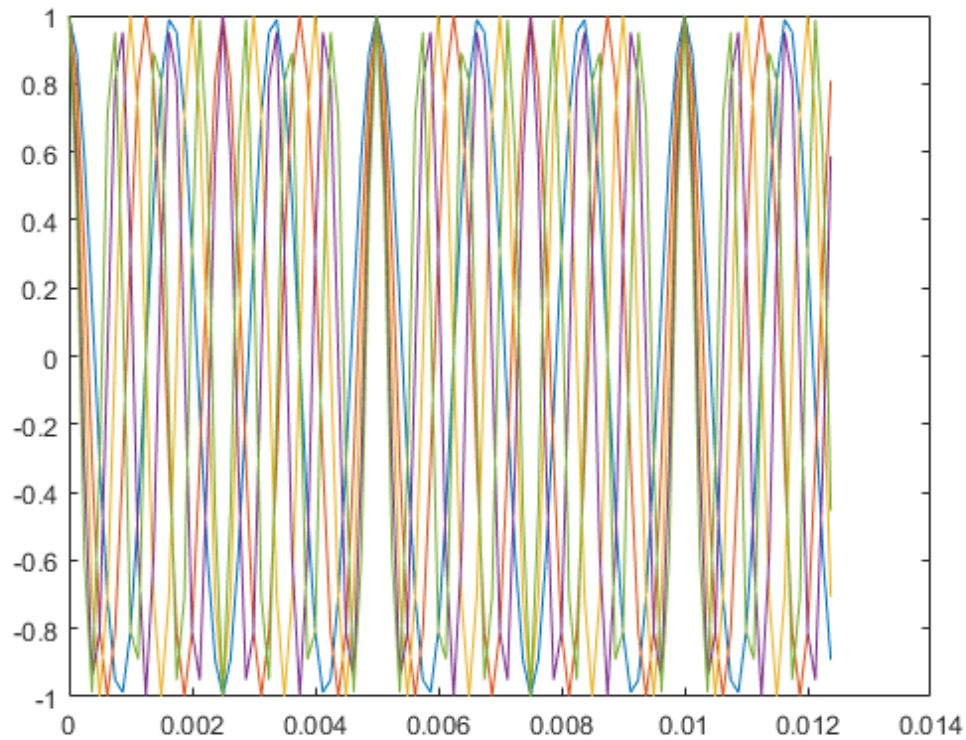


Рис. 3: Многоканальный сигнал.

$$s6 = A \cdot t / T \cdot (t \geq 0) \cdot (t \leq T);$$

Второй способ – выполнять вычисления только для тех моментов времени, для которых это необходимо. Пример создания трех кусочно-заданных сигналов:

- а – односторонний экспоненциальный импульс:

```
% заполняем вектор сигнала нулями
s4 = zeros(size(t));
% находим номера неотрицательных элементов вектора t*f
inds = find(t >= 0);
% рассчитываем сигнал только в нужных точках
s4(inds) = A * exp(-alpha * t(inds));
```

- b – прямоугольный импульс, центрированный относительно начала отсчета времени:

```
s5 = zeros(size(t));
s5(find(abs(t) <= T/2)) = A;
```

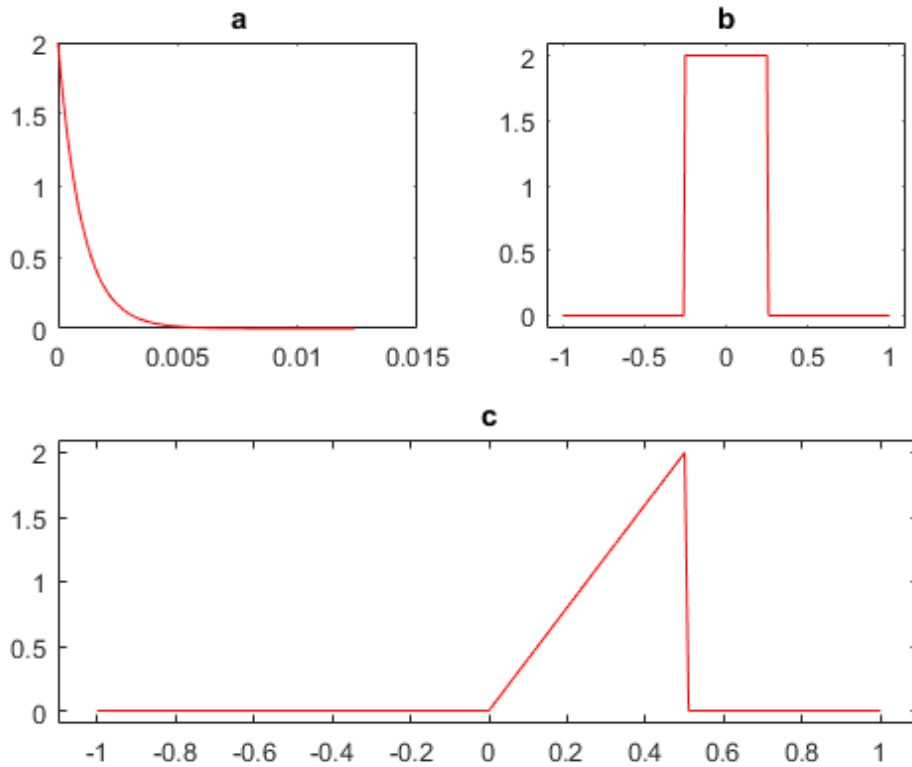


Рис. 4: Графики сигналов, полученные первым способом.

- с – несимметричный треугольный импульс:

```
s6 = zeros(size(t));
inds = find((t>=0) & (t <= T));
s6(inds) = A*t(inds)/T;
```

Результат визуализации не отличается от Рис. 4.

1.4.2 Прямоугольный импульс

Для формирования одиночного прямоугольного импульса с единичной амплитудой используется функция $y = \text{rectpuls}(t, \text{width})$, где t – вектор значений времени, width – ширина импульса. Возвращаемый результат y – вектор рассчитанных значений сигнала, определяемый формулой:

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

Пример создания прямоугольных импульсов с амплитудой 5 В и длительностью 20 мс каждый, расположенных справа и слева от начала отсчета времени. Результат показан на Рис. 5:

```

Fs = 1e3;           % частота дискретизации
t = -40e-3:1/Fs:40e-3; % дискретное время
T = 20e-3;          % длительность импульсов
A = 5;              % амплитуда
s = -A*rectpuls(t+T/2, T)+A*rectpuls(t-T/2,T);
plot(t,s);
ylim([-6 6]);

```

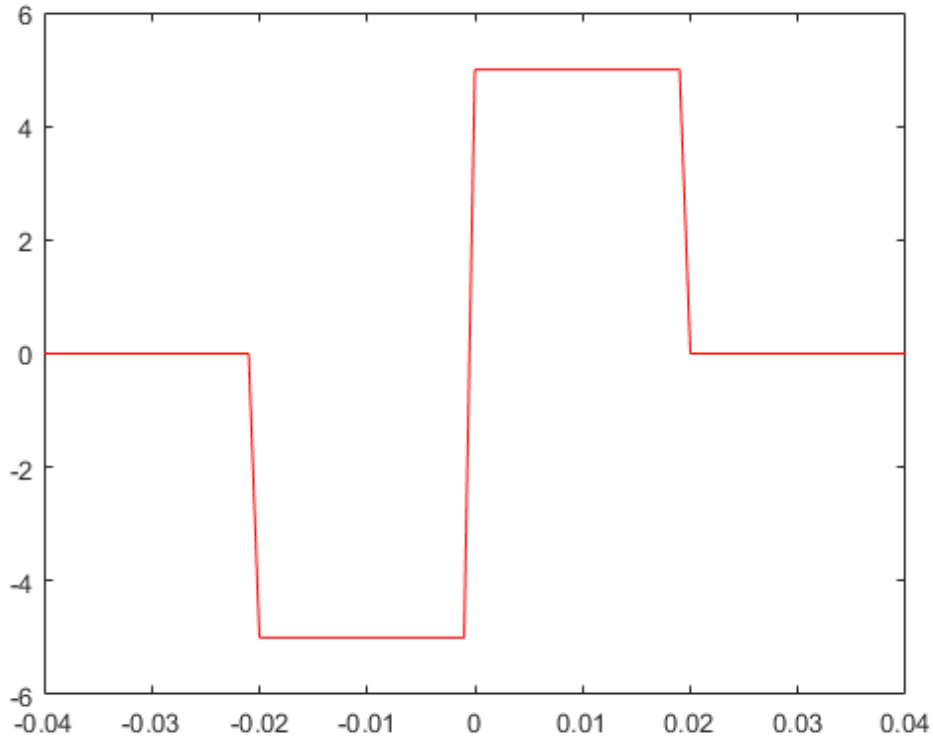


Рис. 5: Сигнал, сформированный с помощью rectpuls.

1.4.3 Треугольный импульс

Для того, чтобы сформировать одиночный треугольный импульс с единичной амплитудой используется функция $y = \text{tripuls}(t, \text{width}, \text{skew})$, где t – вектор значений времени, width – ширина импульса, skew – коэффициент асимметрии импульса, определяющий положение его вершины. Пик импульса расположен при $t = \text{width} \cdot \text{skew} / 2$. Параметр skew должен лежать в диапазоне от -1 до 1. Возвращаемый результат y – вектор рассчитанных значений сигнала, определяемый формулой:

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

Ниже приведен пример формирования симметричного трапецевидного импульса с амплитудой 10 В и размерами верхнего и нижнего оснований 0 и 50 мс соответственно. Результат показан на Рис. 6:

```

Fs = 1e3;
t = -50e-3:1/Fs:50e-3;
A = 10;
T1 = 20e-3;
T2 = 60e-3;
s = A*(T2*tripuls(t,T2)-T1*tripuls(t,T1))/(T2-T1);
plot(t,s);

```

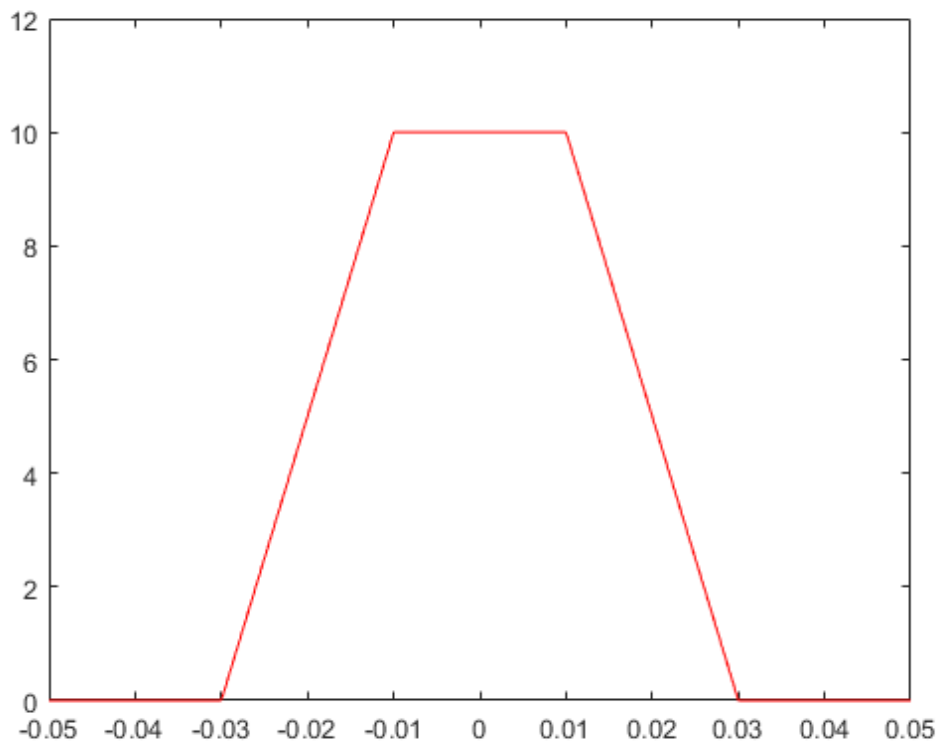


Рис. 6: Сигнал, сформированный с помощью tripuls.

1.4.4 Импульс с ограниченной полосой частот

Для формирования сигнала, имеющего прямоугольный импульс, служит функция `sinc(t)`, где `t` – то же, что и в предыдущих случаях. Возвращаемый результат `y` – вектор рассчитанных значений сигнала, определяемый формулой:

$$y = \frac{\sin(\pi x)}{\pi x}. \quad (3)$$

Для примера построю график `sinc` для линейного вектора (Рис. 7):


```

t = linspace(-5,5);
y = sinc(t);
plot(t,y);
xlabel('Time (sec)');
ylabel('Amplitude');
title('Sinc Function')

```

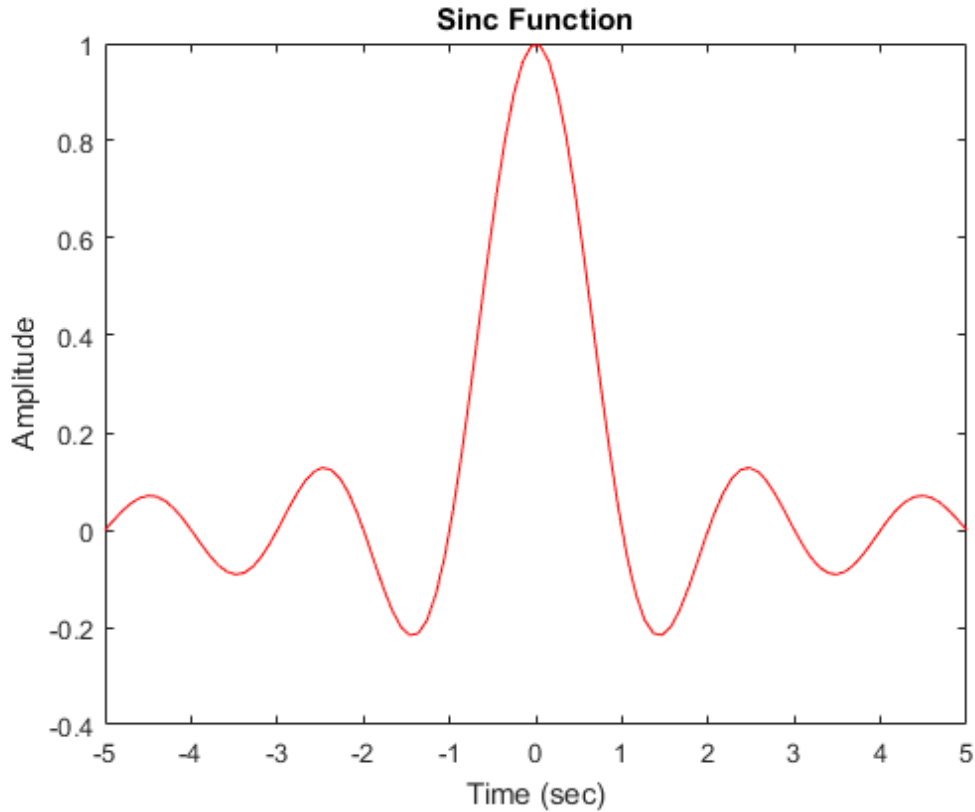


Рис. 7: Сигнал, сформированный с помощью sinc.

1.4.5 Гауссов радиоимпульс

Формирование одиночного радиоимпульса с гауссовой огибающей и единичной амплитудой осуществляется при помощи специальной функции $y = \text{gauspuls}(t, fc, bw, bwr)$, где t – вектор значений времени, fc – несущая частота в герцах, bw – относительная ширина спектра, bwr – уровень в децибелах, по которому производится измерение ширины спектра. Возвращаемый результат y – вектор рассчитанных значений сигнала, определяемый формулой:

$$y = \exp(-at^2) \cos(2\pi f_c t). \quad (4)$$

Коэффициент a используется для управления длительностью импульса и шириной его спектра. Сигнал (4) имеет спектральную функцию, равную

$$S(\omega) = \frac{1}{2} \sqrt{\frac{\pi}{a}} \left(\exp\left(-\frac{(\omega + 2\pi f_c)^2}{4a}\right) + \exp\left(-\frac{(\omega - 2\pi f_c)^2}{4a}\right) \right). \quad (5)$$

Если $f_c \gg \sqrt{a}$, то можно пренебречь наложением «хвостов» сдвинутых копий спектра. Тогда параметр a связан с относительной шириной спектра и уровнем (в дБ), по которому она определяется, следующим образом:

$$a = -\frac{5(2\pi f_c \cdot bw)^2}{bwr \cdot \ln 10} \quad (6)$$

Параметры bwr , bw и f_c можно опустить, при этом используются их значения по умолчанию: $bwr = -6$ дБ, $bw = 0.5$ и $f_c = 1000$ Гц.

При вызове функции `gauspuls` используется от одного до трех выходных параметров: `[y, yq, ye] = gauspuls()`. О значении второго и третьего выходного параметра можно узнать из справки MATLAB.

В качестве примера был сформирован гауссов радиоимпульс с несущей частотой 4 кГц и относительной шириной спектра 10%, измеренной по уровню -20 дБ. Результат продемонстрирован на Рис. 8:

```

Fs = 16e3;           % частота дискретизации
t = -10e-3:1/Fs:10e-3; % дискретное время
Fc = 4e3;            % несущая частота
bw = 0.1;            % относительная ширина спектра
bwr = -20;           % уровень измерения ширины спектра
s = gauspuls(t, Fc, bw, bwr);
plot(t, s);          % график сигнала
ylim([-1.1 1.1]);

```

Функция `gauspuls` позволяет рассчитать *время*, за которое огибающая гауссового импульса упадет до заданного уровня относительно максимума. При этом вместо вектора значений времени в качестве первого входного параметра используется строка `'cutoff':tc = gauspuls('cutoff', t, fc, bw, bwr, tpe)`, где `tpe` – уровень огибающей, момент достижения которого нужно определить. Возвращаемый результат `tc` – момент достижения огибающей уровня `tpe`.

В качестве примера рассчитывается время, за которое огибающая сформированного в предыдущем примере импульса уменьшается на 6 дБ (примерно в два раза):

```

>> gauspuls('cutoff', Fc, bw, bwr, -6)
ans =
    0.0020

```

1.4.6 Генерация последовательности импульсов

Конечную последовательность импульсов (pulse train) одинаковой формы с произвольно задаваемыми задержками и уровнями можно получить при помощи `y=pulstran(t, d, 'func', p1, p2, ...)` – если импульсы задаются именем генерирующей функции. Здесь `t` – вектор значений времени, `d` – вектор задержек, `'func'` – имя функции, генерирующей одиночный импульс. Оставшиеся параметры `p1`, `p2` и другие – дополнительные, передающиеся функции `func` при вызове.

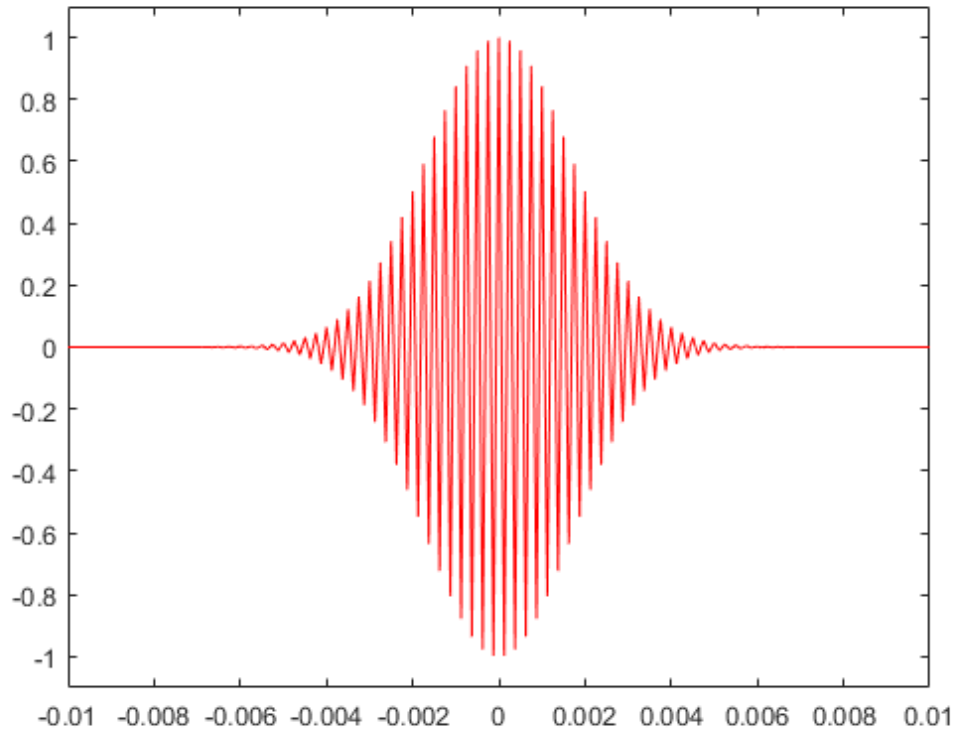


Рис. 8: Сигнал, сформированный с помощью gausspuls.

В качестве примера используется последовательность из пяти симметричных треугольных импульсов, интервалы между которыми линейно увеличиваются, а амплитуды экспоненциально уменьшаются. Результат показан на Рис. 9, а.

```

Fs = 1e3;                % частота дискретизации
t = 0:1/Fs:0.5;          % дискретное время
tau = 20e-3;             % длительность импульса
d = [20 80 160 260 380]'*1e-3; % задержки импульсов
d(:,2) = 0.8.^(0:4)';    % амплитуды импульсов
y = pulstran(t,d,'tripuls',tau);
plot(t, y)

```

Если для генерации одиночного импульса нет готовой функции, то можно рассчитать вектор отсчетов импульса, а затем использовать второй вариант вызова функции: `pulstran(t, d, p, fs, 'method')`.

Ниже приведен пример формирования последовательности из шести импульсов, имеющих форму одного периода функции \sin^2 . Результат показан на Рис. 9, b.

```

Fs0 = 400;               % частота дискретизации
tau = 60e-3;             % длительность импульса
t0 = 0:1/Fs0:tau;        % дискретное время для импульса
s0 = sin(pi*t0/tau).^2; % вектор отсчетов импульса

```

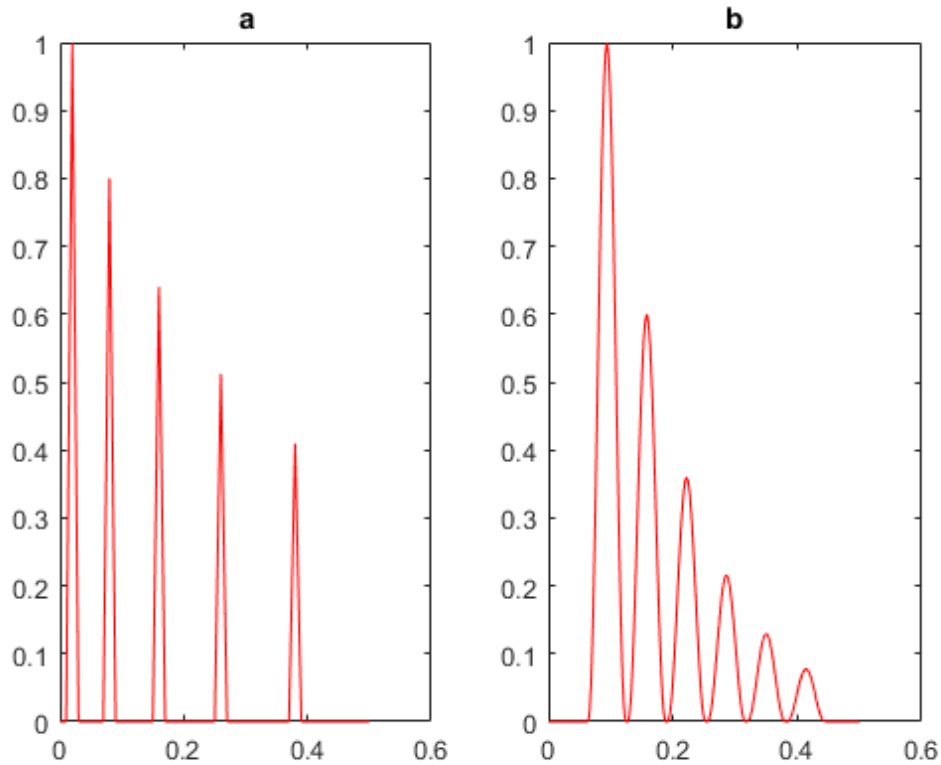


Рис. 9: Сигнал, сформированный с помощью pulstran.

```
% генерируем последовательность импульсов
Fs = 1e3; % частота дискр. послед-сти
t = 0:1/Fs:0.5; % дискретное время для посл-сти
d = (1:6)'*64e-3; % задержки импульсов
d(:,2) = 0.6.^(0:5)'; % амплитуды импульсов
y = pulstran(t,d,s0,Fs0);
plot(t,y);
```

1.4.7 Последовательность прямоугольных импульсов

Последовательность прямоугольных импульсов можно сформировать с помощью функции $y = \text{square}(t)$, где t – вектор значений времени. Последовательность будет иметь период 2π и скважность 2. С помощью второго входного параметра 'duty' можно регулировать скважность.

Сформировать последовательность с периодом T можно так: $y = \text{square}(2\pi * t/T)$.

Пример последовательности однополярных прямоугольных импульсов с амплитудой 3 В, частотой следования 50 Гц, длительностью 5 мс. Результат на Рис. 10.

```
Fs = 1e3; % частота дискретизации
t = -10e-3:1/Fs:50e-3; % дискретное время
A = 3; % амплитуда
```

```

f0 = 50;                % частота следования импульсов
tau = 5e-3;             % длительность импульсов
s = (square(2*pi*t*f0, f0*tau*100)+1)*A/2;
plot(t,s)
ylim([0 5])

```

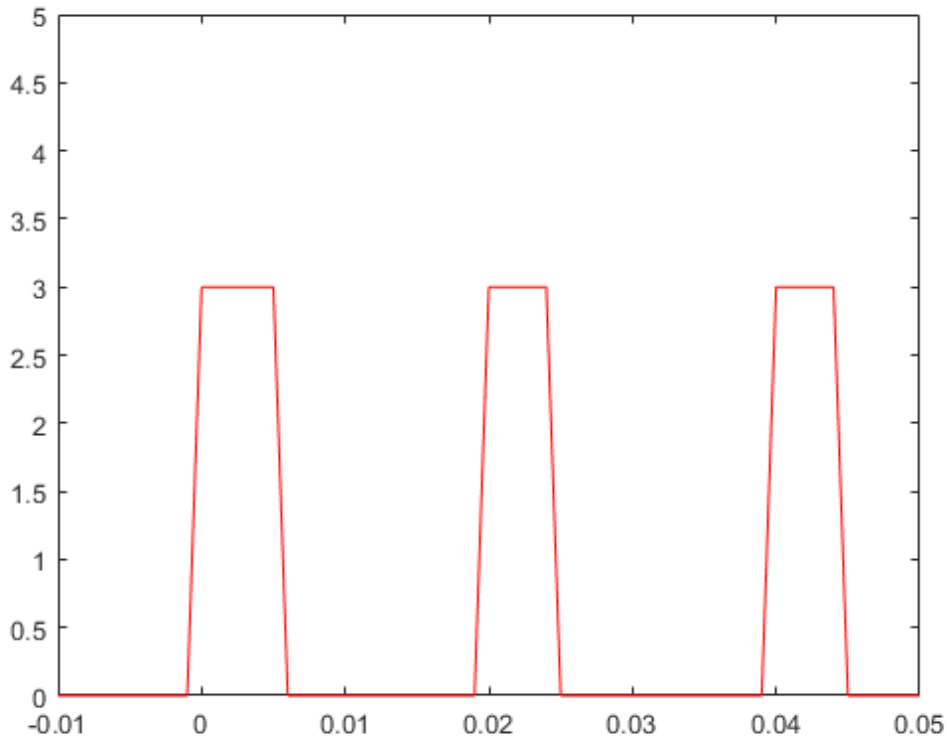


Рис. 10: Последовательность прямоугольных импульсов, полученная с помощью функции `square`.

1.4.8 Последовательность треугольных импульсов

Формирование последовательности треугольных импульсов происходит с использованием функции `y = sawtooth(t)`, где `t` – то же, что и в предыдущем случае. Период – 2π . С помощью второго параметра `width` можно регулировать длительность «обратного хода». По умолчанию этот параметр равен единице.

Сформировать последовательность с периодом T можно так: $y = \text{sawtooth}(2\pi * t/T)$.

В качестве примера построена последовательность треугольных импульсов отрицательной полярности с амплитудой 5 в, периодом 50 мс и длительностью падающего участка 5 мс. Результат приведен на Рис. 11.

```

Fs = 1e3;                % частота дискретизации
t = -25e-3:1/Fs:125e-3; % дискретное время

```

```

A = 5; % амплитуда
T = 50e-3; % период
t1 = 5e-3; % длительность падающего участка
s = (sawtooth(2*pi*t/T, 1-t1/T)-1)*A/2;
plot(t,s)

```

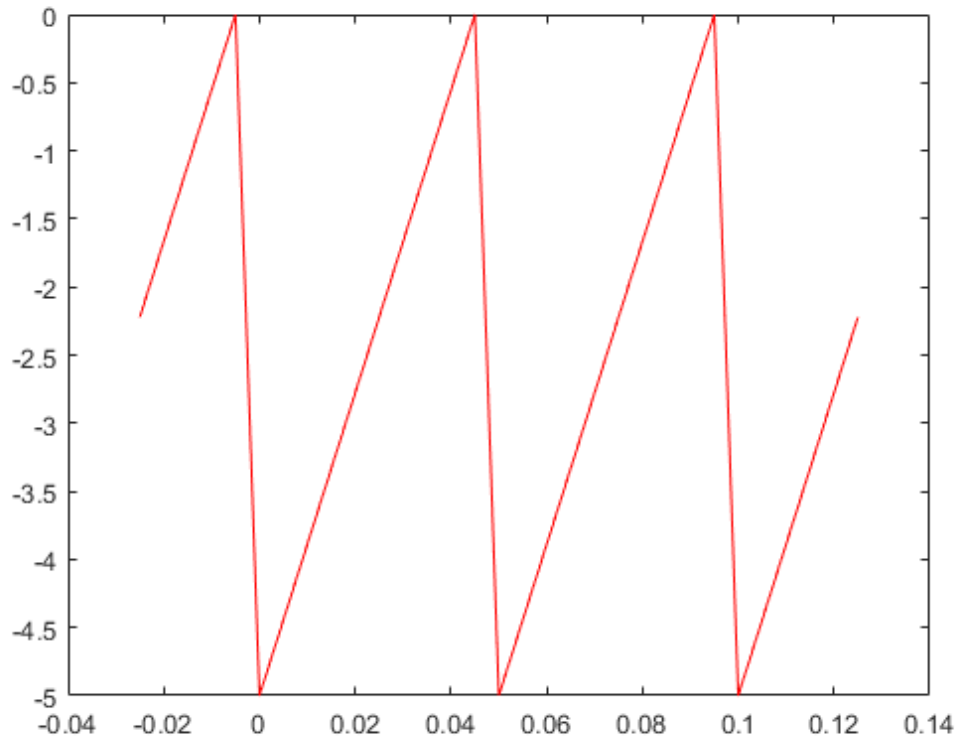


Рис. 11: Последовательность прямоугольных импульсов, полученная с помощью функции sawtooth.

1.4.9 Функция Дирихле

Функция Дирихле описывается следующей формулой

$$\text{diric}_n(x) = \sum_{k=-\infty}^{\infty} \text{sinc} \left(n \left(\frac{t}{2\pi} - k \right) \right), \quad (7)$$

где n – целое положительное число.

Сформировать такую последовательность можно при помощи функции $y=\text{diric}(x,n)$, где x и n соответствуют формуле (7).

Для примера построил графики функции Дирихле при нечетном и четном значениях n ($n = 7$ и $n = 8$, Рис. 12).

```

x = 0:0.01:15;
subplot(2,1,1);

```

```

plot(x, diric(x, 7));
grid on;
title('n = 7');
subplot(2,1,2);
plot(x, diric(x, 8));
grid on;
title('n = 8');

```

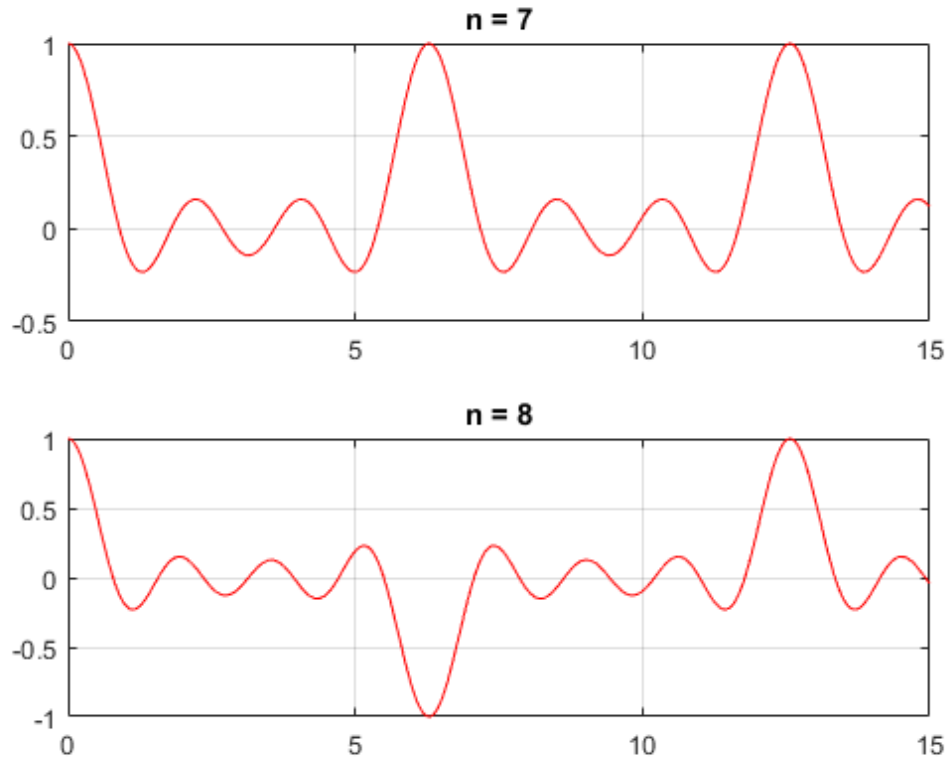


Рис. 12: Функция Дирихле нечетного (сверху) и четного (снизу) порядка.

1.4.10 Генерация сигнала с меняющейся частотой

Функция `y=chirp(t, f0, t1, f1, 'method', phi)` предназначена для генерации колебаний с единичной амплитудой, *мгновенная частота* которых меняется по заданному закону. Здесь `t` – вектор значений времени, `phi` – начальная фаза колебания. Остальные параметры определяют закон изменения частоты.

Строковый параметр `'method'` определяет тип зависимости мгновенной частоты от времени – `'linear'`, `'quadratic'` или `'logarithmic'`. Математически они описываются следующим образом:

'linear' :

$$f(t) = f_0 + \beta t, \text{ где } \beta = \frac{f_1 - f_0}{t_1}; \quad (8)$$

'quadratic' :

$$f(t) = f_0 + \beta t^2, \text{ где } \beta = \frac{f_1 - f_0}{t_1^2}; \quad (9)$$

'logarithmic' :

$$f(t) = f_0 + e^{\beta t}, \text{ где } \beta = \frac{\ln(f_1 - f_0)}{t_1}; \quad (10)$$

Ниже приведен пример формирования трех сигналов, определенных на промежутке $0 \dots 1$ с и имеющих разные законы изменения мгновенной частоты.

```
Fs = 8e3;      % частота дискретизации
t = 0:1/Fs:1; % дискретное время
f0 = 1e3;
t1 = 1;
f1 = 2e3;
s1 = chirp(t,f0,t1,f1,'linear');
s2 = chirp(t,f0,t1,f1,'quadratic');
s3 = chirp(t,f0,t1,f1,'logarithmic');
specgram(s1,[],Fs)
title('linear');
colormap gray;
figure;
specgram(s2,[],Fs)
title('quadratic');
colormap gray;
figure;
specgram(s3,[],Fs)
title('logarithmic');
colormap gray;
```

На Рис. 13-15 изображены спектрограммы сформированных сигналов, наглядно демонстрирующие характер изменения мгновенной частоты при различных значениях параметра 'method'.

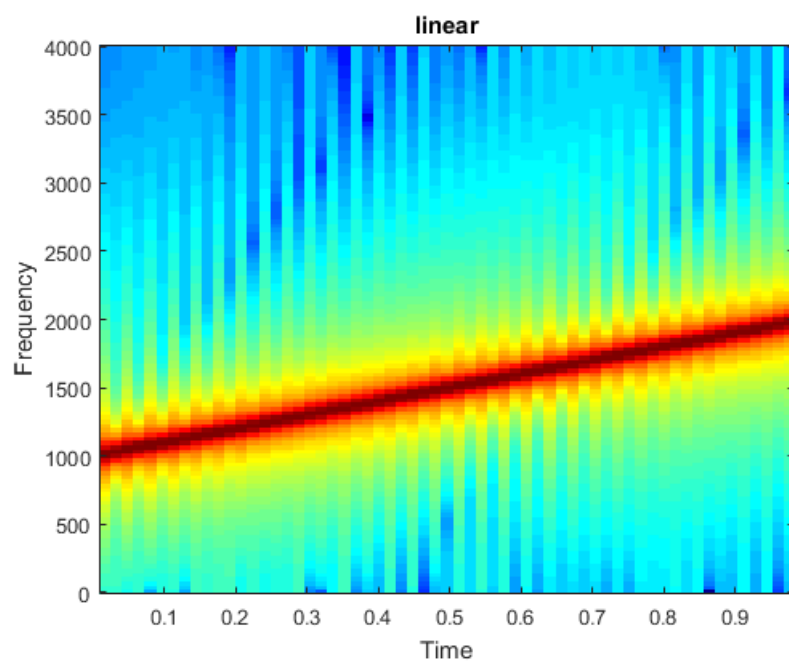


Рис. 13: Спектрограмма сигнала, сформированного функцией `chirp` при линейном законе изменения мгновенной частоты.

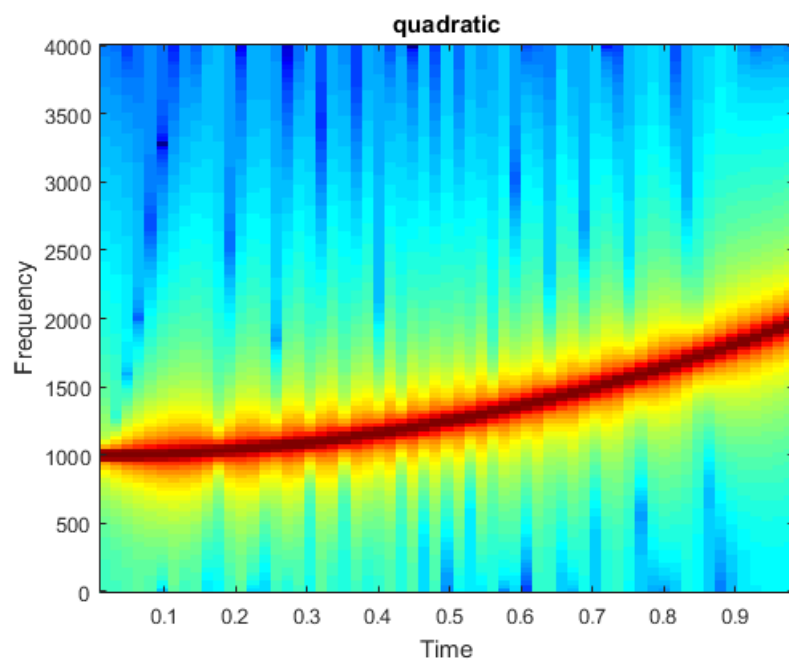


Рис. 14: Спектрограмма сигнала, сформированного функцией `chirp` при квадратичном законе изменения мгновенной частоты.

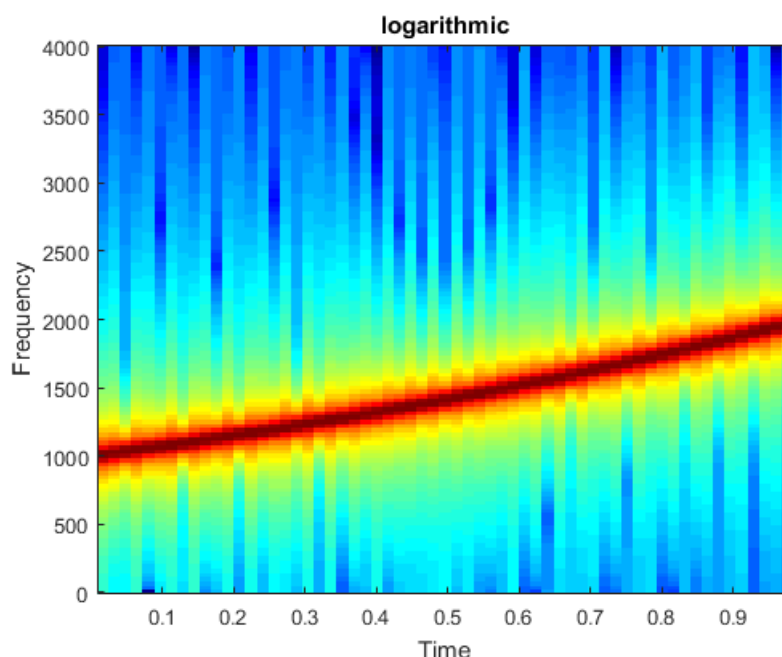


Рис. 15: Спектрограмма сигнала, сформированного функцией chirp при экспоненциальном законе изменения мгновенной частоты.

1.5 Вывод

Сигнал – физическая величина, изменяющаяся в зависимости от времени и/или пространственной переменной или набора (вектора) переменных (многомерные сигналы).

Классификация сигналов представляет собой две группы: **детерминированные** и **случайные**. В зависимости от того, известен ли сигнал *точно*. В данной работе был рассмотрен класс детерминированных сигналов. Однако, возможность генерации случайных сигналов есть, проблема их генерации состоит в том, что они могут быть предсказаны лишь с некоторой вероятностью и отображают случайное физическое явление или физический процесс, причем зарегистрированный в единичном наблюдении сигнал не воспроизводится при повторных наблюдениях и не может быть описан явной математической зависимостью.

При анализе систем, в качестве тестовых сигналов используют детерминированные, так как в этом случае важно знать точный сигнал.

Если сигнал может быть описан функцией $s(t) = s(t + T)$, где T – период, он называется **периодическим**. Если сигнал невозможно описать функцией, он называется **непериодическим**.

По времени сигналы могут делиться на **непрерывные** и **дискретные**.

Существуют и другие критерии классификации сигналов. Например, различают ауди- и видеосигналы, с конечной и бесконечной энергией, широкополосные и узкополосные и так далее.

В пакете математического обеспечения MATLAB есть расширение Signal Processing, в котором имеются следующие функции, генерирующие часто встречающиеся на практике непериодические сигналы (одиночные импульсы):

rectpuls – прямоугольный импульс;

tripuls – треугольный импульс;

sinc – импульс вида $\sin(\pi t)/(\pi t)$;

gauspuls – радиоимпульс с гауссовской огибающей;

pulstran – последовательность из конечного числа импульсов произвольной формы.

Кроме того, реализованы функции, позволяющие формировать отсчеты периодических сигналов различной формы:

square – последовательность прямоугольных импульсов;

sawtooth – последовательность треугольных импульсов;

diric – функция Дирихле.