

# 1 ЛАБОРАТОРНАЯ РАБОТА №1. ДИСКРЕТНЫЕ СИГНАЛЫ

ЦЕЛЬ РАБОТЫ – изучение пакета MATLAB, программирование базовых сигналов цифровой обработки (ЦОС) в пакете MATLAB.

## 1.1 Теоретические сведения

### *Пакет MATLAB*

Для выполнения лабораторных работ используется система научных расчетов MATLAB (сокращение от MATrix LABoratory – матричная лаборатория). Основным объектом MATLAB является матрица. Все вычисления система осуществляет в арифметике с плавающей точкой.

Система MATLAB работает в режиме интерпретатора. Команды вводятся в командной строке (рис. 1.1), а MATLAB выполняет их немедленную обработку и выдает вычисленный результат. MATLAB также может выполнять заранее подготовленную последовательность команд, записанную в виде текстового файла с расширением `.m` (М-файл).

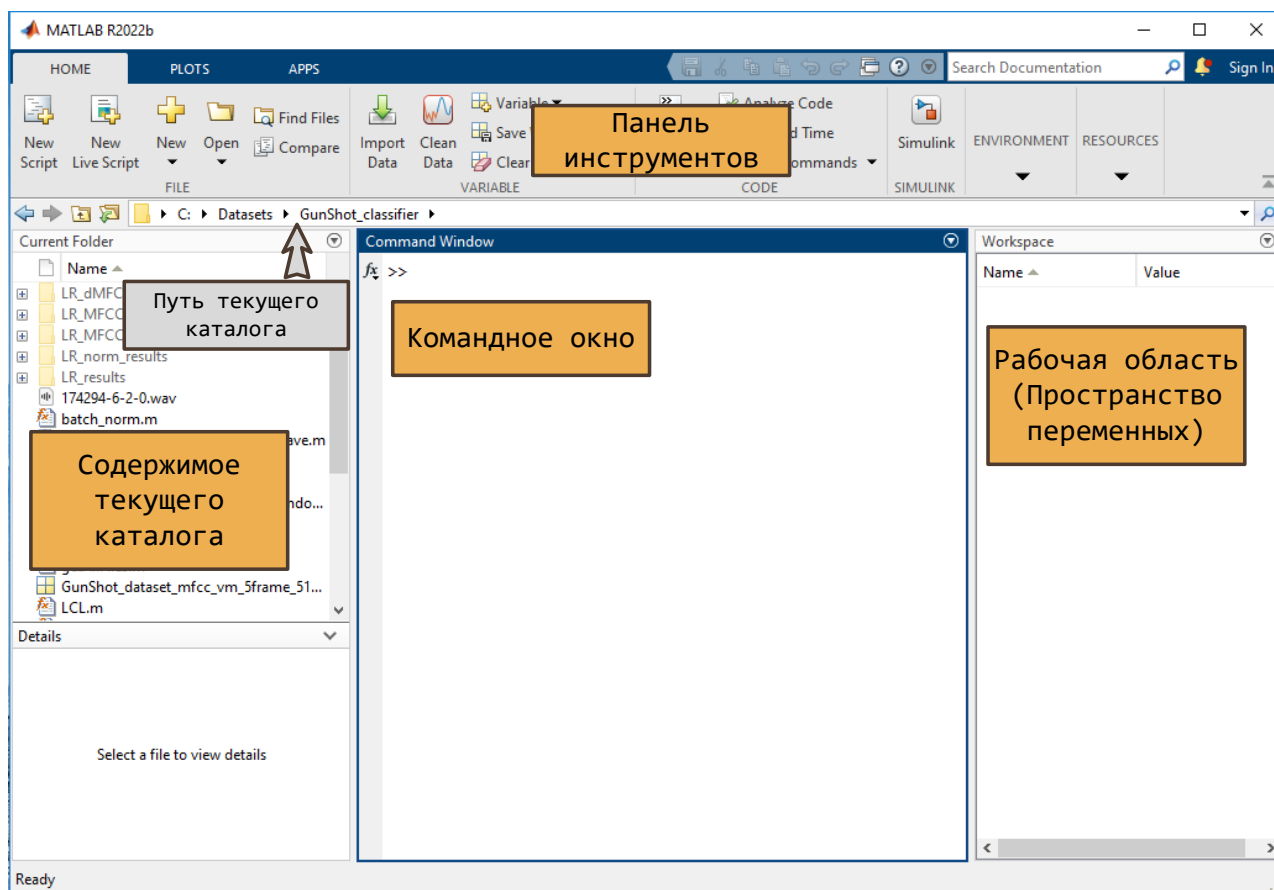


Рис. 1.1. Вид системы MATLAB

M-файлы разделяются на два вида: файлы-сценарии и процедуры-функции. Все M-файлы должны располагаться в рабочем каталоге или каталоге, зарегистрированном в списке путей системы MATLAB. Для изменения рабочего каталога необходимо выбрать указать путь его в поле текущего каталога.

### **Файлы-сценарии**

Файл-сценарий – это текстовый файл, содержащий последовательности команд и операторов. В языке MATLAB нет специальных операторов для обозначения начала и конца файла-сценария, а также точки входа.

Имя файла сценария может содержать только символы латинского алфавита от A до Z и от a до z, цифры от 0 до 9 и символ подчеркивания;

Результат выполнения каждого оператора отображается в окне командной строки. Для того чтобы система не выводила результат, необходимо в конце оператора ставить точку с запятой.

### **Файлы -функции**

Файл-функция также является текстовым файлом, содержащим последовательности команд. Файл-функция имеет несколько отличий от файла-сценария.

Первой строкой файла –функции должен быть заголовок функции:

```
function [<список выходных параметров>] = <имя функции>(<список входных параметров>)
```

Имя функции должно совпадать с именем файла. Для примера рассмотрим функцию, вычисляющую минимальное и максимальное значения массива, которая должна быть сохранена в файл с названием minmax.m :

```
function [minimum, maximum] = minmax(array)
minimum = min(array)
maximum = max(array)
```

Как видно из этого примера, функция может иметь несколько возвращаемых значений. Представленную в примере функцию можно использовать следующим образом:

```
[a,b] = minmax(x)    % В a будет минимум, в b - максимум
a = minmax(x)       % В a будет минимум, максимум теряется
```

По соглашению, принятому в MATLAB, начиная со второй строки функции может располагаться несколько строк комментария. Этот комментарий считается справкой по использованию функции и может быть вызван с помощью команды help, например:

```
help filter % справка по функции filter
```

### ***Создание переменных в системе MATLAB***

В MATLAB не требуется объявлять переменные. Для создания переменной ей просто необходимо присвоить некоторое значение. Рассмотрим несколько вариантов создания переменных в системе MATLAB.

В простейшем случае мы можем создать пустой массив:

```
a = []
```

Этот вариант может потребоваться, например, в том случае, когда мы заранее не знаем сколько элементов будет содержать наш массив. В процессе работы можно добавлять данные в массив путем конкатенации, как показано в следующем примере:

```
>> a = [];  
>> a = [a 1];  
>> a = [a 2]  
a =  
    1    2
```

В MATLAB матрица  $1 \times 1$  называется скаляром. Покажем пример объявления скаляра:

```
b = 8  
c = 0.25 + 18i % Комплексное число
```

Для записи комплексного числа используют символ мнимой единицы. В MATLAB для обозначения мнимой единицы используют символ  $i$  либо  $j$ . Ниже показаны два равнозначных способа задания комплексного числа:

```
c1 = 0.5 + 2i % Первое комплексное число  
c2 = 2 + 0.5j % Второе комплексное число
```

Если нам необходимо объявить вектор, то достаточно записать его элементы, разделенные пробелами, в квадратных скобках:

```
x = [1 2]
y = [2.2 -5]
z = [x 4 y] % Результат z = [1 2 4 2.2 -5]
```

### ***Монотонно возрастающие или убывающие последовательности***

В MATLAB предусмотрен следующий формат объявления монотонных последовательностей (или векторов):

первый\_элемент : [шаг :] конечный\_элемент.

Если шаг равен 1, то его разрешается опустить:

```
x = 1:5 % Результат x = [1 2 3 4 5]
y = 2:-2:-6 % Результат y = [2 0 -2 -4 -6]
z = -1:3:10 % Результат z = [-1 2 5 8]
```

### ***Создание массивов***

Для объявления массива нужно записать его элементы в квадратных скобках, притом элементы строки разделяются пробелами, а строки между собой – точкой с запятой.

Например, массив

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 7 & 11 & 13 \end{bmatrix}$$

в MATLAB будет записываться как

```
A = [2 3 5; 7 11 13];
```

Для объявления специальных массивов в MATLAB предусмотрены следующие функции:

<code>zeros(m, n)</code>	– возвращает массив размером $m \times n$ , заполненный нулями;
<code>ones(m, n)</code>	– возвращает массив размером $m \times n$ , заполненный единицами;
<code>rand(m, n)</code>	– возвращает массив размером $m \times n$ , заполненный псевдослучайными величинами, распределенными по равномерному закону;
<code>randn(m, n)</code>	– возвращает массив размером $m \times n$ , заполненный псевдослучайными величинами, распределенными по нормальному закону.

В данных функциях может опускаться первый операнд. В этом случае возвращается квадратная матрица  $n \times n$ . Пример:

```
a = zeros(1, 100)% Вектор-строка из 100 нулей
b = ones(20)      % Матрица 20x20 единиц
c = rand(2, 50)  % Матрица 2x50 псевдослучайных чисел
```

### ***Доступ к массивам и его элементам***

Доступ к массивам осуществляется по их именам. Все операции с массивами осуществляются по значению, т.е. например, при выполнении операции присваивания будет копироваться все содержимое массива. Пример:

```
a = [2 4 5; 4 6 1]
b = a
```

В результате выполнения данной программы переменная *b* будет иметь размер  $2 \times 3$  и содержать значение  $[2\ 4\ 5; 4\ 6\ 1]$ , притом *a* и *b* будут размещаться в разных участках памяти.

Для доступа к элементу массива необходимо после имени задать в круглых скобках индекс элемента, состоящий из двух чисел или векторов, разделенных запятой. Нумерация начинается с единицы. Первое число или вектор определяет номер строки (номера строк), второе – соответственно столбца. Если номер строки пропущен (обычно при обращении к элементам вектора), то он считается равным единице. Пример:

```
a = [11 12 13 14 15;      % Исходный массив
     21 22 23 24 25;
     31 32 33 34 35;
     41 42 43 44 45]
b = a(2, 3)              % Результат: b = 23
a(3) = 6                 % Результат: a(1,3) = 0
c = a(3, 2:4)           % Результат: c = [32 33 34]
d = a([1 3], 1:3)       % Результат: d = [11 12 13;
                                     %          31 32 33]
a([1 4], [1 5]) = zeros(2,2); % Результат:
                                     % a = [0 12 13 14 0;
                                     %       21 22 23 24 25;
                                     %       31 32 33 34 35;
                                     %       0 42 43 44 0]
```

### ***Арифметические операции***

Операции над матрицами в MATLAB определены в соответствии с правилами линейной алгебры. К таким операциям относят сложение, вычитание и

умножение, также отдельно выделяют умножение матрицы на скаляр. Приведем пример сложения матриц.

```
>> A = [1 2; 3 4];
>> B = [2 4; 1 3];
>> C = A+B

C =

     3     6
     4     7
```

Согласно законам линейной алгебры, складывать и вычитать можно матрицы одного и того же размера. Ниже приведен пример умножения матрицы A из предыдущего примера на константу

```
>> D = 1.5*A

D =

     1.50     3.00
     4.50     6.00
```

Если у нас есть матрица A размером  $m_1 \times n_1$  и матрица B размером  $m_2 \times n_2$ , то их можно перемножить лишь в случае, когда  $n_1 = m_2$ . В результате умножения будет получен матрица размера  $m_1 \times n_2$ . Приведем пример.

```
>> A = [1 1; 1 2; 1 3];
>> B = [1 1 1; 3 3 3];
>> E = A*B

E =

     4     4     4
     7     7     7
    10    10    10
```

В данном примере матрица A имела размер  $3 \times 2$ , а матрица B размер  $2 \times 3$ . В результате перемножения получилась матрица E размером  $3 \times 3$ .

В MATLAB предусмотрена операция добавления скаляра к матрице. В этом случае значение скаляра будет добавлено ко всем элементам матрицы, как показано в следующем примере.

```
>> B = [1 1 1; 3 3 3];
>> E = B-1

E =

     0     0     0
```

Часто встречаются ситуации, когда необходимо умножить элементы одной матрицы на элементы другой. В это случае необходимо использовать поэлементную операцию умножения, которая обозначается «.\*». Ниже приведен пример поэлементного умножения.

```
>> A = [2 4; 6 8];
>> B = [4 2; 4/3 1];
>> C = A.*B
```

C =

```
8 8
8 8
```

Аналогичным образом выполняется операция поэлементного деления «./».

Операцию транспонирования в MATLAB осуществляет оператор «`» (апостроф).

```
>> B = [1 1 1; 3 3 3];
>> C = B`
```

C =

```
1 3
1 3
1 3
```

Используя транспонирование в MATLAB следует быть аккуратным, поскольку с точки зрения математики, если матрица содержит комплексные числа, то при транспонировании каждое число заменяется на комплексно сопряженное. Покажем, как это работает.

```
>> B = [1 1 1+2*1j; 3 3 3-1j];
>> C = B`
```

C =

```
1.0 + 0.0i 3.0 + 0.0i
1.0 + 0.0i 3.0 + 0.0i
1.0 - 2.0i 3.0 + 1.0i
```

Если же мы заинтересованы только в том, чтобы переставить элементы матрицы и не выполнять операцию комплексного сопряжения, то необходимо использовать оператор «.`» (точка и апостроф). Действие этого оператора представлено в примере ниже.

```
>> B = [1 1 1+2*1j; 3 3 3-1j];
>> C = B.\`

C =

    1.0 + 0.0i    3.0 + 0.0i
    1.0 + 0.0i    3.0 + 0.0i
    1.0 + 2.0i    3.0 - 1.0i
```

**Логические операторы. Операции отношения.**

В языке MATLAB используются следующие логические операторы (табл.1.1):

Таблица 1.1

Операции отношения в MATLAB

Операция	Отношение
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно
==	Равно
~=	Не равно

Все логические операторы осуществляют операцию поэлементного сравнения двух массивов. Если один из операндов является скаляром, то он поэлементно сравнивается со всеми элементами другого операнда. Логические операторы возвращают в качестве результата массив того же размера, элементы которого равны единице, если результат сравнения соответствующих элементов равен ИСТИНА, и нулю – в противоположном случае.

Операторы <, >, <=, >= используются для сравнения только действительных частей комплексных элементов, а операции == и ~= осуществляют сравнение как действительных, так и мнимых частей. Пример:

```
X = [2 4 2.5;
     12i 6 3]
Y = [3 4 12;
     2 3 6]
Z = X >= Y

% Результат

Z = [0 1 0;
     0 0 1]
```



## Логические операции

В языке MATLAB есть три логические операции (табл. 1.2):

Таблица 1.2

Логические операции в MATLAB

Операция	И	ИЛИ	НЕ
Обозначение	&		~

При выполнении логических операций массив рассматривается как совокупность булевых переменных, так что значение 0 соответствует булеву значению false, а любое другое значение – булеву значению true. Функция ИСКЛЮЧАЮЩЕЕ ИЛИ реализована в виде функции xor(A, B).

Логические операции имеют низший приоритет по отношению к операциям отношения и арифметическим операциям.

### Оператор цикла с определенным числом операций

Оператор цикла с определенным числом операций имеет следующий вид:

```
for v = <выражение-массив>  
<операторы>  
end
```

В отличие от универсальных языков программирования переменная цикла в языке MATLAB является массивом. Поэтому исполнение цикла состоит в том, что переменной цикла присваиваются значения столбцов массива и затем выполняются операторы, которые должны зависеть от переменной v. Тем не менее на практике чаще всего применяют линейные конструкции вида m : n или a : <шаг> : b.

Для прерывания выполнения цикла используется оператор break. Пример:

```
for i = 1:10  
    x(i) = i .^ 2  
end  
% Результат x = [1 4 9 16 ... 100]  
  
y = []  
for k = [0 3 1 2]  
    y = [y 2.^k]  
end  
% Результат y = [1 8 2 4]
```

### Добавление элементов к массивам

Переменная-вектор или переменная-матрица может быть изменена путем добавления новых элементов.

Добавление новых элементов к переменной-вектору осуществляется путем присваивания значений новым элементам вектора. Ниже показан пример добавления новых элементов к вектору **a**:

```
a = [1 2 3 4 5]
a = 1 2 3 4 5
a(6:10) = 6:10
a = 1 2 3 4 5 6 7 8 9 10
```

Для переменных-матриц добавление новых элементов производится путем присвоения новых значений строкам и столбцам исходной матрицы. Рассмотрим пример добавления новых элементов к матрице **A**:

```
A = [1 2; 3 4]
A =
     1     2
     3     4
A(3,:) = [5 6]
A =
     1     2
     3     4
     5     6
A(:,3) = [7; 8; 9]
A =
     1     2     7
     3     4     8
     5     6     9
```

### ***Удаление элементов массива***

Удаление элементов из переменной-вектора или переменной матрицы осуществляется путем присвоения пустого множества этим элементам. В MATLAB пустое множество обозначается квадратными скобками []. Рассмотрим пример удаления элементов из вектора **a**:

```
a = 1:8
a = 1 2 3 4 5 6 7 8
a(1:5) = []
a = 6 7 8
```

Ниже показан пример удаления последнего столбца из матрицы **A**:

```
A = [1 2 3; 4 5 6; 7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
A(:,3) = []
A =
     1     2
```

```
4 5
7 8
```

### ***Функции управления массивами***

В MATLAB существует множество полезных функций для управления и обработки массивов.

Функция `length(a)` возвращает число элементов вектора `a`.

```
a = -10:0.1:100;
length(a)
ans =
    1101
```

Функция `size(A)` возвращает параметры `[m,n]`, где `m` и `n` – размеры `m × n` матрицы `A`:

```
A = [1 2 3; 4 5 6; 7 8 9];
size(A)
ans =
     3     3
```

Функция `reshape(A,m,n)` позволяет создать матрицу размером `m × n` из элементов матрицы `A`. Элементы берутся по столбцам – столбец за столбцом и расставляются по столбцам. При этом матрица `A` должна иметь `m × n` элементов.

```
A = [1 2 3 4; 5 6 7 8]
A =
     1     2     3     4
     5     6     7     8
reshape(A,4,2)
ans =
     1     3
     5     7
     2     4
     6     8
```

### ***Оператор цикла с неопределенным числом операций***

Оператор цикла с неопределенным числом операций имеет следующий вид:

```
while <логическое выражение>
<операторы>
end
```

Цикл `while ... end` выполняется до тех пор, пока массив логического выражения не станет нулевым.

Логическое выражение имеет форму

выражение <оператор отношения> выражение,

где допустимы следующие операторы отношений : `==`, `~=`, `<=`, `>=`, `<`, `>`. Пример:

```
i = 1
s = 0
while x(i) ~= 0 & i < 4
    s = s + x(i)
    i = i + 1
end
```

### ***Условное выражение***

Как и во всех языках программирования, в языке MATLAB есть конструкции для организации условного выполнения операторов. Одна из конструкций имеет следующий формат:

```
if <логическое выражение>
    <операторы>
elseif <логическое выражение>
    <операторы>
else
    <операторы>
end
```

В условном выражении может присутствовать несколько блоков `elseif` или данный блок может отсутствовать. Также может отсутствовать блок `else`.

Логическое выражение имеет форму

выражение <оператор отношения> выражение,

где допустимы следующие операторы отношений: `==`, `~=`, `<=`, `>=`, `<`, `>`. Пример:

```
if a > 0
    b = 2
elseif a < 0
    b = 0
else
    b = 1
end
```

В MATLAB имеется еще одна конструкция для организации условного выполнения операторов:

```
switch <анализируемое выражение>
  case <выражение 1>
<операторы>
  case <выражение 2>
<операторы>
  ...
  otherwise
<операторы>
end
```

Оператор `switch` последовательно сравнивает анализируемое выражение с выражениями, записанными после `case`, и если выражения равны, то выполняются соответствующие операторы. Если не было найдено ни одного равенства, то выполняются операторы, записанные после `otherwise`.

### ***Построение графиков в MATLAB***

В MATLAB имеется способов визуализации данных. Рассмотрим основные функции, которые понадобятся для выполнения лабораторных работ.

Простейшим вариантом для построения графика сигнала является функция `plot`. Ниже показан пример использования `plot` для построения графика параболы.

```
x=-2:0.01:2;
y=x.^2;
figure;
plot(x,y)
```

В результате выполнения этого блока кода появится окно (`figure`) с изображением параболы (рис. 1.2).

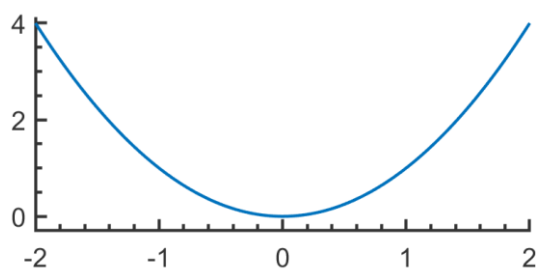


Рис. 1.2. Отображение параболы при помощи функции `plot`

Графику на рис. 1.2 не хватает подписей координатных осей. Этот недостаток можно устранить, используя команды `xlabel` и `ylabel`, как показано в следующем примере:

```
xlabel('x')
```

```
ylabel('y')
```

В результате получится график, показанный на рис. 1.3.

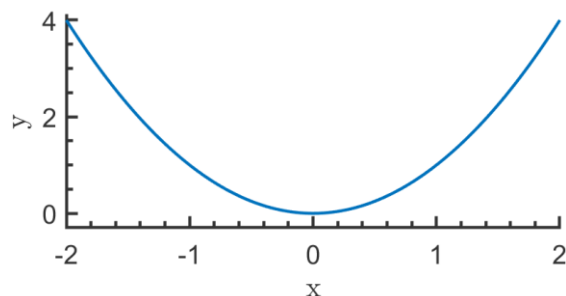


Рис. 1.3. Подпись координатных осей в текстовом режиме

Подписи осей отобразились, как обычный текст. Если мы хотим указать MATLAB, что подписи являются математическими символами, то необходимо воспользоваться режимом интерпретации LaTeX. Для этого вызов команд подписи осей нужно модифицировать как показано ниже:

```
xlabel('$ x $','Interpreter','latex')  
ylabel('$ y $','Interpreter','latex')
```

В результате получится график параболы с подписанными осями, как показано на рис. 1.4.

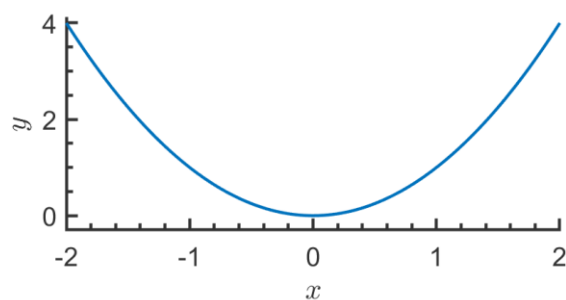


Рис. 1.4. Подпись координатных осей в режиме интерперетатора LaTeX

Для добавления заголовка к графику используется команда `title`. Текст заголовка располагается вверху окна графика. Добавим заголовок к графику, показанному на рис. 1.4:

```
title('График параболы')
```

В результате получится график параболы с указанным заголовком (рис. 1.5).

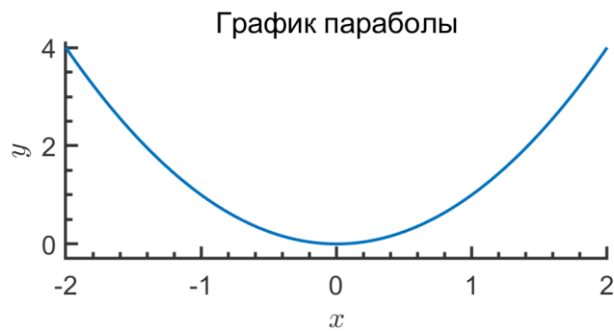


Рис. 1.5. Добавление заголовка к графику

Иногда бывает полезным добавить линии сетки на график. Для этого в MATLAB используется команда `grid on`. Для удаления сетки используется команда `grid off`.

Добавим сетку к графику параболы из рис. 1.5:

```
grid on;
```

В результате на графике отобразится сетка как показано на рис.1.6.

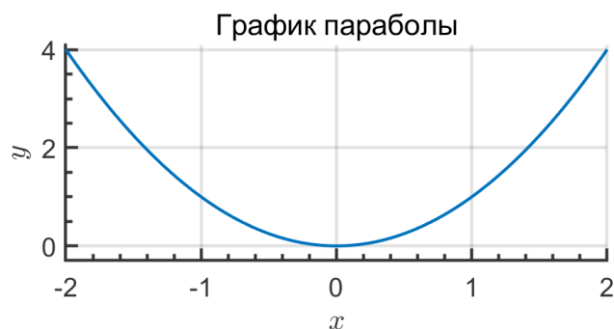


Рис. 1.6. Добавление сетки к графику

В MATLAB возможно добавление легенды к графику. Это может быть полезно при отображении нескольких графиков в одном окне. Для добавления легенды к графику используется команда `legend`.

Добавим легенду к графику параболы из рисунка 1.6, при этом будет использован режим интерпретации LaTeX:

```
legend('$ x^2 $','Interpreter','latex');
```

В результате на графике появится легенда как показано на рис. 1.7.

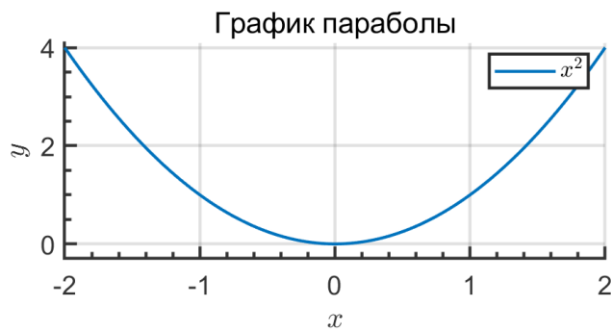


Рис. 1.7. Добавление легенды к графику

Функция `plot` может принимать дополнительные параметры: `plot(x,y,s)`. Строка `s` может содержать до трех символов (табл. 1.3), которые определяют цвет, стиль линии, а также вид точек графика.

Таблица 1.3

Управление функцией `plot`

Тип линии	
Непрерывная	-
Штриховая	--
Двойной пунктир	:
Штрихпунктирная	-. .
Тип точки	
Точка	.
Плюс	+
Звездочка	*
Кружок	o
Цвет	
Фиолетовый	m
Голубой	c
Красный	r
Зеленый	g
Синий	b
Черный	k

### Функция `stem`

Функция `stem` предназначена для вывода графика элементов одномерного массива в виде вертикальных линий, которые заканчиваются в точках графика, помечаемых кружочком. Функция `stem` имеет следующий синтаксис:

```
stem(y)
stem(x,y)
stem(x,y,s)
```



Первый вариант функции строит зависимость значений элементов массива от номера элемента, вторая зависимость –  $y(x)$ , а третья – аналогична второй, но позволяет задавать цвет и стиль линий с помощью строки  $s$ . Правила задания стиля линий аналогичны функции `plot`. Пример:

```
x = -pi:pi/25:pi
y = sin(x)
stem(y)           % Результат показан на рис. 1.8,а
stem(x, y)       % Результат показан на рис. 1.8,б
stem(x, y, 'r')  % Результат показан на рис. 1.8,в
```

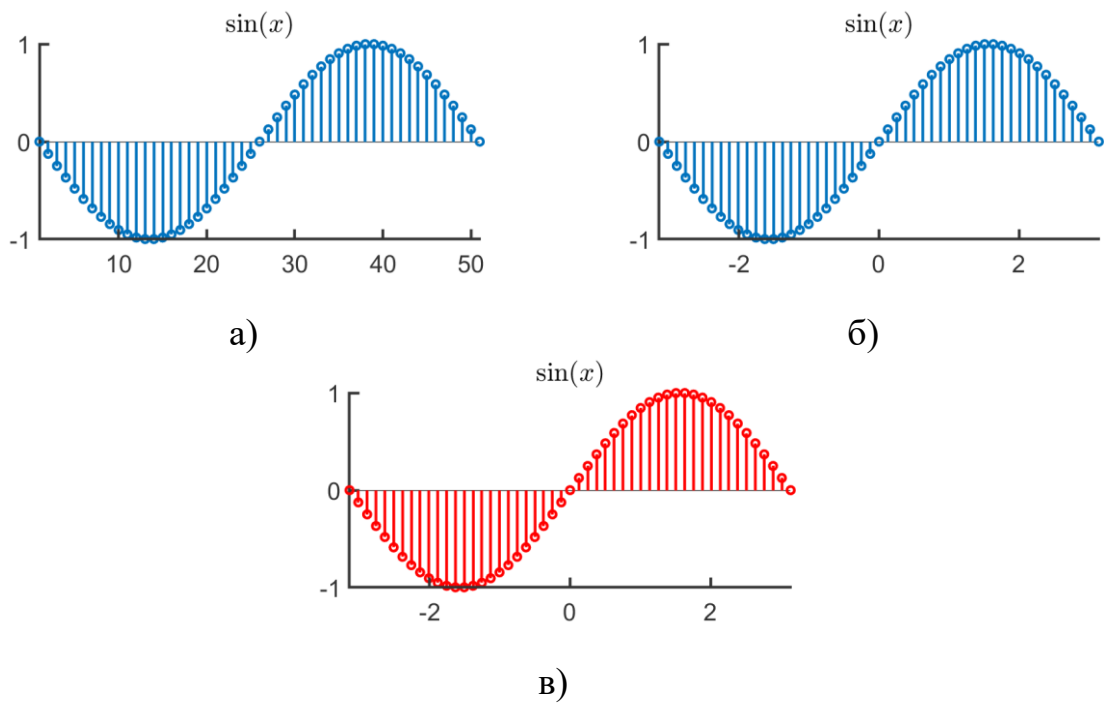


Рис. 1.8. Демонстрация работы функции `stem`

**Функция `figure`**

Функция `figure` создает новое окно для вывода графика, т.е. дальнейший вывод графической информации будет осуществляться в данном окне, пока не встретится вызов следующей функции `figure`.

При вызове функции `figure` можно при помощи свойства `'Position'` регулировать форму области отображения графика. Ниже показан синтаксис, который используется для задания свойства `'Position'` у фигуры:

```
figure('Position', [X Y W H])
```

На рис. 1.9 объясняется смысл параметров  $x$   $y$   $w$  и  $h$ . Видно, что  $x$  и  $y$  задают положение в пикселях нижнего левого угла фигуры, относительно экран-ных координат, а  $w$  и  $h$  определяют ширину и высоту фигуры, соответственно.

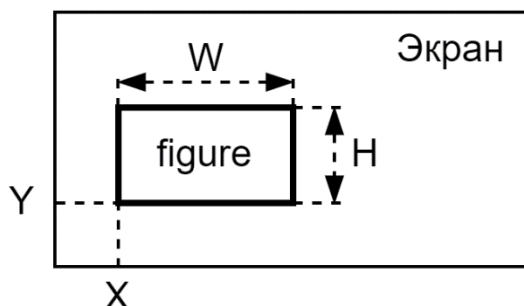


Рис. 1.9. Параметры свойства 'Position' функции `figure`

### Функция `subplot`

Функция `subplot` предназначена для разбиения активного окна (`figure`) на области размером  $m \times n$  и выбора активной области  $p$  для вывода графических данных. Функция `subplot` имеет следующий формат:

```
subplot(mnp);  
subplot(m,n,p);
```

где параметр  $m$  указывает, на сколько частей разбивается окно по вертикали,  $n$  – по горизонтали,  $p$  определяет номер области, в которую будут выводиться графические данные.

Если числа  $n$ ,  $m$  и  $p$  находятся в диапазоне от 1 до 9, то они обычно пишутся слитно (первый вариант записи функции), в противном случае они разделяются запятыми (второй вариант). Верхняя левая область имеет номер 1, правая нижняя –  $m \times n$ .

Например, команда `figure; subplot(321)` создаст в окне `figure` шесть областей для вывода графиков (как показано на рис. 1.10): три строки (параметр  $m$ ), два столбца (параметр  $n$ ) и делает активной область с номером 1 (параметр  $p$ ). Для того, чтобы сделать активной область с номером два, необходимо выполнить команду `subplot(321)`.

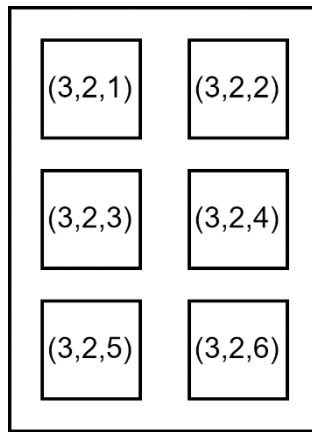


Рис. 1.10. Разбиение области вывода на шесть подобластей с использованием функции `subplot(3,2,1)`

Пример использования функции `subplot`:

```
n = 0:99
x = sin(0.3*n)
y = cos(0.4*n)
subplot(211)
plot(x)
subplot(212)
plot(y)
```

Данная программа разбивает окно вывода на две части по вертикали и выводит график  $x$  в верхней части, а график  $y$  – в нижней. Результат работы программы показан на рис. 1.11.

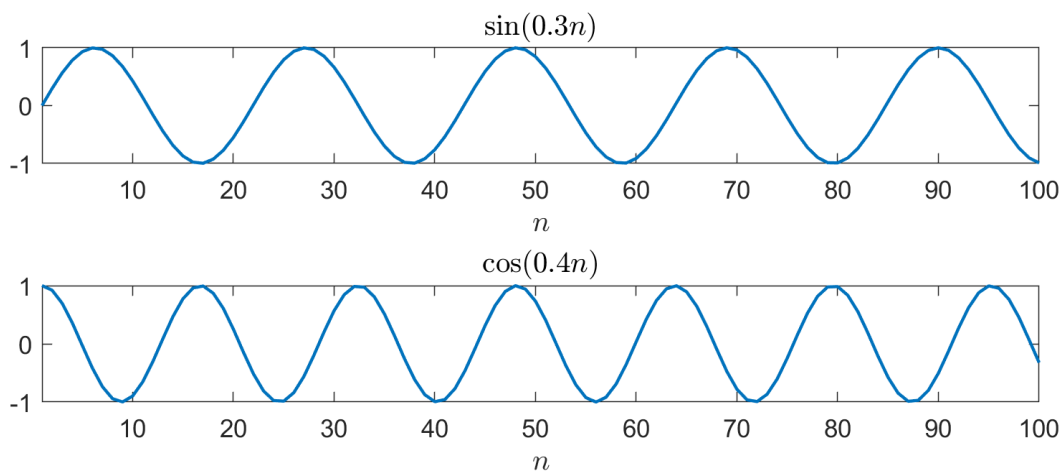


Рис. 1.11. Результат работы программы, демонстрирующей работу функции `subplot`

### ***Отображение нескольких графиков в одном окне***

Существует несколько методов для отображения двух и более графиков в одном окне. Например, данную задачу можно решить при помощи использования функции `plot` или команд удержания – `hold on` и `hold off`.

Для отображения двух и более графиков в одном окне при помощи функции `plot` необходимо ввести пары векторов непосредственно в команду `plot`.

Например, команда `plot (x1,y1,x2,y2,x3,y3)` генерирует три графика:  $y_1$  – функция от  $x_1$ ,  $y_2$  – функция от  $x_2$ ,  $y_3$  – функция от  $x_3$ . При этом длина векторов из одной пары должна быть одинаковой.

При выводе в одном окне нескольких графиков MATLAB автоматически отображает их в разных цветах, чтобы графики можно было легче идентифицировать. Также существует возможность добавить спецификаторы линии после каждой пары. Например, команда

```
plot(x,y,'-k',a,b,':r',c,d,'-.b')
```

изображает график зависимости  $y$  от  $x$  сплошной черной линией,  $b$  от  $a$  – пунктирной красной линией,  $a$  от  $c$  – штрихпунктирной синей линией.

Пример отображения графиков трех функций в одном окне при помощи функции `plot`:

```
x = -1.5:0.01:1.5;  
y1 = x.^2;  
y2 = abs(x);  
y3 = sqrt(abs(x));  
plot(x,y1,'-k',x,y2,':r',x,y3,'-.b');
```

На рис.1.12 показан результат выполнения приведенного кода.

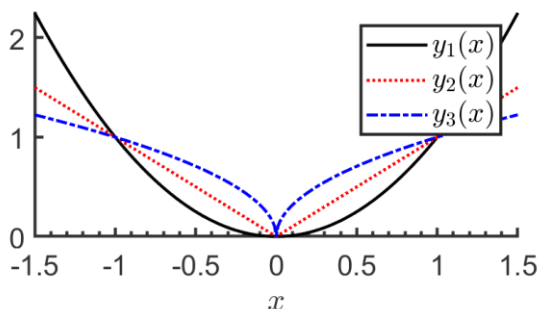


Рис. 1.12. Отображение нескольких графиков в одном окне вывода

Вторым методом отображения нескольких графиков в одном окне является использование команд удержания `hold on` и `hold off`. Команда `hold on` позволяет

сохранить содержимое открытого окна figure. Команда hold off отключает функцию сохранения графической информации в текущем окне figure. Второй и последующие графики добавляются к предыдущим при помощи команд отображения графиков (stem, plot и т.д.). После применения команды hold off каждая новая команда отображения графика будет затирать предыдущий график.

Пример отображения графиков трех функций в одном окне при помощи команды hold on:

```
x = -1.5:0.01:1.5;
y1 = x.^2;
y2 = abs(x);
y3 = sqrt(abs(x));
figure;
hold on;
plot(x,y1, '-k');
plot(x,y2, ':r');
plot(x,y3, '-.b');
```

Результат выполнения данного кода показан на рис.1.12.

### **Базовые сигналы ЦОС: единичный скачок**

Единичный скачок (рис. 1.13) описывается выражением:

$$u(n) = \begin{cases} 1, & n \geq 0, \\ 0 & \text{иначе.} \end{cases}$$

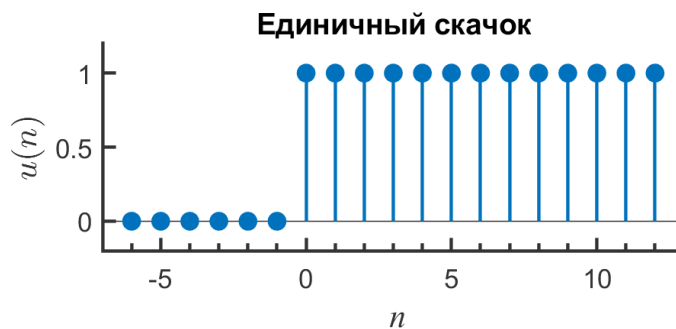


Рис. 1.13. Единичный скачок

Единичный скачок применяют для аналитического описания сигналов, которые начинаются в момент времени  $n = 0$ . Другими словами, если необходимо, чтобы сигнал имел нулевые значения для всех  $n < 0$ , то достаточно умножить его на единичный скачок  $u(n)$ . На рис. 1.14 показан пример такого использования единичного скачка.

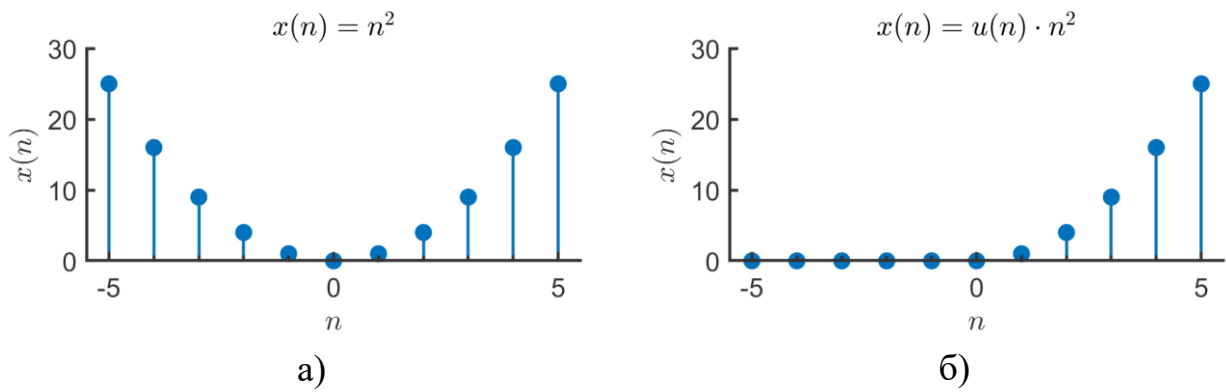


Рис. 1.14. Применение единичного скачка: а) график дискретного сигнала  $x(n) = n^2$ ; б) результат применения единичного скачка  $u(n)$  для ограничения длительности  $x(n)$

Если из единичного скачка вычесть его сдвинутой версию, то можно сформировать импульс, который позволит выделить определенный временной интервал. Так, например, сигнал

$$x(n) = u(n) - u(n - 21)$$

равняется единице в интервале  $0 \leq n \leq 20$  и нулю во всех остальных случаях (рис. 1.15).

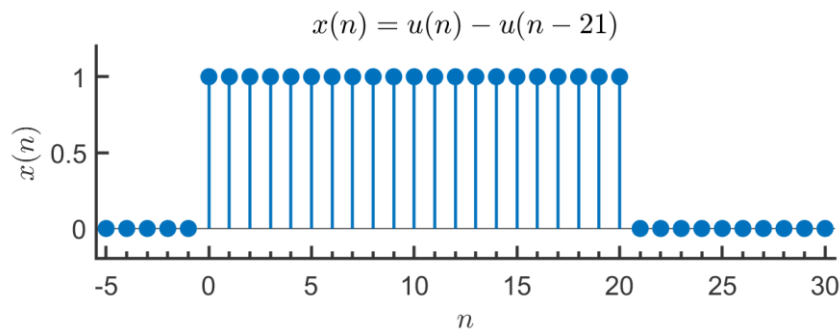


Рис. 1.15 – Формирование импульса с использованием единичного скачка

### **Единичный импульс (дельта-функция)**

Единичный импульс является базовым сигналом в цифровой обработке сигналов (рис. 1.2). Математически единичный импульс определяется как:

$$\delta(n) = \begin{cases} 1, & n = 0, \\ 0 & \text{иначе.} \end{cases}$$

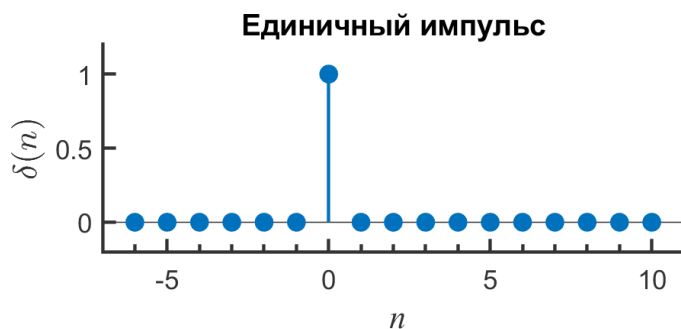


Рис. 1.2. Единичный импульс

единичный импульс обладает несколькими важными свойствами:

1. *Умножение на единичный импульс*: если сигнал  $x(n)$  умножить на сдвинутый на  $m$  отсчетов единичный импульс, то в результате получится такой же единичный импульс, но который будет иметь амплитуду, совпадающую со значением сигнала  $x(n)$  в момент времени  $n = m$ , т.е.

$$x(n)\delta(n - m) = x(m)\delta(n - m). \quad (1.1)$$

2. *Свойство просеивания*: исходя из (1.1) можно прийти к заключению, что любой дискретный сигнал можно записать как линейную комбинацию сдвинутых единичных импульсов:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n - k), \quad (1.2)$$

где каждый член суммы  $x(k)\delta(n - k)$  сдвинутым на  $k$  отсчетов единичным импульсом с амплитудой  $x(k)$ .

3. *Взаимосвязь между  $\delta(n)$  и  $u(n)$* : единичный скачек выражается через единичный импульс следующим образом:

$$u(n) = \sum_{k=-\infty}^n \delta(k), \quad (1.3)$$

В свою очередь единичный импульс можно выразить, как разницу между двумя единичными скачками:

$$\delta(n) = u(n) - u(n - 1). \quad (1.4)$$

### **Синусоидальный сигнал**

Синусоидальные сигналы имеют важное значение в теории и применении ЦОС, поэтому мы уделим им особое внимание. Также на примере синусоидального сигнала удобно вводятся понятия, связанные с процессом дискретизации непрерывных или, как еще говорят, аналоговых сигналов. Возьмем непрерывный синусоидальный сигнал с единичной амплитудой и нулевой фазой, который имеет частоту  $F$ :

$$x(t) = \sin(2\pi Ft). \quad (1.5)$$

Представим, что мы хотим получить дискретную версию этого сигнала. Дискретизация означает, что из всех значений  $x(t)$  мы оставляем лишь конечное множество. Вместо непрерывного аргумента  $t$  появляется сетка дискретных значений:

$$t \rightarrow n\Delta T, \quad (1.6)$$

где  $n$  – номер отсчета, а  $\Delta T$  – шаг между двумя соседними отсчетами. Параметр  $\Delta T$  также называют интервалом дискретизации. На практике обычно указывают не сам интервал дискретизации, а величину обратную ему – **частоту дискретизации**:

$$f_s = 1/\Delta T. \quad (1.7)$$

В англоязычной литературе параметр  $f_s$  называют *sampling rate* т.е. «скорость получения выборок». Этот термин лучше отражает сущность понятия, поскольку  $f_s$  показывает сколько отсчетов образуется в результате дискретизации одной секунды непрерывного сигнала. Частота дискретизации звуковых сигналов как правило выбирается из ряда 8 кГц, 16 кГц, 32 кГц, 48 кГц.

Подставляя выражения (1.6) и (1.7) в (1.5) получаем:

$$x(n\Delta T) = \sin\left(2\pi \frac{F}{f_s} n\right). \quad (1.8)$$



Выражение (1.8) неудобно тем, что аргументом является вещественное число  $n\Delta T$ , поэтому для удобства в качестве аргумента используют индекс  $n$ , а значение  $\Delta T$  опускают. Таким образом, дискретный сигнал – это индексированная последовательность чисел  $x(n)$ . Такой подход удобен для практической реализации систем цифровой обработки сигналов, поскольку сигнал можно представлять в виде простого массива чисел. Единственное, что надо помнить, для корректного отображения и восстановления сигнала  $x(t)$  по его дискретным отсчетам  $x(n)$  значение интервала дискретизации  $\Delta T$  (или частоты дискретизации  $f_s$ ) критически важно. Поскольку, если сама по себе последовательность  $x(n)$  не говорит о том, какое именно физическое время разделяет два её соседних отсчета. В итоге дискретная синусоида записывается, как

$$x(n) = \sin\left(2\pi \frac{F}{f_s} n\right).$$

Согласно критерию Найквиста, частота дискретизации синусоиды должна быть как минимум в два раза больше частоты синусоиды, т.е.

$$f_s \geq 2F. \quad (1.9)$$

Поэтому дискретную синусоиду обычно записывают в виде

$$x(n) = \sin(2\pi f n), \quad (1.10)$$

где  $f = F/f_s$  – *нормированная частота сигнала*. Учитывая критерий Найквиста нормированная частота должна находиться в диапазоне от 0 до 0,5.

Иногда дискретную синусоиду записывают с использованием нормированную круговую частоту  $\omega = 2\pi f$ , поскольку в этом случае выражение носит более простой характер

$$x(n) = \sin(\omega n). \quad (1.11)$$

В общем случае у дискретной синусоиды помимо параметра частоты  $f$  есть параметр амплитуды  $A$  и фазы  $\varphi$ :

$$x(n) = A \sin(2\pi f n + \varphi). \quad (1.12)$$

Приведем несколько примеров дискретизации сигнала. Предположим, что мы дискретизируем синусоиду (1.5) с частотой  $f_0 = 0,5$  Гц. Если установить частоту дискретизации равной  $f_s = 8$  Гц, то получим дискретный сигнал

$$x(n) = \sin\left(2\pi \frac{f_0}{f_s} n\right) = \sin\left(2\pi \frac{0,5}{8} n\right) = \sin(2\pi \cdot 0,0625 \cdot n),$$

который показан на рис. 1.16.

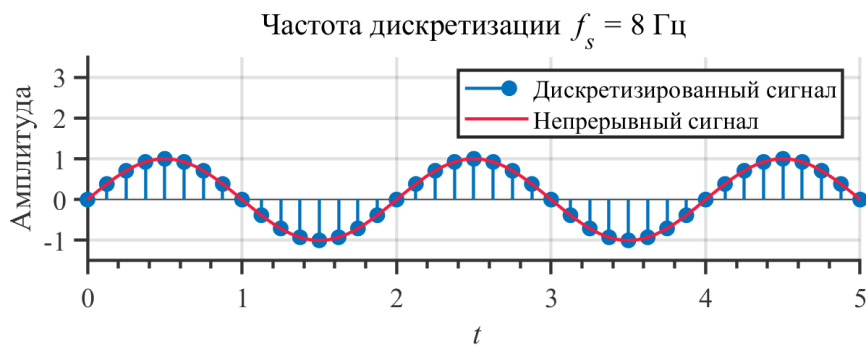


Рис. 1.16. Дискретная синусоида (0,5 Гц) с частотой дискретизации  $f_s = 8$  Гц

Если для той же синусоиды с частотой 0,5 Гц применить частоту дискретизации  $f_s = 2$  Гц, то получим уже другое дискретное описание сигнала

$$x(n) = \sin\left(2\pi \frac{f_0}{f_s} n\right) = \sin\left(2\pi \frac{0,5}{2} n\right) = \sin(2\pi \cdot 0,25 \cdot n),$$

которое представлено на рис. 1.17.

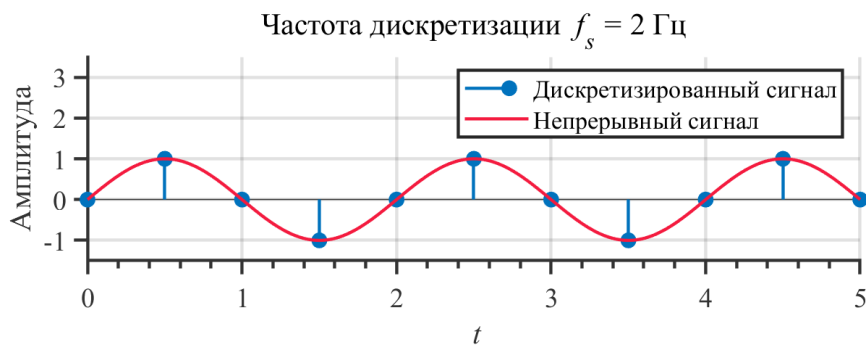


Рис. 1.17. Дискретная синусоида (0,5 Гц) с частотой дискретизации  $f_s = 2$  Гц

Если же выполнить дискретизацию рассматриваемой синусоиды с частотой  $f_s = 0,6$  Гц, то получим сигнал, показанный на рис. 1.18. Из полученного графика видно, что дискретный сигнал имеет частоту, которая отличается от частоты исходного непрерывного сигнала. Это произошло, поскольку критерий Найквиста (1.9) в данном случае не был выполнен.

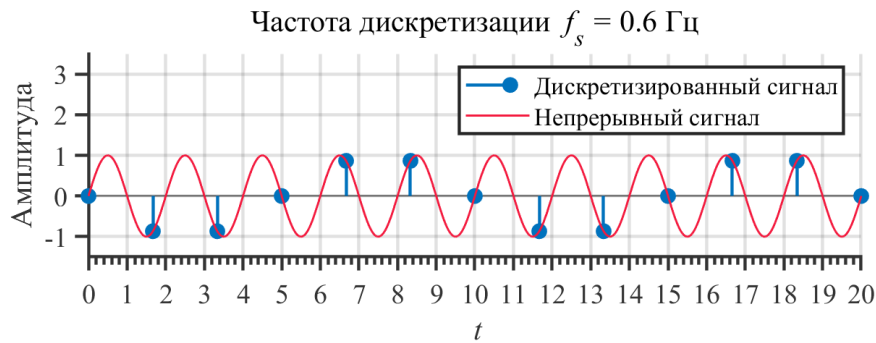


Рис. 1.18. Дискретная синусоида (0,5 Гц) с частотой дискретизации  $f_s = 0,6$  Гц

### **Комплексная экспонента**

В природе многие явления описываются экспоненциальными зависимостями, поэтому и в ЦОС экспоненты имеют важное значение. Дискретная комплексная экспонента определяется выражением

$$x(n) = z^n, \quad (1.13)$$

где  $z \in \mathbb{C}$ . Представим число  $z$  в полярной системе координат

$$z = re^{j\omega},$$

где  $|z| = r$ ,  $\arg z = \omega$ ,  $j^2 = -1$ .

В этом случае комплексную экспоненту (1.13) можно записать как:

$$x(n) = z^n = (re^{j\omega})^n = r^n e^{j\omega n} = r^n (\cos \omega n + j \sin \omega n).$$

Для получения последнего выражения использовалась формула Эйлера:

$$e^{j\omega n} = \cos \omega n + j \sin \omega n. \quad (1.14)$$

Частными случаями дискретной комплексной экспоненты  $z^n$  являются следующие четыре функции:

1) Константная функция при  $z = 1e^{j0}$ :

$$x(n) = k1^n = k;$$

2) Вещественная экспонента при  $z = re^{j0}$ :

$$x(n) = r^n \quad (1.15)$$

3) Синусоида при  $z = 1e^{j\omega}$ :

$$\begin{aligned} x(n) &= \operatorname{Re}\{z^n\} = \operatorname{Re}\{1^n e^{j\omega n}\} = \cos \omega n, \\ x(n) &= \operatorname{Im}\{z^n\} = \operatorname{Im}\{1^n e^{j\omega n}\} = \sin \omega n; \end{aligned} \quad (1.16)$$

4) Затухающая/возрастающая синусоида при  $z = re^{j\omega}$ :

$$\begin{aligned} x(n) &= \operatorname{Re}\{z^n\} = \operatorname{Re}\{r^n e^{j\omega n}\} = r^n \cos \omega n, \\ x(n) &= \operatorname{Im}\{z^n\} = \operatorname{Im}\{r^n e^{j\omega n}\} = r^n \sin \omega n. \end{aligned} \quad (1.17)$$

Для иллюстрации выражений (1.15)–(1.17) на рис. 1.19 показаны различные варианты графиков комплексных экспонент. Можно видеть, что характер графика определяется значением модуля основания экспоненты  $|z|$ . При  $|z| < 1$  (рис. 1.19, а–б) последовательность имеет затухающий характер. При  $|z| = 1$  сигнал не затухает и не возрастает, а остается стабильным во времени. При  $|z| > 1$  сигнал возрастает. Также можно видеть, что если  $z$  является комплексным значением, т.е. имеет и действительную и мнимую части, то результирующая последовательность  $z^n$  имеет колебательный характер.

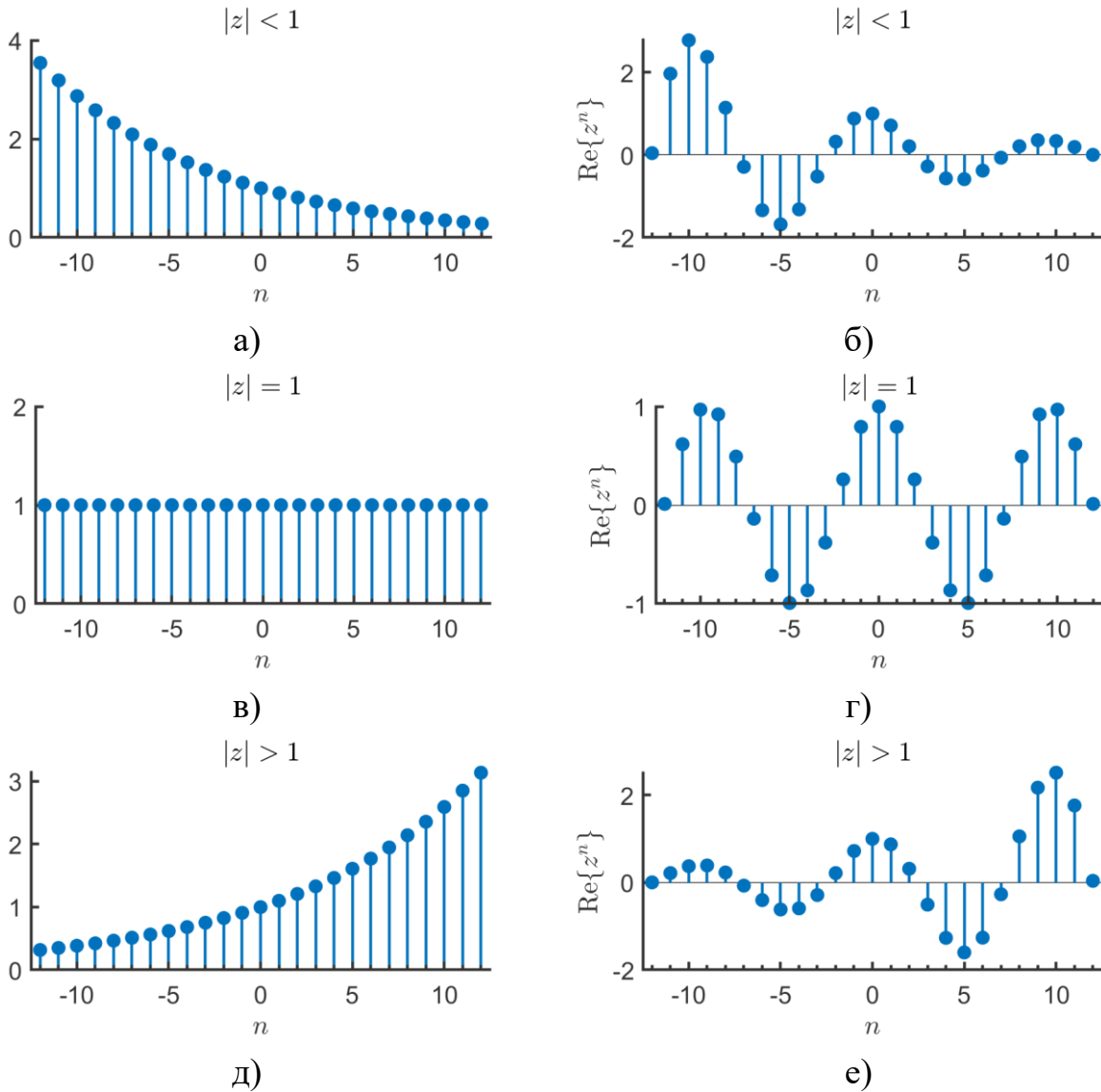


Рис. 1.19. График комплексных экспонент  $z^n$ : а)  $z = 0,9$ ; б)  $z = 0,9e^{j\pi/2}$ ; в)  $z = 1$ ; г)  $z = e^{j\pi/2}$ ; д)  $z = 1,1$ ; е)  $z = 1,1e^{j\pi/2}$

На рис. 1.20 показан трехмерный график комплексной экспоненты (1.13) для случая  $z = e^{j3\pi/50}$ . По оси  $x$  откладывается номер отсчета  $n$ , по оси  $y$  действительная часть сигнала  $\text{Re}\{x(n)\}$ , а по оси  $z$  мнимая часть сигнала  $\text{Im}\{x(n)\}$ .

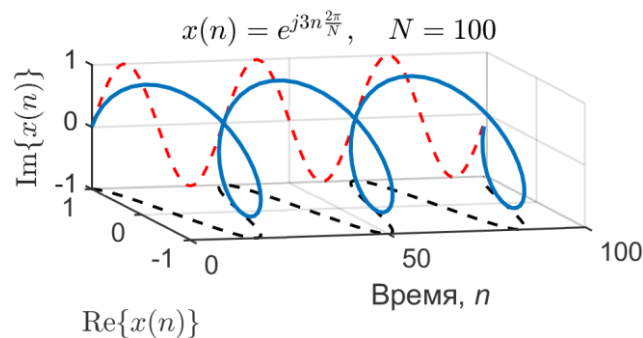


Рис. 1.20. График комплексной экспоненты с частотой  $\omega = 3\pi/50$

В соответствии с (1.16) проекция комплексной экспоненты на плоскость  $xu$  представляет из себя функцию косинус. Проекцией на плоскость  $xz$  является синус, а проекцией комплексной экспоненты на плоскость  $yz$  будет единичная окружность.

Фундаментальную значимость для ЦОС также представляет то, что функции синус и косинус могут быть представлены, как линейные комбинации комплексных экспонент. Покажем это, взяв за основу формулу Эйлера (1.14). Вначале посмотрим, что произойдет, если мы изменим знак аргумента экспоненты (1.14):

$$e^{-j\omega n} = \cos(-\omega n) + j \sin(-\omega n). \quad (1.18)$$

Учитывая то, что косинус четная функция, а синус нечетная, мы можем переписать (1.18) в следующем виде

$$e^{-j\omega n} = \cos(\omega n) - j \sin(\omega n). \quad (1.19)$$

Теперь если мы сложим (1.14) и (1.19), то получим, что

$$\cos(\omega n) = \frac{1}{2}(e^{j\omega n} + e^{-j\omega n}). \quad (1.20)$$

Аналогично, если найти разность (1.14) и (1.19), то легко установить, что

$$\sin(\omega n) = \frac{1}{2j}(e^{j\omega n} - e^{-j\omega n}). \quad (1.21)$$

Еще раз отметим, что переменная  $n$  в приведенных формулах принимает только целые значения. Это подводит к важным отличиям в свойствах непрерывных и дискретных комплексных экспонент. Эту разницу можно увидеть, если рассмотреть комплексную экспоненту (1.13), у которой  $|z| = 1$ , а частота равна  $\omega + 2\pi$ :

$$x(n) = e^{j(\omega+2\pi)n} = e^{j\omega n} \underbrace{e^{j2\pi n}}_{=1} = e^{j\omega n}.$$

Так, что при рассмотрении дискретных экспонент мы должны ограничиться частотами в интервале от 0 до  $2\pi$ .

### **Периодические дискретные сигналы**

Важным классом дискретных сигналов являются *периодические* последовательности. Дискретный сигнал считается периодическим только если существует целое число  $N \neq 0$  такое, что

$$x(n) = x(n + N), \quad \forall n \in \mathbb{Z}, \quad (1.22)$$

где  $N$  – период (обязательно целый).

Если сигнал имеет период  $N$ , то его периодом также будет и значение  $2N$ ,  $3N$  и любое значение  $kN$ , где  $k$  – целое число. **Основным периодом** называется наименьшее значение  $N_0$ , удовлетворяющее условию (1.22).

Используя определение (1.22) мы можем определить период  $N$  дискретной синусоиды (1.11):

$$\sin(\omega n) = \sin(\omega(n + N)), \quad \forall n \in \mathbb{Z}, \quad (1.23)$$

Равенство (1.23) выполняется только если существует  $k \in \mathbb{N}$  такое, что  $\omega N = 2\pi k$ . Основным периодом дискретной синусоиды тогда определяется выражением:

$$N_0 = \min_{\substack{k \in \mathbb{N} \\ (\frac{2\pi k}{\omega}) \in \mathbb{N}}} \left\{ \frac{2\pi}{\omega} k \right\}. \quad (1.24)$$

### **Генерирование периодических сигналов**

Предположим, что необходимо сгенерировать периодический сигнал общего вида. Пусть также задан закон изменения сигнала в пределах одного периода  $S_b(\alpha)$ , где  $\alpha$  – фазовое значение находящееся в диапазоне от 0 до  $2\pi$ . Для примера рассмотрим «смещенный» синус:

$$S_b(\alpha) = \begin{cases} 0,5 \left( 1 - \cos \left( \frac{1+r}{2} \alpha \right) \right), & 0 \leq \alpha < \frac{2\pi}{1+r}, \\ 0,5 \left( 1 - \cos \left( \frac{1+r}{2r} \alpha + \frac{r-1}{r} \pi \right) \right), & \frac{2\pi}{1+r} \leq \alpha < 2\pi, \end{cases} \quad (1.25)$$

Степень «смещенности» синуса определяется параметром  $r$ . При  $r = 1$  получается чистый синус. Обратите внимание, что если в качестве аргумента в  $S_b(\alpha)$  попадет  $\alpha > 2\pi$ , то необходимо вычесть из него  $2\pi$  и только затем произвести вычисление по формуле (1.25). В случае, если после первого вычитания окажется, что  $\alpha$  все ещё больше  $2\pi$  вычитание следует повторить. И так до тех пор, пока  $\alpha$  не попадет в диапазон  $0 \leq \alpha < 2\pi$ .

Допустим, что у нас стоит задача сгенерировать сигнал с частотой дискретизации  $f_s$ , частотой основного тона  $f_0$  и длительностью  $T$ .

Получим вначале выражения для непрерывной версии сигнала

$$S(t) = S_b(2\pi f_0 t), \quad t \in [0; T]. \quad (1.26)$$

Дискретизация сигнала определяется заменой (1.6), следовательно, дискретная версия смещенного синусоидального сигнала рассчитывается как:

$$S(n) = S_b \left( \frac{2\pi f_0}{f_s} n \right), \quad n = 0, 1, \dots, \frac{T}{f_s} - 1. \quad (1.27)$$

### **Генерирование частотно-модулированного (ЧМ) сигнала**

Частота сигнала не обязательно должна оставаться постоянной на протяжении всего временного интервала. Сигналы, которые имеют периодическую природу, но у которых во времени происходит изменение их частоты называются **частотно-модулированными (ЧМ)**.

Параметрами (ЧМ) сигнала являются:  $f_m$  – частота модуляции (Гц);  $k_f$  – индекс модуляции;  $T$  – длительность сигнала в секундах. Начнем с того, что опишем непрерывную версию сигнала:



$$S(t) = S_b(\alpha(t)), \quad t \in [0; T]. \quad (1.28)$$

где

$$\alpha(t) = 2\pi \int_0^t \phi(\tau) d\tau. \quad (1.29)$$

$$\phi(t) = f_0(1 + k_f \cos(2\pi f_m t)). \quad (1.30)$$

Выполним дискретизацию (1.6) модели сигнала (1.28) путем замены при этом интеграл заменим на сумму:

$$S(n) = S_b(\alpha(n)), \quad (1.31)$$

$$\alpha(n) = 2\pi \sum_{m=0}^n \frac{f_0}{f_s} \left( 1 + k_f \cos\left(\frac{2\pi f_m}{f_s} m\right) \right). \quad (1.32)$$

Заметьте, что если  $k_f = 0$ , то  $\alpha(n) = \frac{2\pi f_0}{f_s} n$ , т.е. по сути (1.32) повторяет (1.27).

### **Генерирование амплитудно-модулированного (АМ) сигнала**

Помимо частоты во времени может изменяться и амплитуда сигнала. Такие сигналы называются **амплитудно-модулированными (АМ)**. Параметрами АМ-сигнала являются:  $f_{am}$  – частота модуляции (Гц);  $k_a$  – индекс модуляции. Для генерирования АМ-сигнала используется выражение:

$$s_{AM}(n) = S_b(n) \cdot \left( 1 + k_a \cos\left(\frac{2\pi f_{am}}{f_s} n\right) \right). \quad (1.33)$$

В выражении (1.33) в качестве  $S_b(n)$  может использоваться как обычный периодический сигнал (1.27), так и ЧМ-сигнал (1.31).

### **DTMF-сигналы**

DTMF сигналы или тоны передаются при нажатии кнопок на номеронабирателе и используются, как правило, для [до]набора внутреннего номера абонента офисной АТС или для навигации по голосовому меню (IVR).

## 1.2 Порядок выполнения работы

### Задание 1 (Письменное)

*Вариант 1.*

Определите являются ли периодическими сигналы:

а)  $x_1(n) = \cos(n^2)$

б)  $x_2(n) = \sin\left(\frac{\pi n}{8}\right)$ .

В случае положительного ответа определите период сигнала.

*Вариант 2.*

Упростите выражение  $(u(n) + (-1)^n u(n)) \cdot \sin\left(\frac{\pi n}{2}\right)$ . Постройте график соответствующий данному выражению.

*Вариант 3.*

Пусть  $x(n) = \left(\frac{1}{2}\right)^n \sin\left(\frac{n\pi}{2}\right) u(n)$ . Изобразите график сигнала  $x(\langle n \rangle_5)$ .

*Вариант 4.*

Пусть  $x(n) = \left(\frac{1}{2}\right)^n \sin\left(\frac{n\pi}{2}\right) u(n)$ . Изобразите график сигнала  $x(\langle n - 2 \rangle_4)$ .

*Вариант 5.*

Определите являются ли периодическими сигналы:

а)  $x_1(n) = \cos(\sqrt{2}\pi n + 1,2)$

б)  $x_2(n) = e^{j\frac{22}{7}n}$ .

В случае положительного ответа определите период сигнала.

*Вариант 6-8.*

Для сигналов, показанных на рис. 1.21–1.23, найдите два различных аналитических выражения, которые были верными для всех  $n$ . Для описания сигнала разрешается использовать только единичный скачок  $u(n)$  и единичный импульс  $\delta(n)$ .

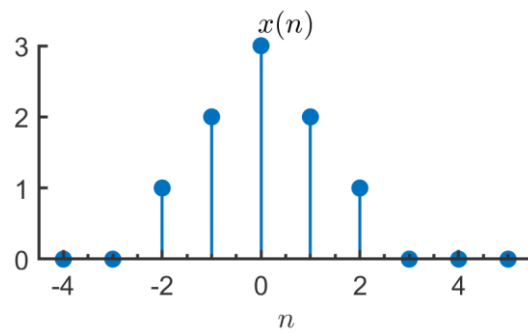


Рис. 1.21. Дискретный сигнал из задания 1 (вариант 6)

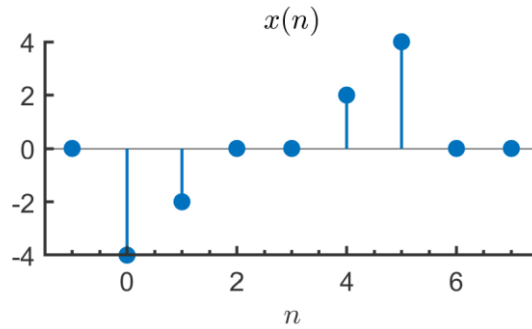


Рис. 1.22. Дискретный сигнал из задания 1 (вариант 7)

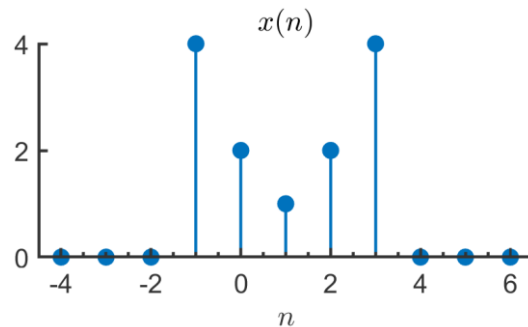


Рис. 1.23. Дискретный сигнал из задания 1 (вариант 8)

### **Задание 2**

Построить график дискретного синусоидального сигнала с периодом  $N = 8 \times n_0$ , где  $n_0$  – номер варианта.

### **Задание 3**

Построить график дискретной непериодической синусоидальной последовательности (частоту  $\omega_0$  выберите самостоятельно).

### **Задание 4**

Сформируйте и постройте графики единичного импульса и единичного скачка:

$$y(n) = \delta(n - n_0),$$

$$y(n) = u(n - n_0),$$

где  $n_0$  – номер варианта.

Для выполнения задания напишите в MATLAB функции `delta(n)` и `unit_step(n)` для расчета единичного импульса и единичного скачка соответственно.

### Задание 5

Напишите функцию для формирования синусоиды, получаемой в результате дискретизации с частотой  $f_s$  непрерывной синусоиды:

$$s(t) = A \sin(2\pi f_0 t + \varphi_0),$$

где  $A$  – амплитуда;  $f_0$  – частота синусоиды;  $\varphi_0$  – начальная фаза.

Функция должна иметь шесть входных аргументов:  $A, f, \varphi_0, F_s, t_0$  – начальное время,  $t_1$  – конечное. Выходными параметрами функции должны быть временные отметки (моменты времени в которые выполняется дискретизация синусоиды) и значения синусоиды в эти моменты.

Сформируйте синусоиду дискретного времени дискретизацией синусоиды непрерывного времени со следующими параметрами (табл. 1.4):

Таблица 1.4

Числовые значения параметров для формирования синусоид дискретного времени

Номер варианта	1	2	3	4	5	6	7	8
$f_0$ , кГц	0,9	1	3,0	1,2	4,0	1,5	0,44	0,675
$\varphi_0$	$\pi/10$	$\pi/6$	$\pi/11$	$\pi/8$	$\pi/2$	$-\pi$	$\pi/4$	$3\pi/4$
$f_s$ , кГц	8	16	11,25	16	22,5	32	44,1	48
$t_0$ , с	0	-1	-2	0	-1	-2	2	-0,5
$t_1$ , с	4	3	1	5	3	2	4	5

### Задание 6

Напишите функцию MATLAB для формирования комплексной экспоненты (1.13), которая имеет сигнатуру  $x = \text{sprlx\_exp}(z, n)$ , где  $z$  – скалярное значение, а  $n$  – вектор целочисленных значений (номера отсчетов). Используйте эту функцию для формирования сигнала вида

$$x(n) = r^n e^{j\omega n}$$

Параметры  $r$  и  $\omega$  выбираются в соответствии с вариантом (табл. 1.9\*\*).

Таблица 1.9\*\*

Параметры комплексных экспонент

Номер варианта	1	2	3	4	5	6	7	8
$r$	0,89	0,9	0,94	0,95	0,97	0,99	0,87	0,85
$\omega$	$\pi/10$	$2\pi/7$	$3\pi/11$	$5\pi/13$	$11\pi/31$	$17\pi/41$	$13\pi/43$	$7\pi/17$

Постройте график действительной и мнимой части для  $n = 0 \dots 100$ . Как сказывается изменение  $\theta$ ? Постройте график, откладывая по оси ординат действительную часть, а по оси абсцисс – мнимую (должна получиться спираль). Поэкспериментируйте с углом  $\theta$  для получения спиралей различного вида. Используя функцию `plot3` постройте трехмерный графики комплексной экспоненты.

### Задание 7

Сформируйте и постройте графики следующих последовательностей используя выражения (1.16)–(1.17):

$$\begin{aligned} x_1(n) &= \sin(\omega_1 n), & n &= 0 \dots 25, \\ x_2(n) &= \cos(\omega_1 n), & n &= -15 \dots 25, \\ x_3(n) &= a^n \sin(\omega_2 n), & n &= -20 \dots 0, \\ x_4(n) &= b^n \cos(\omega_2 n), & n &= 0 \dots 50. \end{aligned}$$

Значения параметров  $\omega_1, \omega_2, a, b$  выберите из табл. 1.10.

Таблица 1.10

Значения параметров  $\omega_1, \omega_2, a, b$

Номер варианта	1	2	3	4	5	6	7	8
$\omega_1$	$\pi/3$	$\pi/6$	$\pi/5$	$\pi/\sqrt{7}$	$\pi/8$	$\pi/13$	$\pi/15$	$2\pi/7$
$\omega_2$	$\pi/4$	$\pi/9$	$\pi/16$	$\pi/6$	$\pi/\sqrt{11}$	$\pi/3$	$\pi/5$	$\pi/\sqrt{13}$
$a$	1,1	1,15	1,4	1,3	1,2	1,05	1,05	1,5

$b$	0,89	0,9	0,94	0,95	0,97	0,99	0,98	0,96
-----	------	-----	------	------	------	------	------	------

### 1.3 Дополнительные задания

1) Сгенерируйте три периода «смещенного» синусоидального сигнала (1.25) для значений параметра  $r = 0,2$  и  $r = 0,8$  и отобразите их на одном графике.

2) Сгенерируйте АМ-сигнал (1.33) с параметрами  $f_s = 16$  кГц,  $f_{am} = 2$  Гц,  $k_a = 0,5$ , длительностью 2 секунды. В качестве  $S_b(n)$  «смещенную» синусоиду (1.25) с параметром  $r = 0,4$ . Сохраните полученный сигнал при помощи команды

```
audiowrite('am_skew_sin.wav', x, fs);
```

Как изменится звучание сигнала, если установить значение параметре  $k_a = 0$ ?

3) Сгенерируйте в MATLAB wav-файл ( $f_s = 16$  кГц), который содержит последовательность из 9 синусоидальных сигналов с частотами 659, 622, 659, 622, 494, 587, 523 и 440 Гц. Длительность каждого сигнала 0,5 секунды. Сохраните полученный сигнал при помощи команды `audiowrite()`. Продемонстрируйте сгенерированный сигнал преподавателю.

4) Напишите MATLAB-функцию для формирования односторонней затухающей экспоненты

$$x(n) = a^n u(n).$$

где  $|a| < 1$ ,  $u(n)$  – единичный скачок.

Постройте график этой функции для  $n = -10, \dots, 20$ .

5) Определите являются ли периодическими сигналы:

$$x_1(n) = \cos\left(\frac{12\pi}{5}n\right),$$

$$x_2(n) = 10 \sin\left(\left(\frac{7\pi}{12}\right)n + \sqrt{2}\right),$$

$$x_3(n) = 2 \cos(0,02n + 3).$$

В случае положительного ответа определите основной период сигнала.

6) Постройте графики сигналов из дополнительного задания 5 и отобразите их на одной фигуре. Задайте для каждого графика свой цвет и тип линии.

7) Изобразите графики следующих дискретных сигналов:

$$\begin{aligned}x_1(n) &= 2\delta(n+3) + 4\delta(n-1) + \delta(n-2) + \delta(n-3), \\x_2(n) &= u(n-2) - u(n-6), \\x_3(n) &= n(u(n) - u(n-7)).\end{aligned}$$

8) Пусть задан сигнал

$$x(n) = n(u(n) - u(n-5)), \quad n = 0, 1, \dots, 7.$$

Найдите и изобразите сигнал  $y(n) = x(\langle -n - 2 \rangle_8)$ .

9) Для сигнала  $x(n)$  показанного на рис. 1.24 изобразите графики следующих сигналов:

- а)  $x(-n)$
- б)  $x(n+6)$
- в)  $x(3-n)$

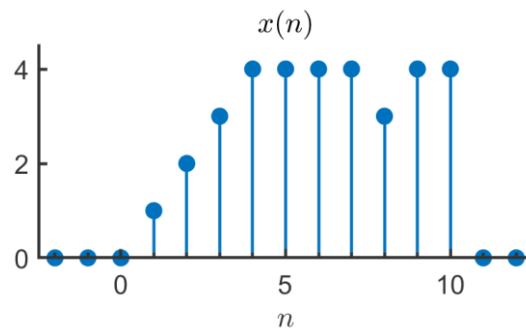


Рис. 1.24 – Дискретный сигнал

10) Пусть задан сигнал вида:

$$x(n) = \begin{cases} 0, & k < 0, \\ 3, & k = 0, \\ 1, & 1 \leq k < 9, \\ 0, & k \geq 9. \end{cases}$$

Запишите сигнал  $x(n)$  аналитически, используя базовые сигналы  $\delta(n)$  и  $u(n)$ .

11) Выполняется дискретизация сигнала с частотой дискретизации  $F_s = 4000$  Гц.

$$x(t) = 10\cos(2000\pi t) + \sqrt{2}\sin(300\pi t) + 2\cos\left(5000\pi t + \frac{\pi}{4}\right),$$

Найдите результирующий дискретный сигнал  $x(n)$ .

12) Пусть непрерывный периодический сигнал с периодом 1 мс имеет вид:

$$x_c(t) = \sum_{k=-9}^9 a_k e^{j\left(\frac{2\pi kt}{10^{-3}}\right)}.$$

В результате дискретизации сигнала  $x_c(t)$  с шагом  $T = \frac{1}{6} 10^{-3}$  получается последовательность  $x(n) = x_c(nT)$ . а) Является ли последовательность  $x(n)$  периодической и, если да, то каков её период?

13) Изобразите следующие сигналы:

а)  $u(n - 2) - u(n - 6)$

б)  $n(u(n) - u(n - 7))$

в)  $(n - 2)(u(n - 2) - u(n - 6))$

14) После дискретизации непрерывного сигнала  $\cos(100\pi t)$  был получен дискретный сигнал  $\cos(\pi n)$ . Найдите частоту дискретизации  $F_s$ . Является ли это значение единственно возможным? Поясните свой ответ.