

ЛАБОРАТОРНАЯ РАБОТА №1. Базовые сигналы в ЦОС

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 инженерных и научных расчетов Matlab (сокращение от MATrixLABoratory – матричная лаборатория). Основным объектом системы Matlab является матрица. Все вычисления система осуществляет в арифметике с плавающей точкой.

Система Matlab работает в режиме интерпретации команд и операторов, которые вводятся в ходе сеанса в командной строке, а Matlab выполняет их немедленную обработку и выдает вычисленный результат. Однако в Matlab можно запустить на выполнение заранее подготовленную последовательность команд и операторов, записанную в виде файла с расширением .m (М-файл).

М-файлы разделяются на два вида: файлы-сценарии и процедуры-функции.

Все М-файлы должны располагаться в рабочем каталоге или каталоге, зарегистрированном в списке путей системы Matlab. Для изменения рабочего каталога необходимо выбрать в меню File пункт SetPath.... При проведении лабораторной работы система разрешает изменять только рабочий каталог и не разрешает добавлять и удалять каталоги из списка путей. При выходе из системы сведения о рабочем каталоге не сохраняются, поэтому при новом запуске Matlab его необходимо ввести заново.

Файлы-сценарии

Файл-сценарий – это текстовый файл, содержащий последовательности команд и операторов. В языке нет специальных операндов для обозначения начала и конца файла-сценария, а также точки входа. Началом сценария является начало файла, а концом соответственно конец файла.

Именем сценария является имя его файла, отсюда следуют ограничения, накладываемые на имена файлов:

- 1) 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;
- 2) имя файла должно начинаться с символа A...Z или a...z либо с символа 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 процедуры-функции

Файл процедуры-функции также является текстовым файлом, содержащим последовательности команд и операторов. Отличия файла процедуры-функции от файла-сценария заключаются в следующем:

1. Первой строкой должен быть заголовок функции, имеющий следующий формат:

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

Имя функции должно совпадать с именем файла.

2. При выполнении функции она сначала компилируется во внутренний формат, после чего исполняется. В связи с этим функция выполняется на порядок быстрее файла-сценария, который работает в режиме интерпретации команд.

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` и имени функции. Такую справку содержат абсолютно все функции пакета Matlab. Например:

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

Рассмотрим функцию, вычисляющую минимальное и максимальное значения массива (файл `minmax.m`):

```
function [minim, maxim] = minmax(a)
minim = min(a)
maxim = max(a)
```

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

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

Данные в системе Matlab

В системе Matlab присутствует только один тип данных – это прямоугольный массив комплексных чисел. Для создания переменной ей просто необходимо присвоить некоторое значение. Существует несколько вариантов записи массива:

1. *Объявление пустого массива.* Объявление пустого массива, т.е. массива нулевого размера, имеет следующий вид:

```
a = []
```

2. *Объявление скаляра.* Скаляр является частным случаем матрицы, т.е. это матрица размером 1×1 . Пример:

```
a = 5  
b = -4.8  
c = 0.25 + 18i % Комплексное число
```

3. *Объявление вектора.* Для объявления вектора необходимо записать его элементы, разделенные пробелами, в квадратных скобках. Пример:

```
a = [0 1]  
b = [3.1 -6]  
c = [a 4 b] % Результат c = [0 1 4 3.1 -6]
```

4. *Объявление монотонно возрастающего или убывающего вектора.* Формат объявления монотонно возрастающего или убывающего вектора следующий: *первый_элемент* : [шаг :] *конечный_элемент*. Если шаг равен 1, то его разрешается опустить. Пример:

```
a = 1:10 % Результат b = [1 2 3 4 5 6 7 8 9 10]  
b = 4:-2:-4 % Результат b = [4 2 0 -2 -4]  
c = -1:3:10 % Результат b = [-1 2 5 8]
```

5. *Объявление массива.* Для объявления вектора необходимо записать его элементы в квадратных скобках, притом элементы строки разделяются пробелами, а строки между собой – точкой с запятой. Пример:

```
a = [1 2 3; 4 5 6; 7 8 9] % Результат  $a = \begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Bmatrix}$ 
```

Для объявления массивов также используются специальные функции. Вот некоторые наиболее часто используемые:

<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 из 100 нулей
b = ones(20)      % Матрица 20×20 единиц
c = rand(2, 50)   % Матрица 2×50 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 2×3 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 операторы

В системе реализовано два типа арифметических операций. Операции над матрицами определены в соответствии с правилами линейной алгебры, а операции над массивами выполняются поэлементно. Для обозначения операций выполняемыми над элементами массива используется знак точки«.» (табл. 1.1).

Операции Matlab

Знак	Операция
+	Сложение: $C=A+B$ Если A – скаляр, B – массив, то $C_{ij} = A + B_{ij}$. Если A – массив, B – скаляр, то $C_{ij} = A_{ij} + B$. Если A – массив, B – массив, то $C_{ij} = A_{ij} + B_{ij}$. Для выполнения операции сложения массивов они должны иметь одинаковый размер.
–	Вычитание: $C=A-B$ Если A – скаляр, B – массив, то $C_{ij} = A - B_{ij}$. Если A – массив, B – скаляр, то $C_{ij} = A_{ij} - B$. Если A – массив, B – массив, то $C_{ij} = A_{ij} - B_{ij}$. Для выполнения операции вычитания массивов они должны иметь одинаковый размер.
*	Умножение матриц: $C=A*B$ Если A – скаляр, B – массив, то $C_{ij} = A \cdot B_{ij}$. Если A – массив, B – скаляр, то $C_{ij} = A_{ij} \cdot B$. Если A – массив, B – массив, то $C_{ik} = \sum_{j=1}^n A_{ij} \cdot B_{jk}$. Для выполнения операции число столбцов первого массива должно быть равно числу строк второго массива.
.*	Поэлементное умножение: $C=A.*B$ Если A – скаляр, B – массив, то $C_{ij} = A \cdot B_{ij}$. Если A – массив, B – скаляр, то $C_{ij} = A_{ij} \cdot B$. Если A – массив, B – массив, то $C_{ij} = A_{ij} \cdot B_{ij}$. Для выполнения операции поэлементного умножения массивов они должны иметь одинаковый размер.
`	Транспонирование матрицы: A' Для действительных массивов результатом является транспонированная матрица. Для комплексных массивов транспонирование дополняется комплексным сопряжением.
.'`	Транспонирование массива: $A.'$ Для действительных и комплексных массивов строки просто заменяются столбцами. Комплексное сопряжение не выполняется.

Логические операторы. Операции отношения.

В языке Matlab используются следующие логические операторы (табл.1.2):

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. 1.3):

Таблица 1.3

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]
```

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`.

Графические возможности языка Matlab

Среда обладает богатыми возможностями для графического представления массивов как в двухмерном, так и в трехмерном виде. В пакет Matlab включены демонстрационные программы для показа возможностей системы, в том числе и графических. Для вызова демонстрационных примеров необходимо в командном окне ввести команду `demo`.

Далее будут рассмотрены графические функции, которые необходимы для выполнения лабораторных работ.

Функция `plot`

Функция `plot` имеет следующий синтаксис:

```
plot(y)
plot(x,y)
plot(x,y,s)
plot(x1,y1,s1,x1,y1,s1)
```

Функция `plot(y)` строит график элементов одномерного массива `y` в зависимости от номера элемента. Если элементы массива `y` комплексные, то строится график, каждая точка которого определяется соответствующей действительной и мнимой частями числа.

Функция `plot(x,y)` соответствует построению обычной функции, когда одномерный массив `x` соответствует значениям аргумента, а одномерный массив `y` – значениям функции (табл. 1.4).

Таблица 1.4

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 `plot(x, y, s)` аналогична `plot(x, y)`, с той разницей, что текстовая строка `s` определяет цвет и стиль линии, а также вид точек графика. Строка `s` может содержать до трех символов из табл. 1.4.

Если цвет линии не указан, он выбирается по умолчанию из шести первых цветов, с желтого до синего, повторяясь циклически.

Функция `plot(x1, y1, s1, x2, y2, s2)` позволяет объединить несколько функций `plot(x, y, s)`. Пример:

```
x = -pi:pi/100:pi
y = sin(x)
plot(y)           % Результат показан на рис. 1.1,а
plot(x, y)        % Результат показан на рис.1.1,б
```

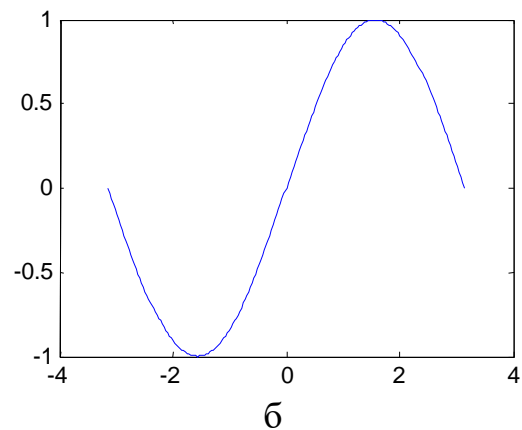
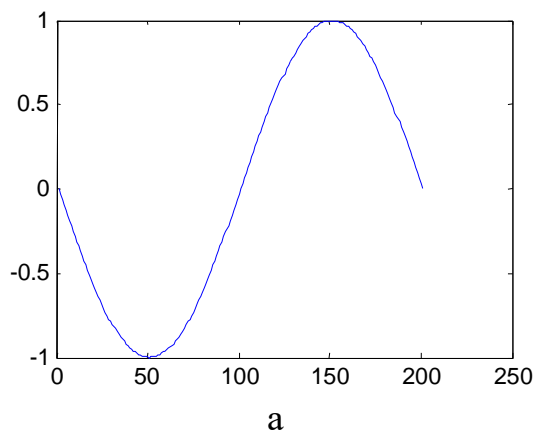


Рис. 1.1. Результат работы примера, демонстрирующего работу функции `plot`

Функция *stem*

Функция `stem` выводит график 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, за исключением того, что 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. 1.2,а
stem(x, y)        % Результат показан на рис.1.2,б
```

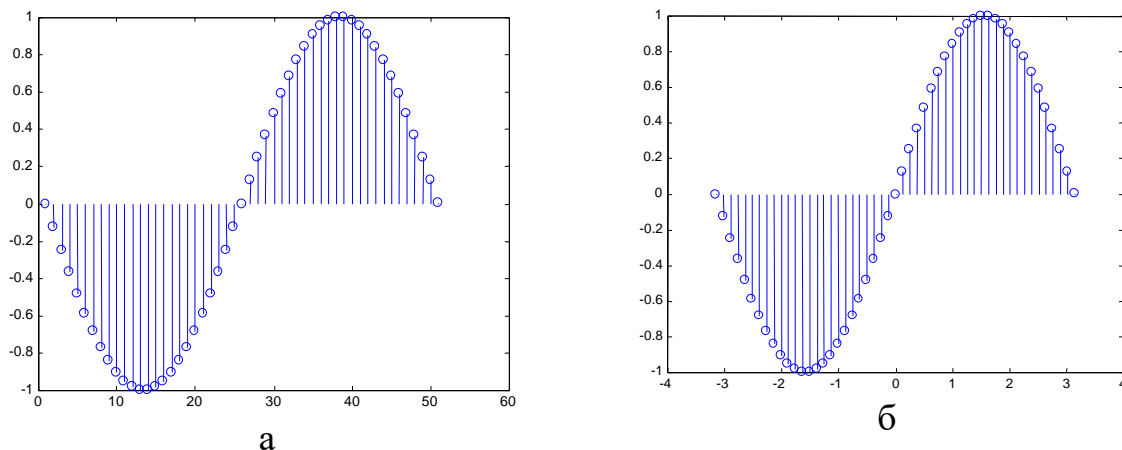


Рис. 1.2. Результат работы примера, демонстрирующего работу функции `stem`

Функция *figure*

Данная функция создает окно для вывода графика, т.е. дальнейший вывод графической информации будет осуществляться в данном окне, пока не встретится функция `figure`.

Функция *subplot*

Функция `subplot` предназначена для разбиения активного окна на области и выбора активной области для вывода графических данных. Функция `subplot` имеет следующий формат:

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

где `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). Пример:

```
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..

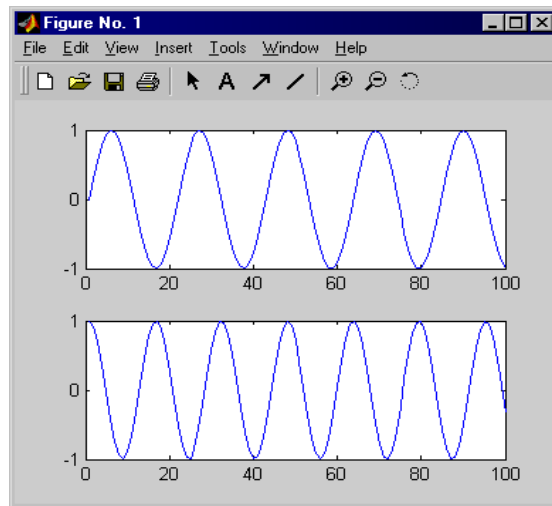


Рис. 1.3. Результат работы программы, демонстрирующей работу функции subplot

1.2 Базовые сигналы ЦОС

Наиболее важными последовательностями, которые часто используются в цифровой обработке сигналов, являются:

а) **единичный импульс** (рис. 1.2):

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

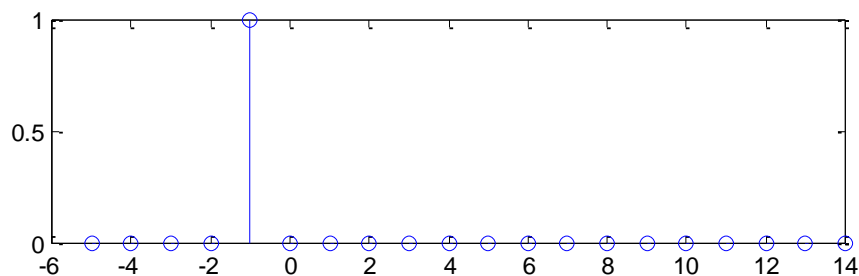


Рис. 1.2. Единичный импульс

б) **единичный скачок** (рис. 1.3):

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

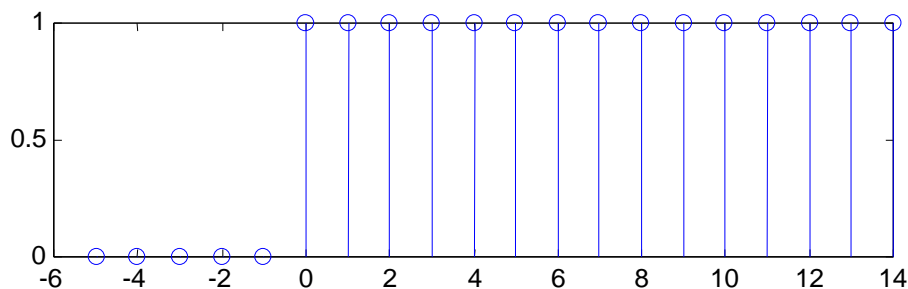


Рис. 1.3. Единичный скачок

в) **убывающая экспонента** (рис. 1.4):

$$g(n) = a^n u(n), \quad |a| < 1;$$

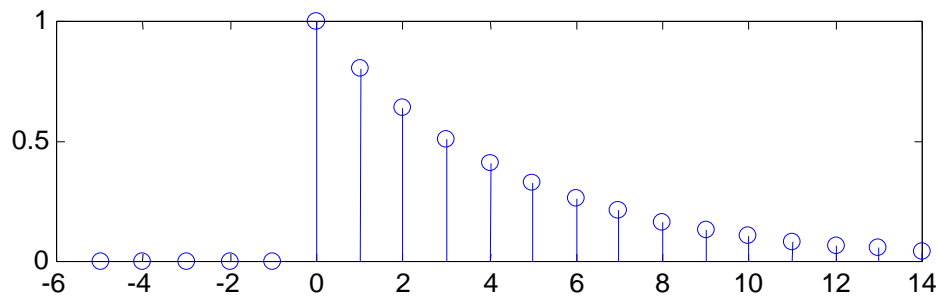


Рис. 1.4. Убывающая экспонента ($a = 0,8$)

г) **косинусоида** (рис.1.5):

$$h(n) = A \cdot \cos\left(\frac{2\pi n}{n_0} + \varphi\right) = A \cdot \cos(\omega n + \varphi);$$

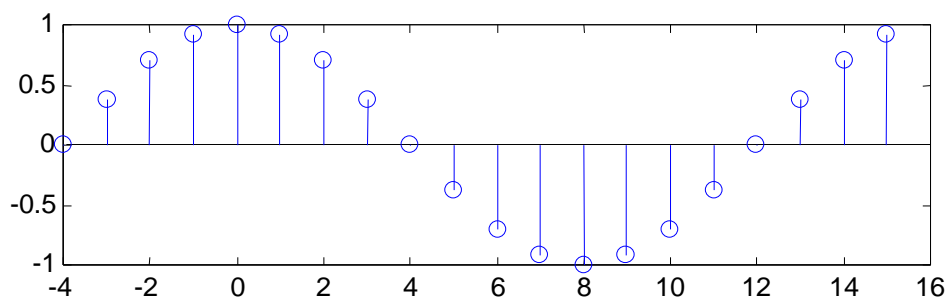


Рис. 1.5. Косинусоида $\cos\left(\frac{2\pi n}{16}\right)$

д) **комплексная экспонента**:

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

1.3 Порядок выполнения работы

1.3.1 Сформируйте и постройте графики следующих последовательностей. Используйте для этого возможности вычисления синуса или косинуса от векторного аргумента:

$$\begin{aligned} x_1(n) &= \sin(\omega_1 n), & n &= 0 \dots 25, \\ x_2(n) &= \sin(\omega_1 n), & n &= -15 \dots 25, \\ x_3(n) &= \sin\left(\omega_2 n + \frac{\pi}{2}\right), & n &= -10 \dots 10, \\ x_4(n) &= \cos(\omega_3 n), & n &= 0 \dots 50. \end{aligned}$$

Значения $\omega_1 \dots \omega_3$ выберите из табл. 1.5.

Таблица 1.5

Значения $\omega_1 \dots \omega_3$						
Номер варианта	1	2	3	4	5	6
ω_1	$\pi/13$	$\pi/14$	$\pi/15$	$\pi/16$	$\pi/17$	$\pi/18$
ω_2	π	2π	3π	π	2π	3π
ω_3	$\pi/\sqrt{19}$	$\pi/\sqrt{21}$	$\pi/\sqrt{23}$	$\pi/\sqrt{20}$	$\pi/\sqrt{22}$	$\pi/\sqrt{18}$

Упростите $x_3(n)$, чтобы не использовать тригонометрические функции.

1.3.2. Сформируйте и постройте график следующих последовательностей:

$$\begin{aligned}
 x_1(n) &= a_1 \delta(n - b_1), & n &= 1 \dots 20, \\
 x_2(n) &= a_2 \delta(n), & n &= -15 \dots 15, \\
 x_3(n) &= a_3 u(n - b_3), & n &= 300 \dots 350, \\
 x_4(n) &= a_4 u(n + b_4), & n &= -10 \dots 0.
 \end{aligned}$$

Для выполнения задания напишите в Matlab функции `delta(n)` и `unit_step(n)` для расчета дельта-функции и функции единичного скачка соответственно. Значения $a_1 \dots a_4$ и $b_1 \dots b_4$ выберите из табл. 1.6.

Таблица 1.6

Числовые значения $a_1 \dots a_4$ и $b_1 \dots b_4$						
Номер варианта	1	2	3	4	5	6
a_1	1,2	1,3	1,4	1,5	1,6	1,7
b_1	4	5	6	7	8	9
a_2	0,6	0,7	0,8	0,9	1,1	1,2
a_3	1,8	1,7	1,6	1,5	1,4	1,3
b_3	310	312	318	321	328	333
a_4	3,8	4,0	4,2	4,4	4,6	4,8
b_4	3	4	5	6	7	8

1.3.3. Напишите функцию для формирования синусоиды, получаемой в результате дискретизации с частотой F_s непрерывной синусоиды:

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

где $t = \dots -3\Delta T, -2\Delta T, -\Delta T, 0, \Delta T, 2\Delta T, \dots \Delta T = 1/F_s$; A – амплитуда; f – частота синусоиды; φ_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. 1.7):

Таблица 1.7

Числовые значения параметров для формирования синусоид дискретного времени

Номер варианта	1	2	3	4	5	6
Частота сигнала	900 Гц	1 кГц	1,1 кГц	1,3 кГц	1,4 кГц	1,5 кГц
Начальная фаза	45°	60°	30°	45°	60°	30°
Нормализованная амплитуда	40	45	50	55	60	65
Частота дискретизации	8 кГц	8 кГц	8 кГц	10 кГц	10 кГц	10 кГц
Начальное время	0 с	0 с	0 с	0 с	0 с	0 с
Конечное время	6 мс	7 мс	8 мс	4 мс	5 мс	6 мс

1.3.4. Напишите функцию для формирования затухающей экспоненты $g(n) = a^n u(n)$. Постройте график функции, выбрав параметр a согласно варианту (табл. 1.8).

Таблица 1.8

Значения параметра a

Номер варианта	1	2	3	4	5	6
a	0,7	0,75	0,8	0,85	0,9	0,95

1.3.5. Для формирования комплексной экспоненты может быть использована формула Эйлера:

$$x(n) = z^n = r^n e^{j\theta n} = r^n (\cos(\theta n) + j \sin(\theta n)),$$

где $z = re^{j\theta}$, $j^2 = -1$.

Используйте данное выражение со следующими параметрами (табл. 1.9).

Таблица 1.9

Параметры комплексных экспонент (угол θ задан в радианах)

Номер варианта	1	2	3	4	5	6
r	0,89	0,9	0,94	0,95	0,97	0,99
θ	30°	45°	60°	30°	45°	60°

Постройте график действительной и мнимой части для $n = 0 \dots 100$. Как сказывается изменение θ ? Постройте график, откладывая по оси ординат действительную часть, а по оси абсцисс – мнимую (должна получиться спираль). Поэкспериментируйте с углом θ для получения спиралей различного вида.

1.3.6. При помощи формулы Эйлера из комплексных экспонент можно получать функции косинуса и синуса:

$$\sin(\varphi) = \frac{1}{2j}(e^{j\varphi} - e^{-j\varphi}),$$

$$\cos(\varphi) = \frac{1}{2}(e^{j\varphi} + e^{-j\varphi}).$$

Сформируйте и постройте графики следующих последовательностей используя приведенные формулы (комплексная единица в Matlab задается как `1j` или `1i`):

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

Значения параметров ω_1, ω_2, a, b выберите из табл. 1.10.

Таблица 1.10

Значения параметров ω_1, ω_2, a, b

Номер варианта	1	2	3	4	5	6
ω_1	$\pi/3$	$\pi/6$	$\pi/5$	$\pi/\sqrt{7}$	$\pi/8$	$\pi/13$
ω_2	$\pi/4$	$\pi/9$	$\pi/16$	$\pi/6$	$\pi/\sqrt{11}$	$\pi/3$
a	1,1	1,15	1,4	1,3	1,2	1,05
b	0,89	0,9	0,94	0,95	0,97	0,99