

# 1. ДИСКРЕТНЫЕ СИГНАЛЫ

## 1.1. Цели работы

- Знакомство со средой MATLAB.
- Формирование и построение графика кусочно-линейного дискретного сигнала.
- Расчет и построение графика спектра дискретного сигнала.
- Расчет и построение графика аналогового сигнала, восстановленного по дискретным отсчетам в соответствии с теоремой Котельникова.

## 1.2. Индивидуальное задание

В данной лабораторной работе рассматривается кусочно-линейный сигнал  $u(t)$ , заданный параметрами  $U_k, T_k$ , как показано на рис. 1.1.

При выполнении работы производится дискретизация данного сигнала с заданной частотой  $F_s$ .

Параметры сигнала и используемая частота дискретизации являются индивидуальными для каждой бригады и выдаются преподавателем в виде таблицы следующего вида.

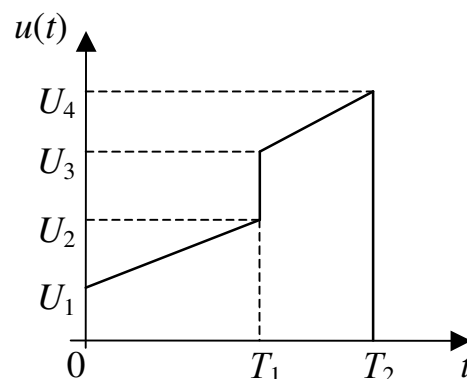



Рис. 1.1

$U_1, \text{В}$	$U_2, \text{В}$	$U_3, \text{В}$	$U_4, \text{В}$	$T_1, \text{мс}$	$T_2, \text{мс}$	$F_s, \text{кГц}$

## 1.3. Указания к выполнению работы

### 1. Запуск MATLAB и знакомство с его средой

Запустите программу MATLAB (ее значок на рабочем столе имеет вид ) . Откроется окно MATLAB.

MATLAB — интерпретируемый язык программирования, позволяющий осуществлять вычисления как в интерактивном режиме, так и путем запуска программ (*сценариев*, *scripts*). Работа в интерактивном режиме сводится к набору в командной строке MATLAB необходимых команд (выполнение команды производится после нажатия клавиши **Enter**), например:

```
>> A = sqrt(cos(pi/12)^2 + 1)
A =
    1.3903
```

В результате выполнения данной команды будет создана переменная **A** со значением, равным  $\sqrt{\cos^2(\pi/12) + 1}$ .

- Создайте переменные для всех параметров сигнала ( $U_1 \dots U_4$ ,  $T_1$ ,  $T_2$ ) и частоты дискретизации  $F_s$ .

## 2. Формирование дискретного сигнала

MATLAB ориентирован на работу с векторами и матрицами. Используйте эти возможности для создания *вектора отсчетов* заданного сигнала:  $x(k) = u(kT)$ , где  $T = 1/F_s$  — интервал дискретизации.

- Сформируйте вектор моментов времени взятия отсчетов.

**Справка.** Для создания векторов в виде арифметических прогрессий используется символ двоеточия. Конструкция  **$\mathbf{x1} : \mathbf{dx} : \mathbf{x2}$**  означает создание равномерной последовательности чисел с первым элементом  **$\mathbf{x1}$** , шагом  **$\mathbf{dx}$**  и конечным значением, равным  **$\mathbf{x2}$**  или меньшим его не более чем на  **$\mathbf{dx}$** . Если шаг последовательности равен единице,  **$\mathbf{dx}$**  можно не указывать:  **$\mathbf{x1} : \mathbf{x2}$** .

- Рассчитайте значения отсчетов первого линейного фрагмента сигнала ( $u(t) = a_1 t + b_1$  при  $t = 0 \dots T_1$ ). Значения параметров  $a_1$  и  $b_1$ , а также диапазоны используемых номеров элементов вектора моментов времени определите самостоятельно.

**Справка.** Для обращения к части элементов вектора нужно указать в круглых скобках *вектор номеров* этих элементов. Чаще всего требуется обращение к одиночному элементу ( **$\mathbf{x}(\mathbf{n})$** ) или сплошному диапазону номеров ( **$\mathbf{x}(\mathbf{n1} : \mathbf{n2})$** ). Для ссылки на последний элемент можно использовать ключевое слово **end**.

**В отличие от многих языков программирования, нумерация элементов в MATLAB всегда начинается с единицы!**

- Аналогичным образом рассчитайте значения отсчетов второго линейного фрагмента сигнала ( $u(t) = a_2 t + b_2$  при  $t = T_1 \dots T_2$ ). Значения параметров  $a_2$  и  $b_2$ , а также диапазон используемых номеров элементов вектора моментов времени определите самостоятельно.

**Замечание.** Если на момент взятия отсчета приходится скачок уровня аналогового сигнала, для целей данной работы значение дискретного отсчета может быть выбрано произвольно в пределах диапазона скачка. Однако с теоретической точки зрения правильнее всего принять значение отсчета равным *середине* скачка (полусумме значений аналогового сигнала справа и слева от скачка).

- Постройте график дискретного сигнала, используя сначала функцию **plot**, а затем функцию **stem**, чтобы сравнить результаты их работы.

**Справка.** Наиболее часто для построения графиков используется функция **plot** в форме **plot(y)** или **plot(x, y)** (в первом случае по горизонтали откладываются номера элементов вектора, во

втором варианте строится зависимость  $y(x)$ ). Параметры **x** и **y** должны быть векторами одинаковой длины.

Для построения графиков дискретных сигналов полезна функция **stem**, которая имеет такой же синтаксис. Она строит графики в виде вертикальных «стебельков».

### 3. Создание MATLAB-программы

Для дальнейшей работы необходимо превратить код, созданный в интерактивном режиме, в MATLAB-программу (сценарий, script). Для этого выполните следующее:

- Создайте (средствами Windows) на сервере дисплейного класса папку для лабораторных работ вашей бригады. Место расположения этих папок обычно имеет следующий вид (*уточнить у преподавателя*):

**W:\stud\2320\номер\_группы\папка\_бригады**

Здесь «2320» — номер дисплейного класса, в котором проводятся занятия.

- В окне MATLAB сделайте созданную папку бригады текущей папкой. Для этого служит поле **Current Folder** в верхней части окна.
- В панели истории введенных команд (**Command History**) выделите команды, с помощью которых задавались параметры дискретного сигнала и производилось его формирование (выделение диапазона строк производится путем щелчка мышью при нажатой клавише **Shift**). Скопируйте выделенные команды в буфер обмена.

**Справка.** Если панель **Command History** отсутствует на экране, включите ее отображение командой меню **Desktop ► Command History**.

- Создайте новый файл MATLAB-программы (самый левый значок (**New script**) панели инструментов, команда меню **File ► New ► Script** или клавиши **Ctrl + N**) и вставьте программный код из буфера обмена.
- Сохраните файл, дав ему осмысленное имя.

**Справка.** Имена файлов в MATLAB фактически являются *идентификаторами переменных*, поэтому на них налагаются соответствующие ограничения. Допустимые символы: латинские буквы, цифры, символ подчеркивания; первый символ — не цифра.

**Русские буквы в именах файлов использовать нельзя!**

- Отредактируйте код, удалив ненужные строки, возможно, оставшиеся после экспериментов, производившихся в интерактивном режиме. Запустите MATLAB-программу клавишей **F5**, кнопкой ► панели инструментов или командой меню **Debug ► Run**. Добейтесь того, чтобы программа работала без ошибок.

**Справка.** На данном этапе уже не нужен вывод на экран результатов выполнения каждой команды — это было полезно при освоении системы в интерактивном режиме. Чтобы отключить вывод результатов расчета на экран, в конце строки кода нужно поставить символ «точка с запятой».

Последующие пункты работы выполняются путем дополнения созданной MATLAB-программы.

#### 4. Построение графика спектра дискретного сигнала

Спектр дискретного сигнала рассчитывается по формуле

$$\dot{X}(\omega) = \sum_{k=0}^{N-1} x(k)e^{-j\omega k}. \quad (1)$$

В этой формуле учтено, что рассматриваемый сигнал содержит конечное число ( $N$ ) ненулевых отсчетов. Необходимо рассчитать спектр в *основном* частотном диапазоне ( $\omega = -\pi \dots +\pi$ ) и построить графики амплитудного и фазового спектров.

На примере расчета спектра дискретного сигнала можно познакомиться с выполнением в MATLAB векторно-матричных операций. Примерная последовательность выполнения расчета:

- Сформируйте *вектор-столбец* номеров отсчетов  $k$ :

$$\mathbf{k} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ N-1 \end{bmatrix}.$$

**Справка.** Рассмотренный ранее способ создания равномерных последовательностей создает *вектор-строку*. Превратить строку в столбец можно с помощью операции транспонирования, которая в MATLAB задается символами «. '». Апостроф без точки («'») означает *эрмитово сопряжение* матрицы (транспонирование в сочетании с комплексным сопряжением). Для вещественных матриц обе операции эквивалентны.

- Сформируйте *вектор-строку* равномерно расположенных частот для расчета спектра. Диапазон частот — от  $-\pi$  до  $+\pi$ , шаг по частоте рекомендуется выбирать так, чтобы вектор содержал от 500 до 1000 элементов.

$$\mathbf{w} = [\omega_1 \quad \omega_2 \quad \dots \quad \omega_M].$$

Здесь  $M$  — число частот в векторе  $\mathbf{w}$ .

**Справка.** Константа  $\pi$  в MATLAB задается как **pi**.

- Вычислите произведение столбца  $\mathbf{k}$  и строки  $\mathbf{w}$  — это даст матрицу размером  $N \times M$ , содержащую все попарные произведения значений  $k$  и  $\omega$ .

- Умножьте полученную матрицу на  $-j$ , чтобы получить показатель комплексной экспоненты из формулы (1).

**Справка.** Мнимую единицу в MATLAB можно задать с помощью констант **1i** или **1j**. После запуска системы значение мнимой единицы также присвоено переменным **i** и **j**, однако эти значения могут быть переопределены в процессе работы.

- Вычислите значения комплексной экспоненты из формулы (1).

**Справка.** Функции **exp**, **sin**, **sqr** и т. п. обрабатывают матричные аргументы *поэлементно*.

- Умножьте сформированный ранее вектор отсчетов сигнала **x** на матрицу значений комплексных экспонент. Эта операция реализует суммирование по  $k$  в формуле (1) и в результате дает вектор-строку значений спектра, рассчитанных для  $M$  частот из вектора **w**.
- Постройте графики амплитудного и фазового спектров сигнала, разместив их друг под другом в одном графическом окне. Зависимости следует строить не от нормированной частоты  $\omega$ , а от линейной частоты ( $f = F_s \omega / (2\pi)$ ).

**Справка.** Для создания нового графического окна служит команда **figure**. Команда **subplot(r, c, n)** делит графическое окно на «клетки» (**r** строк и **c** столбцов) и делает текущей «клетку» **n** (нумерация осуществляется по строкам).

Чтобы избежать увеличения числа графических окон после каждого запуска программы, добавьте в ее начало команду **close all**, закрывающую все графические окна.

Для получения амплитудного и фазового спектров понадобятся следующие функции:

- **abs(x)** — вычисление модуля, в том числе для комплексных чисел.
- **angle(x)** — вычисление аргумента (фазы) комплексных чисел (в радианах).
- **unwrap(x)** — устранение скачков фазы на  $2\pi$  в векторах фазовых значений. Сравните графики фазового спектра, получаемые без использования и с использованием данной функции.

## 5. Восстановление аналогового сигнала по теореме Котельникова

Восстановление аналогового сигнала  $s(t)$  с ограниченным спектром по его дискретным отсчетам  $x(k)$  в соответствии с теоремой Котельникова производится по формуле

$$s(t) = \sum_{k=-\infty}^{\infty} x(k) \frac{\sin\left(\frac{\pi}{T}(t-kT)\right)}{\frac{\pi}{T}(t-kT)}, \quad (2)$$

где  $T = 1/F_s$  — интервал дискретизации.

При расчетах с помощью компьютера значения сигнала  $s(t)$  могут быть вычислены для произвольного *конечного* набора значений времени  $t$ . В данной лабораторной работе сигнал  $s(t)$  рассчитывается для равномерно следующих значений времени, расположенных в 10 раз чаще, чем было при дискретизации сигнала.

Для вычисления отдельных слагаемых формулы (2) удобно использовать матричные возможности MATLAB, рассчитывая эти слагаемые сразу для всех требуемых значений  $t$ , при этом для суммирования слагаемых придется организовать традиционный цикл.

**Справка.** Цикл с заранее заданным числом повторений (цикл **for**) в MATLAB чаще всего записывается следующим образом:

```
for k = 1:N
    % тело цикла
end
```

В данном случае тело цикла будет выполнено **N** раз, для целочисленных значений переменной **k**, меняющихся от единицы до **N**.

- Сформируйте вектор моментов времени для расчета восстановленного сигнала по формуле (2). Шаг по времени должен быть в 10 раз меньше исходного интервала дискретизации. Охватываемый диапазон времени должен выходить за края сигнала как минимум на 5 исходных интервалов дискретизации (это необходимо для того, чтобы увидеть на графике затухающие «хвосты» функций  $\sin(x)/x$ ).
- Сформируйте заполненную нулями «заготовку» для вектора значений восстановленного сигнала  $s(t)$ . Длина этого вектора должна совпадать с длиной созданного на предыдущем шаге вектора моментов времени.

**Справка.** Функция **zeros(m, n)** возвращает заполненную нулями матрицу с **m** строками и **n** столбцами. Функция **length(x)** возвращает длину вектора **x**.

- Реализуйте вычисление суммы по  $k$  в формуле (2), используя для этого цикл по отсчетам сигнала. В теле цикла к созданной ранее «заготовке» для сигнала должно прибавляться очередное слагаемое формулы (2).

**Справка.** Поскольку слагаемые формулы (2) рассчитываются сразу для всех моментов времени  $t$ , числитель и знаменатель функции  $\sin(x)/x$  являются *векторами*, и их деление друг на друга должно выполняться *поэлементно*. По умолчанию MATLAB выполняет операции умножения и деления по *матричным* правилам. Чтобы

осуществить поэлементные действия с матрицами, перед знаком операции необходимо поставить точку: «**.** \*», «**.** /».

- Выведите график восстановленного сигнала с помощью функции **plot**. Разрывы на графике вызваны неопределенностями (0/0) при расчете функции  $\sin(x)/x$  с нулевым аргументом.

**Справка.** Избавиться от этих разрывов можно, используя для расчета функции  $\sin(x)/x$  готовые средства MATLAB. Функция **sinc** возвращает результат, равный  $\sin(\pi x)/(\pi x)$  (именно так определяется функция sinc в зарубежной литературе).

- Постройте графики исходного дискретного и восстановленного аналогового сигналов в общих координатных осях, чтобы наглядно видеть, как они соотносятся друг с другом.

**Справка.** Команды **hold on** и **hold off** включают и выключают режим сохранения текущего содержимого окна при построении нового графика.

Для управления цветом графиков можно добавлять к командам **plot** и **stem** дополнительные строковые параметры, например, **plot(x, y, 'r')** задает вывод графика красным цветом.

## 6. Подготовка материалов для отчета

В завершение работы необходимо скопировать в документ Microsoft Word (в качестве заготовки для отчета) созданный программный код и все полученные графики (дискретный сигнал; его амплитудный и фазовый спектры; совместно показанные дискретный и восстановленный аналоговый сигналы).

**Справка.** Для переноса графиков в документы Word следует использовать команду **Copy Figure** из меню **Edit** окна графика. Перед этим целесообразно задать векторный формат копирования, выбрав в том же меню команду **Copy Options** и установив переключатели **Preserve Information** и **Transparent Background**.

### 1.4. Содержание отчета

Отчет должен содержать созданный в процессе работы программный код, оформленный в виде законченного документа (с заголовками разделов, формулами и комментариями к коду). Полученные в ходе работы графики размещаются в соответствующих разделах отчета.