

## 3. ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ

### 3.1. Цели работы

- Расчет дискретного преобразования Фурье (ДПФ) средствами MATLAB.
- Оценка с помощью прямого и обратного ДПФ полосы частот, в которой сосредоточена основная доля энергии сигнала.
- Наблюдение изменений результатов вычисления ДПФ, происходящих при дополнении сигнала нулевыми отсчетами.
- Оценка степени ускорения вычислений за счет алгоритмов быстрого преобразования Фурье (БПФ).

### 3.2. Индивидуальное задание

В данной лабораторной работе используется сигнал из индивидуального задания для лабораторной работы № 1. Дополнительные индивидуальные параметры отсутствуют.

### 3.3. Указания к выполнению работы

Прямое и обратное ДПФ рассчитываются по следующим формулам:

$$\text{Прямое: } \dot{X}(n) = \sum_{k=0}^{N-1} x(k) e^{-j \frac{2\pi nk}{N}}, \quad \text{обратное: } x(k) = \frac{1}{N} \sum_{n=0}^{N-1} \dot{X}(n) e^{j \frac{2\pi nk}{N}}.$$

#### 1. Подготовка к началу работы

Запустите MATLAB и сделайте текущей папкой папку вашей бригады на сервере дисплейного класса.

Создайте в редакторе MATLAB новый файл MATLAB-программы и скопируйте в него часть кода из программы, созданной при выполнении лабораторной работы № 1 или № 2. Необходимо выбрать фрагмент кода, с помощью которого формируется исходный дискретный сигнал и строится его график.

Сохраните новую программу с осмысленным именем и добейтесь ее работоспособности (программа должна формировать вектор отсчетов дискретного сигнала и строить его график).

Последующие пункты работы выполняются путем дополнения созданной MATLAB-программы.

#### 2. Расчет ДПФ

Вычислите ДПФ для дискретного сигнала из индивидуального задания. Используя функцию **stem**, постройте графики модуля и фазы спектральных отсчетов в одном графическом окне друг под другом (с помощью функции **subplot**).

**Справка.** Для вычисления ДПФ в MATLAB имеется функция **fft** (она вычисляет ДПФ с использованием быстрых алгоритмов — FFT, Fast Fourier Transform). В простейшем случае она вызывается

без дополнительных параметров:  $\mathbf{y} = \mathbf{fft}(\mathbf{x})$ . Здесь  $\mathbf{x}$  — вектор отсчетов сигнала,  $\mathbf{y}$  — вектор результатов вычисления ДПФ.

**Замечание.** Чтобы улучшить внешний вид частотных графиков, целесообразно изменить принятые по умолчанию пропорции графического окна, растянув его в ширину.

### 3. Оценка ширины спектра сигнала

Используя обратное ДПФ, определите минимальное число низкочастотных гармонических составляющих сигнала, содержащих не менее 90% его энергии. Примерная последовательность действий:

1. Рассчитайте энергию исходного сигнала:

$$E_0 = \sum_k |x(k)|^2. \quad (3.1)$$

2. Создайте копию вектора результатов ДПФ под новым именем.
3. Обнулите в созданной копии результатов ДПФ те элементы, которые соответствуют *вещественным* гармоникам с номерами, превышающими некоторое пороговое значение  $N_{\max}$ . Исходное значение  $N_{\max}$  принять равным 0.
4. Вычислите обратное ДПФ и постройте график получившегося сигнала функцией **stem**.

**Справка.** Вычисление обратного ДПФ производится с помощью функции **ifft**:

**$\mathbf{x} = \mathbf{ifft}(\mathbf{y})$  ;**

5. Вычислите энергию получившегося сигнала по формуле (3.1). Если она меньше, чем 90% энергии исходного сигнала  $E_0$ , увеличьте  $N_{\max}$  на единицу и повторите шаги 3–5.

Поиск значения  $N_{\max}$  можно реализовать вручную (задавая новое значение, запуская программу и анализируя результаты) либо написать MATLAB-программу, которая будет это делать автоматически.

В результате выполнения данного пункта работы должно быть получено значение  $N_{\max}$  и построен график низкочастотной части сигнала, содержащей не менее 90% энергии исходного сигнала.

**Замечание.** При определении диапазона номеров обнуляемых элементов ДПФ следует принять во внимание следующее:

- Нумерация элементов массивов в MATLAB начинается с единицы.
- Первое значение в массиве результатов ДПФ соответствует постоянной составляющей сигнала ( $\dot{X}(0)$ ).

- Для получения корректного спектра *вещественного* сигнала в векторе результатов ДПФ должны сохраняться *пары* значений, соответствующие парам одинаковых по модулю положительных и отрицательных частот.
- Вторая половина вектора результатов ДПФ, с учетом периодичности спектров дискретных сигналов, соответствует отрицательным частотам.

В качестве примера ниже приведены разные способы нумерации элементов 8-точечного ДПФ. Двойной рамкой и серой заливкой в таблице выделен диапазон элементов вектора, который необходимо обнулить в случае  $N_{\max} = 2$ .

Номер элемента вектора MATLAB	1	2	3	4	5	6	7	8
Номер $n$ из теоретической формулы ДПФ	0	1	2	3	4	5	6	7
Нумерация с использованием отрицательных частот	0	1	2	3	$\pm 4$	$-3$	$-2$	$-1$

#### 4. Дополнение сигнала нулями

Скопируйте исходный дискретный сигнал в переменную с новым именем и добавьте к концу этой копии сигнала нулевые отсчеты в количестве, равном длине сигнала (длина вектора должна, таким образом, увеличиться в два раза). Вычислите ДПФ для дополненного нулями сигнала, постройте (с помощью функции **stem**) графики модуля и фазы спектральных отсчетов в одном графическом окне друг под другом (с помощью функции **subplot**).

**Замечание.** Чтобы улучшить внешний вид частотных графиков, целесообразно изменить принятые по умолчанию пропорции графического окна, растянув его в ширину.

### 5. Сравнение скорости прямого и быстрого алгоритмов вычисления ДПФ

#### 5.1. Измерение скорости расчетов при вычислении ДПФ непосредственно по теоретической формуле

Функция **fft** гибко выбирает алгоритм вычисления в зависимости от длины преобразуемого сигнала, поэтому измерить с ее помощью скорость вычислений по прямой формуле не удастся. Для реализации такого измерения придется вычислять ДПФ как произвольное линейное преобразование — путем перемножения вектора сигнала и квадратной матрицы преобразования.

**Справка.** Матрица преобразования для ДПФ рассчитывается функцией **dftmtx** ( $N$  — размер вычисляемой матрицы ДПФ):

**D = dftmtx(N) ;**

Для измерения скорости вычислений необходимо реализовать следующий код (предполагается, что вектор сигнала **x** — строка). Для измерения времени выполнения фрагмента кода используется пара функций **tic** и **toc** (первая запускает таймер, вторая выводит результат), вычисления повторя-

ются в цикле много раз, чтобы время выполнения кода было достаточно велико.

**Внимание!** Число повторений цикла (в приведенном примере — 1000) необходимо подобрать экспериментально, чтобы время расчета при  $N = 1024$  составляло *примерно 1 секунду*. Это время ощутимо варьируется от запуска к запуску, так что стремиться к большей точности здесь не следует.

```
N = 1024;           % размер ДПФ
% дополнение сигнала нулями до длины N
x1 = [x zeros(1, N-length(x))];
D = dftmtx(N);      % матрица ДПФ
y = zeros(1, N);    % массив для результатов ДПФ
tic                % старт таймера
for k = 1:1000      % цикл для измерения времени
    y = x1 * D;      % вычисление ДПФ по прямой формуле
end
toc                % отображение измеренного времени
```

Занесите измеренное время выполнения кода при  $N = 1024$ , а также подобранное число повторений цикла, в следующую таблицу.

Число повторений цикла:								
$N$	64	128	256	512	1024	2048	4096	8192
Время, с								
Время однократного вычисления ДПФ, мкс								

Повторите эксперимент для остальных значений  $N$ , указанных в таблице, *не меняя* подобранное число повторений цикла.

Последняя строка таблицы заполняется при подготовке отчета путем деления измеренных значений времени на использованное число повторений цикла.

**Внимание!** Время работы кода растет пропорционально  $N^2$ , поэтому для последнего значения 8192 оно будет составлять *около одной минуты*.

## 5.2. Измерение скорости расчетов при вычислении ДПФ с использованием быстрого алгоритма

Функция **fft** гибко выбирает алгоритм вычисления в зависимости от длины преобразуемого сигнала, и при длине, равной степени двойки, будет использован широко известный алгоритм Кули-Тьюки (Cooley-Tukey), число операций в котором пропорционально  $N \log_2(N)$ .

Для измерения скорости вычислений необходимо реализовать следующий код. Его общая структура аналогична коду, приведенному выше, однако в явном виде дополнять сигнал нулями не нужно — функции **fft** можно передать дополнительный входной параметр, указывающий размерность вычисляемого ДПФ: **y = fft(x, N)**.

**Внимание!** Число повторений цикла (в приведенном примере — 100000) необходимо подобрать экспериментально, чтобы время расчета при  $N = 1024$  составляло *примерно 1 секунду*. Это число должно оказаться *примерно в 100 раз больше*, чем в предыдущем случае.

```
N = 1024;           % размер ДПФ
y = zeros(1, N);    % массив для результатов ДПФ
tic                 % старт таймера
for k = 1:100000     % цикл для измерения времени
    y = fft(x, N);   % вычисление БПФ
end
toc                 % отображение измеренного времени
```

Занесите измеренное время выполнения кода при  $N = 1024$ , а также подобранное число повторений цикла, в следующую таблицу.

Число повторений цикла:								
$N$	64	128	256	512	1024	2048	4096	8192
Время, с								
Время однократного вычисления ДПФ, мкс								

Повторите эксперимент для остальных значений  $N$ , указанных в таблице, *не меняя* подобранное число повторений цикла.

Последняя строка таблицы заполняется при подготовке отчета путем деления измеренных значений времени на использованное число повторений цикла.

## 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:

- исходный сигнал;
- модуль и фаза результатов ДПФ;
- низкочастотная часть сигнала, восстановленная после обнуления части гармоник;
- модуль и фаза результатов ДПФ при дополнении сигнала нулями.

**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**.

### 3.4. 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 и заполненные таблицы размещаются в соответствующих разделах отчета.

Кроме того, в соответствующем разделе должны быть построены графики затрат времени на *однократное* вычисление ДПФ прямым и быстрым методами. Для получения этих данных следует разделить занесенные в таблицы значения времени на использованное количество повторений цикла. Вдоль горизонтальной оси следует откладывать  $\log_2(N)$ .

Совместно с этими экспериментальными графиками следует построить аппроксимирующие зависимости:

- Для прямого расчета ДПФ:  $t = k_1 N^2$ .
- Для быстрого расчета ДПФ:  $t = k_2 N \log_2(N)$ .

Коэффициенты  $k_1$  и  $k_2$  необходимо подобрать самостоятельно; их значения должны быть приведены в отчете.

В конце отчета должны быть приведены выводы по результатам работы.