

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Севастопольский государственный университет

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторным работам
по дисциплине
"Технологии обработки мультимедиа данных"
для студентов дневной и заочной форм обучения
специальности 09.04.02 — "Информационные
системы и технологии"

Часть 1

Севастополь
2015

УДК 004.383.3 (075)

Методические указания к лабораторным работам по дисциплине "Технологии обработки мультимедиа данных" для студ. дневной и заочной форм обучения специальности 09.04.02 — "Информационные управляющие системы и технологии" / СевГУ ; сост. **В. Н. Бондарев**. — Севастополь : Изд-во СевГУ, 2015 . — часть 1. — 56 с.

Методические указания предназначены для проведения лабораторных работ и обеспечения самостоятельной подготовки студентов по дисциплине "Технологии обработки мультимедиа данных" специальности 09.04.02 — "Информационные системы и технологии". **Целью методических указаний** является обучение студентов основным алгоритмам обработки одномерных и двумерных сигналов, используемым при построении мультимедийных систем различного назначения

Методические указания составлены в соответствии с требованиями программы дисциплины "Технологии обработки мультимедиа данных" для студентов дневной и заочной форм обучения специальности 09.04.02 — "Информационные системы и технологии" и утверждены на заседании кафедры Информационных систем (протокол № от 2015 г.)

Допущено учебно-методическим центром СевГУ в качестве методических указаний.

Рецензент:

Кожаев Е.А., канд. техн. наук, доцент информационных технологий и компьютерных систем.

СОДЕРЖАНИЕ

Введение	4
1. Лабораторная работа №1. Изучение программного пакета matlab (scilab).....	5
2. Лабораторная работа №2. Синтез периодических сигналов на основе рядов Фурье	14
3. Лабораторная работа №3. Гармонический анализ сигналов на основе дискретного преобразования Фурье.....	15
4. Лабораторная работа №4. Линейные дискретные системы и цифровые фильтры.....	26
5. Лабораторная работа №5. Системы с многочастотной дискретизацией.....	31
6. Лабораторная работа №6. Кратковременный спектральный анализ и гомоморфная обработка речевых сигналов.....	32
7. Лабораторная работа №7. Линейный предиктивный анализ речи	39
8. Лабораторная работа №8. Цифровая фильтрация изображений	48
Библиографический список.....	56

ВВЕДЕНИЕ

Цели и задачи лабораторных работ

Основная цель выполнения лабораторных работ — получение практических навыков в решении прикладных задач цифровой обработки сигналов. В результате выполнения лабораторных работ студенты должны углубить знания следующих разделов курса: разложение периодических сигналов в ряды Фурье, дискретное преобразование Фурье, цифровая фильтрация данных, системы с изменением частоты дискретизации, цифровая обработка речи, цифровая фильтрация изображений, а также приобрести практические навыки работы с пакетом MATLAB (Scilab).

Порядок выполнения лабораторных работ

При изучении дисциплины "Технологии обработки мультимедиа данных" выполняется семь лабораторных работ (часть 1).

В лабораторных работах студент решает заданную индивидуальным вариантом задачу. Варианты заданий приведены к каждой лабораторной работе. Номер варианта выбирается в соответствии с номером студента в списке группы. Студенты заочники выбирают номер варианта как остаток от деления числа, соответствующего двум последним цифрам в номере зачетной книжки, на 30. Например, две последние цифры зачетки 46. Тогда остаток от деления 46 на 30 будет равен 16. Следовательно, студент выбирает вариант 16.

Лабораторная работа выполняется в два этапа. На первом этапе — этапе самостоятельной подготовки — студент должен выполнить следующее:

- проработать по конспекту и рекомендованной литературе, приведенной в конце настоящих методических указаний, основные теоретические положения лабораторной работы и ответить на контрольные вопросы;
- разработать алгоритм решения задачи;
- написать программу на языке пакета Matlab и описать ее;
- оформить результаты первого этапа в виде заготовки отчета по выполнению лабораторной работы.

На втором этапе, выполняемом в лабораториях кафедры студент должен:

- отладить программу;
- получить необходимые результаты и графики;
- сформулировать выводы по работе.

По завершении выполнения лабораторной работы студентом оформляется и защищается отчет по лабораторной работе.

1. ЛАБОРАТОРНАЯ РАБОТА № 1

«ИЗУЧЕНИЕ ПРОГРАММНОГО ПАКЕТА MATLAB (Scilab)»

1.1. Цель работы

Изучение технологии подготовки и выполнения программ в среде MATLAB (Scilab), исследование элементарных функций генерации, обработки сигналов и отображения сигналов.

1.2. Краткие теоретические сведения

1.2.1. Назначение пакетов MATLAB и Scilab

Пакет Matlab предназначен для выполнения научных и инженерных расчетов. Эти расчеты могут относиться к области вычислительной математики (численное интегрирование, дифференцирование, решение оптимизационных задач, выполнение различного рода аппроксимаций исходных данных), линейной алгебре и аналитической геометрии, математической статистике, а также к обработке сигналов: спектральный и корреляционный анализ, расчет фильтров и прочее [3]. Пакет Scilab выполняет аналогичные функции, но является свободно распространяемым программным обеспечением и также может использоваться для выполнения лабораторных работ (<http://ru.opensuse.org/Scilab>). Набор базовых функций и функций обработки сигналов, синтаксические конструкции пакета Scilab в значительной степени соответствует функциям и синтаксису пакета Matlab. Поэтому в дальнейшем рассмотрим только краткое описание пакета Matlab. Описание особенностей пакета Scilab приведено в [5].

1.2.2. Символы и знаки операций пакета MATLAB

[] Квадратные скобки. Используются для формирования векторов и матриц. Элементы отделяются пробелами, а строки отделяются точкой с запятой: [6.9 6.96 sqrt(-1)] — вектор; [11 12 13; 21 22 23] — матрица 2x3. Запись A = [] означает пустую матрицу.

. Десятичная точка. Например, $314/100=3.14=.314E1$. Операции типа "элемент на элемент" получаются с использованием символов .*, .^, ./, .\.. Например, выражение C=A./B дает матрицу с элементами $c(i,j)=a(i,j)/b(i,j)$.

, Запятая. Если после арифметического выражения стоит запятая, то его результат выводится на экран дисплея.

; Точка с запятой. Если после арифметического выражения стоит точка с запятой, то результат вычислений не выводится на экран дисплея.

% Процент. Все, что следует за этим символом, воспринимается как комментарий.

! Восклицательный знак. Любой текст, следующий за восклицательным знаком, воспринимается как команда операционной системы.

: Двоеточие. Используется в операторе FOR и еще в ряде случаев. Запись J:K тождественна записи [J,J+1,...,K], если $J > K$, то вектор пуст. Запись J:I:K тождественна записи [J,J+I,J+2I,...,K]. Причем, вектор пуст, если $I > 0$ и $J > K$ или $I < 0$ и $J < K$. Двоеточие можно использовать для выбора нужных строк, столбцов и элементов векторов и матриц. $A(:)$ все элементы A, взятые как один столбец (столбец к столбцу). Запись $A(:,J)$ — J-й столбец A; $A(J:K)$ это $A(J), A(J+1), \dots, A(K)$ и наконец $A(:,J:K)$ это $A(:,J), A(:,J+1)$. Запись $A(J,:)$ — J-я строка A; $A(J:K,:)$ — это $A(J,:), A(J+1,:) \dots, A(K,:)$.

' Верхняя кавычка. Используется для обозначения транспонирования матрицы: X' — транспонированная комплексно-сопряженная матрица X. Кроме того, верхняя кавычка может быть использована для текстовых сообщений: все, что заключено в кавычки воспринимается как текстовое сообщение. Например, 'ANY TEXT'.

+ Сложение. $X+Y$, при этом X и Y должны иметь одинаковую размерность. Скаляр можно складывать с чем угодно. Поэлементное сложение матриц (векторов): $X.+Y$

- Вычитание. $X-Y$, X и Y должны иметь одинаковый размер, если один из них не скаляр. Скаляр можно вычитать из чего угодно. Поэлементное вычитание матриц (векторов): $X.-Y$

*** Умножение.** Матричное умножение: $X*Y$. Любой скаляр (в MATLAB он рассматривается как матрица $1*1$) можно умножать на что угодно. В остальных случаях число столбцов X должно быть равно числу строк Y. Поэлементное умножение матриц (векторов): $X.*Y$

/ Символ деления или правое деление матриц. B/A , грубо говоря, то же самое, что $B*INV(A)$. Более точно $B/A=(A' \setminus B)'$. Если $\dim A = \dim B$, то $A./B$ означает поэлементное деление матриц.

< Символ меньше. Запись $<=$ означает меньше или равно.

> Символ больше. Запись $>=$ означает больше или равно.

= Символ присвоения в арифметических выражениях. Используется также в операторе логического равенства $==$.

1.2.3. Описание встроенных функций и операторов

ABS. $ABS(X)$ вычисляет абсолютную величину элементов X. Когда X комплексная, то $ABS(X)$ вычисляет модуль элементов X.

ACOS, ASIN. Арксинус или арккосинус элементов матрицы X.. Результат выражен в радианах.

ATAN. $ATAN(X)$ вычисляет арктангенс элементов X. Результат выражен в радианах и принадлежит отрезку $[-\pi/2, \pi/2]$.

AXIS. Масштабирование осей. $AXIS(h)$, где вектор $h=[xmin, xmax, ymin, ymax]$ — вектор, элементы которого указывают разметку осей x и y. $AXIS('square')$ — оси выбираются таким образом, чтобы рисунок был приблизительно квадратным. С помощью функции $AXIS('normal')$ можно вернуться к прежним осям.

CEIL. $CEIL(X)$ округляет элементы X к ближайшему целому числу в сторону плюс бесконечности.

COS. Функция COS(X) вычисляет косинус X, причем аргумент X должен быть задан в радианах.

DET. Функция DET(A) вычисляет определитель квадратной матрицы A.

ELSE. Используется совместно с IF. Операторы, следующие за ELSE выполняются, если не выполнено условие, указанное в IF.

ELSEIF. Используется совместно с IF. Операторы, следующие за ELSEIF выполняются, если не выполнено условие, указанное в IF.

END. Заканчивает операторы FOR, WHILE, IF. Каждый END связан с ближайшим оператором FOR, WHILE или IF, не имеющим END.

EVAL. Текстовые макро возможности. EVAL(X) заставляет вычислять значение текста, содержащееся в X. Например: t = '1/(i+j-1);for i=1:3, for j=1:3, a(i,j)=eval(t); end, end,' приводит к генерированию матрицы Гильберта третьего порядка.

EXP. Функция EXP(X) вычисляет экспоненты элементов X.

EYE. Единичная матрица. EYE(N) — матрица NxN с единицами на диагонали, EYE(M,N) — матрица MxN с единицами на диагонали и нулями на остальных позициях. EYE(A) — единичная матрица таких же размеров как A.

FFT. FFT(X) вычисляет дискретное преобразование Фурье вектора X с помощью быстрого алгоритма. Если длина вектора X не равна точной степени двойки, то он дополняется справа нулями до степени двойки. Функция IFFT выполняет обратное преобразование Фурье. Функции FFT2, IFFT2 выполняют соответственно прямое и обратное двумерное преобразование Фурье. При этом строки или столбцы X, которые не являются целой степенью двойки, дополняются нулями.

FILTER. Y = FILTER(B,A,X) — фильтрация данных из вектора X с помощью рекурсивного фильтра

$$y[n] = \sum_{k=0}^{N-1} b[k]x[n-k] - \sum_{k=1}^{M-1} a[k]y[n-k]$$

коэффициенты которого $a[k]$, $b[k]$ хранятся в векторах A, B.

FIND. Поиск индексов ненулевых элементов вектора. I = FIND(X) дает индексы вектора X, которые соответствуют ненулевым компонентам X. Например, I = FIND(X > 100) дает индексы тех компонент X, которые больше 100.

FIX. FIX(X) округляет элементы X до ближайшего в направлении нуля числа.

FLOOR. FLOOR(X) округляет элементы X до ближайшего целого числа в сторону минус бесконечности.

FOR. Повтор команд определенное количество раз. Используется так: FOR переменная = выражение, команда,...,команда END.

FORMAT. Устанавливает выходной формат. Все вычисления выполняются с двойной точностью. Для изображения чисел на дисплее можно использовать следующие форматы:

FORMAT (стандартный : фиксированная точка, 5 разрядов) ;

FORMAT SHORT (фиксированная точка , 5 разрядов) ;

FORMAT LONG (фиксированная точка , 15 разрядов) ;

FORMAT SHORT E (плавающая точка , 5 разрядов) ;

FORMAT LONG E (плавающая точка , 15 разрядов).

FUNCTION . В MATLAB предусмотрено создание новых функций. Новая функция должна использовать уже существующие функции. Новая функция должна быть помещена в новый файл с расширением M, имя которого совпадает с именем новой функции. В первой строке файла должна быть строка, содержащая синтаксическое описание новой функции.

GLOBAL. Определение глобальных переменных. GLOBAL X Y Z определяет переменные X,Y,Z как глобальные.

GRID. Нанесение сетки на текущий график.

HOLD. Сохранение текущего графика на экране. Последующие команды PLOT будут добавлять графики (накладывать на ранее созданный), используя уже существующие оси. Оператор HOLD OFF — прекращает наложение графиков, а оператор HOLD ON возобновляет его. Оператор HOLD сам по себе прекращает режим накопления графиков.

IF. Операции, выполняемые по условию. В простейшей форме формат IF следующий: IF переменная, операция, END. Операции выполняются, если выполняется условие. Например:

```
if i == j,
    A(i,j)=2;
elseif ABS(i-j)=-1;
else
    A(i,j)=0;
end
```

IMAG. Функция IMAG(X) вычисляет мнимую часть X. Для задания мнимых чисел нужно использовать SQRT(-1). Например: число $3+2i$ можно задать как $3+2\text{sqrt}(-1)$.

INPUT. Ввод исходных данных с пульта. INPUT ('How many apples') выводит на экран сообщение, заключенное в верхние скобки, и ожидает пока пользователь не наберет с пульта нужное число. Это число (после нажатия клавиши "возврат каретки") передается в программу. INPUT('WHAT IS YOUR NAME','S') выводит текст в первых верхних кавычках на экран и ждет ответа пользователя. Текст, набранный пользователем, передается в программу.

INV. Инвертирование матрицы. INV(A) обращает квадратную матрицу A. Предостерегающее сообщение появляется, если A плохо масштабирована (большой разброс элементов) или близка к сингулярной.

LOAD. LOAD file загружает в оперативную память все переменные из файла, названного 'file.mat '. Функция LOAD без аргументов загружает переменные из файла MATLAB.MAT. С помощью LOAD можно также читать ASCII файлы: в этом случае результат передается в те же переменные, что и в файле.(CM. SAVE).

LOG. Функция LOG(X) вычисляет натуральный логарифм элементов X. Если X меньше нуля, то результат комплексный. Функция LOG10(X) вычисляет логарифм по основанию десять от X.

LOGLOG. Эта функция вычерчивает график с логарифмическим масштабом по осям X и Y.

MAX. Для вектора MAX(X) выбирает максимальный элемент из X. Для матрицы MAX(X) это вектор, содержащий максимальные элементы каждого из столбцов.

$[Y,I] = \text{MAX}(X)$ содержит в Y максимальные элементы столбцов A , а в векторе I содержит индексы этих элементов. $\text{MAX}(X,Y)$ формирует матрицу Z , элементы которой равны $z_{ij} = \max(x_{ij}, y_{ij})$. Если матрица комплексная, то используются величины $\text{MAX}(\text{ABS}(X))$.

MESH. Построение трехмерных изображений. $\text{MESH}(Z)$ изображает трехмерную поверхность, причем матрица Z содержит "высоты" точек поверхности (элемент Z можно рассматривать как значение функции двух переменных X и Y) $\text{MESH}(Z,M)$ изображает поверхность, видимую из определенной точки пространства. Координаты этой точки содержатся в векторе $M = [AZ \text{ } EL]$, где AZ угол в плоскости XOY , а EL — угол в азимутальной плоскости, изменяющийся от 90 градусов до минус 90 градусов (отрицательные значения угла соответствуют как бы взгляду на поверхность снизу).

Функции $\text{MESH}(Z,S)$ и $\text{MESH}(Z,M,S)$, где $S = [sx \ sy \ sz]$ вектор, параметры которого задают размеры поверхности по каждой из осей X,Y,Z .

MESHDOM. Генерирование матриц X и Y для трехмерных изображений. $[XX \ , \ YY \] = \text{MESHDOM}(X,Y)$ преобразует отрезки, определенные векторами X,Y в матрицы XX, YY которые можно использовать для построения трехмерных изображений.

MIN. Выбор минимального элемента вектора или минимальных элементов в столбцах матрицы.

NARGIN. В теле пользовательской функции, постоянная переменная $NARGIN$ указывает число входных аргументов, которые были использованы при вызове функции, а $NARGOUT$ указывает число выходных аргументов.

ONES. Все единицы. $\text{ONES}(N)$ — матрица $N \times N$ из единиц. $\text{ONES}(M,N)$ — матрица $M \times N$ из единиц. $\text{ONES}(A)$ тех же размеров, что и A , но со всеми единицами в качестве элементов.

PAUSE. Заставляет процедуру остановиться и ждать, пока пользователь не нажмет любую клавишу. $\text{PAUSE}(n)$ останавливает выполнение программы на n секунд.

PI. $\text{PI} = 4 * \text{atan}(1) = 3.1415926535897...$

PLOT. Изображение векторов или матриц. $\text{PLOT}(X,Y)$ рисует вектор X против вектора Y (X — аргумент, Y — функция). Если X и Y матрицы, то также вычерчиваются графики. $\text{PLOT}(X1,Y1,X2,Y2)$ другой способ построения нескольких кривых на одном графике, $\text{PLOT}(X1,Y1,':',X2,Y2, '+')$ наносит первую кривую точками, а вторую символами $+$. Можно использовать символы : (сплошная -), (пунктирная --), (точки :), (точки и пунктиры -.), (запятые,), (плюсы +), (звездочки *), (кружечки o), (крестики x). При наличии цветного дисплея можно задавать цвета кривых: r — красный, g — зеленый, b — голубой, w — белый, i — невидимый. $\text{PLOT}(Y)$ вычерчивает Y , как функцию индекса, и она же эквивалентна $\text{PLOT}(\text{real}(Y), \text{imag}(Y))$, если Y комплексный. Во всех остальных случаях использования PLOT мнимая часть игнорируется.

POLAR. Построение графика в полярных координатах. Функция $\text{POLAR}(T,Y)$ строит график функции $y = y(t)$ в полярных координатах. Аргумент t должен быть задан в радианах, а значение функции — это полярный радиус соответствующего аргумента.

PROD. Для вектора PROD(X) произведение элементов X, для матрицы вектор - строка с произведениями элементов столбцов.

RAND. Случайные числа и матрицы. RAND(N) — матрица NxN со случайными элементами. RAND(M,N) матрица MxN со случайными элементами. RAND(A) — матрица тех же размеров, что и A, но со случайными элементами. RAND — случайное число, значение которого изменяется каждый раз при вызове. Обычно, случайные числа распределены равномерно на отрезке [0,1], но можно с помощью RAND('normal') переключиться на генерирование случайных чисел, распределенных нормально с нулевым матожиданием и единичной дисперсией. Имеется команда RAND('seed',n), которая при n = 0 устанавливает датчик случайных чисел в исходное состояние, при других значениях n происходит установка счетчика в некоторое состояние. После чего командой RAND можно генерировать последовательность случайных чисел: каждому n соответствует своя, строго определенная последовательность.

REAL. REAL(X) — вещественная часть числа.

REM. Остаток. REM(X,Y) — это $(x - y * n)$, где $n = \text{fix}(x/y)$ — целое число, ближайшее к точной величине x/y .

ROUND. Округление элементов X до ближайшего целого: ROUND(X).

SAVE. SAVE file — сохраняет текущие переменные в файле с названием 'file.mat'. SAVE file X — сохраняет только X. SAVE file X,Y,Z сохраняет X, Y, Z. SAVE само по себе сохраняет все в файле MATLAB.MAT.

SEMILOGX. Изображение графика с логарифмическим (по основанию 10) масштабом по оси X и линейным по оси Y.

SEMILOGY. Изображение графика с логарифмическим (по основанию 10) масштабом по оси Y и линейным по оси X.

SIGN. Знаковая функция.

SIN. Функция SIN(X) вычисляет синус X, причем, X должен быть в радианах.

SIZE. Если X матрица размерности MxN, то SIZE(X) это вектор с элементами [M,N]. Можно использовать SIZE и в такой форме записи: [M,N] = SIZE(X).

SORT. SORT(X) сортирует каждый столбец X в возрастающем порядке элементов. [Y,I] = SORT(X) также дает матрицу I, содержащую индексы элементов в исходной матрице (по ним можно определить, где в X находился соответствующий элемент Y). Если X вектор, то $Y = X(I)$. Когда X комплексная матрица, то элементы сортируются по абсолютной величине.

SUBPLOT. Управление графическими окнами. Функция SUBPLOT(MNP), где M,N,P — целочисленные параметры, помещает далее следующий график в MxN окон ($M \leq 2, N \leq 2$) и выбирает P -ое окно для текущего графика.

SUM. Для вектора X функция SUM(X) вычисляет сумму всех элементов X. Для матрицы X функция SUM(X) вычисляет вектор-строку с суммами элементов каждого столбца матрицы X. Выражение SUM(DIAG(X)) вычисляет след матрицы X.

TEXT. TEXT (X,Y,'string') отображение текста в области графика в точке с координатами (X,Y) Если X,Y — векторы, то текст пишется во всех точках графического экрана, соответствующих координатам X и Y. TEXT (X,Y,'string','sc') вос-

принимает точки (X,Y) уже в координатах экрана; где (0.0,0.0) — нижний левый угол экрана, а(1.0,1.0) — верхний правый.

TITLE. TITLE ('text') пишет текст в кавычках над графиком как его заголовок. TITLE(text) в качестве заголовка графика помещает значение строковой переменной text.

WHILE. Повтор выражений неопределенное число раз. Используется так: WHILE переменная, выражение, ..., выражение, END. Выражения вычисляются до тех пор, пока все элементы переменной отличны от нуля.

XLABEL. Подпись под осью X. XLABEL('text') пишет текст в кавычках под осью X на текущем графике.

YLABEL. Подпись возле оси Y. YLABEL('text') пишет текст в кавычках слева от оси Y, располагая его вертикально.

ZEROS. Все нули. ZEROS(N) — матрица NxN со всеми нулевыми элементами. ZEROS(M,N) — матрица MxN со всеми нулевыми элементами. ZEROS(A) — матрица тех же размеров, что и A, но со всеми нулевыми элементами.

1.3. Варианты заданий

Варианты заданий приведены в таблице 1.1

Таблица 1.1 — Варианты заданий

№ варианта	M	N	Frec	Fsample
1	10	5	15	120
2	6	12	12	96
3	5	17	7	60
4	10	11	30	240
5	15	4	37	300
6	7	11	22	160
7	9	5	36	300
8	10	8	33	260
9	5	7	10	80
10	8	12	12	92
11	7	14	17	132
12	16	4	20	160
13	8	11	12	100
14	5	7	9	68
15	12	8	30	240
16	15	5	36	300
17	5	12	23	160
18	15	4	15	120
19	5	7	8	60
20	12	5	30	240
21	13	3	35	300
22	15	8	25	160

Продолжение таблицы 1.1

№ варианта	M	N	Frec	Fsample
23	5	13	30	300
24	12	8	35	240
25	11	4	17	128
26	8	13	12	96
27	5	7	8	60
28	16	4	25	200
29	19	3	38	300
30	15	8	23	160

1.4. Порядок выполнения лабораторной работы

1.4.1. Запустите MATLAB. Команда `>>Intro` продемонстрирует отдельные возможности пакета. Ознакомьтесь с возможностью ввода данных в командной строке MATLAB, простейшими операциями над векторами и матрицами, возможностью отображения данных на графиках различного вида.

1.4.2. Создайте новый программный файл (File->New->M-File запустится редактор-отладчик MATLAB Editor/Debugger).

1.4.3. Напишите на языке пакета MATLAB программу, реализующую следующие операции:

- чтение из внешнего файла входной двухмерной матрицы (для этого необходимо предварительно подготовить файл с расширением '.asc'(например 'x1.asc'), содержащей произвольную матрицу (размерность матрицы $M \times N$ определяется вариантом задания)). Для чтения необходимо воспользоваться командой Load ("`>>load x1.asc`" и переменная **x1** инициализируется значениями из файла x1.asc);
- генерацию случайной матрицы **y1** размерностью $M \times N$ (воспользоваться функцией Rand);
- вычисление произведения матриц **x1** и **y1** (для этого необходимо сначала транспонировать матрицу **y1**);
- построение графика поверхности заданной результирующей матрицей (воспользоваться функцией surf) с подписями осей (X,Y,Z), заголовком графика, координатной сеткой (воспользоваться функциями Xlabel,Ylabel,Zlabel,Title, Grid).

1.4.4. Напишите на языке пакета MATLAB программу, генерирующую синусоидальные колебания с частотой **Frec**, продискретизированные с шагом **1/Fsample**, **1/(2*Fsample)**, **1/(0.5*Fsample)** (значения **Frec** и **Fsample** определяются вариантом задания), сохраняющую полученные результаты, соответственно, в векторах **x1**, **x2** и **x3** (время реализации сигнала равно двум периодам колебания) и отображающую в одном графическом окне график **x1(t)**, а другом — **x2(t)** и **x3(t)**, а также сохраняющую значения **x1**, **x2** и **x3** во внешнем файле.

Обратите внимание на то, что MATLAB не является символьным пакетом, и поэтому не в состоянии генерировать аналоговые сигналы (например, ввод строки

>>x=sin(t) приведет к возникновению ошибки). Вместо этого аналоговые сигналы могут быть представлены отдельными отсчетами:

```
>>fSample=1000;
>>frec=400;
>>t=[0:1/fSample:0.5]; %т.е. t=[0 0.001 0.002 0.003 ... 0.05]
>>x=sin(2*pi*frec*t);
>>plot(t,x)
```

1.5. Содержание отчета

Отчет должен содержать следующие разделы:

- постановка задачи (вариант задания);
- тексты программ с комментариями;
- графики реализаций сгенерированных сигналов;
- выводы

1.6. Контрольные вопросы

- 1.6.1. Как в матлабе задаются векторы и матрицы?
- 1.6.2. Как в матлабе определить размерность вектора или матрицы?
- 1.6.3. Как в матлабе обеспечить выборку заданной строки матрицы? заданного столбца матрицы?
- 1.6.4. Для чего предназначены функции матлаба: zeros, ones, eye, diag?
- 1.6.5. Для чего предназначены функции матлаба: floor, ceil, round, rem, fix?
- 1.6.6. Для чего предназначены функции матлаба: load и save?
- 1.6.7. Как в матлабе ввести значение с клавиатуры и отобразить его на дисплее?
- 1.6.8. Запишите и объясните условный оператор языка матлаб.
- 1.6.9. Запишите и объясните оператор цикла for.
- 1.6.10. Запишите и объясните оператор цикла while.
- 1.6.11. Какие средства имеются в матлабе для построения графиков одномерных функций?
- 1.6.12. Какие средства имеются в матлабе для построения графиков двумерных функций?
- 1.6.13. Какими средствами матлаба можно выполнять надписи на графиках?
- 1.6.14. Как сгенерировать случайный вектор или матрицу?
- 1.6.15. Каким образом осуществляется ввод данных в пакете MATLAB?
- 1.6.16. Как в MATLAB создать новое графическое окно?
- 1.6.17. Как в одном графическом окне построить несколько графиков функций?
- 1.6.18. Напишите на языке MATLAB программу, вычисляющую сумму квадратов элементов двухмерной матрицы?
- 1.6.19. Как получить список переменных, содержащихся в оперативной памяти MATLAB?
- 1.6.20. Что такое дискретизация по времени и дискретизация по амплитуде?

1.6.21. Чему равен минимальный шаг дискретизации во времени гармонического колебания с частотой 400Гц для его последующего восстановления?

2. ЛАБОРАТОРНАЯ РАБОТА №2

«СИНТЕЗ ПЕРИОДИЧЕСКИХ СИГНАЛОВ НА ОСНОВЕ РЯДОВ ФУРЬЕ»

2.1. Цель работы

Исследование возможности аппроксимации периодических сигналов рядом Фурье по ортогональной системе гармонических функций, синтез сигналов различной формы и исследование влияния числа ортогональных составляющих на погрешности аппроксимации.

2.2. Краткие теоретические сведения

Математический аппарат, с помощью которого решают задачу синтеза сигналов, основывается на использовании обобщенного ряда Фурье [1]:

$$x(t) = \sum_k c_k \varphi_k(t) \quad , \quad (2.1)$$

где $x(t)$ — аппроксимируемый сигнал; $\{\varphi_k(t)\}$ — ортогональная система базисных функций; c_k — коэффициенты разложения. Формула (2.1) позволяет по заданным коэффициентам и базисным функциям синтезировать сигнал $x(t)$ произвольной формы.

Система базисных функций $\{\varphi_k(t)\}$ называется ортогональной на отрезке $[a, b]$, если скалярное произведение любых 2-х функций (φ_k, φ_m) удовлетворяет условию:

$$(\varphi_k, \varphi_m) = \int_a^b \varphi_k(t) \varphi_m(t) dt = \begin{cases} 0, & k \neq m \\ \|\varphi_k\|^2, & k = m \end{cases} \quad ,$$

где квадрат нормы функции $\varphi_k(t)$: $\|\varphi_k\|^2 = \int_a^b \varphi_k^2(t) dt$.

Коэффициенты c_k обобщенного ряда Фурье могут быть найдены из выражения:

$$c_k = \frac{\int_a^b x(t) \varphi_k(t) dt}{\int_a^b \varphi_k^2(t) dt} = \frac{(x, \varphi_k)}{\|\varphi_k\|^2} \quad .$$

На практике приходится ограничивать ряд Фурье конечным числом членов N . Это приводит к ошибке аппроксимации:

$$\varepsilon(t) = x(t) - \sum_{k=0}^N c_k \varphi_k(t) \quad .$$

Обычно рассматривают норму ошибки:

$$\|\varepsilon\| = \left\{ \int_a^b \varepsilon^2(t) dt \right\}^{1/2} \quad .$$

Аппроксимация сигнала $x(t)$ рядом Фурье при фиксированном числе слагаемых обеспечивает наилучшее приближение в смысле минимума нормы ошибки.

Если ортогональная система является полной, то увеличением количества членов в сумме (2.1) можно сделать норму ошибки сколь угодно малой. Относительную среднеквадратическую погрешность аппроксимации можно определить из выражения:

$$\delta = \frac{\|\varepsilon\|}{\|x(t)\|}, \quad \text{где} \quad \|x(t)\| = \left\{ \int_a^b x^2(t) dt \right\}^{1/2}.$$

Существует большое число полных и ортогональных систем базовых функций. Так, в качестве систем базисных функций можно использовать тригонометрические функции кратных аргументов, полиномы Чебышева, Лежандра, Эрмита, функции Бесселя, Лагерра, Хаара, Уолша и др [1].

Если в качестве базисных функций использовать тригонометрические функции кратных аргументов, то периодическую функцию $x(t)$ с периодом T можно представить в виде ряда Фурье [1, 4]:

$$x(t) = a_0 / 2 + \sum_{k=1}^{\infty} (a_k \cos k\omega_1 t + b_k \sin k\omega_1 t). \quad (2.2)$$

Коэффициенты этого ряда определяются по формулам:

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} x(t) dt, \quad a_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cos k\omega_1 t dt, \quad b_k = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin k\omega_1 t dt$$

Амплитуда и фаза k -ой гармоники разложения

$$A_k = \sqrt{a_k^2 + b_k^2}, \quad \varphi_k = \arctg(b_k / a_k).$$

Совокупности коэффициентов A_k и φ_k называют амплитудным и фазовым частотным спектром периодического сигнала $x(t)$.

Ряд (2.1) приводится к форме (2. 2), если положить

$$c_k = A_k \cdot e^{j\varphi_k}.$$

и в качестве базисных использовать комплексные экспоненциальные функции.

2.3. Варианты заданий

Варианты заданий представлены в таблице 2.1

Таблица 2.1 — Варианты заданий

№ варианта	T_n	T	Вид сигнала
1	1	4	рис. 2.1
2	1	4	рис. 2.2
3	1	4	рис. 2.3
4	1	4	рис. 2.4
5	1	2	рис. 2.1

Продолжение таблицы 2.1

№ варианта	$T_{\text{и}}$	T	Вид сигнала
6	1	2	рис. 2.2
7	1	2	рис. 2.3
8	1	2	рис. 2.4
9	1	8	рис. 2.1
10	1	8	рис. 2.2
11	1	8	рис. 2.3
12	1	8	рис. 2.4
13	1	1	рис. 2.1
14	1	1	рис. 2.2
15	1	3	рис. 2.3
16	1	3	рис. 2.4
17	1	3	рис. 2.1
18	1	3	рис. 2.2
19	1	5	рис. 2.3
20	1	5	рис. 2.4
21	1	5	рис. 2.1
22	1	5	рис. 2.2
23	1	6	рис. 2.3
24	1	6	рис. 2.4
25	1	6	рис. 2.1
26	1	6	рис. 2.2
27	1	6	рис. 2.3
28	1	7	рис. 2.4
29	1	7	рис. 2.1
30	1	7	рис. 2.2

2.4. Порядок выполнения лабораторной работы

2.4.1. Рассчитать по формулам (2.2 значения первых десяти коэффициентов разложения в ряд Фурье по системе тригонометрических функций для следующих сигналов, изображенных на рис.2.1 —2.4.

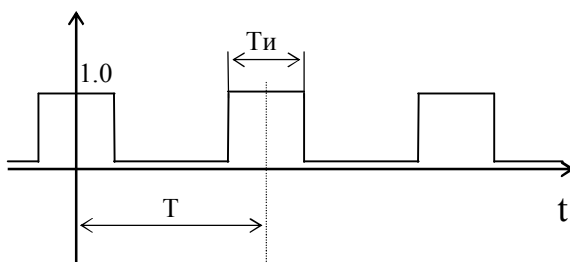


Рисунок 2.1 — Прямоугольные импульсы

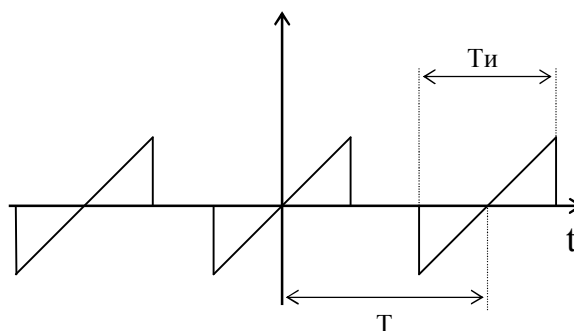


Рисунок 2.2 — Пилообразные импульсы

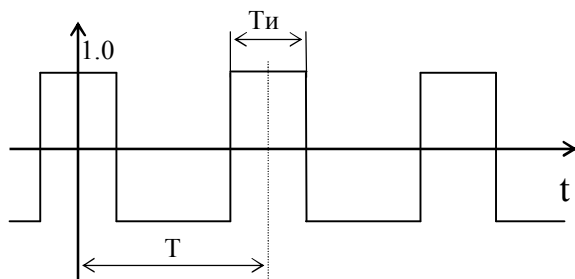


Рисунок 2.3 — Разнополярные
прямоугольные импульсы

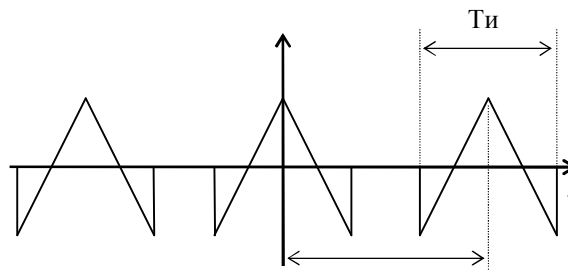


Рисунок 2.4 — Треугольные импульсы

2.4.2. Написать программу на языке пакета MATLAB, позволяющую:

- отображать рассчитанный спектр амплитуд заданного сигнала;
- определять относительную среднеквадратическую погрешность аппроксимации;
- синтезировать заданный сигнал по рассчитанным значениям коэффициентов и отображать как заданный исходный сигнал, так и синтезированный.

2.5. Содержание отчета

Отчет должен содержать следующие разделы:

- цель работы;
- постановка задачи (вариант задания);
- расчет коэффициентов разложения заданного сигнала в ряд Фурье;
- спектр амплитуд заданного сигнала;
- расчет погрешности аппроксимации;
- графики исходного и синтезированного сигнала;
- текст программы с комментариями;
- выводы.

2.6. Контрольные вопросы

- 2.6.1. Какая система базисных функций называется ортогональной (ортонормальной)?
- 2.6.2. Как определяется норма сигнала, и какой физический смысл имеет это понятие?
- 2.6.3. Запишите обобщенный ряд Фурье по системе ортогональных функций?
- 2.6.4. Как определяются коэффициенты обобщенного ряда Фурье?
- 2.6.5. Как определяется погрешность синтеза сигналов при конечном числе ортогональных составляющих ряда Фурье?
- 2.6.6. Какими свойствами обладают спектры периодических сигналов?
- 2.6.7. Как влияет изменение длительности импульса и периода повторения на спектр периодической последовательности прямоугольных видеоимпульсов?

- 2.6.6. Как отразится на спектре периодического сигнала изменение положения начала отсчета времени?
- 2.6.7. Как показать, что система функций $\{\cos k\Omega_1 t, \sin k\Omega_1 t\}$ является полной и ортогональной?
- 2.6.8. Как вычисляются коэффициенты ряда Фурье при разложении периодического сигнала по системе тригонометрических функций (по системе мультипликативно-ортогональных функций, по системе полиномиальных функций Чебышева, по системе функций Уолша)?

3. ЛАБОРАТОРНАЯ РАБОТА №3

«ГАРМОНИЧЕСКИЙ АНАЛИЗ СИГНАЛОВ НА ОСНОВЕ ДИСКРЕТНОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ»

3.1. Цель работы

Исследовать особенности дискретного преобразования Фурье (ДПФ) при гармоническом анализе детерминированных сигналов. Исследовать явления «размывания» спектра и «смещения» частот.

3.2. Краткие теоретические сведения

В практике цифровой обработки сигналов типична такая ситуация: непрерывный сигнал $x(t)$ на конечном интервале времени $(0, T)$ описывается своими отсчетными значениями $x[0], x[1], \dots, x[N-1]$, взятыми в моменты времени $t=nT$, где T — шаг дискретизации. Требуется по отсчетам данных выявить спектральный состав рассматриваемого сигнала.

Для решения рассматриваемой задачи может быть применено дискретное преобразование Фурье. Пара преобразований для обычного определения ДПФ задается выражениями [1, 4]:

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \exp(-j2\pi kn/N) \quad (3.1)$$

и

$$x[n] = \sum_{k=0}^{N-1} X[k] \exp(j2\pi kn/N), \quad (3.2)$$

где $x[n]$ — исходная N -точечная последовательность данных; $X[k]$ — результирующая N -точечная последовательность спектральных отсчетов.

Отметим следующие свойства ДПФ [1]:

- число спектральных отсчетов $X[k]$, вычисляемых по формуле (3.1), равно числу N отсчетов исходного временного ряда;
- спектральный отсчет $X[0]$ является средним значением всех отсчетов $x[n]$:

$$X[0] = \frac{1}{N} \sum_{n=0}^{N-1} x_n;$$

- если отсчетные значения $x[n]$ — вещественные числа, то спектральные отсчеты $X[k]$, номера которых располагаются симметрично относительно отсчета с номером $N/2$, образуют комплексно-сопряженные пары. Следовательно, для анализа амплитудного спектра можно использовать только первую половину отсчетов $X[k]$.

Одной из особенностей спектра дискретных сигналов является возможность появления ложных спектральных составляющих. На рис. 3.1 показано соотношение спектров исходного непрерывного сигнала и его дискретного аналога. Спектр

дискретного сигнала является периодическим с периодом, равным частоте дискретизации f_0 . Если частота дискретизации меньше верхней граничной частоты в спектре сигнала, то в результате наложения повторяющихся спектров образуются ложные спектральные составляющие [1, 4].

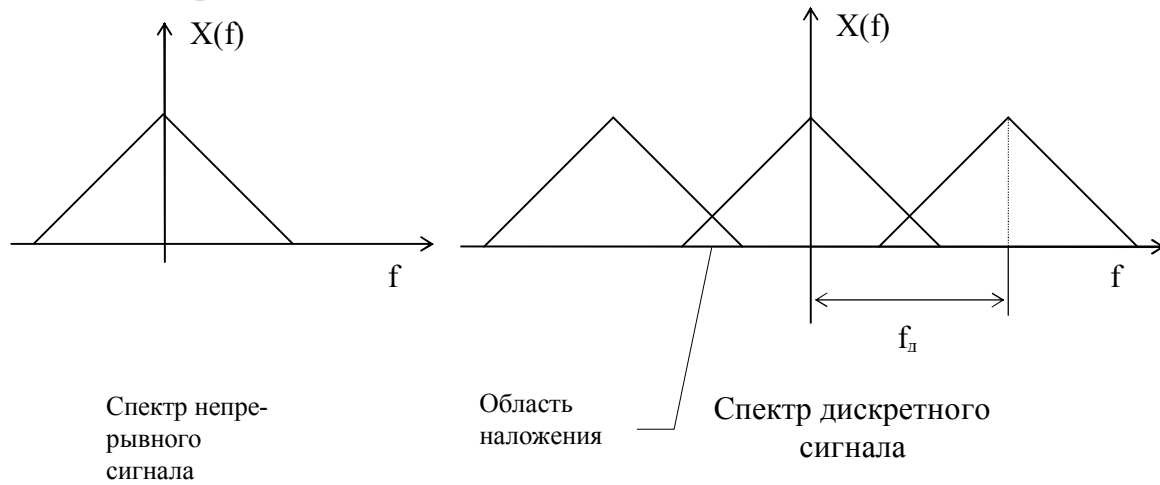


Рисунок 3.1 — Спектры непрерывного и дискретизированного сигналов

Иная особенность ДПФ состоит в «размывании» спектра [1]. Ограничение временного ряда $x[n]$ конечным числом значений равносильно умножению его на временное окно прямоугольной формы.

$$w[n] = \begin{cases} 1, n \leq N-1 \\ 0, n > 0 \end{cases}.$$

Поэтому последовательность наблюдаемых данных $x[n]$ из N отсчетов можно описать как произведение бесконечной последовательности $x_0[n]$ на $w[n]$:

$$x[n] = x_0[n] w[n].$$

ДПФ последовательности $x[n]$, выраженной через ДПФ последовательности $x_0[n]$ и временного окна $w[n]$, равно свертке этих преобразований:

$$X(f) = X_0(f) * W(f),$$

где

$$W(f) = \exp(-j2\pi f T [N-1]) \frac{\sin(\pi f T N)}{\sin(\pi f T)}.$$

Функция $W(f)$, называемая дискретной функцией **sinc**, или ядром Дирихле, представляет собой ДПФ прямоугольной функции. Преобразование Фурье наблюдаемой конечной последовательности $x[n]$ является искаженной версией преобразования Фурье бесконечной последовательности $x_0[n]$. Влияние прямоугольного окна на дискретно-временную синусоиду с частотой f_0 иллюстрирует рис.3.2.

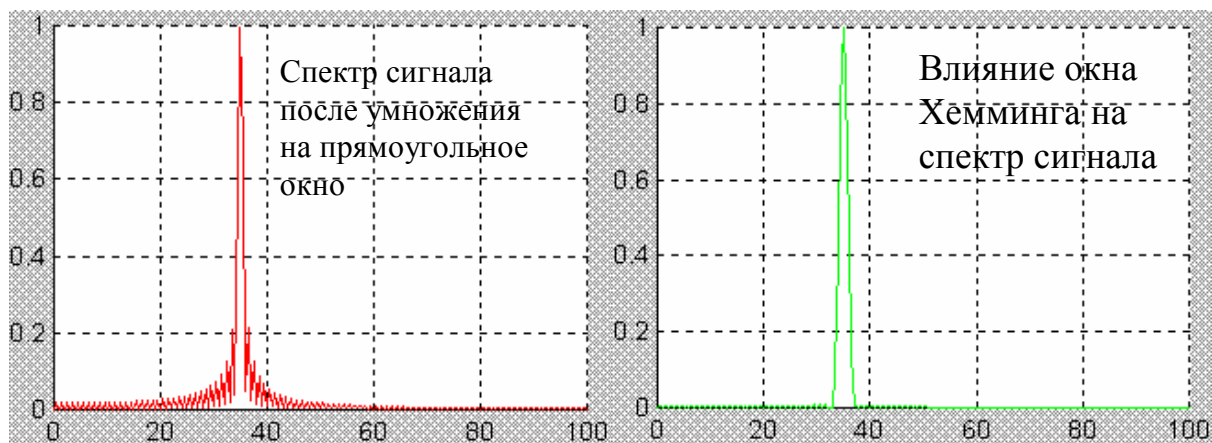
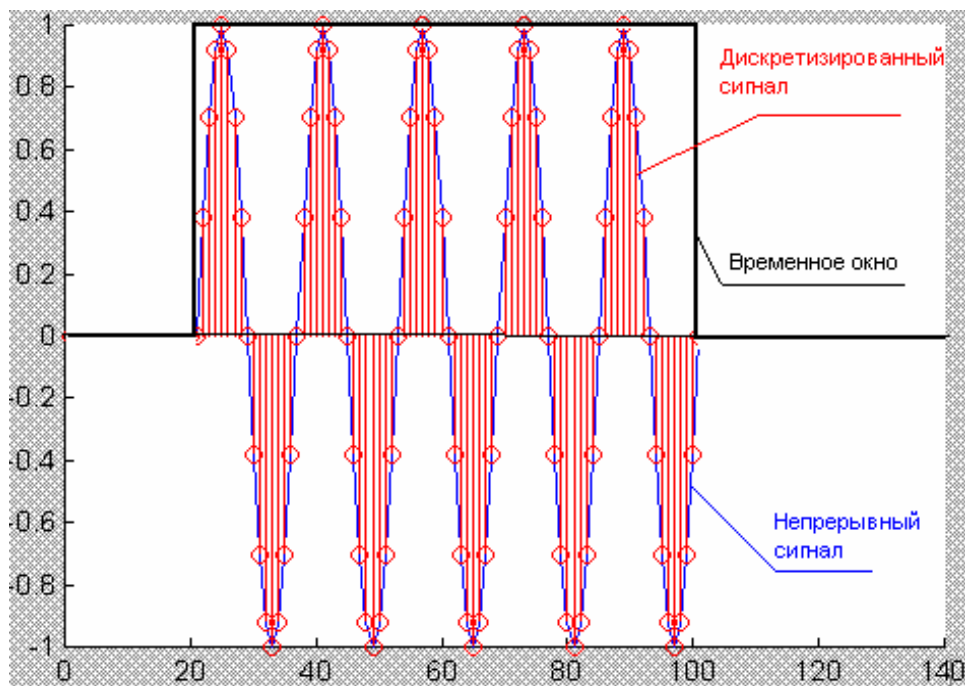


Рисунок 3.2 — Влияние окон на спектр дискретизированного сигнала

Из рисунка 3.2 видно, что острый спектральный пик ДПФ бесконечной последовательности расширяется за счет свертки со спектром окна, т.е. происходит «утечка» энергии в боковые лепестки. Боковые лепестки будут изменять (искажать) амплитуды соседних спектральных пиков.

В целях устранения влияния этого явления необходимо принять меры по локализации энергии у центральных частот. Обычно это достигается умножением временного ряда на соответствующим образом подобранное временное окно. Вместе с тем необходимо учитывать, что при «гашении» энергии в боковых лепестках возможна потеря информации об отдельных спектральных составляющих временного ряда.

Ниже приведены определения некоторых наиболее часто используемых функций $W[n]$ окон (в фигурных скобках приведены имена MATLAB-функций, реализующих окна [3]), $-N/2 \leq n \leq N/2-1$:

- треугольное {`triang`}:

$$w[n] = 1 - \frac{|n|}{N} \quad ;$$

- косинус-квадрат (окно Ханна) {hanning}:

$$w[n] = \cos^2\left(\frac{\pi n}{N}\right) = \frac{1}{2} \left(1 - \cos \frac{2\pi n}{N}\right) \quad ;$$

- приподнятый-косинус (окно Хэмминга) {hamming}:

$$w[n] = \alpha + (1 - \alpha) \cos \frac{2\pi n}{N} \quad , \quad \text{обычно } \alpha=0.54;$$

- усеченное гауссовское:

$$w[n] = \exp \left[-\frac{1}{2} \left(\frac{\alpha n}{N/2} \right)^2 \right], \quad \alpha \geq 2.5;$$

- окно Блекмана {blackman}:

$$w[n] = 0.42 - 0.50 \cos\left(\frac{2\pi n}{N}\right) + 0.08 \cos\left(\frac{4\pi n}{N}\right);$$

- окно Кайзера {kaiser}:

$$w[n] = \frac{I_0 \left[\pi \alpha \sqrt{1 - \left(\frac{n}{N/2} \right)^2} \right]}{I_0(\pi \alpha)} \quad ;$$

где $I_0(x)$ — модифицированная функция Бесселя первого рода нулевого порядка, α — константа (обычно $\alpha=3,0$).

- окно Чебышева {chebwin} задается частотной характеристикой.

Вычисление ДПФ требует выполнения N^2 операций. С целью сокращения количества операций ДПФ вычисляют по алгоритму быстрого преобразования Фурье (БПФ). С использованием БПФ требуется выполнить всего $2N \log_2 N$ операций. Например, при $N=2^{10}=1024$ объем вычислений сокращается примерно в 50 раз. Наиболее широкое распространение алгоритм БПФ получил при вычислении дискретных последовательностей, длина которых кратна двум. Вместе с тем разработаны и другие методы, основанные на преобразованиях числовых последовательностей иной длины. Для обращения к прямому и обратному алгоритму БПФ в MATLAB используются функции **fft** и **ifft**.

3.3. Варианты заданий

Варианты заданий приведены в таблице 3.1.

Таблица 3.1— Варианты заданий

№ варианта	Вид окна	f1	f2	f3	fsample 1	fsample 2
1	bartlett	10	14	30	120	50
2	triang	8	12	24	96	40
3	hanning	5	7	15	60	25
4	hamming	20	28	60	240	100
5	blackman	25	35	75	300	140
6	kaiser	15	22	45	160	80
7	bartlett	25	35	75	300	140
8	triang	20	28	60	240	100
9	hanning	5	7	15	60	25
10	hamming	8	12	24	96	40
11	blackman	10	14	30	120	50
12	kaiser	16	24	40	160	80
13	bartlett	8	11	24	56	40
14	triang	5	7	15	60	25
15	hanning	20	28	60	240	100
16	hamming	25	35	75	300	140
17	blackman	15	22	45	160	80
18	kaiser	10	14	30	120	50
19	bartlett	5	7	15	60	25
20	triang	20	28	60	240	100
21	hanning	25	35	75	300	140
22	Hamming	15	22	45	160	80
23	Blackman	25	35	75	300	140
24	Kaiser	20	28	60	240	100
25	Bartlett	10	14	30	120	50
26	Chebwin	8	12	24	96	40
27	Chebwin	5	7	15	60	25
28	Chebwin	20	28	60	240	100
29	Chebwin	25	35	75	300	140
30	Chebwin	15	22	45	160	80

3.4. Порядок выполнения лабораторной работы

3.4.1. Сформировать, используя пакет MATLAB, гармонические сигналы a, b, c:
secs=5;
t=(0:1/fsample:secs);


```
a=sin(2*pi*f1*t);
b=sin(2*pi*f2*t);
c=sin(2*pi*f3*t);
```

Здесь f_1 , f_2 , f_3 — частоты определяемые вариантом задания; f_{sample} — частота дискретизации; secs — длительность реализации сигнала. Варианты заданий см. в таблице 3.1.

3.4.2. Построить амплитудные спектры сигналов **a**, **b** и **c**, используя функцию **fft**. Например:

```
xa=abs(fft(a));
```

Для отображения графиков спектров воспользоваться функцией **plot**. Ось частот должна иметь значения от 0 до $f_{\text{sample}} - (f_{\text{sample}}/N)$, где N — количество отсчетов сигнала.

3.4.3. Сформировать сигнал, являющийся суммой сигналов **a**, **b** и **c**:

```
d=a(1:N)+0.75*b(1:N)+0.5*c(1:N);
```

и построить его амплитудный спектр:

```
AmpSpec=(1/(N/2))*abs(fft(d));
```

(для того, чтобы пронаблюдать "размытый" спектр дискретизированного сигнала необходимо уменьшить шаг дискретизации по частоте, для этого при формировании сигнала **d** необходимо заполнить нулями $N/2$ отсчетов перед началом реализации и $N/2$ отсчетов после окончания реализации).

3.4.4. Построить временные и спектральные графики прямоугольного окна и окна заданного по варианту задания. Описать свойства окна.

3.4.6. Сформировать сигнал **dw**, являющийся произведением сигнала **d** и сигнала **w**, представляющего заданное временное окно. Например:

```
w=hanning(N);
dw=(w1)*d;
```

3.4.7. Построить графики сигнала **dw** и его амплитудного спектра.

3.4.8. Сравнить реализации и спектры сигналов **d** и **dw**. Сформулировать выводы.

3.4.9. Повторить пункты 3.5.1-3.5.7 для иной частоты дискретизации в соответствии с вариантом задания.

3.5. Содержание отчета

Цель работы, постановка задачи и вариант задания, краткие теоретические сведения, графики реализаций и амплитудных спектров сигналов **a**, **b**, **c**, **d**, **dw** для различных частот дискретизации, временные и спектральные графики окон в соответствии с вариантом, текст программы с комментариями, выводы.

3.6. Контрольные вопросы

3.6.1. Как выражаются прямое и обратное дискретное преобразования Фурье?

3.6.2. Назовите и объясните свойства ДПФ?

- 3.6.3. Объясните «механизм» наложения частот?
- 3.6.4. Объясните причину «размывания» спектральных составляющих при выполнении ДПФ?
- 3.6.5. Почему весовые окна позволяют уменьшить «утечку» энергии?
- 3.6.6. Назовите и охарактеризуйте основные параметры весовых окон?
- 3.6.7. Приведите определения и свойства окон Ханна и Хемминга?
- 3.6.8. В чем преимущества применения БПФ?
- 3.6.9. Сформулируйте основную идею алгоритма БПФ?
- 3.6.10. Выполните вывод формул БПФ.
- 3.6.11. Объясните операцию БПФ «бабочка».
- 3.6.12. Нарисуйте граф 8-точечного БПФ.
- 3.6.13. Объясните принцип бит-инверсного упорядочения при вычислении БПФ.
- 3.6.14. Напишите на языке пакета MATLAB программу прямого и обратного дискретного преобразования Фурье?

4. ЛАБОРАТОРНАЯ РАБОТА №4 «ЛИНЕЙНЫЕ ДИСКРЕТНЫЕ СИСТЕМЫ И ЦИФРОВЫЕ ФИЛЬТРЫ»

4.1. Цель работы

Изучение основных понятий линейных дискретных систем и исследование методов реализации цифровых фильтров во временной и частотной областях.

4.2. Краткие теоретические сведения

Выходная последовательность $y(nT)$ линейной дискретной системы является откликом на входную последовательность $x(nT)$ [1]:

$$y(nT) = \sum_{k=-\infty}^{\infty} h(kT)x(nT - kT) \quad . \quad (4.1)$$

Выходная последовательность $y(nT)$ это свертка входной последовательности $x(nT)$ с импульсной характеристикой линейной дискретной системы, т. е.

$$Y(nT) = h(nT)*x(nT) \quad . \quad (4.2)$$

Импульсная характеристика (ИХ) дискретной линейной системы определяется как отклик системы на входное воздействие

$$x(nT) = \begin{cases} 1, & \text{при } n = 0 \\ 0, & \text{при } n \neq 0 \end{cases} \quad .$$

В цифровой обработке сигналов рассматриваются устойчивые и физически реализуемые линейные дискретные системы. Система является устойчивой, если

$$\sum_{n=-\infty}^{\infty} |h(nT)| < \infty \quad .$$

Система является физически реализуемой, если $h(nT)=0$ при $n<0$.

В общем случае для линейной дискретной системы последовательности $x(nT)$ и $y(nT)$ связаны между собой разностным уравнением [1, 4]:

$$y[n] = \sum_{k=0}^{N-1} b[k]x[n - k] - \sum_{k=1}^{M-1} a[k]y[n - k], \quad (4.3)$$

где $a[k]$ и $b[k]$ — постоянные коэффициенты, определяющие свойства системы.

Выражение (4.3) представляет алгоритм функционирования цифрового фильтра. В зависимости от вида импульсной характеристики цифровые фильтры принято делить на два класса: КИХ-фильтры (фильтры с конечной импульсной характеристикой) и БИХ (фильтры с бесконечной импульсной характеристикой). Отметим, что уравнение (4.3) определяет БИХ-фильтр. Одна из структурных схем БИХ-фильтра приведена рис.7. Фильтр, функционирующий на основе (4.3), также называют рекурсивным фильтром.

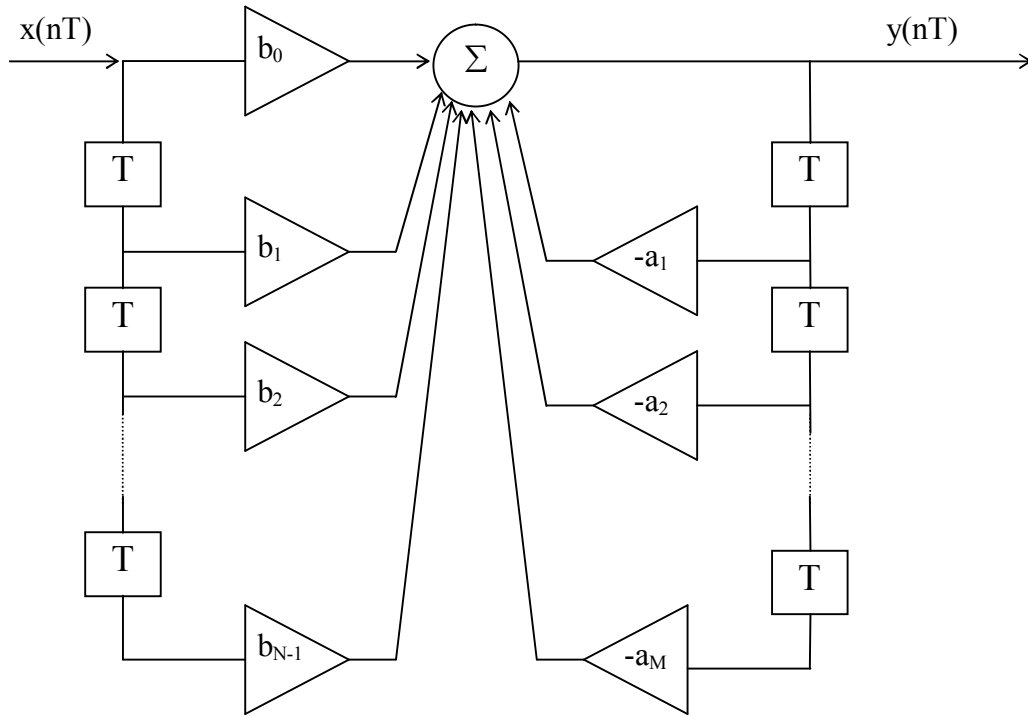


Рисунок 4.1 — Структурная схема БИХ-фильтра

Если все коэффициенты $a[k]$ в уравнении (4.3) равны нулю, то

$$y[n] = \sum_{k=0}^N b[k] x[n-k]. \quad (4.4)$$

Фильтр (4.4) имеет конечную длительность импульсной характеристики, равную NT . В этом случае $b[k]=h[k]$, т. е. коэффициенты КИХ-фильтра представляют отсчеты его импульсной характеристики. Выражение (4.4) определяет нерекурсивный фильтр.

Передаточной функцией $H(z)$ называют отношение z -образов выходного $Y(z)$ и входного $X(z)$ сигналов фильтра при нулевых начальных условиях: $H(z)=Y(z)/X(z)$ [1, 4]. Для рекурсивного и нерекурсивного фильтров передаточные функции соответственно равны:

$$H(z) = \frac{\sum_{k=0}^{N-1} b[k] z^{-k}}{1 + \sum_{k=1}^{M-1} a[k] z^{-k}},$$

$$H(z) = \sum_{k=0}^{N-1} b[k] z^{-k}.$$

Коэффициенты фильтров являются коэффициентами соответствующих передаточных функций. Комплексные частотные характеристики представляют функции, полученные в результате подстановки $z=\exp(j\omega T)$ в передаточные функ-

ции. Модуль комплексной частотной характеристики $A(\omega) = |H(\exp(j\omega T))|$ называется амплитудно-частотной характеристикой (АЧХ). Аргумент комплексной частотной характеристики $\varphi(\omega) = \arg[H(\exp(j\omega T))]$ называется фазочастотной характеристикой (ФЧХ).

Частотные характеристики ЦФ с вещественными коэффициентами обладают следующими свойствами:

- 1) все частотные характеристики представляют собой периодические функции частоты с периодом ω_d ;
- 2) АЧХ представляет собой четную функцию частоты, а ФЧХ — нечетную функцию.

Требования к частотным характеристикам необходимо задавать на интервале $[0, \pi/T]$. С целью упрощения сопоставимости частотных характеристик фильтров выполняют нормировку частоты одним из двух способов. При первом способе используется нормированная частота $\bar{\omega} = \omega T$, тогда $\bar{\omega}_d = \omega_d * T = 2\pi$ и требования к частотным характеристикам задают на интервале $[0, \pi]$. При втором способе полагают нормированной частоту $w = \omega T/2\pi$, тогда $w_d = \omega_d * T/2\pi = 1$ и требования к частотным характеристикам задают в интервале $[0; 0.5]$. При этом изменяются аргументы в обозначениях частотных характеристик $H(e^{j\omega T})$, $A(w)$, $\varphi(w)$.

Импульсная и частотная характеристики фильтров связаны парой преобразования Фурье:

$$\begin{aligned} h(nT) &= F^{-1} \{H(e^{j\omega T})\}, \\ H(e^{j\omega T}) &= F \{h(nT)\}. \end{aligned} \quad (4.5)$$

Исходя из этого, возможна реализация фильтров не только во временной области, в соответствии с (4.1), но и в частотной. При этом свертке (4.1) будут соответствовать умножение преобразований Фурье импульсной характеристики $h(nT)$ и входного сигнала $x(nT)$.

Пакет MATLAB содержит ряд функций для исследования линейных дискретных систем и цифровой фильтрации.

Y=FILTER(B,A,X) — фильтрация данных из вектора X с помощью фильтра, функционирующего на основе выражения (4.3). Коэффициенты фильтра хранятся в векторах B, A.

[h,w]=freqz(B,A,N) — вычисление нормированной комплексной частотной характеристики фильтра с коэффициентами, которые хранятся в векторах B, A. Результат h — вектор значений комплексной частотной характеристики, w — вектор частот, N — размерность векторов h и w.

В общем случае для реализации БИХ-фильтрации в частотной области, необходимо умножить ДПФ от входной последовательности $x(nT)$ на отношение ДПФ коэффициентов фильтра:

$$\begin{aligned} n &= \text{lenght}(x); \\ y &= \text{ifft}(\text{fft}(x) .* \text{fft}(b,n) ./ \text{fft}(a,n)). \end{aligned}$$

Это позволяет вычислить вектор y , который соответствует применению функции **filter**, с точностью до переходных процессов.

4.3. Варианты заданий

Заданы передаточные функции линейных дискретных систем вида:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}}$$

В таблице 4.1 приведены коэффициенты передаточной функции, согласно варианту задания.

Таблица 4.1— Варианты заданий

Вариант	a0	a1	a2	b0	b1	b2
1	1	-1.5610	0.6414	0.0201	0.0402	0.0201
2	1	-1.1430	0.4128	0.0675	0.1349	0.0675
3	1	-0.7478	0.2722	0.1311	0.2622	0.1311
4	1	-0.3695	0.1958	0.2066	0.4131	0.2066
5	1	0	0.1716	0.2929	0.5858	0.2929
6	1	-1.1430	0.4128	0.6389	-1.2779	0.6389
7	1	-1.5610	0.6414	0.8006	-1.6012	0.8006
8	1	-0.7478	0.2722	0.5050	-1.0100	0.5050
9	1	-0.3695	0.1958	0.3913	-0.7827	0.3913
10	1	0	0.1716	0.2929	-0.5858	0.2929
11	1	-0.1842	0.1776	0.3404	-0.6809	0.3404
12	1	-0.5570	0.2270	0.4460	-0.8920	0.4460
13	1	-0.9428	0.3333	0.5690	-1.1381	0.5690
14	1	-1.3490	0.5140	0.7157	-1.4315	0.7157
15	1	-1.7786	0.8008	0.8949	-1.7897	0.8949
16	1	-1.7786	0.8008	0.0055	0.0111	0.0055
17	1	-1.3490	0.5140	0.0413	0.0825	0.0413
18	1	-0.9428	0.3333	0.0976	0.1953	0.0976
19	1	-0.5570	0.2270	0.1675	0.1675	0.3350
20	1	-0.1842	0.1776	0.2483	0.4967	0.2483
21	1	-1.5610	0.6414	0.0201	0.0402	0.0201
22	1	-1.1430	0.4128	0.0675	0.1349	0.0675
23	1	-0.7478	0.2722	0.1311	0.2622	0.1311
24	1	-0.3695	0.1958	0.2066	0.4131	0.2066
25	1	0	0.1716	0.2929	0.5858	0.2929
26	1	-1.1430	0.4128	0.6389	-1.2779	0.6389
27	1	-1.5610	0.6414	0.8006	-1.6012	0.8006
28	1	-0.7478	0.2722	0.5050	-1.0100	0.5050
29	1	-0.3695	0.1958	0.3913	-0.7827	0.3913
30	1	0	0.1716	0.2929	-0.5858	0.2929

4.4. Порядок выполнения лабораторной работы

4.4.1. Построить частотную характеристику фильтра с помощью функции `freqz`.

4.4.2. Построить импульсную характеристику фильтра, как реакцию фильтра на единичное входное воздействие.

4.4.3. Сформировать сигнал $x(nT)$ в виде суммы 3-х гармонических косинусоидальных сигналов. Частоты косинусов $\omega_1, \omega_2, \omega_3$ выбрать таким образом, чтобы фильтр пропускал гармоники ω_1 и ω_2 , и подавлял ω_3 .

4.4.4. Выполнить фильтрацию сигнала $x(nT)$ во временной области.

4.4.5. Выполнить фильтрацию сигнала $x(nT)$ в частотной области.

4.4.6. Построить графики выходных процессов в соответствии с п.4.4.4. и п.4.4.5 в одной системе координат.

4.5. Содержание отчета

Цель работы, постановка задачи, краткие теоретические сведения, текст программы с комментариями, частотная характеристика фильтра, импульсная характеристика фильтра, входной процесс $x(nT)$, графики выходных процессов при фильтрации во временной и частотной областях, выводы.

4.6. Контрольные вопросы

4.6.1. Запишите соотношение, устанавливающее взаимосвязь входного и выходного процесса линейной дискретной системы.

4.6.2. Что понимают под импульсной характеристикой?

4.6.3. Что такое передаточная функция линейной дискретной системы?

4.6.6. Что такое АЧХ и ФЧХ?

4.6.7. Сформулируйте свойства комплексной частотной характеристики?

4.6.8. Сформулируйте понятия устойчивости и физической реализуемости дискретной линейной системы?

4.6.9. Напишите на языке Паскаль (Си) функцию, вычисляющую выходной сигнал рекурсивного и нерекурсивного фильтра.

4.6.10. Объясните возможность реализации фильтрации в частотной области?

4.6.11. Каким образом выполняют нормирование частоты при построении АЧХ дискретных линейных систем?

4.6.12. Как взаимосвязаны импульсная и частотная характеристики?

5. ЛАБОРАТОРНАЯ РАБОТА №5 «СИСТЕМЫ С МНОГОЧАСТОТНОЙ ДИСКРЕТИЗАЦИЕЙ»

5.1. Цель работы

Исследование свойств линейных дискретных систем с повышением и понижением частоты дискретизации.

5.2. Краткие теоретические сведения

Восходящей дискретной системой (ВДС) называется система, частота дискретизации сигнала на выходе которой, выше частоты дискретизации входного сигнала. **Нисходящей дискретной системой (НДС)** называется система, частота дискретизации на выходе которой, ниже частоты дискретизации входного сигнала [1].

Каждая ВДС содержит элемент, увеличивающий частоту дискретизации — интерполятор, находящийся на входе системы, и дискретный фильтр, выполняющий последующую обработку сигнала с выходной частотой дискретизации. Каждая НДС состоит из дискретного фильтра, выполняющего предварительную обработку входного сигнала с входной частотой дискретизации, и элемента, уменьшающего частоту дискретизации — дециматора, находящегося на выходе системы.

Интерполятор, увеличивающий частоту дискретизации в M раз (M -целое), представляет блок, преобразующий входной дискретный сигнал, описываемый решетчатой функцией $x(nT')$ с периодом повторения T' , в выходной дискретный сигнал, описываемый решетчатой функцией $y(nT)$ с периодом повторения $T=T'/M$ по алгоритму:

$$y(nT) = \begin{cases} x\left(\frac{n}{M} \cdot T'\right) & \text{при } n = 0, M, 2M, \dots \\ 0 & \text{при других } n \end{cases}.$$

Таким образом, последовательность $y(nT)$ получается путем ввода между соседними отсчетами последовательности $x(nT')$ дополнительных $(M-1)$ отсчетов с нулевыми значениями. На рис.5.1. показан интерполятор и последовательности на входе и выходе интерполятора при $M=3$.

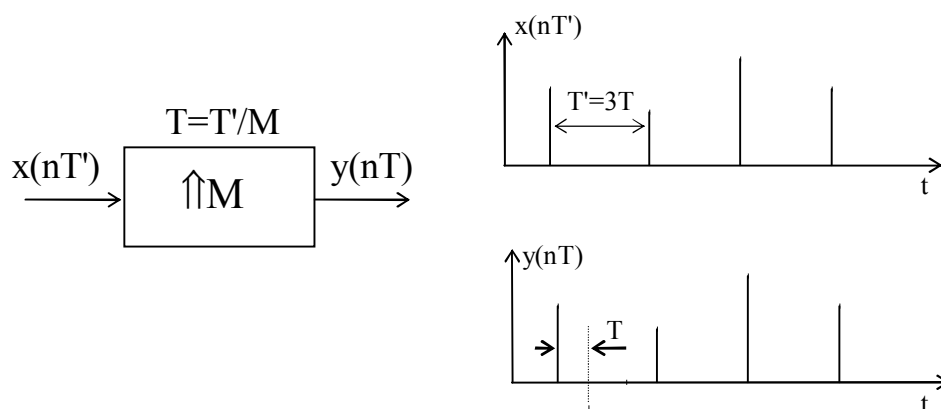


Рисунок 5.1 — Интерполятор

Спектр дискретного сигнала $x(nT')$ периодичен по оси частот с частотой дискретизации $\omega_d = 2\pi/T'$. Модуль спектра $X(e^{j\omega T'})$ условно показан на рис.5.2 (график 2),

Спектр сигнала на выходе интерполятора будет иметь тот же вид, что и спектр входного сигнала (рис.5.2, график 2). Действительно, поскольку последовательность $y(nT)$ отлична от нуля только при $n=0, M, 2M \dots$, то в выражении для спектра сигнала $y(nT)$

$$Y(e^{j\omega T}) = \sum_{n=0}^{\infty} y(nT) e^{-j\omega nT}$$

можно заменить n на nM . Тогда

$$\begin{aligned} Y(e^{j\omega T}) &= \sum_{n=0}^{\infty} y(nMT) e^{-j\omega nMT} = \sum_{n=0}^{\infty} x(nT') e^{-j\omega nMT} = \\ &= X(e^{j\omega T'}) = \frac{1}{MT} \sum_{k=-\infty}^{\infty} X \left[j \left(\omega + k \frac{2\pi}{T'} \right) \right]. \end{aligned}$$

Отсюда следует, что спектр сигнала $y(nT)$ будет периодически повторяться со "старой" частотой дискретизации $\omega'_d = 2\pi/T'$, действующей на входе интерполятора.

Если бы сигнал $y(nT)$ получался непосредственно путем дискретизации исходного аналогового сигнала $x(t)$, то его спектр определялся бы выражением

$$Y^*(e^{j\omega T}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X \left[j \left(\omega + k \frac{2\pi}{T} \right) \right]$$

и был бы периодичен с частотой $\omega_d = 2\pi/T$ (рис. 5.2., график 3).

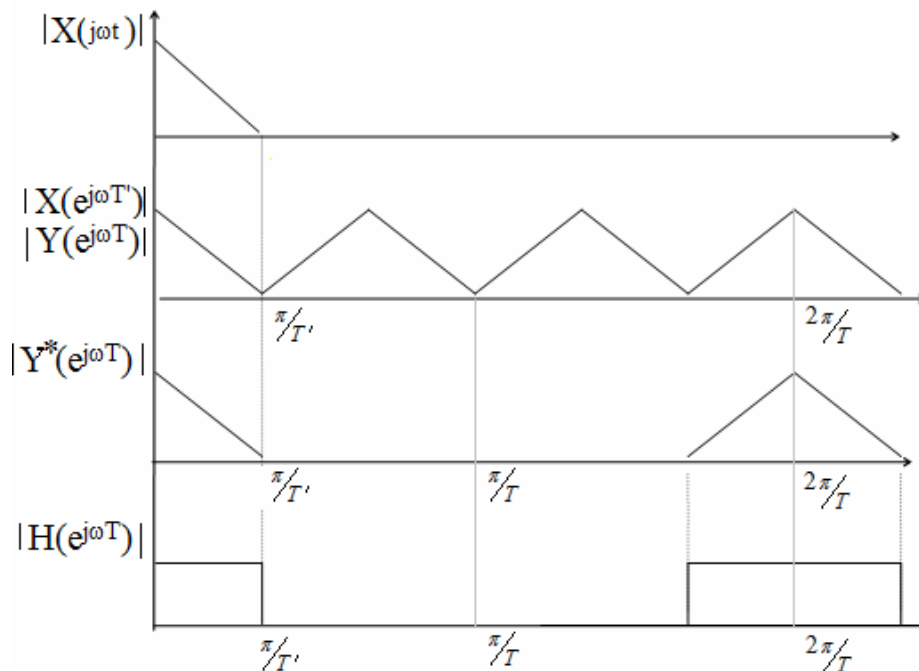


Рисунок 5.2 — Спектры сигналов $x(nT)$ и $y(nT)$

Из сравнения графиков 2 и 3 видно, что спектр $Y(e^{j\omega T})$ сигнала на выходе интерполятора отличается от требуемого $Y^*(e^{j\omega T})$ наличием "лишних" частотных составляющих и величиной постоянного множителя [1]. Следовательно, требуемый сиг-

нал $y^*(nT)$ может быть получен из $y(nT)$ с помощью дискретного фильтра нижних частот, подавляющего "лишние" частотные составляющие. Идеализированная АЧХ такого фильтра показана на рис. 5.2 (график 4). В общем случае идеализированная АЧХ фильтра должна удовлетворять требованиям:

$$|H(e^{j\omega T})| = \begin{cases} M & \text{при } \omega < \pi/T, \\ 0 & \text{при } \pi/T < \omega < \pi/T. \end{cases}$$

Такой фильтр устанавливают на выходе интерполятора.

Уменьшение частоты дискретизации в M раз осуществляется с помощью дециматора. В этом случае из входного дискретного сигнала, описываемого решетчатой функцией $x(nT)$ с периодом повторения T , берется только каждый M -ый отсчет: $y(nT) = y(nMT) = x(nMT)$.

Спектр $Y(e^{j\omega MT})$ выходного сигнала дециматора, уменьшающего частоту дискретизации сигнала $x(nT)$ в M раз, имеет вид:

$$Y(e^{j\omega MT}) = \frac{1}{M} \sum_{n=0}^{M-1} X(e^{j\omega T + j2\pi \frac{n}{M}}).$$

Уменьшение частоты дискретизации сигнала $x(nT)$, основной спектр которого занимает полосу частот от 0 до π/T , приводит к наложению спектров (рис. 5.3).

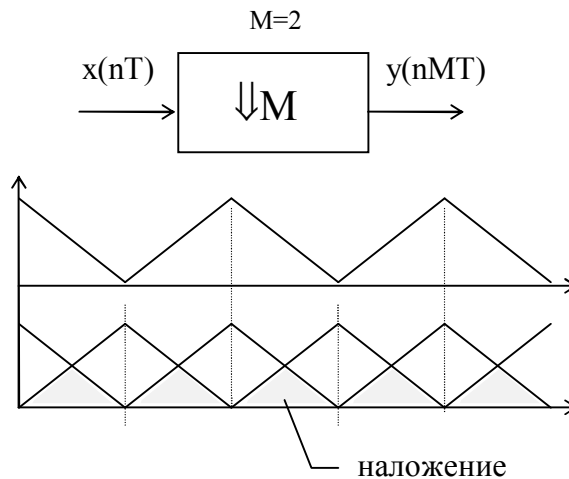


Рисунок 5.3— Децимация и наложение спектров

Для того, чтобы уменьшение частоты дискретизации в M раз не приводило к наложению спектров, спектр $X(e^{j\omega T})$ входного сигнала $x(nT)$ дециматора не должен содержать гармонических составляющих в диапазоне от частот от π/MT до π/T . Следовательно, на входе дециматора должен быть установлен фильтр нижних частот с АЧХ:

$$|H(e^{j\omega T})| = \begin{cases} 1 & \text{при } |\omega| < \pi/MT \\ 0 & \text{при } |\omega| > \pi/MT \end{cases}.$$

Система, в которой увеличение (уменьшение) частоты дискретизации производится за один прием (однократно), называется соответственно простейшей восходящей линейной системой (ПВДС) и простейшей нисходящей линейной системой (ПНДС).

ПВДС представлена на рис. 5.4. Входной сигнал $x(nT')=x(nMT)$ с интервалом дискретизации $T'=MT$ поступает на интерполятор, увеличивающий частоту дискретизации в M раз. Выходной сигнал $y(nT)$ ПВДС

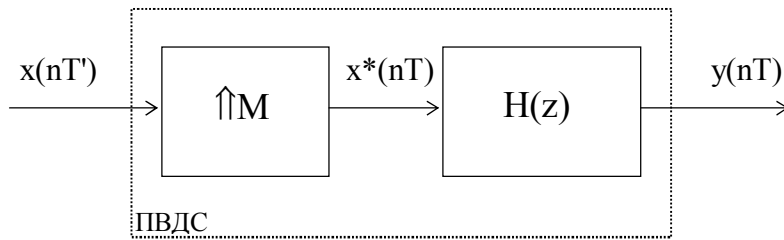


Рисунок 5.4 — Простейшая восходящая дискретная система

получается в результате обработки сигнала $x^*(nT)$ дискретным фильтром с передаточной функцией $H(z)$, работающим на повышенной частоте дискретизации.

ПНДС изображена на рис.5.5. Входной сигнал $x(nT)$ с периодом дискретизации T обрабатывается дискретным фильтром с передаточной функцией $H(z)$. На выходе фильтра установлен дециматор, уменьшающий частоту дискретизации в M раз.

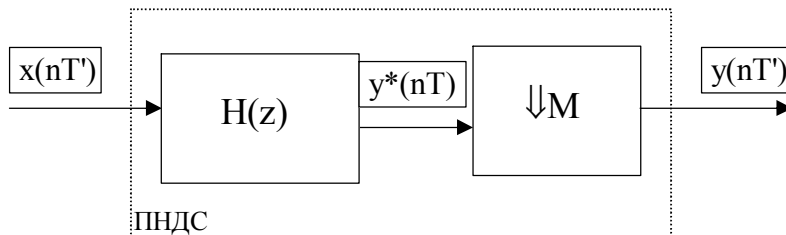


Рисунок 5.5 — Простейшая нисходящая дискретная система

Возможно последовательное соединение ПВДС и ПНДС (рис. 5.6). В этом случае выполняется изменение частоты дискретизации в M/L раз, т.е. выполняется дробное преобразование частоты [1].

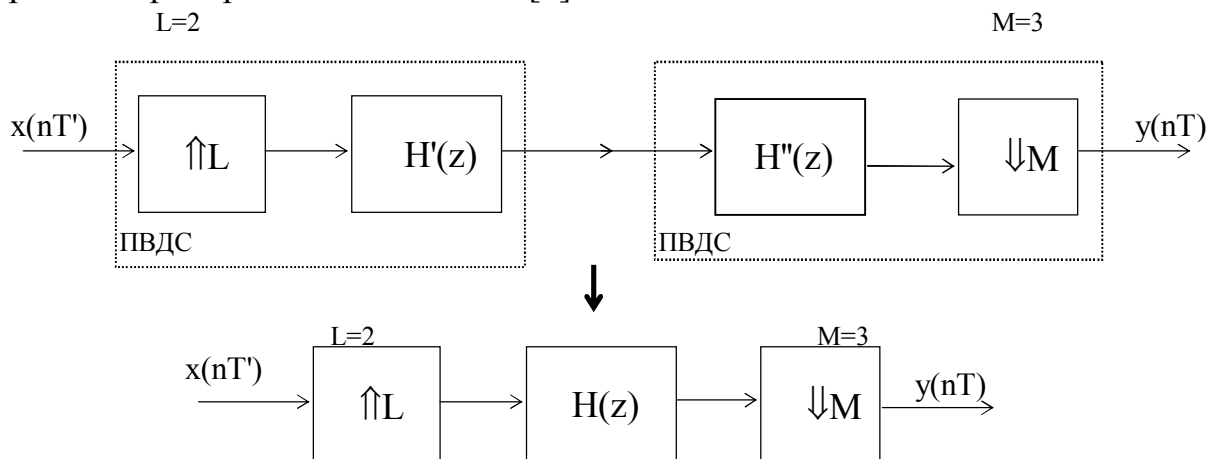


Рисунок 5.6 — Дробное преобразование частоты дискретизации

5.3. Варианты заданий

Варианты заданий приведены в таблице 5.1.

Таблица 5.1 — Варианты заданий

Вариант	$f_1, \text{Гц}$	$f_2, \text{Гц}$	$f_d', \text{Гц}$	$f_d'', \text{Гц}$	$f_d''', \text{Гц}$
1	750	1050	8000	2000	40000
2	1000	1550	9000	3000	36000
3	800	2550	10000	5000	30000
4	900	1150	11000	2200	22000
5	850	1550	12000	3000	60000
6	1200	1675	13000	3250	65000
7	2350	3550	14000	7000	42000
8	750	1550	15000	3000	30000
9	1500	2050	16000	4000	80000
10	900	2050	8000	4000	16000
11	1700	2300	9000	4500	27000
12	560	1050	10000	2000	20000
13	870	1425	11000	2750	55000
14	1350	2050	12000	4000	48000
15	2700	3300	13000	6500	39000
16	950	1800	14000	3500	70000
17	2600	3800	15000	7500	45000
18	1200	1650	16000	3200	32000
19	500	8050	8000	1600	16000
20	600	1175	9000	2250	45000
21	850	1300	10000	2500	50000
22	2000	2800	11000	5500	33000
23	750	1250	12000	2400	24000
24	1100	1350	13000	2600	26000
25	1200	1450	14000	2800	28000
26	1900	2550	15000	5000	30000
27	3850	4050	16000	8000	48000
28	760	950	9000	1800	18000
29	880	1050	10000	2000	20000
30	1350	1425	11000	2750	55000

5.4. Порядок выполнения лабораторной работы

5.4.1 Исследовать ПНДС:

- сформировать двухтональный дискретный сигнал на интервале $1c$: $x(t) = \sin(2\pi f_1 t) + 0.75 \sin(2\pi f_2 t)$ с частотой дискретизации f_d' ;
- используя окно Хемминга построить спектр сигнала $x(t)$. Конечное значение на оси частот равно f_d' ;

- задать новую частоту дискретизации f_d'' и определить коэффициент децимации f_d'/f_d'' ;
- синтезировать ФНЧ ПНДС с помощью функции FIR1 пакета MATLAB ($h=fir1(\text{"порядок фильтра"}, \text{"частота среза"})$, где h — импульсная характеристика фильтра). Рекомендуется использовать 20-ый порядок фильтра;
- построить частотную характеристику синтезированного фильтра с помощью БПФ. Конечное значение на оси частот равно f_d' ;
- реализовать ПНДС в соответствии с рис 5.5. Фильтрацию сигнала реализовать с использованием функции filter пакета MATLAB. Децимацию выходного сигнала фильтра можно выполнить следующим образом:


```
d=filter(h,1,x);           % фильтрация
      i=(1:1:N/decimateBy);   % новая индексация
      y=d(i*decimateBy);      % децимация.
```

 Здесь переменная decimateBy — коэффициент децимации; N — соответствует размерности вектора x, хранящего отсчеты сигнала $x(t)$;
- используя окно Хеннинга построить спектр сигнала y. Конечное значение на оси частот равно f_d'' ;
- выполнить анализ полученных результатов и сделать выводы.

5.4.2. Исследовать ПВДС:

- сформировать двухтональный дискретный сигнал на интервале 0.1с: $x(t)=\sin(2\pi f_1 t)+0.75\sin(2\pi f_2 t)$ с частотой дискретизации f_d' ;
- используя окно Хемминга построить спектр сигнала $x(t)$. Конечное значение на оси частот равно f_d' ;
- задать новую частоту дискретизации f_d''' и определить коэффициент интерполяции f_d''/f_d''' ;
- выполнить интерполяцию сигнала $x(t)$:


```
d=zeros(1,N*interpolateBy);
      i=(1:1:N);
      d(i*interpolateBy)=x(i);
```

 Здесь переменная interpolateBy — коэффициент интерполяции; N — соответствует размерности вектора x, хранящего отсчеты сигнала $x(t)$.
- используя окно Хемминга построить спектр сигнала d. Конечное значение на оси частот равно f_d''' ;
- синтезировать ФНЧ ПВДС с помощью функции FIR1 пакета MATLAB. Порядок фильтра — 100.
- построить частотную характеристику синтезированного фильтра с помощью БПФ. Конечное значение на оси частот равно f_d''' ;
- реализовать ПВДС выполнив фильтрацию вектора d:


```
y=filter(h,1,d);
```
- построить спектр сигнала y;
- выполнить анализ полученных результатов и сделать выводы.

5.5. Содержание отчета

Цель работы, постановка задачи, краткие теоретические сведения, текст программы с комментариями, временная реализация и спектр двухтонального дискретного сигнала, частотная и импульсная характеристики ФНЧ ПНДС, временная реализация и спектр сигнала после фильтрации ФНЧ ПНДС, временная реализация и спектр сигнала после децимации ПНДС, временная реализация и спектр сигнала после интерполяции ПВДС, частотная и импульсная характеристики ФНЧ ПВДС, временная реализация и спектр сигнала после фильтрации ФНЧ ПВДС, выводы по работе.

5.6. Контрольные вопросы

- 5.6.1. Что понимают под ВДС?
- 5.6.2. Что понимают под НДС?
- 5.6.3. Сформулируйте алгоритм функционирования интерполятора?
- 5.6.4. Как соотносятся спектры входного и выходного сигналов интерполятора?
- 5.6.5. Какие требования предъявляются к ФНЧ ПВДС?
- 5.6.6. Сформулируйте алгоритм функционирования дециматора?
- 5.6.7. Как соотносятся спектры входного и выходного сигналов дециматора?
- 5.6.8. Какие требования предъявляются к ФНЧ ПНДС?
- 5.6.9. Каким образом выполняется дробное преобразование частоты?
- 5.6.10. Какие требования предъявляются к фильтру при дробном преобразовании частоты?

6. ЛАБОРАТОРНАЯ РАБОТА №6

«КРАТКОВРЕМЕННЫЙ СПЕКТРАЛЬНЫЙ АНАЛИЗ И ГОМОМОРФНАЯ ОБРАБОТКА РЕЧЕВЫХ СИГНАЛОВ»

6.1. Цель работы

Исследование спектральных методов обработки речевых сигналов. Определение параметров речевых сигналов по спектрограмме и кепстру.

6.2. Краткие теоретические сведения

Голосовой тракт человека представляет собой акустическую трубу, которая с одной стороны оканчивается голосовыми связками, а с другой губами. Форма голосового тракта определяется положением губ, челюстей языка и мягкого неба.

Звуки в этой системе образуются тремя способами. Вокализованные (звонкие) звуки — путем возбуждения голосового тракта квазипериодическими импульсами воздушного давления, создаваемыми вибрациями голосовых связок. Фрикативные звуки образуются проталкиванием воздуха через сужения в определенных областях голосового тракта, в результате чего возникает турбулентия, которая является источником шума, возбуждающего голосовой тракт. Взрывные звуки образуются путем создания избыточного давления в области полного смыкания голосового тракта с последующим его быстрым размыканием. Все эти источники создают широкополосное возбуждение голосового тракта, который в свою очередь действует как линейный фильтр с изменяющимися во времени параметрами.

На рис. 6.1 приведена цифровая модель источника речи.

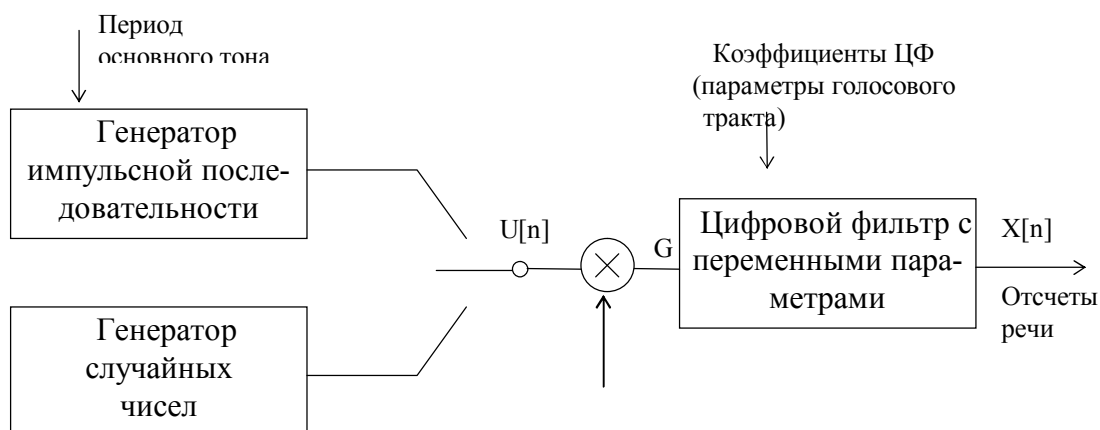


Рисунок 6.1— Цифровая модель источника речи

Предполагается, что в этой модели дискретные отсчеты речевого сигнала формируются на выходе ЦФ с переменными параметрами, который аппроксимирует передаточные свойства голосового тракта. На временном интервале

порядка 10ms характеристики ЦФ можно считать неизменными. На каждом таком интервале ЦФ может быть охарактеризован совокупностью своих коэффициентов. В случае вокализованной речи ЦФ возбуждается генератором квазипериодической импульсной последовательности, расстояние между соседними импульсами которого соответствует периоду основного тона. На интервалах невокализованной речи ЦФ возбуждается генератором случайных чисел, который вырабатывает шумовой сигнал с равномерной спектральной плотностью. В обоих случаях сигнал, поступивший на ЦФ, управляется по амплитуде.

На рассмотренной модели базируются многочисленные способы обработки речевых сигналов. В лабораторной работе рассматриваются кратковременный спектральный анализ и гомоморфная обработка речи как наиболее широко используемые способы обработки речевых сигналов.

В основу кратковременного спектрального анализа положено фундаментальное допущение о том, что на длительном интервале речь является нестационарным процессом, однако в пределах короткого интервала ее можно считать стационарной. Таким образом, преобразование Фурье, выполненное над коротким отрезком речи, должно обеспечить удовлетворительное спектральное описание речи на этом временном интервале. На измерении кратковременного спектра основана работа полосного вокодера, спектрографов и некоторых систем распознавания речи. Для реализации кратковременного анализа Фурье обычно применяется алгоритм быстрого преобразования Фурье (БПФ).

Кратковременное дискретное преобразование Фурье определяется следующим образом

$$X_l(\omega) = T_0 \sum_{n=0}^{N-1} x_l[n] e^{-j\omega n T_0}, \quad (6.1)$$

где $x_l[n]$ представляет отрезок речи, взвешенный окном $w[n]$, длиной N отсчетов:

$$x_l[n] = w[n]x[n + lN], \quad l = 0, 1, 2, \dots \quad (6.2)$$

Частотное разрешение спектрального анализа обратно пропорционально длительности окна N , что иллюстрируется на рис. 6.2. В качестве весовой функции здесь использовано окно Хемминга с большим значением $N=1024$ (при $f_d=22050$ Гц частотное разрешение равно $22050/1024=21,5$ Гц).

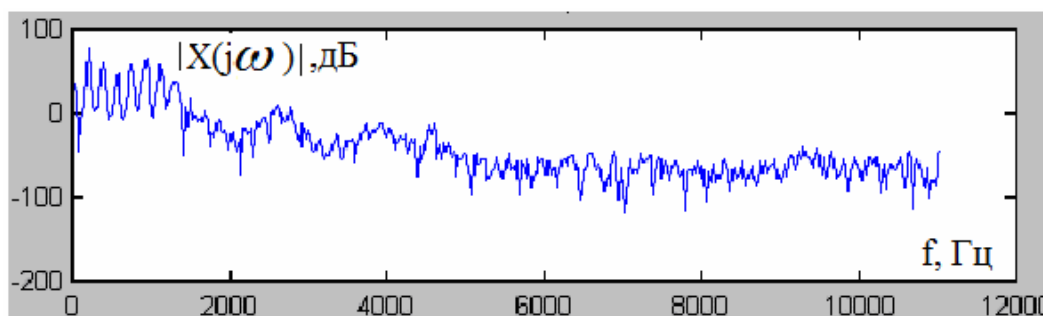


Рисунок 6.2— Узкополосный спектр

Заметим, что такое преобразование позволяет выявить каждую гармонику частоты основного тона в отдельности. На рис. 6.3 показаны результаты кратковременного анализа при $N=64$. В данном случае разрешающая способность по частоте получилась много меньшей, чем на рис 6.2.

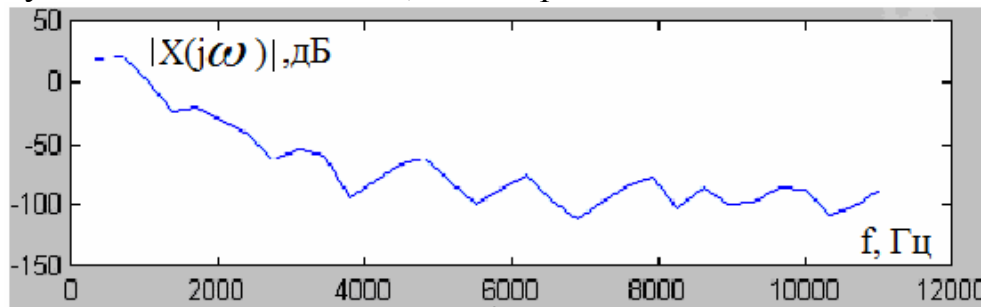


Рисунок 6.3 — Широкополосный спектр

Заметим, что спектр на рис. 6.2 весьма схож со спектром, представленным на рис. 6.3. На первом из них присутствует информация, как об основном тоне, так и о передаточной функции голосового тракта. Соответственно спектр на рис. 6.2 называют узкополосным, а на рис.6.3 — широкополосным.

При узкополосном кратковременном преобразовании Фурье функция возбуждения голосового тракта (см. рис. 6.1) проявляется узкими пиками на графике спектра на частотах, кратных основной частоте (см. рис.6.2). Этот факт положен в основу определения частоты основного тона путем вычисления произведения гармоник спектра [1]:

$$P_l(\omega) = \prod_{k=1}^K |X_l(\omega k)|^2 \quad (6.3)$$

Взяв логарифм, получим логарифмическое произведение гармоник спектра:

$$\bar{P}_l(\omega) = 2 \cdot \sum_{r=1}^R \log |X_l(\omega r)|.$$

Видно, что $\bar{P}_l(\omega)$ представляет собой сумму из R сжатых по частоте составляющих $\log |X_n(\omega r)|$. Введение функции $P_l(\omega)$ мотивируется тем, что в вокализованной речи сжатие частотной шкалы в целое число раз должно привести к усилению гармоник на основной частоте. Было обнаружено, что этот метод особенно устойчив к шумам, поскольку вклад шумов в $X_l(\omega)$ не имеет коррелированной структуры. По этой же причине невокализованная речь не даст пика в $\bar{P}_l(\omega)$.

При изучении речи одним из полезных инструментов является спектрограмма. Спектрограмма представляет энергию речевого сигнала в виде функции частоты и времени. Иными словами, спектрограмма это функция $|X_l(\omega)|$ в координатах частота-время-интенсивность.

Образец спектрограммы показан на рис. 6.4. В пакете MATLAB имеется специальная функция **specgram** для построения спектрограмм. Вызов функции выполняется в соответствии с форматом:

`specgram(X,NFFT,FS,WINDOW,NOVERLAP),`

где X — вектор отсчетов речевого сигнала; NFFT — число отсчетов БПФ; FS — частота дискретизации, Гц; WINDOW — имя и необходимые параметры функции окна; NOVERLAP — число отсчетов перекрытия спектров.

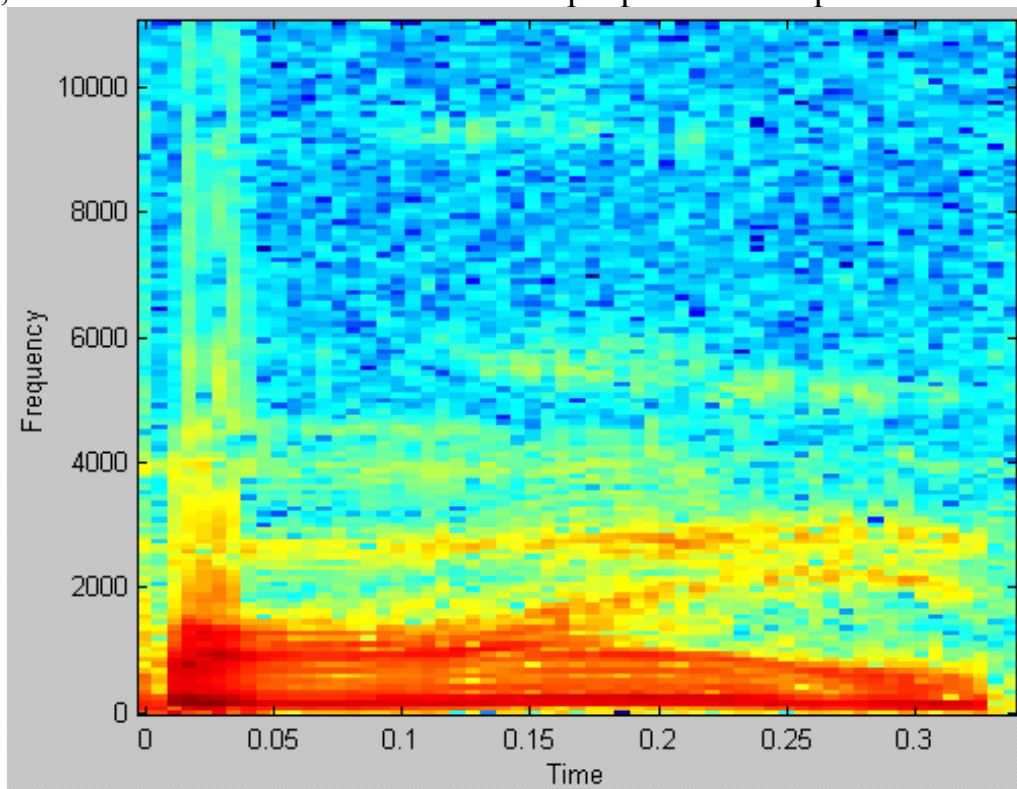


Рисунок 6.4 — Спектрограмма речи

В соответствии с рис.6.1 речевой сигнал $x[n]$ является сверткой функции возбуждения (случайного шума либо квазипериодической последовательности импульсов) и импульсной характеристики голосового тракта (цифрового фильтра). Гомоморфный анализ речи позволяет разделить эти компоненты. Схема гомоморфного анализа речи изображена на рис. 6.5.

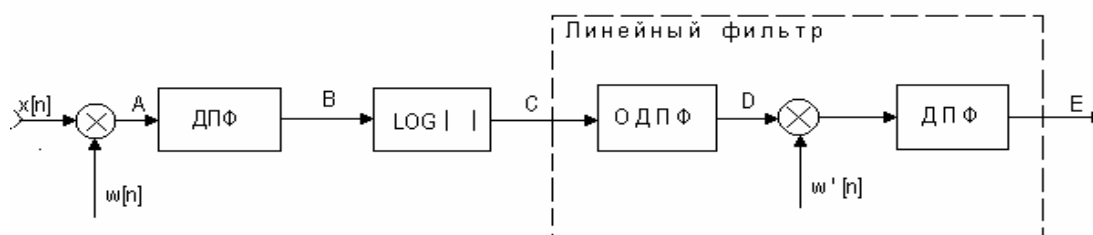


Рисунок 6.5 — Схема гомоморфного анализа речи

ДПФ от речевого сигнала $x[n]$ (точка В на рис. 6.5) будет представлять собой произведение Фурье преобразований функции возбуждения и импульсной характеристики цифрового фильтра. Логарифмируя найденное значение спектра, получим (точка С на рис. 6.5) сумму логарифмов спектра функции возбуждения и частотной характеристики голосового тракта. Поскольку обратное дискретное преобразование Фурье (ОДПФ) является линейной операцией, сигнал в точке D, называемый *кепстром*, равен результату сложения *кепстров* функции возбужде-

ния и голосового тракта. Результаты указанных преобразований применительно к вокализованной речи представлены на рис. 6.6.

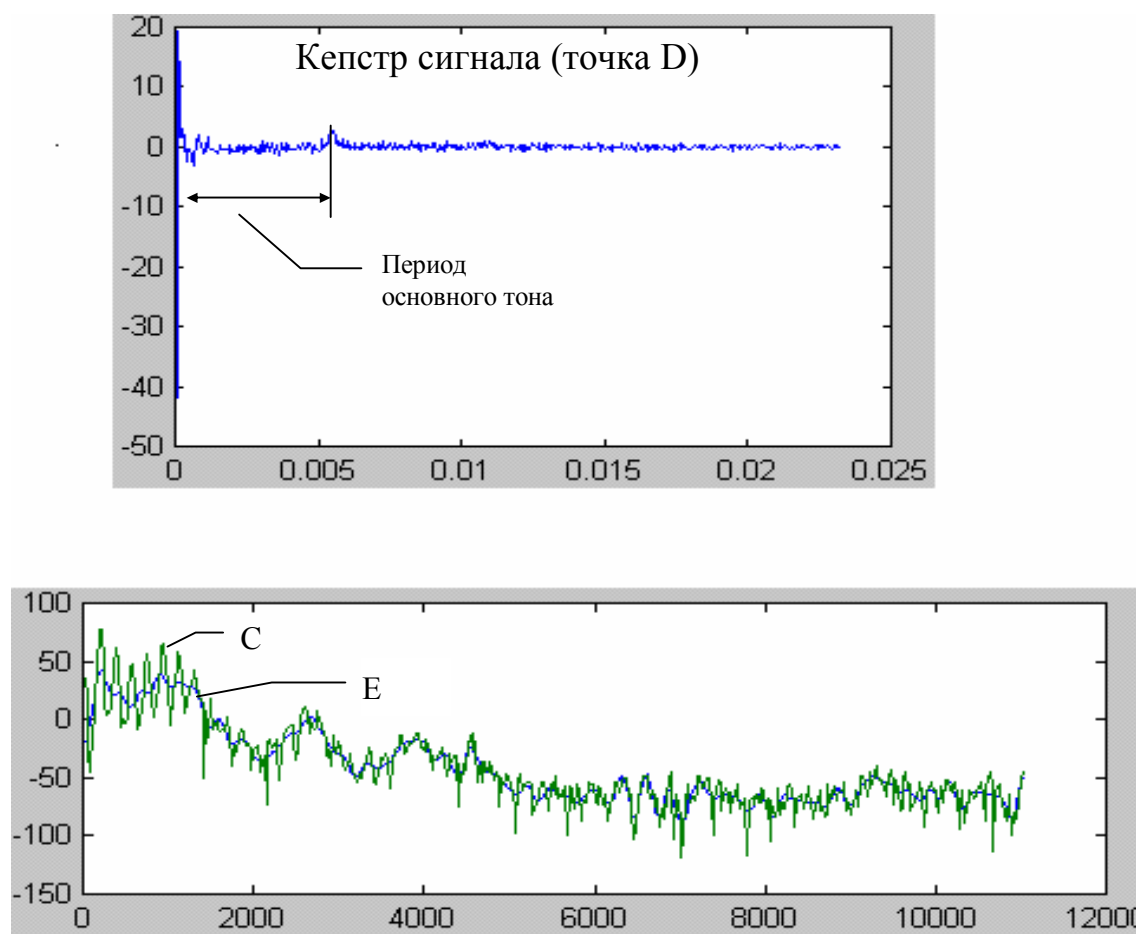


Рисунок 6.6 — Кепстр и спектр вокализованной речи

Пульсирующая кривая, помеченная буквой С на нижнем графике, представляет логарифм кратковременного спектра. Она содержит медленноменяющуюся составляющую, обусловленную передаточными свойствами голосового тракта и быстроменяющуюся периодическую составляющую, обусловленную периодическим сигналом возбуждения. Медленноменяющаяся составляющая способствует формированию в кепстре (точка D) области малых времен, а быстроменяющаяся составляющая проявляется в виде резкого пика в кепстре, возникающего через интервал времени, равный периоду входного сигнала. Важно отметить, что кепстр является аддитивной комбинацией, в которой составляющие голосового тракта и сигнала возбуждения практически не перекрываются. Таким образом, кепстр весьма удобен для определения периода основного тона вокализованной речи и для определения характера — вокализованный или невокализованный — данного образца речи.

Амплитудно-частотная характеристика голосового тракта, т.е. огибающая спектра, может быть получена посредством линейной фильтрации прологарифмированных значений спектра, в результате которой будут подавлены его быстроменяющиеся составляющие. Один из возможных способов такой фильтрации состоит в перемножении кепстральных значений на подходящее временное окно, пропускающее лишь составляющие из области малых времен с последующим вычис-

лением ДПФ кепстра. Кривая, полученная в результате такой обработки, помечена буквой Е (см. рис. 6.6).

6.3. Варианты заданий

Варианты заданий приведены в таблице 6.1

Таблица 6.1 — Варианты заданий

Номер варианта	Файл 1	Файл 2
1	ta0	ta50
2	ta1	ta40
3	ta2	ta30
4	ta3	ta20
5	ta4	ta10
6	ta5	ta19
7	ta6	ta18
8	ta7	ta17
9	ta8	ta16
10	ta9	ta15
11	ta10	ta14
12	ta11	ta13
13	ta12	ta20
14	ta13	ta11
15	ta14	ta10
16	ta15	ta9
17	ta16	ta8
18	ta17	ta7
19	ta18	ta6
20	ta19	ta5
21	ta20	ta3
22	ta30	ta2
23	ta40	ta1
24	ta50	ta0
25	ta0	ta5
26	ta1	ta10
27	ta2	ta20
28	ta3	ta30
29	ta4	ta40
30	ta5	ta50

6.4 Порядок выполнения лабораторной работы

6.4.1 Выполнить анализ речи с помощью кратковременного спектрального анализа:

- ввести данные из файлов соответствующих варианту задания, и сохранить их в векторах x_1 и x_2 (функция `waveread`);
- путем просмотра речевых сигналов выбрать вокализованные и невокализованные участки x_1 и x_2 ;

- выполнить кратковременный узкополосный и широкополосный анализ участков x_1 и x_2 , построить графики соответствующих реализаций и логарифмов модулей их спектров;
 - по результатам спектрального анализа определить частоту основного тона, пользуясь (6.3) ($R=5$). Построить график $P_1(\omega)$;
 - построить спектрограммы речевых процессов x_1 и x_2 .
- 6.4.2. Выполнить гомоморфный анализ речи:
- для фрагментов речи x_1 и x_2 выполнить обработку по схеме, представленной на рис. 6.5;
 - представить на графиках процессы в точках А, В, С, D, Е;
 - определить по кепстру частоту основного тона.

6.5. Содержание отчета

Цель работы, постановка задачи (вариант задания), краткие теоретические сведения с расчетными формулами, тексты программ обработки с комментариями, графики процессов x_1 и x_2 , узкополосный и широкополосный спектры вокализованных и невокализованных участков речи, график $P_1(\omega)$, частота основного тона, спектрограммы x_1 и x_2 , графики процессов в точках А,В,С,D,Е в соответствии со схемой гомоморфной обработки, значения частоты основного тона, определенное по кепстру, выводы по работе.

6.6. Контрольные вопросы

- 6.6.1. Как формируются вокализованные звуки?
- 6.6.2. Как формируются фрикативные звуки?
- 6.6.3. Опишите цифровую модель формирования речи?
- 6.6.4. Объясните суть кратковременного спектрального анализа?
- 6.6.5. Что такое узкополосный и широкополосный спектральный анализ?
- 6.6.6. Как по кратковременному спектру определить частоту основного тона?
- 6.6.7. Основные этапы гомоморфной обработки речи?
- 6.6.8. Что такое кепстр, и как определить частоту основного тона по кепстру?
- 6.6.9. Объясните на графиках, что понимают под быстроменяющейся и медленноменяющейся составляющими спектра речи?
- 6.6.10. Объясните на графиках, что понимают под областью малых и больших времен кепстра?
- 6.6.11. Чему на графике спектра соответствует область малых (больших) времен кепстра?
- 6.6.12. Каким должно быть временное окно, что сохранить область малых времен кепстра?

7. ЛАБОРАТОРНАЯ РАБОТА №7 «ЛИНЕЙНЫЙ ПРЕДИКТИВНЫЙ АНАЛИЗ РЕЧИ»

7.1. Цель работы

Исследование метода линейного предсказания речи. Определение коэффициентов линейного предсказания.

7.2. Краткие теоретические сведения

Способ анализа речи на основе линейного предсказания базируется на использовании цифровой модели речи, изображенной на рис. 6.1. Предположим, что цифровой фильтр на рис. 6.1 имеет передаточную функцию вида

$$H(z) = \frac{G}{1 + \sum_{k=1}^p a[k] z^{-k}}. \quad (7.1)$$

В случае вокализованной речи фильтр возбуждается последовательностью импульсов, а на участках невокализованной речи — белым шумом. В соответствии с (7.1) речевые отсчёты $x[n]$ связаны с сигналом возбуждения уравнением

$$x[n] = -\sum_{k=1}^p a[k] x[n-k] + u[n]. \quad (7.2)$$

Отметим, что здесь значения сигнала $u[n]$ неизвестны.

Если предположить, что известны предыдущие значения речевого процесса $x[n-1]$, $x[n-2]$ и т.д., то текущее значение $x[n]$ можно предсказать по предыдущим значениям, используя формулу

$$\tilde{x}[n] = -\sum_{k=1}^p \hat{a}[k] x[n-k]. \quad (7.3)$$

Коэффициенты $\hat{a}[k]$ называют коэффициентами линейного предсказания (ЛПК-коэффициентами). Ошибка предсказания будет равна

$$e[n] = x[n] - \tilde{x}[n] = x[n] + \sum_{k=1}^p \hat{a}[k] x[n-k]. \quad (7.4)$$

Один из способов определения ЛПК-коэффициентов основан на минимизации среднеквадратического значения ошибки (7.4):

$$E_e = \sum_{n=0}^{N-1} e^2[n] = \sum_{n=0}^{N-1} \left[x_e[n] + \sum_{k=1}^p \hat{a}[k] x_e[n-k] \right]^2, \quad (7.5)$$

где $x_e[n]$ — стационарные участки речи. Для нахождения $\hat{a}[k]$ необходимо решить систему уравнений

$$\frac{\partial E}{\partial \hat{a}[k]} = 0, \quad 1 \leq k \leq p \quad (7.6)$$

Выполнив дифференцирование, получим систему нормальных уравнений:

$$\sum_{n=0}^{N-1} x_e[n-i] x_e[n] = \sum_{k=1}^p \hat{a}[k] \sum_{n=0}^{N-1} x_e[n-k] x_e[n-i], \quad 1 \leq i \leq p. \quad (7.7)$$

Вводя обозначение

$$r[i] = \sum_{n=0}^{N-1} x_e[n]x_e[n+i], \quad (7.8)$$

перепишем систему (7.7) компактной форме:

$$\sum_{k=1}^p r[i-k] \hat{a}[k] = r[i] \quad , 1 \leq i \leq p, \quad (7.9)$$

где $r[i]$ — автокорреляционная функция процесса $x_e[n]$. Участки речи $x_e[n]$ получаются путём умножения $x[n]$ на временное окно $w[n]$ длиной в N отсчётов.

Существует несколько процедур решения уравнения (7.9). Одна из процедур нахождения ЛПК коэффициентов использует рекурсивный алгоритм Левинсона-Дарбина [1]. ЛПК-коэффициенты широко применяются при решении следующих задач анализа речи:

1. Оценка спектра. Если известны коэффициенты ЛПК, то нетрудно определить амплитудно-частотную характеристику речевого тракта:

$$H(e^{j\omega T_0}) = \frac{G}{1 + \sum_{k=1}^P \hat{a}[k] e^{-jk\omega T_0}}. \quad (7.10)$$

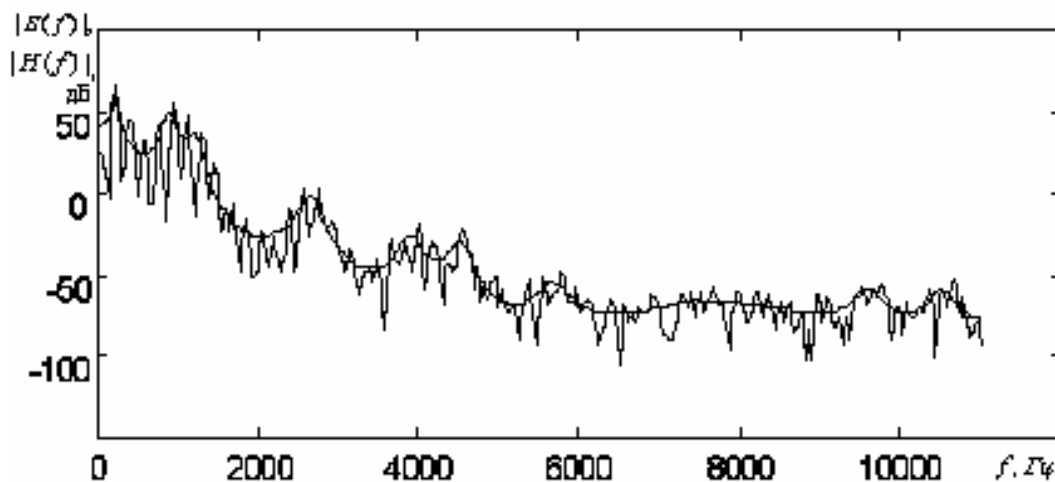


Рисунок 7.1 — Кратковременный спектр и АЧХ голосового тракта

Данная характеристика соответствует медленно меняющейся составляющей кратковременного спектра речевого сигнала. На рис. 7.1 представлен кратковременный спектр речевого сигнала и АЧХ голосового тракта, вычисленная с помощью (7.10). Порядок фильтра $P=28$.

2. Оценка формантных частот. Пики на графике АЧХ соответствуют так называемым формантным частотам. Эти пики связаны с полюсами передаточной функции (7.1).

3. Определение основного тона. Отметим, что ошибка предсказания (7.4) будет похожа на функцию возбуждения $u[n]$. Поэтому по $e[n]$ можно определить период основного тона.

4. Синтез речи. Коэффициенты предсказания и информация о функции возбуждения могут быть использованы для восстановления речи по схеме, изображенной на рис.6.1.

7.3. Варианты заданий

Варианты заданий выбираются в соответствии с табл. 6.1.

7.4. Порядок выполнения лабораторной работы

7.4.1. Выполнить кратковременный узкополосный и широкополосный анализ вокализованного участка речи.

7.4.2. Выполнить гомоморфный спектральный анализ этого же участка речи.

7.4.3. Определить ЛПК-коэффициенты для выбранного участка речи. Для этого можно использовать функцию `lpc` пакета Matlab.

7.4.4. Построить оценку спектра, используя (7.10).

7.4.5. Выполнить сравнение указанных четырёх спектров.

7.4.6. Вычислить ошибку $e[n]$, используя (7.4).

7.4.7. Определить частоту основного тона по графику $e[n]$.

7.4.8. По спектру, полученному методом линейного предсказания, определить формантные частоты.

7.5. Содержание отчёта

Цель работы, краткие теоретические сведения, тексты программ обработки с комментариями, графики вокализованных участков речи, графики узкополосного и широкополосного спектров, график спектра, полученного путём гомоморфной обработки, значения ЛПК-коэффициентов, график спектра, полученного методом линейного предсказания, график ошибки предсказания, значение частоты основного тона, выводы.

7.6. Контрольные вопросы

7.6.1. Запишите уравнение предсказания речи.

7.6.2. Как определяется ошибка предсказания?

7.6.3. Выполните вывод системы нормальных уравнений.

7.6.4. Как перейти от системы нормальных уравнений к системе с использованием автокорреляционных функций?

7.6.5. Каким методом решается система автокорреляционных уравнений?

7.6.6. Как получить оценку спектра на основе ЛПК-коэффициентов?

7.6.7. Как по графику спектра, полученному на основе ЛПК-коэффициентов, определить формантные частоты?

7.6.8. Какому процессу на рис. 6.1 соответствует ошибка предсказания?

7.6.9. Как по графику ошибки предсказания определить период основного тона?

7.6.10. Приведите примеры применения линейного предсказания.

8. ЛАБОРАТОРНАЯ РАБОТА № 8 «ЦИФРОВАЯ ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ»

8.1. Цель работы

Исследование способов улучшения изображений с помощью цифровой фильтрации. Исследование способов представления изображений в ЭВМ и алгоритмов двумерной обработки сигналов.

8.2. Краткие теоретические сведения

8.2.1. Представление изображений в системе MATLAB

MATLAB поддерживает четыре формата представления изображений [2,3]:

- индексированные изображения;
- полутоновые изображения;
- бинарные изображения;
- цветные(RGB) изображения.

Основным типом данных, используемым для представления изображений, является матрица изображений, которая соответствует двумерной последовательности чисел $x[n_1, n_2]$. Элемент последовательности $x[n_1, n_2]$ соответствует пикселю на изображении.

Индексированные изображения в MATLAB используют две матрицы: матрицу цветов и матрицу изображения. Матрица цветов является упорядоченным множеством значений, которые представляют цвета изображения. Каждому пикселю матрицы изображений соответствует определенный цвет, задаваемый индексом матрицы цветов. Матрица изображений состоит из целых чисел от 1 до 128, которые являются индексами матрицы цветов. Каждая строка матрицы цветов, выбираемая с помощью индекса, представляет вектор $[R \ G \ B]$, где R, G, B – вещественные числа, задающие интенсивность соответственно красного, зеленого и синего цветов. Поэтому матрицу цветов часто называют палитрой. Одно и тоже изображение можно отображать, используя различные палитры. Для отображения матрицы изображения x с помощью палитры **pal** в MATLAB можно применить функцию **imshow(x, pal)**. При этом необходимо предварительно задать значение матриц x и **pal**. Если второй параметр функции **imshow** не задан, то используется палитра по умолчанию.

Полутоновые изображения в MATLAB представляются матрицей интенсивностей. Каждый элемент такой матрицы является вещественным значением в диапазоне от 0.0 до 1.0. Элемент матрицы интенсивностей соответствует пикселю, отображаемому градациями серого цвета. Нулевому значению интенсивности соответствует черный цвет, а единичному — белый. Полутоновый формат изображений идеально подходит для изображения черно-белых фотографий, медицинских снимков. Для отображения матрицы интенсивности, которую так же можно рассматривать как двумерную последовательность вещественных чисел $x[n_1, n_2]$, в MATLAB используют функцию **imshow**. Например, **imshow(x, 64)**. Здесь второй параметр функции **imshow** обозначает количество градаций серого цвета.

Во многих случаях полутоновые изображения, полученные с помощью других систем, представляются как индексированные изображения с палитрой серого цвета, имеющего 256 градаций. Для преобразования таких изображений в полутоновые можно использовать функцию **ind2gray(x+1, gray(256))**. Здесь значение индексированной матрицы изображений увеличивается на 1, потому что начальный индекс матрицы цветов равен 1, а не 0.

Бинарные изображения представляют частный случай полутоновых изображений, когда используются всего две градации серого цвета. Матрица интенсивностей для бинарного изображения имеет всего два уровня значений 0(черный) и 1(белый). Такая матрица интенсивностей иногда называется логической или булевой. Она может быть создана с помощью операций отношений ($>$, $<$, $=$). Для отображения бинарных изображений в MATLAB используют функцию **imshow(x,2)**.

Формат цветных RGB-изображений является стандартной формой для представления цветных изображений. Например, большинство сканеров поддерживают RGB-формат. В этом случае для каждого цвета (красного, зеленого, синего) создается своя матрица интенсивностей. Для отображения цвета пиксела на экране компьютера выполняется выборка значений из трех указанных матриц интенсивностей. Для отображения цветных RGB-изображений необходимо сначала создать три матрицы интенсивностей r, g, b , а затем применить функцию **imshow(r,g,b)**.

В некоторых случаях необходимо выполнять преобразование индексированных изображений в цветные RGB-изображения. Данная операция может быть выполнена с помощью функции **[r,g,b] = ind2rgb(x,pal)**.

Данная операция позволяет в дальнейшем обрабатывать отдельно каждую из матриц интенсивностей r, g или b (например, выполнять фильтрацию). В противном случае операция обработки будет применена не к интенсивностям, а к индексам.

В некоторых системах RGB-изображения хранятся с использованием четырех байтовых целых чисел. В этом случае интенсивности красного, синего и зеленого цветов имеют по 256 градаций и хранятся, соответственно, в первых трех байтах. Четвертый байт заполнен нулями. Для того чтобы обрабатывать такие изображения в MATLAB, необходимо выделить байты, соответствующие красному, синему и зеленому цветам, и сформировать соответствующие им матрицы интенсивности.

Пакет MATLAB имеет ряд функций для чтения распространенных форматов изображения: GIF, TIFF, BMP, PCX.

Примеры вызовов функций, выполняющих ввод изображений в указанных форматах:

```
[x, pal] = gifread('img.gif');
[r, g, b] = tiffread('rgb.tif');
[x, pal] = bmpread('img.bmp');
[x, pal] = pcxread('img.pcx').
```

При обработке изображений в MATLAB в большинстве случаев используется система координат, соответствующая индексированию матрицы. Например, обращение $x(2,3)$ будет соответствовать отображению третьего пикселя второй строки. Отсчет номеров пикселей начинается с верхнего левого угла дисплея.

8.2.2 Цифровая фильтрация изображений

Для цифровой фильтрации изображений наиболее часто используют фильтры с конечной импульсной характеристикой (КИХ). Такие фильтры обладают следующими положительными свойствами [1, 4]:

- импульсная характеристика представляется в виде матрицы;
- двумерные КИХ-фильтры являются естественным расширением одномерных КИХ-фильтров;
- имеются несколько хороших методов синтеза КИХ-фильтров;
- КИХ-фильтры просты в применении;
- легко выполнить проектирование КИХ-фильтра с нулевым фазовым сдвигом.

Последнее свойство очень важно для обработки изображений. Нулевой фазовый сдвиг предотвращает искажение изображения. КИХ-фильтр будет иметь нулевой фазовый сдвиг, если его частотная характеристика является действительной функцией, т. е. $H(\omega_1, \omega_2) = H^*(\omega_1, \omega_2)$, или импульсная характеристика будет симметричной относительно начала координат $h[n_1, n_2] = h^*[-n_1, -n_2]$.

Для фильтров, имеющих нечетное число вещественных коэффициентов в каждом направлении, частотная характеристика будет вещественной. Для фильтров, имеющих четное число коэффициентов, частотная характеристика будет иметь линейную зависимость фазы от частоты.

Фильтры с бесконечной импульсной характеристикой (БИХ) для фильтрации изображений практически не применяются. Они требуют дополнительного обеспечения устойчивости и не обеспечивают линейности фазовой характеристики.

Фильтрация изображений выполняется с помощью линейной двумерной свертки

$$y[n_1, n_2] = \sum_{k_1} \sum_{k_2} x[k_1, k_2] * h[n_1 - k_1, n_2 - k_2],$$

где $x[n_1, n_2]$ — матрица интенсивностей; $h[n_1, n_2]$ — двумерная импульсная переходная характеристика фильтра.

Пусть

$$h = \begin{bmatrix} 2 & 6 & 4 \\ 1 & -3 & 4 \end{bmatrix}.$$

Тогда свертка $y[n_1, n_2] = x[n_1, n_2] * h[n_1, n_2]$ будет соответствовать вычислениям по схеме:

$$y[n_1, n_2] = 2x[n_1-1, n_2-1] + 6x[n_1-1, n_2] + 4x[n_1-1, n_2+1] + \\ + x[n_1, n_2-1] - 3x[n_1, n_2] + 4x[n_1, n_2+1].$$

В соответствии с этой схемой вычисляется сумма произведений отсчетов последовательности $h[n_1, n_2]$ на значения отсчетов изображения $x[n_1, n_2]$, которые находятся в пределах «окна», соответствующего импульсной характеристике (рис.8.1).

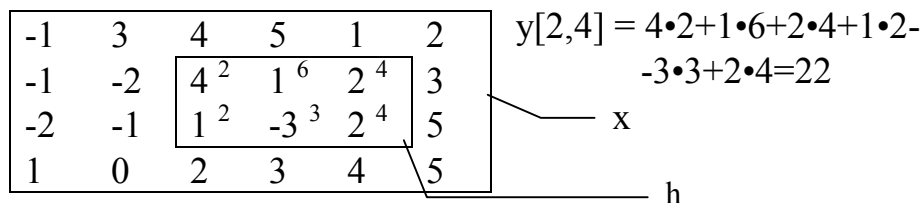


Рисунок 8.1— Вычисление двумерной свертки

Вычисленное значение $y[n1,n2]$ связано с центром «окна». Положение центра окна зависит от его формы:

- для окна с нечетным количеством строк и столбцов центр соответствует центральному элементу окна;
- для окна с четным числом строк или столбцов центр состоит из четырех или двух элементов и соответственно результат связывают с позицией верхнего левого элемента (или только левого, или только верхнего).

Для фильтрации изображений в MATLAB используют функцию `filter2(x,h)`.

8.2.3. Проектирование двумерных КИХ-фильтров

Существует несколько методов синтеза двумерных КИХ-фильтров [1]:

- метод частотной выборки, который позволяет синтезировать фильтр на основе заданной частотной характеристики;
- метод оконных функций, который позволяет синтезировать фильтр на основе заданной частотной характеристики, взвешенной оконной функцией;
- метод частотного преобразования, который преобразует одномерный КИХ-фильтр в двумерный КИХ-фильтр.

При синтезе КИХ-фильтра методом частотной выборки задается желаемая частотная характеристика. Эта характеристика представляется матрицей, числовые значения которой соответствуют модулю частотной характеристики в заданных точках двумерной плоскости (ω_1, ω_2) . Метод позволяет синтезировать КИХ-фильтр с АЧХ, проходящий через заданные точки. Поведения АЧХ между двумя заданными точками не контролируется. Для синтеза КИХ-фильтров методом частотной выборки в пакете MATLAB имеется функция `fsamp2`. Формат вызова функции

$$h = \text{fsamp2}(H_d),$$

где H_d — желаемая АЧХ фильтра, представленная матрицей; h — результирующая матрица, соответствующая импульсной характеристике фильтра.

Ниже приведен пример синтеза двумерного фильтра 11-го порядка и отображения его частотного отклика:

```
Hd = zeros(11,11);           % создаем нулевую матрицу 11*11
Hd(4:8,4:8) = ones(5,5);    % задаем желаемую АЧХ
[f1,f2] = fregspace([11,11]); % формируем нормированные частоты
[x,y] = meshgrid(f1,f2);     % формируем сетку на опорной области
colormap(jet(64));           % выбираем палитру
subplot(1,2,1), mesh(x,y,Hd), % отображаем желаемую АЧХ
axis([-11 -11 0 1.2]);
```

```
h=fsam2(Hd); % синтезируем КИХ-фильтр
subplot(1,2,2), freqz2(h, [32 32]), axis([-11 -11 0 1.2])
```

КИХ-фильтр, синтезированный указанным способом, имеет большие пульсации в полосе пропускания.

Лучшие результаты дает метод оконных функций. В этом случае желаемая АЧХ фильтра умножается на весовое окно, а затем выполняется синтез фильтра. В MATLAB имеются две функции, используемые для синтеза двумерных фильтров `fwind1` и `fwind2`. Функция `fwind1` создает двумерное окно путем трансформации одномерного окна. Функция `fwind2` формирует двумерное окно путем непосредственных вычислений. Пример вызова функции `fwind1`:

```
h= fwind1(Hd, hamming(11));
```

Метод частотного преобразования позволяет преобразовывать одномерный КИХ-фильтр в двумерный. Этот метод обеспечивает сохранение свойств одномерного фильтра. В частности, сохраняется ширина переходных полос и заданные уровни пульсаций. В MATLAB методу частотного преобразования соответствует функция `ftrans2`. До вызова функции `ftrans2` необходимо синтезировать одномерный КИХ-фильтр одним из известных методов. Например:

```
b=remez(10,[0 0.45 0.65 1],[1 1 0.05 0.05]);
h=ftrans2(b);
```

Здесь предварительно синтезируется низкочастотный одномерный КИХ-фильтр 10-го порядка по методу Ремеза, а затем он преобразуется в двумерный фильтр.

8.2.4. Улучшение изображений с помощью фильтрации

Линейная фильтрация наиболее часто используется для подавления ограниченного по частоте аддитивного шума. При этом для каждого случая необходимо синтезировать соответствующий фильтр, учитывая частотное распределение шумов [1,2].

Сглаживание шума обеспечивается низкочастотной фильтрацией. Существенным недостатком такой обработки является размывание границ. Для подчеркивания границ можно применить фильтр верхних частот. Другой способ подавления шумов на изображении основан на применении медианной фильтрации. Суть этого метода фильтрации состоит в упорядочении значений матрицы интенсивностей, попавших в окно весовой функции и в замене значений опорной точки значением соответствующим срединному значению упорядоченного ряда чисел. Медианная фильтрация является нелинейным методом обработки. Медианная фильтрация весьма эффективна для обработки изображений искаженными помехами типа «соль и перец», т.е., когда на светлом фоне имеются темные точки («перец») и на темном фоне светлые точки («соль»). В пакете MATLAB медианная фильтрация реализуется функцией `medfilt2`.

8.3. Варианты заданий

Варианты задания приведены в таблице 8.1.

Таблица 8.1— Варианты заданий

№ варианта	NxN	Метод синтеза	Изображение
1	5x5	Fsamp2	Saturn tif
2	6x6	Fwind1	Bonemarr tif
3	7x7	Ftrans2	Moon tif
4	8x8	Fsamp2	Rice tif
5	9x9	Fwind1	Moon tif
6	10x10	Ftrans2	Saturn tif
7	11x11	Fsamp2	Bonemarr tif
8	5x5	Fwind1	Moon tif
9	6x6	Ftrans2	Rice tif
10	7x7	Fsamp2	Moon tif
11	8x8	Fwind1	Saturn tif
12	9x9	Ftrans2	Bonemarr tif
13	10x10	Fsamp2	Moon tif
14	11x11	Fwind1	Rice tif
15	5x5	Ftrans2	Moon tif
16	6x6	Fsamp2	Saturn tif
17	7x7	Fwind1	Bonemarr tif
18	8x8	Ftrans2	Moon tif
19	9x9	Fsamp2	Rice tif
20	9x9	Ftrans2	Bonemarr tif
21	7x7	Fwind1	Bonemarr tif
22	10x10	Fwind1	Moon tif
23	10x10	Fsamp2	Moon tif
24	5x5	Ftrans2	Moon tif
25	8x8	Ftrans2	Moon tif
26	10x10	Fwind1	Moon tif
27	11x11	Fwind1	Rice tif
28	9x9	Fsamp2	Rice tif
29	8x8	Fwind1	Saturn tif
30	6x6	Fsamp2	Saturn tif

8.4. Порядок выполнения лабораторной работы

8.4.1. Загрузить в память ЭВМ заданное изображение, используя функцию `imread`. Формат вызова функции `a=imread('имя файла', 'формат')`, где 'формат' — строка, соответствующая формату хранения изображения 'tiff', 'bmp', 'psx' и др.

8.4.2. Если необходимо, то преобразовать изображение в полутоновое, используя функцию `rgb2gray(R,G,B)`.

8.4.3. Отобразить на экране исходное изображение.

8.4.4. Добавить к исходному изображению гауссовский шум с помощью функции `imnoise`. Формат вызова функции

`I=imnoise(a,'gaussian',M,V),`

где `a` — исходное изображение, заданное матрицей интенсивностей; `M` — среднее значение шума; `V` — дисперсия. Если параметры `M` и `V` не заданы, то считается, что `M=0`, а `V=0.1`.

8.4.5. Синтезировать фильтр в соответствии с вариантом задания. Для синтеза фильтра необходимо использовать два параметра — порядок и частоту среза.

Если по варианту для синтеза фильтра необходимо использовать функции `fsamp2` или `fwind1`, то желаемую АЧХ можно сформировать следующим образом:

```
order = 11;
cutoff = 0.5;
[f1,f2] = freqspace(order);
[x,y] = meshgrid(f1,f2);
Hd = zeros(size(x));
d = finde(sqrt(x.*x+y.*y)<cutoff);
Hd(d) = ones(size(d));
```

Здесь `order` — порядок фильтра; `cutoff` — нормированная частота среза фильтра; `d` — определяет индексы элементов матрицы; `Hd` — желаемая АЧХ, которая будет иметь единичные значения. При этом `d` соответствует круговой области на плоскости частот. Радиус этой области меньше значения частоты среза.

Приведенный фрагмент программы формирует желаемую АЧХ низкочастотного фильтра. Если требуется построить ФВЧ, то выполняют оператор

`Hd = 1-Hd`

и АЧХ становится инверсной.

Если в соответствии с вариантом, необходимо синтезировать фильтр, используя функцию `ftrans2`, то можно воспользоваться следующей последовательностью операторов:

```
F = [0 cutoff-0.5 cutoff+0.5 1];           % ось частот
M = [1 1 0 0];                             % значение АЧХ в заданных точках
H = ftrans2(remez(order-1,F,M)); % синтез фильтра
```

Построить импульсную и амплитудно-частотную характеристики желаемого и синтезированного фильтра.

8.4.6. Выполнить фильтрацию исходного изображения, используя функцию `filter2`. Подобрать частоту среза фильтра так, чтобы получить наилучшее качество изображения.

8.4.7. Отобразить полученное изображение. Если после фильтрации элементы матрицы интенсивностей имеют тип `double`, то необходимо привести их к типу `uint8`. Операция приведения типов в пакете MATLAB выполняется оператором `uint8(x)`.

8.4.8. Добавить к исходному изображению шум типа «соль и перец». Отобразить полученное изображение.

8.4.9. Обработать изображение, полученное в пункте 8.4.1 синтезированным фильтром и с помощью медианного фильтра, сравнить результаты.

8.5. Содержание отчета

Цель работы, краткие теоретические сведения, текст программы с комментариями, АЧХ желаемого и синтезированного фильтров, импульсные характеристики желаемого и синтезированного фильтров, алгоритмы фильтрации, результаты фильтрации, выводы.

8.6. Контрольные вопросы

- 8.6.1. Какие существуют способы представления изображений в MATLAB?
- 8.6.2. Объясните индексное представление изображений в ЭВМ.
- 8.6.3. Объясните полутоновое представление изображений в ЭВМ.
- 8.6.4. Объясните бинарное представление изображений в ЭВМ.
- 8.6.5. Объясните RGB-представление изображений в ЭВМ.
- 8.6.6. Запишите уравнение двумерной свертки и объясните его.
- 8.6.7. Запишите разностное уравнение двумерного фильтра и найдите его Z-преобразование.
- 8.6.8. Достоинства КИХ-фильтров?
- 8.6.9. Какие существуют методы синтеза двумерных фильтров? В чем их особенности?
- 8.6.10. Объясните алгоритм медианной фильтрации?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бондарев В.Н. Цифровая обработка сигналов: методы и средства. Учеб. пособие для вузов / В.Н. Бондарев, Г.Трестер, В.С. Чернега. — 2-е изд. — Х.: Конус, 2001. — 398с.
2. Гонсалес Р. Цифровая обработка изображений в среде MATLAB: пер. с англ. / Р. Гонсалес, Р. Вудс, С. Эддинс. — М.: Техносфера, 2006. — 616 с.
3. Поршнев С. В. MATLAB 7: основы работы и программирования: учеб. пособие для студ. вузов, обуч. по напр. 654600 "Информатика и вычислительная техника" / С. В. Поршнев. — М.: БИНОМ, 2011. — 320 с.
4. Сергиенко А. Б. Цифровая обработка сигналов: учеб. пособие для студ. вузов / А. Б. Сергиенко. — 2-е изд. — М. и др. : Питер, 2006. — 751 с.
5. Baudin Michael Введение в Scilab [электронный ресурс] / перевод А. Глебов, 2013 . — 89 с. — Режим доступа:
<http://forge.scilab.org/upload/docintrotoscilab/files/introscilab-v1.3-ru.pdf> . — Последний доступ 10.10.2013г. — Название с экрана.