



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Кафедра «Математики и информатики»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
для проведения практических занятий
по дисциплине
«Методы обработки сигналов и изображений»**

Ростов-на-Дону
ДГТУ
2022

УДК 004.8(075.8)

Составитель: М.Н. Богачева

Методические указания для проведения практических занятий по дисциплине «Методы обработки сигналов и изображений» – Ростов-на-Дону : Донской гос. техн. ун-т, 2022. – 82 с.

Методические указания предназначены для обучающихся по направлению подготовки магистратуры направления 090402 «Информационные системы и технологии».

УДК 004.8(075.8)

Печатается по решению редакционно-издательского совета
Донского государственного технического университета

Ответственный за выпуск зав. кафедрой «Математика и информатика» д-р физ.-мат. наук, профессор А.И. Сухинов

В печать _____ г.
Формат 60×84/16. Объем _____ усл. п. л.
Тираж _____ экз. Заказ № _____

Издательский центр ДГТУ
Адрес университета и полиграфического предприятия:
344000, г. Ростов-на-Дону, пл. Гагарина, 1

© Донской государственный
технический университет, 2022

Оглавление

| | |
|--|----|
| Лабораторная работа № 1 | |
| Разложение дискретизированных сигналов в действительный и комплексный ряд Фурье. Цифровая фильтрация шумов..... | 4 |
| Лабораторная работа № 2 | |
| Кластеризация данных алгоритмом FOREL. Нахождения дискриминантной функции по прецедентам методом потенциальных функций. Построение гистограмм функций плотности распределения..... | 36 |
| Лабораторная работа № 3 | |
| Подгонка кривых | 50 |
| Лабораторная работа № 4 | |
| Частотно-временной анализ нестационарных сигналов методом Гильберта-Хуанга | 55 |
| Лабораторная работа № | 4 |
| Средства идентификации объектов на изображениях Базовые средства фильтрации шумов на изображениях, | 62 |
| Лабораторная работа № 6 | 76 |
| Построение Байесовского классификатора по выборке двумерных нормально распределенных векторов. Построение Байесовского классификатора по прецедентам | 77 |

Лабораторная работа № 1

Разложение дискретизированных сигналов в действительный и комплексный ряд Фурье. Цифровая фильтрация шумов

Цель работы

изучение методики разработки программ разложения дискретизированных сигналов в действительный и комплексный ряд Фурье на интервалах $[-\pi, \pi]$ и $[-T/2, T/2]$ с конечным числом членов. Изучение методики разработки программ цифровой обработки сигналов, включающей различные способы улучшения отношения сигнал/шум (накопление, использование НЧ и ВЧ-фильтров, оптимального фильтра Колмогорова-Винера, прямого и обратного БПФ).

Теоритические сведения

Программные средства систем ЦОС обычно создаются на языке C/C++, так как в этом языке предусмотрены средства, обычно необходимые при аппаратной реализации на базе программируемых логических интегральных схем (ПЛИС) и сигнальных процессоров: удобные средства работы с битами, логические операции, средства работы с аппаратными прерываниями и др. Кроме того, компилятор C обеспечивает формирование исполняемой программы, близкой по скорости исполнения к программе на ассемблере.

При разработке систем жесткого реального времени на базе ПЛИС и быстродействующих сигнальных процессоров необходимо производить отработку алгоритмов ЦОС на универсальных средствах (с помощью компьютерных программ) используя низкоуровневое программирование без библиотечных функций (на C/C++) с тем, чтобы на следующем этапе можно было реализовать разработанные алгоритмы ЦОС на ПЛИС и сигнальных процессорах. Кроме того, как известно, ЦОС должна быть выполнена в течение критического срока обслуживания и это так же важно, как и корректность алгоритма ЦОС. Невыполнение задачи реального времени в течение критического срока обслуживания равносильно невыполнению задачи в целом. Поэтому очень важно, чтобы на втором этапе разработки отработка алгоритмов ЦОС производилась с использованием соответствующей аппаратуры и скорость выполнения операций ЦОС контролировалась, например, с использованием системного таймера компьютера.

Конечной целью разработки и исследования различных алгоритмов ЦОС является их практическая реализация в устройствах на базе ПЛИС и сигнальных процессоров, таких, например, как модуль ЦОС SAMC-401, содержащий ПЛИС серии Virtex-4, в которой интегрированы два процессора

Power PC с тактовой частотой до 450 МГц, и сигнальный процессор Texas Instruments TMS320C6455, работающий на частоте 1.2ГГц с submodule АЦП SAMC-ADC, реализующий аналого-цифровое преобразование 12/14 бит с предельной частотой тактирования до 210 МГц.

Основные стадии разработки алгоритмов ЦОС:

1. Высокоуровневая программная (MATLAB, LabView).
2. Низкоуровневая программная (C/C++).
3. Аппаратно-программная (C/C++, LabWindows/CVI).
4. Аппаратная (VHDL, C/C++).

На первом этапе разработки наиболее совершенным средством является MATLAB, так как он содержит библиотеки функций для сложных видов математической обработки, таких как быстрое преобразование Фурье, цифровая фильтрация, корреляционная обработка, преобразование Гильберта и др., библиотеки программ для создания объектов графического пользовательского интерфейса (панелей, кнопок управления, окон цифрового ввода/вывода, окон графического вывода и др.).

Программные средства, создаваемые для работы в среде MATLAB, содержат две составляющие:

1. Собственно программу (mat-файл) на языке C¹, в которой содержатся необходимые функции математической обработки, отображения таблиц и графиков. При ее создании следует пользоваться описаниями функций библиотек математической обработки и графического пользовательского интерфейса.
2. Файлы ресурсов графического пользовательского интерфейса <имя программы>.fig.

Создаваемая пользователем прикладная программа представляет проект, содержащий два файла:

- файл основной программы <имя>.mat;
- файл макета <имя>.fig.

Программа на языке C может быть написана и редактироваться пользователем. Файлы макета <имя>.fig создаются автоматически при создании и редактировании пользователем графических панелей. Эти файлы нельзя редактировать!

¹ Синтаксис языка C в MATLAB имеет ряд особенностей, в частности, связанных с тем, что объектами при вычислениях являются матрицы.

Создание средств графического пользовательского интерфейса в традиционных системах программирования, таких как Visual C++ возможно, но достаточно сложно, так как для этого необходимо создание большого количества нестандартных графических объектов.

Задание к работе

Имеется дискретизированная функция в виде числового массива. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки данных, реализующую разложение этого сигнала в ряд Фурье с конечным числом членов и исследовать зависимость точности представления этой функции с помощью ряда Фурье от числа членов разложения и от шага дискретизации исходного непрерывного сигнала.

Теоретические основы

Известно, что любую функцию $f(t)$ можно представить в виде:

$$f(t) = c_0\varphi_0(t) + c_1\varphi_1(t) + c_2\varphi_2(t) + \dots + c_k\varphi_k(t) + \dots = \sum_{k=0}^{\infty} c_k\varphi_k(t)$$

где φ_k - ортонормированные функции.

Коэффициенты c_k вычисляются по формуле:

$$c_k = \langle f(t), \varphi_k(t) \rangle = \frac{1}{b-a} \int_a^b f(t) * \varphi_k(t) dt$$

Условие ортонормированности выполняется, если скалярное произведение любых двух функций, входящих в набор, равно нулю, а норма любой функции равна единице:

$$\langle \varphi_m(t), \varphi_n(t) \rangle = \frac{1}{b-a} \int_a^b \varphi_m(t) * \varphi_n(t) dt = 0, \quad m \neq n$$

$$\|\varphi_m(t)\| = \langle \varphi_m(t), \varphi_m(t) \rangle = \frac{1}{b-a} \int_a^b \varphi_m^2(t) dt = 1$$

Разложение функции на интервале $[-\pi, \pi]$

Функцию $f(t)$ можно разложить по системе тригонометрических функций на отрезке $[-\pi, \pi]$ следующим образом:

$$f(t) = \alpha_0 + \alpha_1 \sqrt{2} \cos t + \alpha_2 \sqrt{2} \cos 2t + \alpha_3 \sqrt{2} \cos 3t + \dots + \\ + \beta_1 \sqrt{2} \sin t + \beta_2 \sqrt{2} \sin 2t + \beta_3 \sqrt{2} \sin 3t + \dots$$

Коэффициенты α_k, β_k , как было показано ранее, можно выразить через скалярные произведения:

$$\alpha_0 = \langle f(t), 1 \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) dt$$

$$\alpha_k = \langle f(t), \sqrt{2} \cos kt \rangle = \frac{\sqrt{2}}{2\pi} \int_{-\pi}^{\pi} f(t) \cos kt dt$$

$$\beta_k = \langle f(t), \sqrt{2} \sin kt \rangle = \frac{\sqrt{2}}{2\pi} \int_{-\pi}^{\pi} f(t) \sin kt dt$$

В общем виде функцию $f(t)$ можно представить следующим образом:

$$f(t) = \frac{a_0}{2} + a_1 \cos t + a_2 \cos 2t + a_3 \cos 3t + \dots + \\ + b_1 \sin t + b_2 \sin 2t + b_3 \sin 3t + \dots = \\ = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt).$$

Коэффициенты a_0, a_k, b_k называют коэффициентами Фурье, а подобное представление функции называется разложением в ряд Фурье. Иногда такое представление называют действительным разложением в ряд Фурье, а коэффициенты — действительными коэффициентами Фурье. Термин «действительный» вводится для того, чтобы отличить представленное разложение от разложения в ряд Фурье в комплексной форме, о котором будет сказано далее. Как уже было сказано раньше, произвольную функцию можно разложить по системе ортогональных функций, даже если функции из этой системы не представляются в виде тригонометрического ряда. Обычно под разложением в ряд Фурье подразумевается разложение в тригонометрический ряд. Если коэффициенты Фурье выразить через a_0, a_k, b_k , получим:

$$a_0 = 2\alpha_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) dt \quad (2.1)$$

$$a_k = \sqrt{2}\alpha_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos kt dt$$

$$b_k = \sqrt{2}\beta_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin kt dt$$

Поскольку при $k = 0$ $\cos(kt) = 1$, то константа $a_0/2$ выражает общий вид коэффициента a_k при $k = 0$.

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(kt) dt \quad (k = 0, 1, 2, \dots)$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(kt) dt \quad (k = 0, 1, 2, \dots)$$

В последнем соотношении колебание самого большого периода, представленное суммой $\cos(t)$ и $\sin(t)$, называют колебанием основной частоты или *первой гармоникой*. Колебание с периодом, равным половине основного периода, называют *второй гармоникой*. Колебание с периодом, равным $1/3$ основного периода, называют *третьей гармоникой* и т.д. Как видно из соотношения (2.1) a_0 является постоянной величиной, выражающей среднее значение функции $f(t)$. Если функция $f(t)$ представляет собой электрический сигнал, то a_0 представляет его постоянную составляющую. Следовательно, все остальные коэффициенты Фурье выражают его переменные составляющие.

Первая гармоника является периодической функцией с периодом 2π . Прочие гармоники также имеют период, кратный 2π . Исходя из этого, при формировании сигнала из составляющих ряда Фурье мы, естественно, получим периодическую функцию с периодом 2π . А если это так, то разложение в ряд Фурье — это, собственно говоря, способ представления периодических функций.

Итак, каждая из гармоник, представленных рядом Фурье, представляет собой сумму вида $a^* \cos(x) + b^* \sin(x)$. Эта сумма может быть преобразована, как будет показано ниже, к виду $\sqrt{a^2 + b^2} * \sin(x + \varphi)$.

Лемма. Если сумма квадратов двух действительных чисел равна единице, то одно из этих чисел можно рассматривать как косинус, а другое как синус некоторого угла.

Другими словами, если $a^2 + b^2 = 1$, то существует угол φ , такой, что

$$a = \cos \varphi; \quad b = \sin \varphi.$$

Прежде чем доказывать эту лемму, поясним ее на следующем примере:

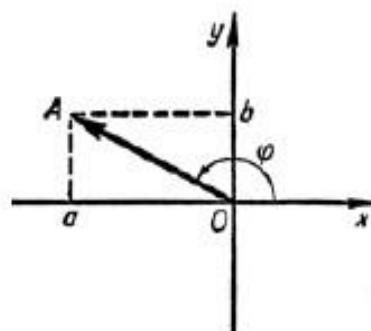
$$(\sqrt{3}/2)^2 + (1/2)^2 = 3/4 + 1/4 = 1$$

Поэтому существует угол φ , такой, что $\sqrt{3}/2 = \cos \varphi$; $1/2 = \sin \varphi$.

В качестве φ в данном случае можно выбрать любой из углов 30° , $30^\circ \pm 360^\circ$, $30^\circ \pm 2 \cdot 360^\circ$ и т. д.

Доказательство леммы.

Рассмотрим вектор OA с координатами (a, b) . Поскольку $a^2 + b^2 = 1$, длина этого вектора равна 1. Но в таком случае его координаты должны быть равны $\cos \varphi$ и $\sin \varphi$, где φ — угол, который образует данный вектор с осью абсцисс.



Итак, $a = \cos \varphi$; $b = \sin \varphi$, что и требовалось доказать.

Доказанная лемма позволяет преобразовать выражение $a \sin x + b \cos x$ к более удобному для изучения виду.

Прежде всего вынесем за скобки выражение $\sqrt{a^2 + b^2}$

$$a \sin x + b \cos x = \sqrt{a^2 + b^2} \left(\frac{a}{\sqrt{a^2 + b^2}} \sin x + \frac{b}{\sqrt{a^2 + b^2}} \cos x \right)$$

Поскольку

$$\left(\frac{a}{\sqrt{a^2+b^2}}\right)^2 + \left(\frac{b}{\sqrt{a^2+b^2}}\right)^2 = 1,$$

первое из чисел $\frac{a}{\sqrt{a^2+b^2}}$ и $\frac{b}{\sqrt{a^2+b^2}}$ можно рассматривать как косинус некоторого угла φ , а второе как синус того же угла φ :

$$\frac{a}{\sqrt{a^2+b^2}} = \cos \varphi, \quad \frac{b}{\sqrt{a^2+b^2}} = \sin \varphi.$$

Но в таком случае

$$a \sin x + b \cos x = \sqrt{a^2+b^2} (\cos \varphi \sin x + \sin \varphi \cos x) = \sqrt{a^2+b^2} \sin (x + \varphi)$$

Итак,

$$a \sin x + b \cos x = \sqrt{a^2+b^2} \sin (x + \varphi), \text{ где угол } \varphi \text{ определяется из условий}$$

$$\sin \varphi = \frac{b}{\sqrt{a^2+b^2}} \quad \cos \varphi = \frac{a}{\sqrt{a^2+b^2}}$$

Разложение функции, представленной в дискретизированном виде, в ряд Фурье на интервале $[-\pi, +\pi]$

В дискретизированном виде (т.е. в виде набора дискретных значений или, что то же, в виде числового массива, содержащего N значений) функция $f(t)$ на интервале $[-\pi, +\pi]$ будет иметь вид:

$$f(t) \rightarrow f\left(\frac{2\pi i}{N}\right), \quad i = \left[-\frac{N}{2}, \frac{N}{2}\right]$$

где N – количество дискретных значений сигнала. При $n \rightarrow \infty$ дискретизированная функция будет приближаться к непрерывной $f(t)$.

Разложение в ряд Фурье будет иметь вид:

$$f(t) = \alpha_0 + \sum_{k=1}^{\infty} (\alpha_k \sqrt{2} \cos kt + \beta_k \sqrt{2} \sin kt)$$

а коэффициенты α_0, α_k и β_k :

$$\alpha_0 = \left\langle f\left(\frac{2\pi i}{N}\right), 1 \right\rangle = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right);$$

$$\alpha_k = \left\langle f\left(\frac{2\pi k}{N}\right), \sqrt{2} \cos\left(\frac{2\pi k}{N}\right) \right\rangle = \frac{\sqrt{2}}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right) \cos\left(\frac{k * 2\pi i}{N}\right);$$

$$\beta_k = \left\langle f\left(\frac{2\pi i}{N}\right), \sqrt{2} \sin\left(\frac{2\pi k}{N}\right) \right\rangle = \frac{\sqrt{2}}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right) \sin\left(\frac{k * 2\pi i}{N}\right).$$

и коэффициенты a_0 , a_k и b_k :

$$a_0 = 2\alpha_0 = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right)$$

$$a_k = \sqrt{2}\alpha_k = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right) \cos\left(\frac{k * 2\pi i}{N}\right)$$

$$b_k = \sqrt{2}\beta_k = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2\pi i}{N}\right) \sin\left(\frac{k * 2\pi i}{N}\right)$$

а общий вид разложения:

$$\begin{aligned} f\left(\frac{2\pi i}{N}\right) &= \frac{a_0}{2} + a_1 \cos\left(\frac{2\pi i}{N}\right) + a_2 \cos\left(\frac{4\pi i}{N}\right) + a_3 \cos\left(\frac{6\pi i}{N}\right) + \dots \\ &\quad + b_1 \sin\left(\frac{2\pi i}{N}\right) + b_2 \sin\left(\frac{4\pi i}{N}\right) + b_3 \sin\left(\frac{6\pi i}{N}\right) + \dots \end{aligned}$$

или

$$\begin{aligned} f\left(\frac{2\pi i}{N}\right) &= \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos\left(\frac{k * 2\pi i}{N}\right) + b_k \sin\left(\frac{k * 2\pi i}{N}\right)) = \\ &= \frac{a_0}{2} + \sum_{k=1}^{\infty} (\sqrt{a_k^2 + b_k^2} \sin\left(\frac{k * 2\pi i}{N} + \varphi_k\right)), \text{ где } \varphi_k = \arctg \frac{b_k}{a_k} \end{aligned}$$

Массив $C_k = \sqrt{a_k^2 + b_k^2}$ представляет спектр амплитуд, а φ_k - спектр фаз.

Оставив функцию $f(t)$ без изменения, получим

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos \omega_0 k t \, dt \quad (k = 0, 1, 2, \dots)$$

Аналогичным образом выводится следующее соотношение:

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin \omega_0 k t \, dt \quad (k = 1, 2, \dots).$$

Разложение функции, представленной в дискретизированном виде, в действительный ряд Фурье на интервале $[-T, T]$

В дискретизированном виде (т.е. в виде набора дискретных значений или, что то же, в виде числового массива, содержащего N значений) функция $f(t)$ на интервале $[-T, T]$ будет иметь вид:

$$f(t) \rightarrow f\left(\frac{2Ti}{N}\right), \quad i = \left[-\frac{N}{2}, \frac{N}{2}\right]$$

где N – количество дискретных значений сигнала. При $n \rightarrow \infty$ дискретизированная функция будет приближаться к непрерывной $f(t)$.

Разложение в ряд Фурье будет иметь приведенный ранее вид:

а коэффициенты α_0 , α_k и β_k :

$$\alpha_0 = \left\langle f\left(\frac{2Ti}{N}\right), 1 \right\rangle = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right);$$

$$\alpha_k = \left\langle f\left(\frac{2Ti}{N}\right), \sqrt{2} \cos\left(\frac{2\pi k}{N}\right) \right\rangle = \frac{\sqrt{2}}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right) \cos\left(\frac{k * 2\pi i}{N}\right);$$

$$\beta_k = \left\langle f\left(\frac{2Ti}{N}\right), \sqrt{2} \sin\left(\frac{2\pi k}{N}\right) \right\rangle = \frac{\sqrt{2}}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right) \sin\left(\frac{k * 2\pi i}{N}\right).$$

и коэффициенты a_0 , a_k и b_k :

$$a_0 = 2\alpha_0 = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right)$$

$$a_k = \sqrt{2}\alpha_k = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right) \cos\left(\frac{k*2\pi i}{N}\right)$$

$$b_k = \sqrt{2}\beta_k = \frac{2}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} f\left(\frac{2Ti}{N}\right) \sin\left(\frac{k*2\pi i}{N}\right)$$

а общий вид разложения:

$$f\left(\frac{2Ti}{N}\right) = \frac{a_0}{2} + a_1 \cos\left(\frac{2\pi i}{N}\right) + a_2 \cos\left(\frac{4\pi i}{N}\right) + a_3 \cos\left(\frac{6\pi i}{N}\right) + \dots \\ + b_1 \sin\left(\frac{2\pi i}{N}\right) + b_2 \sin\left(\frac{4\pi i}{N}\right) + b_3 \sin\left(\frac{6\pi i}{N}\right) + \dots$$

Разложение функции в комплексный ряд Фурье

Система функций $\{e^{jki}, k = 0, \pm 1, \pm 2, \dots\}$ на отрезке $[-\pi, \pi]$ образует ортонормированную систему функций. Значит, произвольная функция $f(t)$ может быть представлена по этой системе следующим образом:

$$f(t) = \sum_{k=-\infty}^{\infty} C_k e^{jki}$$

Это и есть разложение в комплексный ряд Фурье. Коэффициенты C_k называются комплексными коэффициентами Фурье и, подобно действительным коэффициентам Фурье, вычисляются как скалярные произведения $f(t)$ и e^{jki} :

$$C_k = \langle f(t), e^{jki} \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-jki} dt$$

Если период функции не равен 2π , а, например, равен T , то получим следующее общее выражение для комплексных коэффициентов:

$$f(t) = \sum_{k=-\infty}^{\infty} C_k e^{j\omega_0 ki} \\ C_k = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j\omega_0 ki} dt \quad (\omega_0 = 2\pi / T)$$

Содержание работы

1. Разработайте программы разложения дискретизированного сигнала $f(x)=x, x^2, |x|, |\sin(x)|, x*\sin(x), x*\cos(x), \exp(ax), \ln[2\cos(x/2)], -\ln[\lg(x/2)]/2, \text{sign}[\sin(x)]$ в дискретизированный и комплексный ряд Фурье с конечным числом членов.
2. Исследуйте зависимость точности представления заданной функции с помощью ряда Фурье от числа членов разложения и от шага дискретизации функции.

Указания к выполнению

Используйте в качестве основы программы, приведенные в приложении.

Содержание отчета

1. Задание к работе.
2. Тексты разработанных программ с комментариями.
3. Графики зависимости погрешности представления заданной функции в действительный и комплексный ряд Фурье на интервале $[-\pi, \pi]$ и $[-\frac{T}{2}, \frac{T}{2}]$ от количества членов разложения в ряд Фурье и от шага дискретизации.
4. Выводы.

Приложение. Тексты базовых программ

Приложение 1. Программа вычисления коэффициентов разложения в

действительный ряд Фурье для функции $y(t)=t \rightarrow \frac{2\pi i}{N}, i = [-\frac{N}{2}, \frac{N}{2}]$.

% Разложение функции t в ряд Фурье

%в дискретизированном виде на интервале $[-T, T]$, например, $[-\pi, \pi]$

N=255; %Количество отсчетов (элементов массива y(t))

K=16; %Количество членов ряда Фурье

T=pi; %диапазон изменения функции f(i) равен +/-T

kp=2.4; %количество периодов гармонической функции

y=zeros(1,N+1);

Sa = zeros(1,K);

Sb = zeros(1,K);

p=1; % показатель степени функции t^p

f=zeros(1,N+1);

Sa0=0;

for i=1:N+1

 f(i)=sin(2*pi*kp*(i-1)/N); % гармоническая функция

 % f(i)= (2*T*((i-1-N/2)/N))^p; %функция t^p

 Sa0=Sa0+f(i);

end

Sa0=Sa0/N

for i=1:N+1

 for j=1:K

 Sa(j) = (Sa(j)+f(i)*cos((j)*2*pi*(i-1-N/2)/N));

 Sb(j) = (Sb(j)+f(i)*sin((j)*2*pi*(i-1-N/2)/N));

 end

end

for j=1:K

 Sa(j)=Sa(j)*(1/(N/2));

 Sb(j)=Sb(j)*(1/(N/2));

end

%Вычисление и отображение спектра амплитуд (начало)


```

for j=1:K
    Sab(j)=sqrt(Sa(j)^2+Sb(j)^2);
end
i=1:K;
figure
plot(i,Sab);
stem(Sab(1:K)); %вывод графика дискретной последовательности данных
axis([1 8 -0.2 1.2]); %задание осей: [xmin xmax ymin ymax]
title('Амплитуды частотных составляющих спектра');
xlabel('Количество периодов')
axis tight;
%Вычисление и отображение спектра амплитуд (конец)
y=zeros(1,N+1);
for i=1:N+1
    for j=1:K
        y(i)= y(i)+Sa(j)*cos(j*2*pi*(i-1-N/2)/N)+Sb(j)*sin(j*2*pi*(i-1-N/2)/N);
    end
    y(i)=Sa0+y(i);
end
i=1:N+1;
figure
plot(i,f);
axis tight;
hold on;
plot(i,y,'r-')
hold off;
pause;
close all;
clear; %очистка Workspace

```

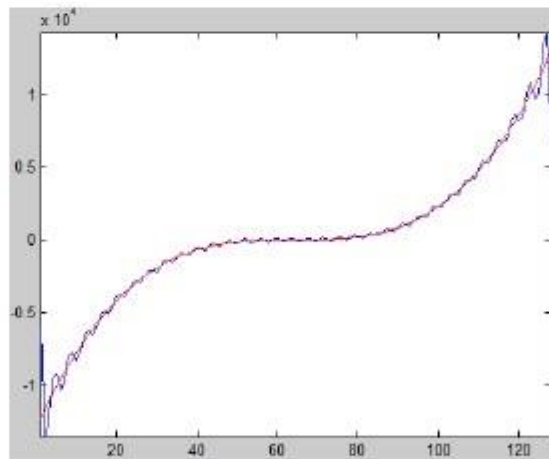


Рис. 2.1. Исходная и восстановленная функция t^3 при $N=128$, $K=32$, $p=3$.

Примечание. Для разложения четной функции из ряда можно исключить члены, содержащие синусы, для разложения нечетной – косинусы. Но можно оставить ряд разложения полностью.

Приложение 2. Программа вычисления коэффициентов разложения в действительный ряд Фурье для функции $y(t) = t \rightarrow \frac{2Ti}{N}, \quad i = [-\frac{N}{2}, \frac{N}{2}]$.

% Разложение функции $y(t)=t$ в ряд Фурье
 % в дискретизированном виде на интервале $[0, T]$, например, $[0, \pi]$

```
N=255; %Количество отсчетов (элементов массива y(t)=t)
K=64; %Количество членов ряда Фурье
T=pi; %диапазон изменения функции f(i)+/-T
kp=2.0
y=zeros(1,N+1);
Sa = zeros(1,K);
Sb = zeros(1,K);
p=3; %показатель степени функции  $t^p$ 
f=zeros(1,N+1);
Sa0=0;
for i=1:N+1
    f(i)=sin(2*pi*kp*(i-1)/N); % гармоническая функция
    % f(i)= (T*(((i-1))/N))^p; %функция  $t^p$ , i-1, если p>0, i, если p<0
    Sa0=Sa0+f(i);
end
Sa0=Sa0/N
for i=1:N+1
    for j=1:K
        Sa(j) = (Sa(j)+f(i)*cos((j)*2*pi*(i-1)/N));
        Sb(j) = (Sb(j)+f(i)*sin((j)*2*pi*(i-1)/N));
    end
end
for j=1:K
    Sa(j)=Sa(j)*(1/(N/2));
    Sb(j)=Sb(j)*(1/(N/2));
end
%Вычисление и отображение спектра амплитуд (начало)
for j=1:K
    Sab(j)=sqrt(Sa(j)^2+Sb(j)^2);
```

```

end
i=1:K;
figure
plot(i,Sab);
stem(Sab(1:K)); %вывод графика дискретной последовательности данных
axis([1 8 -0.2 1.2]); %задание осей: [xmin xmax ymin ymax]
title('Амплитуды частотных составляющих спектра');
xlabel('Количество периодов')
axis tight;
%Вычисление и отображение спектра амплитуд (конец)
y=zeros(1,N+1);
for i=1:N+1
    for j=1:K
        y(i)=y(i)+Sa(j)*cos(j*2*pi*(i-1-N)/N)+Sb(j)*sin(j*2*pi*(i-1-N)/N);
    end
    y(i)=Sa0+y(i);
end
i=1:N+1;
figure
plot(i,f);
axis tight;
hold on;
plot(i,y,'r-')
hold off;
pause;
close all;
clear; %очистка Workspace

```

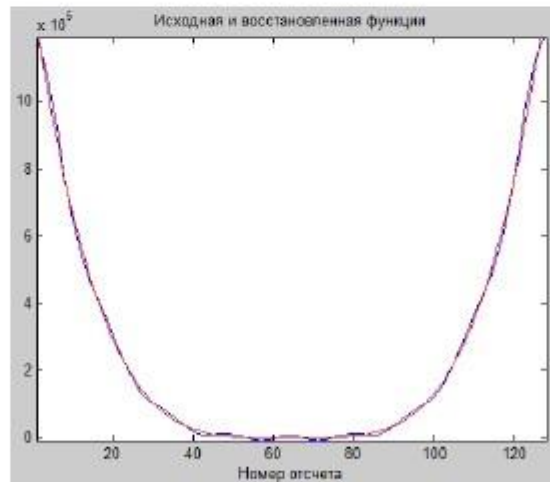


Рис. 2.2. Исходная и восстановленная функция $y=t^p$ после разложения в ряд Фурье, $N=128$, $T=5$, $p=4$, $K=16$.

Приложение 3. Программа вычисления коэффициентов разложения в

комплексный ряд Фурье для функции $y(t)=t^2 \rightarrow \left(\frac{2Ti}{N}\right)^2, \quad i=\left[-\frac{N}{2}, \frac{N}{2}\right]$.

%Разложение функции t^p в комплексный ряд Фурье

%в дискретизированном виде на интервале $[0, T]$

%Восстановление функции производится по формуле

% $f_b(i)=y(i)=\sum(c(k)*\exp(j*2*\pi*i*k/N)), k=[1,M], i=[0,N-1]$

%Чем больше M , тем точнее восстановление

$T=\pi$; %Значение T (произвольное)

$N=128$; %количество значений функции на интервале $[0, T]$

$M=6$; %количество членов ряда Фурье

$p=1$; %показатель степени функции x^p

$k_p=2.4$; %количество периодов гармонического сигнала

$C0=0$;

for $i=1:N+1$ %генерация модельной функции

$f(i)=\sin(2*\pi*k_p*(i-1)/N)$; % гармоническая функция

% $f(i)=(T*((i-1)/N))^p$; %функция t^p

$C0=C0+f(i)$;

end

$C0=C0*(2/N)$;

for $k=1:M$

$C(k)=0$;

end

for $i=1:N+1$

for $k=1:M$

$C(k)=C(k)+f(i)*\exp(-j*2*\pi*k*(i-1)/N)$;

end

end

for $k=1:M$

$C(k)=C(k)*(2/N)$;

end

%Вычисление и отображение спектра амплитуд (начало)

for $k=1:M$

$Cab(k)=abs(C(k))$; %коэффициенты $Cab(k)$ - комплексные числа вида $a+jb$,

%функция abs вычисляет $\sqrt{a^2+b^2}$

```

end
k=1:M;
figure
plot(k,Cab);
stem(Cab(1:M)); %вывод графика дискретной последовательности данных
axis([1 8 -0.2 1.2]);%задание осей: [xmin xmax ymin ymax]
title('Амплитуды частотных составляющих спектра');
xlabel('Количество периодов')
axis tight;
%Вычисление и отображение спектра амплитуд (конец)
for i=1:N+1
    y(i)=0;
    %f3(i)=0;
    for k=1:M
        y(i)=y(i)+C(k)*exp(j*2*pi*k*(i-1)/N);
    end
    y(i)=C0/2+y(i);
end
i=1:N+1;
figure
plot(i,f);
axis tight;
title('Исходная и
восстановленная функция')
xlabel('Номер элемента массива')
hold on;
plot(i,real(y),'r-');
axis tight;
hold off;
pause;
close all;
clear;

```

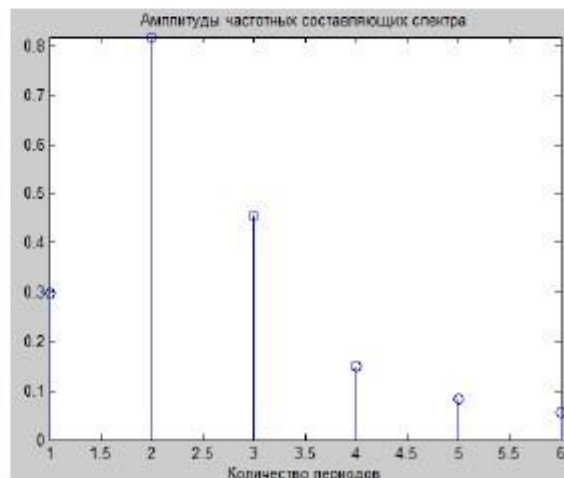


Рис. 2.3. Спектр амплитуд гармонического сигнала с количеством периодов $k_p=2.4$

Задание к работе

Имеется набор экспериментальных данных в виде числового массива. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки данных, реализующую различные способы улучшения отношения сигнал/шум:

1. Накопление.
2. НЧ-фильтр.
3. Фильтр Баттерворта.
4. Фильтр скользящего среднего.
5. Фильтр БПФ - ОБПФ.
6. Оптимальный фильтр Колмогорова-Винера.

и оценить сравнительный эффект улучшения отношения сигнал/шум и степень искажения сигнала в результате обработки.

Теоретические основы

Цифровая обработка сигналов решает задачи обнаружения и определения параметров информативных сигналов и изображений, искаженных шумами и помехами. Для этой цели используются различные средства:

- накопление (временная фильтрация);
- цифровые частотные фильтры (высокой частоты, низкой частоты, полосовые фильтры, фильтр Баттерворта);
- сглаживающие фильтры (скользящего среднего, медианный);
- оптимальные фильтры (фильтр Колмогорова-Винера, LMS и RLS-фильтры);
- адаптивные фильтры;

Выбор способа борьбы с шумами должен производиться с учетом свойств и особенностей информативного сигнала и помехи. Чем в большей степени свойства сигнала и шума априори известны, тем может быть получен больший эффект от цифровой обработки. Кроме того, несмотря на наличие стандартных программ цифровой обработки, с учетом конкретных априори известных свойств информативного сигнала и шума может оказаться полезным разработка новых методов борьбы с шумами.

ЦИФРОВОЙ НИЗКОЧАСТОТНЫЙ ФИЛЬТР

Для фильтрации высокочастотного шума может быть применен фильтр низких частот (ФНЧ). Частотная характеристика ФНЧ выражается как

$$K(\omega) = \frac{1}{1 + j\omega}, \quad \text{где } \omega - \text{частота среза НЧ - фильтра}$$

Для фильтрации сигнала нужно вычислить частотный спектр сигнала с помощью преобразования Фурье, затем перемножить частотный спектр сигнала и частотную функцию фильтра и выполнить обратное преобразование Фурье. Второй способ – вычислить импульсную переходную характеристику фильтра (реакцию на единичный импульс), затем выполнить операцию свертки входного сигнала с импульсной переходной характеристикой фильтра.

В программах обработки дискретизированных сигналов, представленных в форме числовых массивов, цикл

```
for k=1:N
```

```
x(k) = A*sin(2*pi*KP*k/N);
```

```
end
```

создает KP периодов дискретизированного синусоидального сигнала в числовом массиве, содержащем N значений, понятие частоты в этом случае отсутствует и появляется только в том случае, если задать шаг дискретности по времени. Аналогично этому, в результате выполнения быстрого преобразования Фурье

```
I=1:N;
```

```
Y=fft(x,N);
```

мы получаем числовой массив, который может содержать N элементов, причем информативной будет только первая половина массива, вторая будет зеркально отображать первую половину. В первой половине массива, содержащей N/2

элементов, будет представлен «частотный» спектр. Спектр будет отражать не частоту сигнала, установить которую по числовому массиву, представляющему сигнал во временной области, невозможно, а количество периодов. Т.е. в приведенном выше примере пик в массиве частотного спектра будет в элементе с номером КР. Таким образом, положение пика (номер элемента массива частотного спектра) укажет на количество периодов сигнала во временной области.

Частотная характеристика линейного фильтра низких частот может быть вычислена следующим образом:

```
for i=1:N
H(i)=1/((1+j*i/NC));
end
```

Здесь NC - полоса пропускания фильтра по уровню 0,7 амплитуды выражена в количестве отчетов спектра БПФ, пропускаемых фильтром. Остальные отсчеты в массиве частотного спектра будут ослабляться по амплитуде. Таким образом, понятие постоянной времени фильтра, равно как и полосы пропускания, при дискретизированном представлении линейного фильтра отсутствует.

Ниже приведены программы фильтрации сигналов и временные диаграммы.

%Низкочастотный фильтр

```
A=1; %амплитуда сигнала
Q=0.05; %амплитуда шума
KP1=5;% - количество периодов первого сигнала
KP2=5;% - количество периодов второго сигнала
N=1024;%количество отсчетов
NC=5;
%NC - полоса пропускания фильтра по уровню 0,7 амплитуды
% выражена в количестве отчетов спектра БПФ, пропускаемых фильтром
% остальные отсчеты (в частотном спектре!) будут ослабляться по амплитуде
```

```
for k=1:N % генерация сигнала и шума
s(k) = A*sin(2*pi*KP1*k/N);%+ A*sin(2*pi*KP2*k/N);
q(k)=Q*(randn(size(N))); %СКО шума равно Q
x(k)=s(k)+q(k); % суммирование сигнала и шума
```

```

end
figure
plot(x);
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax
title('Зашумленный сигнал до фильтра');
Y=fft(x,N); %БПФ сигнала с шумом
i=1:N/2;
figure
% semilogy(i(1:200),2*abs(SS1(1:200)));
%plot(i(1:100),2*abs(SS1(1:100)));
plot(i(1:N/2),abs(Y(1:N/2)));
title('Частотный спектр сигнала с шумом');
for i=1:N;
H(i)=1/((1+j*i/NC)); %передаточная функция фильтра НЧ 1-го порядка
%в частотной области
end
h=ifft(H);
% HH=fft(h,N);
i=1:N;
plot(i(1:20),abs(h(1:20))); %импульсная характеристика фильтра
title('Импульсная характеристика фильтра');
i=1:200;
figure
%plot(i,abs(H(1:200)));
semilogx(i,abs(H(1:200)));%то же, что и plot, но в логарифмическом
%масштабе по X
grid on;
title('Частотная хар-ка НЧ-фильтра');
i=1:N;
XX1=fft(x,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
XX2=fft(s,N);%частотный спектр сигнала
Z2=ifft(XX2.*H);%свертка незашумленного сигнала с частотной хар-кой
фильтра
DZ(i)=(2*real(Z(i))-2*real(Z2(i)))*100./max(real(Z2));%случайная погрешность
DZ1(i)=(2*real(Z(i))-x(i))*100./max(x);%полная погрешность

```

```

SKO=std(DZ)
SKO1=std(DZ1)
i=1:N;
yy=A*sin((6.28*KP1*i/N));
figure
plot(i,x); %вывод сигнала до фильтра
title('Сигнал до фильтра');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Амплитуда'); % подпись по оси Y
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax
hold on; % "удержание" окна вывода для вывода следующего графика
i=1:N;
plot(i,2*real(Z(1:N)),'r-'),grid; %вывод отфильтрованного сигнала
%представление графика линией красного цвета, отображение сетки
title('Сигнал до и после фильтра');%подпись названия графика
hold off;
i=1:N;
figure
plot(i,DZ(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Случайная погрешность отфильтрованного сигнала');
ylabel('Случайная погрешность, %'); % подпись по оси Y
axis tight;
i=1:N;
figure
plot(i,DZ1(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Полная погрешность отфильтрованного сигнала');
ylabel('Полная погрешность, %'); % подпись по оси Y
axis tight;
pause;
close all;%закрытие окон графического вывода
clear;%очистка Workspace

```

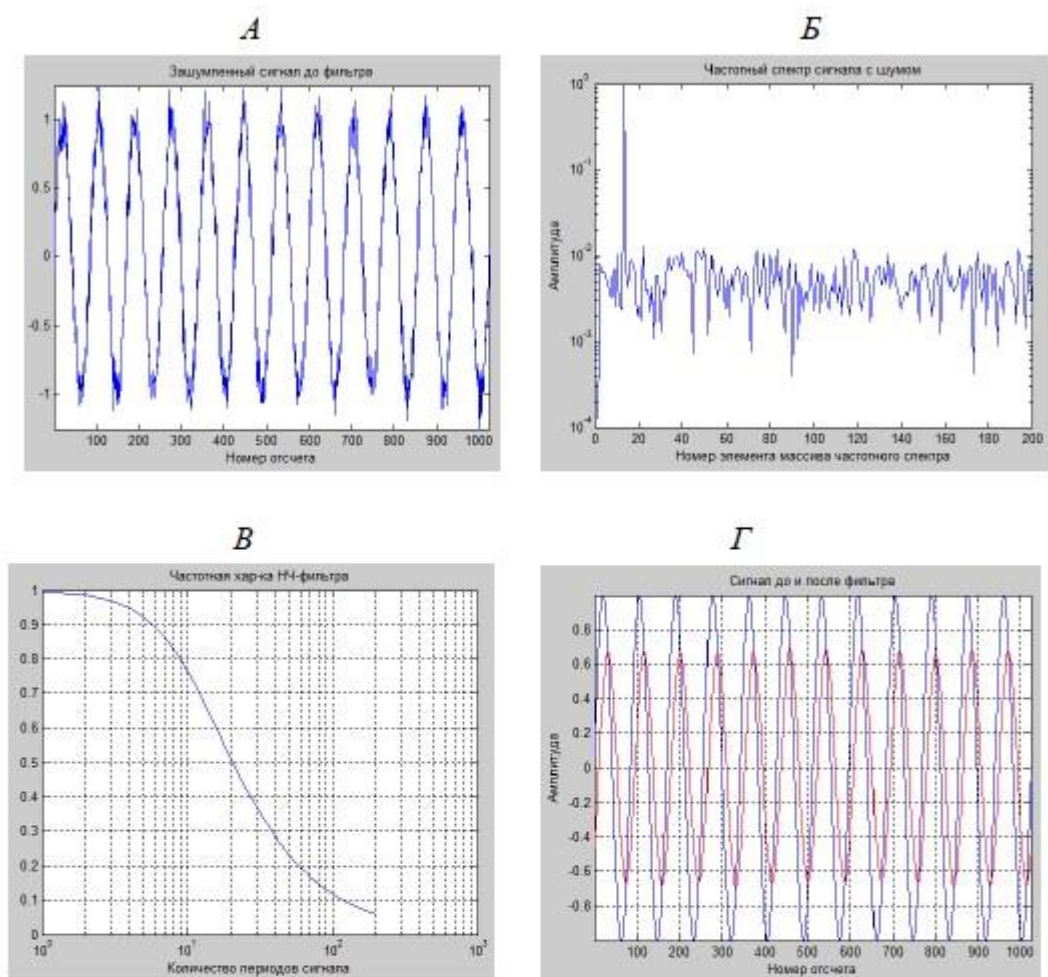



Рис.3.1. Исходный сигнал с шумом (А), его частотный спектр (Б), частотная характеристика фильтра в полулогарифмическом масштабе (В) и сигнал до и после фильтра (Г).

ФИЛЬТР БАТТЕРВОРТА

Недостатком линейного цифрового НЧ-фильтра первого порядка является существенная неравномерность амплитудно-частотной характеристики в полосе пропускания, малая крутизна ее спада. Эти недостатки могут быть в некоторой степени преодолены при использовании фильтров более высоких порядков, в значительной мере указанные недостатки преодолены в фильтрах Баттерворта.

Ниже приведена программа фильтрации сигналов с помощью фильтра Баттерворта 4-го порядка и временные диаграммы.

%Фильтр НЧ Баттерворта 4-го порядка

A=1; %амплитуда сигнала

Q=0.3; %СКО шума

KP1=5;% - количество периодов первого сигнала

KP2=5;% - количество периодов второго сигнала

N=1024;%количество точек расчета

NC=15;

%NC - полоса пропускания фильтра по уровню 0,7 амплитуды

% выражена в количестве отчетов спектра БПФ, пропускаемых фильтром

% остальные отсчеты (в частотном спектре!) будут ослабляться по амплитуде

for k=1:N % генерация сигнала и шума

s(k) = A*sin(2*pi*KP1*k/N); %+ A*sin(2*pi*KP2*k/1000);

q(k)=Q*(randn(size(N))); %СКО шума, амплитуда равна 3Q

x(k)=s(k)+q(k); % суммирование сигнала и шума

end

figure

plot(x);

title('Зашумленный сигнал до фильтра');

Y=fft(x,N)/N; %БПФ сигнала с шумом

i=1:N/2;

figure

% semilogy(i(1:200),2*abs(SS1(1:200)));d

%plot(i(1:100),2*abs(SS1(1:100)));

plot(i(1:N/2),real(Y(1:N/2)));

title('Частотный спектр сигнала с шумом');

for i=1:N;

H(i)=1/(sqrt(1+(j*i/NC).^4)); %передаточная функция фильтра Баттерворта 4-го порядка

end

i=1:200;

figure

%plot(i,abs(H(1:200))); %вывод частотной хар-ки фильтра

```

semilogx(i,abs(H(1:200))),grid;%то же, что и plot, но в логарифмическом
%масштабе по X
%loglog(i,abs(H(1:200))),grid;% в логарифм. масштабе по X и Y
axis tight;
title('Частотная хар-ка фильтра Баттерворта');

i=1:N;
XX1=fft(x,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
XX2=fft(s,N);%частотный спектр сигнала без шума
Z2=ifft(XX2.*H);%свертка сигнала с частотной хар-кой фильтра
DZ=(2*real(Z(i))-2*real(Z2(i)))/A*100;%вычисление случайной составляющей
погрешности
DZ1=(2*real(Z(i))-s(i))/A*100;%вычисление полной погрешности
SKO=std(DZ)
SKO1=std(DZ1)
figure
plot(i,x); %вывод сигнала до фильтра
%title('Сигнал до фильтра');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Амплитуда'); % подпись по оси Y
legend('до фильтра');
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax
hold on; % "удержание" окна вывода для вывода следующего графика
plot(i,2*real(Z(1:N)),'r-'); %вывод отфильтрованного сигнала
%представление графика линией красного цвета
grid on; % отображение сетки
title('Сигнал до и после фильтра');%подпись названия графика
hold off;

i=1:N;
figure
plot(i,real(DZ(1:N))); %вывод случайной погрешности
title('Случайная погрешность отфильтрованного сигнала, %');
i=1:N;
figure

```



```

plot(i,real(DZ1(1:N))); %вывод полной погрешности
title('Полная погрешность отфильтрованного сигнала, %');

pause;
close all; %закрытие окон графического вывода
clear; %очистка Workspace

```

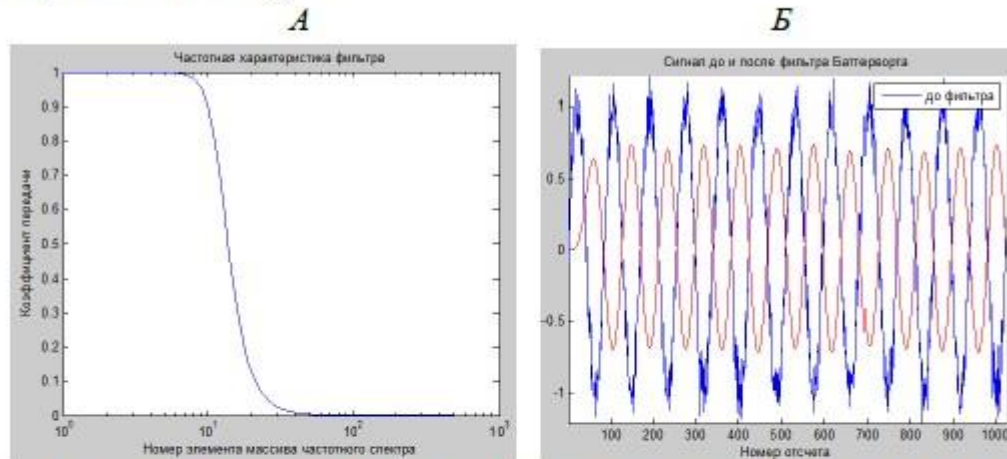


Рис.3.2. Частотная характеристика фильтра в полулогарифмическом масштабе (А) и сигнал до и после фильтра (Б).

ОПТИМАЛЬНЫЙ ФИЛЬТР КОЛМОГОРОВА-ВИНЕРА

Фильтры низкой частоты, высокой частоты и полосовые фильтры эффективны в том случае, когда частотные спектры сигнала и шума не перекрываются.

Наилучшее разделение сигнала и шума цифровыми методами обеспечивает оптимальный фильтр Колмогорова-Винера.

Частотная характеристика фильтра Колмогорова-Винера:

$$H(w) = W_s(w) / [W_s(w) + W_q(w)] ,$$

где $W_s(w)$ и $W_q(w)$ - энергетические спектры (плотности мощности) сигнала и помех.

Спектр плотности мощности W в MATLAB рассчитывается по формуле:

$$W(k) = \frac{1}{N * F_s} X(k) .* conj(X(k))$$

где $X(k)$ – N-точечное ДПФ N-точечной последовательности $x(n)$, N-размерность ДПФ, F_s – частота дискретизации.

$W(k)$ можно вычислять по более простой формуле:

$$W(k) = X(k) \cdot \text{conj}(X(k))$$

т.к. множитель $\frac{1}{N \cdot F_s}$ в выражении $H(w)$ равен единице.

Программа, реализующая оптимальный фильтр Колмогорова – Винера в среде MATLAB для трех часто встречающихся видов сигнала: гармонического, колоколообразного и прямоугольного, приведена ниже.

%Фильтр Колмогорова-Винера

```
A=1; %амплитуда сигнала
Q=0.1; %амплитуда шума (в долях СКО)
N=1024;%количество точек расчета
kp1=12;
kp2=24;
for k=1:N %цикл вычисления сигнала и шума
    %s1(k)=A*exp(-0.0003*(k-200)^2.0); %колоколообразный сигнал
    s1(k)=A*sin(2*pi*kp1*k/N);%+A*sin(2*pi*kp2*k/N);%гармонический сигнал
    % s1(k)=0; % сигнал прямоугольной формы
    % if (k>100)&(k<300) % сигнал прямоугольной формы
    %   s1(k)=A;
    % end
    q(k)=Q*(randn(size(N))); %шум
    x1(k)=s1(k)+q(k); % суммирование сигнала и шума
end
figure
plot(x1(1:N));
title('Зашумленный сигнал до фильтра');
axis tight;
Y=fft(x1,N)/N; %БПФ сигнала с шумом
SS1=Y.*conj(Y)/N; %спектр мощности
i=1:200;
figure
%plot(i,SS1(1:200));
semilogy(i,SS1(1:200)); %вывод спектра мощности сигнала с шумом
title('Частотный спектр сигнала с шумом');
```

```

Y=fft(s1,N)/N; %БПФ сигнала без шума
f = N/2*linspace(0,1,N/2);%вычисление шкалы для частотной области
SS1=Y.*conj(Y)/N; %спектр мощности сигнала без шума
Y1=fft(q,N)/N; %БПФ шума
f = N/2*linspace(0,1,N/2);
SS2=Y1.*conj(Y1)/N; %спектр мощности шума
for i=1:N
H(i)=SS1(i)/(SS1(i)+SS2(i));%частотная характеристика оптимального фильтра
end
i=1:200;
figure
%plot(i,abs(H(1:200)));
semilogx(i,abs(H(1:200)));
%hold on
title('Частотная характеристика оптимального фильтра');
i=1:N;
XX1=fft(x1,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
axis tight;
figure
plot(i,s1(1:N)); %вывод незашумленного сигнала до фильтра сигнала
title('Незашумленный сигнал до фильтра');
axis tight;
figure
plot(i,Z(1:N)); %вывод отфильтрованного сигнала
title('Сигнал после свертки с част. хар-кой оптимального фильтра');
axis tight;
i=1:N;
DZ(i)=Z(i)-s1(i);
DZ1=DZ*100/max(s1);
SKO=std(DZ1)

i=1:N;
figure
plot(i,DZ1(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Погрешность отфильтрованного сигнала');

ylabel('Полная погрешность, %'); % подпись по оси Y
axis tight;
pause;
close all; %закрытие окон графического вывода
clear; %очиска Workspace

```

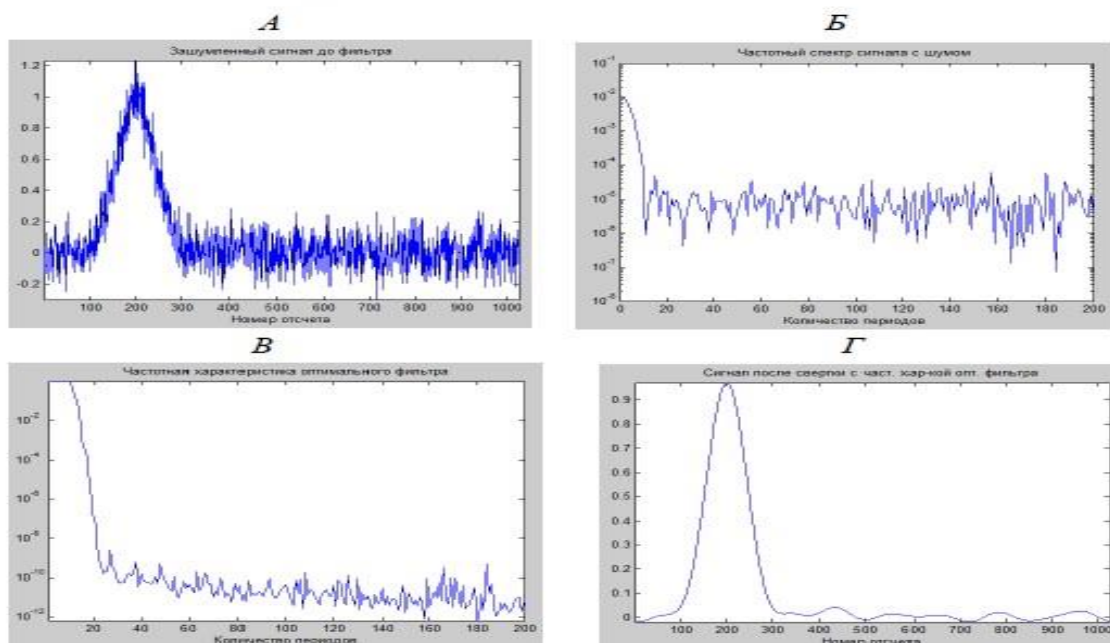


Рис. 3.3. Исходный колоколообразный сигнал с шумом (А), его частотный спектр (Б), частотная характеристика оптимального фильтра Колмогорова-Винера (В) и сигнал после фильтра (Г).

ФИЛЬТРАЦИЯ ШУМОВ С ИСПОЛЬЗОВАНИЕМ ПРЯМОГО И ОБРАТНОГО БПФ

В том случае, когда частотные спектры сигнала и шума не перекрываются, фильтрация шума может быть произведена путем выполнения БПФ, обнуления спектральных линий шума и последующего обратного БПФ.

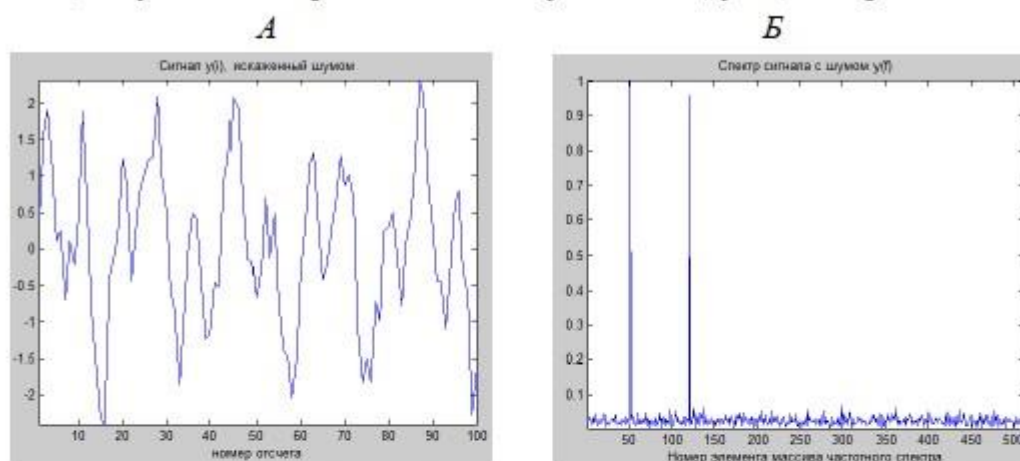


Рис. 3.4. Зашумленный сигнал (А), представляющий сумму двух синусоидальных сигналов разных частот (количество периодов во временном окне 50 и 120), и частотный спектр, полученный с помощью БПФ (Б).

Если теперь обнулить участки спектра от 0 до 40, от 55 до 110 и от 125 до 500, а затем выполнить обратное преобразование БПФ (ifft), то получим спектр и сигнал, представленные на рис. 3.5.

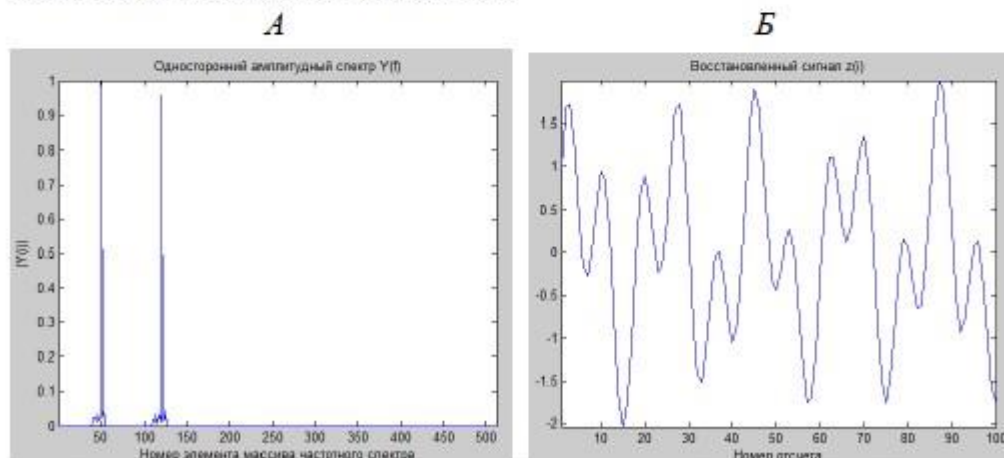


Рис. 3.5. Частотный спектр после удаления спектральных составляющих шума (А) и восстановленный с помощью обратного преобразования Фурье сигнал (Б)

Текст программы фильтрации помех с помощью прямого и обратного преобразования Фурье приведен ниже.

%Программа фильтрации помех с использованием БПФ-ОБПФ

A=1;%амплитуда первого сигнала

B=1; %амплитуда второго сигнала

Q=0.4;%уровень шума, выраженный в долях СКО. СКО шума равно 1.

N=1024;%количество отсчетов сигнала

kp1=50;%количество периодов 1-го сигнала

kp2=120;%количество периодов 2-го сигнала

% Сумма сигналов 1 и 2

for i=1:N

x(i) = A*sin(2*pi*kp1*i/N) + B*sin(2*pi*kp2*i/N);

z(i)=Q*randn(size(N));%

y(i) = x(i) + z(i); % Суммирование сигнала и шума

end

Y = fft(y,N)/N;%вычисление спектра сигнала с шумом

i=1:N/2;

figure

plot(i,2*abs(Y(1:N/2)));

title('Спектр сигнала с шумом y(f)')

xlabel('Номер элемента массива частотного спектра')

axis tight;

for i=1:N %"вырезание" спектральных составляющих шума

if (i>125)|(i<40)|((i>55)&(i<110))

Y(i)=0;

end

end

i=1:N/2;

figure

plot(i,2*abs(Y(1:N/2)));

title('Односторонний амплитудный спектр Y(f)')

xlabel('Номер элемента массива частотного спектра')

ylabel('|Y(i)|')

```

axis tight;
%восстановление исходного сигнала после "вырезания"
%спектральных составляющих шума
z=ifft(Y)*2*N;
i=1:N;
figure
plot(i(1:100),x(1:100));
title('Исходный сигнал x(i)')
xlabel('Номер отсчета');
axis tight;
figure
plot(i(1:100),y(1:100));
title('Сигнал y(i), искаженный шумом')
xlabel('номер отсчета')
axis tight;
figure
plot(i(1:100),z(1:100));
title('Восстановленный сигнал z(i)')
xlabel('Номер отсчета');
axis tight;
for i=1:N
DZ(i)=x(i)-real(z(i));%уровень зашумления в сигнале после фильтра
%результат обратного преобразования Фурье - массив комплексных чисел
%Восстановленный сигнал - действительные части комплексных чисел
%Вычисляются с помощью функции real
end
DZ1=DZ*100/max(x);
SKO=std(DZ1)
figure
i=1:N;
plot(i,DZ1(i:N)); %вывод погрешности отфильтрованного сигнала
title('Погрешность отфильтрованного сигнала');
ylabel('Случайная погрешность, %'); %подпись по оси Y
xlabel('Номер отсчета');
axis tight;

```



```

pause;
close all; %заккрытие всех окон графического вывода
clear;%очистка Workspace

```

ФИЛЬТР СКОЛЬЗЯЩЕГО СРЕДНЕГО

Пусть мы имеем массив N значений измеренного сигнала, представленный в цифровой форме:

$$\{f_1, f_2, \dots, f_N\}, i=1, 2, 3, \dots, N$$

Для нахождения скользящего среднего в окрестности точки f_i берем среднее арифметическое от K предыдущих и K последующих точек, включая и f_i . Таким же образом производим обработку для всех значений i . В результате вычисляем новый массив g_i :

$$g_i = \frac{1}{2K+1} (f_{i-K} + f_{i-K-1} + \dots + f_i + \dots + f_{i+K})$$

или

$$g_i = \frac{1}{2K+1} \sum_{j=-K}^K f_{i+j}$$

Текст программы фильтрации помех с помощью фильтра скользящего среднего приведен ниже.

```
%Фильтр скользящего среднего
```

```

A=1; %амплитуда сигнала
Q=0.05; %СКО шума
KP1=10;% - количество периодов первого сигнала
KP2=5;% - количество периодов второго сигнала
N=1024;%количество точек расчета

```

```

for k=1:N % генерация сигнала и шума
s(k) = A*sin(2*pi*KP1*k/N);%+ A*sin(2*pi*KP2*k/N);
q(k)=Q*(randn(size(N))); %СКО шума равно Q
x(k)=s(k)+q(k); % суммирование сигнала и шума

```

```

end

S=x(1)+x(2)+x(3)+x(4)+x(5)+x(6)+x(7);
y(4)=S/7;
S1=s(1)+s(2)+s(3)+s(4)+s(5)+s(6)+s(7);
y1(4)=S1/7;
for i=1:N-7 %сглаживание зашумленного сигнала
    S=S-x(i)+x(i+7);
    y(i+4)=S/7;
end

for i=4:N-4
    DZ(i)=s(i)-y(i); %уровень зашумления в сигнале после фильтра
end
DZ=DZ*100/max(s); %остаточная погрешность после фильтрации
SKO=std(DZ)

i=4:N-4;
figure;
plot(x);
hold on;
title('Зашумленный сигнал до фильтра');
plot(i,y(4:N-4),'r-');
title('Сигнал после фильтра');
xlabel('Номер отсчета'); % подпись по оси X
axis tight;
hold off;
figure
plot(i,DZ(4:N-4)); %вывод погрешности отфильтрованного сигнала
title('Погрешность отфильтрованного сигнала');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Полная погрешность, %'); % подпись по оси Y
axis tight;
pause;
close all; %закрытие окон графического вывода
clear; %очистка Workspace

```

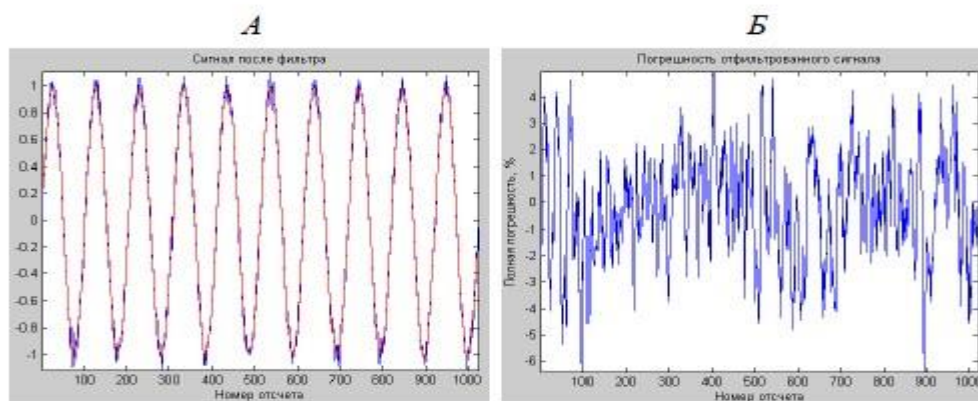


Рис. 3.6. Сигнал до и после сглаживания (А) и полная погрешность после сглаживания (Б).

Указания к выполнению работы

1. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала в сравнении с исходным благодаря накоплению постройте зависимость отношения сигнал/шум и разности исходного и обработанного сигналов от количества накоплений.

2. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала в сравнении с исходным благодаря использованию фильтров (НЧ, Баттерворта, БПФ-ОБПФ и др.) найдите оптимальную полосу пропускания фильтра, при которой эффект будет наибольшим с точки зрения улучшения отношения сигнал/шум и уменьшения полной погрешности обработанного сигнала отдельно.

3. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала в сравнении с исходным благодаря использованию фильтра скользящего среднего (Smoothing) определите зависимость степени подавления шумов от ширины окна (при размере окна 3, 5, 7 элементов) при различном уровне шумов и влияние параметров фильтра на изменения амплитуды и фазы сигнала на выходе фильтра.

4. При исследовании эффекта улучшения отношения сигнал/шум и уменьшения погрешности обработанного сигнала произведите сравнение эффективности исследуемых цифровых фильтров с оптимальным фильтром Колмогорова-Винера при обработке сигналов различной формы: гармонических, импульсных, колоколообразных при различных параметрах частоты и длительности.

Содержание отчета

1. Задание к работе.
2. Таблица исходных данных.
3. Тексты программ фильтрации с комментариями
4. Графики исходного и обработанного сигналов, график погрешности при различных параметрах обработки.
5. Выводы.

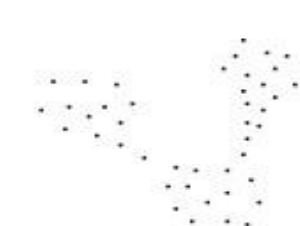
Лабораторная работа № 2

Кластеризация данных алгоритмом FOREL. Нахождения дискриминантной функции по прецедентам методом потенциальных функций. Построение гистограмм функций плотности распределения

Цель работы изучить алгоритм FOREL. Построение гистограмм функций плотности распределения

1. Считывание точечных данных из файла в матрицу A и формирование массива точек (X, Y)

$A := \text{READBMP}("D:/p3")$



A

A =

| | 0 | 1 | 2 | 3 |
|---|-----|-----|-----|-----|
| 0 | 255 | 255 | 255 | 255 |
| 1 | 255 | 255 | 255 | 255 |
| 2 | 255 | 255 | 255 | 255 |
| 3 | 255 | 255 | 255 | 255 |
| 4 | 255 | 255 | 255 | 255 |
| 5 | 255 | 255 | 255 | ... |

```

points(A) :=
  k ← 0
  for i ∈ 0..rows(A) - 1
    for j ∈ 0..cols(A) - 1
      if Ai,j = 0
        Xk ← i
        Yk ← j
        k ← k + 1
  for k ∈ 0..length(X) - 2
    if (Xk+1 - Xk)2 + (Yk - Yk+1)2 < 3
      Xk ← -1
      Yk ← -1
  s ← 0
  for k ∈ 0..length(X) - 1
    if Xk ≠ -1
      X1s ← Xk
      Y1s ← Yk
      s ← s + 1
  (X1)
  (Y1)

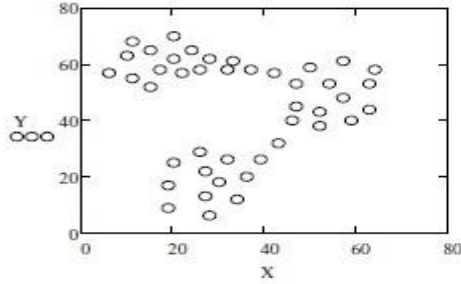
```

функция формирования из матрицы изображения точечных данных A массива точек (X, Y)

$$X := \text{points}(A)_0 \quad X^T = \begin{array}{c|cccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & 6 & 10 & 11 & 11 & 15 & 15 & 17 & 19 & 19 & \dots \end{array}$$

$$Y := \text{points}(A)_1 \quad Y^T = \begin{array}{c|cccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & 57 & 63 & 55 & 68 & 52 & 65 & 58 & 9 & 17 & \dots \end{array}$$

2. Функции алгоритма FOREL



$$d(C, S) := (C_0 - S_0)^2 + (C_1 - S_1)^2$$

$$\text{cent}(F, X, Y, r) := \begin{array}{l} C1 \leftarrow (0 \ 0)^T \\ n \leftarrow 0 \\ \text{for } i \in 0.. \text{length}(X) - 1 \\ \quad C_i \leftarrow (X_i \ Y_i)^T \\ \quad \text{if } d(C_i, F) < r^2 \\ \quad \quad n \leftarrow n + 1 \\ \quad \quad C1 \leftarrow C1 + C_i \\ \frac{C1}{n} \end{array}$$

функция определения центра масс точек массива (X, Y) , содержащихся внутри круга с центром в точке F и радиусом r

$$\text{ext}(F, X, Y, r) := \begin{array}{l} \text{for } i \in 0.. \text{length}(X) - 1 \\ \quad C_i \leftarrow (X_i \ Y_i)^T \\ \quad \text{if } d(C_i, F) > r^2 \\ \quad \quad C1 \leftarrow C_i \\ \quad \quad \text{break} \\ \text{for } i \in 0.. \text{length}(X) - 1 \\ \quad C_i \leftarrow (X_i \ Y_i)^T \\ \quad C1 \leftarrow \text{augmen}(C1, C_i) \text{ if } d(C_i, F) > r^2 \wedge C_i \neq C1 \\ C1 \end{array}$$

функция удаления из массива (X, Y) точек, содержащихся внутри круга с центром в точке F и радиусом r

$$\text{forel}(X, Y, r) := \begin{array}{l} F0 \leftarrow (X_0 \ Y_0)^T \\ F \leftarrow F0 \\ k \leftarrow \text{length}(X) \\ \text{while } k > 0 \\ \quad F1 \leftarrow \text{cent}(F0, X, Y, r) \\ \quad \text{while } d(F0, F1) > 0.1 \\ \quad \quad F0 \leftarrow F1 \\ \quad \quad F1 \leftarrow \text{cent}(F1, X, Y, r) \\ \quad F \leftarrow \text{augmen}(F, F1) \\ \quad X1 \leftarrow (\text{ext}(F1, X, Y, r)^T)^{(0)} \text{ if } \text{ext}(F1, X, Y, r) \neq 0 \\ \quad \quad \text{break otherwise} \\ \quad Y1 \leftarrow (\text{ext}(F1, X, Y, r)^T)^{(1)} \text{ if } \text{ext}(F1, X, Y, r) \neq 0 \\ \quad X \leftarrow X1 \\ \quad Y \leftarrow Y1 \\ \quad F0 \leftarrow (X_0 \ Y_0)^T \\ \quad k \leftarrow \text{length}(X) \\ F \leftarrow \text{submatrix}(F, 0, 1, 1, \text{cols}(F) - 1) \\ F \end{array}$$

функция вычисления формальных элементов массива точек (X, Y) алгоритмом FOREL с параметром r

3. Результаты работы алгоритма FOREL

$r := 10$

$F_{00} := \text{forel}(X, Y, r)$

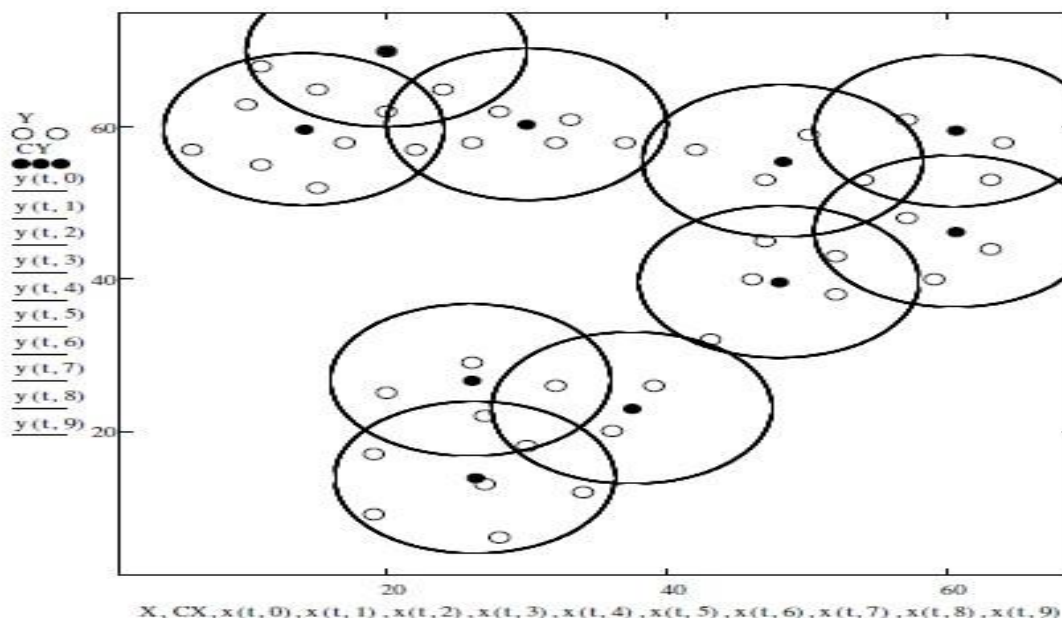
| | 0 | 1 | 2 | 3 | 4 |
|---|--------|--------|--------|--------|-----|
| 0 | 14.111 | 26.286 | 26 | 30 | 20 |
| 1 | 59.667 | 13.857 | 26.667 | 60.333 | ... |

$CX := (F^T)^{(0)}$

$CY := (F^T)^{(1)}$

$x(t, i) := F_{0,i} + r \cdot \cos(t)$

$y(t, i) := F_{1,i} + r \cdot \sin(t)$



Приложение 2. Нахождения дискриминантной функции по прецедентам методом потенциальных функций

1. Формирование множества прецедентов по изображению точечных данных

1.1. Считывание точечных данных из файла

$A := \text{READBMP}('f:\text{выборка_put_2}')$

1.2. Выделение точек из матрицы изображения, формирование массива точечных данных X

```

quan(A, a, b) :=
  s ← 0
  for i ∈ 0..rows(A) - 1
    for j ∈ 0..cols(A) - 1
      if (a ≤ Ai,j) · (Ai,j ≤ b)
        X(s) ← (i j)T
        s ← s + 1
  X

```

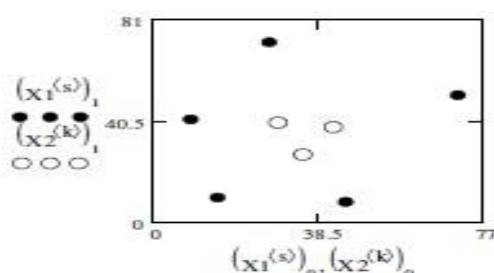
$X1 := \text{quan}(A, 0, 50)$ $X1 = \begin{pmatrix} 9 & 15 & 27 & 45 & 71 \\ 41 & 10 & 72 & 8 & 51 \end{pmatrix}$

- множество векторов первого класса

$X2 := \text{quan}(A, 50, 200)$ $X2 = \begin{pmatrix} 29 & 35 & 42 \\ 40 & 27 & 38 \end{pmatrix}$

- множество векторов второго класса

$s := 0..cols(X1) - 1$ $k := 0..cols(X2) - 1$



- изображение точечных данных массива X

1.3. Определение множества прецедентов

```

prec(A, b) :=
  s ← 0
  for i ∈ 0..rows(A) - 1
    for j ∈ 0..cols(A) - 1
      if Ai,j ≠ 255
        
$$\begin{cases} X^{(s)} \leftarrow (i \ j \ 1)^T & \text{if } A_{i,j} \leq b \\ X^{(s)} \leftarrow (i \ j \ -1)^T & \text{otherwise} \end{cases}$$

        s ← s + 1
        continue
  X

X := prec(A, 40)

$$X = \begin{pmatrix} 9 & 15 & 27 & 29 & 35 & 42 & 45 & 71 \\ 41 & 10 & 72 & 40 & 27 & 38 & 8 & 51 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \end{pmatrix}$$


```

- множество прецедентов

2. Вычисление дискриминантной функции методом потенциальных функций

2.1. Задание метрики и потенциальной функции

$$d(x, y) := \sqrt{(x_0 - y_0)^2 + (x_1 - y_1)^2} \quad u(x, y) := \frac{1}{1 + d(x, y)^2}$$

2.2. Вычисление дискриминантной функции

```

potentia(X) :=
  p ← 0
  for j ∈ 0..cols(X) - 1
    aj ← 0
  while p < cols(X)
    p ← 0
    for k ∈ -1..cols(X) - 2
      if 
$$\left[ X_{2, k+1} \cdot \sum_{j=0}^{\text{cols}(X)-1} a_j \cdot u \left[ \begin{pmatrix} X_{0, k+1} \\ X_{1, k+1} \end{pmatrix}, \begin{pmatrix} X_{0, j} \\ X_{1, j} \end{pmatrix} \right] \right] \leq 0$$

        ak+1 ← ak+1 + 1 · X2, k+1
        p ← 0
    p ← p + 1 otherwise
  a

```

$a := \text{potential}(X)$

$a^T = (1 \ 0 \ 0 \ -1 \ 0 \ 0 \ 1 \ 0)$ - вектор коэффициентов решающей функции

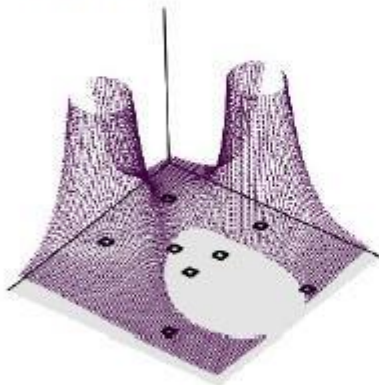
$D(y) := \sum_{j=0}^{\text{cols}(X)-1} a_j \cdot u \left[y, \begin{pmatrix} X_{0,j} \\ X_{1,j} \end{pmatrix} \right]$ - решающая функция

2.3. Визуализация решающей функции и предельентов

$i := 0..\text{cols}(A) - 1 \quad j := 0..\text{rows}(A) - 1$

$D_{i,j} := \begin{cases} D\left(\begin{pmatrix} i \\ j \end{pmatrix}\right) & \text{if } D\left(\begin{pmatrix} i \\ j \end{pmatrix}\right) \geq 0 \\ 0 & \text{otherwise} \end{cases}$

$B_{i,j} := \begin{cases} 0.000001 & \text{if } \sum_{s=0}^{\text{cols}(X)-1} \left[(X_{0,s}-i)^2 + (X_{1,s}-j)^2 \leq 3 \right] \\ 0 & \text{otherwise} \end{cases}$



D, B

3. Вычисление дискриминантной функции путем представления потенциальной с помощью системы базисных функций

3.1. Задание системы базисных функций

$H(x, i, j) := \text{Leg}(i, x_0) \cdot \text{Leg}(j, x_1)$ - многочлены Лежандра

3.2. Нормирование исходных данных

$x_0 := (x^T)^{(0)} \quad x_1 := (x^T)^{(1)}$

$$X0^T = (9 \ 15 \ 27 \ 29 \ 35 \ 42 \ 45 \ 71) \quad \max0 := \max(X0) \quad \min0 := \min(X0)$$

$$X1^T = (41 \ 10 \ 72 \ 40 \ 27 \ 38 \ 8 \ 51) \quad \max1 := \max(X1) \quad \min1 := \min(X1)$$

$$\text{normal}(X) := \begin{array}{|l} \text{for } k \in 0.. \text{cols}(X) - 1 \\ \quad \left| \begin{array}{l} XX_{0,k} \leftarrow \frac{(X_{0,k} - \min0)}{\max0 - \min0} \\ XX_{1,k} \leftarrow \frac{(X_{1,k} - \min1)}{(\max1 - \min1)} \\ XX_{2,k} \leftarrow X_{2,k} \end{array} \right. \\ \quad XX \end{array}$$

$$XX := \text{normal}(X)$$

$$XX = \begin{pmatrix} 0 & 0.097 & 0.29 & 0.323 & 0.419 & 0.532 & 0.581 & 1 \\ 0.516 & 0.031 & 1 & 0.5 & 0.297 & 0.469 & 0 & 0.672 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \end{pmatrix} \quad \begin{array}{l} \text{- нормированное} \\ \text{множество пределов} \end{array}$$

3.3. Вычисление дискриминантной функции (m - порядок аппроксимации)

$$\text{potential2}(X, m) := \begin{array}{|l} p \leftarrow 0 \\ \text{for } i \in 0..m-1 \\ \quad \text{for } j \in 0..m-1 \\ \quad \quad c_{i,j} \leftarrow 0 \\ \text{while } p < \text{cols}(X) \\ \quad \left| \begin{array}{l} p \leftarrow 0 \\ \text{for } k \in -1.. \text{cols}(X) - 2 \\ \quad \left| \begin{array}{l} \text{if } \left[X_{2,k+1} \cdot \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} c_{i,j} \cdot H \left[\begin{pmatrix} X_{0,k+1} \\ X_{1,k+1} \end{pmatrix}, i, j \end{pmatrix} \right] \leq 0 \\ \quad \left| \begin{array}{l} \text{for } i \in 0..m-1 \\ \quad \text{for } j \in 0..m-1 \\ \quad \quad \left| \begin{array}{l} X_{2,k+1} \cdot H \left[\begin{pmatrix} X_{0,k+1} \\ X_{1,k+1} \end{pmatrix}, i, j \end{pmatrix} \right] \\ c_{i,j} \leftarrow c_{i,j} + \frac{\quad}{(i+1) \cdot (j+1)} \end{array} \right. \\ \quad \quad p \leftarrow 0 \\ \quad \quad p \leftarrow p + 1 \text{ otherwise} \end{array} \right. \\ \quad \quad c \end{array} \right. \end{array}$$

$m := 3$

$c := \text{potential}(XX, m)$

$$c^T = \begin{pmatrix} 3 & -2.056 & 3.065 \\ -2.023 & -1.773 & 0.464 \\ 1.756 & 0.09 & -0.478 \end{pmatrix}$$

- матрица
коэффициентов
решающей функции

$$DN(y) := \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} c_{i,j} H(y, i, j)$$

- нормированная
решающая функция

$$D2(y) := DN \left[\begin{pmatrix} \frac{y_0 - \min 0}{\max 0 - \min 0} \\ \frac{y_1 - \min 1}{(\max 1 - \min 1)} \end{pmatrix} \right]$$

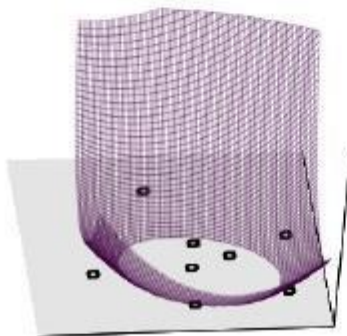
- денормированная
решающая функция

3.4. Визуализация решающей функции и прецедентов

$i := 0..cols(A) - 1 \quad j := 0..rows(A) - 1$

$$D2_{i,j} := \begin{cases} D2\left(\begin{pmatrix} i \\ j \end{pmatrix}\right) & \text{if } D2\left(\begin{pmatrix} i \\ j \end{pmatrix}\right) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,j} := \begin{cases} 0.000001 \text{ if } \sum_{s=0}^{cols(X)-1} \left[(X_{0,s} - i)^2 + (X_{1,s} - j)^2 \leq 3 \right] \\ 0 & \text{otherwise} \end{cases}$$



D2, B

Приложение 4. Построение гистограмм функций плотности распределения

1. Простейшая гистограмма с регулярными ячейками

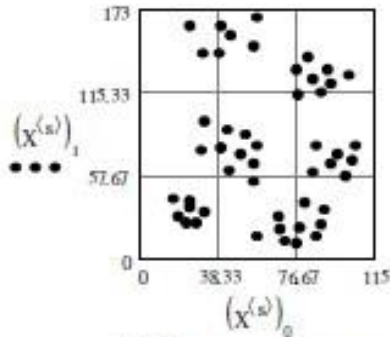
1.1. Считывание точечных данных из файла и формирование вектора данных X

$A := \text{READBMP}('D:/\text{выборка}')$

```
quant(A) :=
  s ← 0
  for i ∈ 0..rows(A) - 1
    for j ∈ 0..cols(A) - 1
      if Ai,j = 0
        X(s) ← (i j)T
        s ← s + 1
  X
```

$X := \text{quant}(A)$

$s := 0..cols(X) - 1$



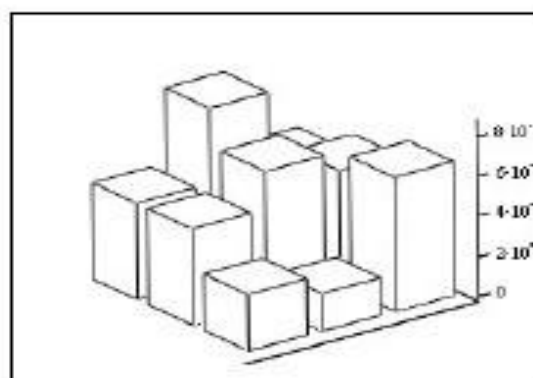
$X^T =$

| | 0 | 1 |
|----|----|-----|
| 0 | 16 | 42 |
| 1 | 18 | 30 |
| 2 | 22 | 25 |
| 3 | 24 | 36 |
| 4 | 24 | 41 |
| 5 | 24 | 162 |
| 6 | 27 | 25 |
| 7 | 29 | 75 |
| 8 | 30 | 142 |
| 9 | 31 | 33 |
| 10 | 31 | 96 |
| 11 | 38 | 142 |
| 12 | 39 | 77 |
| 13 | 39 | 162 |
| 14 | 42 | 89 |
| 15 | 43 | 61 |
| 16 | 44 | 155 |
| 17 | 49 | 73 |
| 18 | 51 | 86 |
| 19 | 55 | 54 |

1.2. Вычисление гистограммы с $m \times n$ регулярными ячейками

```
his(X, m, n) :=
  for i ∈ 0..m - 1
    for j ∈ 0..n - 1
      Hi,j ← 0
  for i ∈ 0..m - 1
    for j ∈ 0..n - 1
      for s ∈ 0..cols(X) - 1
        R1 ← [ rows(A)·i / m ≤ (X(s))0 ] · [ (X(s))0 < rows(A)·(i+1) / m ]
        R2 ← [ cols(A)·j / n ≤ (X(s))1 ] · [ (X(s))1 < cols(A)·(j+1) / n ]
        Hi,j ← Hi,j + 1 if R1·R2
  H · m·n / ((cols(X) - 1)·(rows(A) - 1)·(cols(A) - 1))
```

$H := \text{hist}(X, 3, 3)$



$$H = \begin{pmatrix} 6.694 \times 10^{-5} & 1.912 \times 10^{-5} & 2.869 \times 10^{-5} \\ 5.737 \times 10^{-5} & 6.694 \times 10^{-5} & 4.781 \times 10^{-5} \\ 4.781 \times 10^{-5} & 8.606 \times 10^{-5} & 4.781 \times 10^{-5} \end{pmatrix}$$

H

2. Вычисление гистограммы с нерегулярными ячейками

2.1. Определение взвешенной евклидовой d и евклидовой dc метрик

$$d(x, B, k) := \frac{(x_0 - B_{0,k})^2}{B_{2,k}} + \frac{(x_1 - B_{1,k})^2}{B_{3,k}} \quad dc(x, y) := (x_0 - y_0)^2 + (x_1 - y_1)^2$$

2.2. Определение квадрата расстояния от точки x до границы матрицы изображения A

$$db(x, A) := (\min(x_0 \ x_1 \ \text{rows}(A) - x_0 \ \text{cols}(A) - x_1))^2$$

2.3. Функция распределения основного массива данных по ячейкам

Входные данные

массив X - в каждом i-м столбце - координаты i-го вектора данных;

s0 - значение минимальной дисперсии;

h1 - максимальное расстояние во взвешенной метрике от текущего центра до вектора данных, при котором он будет отнесен к данному классу;

h2 - минимальное расстояние во взвешенной метрике от вектора данных до всех центров классов, при котором вектор станет центром нового класса.

Выходные данные

массив B - массив классов, в каждом k-м столбце записаны координаты центра k-го класса, по координатные дисперсии и количество элементов в классе;

массив X - в каждом i-м столбце - координаты i-го вектора данных и номер класса, к которому он относится.


```

distr(X,s0,h1,h2) :=
  X2,0 ← 1
  B(0) ← (X0,0 X1,0 s0 s0 1)T
  for i ∈ 1..cols(X) - 1
    s ← 0
    p ← 0
    while s ≤ cols(B) - 1
      (m0 m1 s1 s2 N) ← B(s)T
      if d(X(i), B, s) < h1
        X2,i ← s + 1
        B0,s ←  $\frac{1}{N+1} \cdot (N \cdot m0 + X_{0,i})$ 
        B1,s ←  $\frac{1}{N+1} \cdot (N \cdot m1 + X_{1,i})$ 
        B2,s ←  $\max \left[ s0 \cdot \frac{(N-1) \cdot s1}{N} + \frac{1}{N+1} \cdot (m0 - X_{0,i})^2 \right]$ 
        B3,s ←  $\max \left[ s0 \cdot \frac{(N-1) \cdot s2}{N} + \frac{1}{N+1} \cdot (m1 - X_{1,i})^2 \right]$ 
        B4,s ← N + 1
        break
      p ← p + 1 if d(X(i), B, s) > h2
      s ← s + 1
    if p = cols(B)
      B ← augment(B, (X0,i X1,i s0 s0 1)T)
      X2,i ← s + 1
  (B X)T

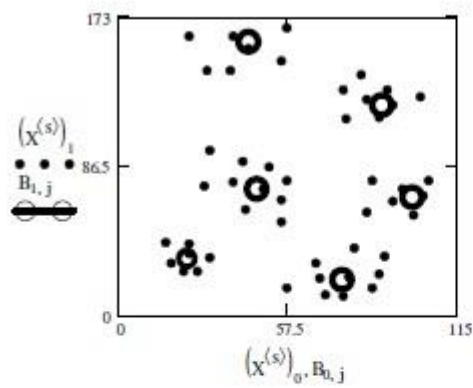
```

DISTR := distr(X,30,10,20)

B := DISTR₀ X := DISTR₁

$$B = \begin{pmatrix} 23.143 & 43.8 & 46.667 & 75.556 & 89.167 & 99.4 \\ 33.143 & 158.8 & 73.333 & 20.667 & 122.333 & 68.8 \\ 37.343 & 178.7 & 84.5 & 123.028 & 72.867 & 33.133 \\ 48.476 & 62.2 & 129.75 & 61.25 & 45.467 & 58.075 \\ 7 & 5 & 9 & 9 & 6 & 5 \end{pmatrix}$$

- массив центров
кластеров, дисперсий и
количества элементов в
кластерах

$$X = \begin{array}{c|cccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & 16 & 18 & 22 & 24 & 24 & 24 & 27 & 29 & 30 & 31 \\ 1 & 42 & 30 & 25 & 36 & 41 & 162 & 25 & 75 & 142 & 33 \\ 2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 3 & 0 & 1 \end{array}$$


2.4. Функция распределения оставшихся нераспределенных элементов по классам методом ближайшего соседа

```
dopdist(B, X) =
  for s ∈ 0..cols(X) - 1
    if X2,s = 0
      for j ∈ 0..cols(B) - 1
        mdj ← 100000000
        for i ∈ 0..cols(X) - 1
          R ← (mdj > dc(X(i), X(s))) · (X2,i = j + 1)
          mdj ← dc(X(i), X(s)) if R
        mn ← 100000000
        for r ∈ 0..cols(B) - 1
          if mdj < mn
            mn ← mdj
            k ← r
        X2,s ← k + 1
        B4,k ← B4,k + 1
  (B)
  (X)
```

$D := \text{dopdistr}(B, X)$

$B := D_0 \quad X := D_1$

$B = \begin{pmatrix} 23.143 & 43.8 & 46.667 & 75.556 & 89.167 & 99.4 \\ 33.143 & 158.8 & 73.333 & 20.667 & 122.333 & 68.8 \\ 37.343 & 178.7 & 84.5 & 123.028 & 72.867 & 33.133 \\ 48.476 & 62.2 & 129.75 & 61.25 & 45.467 & 58.075 \\ 7 & 7 & 10 & 10 & 8 & 7 \end{pmatrix}$

- итоговый массив
центров кластеров,
дисперсий и
количества элементов в
кластерах

$X = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 0 & 16 & 18 & 22 & 24 & 24 & 24 & 27 & 29 \\ 1 & 42 & 30 & 25 & 36 & 41 & 162 & 25 & 75 \\ 2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 3 \\ \hline \end{array}$

- массив элементов с
указанием номера
кластера, к которому
элемент относится

2.5. Функция построения областей Дирикле и вычисления их площадей

Входные данные:

массив X - массив векторов данных,

массив B - массив классов,

массив A - массив с исходной картинкой

Выходные данные:

массив B - массив классов, в который добавлена строка с площадями областей

Дирикле, соответствующих своим классам;

массив AA - массив, в каждой точке которого записан номер класса, к которому эта точка принадлежит.


```

square(A, B, X) :=
  for k ∈ 0..cols(B) - 1
    B5, k ← 0
    for i ∈ 0..rows(A) - 1
      for j ∈ 0..cols(A) - 1
        for k ∈ 0..cols(B) - 1
          mk ← 10000000
          for s ∈ 0..cols(X) - 1
            R ← [dc[X(s), (i, j)] < mk] · (X2, s = k + 1)
            mk ← dc[X(s), (i, j)] if R
          md ← 10000000
          for k ∈ 0..cols(B) - 1
            if mk < md
              md ← mk
              p ← k + 1
          AAi, j ← 0 if db[(i, j), A] < md
          otherwise
            AAi, j ← p
            B5, p-1 ← B5, p-1 + 1
  (B AA)T

```

$G := \text{square}(A, B, X) \quad B := G_0 \quad AA := G_1$

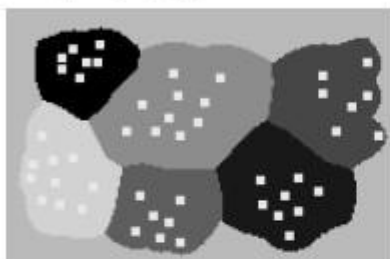
Итоговый массив центров кластеров с добавленной строкой площадей областей Дирихле

$$B = \begin{pmatrix} 23.143 & 43.8 & 46.667 & 75.556 & 89.167 & 99.4 \\ 33.143 & 158.8 & 73.333 & 20.667 & 122.333 & 68.8 \\ 37.343 & 178.7 & 84.5 & 123.028 & 72.867 & 33.133 \\ 48.476 & 62.2 & 129.75 & 61.25 & 45.467 & 58.075 \\ 7 & 7 & 10 & 10 & 8 & 7 \\ 1.403 \times 10^3 & 2.314 \times 10^3 & 3.54 \times 10^3 & 2.067 \times 10^3 & 2.484 \times 10^3 & 1.704 \times 10^3 \end{pmatrix}$$

2.6. Функция "нанесения" точек на изображение областей Дирихле

```
point(AA,X,p) :=
  for s ∈ 0..cols(X) - 1
  for k ∈ 0..2·p - 1
  for l ∈ 0..2·p - 1
    AAX0,s-p+k,X1,s-p+l ← 253
  AA
```

AX := point(AA,X,2)



(AX + 255)·70

2.7. Функция вычисления адаптивной гистограммы

```
hist(AA,B,X) :=
  for k ∈ 0..cols(B) - 1
  for i ∈ 0..rows(A) - 1
  for j ∈ 0..cols(A) - 1
    AAi,j ←  $\frac{B_{4,k}}{\text{cols}(X) \cdot B_{5,k}}$  if AAi,j = k + 1
  AA
```

AA := hist(AA,B,X)



AA

Лабораторная работа № 3

Подгонка кривых

Цель работы является освоение методики получения адекватного аналитического описания зависимости $Y=f(X)$, по данным, содержащим случайные погрешности..

Теоретические основы

Обработка результатов измерений, содержащих случайные погрешности, включает 3 этапа:

1. Выбор вида аппроксимирующей зависимости.
2. Расчет коэффициентов аппроксимирующей зависимости.
3. Проверку адекватности полученной зависимости экспериментальным данным.

Наиболее совершенным программным средством для решения данной задачи является пакет программ MATLAB. MATLAB является средой разработки программ обработки данных и, одновременно, содержит большое количество готовых программ, в частности, программу подгонки кривых CurveFitting.

Программа подгонки кривых CurveFitting предоставляет следующие виды функций для аппроксимации и интерполяции одномерных массивов данных: а) экспоненциальную;

$$y = ae^{bx}$$

$$y = ae^{bx} + ce^{dx}$$

б) гауссиан

$$y = \sum_{i=1}^n a_i e^{\left[-\left(\frac{x-b_i}{c_i}\right)^2\right]}$$

$$n=1,2,\dots,10$$

в)Фурье

$$y = a_0 + \sum_{i=1}^n a_i \cos(nwx) + b_i \sin(nwx)$$
$$n=1,2,\dots,10$$

г)полиномиальную

$$y = \sum_{i=1}^{n+1} p_i x^{n+1-i}$$
$$n=1,2,\dots,10$$

д)показательную

$$y = ax^b$$
$$y = a + bx^c$$

е)рациональную

$$y = \frac{\sum_{i=1}^{n+1} p_i x^{n+1-i}}{x^m + \sum_{i=1}^m q_i x^{m-i}}$$
$$n, m=1,2,\dots,10$$

Критерием качества подгонки являются вычисляемые программой коэффициенты множественной детерминации (R-square и Adjusted R-square). Коэффициент R-square показывает насколько разброс данных относительно аппроксимирующей кривой может быть объяснен наличием случайных погрешностей в данных. Максимально возможное значение R-square равно 1.

Коэффициент Adjusted R-square характеризует то же самое, но учитывает количество степеней свободы дисперсий воспроизводимости и адекватности. Поэтому Adjusted R-square лучше характеризует степень соответствия экспериментальных данных аппроксимирующей кривой в том случае, если мы увеличиваем степень аппроксимирующего полинома и хотим проверить, происходит ли при этом улучшение качества подгонки. Максимально возможное значение Adjusted R-square также равно 1.

$$SSE = \sum_{i=1}^n w_i (y_i - \hat{y}_i)^2 \quad \text{SSE – sum square error}$$

$$SST = \sum_{i=1}^n w_i (y_i - \bar{y})^2 \quad \text{SST – sum square total,}$$

$$w_i = \frac{1}{\sigma_i^2} \quad \text{- коэффициент, выравнивающий степень разброса данных в опытах}$$

\hat{y} – значение полученное по аппроксимирующей зависимости

$$R - square = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$Adjusted R - square = 1 - \frac{SSE(n-1)}{SST(n-m)}$$

n – количество результатов измерений;

m – количество членов в уравнении регрессии;

$n-1$ – количество степеней свободы дисперсии SSE;

$n-m$ – количество степеней свободы дисперсии SST.

Программа работы

6. Освоить технологию работы с программой CurveFitting пакета MATLAB.
7. Научиться производить обоснованный выбор наилучшей аппроксимирующей зависимости из возможных.

Указания к выполнению

1. При выполнении п.1. Программы изучите теоретические основы подгонки кривых, положенные в основу программы Curve Fitting и описание библиотеки аппроксимирующих функций Fitting Library в MATLAB.
2. При выполнении п.2 выполните подгонку кривых, выберите наилучшую функциональную зависимость и обоснуйте сделанный выбор для зашумленного массива данных: файл censuN.mat
3. При работе с программой CurveFitting подгонку кривых производите в следующей последовательности:
 - а) Запустите MATLAB;
 - б) откройте окно Workspace (View/Workspace) и загрузите в это окно mat-файл данных по заданию преподавателя;
 - в) откройте панель CurveFitting для чего выберите Start/ Toolboxes/ CurveFitting в окне MATLAB;
 - г) на панели CurveFittingTool активизируйте кнопку Data. В окнах Xdata, Ydata станут доступными составляющие cdate (X) и pop (Y) из файла census.mat;
 - д) произведите выбор cdate и pop, затем активизируйте кнопку CreateDataSet;
 - е) активизируйте кнопку Fitting;
 - ж) выберите вид аппроксимирующей функции и активизируйте кнопку Apply;

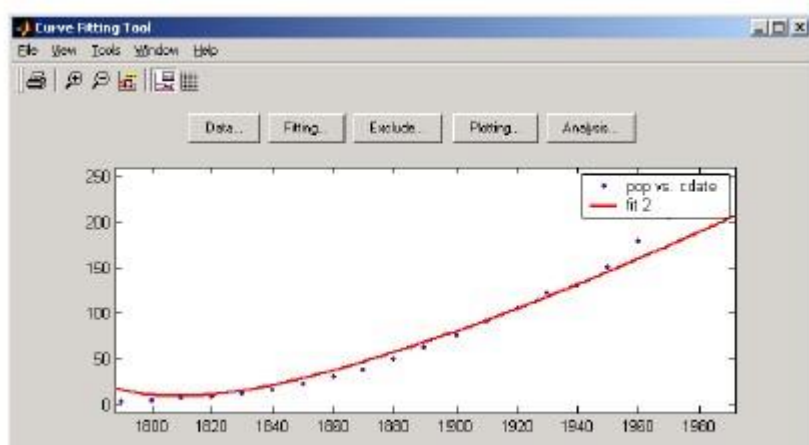


Рис. 6.1

Результаты аппроксимации будут представлены в графической форме в окне графического вывода на панели CurveFittingTool (рис.6.1) и в численной форме в окне Results на панели Tables of Fits (рис. 6.2).

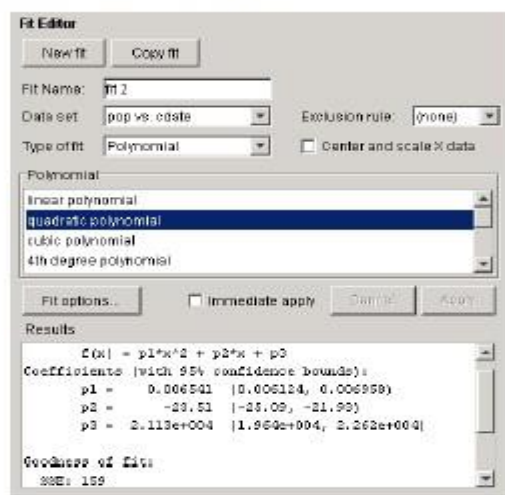


Рис. 6.2

з)выберите пункт меню View/Residuals/Line или View/Residuals/Scatter. В результате в окне графического вывода на панели CurveFittingTool будут одновременно выведен график ошибки интерполяции (см. пример на рис. 6.3).

Окно графического вывода можно вывести на печать или выполнить в виде рисунка: File/Print to Figure.

Полученные в процессе аппроксимации результаты занесите в таблицу:

Табл. 6.1.

| Вид модели | Порядок модели | Качество подгонки (+ / -) | Диапазон погрешностей | R-square | Adjusted R-square | RMSE |
|------------|----------------|---------------------------|-----------------------|----------|-------------------|------|
| | | | | | | |

Исследуйте таким образом все возможные виды моделей, порядок модели - от 1 до 5.

е) по полученным и отраженным в таблице данным выберите три наилучших варианта аппроксимирующих функций. Для этих вариантов сохраните график аппроксимирующей функции и график погрешности аппроксимации, аналитические выражения с численными значениями коэффициентов для аппроксимирующей зависимости.

4. Оценку качества подгонки кривых и окончательный выбор аппроксимирующей зависимости произведите последовательно:

1. Выберите «хорошие» модели.

1.1. Визуально, по степени согласованности графика полученной в результате подгонки аппроксимирующей кривой с отображенными в этом же окне графического вывода значениями y_i . Качество подгонки считается удовлетворительным, если значения y_i «вытянуты» вдоль аппроксимирующей кривой. На рис.6.3 приведены результаты подгонки, которые можно считать неудовлетворительными (А) и удовлетворительными (Б).

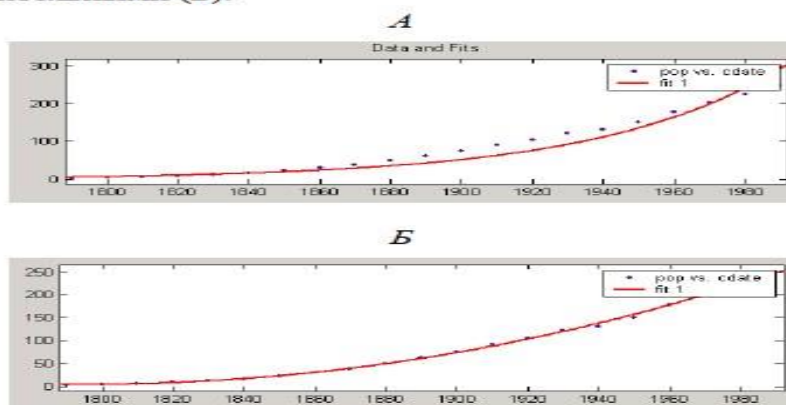


Рис. 6.3.

1.2. Визуально, по графику разности между значениями y_i и вычисленными по аппроксимирующей функции значениями $f(x_i)$. Качество подгонки считается удовлетворительным, если график разности хорошо аппроксимируется «на глаз» прямой $y=0$. На рис.6.4 приведены результаты подгонки, которые можно считать неудовлетворительными (А) и удовлетворительными (Б).

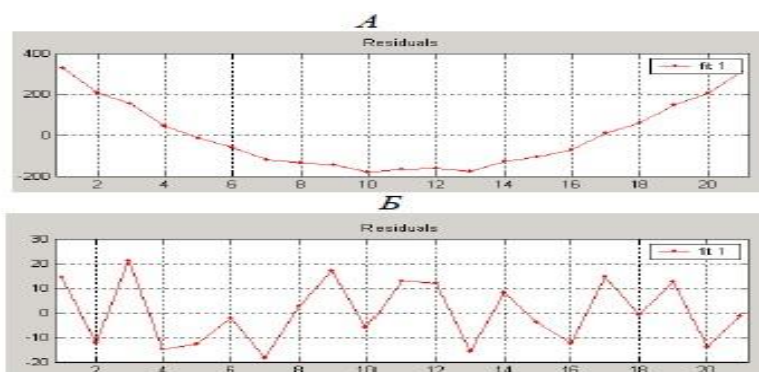


Рис. 4.

2. Выберите лучшую кривую в каждом классе аппроксимирующих зависимостей – по показателю Adjusted R-square. Лучшей кривой в классе является та, которой соответствует максимальное значение Adjusted R-square. Если максимальное значение соответствует двум или более кривым – выбирается кривая с наиболее простым аналитическим описанием.
3. Окончательный выбор самой лучшей аппроксимирующей зависимости – выбрать самую простую модель из лучших с условием примерно тех же значений диапазона погрешностей, R-square и RMSE (Root Mean Squared Error).

Содержание отчета

1. Постановка задачи.
2. Результаты аппроксимации экспериментальных данных из файла и показатели качества подгонки R-square и Adjusted R-square для всех основных классов аппроксимирующих функций в форме таблицы.
3. Для трех наиболее подходящих аппроксимирующих зависимостей – график аппроксимирующей функции и график погрешности аппроксимации, аналитические выражения для аппроксимирующей зависимости.
4. Обоснование выбора наилучшей аппроксимирующей зависимости.
5. Выводы.

Лабораторная работа № 4

Частотно-временной анализ нестационарных сигналов методом Гильберта-Хуанга

Цель работы

изучение методики частотно-временного анализа нестационарных сигналов на основе эмпирического метода декомпозиции Хуанга и частотно-временного анализа на основе преобразования Гильберта.

Задание к работе

Имеется модельный сигнал, представляющий собой зашумленный гармонический сигнал с частотой, изменяющейся по линейному, квадратичному, кубическому закону или сумму двух зашумленных гармонических сигналов с изменяющейся частотой. Требуется исследовать условия их адекватного частотно-временного представления.

Теоретические основы

Под преобразованием Гильберта-Хуанга (Hilbert-Huang transform – ННТ) понимается эмпирический метод декомпозиции (EMD) нелинейных и нестационарных процессов и Гильбертов спектральный анализ (HSA). ННТ представляет собой частотно-временной анализ данных (сигналов) и не требует априорного функционального базиса. Функции базиса получают адаптивно непосредственно из данных процедурами отсеивания функций «эмпирических мод». Мгновенные частоты вычисляются от производных фазовых функций Гильбертовым преобразованием функций базиса. Заключительный результат представляется в частотновременном пространстве

EMD-HSA был предложен Норденом Хуангом в 1995 в США (NASA) для изучения поверхностных волн тайфунов, с обобщением на анализ произвольных временных рядов коллективом соавторов в 1998 г. В последующие годы, по мере расширения применения EMDHSA для других отраслей науки и техники, вместо термина EMD-HSA был принят более короткий термин преобразования: ННТ.

EMD (Empirical Mode Decomposition) - метод разложения сигналов на функции, которые получили название «эмпирических мод». Метод представляет собой итерационную вычислительную процедуру разложения исходных данных (непрерывных или дискретных сигналов) на эмпирические моды или внутренние колебания (intrinsic mode functions, IMF).

IMF представляет собой колебательный режим, как часть простой гармонической функции, но вместо постоянной амплитуды и частоты, как в простой гармонике, у IMF могут быть переменная амплитуда и частота, как функции времени.

IMF, определенные вышеприведенным способом, допускают вычисление физически значимых мгновенных частот, что дает возможность создать частотно-временное представление сигнала на основе преобразования Гильберта.

Преобразование Гильберта позволяет разложить исходный процесс на две составляющие: амплитудную и фазовую. Амплитуда не меняется во времени, а фаза линейно растет с коэффициентом пропорциональности, который носит название частоты. Частота определяет число периодов (повторений значений функции) в единицу времени и является постоянной во времени величиной.

Амплитуда и фаза гармонической функции отражают различную информацию: амплитуда описывает энергию, а фаза характеризует повторяемость процесса во времени, и в этом смысле они могут рассматриваться как независимые характеристики гармонической функции. Поскольку фаза зависит от времени произвольным образом, то вводится понятие мгновенной частоты.

Где ϕ – это фаза процесса.

Преобразование Гильберта действительной функции $x(t)$, определенной во всей временной области $-\infty < t < \infty$, есть действительная функция $\tilde{x}(t)$, также определенная на всей временной оси и задаваемая формулой:

$$\tilde{x}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau$$

Применение преобразования Гильберта позволяет выполнять квадратурную модуляцию сигналов, в каждой текущей координате модулированных сигналов производить определение огибающей и мгновенной фазы и частоты сигналов, выполнять анализ каузальных систем обработки сигналов.

Декомпозиция Хуанга

Декомпозиция основана на предположении, что любые данные состоят из различных внутренних колебаний. В любой момент времени данные могут иметь различные сосуществующие внутренние колебания, нанесенные одно на другое. Каждое колебание, линейное или нелинейное, представляет собой модовую функцию, которая имеет экстремумы и нулевые пересечения. Кроме того, колебание будет в определенной степени «симметрично» относительно локального среднего значения. Результат – конечные сложные данные.

Допустим, что имеется произвольный сигнал $x(t)$. Сущность метода EMD заключается в последовательном вычислении функций эмпирических мод $c_j(t)$ и остатков $r_j(t) = r_{j-1}(t) - c_j(t)$, где $j = 1, 2, 3, \dots, n$ при $r_0 = x(t)$. Результатом разложения будет представление сигнала в виде суммы модовых функций и конечного остатка.

Входной сигнал $x(t)$ раскладывается по базису, который удовлетворяет всем традиционным требованиям базиса. На основании проверки на модельных и опытных данных он является:

- законченным,
- сходящимся,
- ортогональным,
- единственным.

Также он является адаптивным, так как получен непосредственно из анализируемых данных эмпирическим методом.

Алгоритм эмпирической декомпозиции сигнала⁴

Операция 1. Идентифицируем по координатам и амплитудам все локальные экстремумы (максимумы и минимумы) сигнала. Группируем отдельно массивы векторов координат (номеров отсчетов) $x_{\max}(k)$ и соответствующих амплитудных значений $y_{\max}(k)$ максимумов, и аналогичные массивы векторов $x_{\min}(k)$ и $y_{\min}(k)$ минимумов всех выделенных экстремумов.

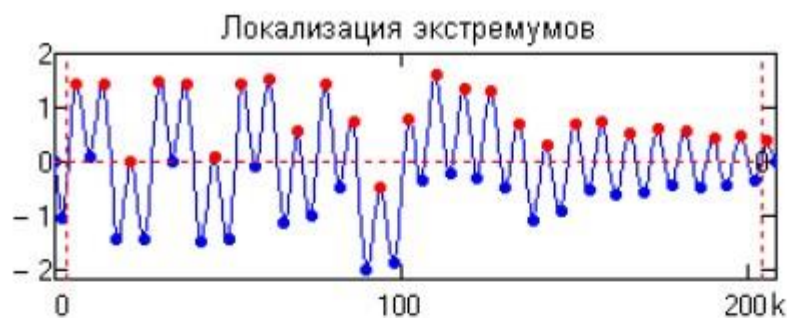


Рис. 7.1

Операция 2. Кубическим (или каким либо другим) сплайном вычисляем верхнюю и нижнюю огибающие сигнала по выделенным максимумам и минимумам, как это показано на рисунке (красный и синий цвет соответственно). Определяем функцию средних значений $m_1(k)$ между огибающими (черный цвет) и находим первое приближение к первой функции IMF:

$$h_1(k) = y(k) - m_1(k).$$

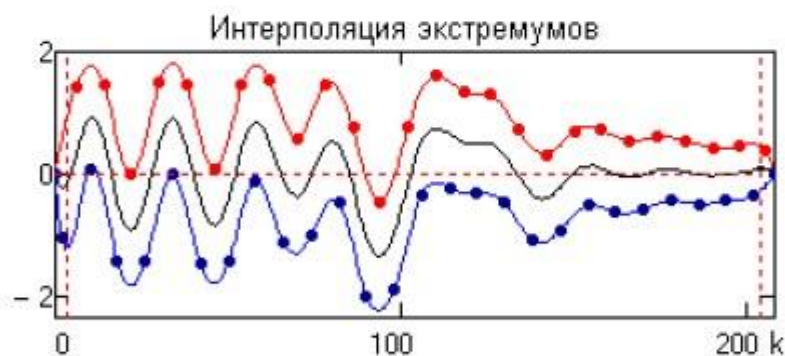


Рис. 7.2.

Операция 3. Повторяем операции 1 и 2, принимая вместо $y(k)$ функцию $h_1(k)$, и находим второе приближение к первой функции IMF – функцию $h_2(k)$.

$$h_2(k) = h_1(k) - m_2(k).$$

Аналогично находим третье и последующие приближения к первой функции IMF. По мере увеличения количества итераций функция $m_i(k)$, равно как и функция $h_i(k)$, стремится к неизменяемой форме. С учетом этого, естественным критерием останова итераций является задание определенного предела по нормализованной квадратичной разности между двумя последовательными операциями приближения, определяемой как

$$d = \frac{\sum_k |h_i(k) - h_{i-1}(k)|}{\sum_k h_{i-1}^2(k)}$$

Результат разложения

Последнее значение $h_i(k)$ итераций принимается за наиболее высокочастотную функцию $c_1(k) = h_i(k)$ семейства IMF, которая непосредственно входит в состав исходного сигнала $y(k)$. Это позволяет вычесть $c_1(k)$ из состава сигнала и оставить в нем более низкочастотные составляющие:

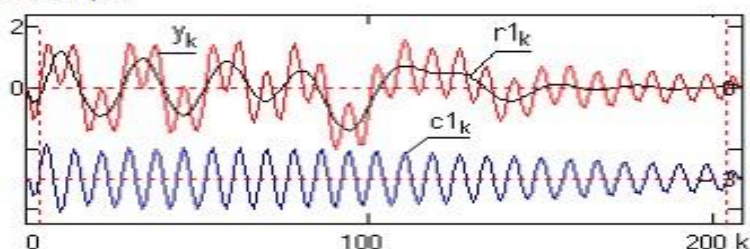


Рис. 7.3.

$$r_1(k) = y(k) - c_1(k).$$

Функция $r_1(k)$ обрабатывается как новые данные по аналогичной методике с нахождением второй функции IMF – $c_2(k)$, после чего процесс продолжается:

$$r_2(k) = r_1(k) - c_2(k), \text{ и т.д.}$$

Таким образом, достигается декомпозиция сигнала в n – эмпирическом приближении: $y(t) = c_n(t) + r_n(t)$.

Критерии останова процесса декомпозиции

1. Остаток $r_n(k)$ во всем интервале задания сигнала становятся несущественными по своим значениям по сравнению с сигналом.
2. Остаток $r_n(k)$ становится монотонной функцией, из которой больше не может быть извлечено функций IMF.
3. Так как в конечном итоге суммирование всех функций IMF (реконструкция сигнала) должно давать исходный сигнал, то можно останавливать разложение заданием относительной погрешности среднеквадратической реконструкции (без учета остатка $r_n(k)$).
4. По мере увеличения количества функций IMF относительная среднеквадратическая погрешность реконструкции достаточно сложных и протяженных сигналов уменьшается, но, как правило, имеет определенный минимум. По-видимому, это определяется попытками алгоритма разложить остаток на функции, частично компенсирующие друг друга. Соответственно, останов программы может выполняться, если следующая выделенная функция IMF увеличивает погрешность реконструкции.

Практический критерий останова процесса декомпозиции

Другими словами, остановка декомпозиции сигнала должна происходить при максимальном «выпрямлении» остатка, т.е. превращения его в тренд сигнала по интервалу задания с числом экстремумов не более 3. Даже для данных с нулевым средним значением конечный остаток может отличаться от нуля. Чтобы применять метод EMD, центрирования данных не требуется, метод нуждается только в локализациях экстремумов. Нулевая линия для каждого компонента декомпозиции будет сформирована процессом отсеивания.

Очистка сигналов от шумов с использованием разложения Хуанга

Особенностью сигналов, получаемых при физических экспериментах, является то, что они, как правило, нестационарные и значительно зашумлены. Применение традиционных способов очистки сигналов от шумов с помощью частотных фильтров (сглаживающих, медианных, Баттерворта и др.) приводят к значительному искажению формы сигнала, что может приводить к ошибочной физической интерпретации изучаемого процесса.

Известен другой способ очистки сигналов от шумов, основанный на использовании эмпирической модовой декомпозиции Хуанга [7,8]. Способ заключается в разложении исходного сигнала на составляющие (моды), затем в исключении первых составляющих (от 1 до 3), которые обычно представляют собой шум, содержащийся в исходном сигнале.

Преобразование Гильберта

Одним из достоинств преобразования Гильберта является то, что с помощью него можно $\varphi(t) = \arctan \left[\frac{Hs(t)}{s(t)} \right]$ вычислить мгновенную частоту сигнала.

Для этого используется понятие аналитического сигнала. Аналитический сигнал – комплексный сигнал $z(t) = s(t) + jHs(t)$, где $s(t)$ – исходный сигнал, который представляется в качестве действительной части комплексного сигнала, $Hs(t)$ – мнимая часть и является преобразованием Гильберта исходного сигнала $s(t)$. Тогда амплитуда сигнала $a(t)$ и фаза $\varphi(t)$ вычисляются по формулам:

$$a(t) = \sqrt{s(t)^2 + Hs(t)^2}$$

Мгновенная частота находится путем вычисления производной от фазы по времени. На рис. 7.4 приведен пример частотно-временного представления сигнала с линейной частотной модуляцией.

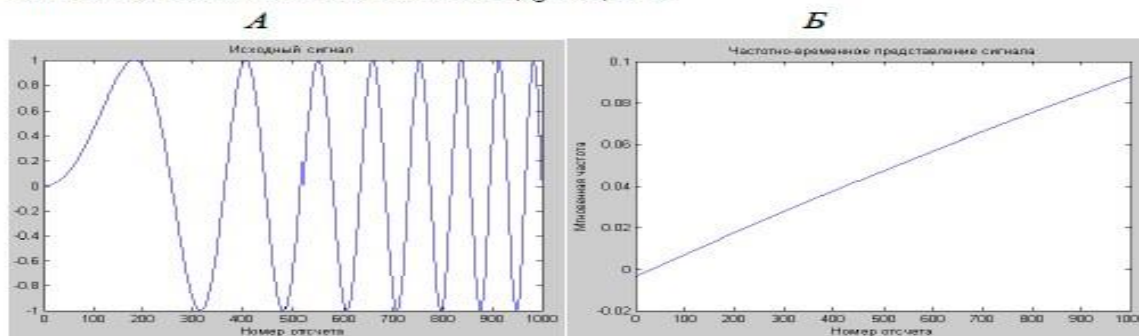


Рис. 7.4. Сигнал с линейной частотной модуляцией (А) и его частотно-временное представление (Б).

Указания к выполнению работы

1. Для изучения эмпирической декомпозиции Хуанга возьмите за основу базовую программу (см. Приложение 1) и выполните декомпозицию для трех видов сигналов: суммы двух стационарных гармонических сигналов; нестационарного гармонического сигнала;

зашумленного гармонического сигнала.

В первом случае представляет интерес возможность разделения гармонических составляющих сигналов в функциях IMF.

Во втором и третьем случае представляет интерес определение зависимости оптимального количества удаляемых функций IMF для очистки сигнала от шума в зависимости от уровня шума в исходном сигнале.

2. Для изучения преобразования Гильберта возьмите за основу базовую программу (см. Приложение 2)

Содержание отчета

1. Задание к работе.
2. Тексты программ эмпирической декомпозиции Хуанга и частотно-временного преобразования Гильберта.
3. Графики исходных сигналов и результатов обработки.
4. Интерпретация полученных результатов обработки. 5. Выводы.

Приложение 1. Текст базовой программы модовой декомпозиции Хуанга

```
%Эмпирическая модовая декомпозиция Хуанга. %Используется для  
очистки сигналов от шумов clc;%очистка Command Window
```

```
N=2401;%количество точек
```

```
d=0.005;%шаг изменения переменной time
```

```
M=(N-1)*d;% time
```

```
= 0:d:M;
```

```
Q=0.2;%уровень шума в долях СКО
```

```
NF=3;%кол-во функций разложения, содержащих шум (NF=1..3)
```

```
KP1=5;%кол-во периодов первого сигнала
```

```
KP2=7;%кол-во периодов второго сигнала
```

```
%ПРОГРАММНАЯ ГЕНЕРАЦИЯ СИГНАЛА
```

```
noise=rand(N);%генерация массива нормально распределенного шума
```

```
% noise=wgn(k,1,0);%генерация массива белого Гауссова шума for
```

```
k=1:N % генерация гармонического сигнала
```

```
s(k) = sin(2*pi*KP1*k/N)+ sin(2*pi*KP2*k/N);% 1-й вид сигнала
```

```
f(k)=s(k)+ Q*noise(k); % суммирование сигнала и шума end
```

```
%Генерация нестационарного сигнала
```

```
s = 4*sin(10*time.*time./100.0); f =
```

```
s(1:N)+ Q*noise(1:N); % Чтение
```

```
сигнала из файла
```

```
% filename = sensor_signal.xls';%3-й вид сигнала
```

```
% x = xlsread(filename);%3-й вид сигнала
```

```
% f = x(1:N);
```

```
% f(N) = f(N-1);
```

```
% f = zeros(1,1+M/d);
```

```
% i=1:1+M/d; % f(i)=x(i);
```

```
figure plot(f); title ('Исходный
```

```
сигнал') % Обнуление матрицы
```

```
остатков
```

```

r = zeros(10,1+M/d);
r(1,:) = f;%Операция создания 1-й строки матрицы остатков r из массива f
%первая строка матрицы остатков равна самому модельному сигналу
%ПОЛУЧЕНИЕ ФУНКЦИЙ РАЗЛОЖЕНИЯ
for j = 1:10 % 10 - максимальное количество функций разложения IMF h
= r(j,:);%r(j,:)-операция выделения строки j из матрицы r maxLastCount
= -10;
for i = 1:N % N - максимальное количество экстремумов
[ymax,imax,ymin,imin] = extrema(h);%нахождение массивов локальных максимумов
%и минимумов с помощью стандартной функции extrema, имеющейся в MATLAB
if((length(imax) < 2) || (length(imin) < 2)) break; end;
kUp = spline(time(imax),ymax,time);%сплайн-интерполяция локальных максимумов
kDown = spline(time(imin),ymin,time);%сплайн-интерполяция локальных минимумов
kMean = (kUp + kDown)/2; h = h - kMean;
[one, maxCount] = size(ymax);
if(abs(maxCount - maxLastCount) < 4) break;
end;
maxLastCount = maxCount;
end; c(j,:)
= h;
%Формирование искусственных экстремумов в начале и в конце массива c(j,1)=c(j,2);%1-ый
элемент массива Cj устанавливается равным 2-му c(j,2)=c(j,3)+0.1*abs(c(j,3));%2-ой элемент
массива Cj равен 3-му + 10% от модуля 3-го c(j,3)=c(j,4)-0.1*abs(c(j,4));%3-ий элемент
массива Cj равен 4-му минус 10% от модуля 4-го c(j,N)=c(j,N-1); %N-й элемент массива Cj
устанавливается равным N-1-му
c(j,N-2)=c(j,N-3)+0.1*abs(c(j,N-3)); %N-2-й элемент массива Cj равен N-3-му +10% от модуля N-3го
c(j,N-3)=c(j,N-4)-0.1*abs(c(j,N-4)) ; %N-3-й элемент массива Cj N-4-му минус 10% от модуля N-4го
r(j + 1,:) = r(j,:) - c(j,:); end;
%Вывод функций разложения IMF
figure; plot(c(1,:));title('IMF_1');
figure;plot(c(2,:)); title('IMF_2');
figure; plot(c(3,:));title('IMF_3');
figure;plot(c(4,:));title('IMF_4'); figure;
plot(c(5,:));title('IMF_5');
figure;plot(c(6,:));title('IMF_6');
figure;plot(c(7,:));title('IMF_7');
%Вывод остатков
figure; plot(r(1,:)); title('Остаток_1');
figure;plot(r(2,:));title('Остаток_2');
figure;plot(r(3,:));title('Остаток_3');
figure;plot(r(4,:));title('Остаток_4');
figure;plot(r(5,:));title('Остаток_5');
figure;plot(r(6,:));title('Остаток_6');
figure;plot(r(7,:));title('Остаток_7'); C1
= zeros(1,1+M/d);
%Суммирование функций разложения, содержащих только шум
for i = 1:NF

```

```

C1=C1+c(i,:);%накопление шумовых функций IMF end i=1:N; f1(i)=f(i)-
C1(i);%исходный зашумленный сигнал за вычетом шумовых IMF
%f1(i)=r(kki,i);
figure;plot(i,f(i),'r-');title('Исходный зашумленный сигнал');
hold on; %figure
plot(i,f1(i));title('Отфильтрованный сигнал'); hold
off;
% Следующий блок программы комментируется, если зашумленные
% данные берутся из файла (в этом случае
% чистый незашумленный сигнал s неизвестен) for
i=1:N
DZ(i)=f1(i)-s(i);%уровень зашумления в сигнале после
%вычитания функций разложения, содержащих только шум end
SKO=std(DZ)%СКО полной погрешности
KSKO=Q/SKO %коэффициент подавления шума
pause; close all; %закрытие всех окон графического
вывода
clear; %очистка Workspace
Приложение 2.Текст базовой программы преобразования Гильберта
%Частотно-временной анализ нестационарного сигнала
%с помощью преобразования Гильберта
n=1000; % количество отсчетов k=5 ; %
количество периодов
h=zeros(1,n); for
i=1:n
h(i)= sin(3*pi*k*i*i/n/n);%линейная частотная модуляция (ЛЧМ) %
h(i)=sin(2*pi*k*i*i*i/n/n/n); %квадратичная частотная модуляция
end figure plot(h)
title('Исходный сигнал') xlabel('Номер
отсчета');
hx=hilbert(h); %преобразование Гильберта
phi = angle(hx); %фаза phi2 = unwrap(phi);
p = polyfit(0:n-1,phi2,2); %аппроксимация полиномом второй степени
dp = polyder(p); %производная полинома f = polyval(dp, 0:n-1);
%Вычисление значений полинома figure plot(f)
title('Частотно-временное представление сигнала')
xlabel('Номер отсчета'); ylabel('Мгновенная
частота'); pause
close all; clear;

```

Лабораторная работа № 5

Средства идентификации объектов на изображениях Базовые средства фильтрации шумов на изображениях

Цель работы

Изучение методики использования базовых средств фильтрации шумов на изображениях в среде MATLAB, включающих усредняющий и медианный фильтры, фильтр Гаусса, адаптивный фильтр Винера, фильтр повышения резкости и ранговый фильтр и оценки их сравнительной эффективности. Изучение методики разработки программ цифровой корреляционной обработки изображений, включающей корреляционное сравнение изображений, формирование порогового правила для идентификации объектов на изображениях.

Задание к работе

1. Исследуйте связь эффективности фильтрации зашумленных изображений с визуальным восприятием изображений и степенью сходства отфильтрованного изображения с исходным незашумленным изображением, найдите оптимальные параметры фильтров;
2. Исследуйте связь эффективности фильтрации от показателей яркости и контрастности исходного изображения;
3. Спроектируйте на внутреннем языке MATLAB программу цифровой фильтрации изображений, содержащей средства интерфейса пользователя, позволяющие выбирать вид и задавать параметры фильтрации. ***Описание программного обеспечения и порядок работы***

Программное обеспечение включает базовую программу фильтрации шумов и файлы фотоизображений. Программа фильтрации шумов позволяет загрузить два изображения, выполнить зашумление одного из них, произвести фильтрацию зашумленного изображения с помощью различных фильтров, вычислить коэффициент корреляции зашумленного и отфильтрованных изображений с исходным, выполнить построение 2D и 3D графика коэффициента корреляции. Виды и параметры шума и параметры фильтров могут задаваться. Текст программы приведен в приложении.

Указания к выполнению работы

1. При выполнении п.1 Программы установите в программе уровень шума 0.005, вид шума – Гауссов, запустите программу Filtering.m, произведите кадрирование изображения, выделив центральную часть лица, и зафиксируйте в таблице получаемые в окне Command Window при выполнении программы максимальные значения коэффициентов корреляции зашумленного изображения и отфильтрованных разными фильтрами изображений. Затем, изменяя параметры фильтров, добейтесь получения наиболее высоких значений коэффициентов корреляции отфильтрованных сигналов с исходным.
2. Повторите выполнение п. 1 при шуме salt & pepper с параметрами $d=0.01$, 0.05 и 0.1 и при мультипликативном шуме speckle с нулевым средним значением и среднеквадратичным отклонением $v=0.02$, 0.04 и 0.06.
3. При выполнении п. 2 Программы спроектируйте на внутреннем языке MATLAB программу цифровой обработки изображений, реализующую уменьшение шумов на изображениях, используя стандартные базовые программы: average – усредняющий фильтр; medfilt – медианный фильтр; gaussian – фильтр Гаусса; ordfilt – ранговый фильтр; unsharp – фильтр повышения резкости; wiener2 – адаптивный фильтр Винера;

Программа должна позволять производить интерактивный выбор вида фильтра и параметров фильтра.

Содержание отчета

1. Задание к работе.
2. Таблицы с показателями корреляционного сравнения изображений по п. 1 и 2 Указаний по выполнению работы с комментариями.
3. Текст программы по п. 3 Указаний по выполнению работы и экранная форма. 4. Выводы.

Приложение. Текст базовой программы фильтрации шумов

```

%Программа производит зашумление изображения
%фильтрацию зашумленного изображения с помощью
%различных фильтров, определение коэффициента корреляции
%зашумленного и отфильтрованных изображений с исходным
% img=imread('m00200021.tif');
img2=imread('m00200001.tif');
img=imread('m00200117.tif'); %
img3=imread('m00200118.tif');
% img4=imread('m00200119.tif');
% img5=imread('m00200120.tif');
% img6=imread('m00200121.tif');
I=imcrop(img);
figure; imshow
(img);
title('Исходное изображение');
img1=imnoise(img,'gaussian',0,0.015);% 0.005
figure; imshow (img1);
title('Зашумленное изображение');
%Медианный фильтр 'medfilt2' hsize=[3
3 ];
F4= medfilt2(img1,hsize);
figure; imshow(F4);
title('После медианного фильтра'); pause;
%Усредняющий фильтр 'Average'
hsize=[3 3 ];
h= fspecial('average',hsize);
F1=imfilter(img1,h,'replicate');
figure; imshow (F1);
title('После усредняющего фильтра'); pause;
ncorr = normxcorr2(I(:, :, 1),img2(:, :, 1)); figure,
surf(ncorr), shading flat; title('Сравнение с
изображением другого человека');
max_c_other_man = max(abs(ncorr(:))) pause; ncorr =
normxcorr2(I(:, :, 1),img1(:, :, 1)); figure, surf(ncorr),
shading flat; title('После зашумления');
max_c_noised = max(abs(ncorr(:))) pause; ncorr =
normxcorr2(I(:, :, 1),F4(:, :, 1)); figure, surf(ncorr),
shading flat; title('После медианного фильтра');
max_c_median = max(abs(ncorr(:)))
pause; ncorr =
normxcorr2(I(:, :, 1),F1(:, :, 1));
figure, surf(ncorr), shading flat; title('После
усредняющего фильтра'); max_c_average
= max(abs(ncorr(:))) pause;
%Фильтр Гаусса 'gaussian'
hsize=[9 9]; sigma=0.99; h=
fspecial('gaussian',hsize,sigma);

```

```

F2=imfilter(img1,h,'replicate');
figure;
imshow(F2);title('После фильтра Гаусса');
pause; ncorr =
normxcorr2(I(:, :, 1),F2(:, :, 1)); figure,
surf(ncorr), shading flat; title('После
фильтра Гаусса'); max_c_gaussian =
max(abs(ncorr(:))) pause;
%Фильтр Лапласа 'laplacian'
alpha=0.5; h=
fspecial('laplacian',alpha);
F3=imfilter(img1,h,'replicate')
; figure;
imshow(F3);title('После фильтра Лапласа');
pause; ncorr = normxcorr2(I(:, :, 1),F3(:, :, 1));
figure, surf(ncorr), shading flat; title('После
фильтра Лапласа'); max_c_laplacian =
max(abs(ncorr(:))) pause;
%Фильтр повышения резкости 'unsharp'
alpha=0.2; h=
fspecial('unsharp',alpha);
F4=imfilter(img1,h); figure;
imshow(F4);title('После фильтра unsharp'); pause;
ncorr = normxcorr2(I(:, :, 1),F4(:, :, 1));
figure, surf(ncorr), shading flat;
title('После фильтра unsharp');
max_c_unsharp = max(abs(ncorr(:)))
pause; % Ранговый фильтр
m=5;n=5;
J = imnoise(img,'gaussian',0,0.015); imshow(J);
J6 = ordfilt2(J, 7, ones(4,4)); figure;
imshow(J6);title('После рангового фильтра'); pause;
ncorr = normxcorr2(I(:, :, 1),J6(:, :, 1));
figure, surf(ncorr), shading flat;
title('После рангового фильтра');
max_c_ordfilt2 = max(abs(ncorr(:))) pause;
%Фильтр Винера m=5;n=5;
[J,noise] = wiener2(img1,[m n]);
J6 = wiener2(J,[5 5]); figure;
imshow(J6);title('После фильтра Винера');
pause; ncorr = normxcorr2(I(:, :, 1),J6(:, :, 1));
figure, surf(ncorr), shading flat; title('После
фильтра Винера'); max_c_wiener =
max(abs(ncorr(:))) pause;
close all; %закрытие всех окон графического вывода clear;
%очистка Workspace

```

Задание к работе

Имеется набор изображений центральной части лица нескольких человек. Требуется спроектировать на внутреннем языке MATLAB программу цифровой обработки изображений, реализующую сравнение изображения человека с различными изображениями других людей из базы данных или различными изображениями того же человека, формирование порогового правила, позволяющего отличать «своего» от «чужого».

Описание программного обеспечения и порядок работы

Основные программы системы лабораторной работы функционируют в среде MATLAB. Программное обеспечение включает файлы 150 полноформатных фотоизображений, программу формирования эталонных фотоизображений и программу сравнения фотоизображений. Первый этап подготовки к работе системы заключается в создании базы данных эталонных изображений. В результате работы первой программы на основе каждого полноформатного изображения формируются 16 эталонных изображений центральной части лица, отличающиеся масштабом от исходного в пределах $\pm 10\%$.

На рис. приведен пример полноформатного фотоизображения, на рис. 9.2 – 16 эталонных фотоизображений центральной части лица, отличающихся масштабом в пределах $\pm 10\%$.



Рис. 9.2

Для выполнения основной функции – сравнения фотоизображений нужно запустить вторую программу. Программа производит корреляционное сравнение выбранного изображения с эталонными. Для исследования эффективности корреляционного принципа распознавания и формирования оптимального порога распознавания необходимо получить функции распределения коэффициентов корреляции при сравнении фотоизображений одних и тех же людей и разных людей.

В первом случае каждое исходное фотоизображение из базы данных масштабируется случайным образом, сравнивается с 16 эталонными изображениями этого же человека и находится максимальное значение коэффициента корреляции для каждого человека.

Во втором случае каждое исходное фотоизображение из базы данных масштабируется случайным образом, сравнивается с 16 эталонными изображениями остальных 149 людей и находится максимальное значение коэффициента корреляции для каждой пары сравниваемых фотоизображений.

Указания к выполнению работы

1. Для формирования эталонных изображений запустите программу формирования эталонных фотоизображений. При появлении окна с полноформатным изображением с помощью курсора выделите центральную часть лица, затем из контекстного меню выберите пункт **Imcrop**. Далее

будет автоматически производиться программная генерация 16 эталонных изображений, отличающихся масштабом от исходного в пределах $\pm 10\%$.

2. Корреляционное сравнение текущих изображений с эталонными производите в следующем порядке:

а) Произведите корреляционное сравнение выбранного изображения, масштаб которого измените случайным образом в пределах $\pm 10\%$ от исходного, с серией эталонных изображений, созданных при выполнении п. 1. Для этого запустите программу сравнения фотоизображений, после чего зафиксируйте результаты корреляционного сравнения, выводимые в окно Command Window (максимальное значение коэффициента корреляции). Аналогичные действия произведите для всех полноформатных изображений;

б) Произведите корреляционное сравнение выбранного изображения с серией эталонных изображений других людей, созданных при выполнении п. 1. Для этого запустите программу сравнения фотоизображений, после чего зафиксируйте результаты корреляционного сравнения, выводимые в окно Command Window (максимальные значения коэффициентов корреляции).

Аналогичные действия произведите для всех полноформатных изображений.

3. По полученным массивам коэффициентов корреляции постройте интервальные гистограммы, затем массивы гистограмм аппроксимируйте непрерывными функциями, используя программу подгонки кривых Curve Fitting и произведите проверку правильности гипотезы о законе распределения по критерию χ^2

4. Произведите анализ результатов корреляционного сравнения изображений «свой - свой» и «свой - чужой» и на основании этого определите пороговое правило по принципу максимума правдоподобия для разделения по значению коэффициента корреляции «своих» и «чужих».

Оценить вероятность ошибок 1-го (принять «своего» за «чужого») и 2-го рода (принять «чужого» за «своего»).

5. Вычислите порог корреляционного сравнения по принципу «идеального наблюдателя» и Неймана-Пирсона, считая допустимой ошибкой «пропуска чужого» - 10^{-3} .

Содержание отчета

1. Задание к работе

2. Текст программы с комментариями.

3. Таблицы полученных значений коэффициентов корреляции.

4. Расчет и обоснование порогового правила для различения «своих» и «чужих» по принципу максимума правдоподобия.

5. Расчет вероятности ошибок 1-го и 2-го рода. 6. Выводы.

Приложение 1. Критерий χ^2 с $k=l-r-1$ степенями свободы.

$$\chi^2 = \sum_{j=1}^l \frac{(m_j - np_j)^2}{np_j}$$

где

r – число параметров распределения, определяемых по выборке;

m_j – табличная частота;

np_j – теоретич. частота;

p_j – вероятность функции распределения, вычисляемая по формуле:

$$P_i^{\text{theor}} = \int_{x_{i-1}}^{x_i} f(x) dx$$

Табл.9.1

Критерий Пирсона (хи - квадрат) при уровне значимости
(0,01; 0,05; 0,1; 0,15; 0,20; 0,25; 0,30)

| n | P - 0,01 | P - 0,05 | P - 0,1 | P - 0,15 | P - 0,2 | P - 0,25 | P - 0,3 |
|----|----------|----------|---------|----------|---------|----------|---------|
| 1 | 6,6349 | 3,8415 | 2,7055 | 2,0723 | 1,6424 | 1,3233 | 1,0742 |
| 2 | 9,2103 | 5,9915 | 4,6052 | 3,7942 | 3,2189 | 2,7726 | 2,4079 |
| 3 | 11,3449 | 7,8147 | 6,2514 | 5,3170 | 4,6416 | 4,1083 | 3,6649 |
| 4 | 13,2767 | 9,4877 | 7,7794 | 6,7449 | 5,9886 | 5,3853 | 4,8784 |
| 5 | 15,0863 | 11,0705 | 9,2364 | 8,1152 | 7,2893 | 6,6257 | 6,0644 |
| 6 | 16,8119 | 12,5916 | 10,6446 | 9,4461 | 8,5581 | 7,8408 | 7,2311 |
| 7 | 18,4753 | 14,0671 | 12,0170 | 10,7479 | 9,8032 | 9,0371 | 8,3834 |
| 8 | 20,0902 | 15,5073 | 13,3616 | 12,0271 | 11,0301 | 10,2189 | 9,5245 |
| 9 | 21,6660 | 16,9190 | 14,6837 | 13,2880 | 12,2421 | 11,3888 | 10,6564 |
| 10 | 23,2092 | 18,3070 | 15,9872 | 14,5339 | 13,4420 | 12,5489 | 11,7807 |
| 11 | 24,7249 | 19,6751 | 17,2750 | 15,7671 | 14,6314 | 13,7007 | 12,8987 |
| 12 | 26,2169 | 21,0260 | 18,5493 | 16,9893 | 15,8120 | 14,8454 | 14,0111 |
| 13 | 27,6882 | 22,3620 | 19,8119 | 18,2020 | 16,9848 | 15,9839 | 15,1187 |
| 14 | 29,1412 | 23,6848 | 21,0641 | 19,4062 | 18,1508 | 17,1169 | 16,2221 |
| 15 | 30,5779 | 24,9958 | 22,3071 | 20,6030 | 19,3107 | 18,2451 | 17,3217 |
| 16 | 31,9999 | 26,2962 | 23,5418 | 21,7930 | 20,4651 | 19,3689 | 18,4179 |
| 17 | 33,4087 | 27,5871 | 24,7690 | 22,9770 | 21,6146 | 20,4887 | 19,5110 |
| 18 | 34,8053 | 28,8693 | 25,9894 | 24,1555 | 22,7595 | 21,6049 | 20,6014 |
| 19 | 36,1909 | 30,1435 | 27,2036 | 25,3288 | 23,9004 | 22,7178 | 21,6891 |
| 20 | 37,5662 | 31,4104 | 28,4120 | 26,4976 | 25,0375 | 23,8277 | 22,7745 |

Приложение 2. Программа создания эталонных изображений

```
%Программа создания эталонных изображений
%из фотоизображений 480x640 формата .tif
clc; for i=1:10
t=num2str(i);%номер исходного фотоизображения password=num2str(i);%номер
группы эталонов
path='D:\LifeCamFiles\P1\'; fname=strcat(path,t,'.tif');
J=imread (fname);%Чтение файла с изображением для создания
%группы эталонов
J3=imresize(J,0.5,'bicubic');%масштабирование figure;
imshow(J3);
I=imcrop(J3);%интерактивное выделение центральной части лица
```

```

figure; imshow(I);
path='D:\LifeCamFiles\'; h =
waitbar(0,'Please wait...'); for
i=1:16 t=num2str(10+i);
k=0.84+i*0.02;
J=imresize(I,k,'bicubic'); fname=strcat(path,password,t,'.tif');
imwrite(J,fname); %
figure;imshow(J);
waitbar(i/16) end
pause(2);
set(0,'ShowHiddenHandles','on')% автоматическое удаление всех figures close
all;
clear; end

```

Приложение 3. Программа корреляционного сравнения изображений

```

%Программа корреляционного сравнения полноформатного
% фотоизображения с эталонными clc;
msgbox('Введите номер полноформатного фотоизображения','Выбор номера изображения','warn');
answer=input('Введите номер фотоизображения ','s'); t=char(answer); msgbox('Введите номер
группы эталонов ','Выбор группы эталонов','warn'); answer=input('Введите номер группы
эталонов ','s'); password=char(answer); path='D:\LifeCamFiles\P1\'; fname=strcat(path,t,'.jpg');
I=imread(fname);
%Изменение масштаба полноформатного изображения по нормальному закону
% 100 - количество случайных значений, 0,05 - СКО,
% макс. отклонение с доверит. вероятностью 0,95 - 3*СКО
k=abs(1.0+ 0.05*(randn(size(100)))));
J=imresize(I,k,'bicubic');%изменение масштаба
X1=imresize(J,0.5,'bicubic');
figure;imshow(X1);
path='D:\LifeCamFiles\'; h =
waitbar(0,'Please wait...'); for
i=1:16 t=num2str(78+i);
fname=strcat(path,password,t,'.tif');
X2=imread(fname); figure;imshow(X2);
% изменен порядок сравнения: X2 с X1
ncorr = normxcorr2(X2(:,1),X1(:,1));
% figure, surf(ncorr), shading flat; max_c
= max(abs(ncorr(:))); y(i)=max_c;
waitbar(i/16) end
set(0,'ShowHiddenHandles','on')% автома
тическое удаление всех figures
delete(get(0,'Children'))% автоматическо
е удаление всех figures y(9)=0;
corr=y; max_corr=max(y)
set(0,'ShowHiddenHandles','on')% автоматическое удаление всех figures
close all;
clear;

```

Расчетные задания

Расчетное задание 1

1. Могут ли векторы v_1 и v_2 образовывать базис, если

$$v_1 = \left(\frac{\sqrt{3}}{2}, \frac{\sqrt{1}}{2}\right), \quad v_2 = \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$$

1. Разложите вектор

$$f = \left(\frac{\sqrt{3}}{2}, \frac{5}{2}\right)$$

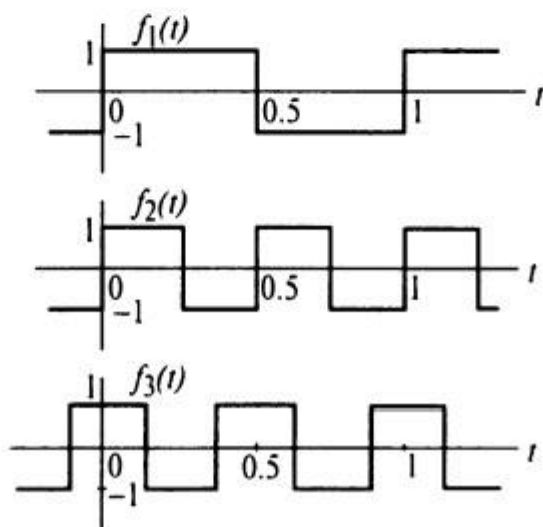
по базису $\{v_1, v_2\}$

2. Найдите расстояние между векторами

$$f = (4, -4, 7), \quad g = (3, -2, 6),$$

их скалярное произведение и коэффициент корреляции.

4. Докажите, что сигналы $f_1(t)$, $f_2(t)$, $f_3(t)$, изображенные на рисунке, взаимно перпендикулярны на отрезке $[0, 1]$



Расчетное задание 2

Вариант 1.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t)=t^2$ на интервале $[-\pi, \pi]$ ($[-\frac{T}{2}, \frac{T}{2}]$).
2. Получите аналитическое выражение для коэффициента корреляции R двух гармонических сигналов на интервале $[0, T]$, отличающихся по частоте (по количеству периодов на интервале определения):

$$x(t) = \cos\left(\frac{2\pi n(t - T/2)}{T}\right) \quad \text{и} \quad y(t) = \cos\left(\frac{2\pi n_2(t - T/2)}{T}\right),$$

где n, n_2 – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$,

$$n - n_2 = \Delta n, \quad \Delta n < \frac{1}{n}.$$

Получите аналитическое выражение, постройте зависимость коэффициента корреляции от относительного сдвига по частоте

$$R = R(\delta), \quad \delta = \frac{\Delta n}{n}, \quad n = 4.2, \quad \delta = +/\!- 0.15$$

и найдите значение δ , при котором достигается максимум функции $R(\delta)$

- 2*. Получите аналитическое выражение для коэффициента корреляции R двух отличающихся по частоте гармонических сигналов, модулированных по амплитуде экспоненциальной оконной функцией:

$$x(t) = \cos\left(\frac{2\pi n(t - T/2)}{T}\right)w(t), \quad y(t) = \cos\left(\frac{2\pi n_2(t - T/2)}{T}\right)w(t),$$

$$w(t) = \exp[-(n(t - T/2)/T)^2]$$

- 2**. Получите аналитическое выражение для коэффициента корреляции R двух отличающихся по частоте гармонических сигналов, модулированных по амплитуде экспоненциальной оконной функцией, в условиях, когда один из сигналов зашумлен:

$$x(t) = \cos\left(\frac{2\pi n(t - T/2)}{T}\right)w(t) \quad \text{и} \quad y(t) = \cos\left(\frac{2\pi n_2(t - T/2)}{T}\right)w(t) + \xi(t)$$

где $\xi(t)$ – шум.

Расчетное задание 2

Вариант 2.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t) = |t|$ на интервале $[-\pi, \pi]$ $(-\frac{T}{2}, \frac{T}{2})$.
2. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов на интервале $[0, T]$, имеющих одинаковую частоту и количество периодов, но отличающихся сдвигом по времени:

$$x(t) = \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right) \quad \text{и} \quad y(t) = \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right), \quad \text{shift1}, \text{shift2} \in [0, T]$$

где n – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$.

Получите аналитическое выражение и постройте зависимость коэффициента ковариации от относительного сдвига по времени δ :

$$R = R(\delta), \quad \delta = \frac{\Delta t}{T}, \quad \Delta t = \text{shift2} - \text{shift1}, \quad n = 9, \quad \delta \in [-0.05, 0.05]$$

и найдите значение δ , при котором достигается максимум функции $R(\delta)$

- 2*. Получите аналитическое выражение для коэффициента ковариации R двух отличающихся сдвигом по времени гармонических сигналов, модулированных экспоненциальными оконными функциями $w_1(t)$ и $w_2(t)$:

$$x(t) = \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right) w_1(t), \quad y(t) = \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right) w_2(t),$$

$$w_1(t) = \exp\left[-\left(\frac{m(t - \text{shift1})}{T}\right)^2\right], \quad w_2(t) = \exp\left[-\left(\frac{m(t - \text{shift2})}{T}\right)^2\right]$$

$m = 10$ – количество периодов несущей частоты радиоимпульса на интервале $[0, T]$

- 2**. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов, модулированных экспонентами и отличающихся сдвигом по времени, в условиях, когда один из сигналов зашумлен:

$$x(t) = \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right) w_1(t), \quad y(t) = \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right) w_2(t) + \xi(t),$$

где $\xi(t)$ – шум, $0 < \text{СКО шума} < 0.2$

и постройте зависимость СКО интеграла R от СКО шума.

Расчетное задание 2

Вариант 3.

Вычислите аналитически коэффициенты разложения в ряд Фурье

функции $f(t) = |\sin(t)|$ на интервале $[-\pi, \pi]$ $(-\frac{T}{2}, \frac{T}{2})$.

В таблице представлены значения среднемесячных температур для двух городов F и G (измерения проводились через месяц).

Температура [$^{\circ}\text{C}$]

| i | 1 | 2 | 3 | 4 | 5 | 6 |
|-------------------|----|----|----|----|----|----|
| Месяцы | 1 | 3 | 5 | 7 | 9 | 11 |
| Город F (f_i) | 4 | 10 | 20 | 25 | 20 | 5 |
| Город G (g_i) | 20 | 18 | 15 | 10 | 17 | 22 |

Составьте таблицу и график значений функции взаимной корреляции температур атмосферы этих городов и прокомментируйте ее. Функция взаимной корреляции определяется соотношением:

$$R_R(j) = \frac{1}{6} \sum_{i=1}^6 (f_i - \bar{f})(g_{i+j} - \bar{g}),$$

где \bar{f} , \bar{g} – среднеарифметические значения f_i и g_i .

Расчетное задание 2

Вариант 4.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t) = t^* \sin(t)$ на интервале $[-\pi, \pi]$ ($[-\frac{T}{2}, \frac{T}{2}]$).
2. Вычислите⁵ и постройте зависимость значения интеграла R модуля суммы двух гармонических сигналов, отличающихся по частоте (по количеству периодов на интервале $[0, T]$):

$$R(\delta) = \frac{1}{T} \int_0^T \left| \sin\left(\frac{2\pi n_1 t}{T}\right) + \sin\left(\frac{2\pi n_2 t}{T}\right) \right| dt$$

где n, n_2 – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$,

$$n - n_2 = \Delta n, \quad \Delta n < \frac{1}{n}.$$

от относительного сдвига по частоте

$$R = R(\delta), \quad \delta = \frac{\Delta n}{n}, \quad n = 4.2, \quad \delta = + / - 0.15.$$

- 2*. Вычислите и постройте зависимость интеграла R модуля суммы двух гармонических сигналов, модулированных экспоненциальной оконной функцией $w(t)$, и отличающихся по частоте:

$$x(t) = \sin\left(\frac{2\pi n_1 t}{T}\right)w(t), \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right)w(t),$$

$$w(t) = \exp(-(n(t - T/2)/T)^2)$$

- 2**. Вычислите и постройте зависимость значения интеграла R модуля суммы двух гармонических сигналов, отличающихся по частоте, в условиях, когда один из сигналов зашумлен:

$$x(t) = \sin\left(\frac{2\pi n_1 t}{T}\right)w(t) \quad \text{и} \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right)w(t) + \xi(t)$$

где $\xi(t)$ – шум, $0 < \text{СКО шума} < 0.2$

и постройте зависимость СКО интеграла R от СКО шума.

⁵ Использовать численное интегрирование методом трапеций. Пример программы приведен ниже

```
x1=0:0.01:2;
y1=x1.^2;
int1=trapz(x1,y1) %числ.интегр.методом трапеций
```

...

Расчетное задание 2

Вариант 5.

1. Вычислить аналитически коэффициенты разложения в ряд Фурье функции $f(t) = \exp(at)$ на интервале $[-\pi, \pi]$ $(-\frac{T}{2}, \frac{T}{2})$.
2. Вычислите⁶ и постройте на интервале $[0, T]$ зависимость значения интеграла R модуля разности двух гармонических сигналов $x(t)$ и $y(t)$, имеющих одинаковую частоту и количество периодов, но отличающихся сдвигом по времени:

$$R(\delta) = \frac{1}{T} \int_0^T \left| \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right) - \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right) \right| dt, \text{shift1}, \text{shift2} \in [0, T]$$

где n – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$.

$$R = R(\delta), \quad \delta = \frac{\Delta t}{T}, \quad \Delta t = \text{shift2} - \text{shift1}, \quad n = 9, \quad \delta = [-0.1, 0.1]$$

- 2*. Вычислите и постройте зависимость интеграла R от модуля разности двух гармонических сигналов, модулированных экспоненциальными оконными функциями $w_1(t)$ и $w_2(t)$:

$$w_1(t) = \exp\left[-\left(\frac{m(t - \text{shift1})}{T}\right)^2\right], \quad w_2(t) = \exp\left[-\left(\frac{m(t - \text{shift2})}{T}\right)^2\right]$$

$m = 10$ – количество периодов несущей частоты радиоимпульса на интервале $[0, T]$

- 2**. Вычислите и постройте зависимость значения математического ожидания интеграла R от модуля суммы двух гармонических сигналов, отличающихся сдвигом по времени, в условиях, когда один из сигналов зашумлен:

$$x(t) = \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right)w_1(t), \quad y(t) = \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right)w_2(t) + \xi(t),$$

где $\xi(t)$ – шум, $0 < \text{СКО шума} < 0.2$

и постройте зависимость СКО интеграла R от СКО шума.

⁶ Использовать численное интегрирование методом трапеций. Пример программы приведен ниже
`x1=0:0.01:2;
y1=x1.^2;
int1=trapz(x1,y1) %числ.интегр.методом трапеций`

Расчетное задание 2

Вариант 6.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t)=x*\exp[ax]$ на интервале $[-\pi, \pi]$ $(-\frac{T}{2}, \frac{T}{2})$.
2. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов, имеющих одинаковую частоту и количество периодов, но отличающихся сдвигом по времени, на интервале $[0, T]$:

$$x(t) = \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right) \quad \text{и} \quad y(t) = \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right), \quad \text{shift1}, \text{shift2} \in [0, T]$$

где n – количество периодов сигнала $x(t)$ и сигнала $y(t)$ на интервале $[0, T]$.

Получите аналитическое выражение, постройте зависимость от относительного сдвига по времени:

$$R = R(\delta), \quad \delta = \frac{\Delta t}{T}, \quad \Delta t = \text{shift2} - \text{shift1}, \quad n = 5, \quad \delta = [-0.1, 0.1]$$

и найдите значение δ , при котором достигается максимум функции $R(\delta)$

- 2*. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов $x(t)$ и $y(t)$, модулированных экспоненциальными оконными функциями $w_1(t)$ и $w_2(t)$:

$$w_1(t) = \exp\left[-\left(\frac{m(t - \text{shift1})}{T}\right)^2\right], \quad w_2(t) = \exp\left[-\left(\frac{m(t - \text{shift2})}{T}\right)^2\right]$$

- 2**. Получите аналитическое выражение для коэффициента ковариации R двух гармонических сигналов $x(t)$ и $y(t)$, отличающихся сдвигом по времени, в условиях, когда один из сигналов зашумлен:

$$x(t) = \cos\left(\frac{2\pi n(t - \text{shift1})}{T}\right)w_1(t), \quad y(t) = \cos\left(\frac{2\pi n(t - \text{shift2})}{T}\right)w_2(t) + \xi(t),$$

где $\xi(t)$ – шум.

Расчетное задание 2

Вариант 7.

1. Вычислите аналитически коэффициенты разложения в ряд Фурье функции $f(t) = x^* \cos(x)$ на интервале $[-\pi, \pi]$ ($[-\frac{T}{2}, \frac{T}{2}]$).
2. Получите аналитическое выражение для коэффициента корреляции R двух гармонических сигналов на интервале $[0, T]$, отличающихся по частоте (по количеству периодов на интервале определения):

$$x(t) = \sin\left(\frac{2\pi n t}{T}\right) \quad \text{и} \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right),$$

где n, n_2 – количество периодов сигнала $x(t)$ и $y(t)$ на интервале $[0, T]$,

$$n - n_2 = \Delta n, \quad \Delta n < \frac{1}{n}.$$

Получите аналитическое выражение, постройте зависимость коэффициента корреляции от относительного сдвига по частоте

$$R = R(\delta), \quad \delta = \frac{\Delta n}{n}, \quad n = 4.2, \quad \delta = +/\!- 0.15$$

и найдите значение δ , при котором достигается максимум функции $R(\delta)$

- 2*. Получите аналитическое выражение для коэффициента корреляции R двух отличающихся по частоте гармонических сигналов, модулированных по амплитуде экспоненциальной оконной функцией:

$$x(t) = \sin\left(\frac{2\pi n t}{T}\right)w(t), \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right)w(t),$$

$$w(t) = \exp\left[-(n(t - T/2)/T)^2\right]$$

- 2**. Получите аналитическое выражение для коэффициента корреляции R двух отличающихся по частоте гармонических сигналов, модулированных по амплитуде экспоненциальной оконной функцией, в условиях, когда один из сигналов зашумлен:

$$x(t) = \sin\left(\frac{2\pi n t}{T}\right)w(t) \quad \text{и} \quad y(t) = \sin\left(\frac{2\pi n_2 t}{T}\right)w(t) + \xi(t)$$

где $\xi(t)$ – шум.

Лабораторная работа № 6

Построение Байесовского классификатора по выборке двумерных нормально распределенных векторов. Построение Байесовского классификатора по прецедентам

Цель работы

Изучение Байесовского классификатора

Приложение 3. Построение байесовского классификатора по выборке двумерных нормально распределенных векторов

1. Априорные вероятности появления классов

$$p1 \approx 0.4 \qquad p2 \approx 0.6$$

2. Генерирование двумерных обучающих векторов, распределенных по нормальному закону

2.1. Задание параметров нормального сферического распределения

$$mx1 \approx 0 \quad my1 \approx 0 \quad \sigma x1 \approx 2 \quad \sigma y1 \approx 1 \text{ - параметры нормального распределения в первом классе}$$

$$mx2 \approx 4 \quad my2 \approx 3 \quad \sigma x2 \approx 2 \quad \sigma y2 \approx 1 \text{ - параметры нормального распределения во втором классе}$$

2.2. Генерирование центрированных нормально распределенных значений - координат случайных векторов

$$N \approx 50 \quad \text{- размер выборки}$$

$$X1 \approx \text{rnorm}(N, 0, \sigma x1) \qquad X2 \approx \text{rnorm}(N, 0, \sigma x2)$$

$$Y1 \approx \text{rnorm}(N, 0, \sigma y1) \qquad Y2 \approx \text{rnorm}(N, 0, \sigma y2)$$

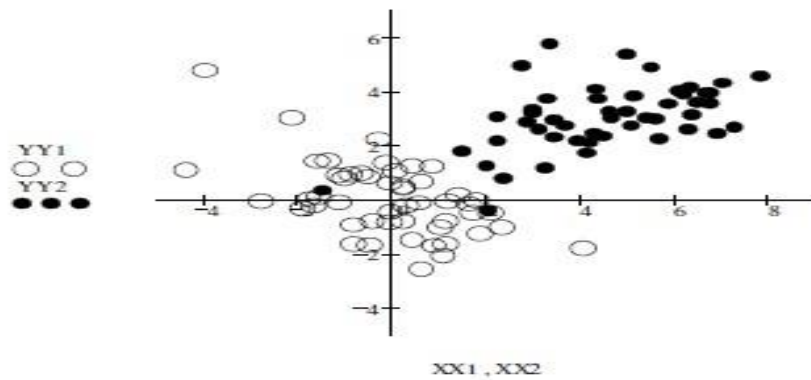
2.3. Поворот и смещение нормально распределенных векторов

$$a1 := \frac{\pi}{6} \qquad a2 \approx \frac{5 \cdot \pi}{6} \quad \text{- углы поворота эллипсов рассеяние}$$

$$i \approx 0..N-1$$

$$XX1_i \approx X1_i \cdot \cos(a1) + Y1_i \cdot \sin(a1) + mx1 \qquad XX2_i \approx X2_i \cdot \cos(a2) + Y2_i \cdot \sin(a2) + mx2$$

$$YY1_i \approx -X1_i \cdot \sin(a1) + Y1_i \cdot \cos(a1) + my1 \qquad YY2_i \approx -X2_i \cdot \sin(a2) + Y2_i \cdot \cos(a2) + my2$$



3. Вычисление оценок числовых характеристик распределений векторов в классах

3.1. Вычисление оценки центра рассеяния в первом классе

$$MX1 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} XX1_i \quad MY1 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} YY1_i$$

$$MX1 = -0.083 \quad MY1 = 0.019 \quad M1 := \begin{pmatrix} MX1 \\ MY1 \end{pmatrix}$$

3.2. Вычисление оценки центра рассеяния во втором классе

$$MX2 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} XX2_i \quad MY2 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} YY2_i$$

$$MX2 = 4.58 \quad MY2 = 3.037 \quad M2 := \begin{pmatrix} MX2 \\ MY2 \end{pmatrix}$$

3.3. Вычисление оценок элементов ковариационной матрицы в первом классе

$$KXX1 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (XX1_i - MX1)^2 \quad KYY1 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (YY1_i - MY1)^2$$

$$KXY1 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (XX1_i - MX1) \cdot (YY1_i - MY1)$$

$$S1 := \begin{pmatrix} KXX1 & KXY1 \\ KXY1 & KYY1 \end{pmatrix} \quad S1 = \begin{pmatrix} 2.388 & -1.122 \\ -1.122 & 1.762 \end{pmatrix}$$

3.4. Вычисление оценок элементов ковариационной матрицы во втором классе

$$KXX2 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (XX2_i - MX2)^2 \quad KYY2 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (YY2_i - MY2)^2$$

$$KXY2 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (XX2_i - MX2) \cdot (YY2_i - MY2)$$

$$S2 := \begin{pmatrix} KXX2 & KXY2 \\ KXY2 & KYY2 \end{pmatrix} \quad S2 = \begin{pmatrix} 3.878 & 1.235 \\ 1.235 & 1.445 \end{pmatrix} \quad \text{- оценки ковариационных матриц}$$

$$S1^{-1} = \begin{pmatrix} 0.598 & 0.381 \\ 0.381 & 0.81 \end{pmatrix} \quad S2^{-1} = \begin{pmatrix} 0.354 & -0.303 \\ -0.303 & 0.951 \end{pmatrix} \quad \text{- обратные к ковариационным матрицы}$$

4. Вычисление решающих функций, определяемых байесовским классификатором

4.1. Определение метрик Махаланобиса в каждом классе

$$Mach_1(x, y) := (S1^{-1})_{0,0} \cdot x^2 + 2 \cdot (S1^{-1})_{0,1} \cdot x \cdot y + (S1^{-1})_{1,1} \cdot y^2$$

$$Mach_2(x, y) := (S2^{-1})_{0,0} \cdot x^2 + 2 \cdot (S2^{-1})_{0,1} \cdot x \cdot y + (S2^{-1})_{1,1} \cdot y^2$$

4.2. Определение решающих функций

$$dk1(x, y) := \ln(p1) - 0.5 \cdot \ln(|S1|) - 0.5 \cdot Mach_1(x - MX1, y - MY1)$$

$$dk2(x, y) := \ln(p2) - 0.5 \cdot \ln(|S2|) - 0.5 \cdot Mach_2(x - MX2, y - MY2)$$

$$d(x, y) := dk1(x, y) - dk2(x, y)$$

5. Прорисовка линии разделения двух классов

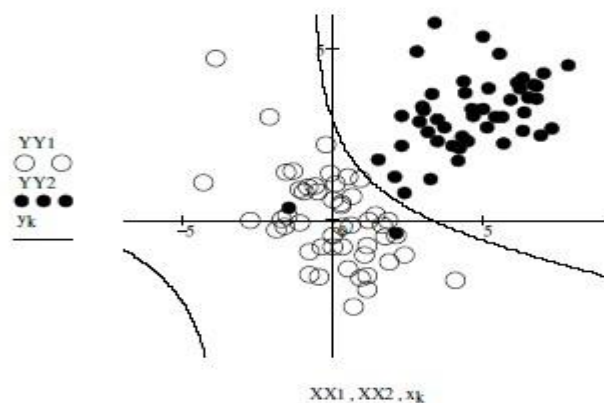
$n := 10000$

$k := 0..n-1$

$a := -7 \quad b := 9$

$x_k := a + \frac{k \cdot (b - a)}{n}$

$y0 := -4 \quad y_k := \text{root}(d(x_k, y0), y0)$



Приложение 5. Построение байесовского классификатора по прецедентам

1. Формирование обучающей выборки по изображению точечных данных

1.1. Считывание точечных данных из файла

$A := \text{READBMP}("d\backslash\text{выборка_2"})$

1.2. Выделение точек из матрицы изображения, формирование массива точечных данных X

```
quant(A, a, b) :=
  s ← 0
  for i ∈ 0..rows(A) - 1
    for j ∈ 0..cols(A) - 1
      if (a ≤ Ai,j) · (Ai,j ≤ b)
        X(s) ←  $\begin{pmatrix} i \\ j \end{pmatrix}$ 
        s ← s + 1
  X
```

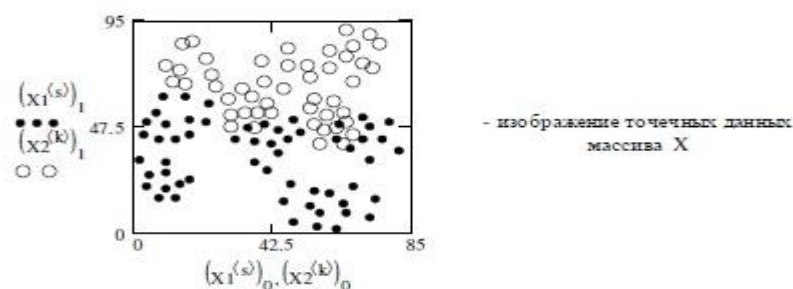
$X1 = \text{quant}(A, 0, 50) \quad X1 =$

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|----|----|----|----|----|----|----|----|----|
| 0 | 0 | 2 | 3 | 4 | 4 | 5 | 7 | 8 | 8 | 9 |
| 1 | 1 | 33 | 44 | 21 | 50 | 26 | 54 | 16 | 42 | 61 |

$X2 = \text{quant}(A, 50, 200) \quad X2 =$

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|----|----|----|----|----|----|----|----|----|
| 0 | 0 | 10 | 12 | 14 | 15 | 16 | 18 | 22 | 24 | 25 |
| 1 | 1 | 75 | 68 | 73 | 85 | 67 | 86 | 78 | 71 | 66 |

$s := 0..cols(X1) - 1 \quad k := 0..cols(X2) - 1$



2. Вычисление гистограмм с нерегулярными ячейками

2.1. Распределение основного массива данных по ячейкам в соответствии со взвешенной метрикой, нахождение центров классов и дисперсий (см. Приложение 2)

$$\text{DISTR1} := \text{distr}(X1, 10, 8, 25) \quad B1 := \text{DISTR1}_0 \quad X1 := \text{DISTR1}_1$$

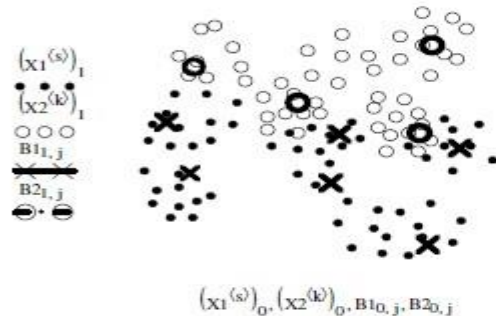
$$B1 = \begin{pmatrix} 12.5 & 7 & 46.2 & 44.333 & 66.333 & 73.571 \\ 32.25 & 51 & 46.6 & 28.667 & 6 & 41.429 \\ 31.318 & 11.75 & 19.2 & 15.083 & 29.083 & 26.952 \\ 141.295 & 10 & 13.675 & 49.333 & 13 & 36.952 \\ 12 & 3 & 5 & 3 & 3 & 7 \end{pmatrix}$$

- массив центров кластеров, дисперсий и количества элементов в кластерах для первого класса

$$\text{DISTR2} := \text{distr}(X2, 10, 8, 25) \quad B2 := \text{DISTR2}_0 \quad X2 := \text{DISTR2}_1$$

$$B2 = \begin{pmatrix} 13 & 36.909 & 67.143 & 64.4 \\ 70.75 & 58.091 & 79.143 & 47 \\ 10.667 & 28.391 & 36.81 & 10 \\ 14.917 & 40.491 & 21.476 & 29 \\ 4 & 11 & 7 & 5 \end{pmatrix}$$

- массив центров кластеров, дисперсий и количества элементов в кластерах для второго класса



2.3. Распределение оставшихся нераспределенных элементов по кластерам методом ближайшего соседа (см. Приложение 2)

$$D1 := \text{dopdistr}(B1, X1) \quad B1 := D1_0 \quad X1 := D1_1$$

$$B1 = \begin{pmatrix} 12.5 & 7 & 46.2 & 44.333 & 66.333 & 73.571 \\ 32.25 & 51 & 46.6 & 28.667 & 6 & 41.429 \\ 31.318 & 11.75 & 19.2 & 15.083 & 29.083 & 26.952 \\ 141.295 & 10 & 13.675 & 49.333 & 13 & 36.952 \\ 15 & 7 & 8 & 12 & 6 & 10 \end{pmatrix}$$

- итоговый массив центров кластеров, дисперсий и количества элементов в кластерах для первого класса

$$X1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 2 & 3 & 4 & 4 & 5 & 7 & 8 & 8 & 9 \\ 1 & 39 & 44 & 21 & 50 & 26 & 54 & 16 & 42 & 61 \\ 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 1 & 2 \end{pmatrix}$$

- массив элементов первого класса с указанием номера кластера, к которому элемент относится

$$D2 := \text{dopdistr}(B2, X2) \quad B2 := D2_0 \quad X2 := D2_1$$

$$B2 = \begin{pmatrix} 13 & 36.909 & 67.143 & 64.4 \\ 70.75 & 58.091 & 79.143 & 47 \\ 10.667 & 28.391 & 36.81 & 10 \\ 14.917 & 40.491 & 21.476 & 29 \\ 9 & 17 & 10 & 12 \end{pmatrix}$$

- итоговый массив центров кластеров, дисперсий и количества элементов в кластерах для второго класса

$$X2 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 10 & 12 & 14 & 15 & 16 & 18 & 22 & 24 & 25 \\ 1 & 75 & 68 & 73 & 85 & 67 & 86 & 78 & 71 & 66 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \end{pmatrix}$$

- массив элементов второго класса с указанием номера кластера, к которому элемент относится

2.4. Построения областей Дирикле и вычисления их площадей (см. Приложение 2)

$$G1 := \text{square}(A, B1, X1, 8) \quad B1 := G1_0 \quad AA1 := G1_1$$

$$B1 = \begin{pmatrix} 12.5 & 7 & 46.2 & 44.333 & 66.333 & 73.571 \\ 32.25 & 51 & 46.6 & 28.667 & 6 & 41.429 \\ 31.318 & 11.75 & 19.2 & 15.083 & 29.083 & 26.952 \\ 141.295 & 10 & 13.675 & 49.333 & 13 & 36.952 \\ 15 & 7 & 8 & 12 & 6 & 10 \\ 920 & 534 & 563 & 925 & 430 & 804 \end{pmatrix}$$

- итоговый массив центров кластеров, дисперсий и количества элементов в кластерах для первого класса с добавленной строкой площадей областей Дирикле

- итоговый массив
центров кластеров
второго класса с
добавленной строкой
площадей областей
Директе

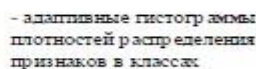
2.5. "Нанесение" точек на изображение областей Дирихле

$$AX2 \approx \text{point}(AA2, X2, 0)$$

Области Дирихле первого и второго классов



2.6. Вычисления адаптивных гистограмм в классах (см. Приложение 2)

$$AA2 \approx \text{hist}(AA2, B2, X2)$$


AA1,AA2

3. Построение байесовского классификатора по гистограммам распределения признаков в классах

3.1. Задание априорных вероятностей появления классов p и $1-p$

 $p := 0.5$

3.2. Функция вычисления областей предпочтения байесовской классификации

```

bayes(A1,A2,p) =
  for i ∈ 0..rows(A1) - 1
    for j ∈ 0..cols(A1) - 1
      AAi,j ← 100 if A2i,j = 0
      AAi,j ← 100 if  $\frac{A1_{i,j}}{A2_{i,j}} > \frac{1-p}{p}$  otherwise
  AA

```

$$BB := \text{bayes}(AA1, AA2, 0.5)$$

3.3. Функция вложения границы разделения областей предпочтения

```

solve_curve(BB) :=
  k ← 0
  for i ∈ 1..rows(BB) - 2
    for j ∈ 1..cols(BB) - 2
      R1 ← (BBi-1,j-1 ≠ 0) + (BBi-1,j ≠ 0) + (BBi-1,j+1 ≠ 0)
      R2 ← (BBi,j-1 ≠ 0) + (BBi,j+1 ≠ 0) + (BBi+1,j-1 ≠ 0)
      R3 ← (BBi+1,j ≠ 0) + (BBi+1,j+1 ≠ 0)
      if [(BBi,j = 0) · (R1 + R2 + R3)]
        CXk ← i
        CYk ← j
        k ← k + 1
  (CX CY)T

```

```

C := solve_curve(BB)      CX := C0      CY := C1
s := 0..cols(X1) - 1      k := 0..cols(X2) - 1      j := 0..rows(CX) - 1

```

Граница разделения областей предпочтения

