

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Владимирский государственный университет имени Александра
Григорьевича и Николая Григорьевича Столетовых»

Кучерик А.О, Кутровская С.В., Бухаров Д.Н.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ВЫПОЛНЕНИЯ
ЛАБОРАТОРНЫХ РАБОТ
ПО ДИСЦИПЛИНЕ
«ОБРАБОТКА И АНАЛИЗ СИГНАЛОВ И СЦЕН»

Владимир 2016

Составители: Кучерик Алексей Олегович, Кутровская Стелла Владимировна, Бухаров Дмитрий Николаевич.

Методические указания для выполнения лабораторных работ по дисциплине «Обработка и анализ сигналов и сцен» / Владим. гос. уни-т имени Александра Григорьевича и Николая Григорьевича Столетовых; Кучерик Алексей Олегович, Кутровская Стелла Владимировна, Бухаров Дмитрий Николаевич. – Владимир: Изд-во ВлГУ, 2016. – 60с.

Предложены лабораторные работы по курсу «Обработка и анализ сигналов и сцен» для магистров по направлению 01.04.02 «Прикладная математика и информатика». Рассмотрены основы разделов моделирование сигналов, корреляционный, спектральный, вейвлет анализы, фрактальный и мультифрактальный анализ сигналов, обработка и анализ двумерных изображений в MATLAB. К лабораторным работам приведена краткая теория, что упрощает их выполнение.

Пособие выполнено в рамках государственного задания ВлГУ на 2014/13 на выполнение государственных работ в сфере научно-технической деятельности.

Рецензент – доктор физико-математических наук, с.н.с. отдела структурно-фазовых превращений Федерального государственного бюджетного учреждения науки Физико-технический институт Уральского отделения Российской Академии наук Карбань О.В.

Оглавление

ВВЕДЕНИЕ	5
ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ.....	5
1. ЛАБОРАТОРНАЯ РАБОТА №1 «МОДЕЛИРОВАНИЕ СИГНАЛОВ»..	6
Теория.....	6
1.1. Понятие сигнала. Классификация сигналов	6
1.2.Основные характеристики случайных сигналов	7
1.3.Задание тестовых сигналов в пакетах MATLAB.....	9
1.4.Предварительная статистическая обработка сигналов	14
Задания к работе:.....	15
Контрольные вопросы	16
2. ЛАБОРАТОРНАЯ РАБОТА №2 «КОРРЕЛЯЦИОННЫЙ АНАЛИЗ»....	17
Теория.....	17
Задания к работе:.....	19
Контрольные вопросы	20
3. ЛАБОРАТОРНАЯ РАБОТА № 3 «СПЕКТРАЛЬНЫЙ АНАЛИЗ»....	21
Теория.....	21
3.1. Фурье-преобразование сигнала	21
3.2.Спектральный анализ случайных сигналов	22
3.3.Преобразования Фурье тестовых сигналов в MATLAB	22
3.4.Фильтрация аналоговых сигналов.....	26
Задания к работе.....	27
Контрольные вопросы	29
4. ЛАБОРАТОРНАЯ РАБОТА № 4 «ВЕЙВЛЕТ АНАЛИЗ»	30
Теория.....	30
4.1. Разложение сигнала по вейвлетам.....	30
4.2.Энергетические характеристики сигналов.....	32

4.3. Вейвлет-анализ сигналов с помощью картин значений коэффициентов вейвлет-преобразований	33
4.4. Численный расчет энергетических характеристик сигналов	37
Задания к работе:	37
Приложение	38
Контрольные вопросы:	39
5. ЛАБОРАТОРНАЯ РАБОТА № 5 «ФРАКТАЛЬНЫЙ И МУЛЬТИФРАКТАЛЬНЫЙ АНАЛИЗ СИГНАЛОВ»	40
5.1. Определения фрактала и мультифрактала	40
5.2. Фрактальный анализ сигналов	41
5.3. Мультифрактальный анализ сигналов	43
Задания к работе:	45
Контрольные вопросы	46
6. ЛАБОРАТОРНАЯ РАБОТА № 6 «ОБРАБОТКА ДВУМЕРНЫХ ИЗОБРАЖЕНИЙ В MATLAB»	47
Теория	47
6.1. Обработка изображений в MATLAB	47
Задания к работе	54
Контрольные вопросы	54
7. ЛАБОРАТОРНАЯ РАБОТА № 7 «АНАЛИЗ ДВУМЕРНЫХ ИЗОБРАЖЕНИЙ В MATLAB. ВЫДЕЛЕНИЕ ГРАНИЦ»	55
Теория	55
7.1. Анализ изображений	55
Задания к работе	58
Контрольные вопросы	58
Список литературы	59

ВВЕДЕНИЕ

В методических указаниях приведены лабораторные работы по курсу «Обработка и анализ сигналов и сцен». Лабораторные работы сопровождаются краткой теорией, что делает более удобным их выполнение.

Для ознакомления с экспериментальной частью курса и выполнения практических заданий настоятельно рекомендуется знакомство с пакетом MATLAB версии 6.0 и выше.

Для облегчения освоения материала курса полезно иметь опыт работы в среде MATLAB. Успешное решение практических задач требует владения программированием на одном из языков высокого уровня

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Работа в лаборатории требует от студента соответствующей подготовки. Для выполнения лабораторной работы, кроме рассмотренной теоретической части также необходимо использовать литературу, которую рекомендует преподаватель на лекции.

В отчет по лабораторной работе необходимо внести:

- номер и название работы;
- цель работы;
- теоретическую часть;
- текст программы, таблицы, расчетные формулы, графики и т.д.
- вывод.

При работе в лаборатории следует быть внимательным и выполнять правила по технике безопасности. Инструкция по технике безопасности находится в лаборатории.

Работа заканчивается составлением краткого заключения (вывода), в котором следует указать:

- в какой программной среде реализована лабораторная работа;
- краткий анализ результатов.

1. ЛАБОРАТОРНАЯ РАБОТА №1 «МОДЕЛИРОВАНИЕ СИГНАЛОВ»

Теория

1.1. Понятие сигнала. Классификация сигналов

Под сигналом в общем случае понимают физический процесс, который осуществляет перенос информации во времени и пространстве. Сигналы описываются математическими моделями, отражающими их общие свойства. Примерами сигналов могут служить ток в цепи микрофона, воспринимающего речь, яркость луча электронно-лучевой трубки, интенсивность излучения лазера и др.

Математически сигналы описываются функциональными зависимостями $u(t)$, $f(s)$ и т. п., в которых аргументом чаще всего является время t или некоторая пространственная переменная s . Функции, описывающие сигналы, могут принимать как вещественные, так и комплексные значения.

Сигнал, описываемый функцией одной переменной, называется одномерным, а сигнал, описываемый функцией M ($M > 2$) независимых переменных, – многомерным.

Сигналы, значения которых можно предсказать в любой момент времени (в любой точке пространства), называются детерминированными. Сигналы, для которых невозможно точно предсказать значения, называют стохастическими или случайными. Сигналы, значения которых изменяются непрерывно при изменении непрерывной временной t или пространственной переменной s , называются непрерывными сигналами. Непрерывные сигналы часто называют аналоговыми сигналами.

Наряду с непрерывным способом передачи и преобразования сигналов, широко применяют дискретные способы, в которых в том или ином виде используется дискретизация сигналов. Дискретизация сигналов состоит в замене «непрерывных» значений дискретными значениями и может производиться как по времени (или пространственной переменной), так и по уровню (по значению величины сигнала), или по времени и уровню одновременно (рис. 1 [1]). В первом случае ее часто называют операцией получения выборки, во втором – квантованием. Уровни обычно отстоят друг от друга на постоянную величину, называемую шагом квантования по уровню.

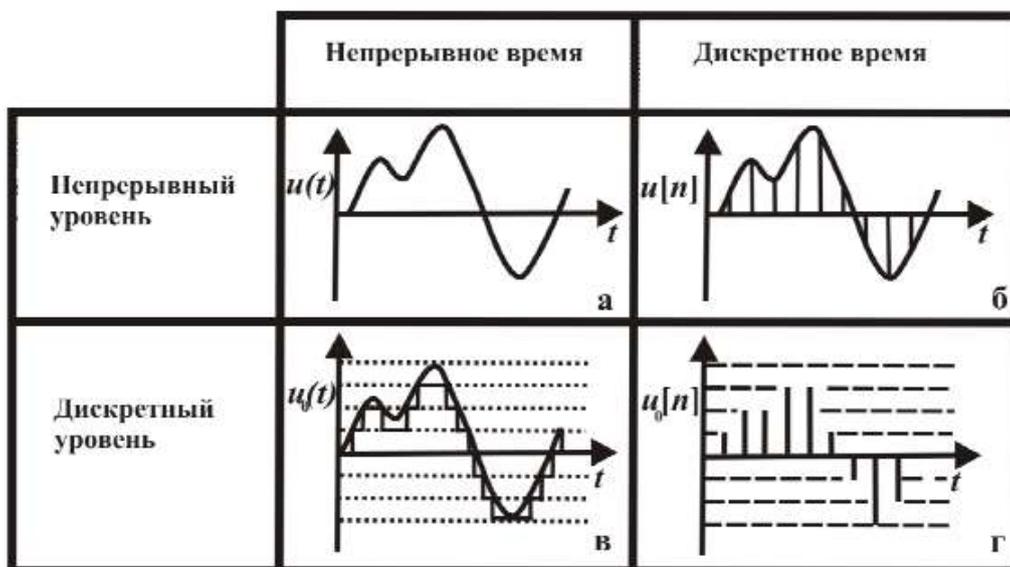


Рис. 1. Дискретизация и квантование сигналов.

Дискретизация сигнала по времени соответствует выделению значений сигнала при фиксированных значениях t . Обычно моменты взятия значений (отсчетов) сигнала отстоят друг от друга на величину T_0 , (или S_0), называемую интервалом или шагом дискретизации. Сигнал, получаемый в ходе указанного процесса, называется дискретным сигналом.

Дискретный сигнал представляется множеством точек nT_0 , в каждой из которых определено отсчетное значение сигнала $u[n] = u(nT_0)$. Таким образом, дискретизация по времени заменяет непрерывную функцию $u(t)$ решетчатой, которая определяется совокупностью выделенных ординат, следующих через равные интервалы T_0 .

Сигналы, дискретизированные как по времени, так и по уровню, называются цифровыми. Цифровые сигналы отличаются от дискретных тем, что для них отсчетные значения представлены в виде чисел. Для формирования цифровых сигналов применяют аналого-цифровые преобразователи (АЦП), которые выполняют дискретизацию аналогового сигнала по времени и уровню, а затем кодируют уровень сигнала, используя ту или иную систему счисления[4].

1.2. Основные характеристики случайных сигналов

Случайные сигналы описываются случайными функциями, значения которых для каждого аргумента представляются случайными величинами.

Случайной величиной называется величина, наблюдаемое значение которой зависит от случайных причин. Случайная величина характеризуется полностью, если указаны вероятности, с которыми она может принимать те или иные значения.

Случайную функцию времени называют случайным процессом.

Полная совокупность функций времени представляет собой ансамбль, и будет обозначаться $\{x(t)\}$, где любая функция $x(t)$ ансамбля есть выборочная функция случайного процесса.

При одном наблюдении случайного процесса получают определенную функциональную зависимость, которую называют реализацией.

Условно случайный процесс можно представить в виде совокупности реализаций.

Сечение случайного процесса $x(t)$ в момент времени t представляет случайную величину x , которая полностью характеризуется одномерной функцией плотности вероятности $f(x)$. Произведение $f(x)dx$ – вероятность того, что значение случайной величины будет находиться в интервале $(x, x + dx)$.

Используя одномерную функцию плотности вероятности, определим важнейшие характеристики случайных сигналов.

Среднее значение случайной величины X , определяется следующим образом:

$$\bar{X} = E[X] = \int_{-\infty}^{\infty} xf(x)dx.$$

Величина $E[X]$ называется математическим ожиданием X .

Средний квадрат случайной величины (или момент 2-го порядка):

$$\overline{X^2} = E[x^2] = \int_{-\infty}^{\infty} x^2 f(x)dx.$$

Дисперсией $D(X)$ случайной величины X называется средний квадрат отклонения этой величины от ее математического ожидания:

$$\mu_2 = \sigma_x^2 = D(X) = Var[X] = \overline{(X - \bar{X})^2} = \int_{-\infty}^{\infty} (x - \bar{X})^2 f(x)dx.$$

Величина σ_x – значение корня квадратного из дисперсии – называется стандартным или средним квадратическим отклонением.

Характеристикой взаимосвязи двух случайных величин является корреляционная функция. Если эти случайные величины являются выборочными значениями одного и того же случайного процесса, то указанная функция называется автокорреляционной (или просто корреляционной) функцией данного процесса, если же они принадлежат различным случайным процессам – взаимной корреляционной функцией.

Автокорреляционная функция определяется следующим образом:

$$R_x(t_1, t_2) = E[x_1, x_2] = \int_{-\infty}^{\infty} dx_1 \int_{-\infty}^{\infty} x_1 x_2 f(x_1, x_2) dx_2,$$

где $x_1 = X(x_1)$, $x_2 = X(x_2)$, x – двумерная плотность распределения вероятностей.

Существует еще один вид взаимной корреляционной функции, которую можно определить для тех же двух моментов времени. Пусть $y_1 = Y(t_1)$, $y_2 = Y(t_2)$. Тогда по определению:

$$R_{YX}(\tau) = E[y_1, x_2] = \int_{-\infty}^{\infty} dy_1 \int_{-\infty}^{\infty} y_1 x_2 f(y_1, x_2) dx_2.$$

1.3. Задание тестовых сигналов в пакетах MATLAB

При моделировании сигнал задается в виде функции одного или нескольких аргументов, причем функции разделяются на два типа:

1. встроенные функции;
2. функции, определяемые пользователем.

Применение функций обоих типов в расчетах совершенно одинаково, с тем исключением, что любую встроенную функцию можно использовать сразу в любом месте документа, а пользовательскую функцию необходимо предварительно определить в документе до момента вычисления ее значения.

Функции в пакете MATLAB – это имеющие специальные имена объекты, выполняющие определенные преобразования над своими аргументами и при этом возвращающие результаты этих преобразований. При этом результат вычисления функции с одним переменным параметром подставляется на место ее вызова, что позволяет использовать функции в математических выражениях.

Функции в общем случае имеют список аргументов (параметров), заключенных в круглые скобки. Если функция возвращает несколько значе-

ний, то она записывается в виде: $[Y_1, Y_2, \dots] = \text{func}(X_1, X_2, \dots)$, где Y_1, Y_2, \dots – список выходных аргументов и X_1, X_2, \dots – список входных аргументов (параметров), func – имя функции.

Со списком элементарных функций можно ознакомиться выполнив команду «helpfun» в командном окне (Commandwindow).

Для того чтобы построить график функции $y = f(t)$ в среде MATLAB, достаточно тем или иным способом сформировать 2 вектора одинаковой размерности – вектор значений аргумента t и вектор соответствующих значений функции y , а затем воспользоваться процедурой «plot». При вызове процедуры «plot», MATLAB автоматически создает окно с заголовком «Figure 1» с графиком заданной функции.

Пример 1. (детерминированный сигнал). Зададим сигнал в виде дискретной функции, меняющейся по гармоническому закону:

$$X_k(t) = A \sin\left(\frac{2\pi t k}{T} + \Psi\right),$$

где A – амплитуда сигнала, Ψ – начальная фаза, T – период сигнала, k – целое число, Δt – интервал времени

Создадим вектор n , воспользовавшись процедурой `linspace(a,b,K)`, которая генерирует вектор-строку из K точек, равномерно расположенных между точками a и b (вторая строка листинга).

Замечание. Можно создать вектор в MATLAB другим способом. Например, записав: `k=a:K:b`. Такая запись сформирует вектор-строку на интервале (a,b) с шагом K . Так, например, запись `k=linspace(1,1023,1023)` будет эквивалентна записи `k=1:1:1023`. В пятой строке листинга задается функция сигнала график, которой затем выводится на экран.

Пример 2. Зададим сигнал в виде возмущенной гармонической дискретной функции:

$$X_k = \cos\left(2\pi \frac{(0,5k)^n \Delta t}{T} + \Psi\right) + 1,$$

где k из $[0, K]$, показатель степени $n = 1,5$, Ψ – начальная фаза $\Psi = \pi / 3$, период сигнала $T = 100$, интервал времени $\Delta t = 1$.

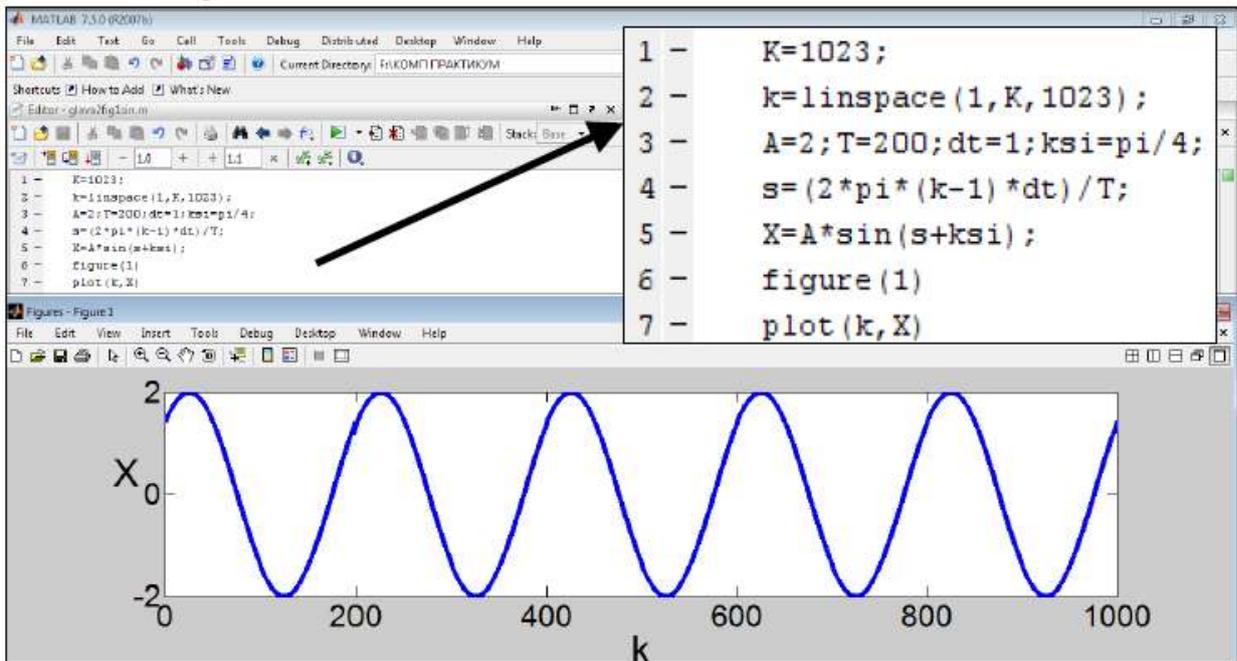


Рис. 2. Задание гармонического сигнала в пакете MATLAB.

Тогда листинг программы примет вид:

```

%Задание сигнала в виде возмущенной гармонической функции
clearall % очищаем память
K=120; % задаем параметры сигнала
k=linspace(0, K, 120);
n=1.5;
T=100;
ksi=pi/3;
dt=1;
z=(2*pi)*dt/T; % введем параметр z для удобной записи функции
% сигнала
X=cos(z*(k*0.5).^n+ksi)+1; % задаем формулу функции сигнала
figure(1) % выводим график на экран
plot(k, X)
xlabel('k') % подпись осей
ylabel('X')

```

На рис. 3 представлен график заданной функции.

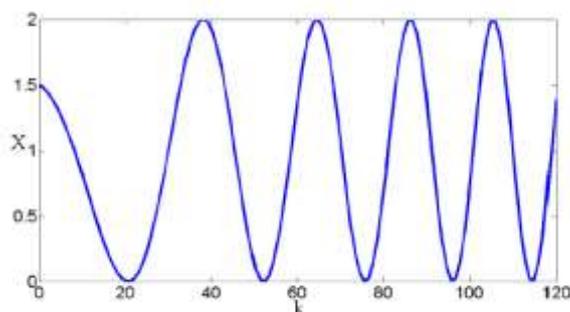


Рис. 3. Задание сигнала в виде возмущенной гармонической функции

При моделировании случайных сигналов в MATLAB можно использовать, например, такие встроенные функции, как «rand», «randn», «random»[3].

Функция «rand» позволяет генерировать случайные числа, равномерно распределенные в интервале от 2^{-53} до $1-2^{-53}$ (примерно [0,1]). Если у функции задается один скалярный аргумент, то функция «rand(n)» возвращает квадратную матрицу n-го порядка, элементами которой являются случайные числа из [0,1]. Функция «rand(n, m)» возвращает прямоугольную матрицу размерностью n на m со случайными числами. Обращение «rand(size(A))» позволяет сгенерировать матрицу случайных чисел, размерность которой совпадает с размерностью массива A. Для перехода к случайным числам из произвольного диапазона [a,b] можно воспользоваться преобразованием: $a + (b - a) * \text{rand}$.

Функция randn позволяет генерировать случайные числа, распределенные по нормальному закону, со следующими параметрами (среднее значение случайной величины 0, дисперсия 1). Форматы обращения к функции randn такие же, как и к функции rand. Если требуются случайные числа, распределенные по нормальному закону со смещенным центром m и заданной дисперсией s, то достаточно воспользоваться преобразованием: $m + \text{sqrt}(s) * \text{randn}$.

Функция Y=random(name,A,B,C) возвращает случайные числа Y с распределением вероятности (name) и параметрами этого распределения A, B, C. Название распределения name заключается в кавычки 'name'. Например: 'beta' (β-распределение), 'exp' (экспоненциальное распределение), 'logn' (логнормальное распределение), 'poiss' (распределение Пуассона). С полным списком возможных распределений можно ознакомиться, набрав в командном окне: help random.

Обратите внимание, что размер всех параметров распределения A, B, C, которые можно задать в виде массивов (матриц), должен совпадать с размером Y.

Пример 3. Зададим случайный сигнал в виде возмущенной гармонической дискретной функции с начальной фазой, которая имеет равномерное распределение вероятностей (или с равномерной плотностью вероятности) и остальными параметрами такими же, как в примере 2. Тогда для

задания сигнала достаточно в листинге примера 2, изменить задание фазы: $k\text{si}=\text{rand}(\text{size}(k))\cdot\pi/3$. На рис. 4 приведен график полученного сигнала.

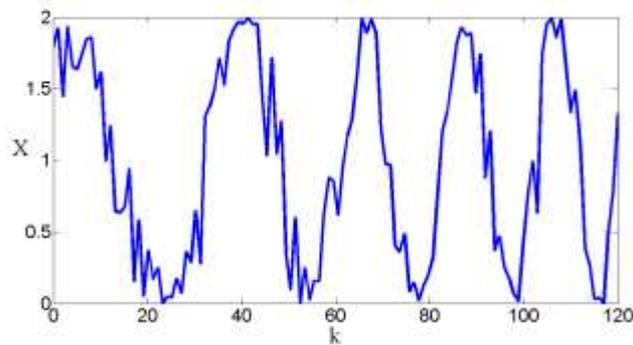


Рис. 4. Задание сигнала в виде возмущенной гармонической функции с равномерно распределенной начальной фазой

Пример 4. Зададим случайный сигнал в виде функции Вейерштрасса

$$X(t) = \sqrt{2}\sigma \frac{\left(1 - b^{2D-4}\right)^{1/2} \sum_{n=0}^N b^{(D-2)} \sin\left(2\pi s b^n t + \Psi_n\right)}{\left(1 - b^{(2D-4)(N+1)}\right)^{1/2}}.$$

При этом параметры сигнала: случайная фаза ψ равномерно на отрезке $[0, 2\pi]$, $\sigma = 3,3$; $b = 2,5$; $s = 0,005$; $D = 1,3$; $N = 10$.

Листинг задания такого сигнала будет иметь вид:

```
clearall % очищаем память
N=10; % задаем параметры сигнала
k=0:1:1023;
sigma=3.3;
b=2.5;
D=1.3;
s=0.005;
h=sigma*((2)^0.5);
h1=(1-b^(2*D-4))^0.5; % введем вспомогательные переменные для удобной записи сигнала
h2=(1-b^((2*D-4)*(N+1)))^0.5;
h3=(h*h1)./h2;
X=zeros(1,length(k)); % создадим матрицу-строку из k нулевых элементов
for n=1:N+1
c1=2*pi*s*k*b.^(n-1);
c2=(D-2).*(n-1);
f=(b.^c2);
c=random('unif',0,n-1);
ksl=2*pi*c;% случайная фаза изменяется от 0 до 2 pi
term=h3.*f.*sin(c1+ksl);
X=X+term; % функция сигнала
end
```

```
figure(1) % вывод графика на экран
clf
plot(k,X);
```

График одной из реализаций сигнала приведен на рис. 5

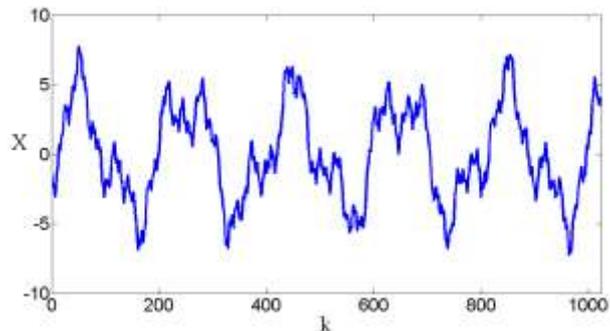


Рис. 5. Задание сигнала в виде функции Вейерштрасса

1.4. Предварительная статистическая обработка сигналов

Предварительную обработку случайных сигналов можно провести в вычислительном пакете MATLAB. Приведем один из возможных вариантов листинга программы анализа случайного сигнала с нормальным распределением вероятностей ($E = 0$, дисперсия 1):

```
K=1000;
X=randn(K,1);
figure(1)
plot(X) % вывод графика сигнала
% построение гистограммы
int=-4:0.2:4;
n=hist(X,int);
figure(2)
hist(X,int) % вывод гистограммы на экран
figure(3)
plot(int,n) % вывод профиля (огibaющей) гистограммы
% Вычисление основных статистических характеристик
M=mean(X); % выборочное среднее значение
V=var(X); % дисперсия
% стандартное отклонение
S=sqrt(V); % по определению
S1=std(X); % с помощью встроенной функции
amin=min(X); % минимальное значение
amax=max(X); % максимальное значение
R=amax-amin; % размах
mod=mode(X); % мода
med=median(X); % медиана
% коэффициент асимметрии
Skew=skewness(X);
% коэффициент эксцесса
Kurt=kurtosis(X);
```

Замечание. При построении гистограммы можно воспользоваться другой формой записи функции «hist», позволяющей в явном виде задавать число ее элементов:

hist(X,bin) % bin – число столбиков гистограммы.

На рис. 6 приведена гистограмма заданного случайного сигнала и график ее огибающей.

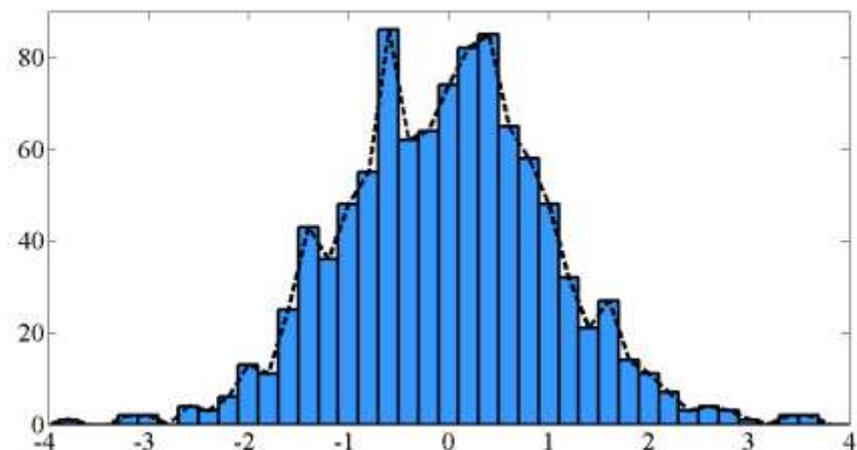


Рис. 6. Гистограмма ограниченного случайного сигнала с нормальным распределением вероятности и график ее огибающей (пунктир).

Задания к работе:

1. Постройте график структуры сигнала $X_k(k)$, заданного в виде возмущенной гармонической функции

$$X_k = \frac{A}{\sqrt{k+1}} \cos\left(2\pi \frac{k\Delta t}{T/\sqrt{k+1}} + \Psi_k\right),$$

где $A = 2$, $T = 200$, $\Delta t = 1$, $k = 0, 1, \dots, 1023$, $n = 5$. Случайная фаза Ψ_k равномерно распределена на отрезке $\left[0, \frac{\pi}{2}\right]$.

2. Задайте сигнал X_k в виде функции Вейерштрасса со следующими параметрами: случайная фаза Ψ равномерно на отрезке $[0, 2\pi]$, $\sigma = 3$; $b = 4$; $s = 0,005$; $D = 1$; $N = 5$, $k = 0, 1, \dots, 1023$.

3. Задайте сигнал X_k с нормальным распределением вероятности со следующими параметрами: математическое ожидание $E = 1,5$, стандартное отклонение $\sigma = 1$, $k = 0, 1, \dots, 1000$.

4. Задать гармонический сигнал с амплитудой 5, периодом 200 и случайной фазой, равномерно распределенной на отрезке $\left[0, \frac{\pi}{2}\right]$. Для заданного сигнала построить гистограмму и ее огибающую (число столбцов 30).

5. Задайте случайный сигнал с однородным распределением вероятности. Для заданного сигнала определите: выборочное среднее значение, выборочную дисперсию, относительную дисперсию, стандартное отклонение, относительное стандартное отклонение, медиану, моду, размах.

Контрольные вопросы

1. Что такое сигнал?
2. Как классифицируются сигналы?
3. Приведите определение цифрового сигнала.
4. Приведите пример детерминированного и случайного сигналов.
5. Как определяется дисперсия и стандартное отклонение случайного сигнала?
6. Приведите характеристики взаимосвязи двух случайных величин.
7. Как связаны огибающая гистограмма распределения случайной величины и функция плотности распределения вероятности?
8. Как определяется мода?
10. Как определяется медиана?

2. ЛАБОРАТОРНАЯ РАБОТА №2 «КОРРЕЛЯЦИОННЫЙ АНАЛИЗ»

Теория

Автокорреляционная функция (АКФ, CF – *correlation function*) сигнала $s(t)$ конечного по энергии является количественной интегральной характеристикой формы сигнала, выявления в сигнале характера и параметров взаимной временной связи отсчетов. АКФ определяется интегралом от произведения двух копий сигнала $s(t)$, сдвинутых относительно друг друга на время τ :

$$B_s(\tau) = \int_{-\infty}^{\infty} s(t)s(t+\tau)dt.$$

Взаимная корреляционная функция (ВКФ) разных сигналов (*cross-correlation function, CCF*) описывает как степень сходства формы двух сигналов, так и их взаимное расположение друг относительно друга по координате (независимой переменной).

$$B_{su}(\tau) = \int_{-\infty}^{\infty} s(t)u(t+\tau)dt.$$

Для расчета АКФ и ВКФ в MATLAB применяется функция «`xcorr()`». В зависимости от указываемых параметров функция рассчитывает АКФ либо ВКФ для определенных значений сдвига τ либо на всем времени существования сигналов, применяя к результату различные варианты нормирования. [1]

В общем случае функция записывается в следующем виде:

$$[c, \tau] = \text{xcorr}(x1, x2, [\tau_min \tau_max], \text{norm})$$

c – массив значений АКФ или ВКФ, полученный в результате работы функции; τ – массив значений сдвигов по оси времени для данной АКФ или ВКФ; $x1, x2$ – исходные сигналы (в случае расчета АКФ, $x2$ не указывается); $[\tau_min \tau_max]$ – пределы изменения сдвига по времени τ (вместо двух значений можно указать одно, тогда расчет будет произведен для значений $[-\tau_max \tau_max]$, можно не указывать параметр вообще, тогда расчет будет произведен на всем времени существования сигналов); norm – метод нормирования результирующей функции: ‘none’ – без нормировки, ‘coeff’ – функция нормируется так, чтобы при нулевом сдвиге значение было равно 1; также предусмотрены значения ‘biased’ и ‘unbiased’

$C = \text{xcorr}(A, B, 'unbiased')$ – для векторов A и B с одинаковой длиной $M > 1$ возвращает оценку взаимной корреляции в виде вектора C с длиной $2*M-1$ и заданной нормировкой 'unbiased';

$C1 = \text{xcorr}(A)$ – для вектора A возвращает оценку автокорреляции.

Параметр `norm` может не указываться, тогда по умолчанию принимается `norm = 'none'`. [9]

Пример 1. Вычислим автокорреляционную функцию сигнала R_n , заданного функцией Вейерштрасса.

Листинг программы примет вид:

```
clearall % очищаем память
N=4; % задаем параметры сигнала
k=0:1:1023;
sigma=3.3;b=2.5;D=1.3;s=0.005;
h=sigma*((2)^0.5);
h1=(1-b^(2*D-4))^0.5; % введем вспомогательные переменные для
% удобной записи сигнала
h2=(1-b^((2*D-4)*(N+1)))^0.5;
h3=(h*h1)./h2;
X2=zeros(1,length(k)); % создадим матрицу-строку из k нулевых
% элементов
for m=1:N+1
c1=2*pi*s*k*b.^(m-1);
c2=(D-2).*(m-1);
f=(b.^c2);
c=random('unif',0,m-1);
c3=2*pi*c;% случайная фаза равномерно изменяется от 0 до 2pi
term=h3.*f.*sin(c1+c3);
X2=X2+term;
end
[R, lags] = xcorr(X2,100,'unbiased');% вычисление функции
% автокорреляции (lags ограничивает область значений R)
figure(1)
clf
plot(lags,R) % вывод графика на экран
```

На рис. 7 представлен фрагмент вычисленной функции R_n .

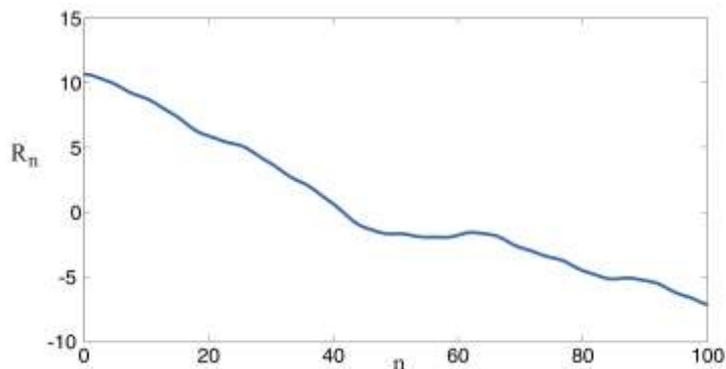


Рис.7. Фрагмент автокорреляционной функции заданного сигнала

Задания к работе:

Задание 1

1. Вычислите автокорреляционную функцию для случайного сигнала, заданного в виде функции с экспоненциальным распределением вероятности и параметрами $\tau = 4$, $k = 0, 1, \dots, 1023$. Постройте график автокорреляционной функции в зависимости от числа n , где $n = 0, 1, \dots, 100$.

2. Вычислите коэффициент и функцию взаимной корреляции сигналов, заданных в виде: X_1 – функция, заданная в пункте 1; X_{2k} – функция с нормальным распределением вероятности и параметрами: $E=1$, $\sigma = 0,7$, $k = 0, 1, \dots, 1023$.

Приведите зависимость функции взаимной корреляции при изменении числа n , где $n = 0, 1, \dots, 200$.

Задание 2

1. Задать вектор времени.
2. Задать прямоугольный импульс на векторе времени. Для этого ознакомиться с описанием функции «rectpuls». См. Приложение
3. Получить АКФ для данного сигнала, построить графики исходной и автокорреляционной функций с помощью функций «subplot» и «plot».
4. Объяснить результат с точки зрения теории.
5. Повторить пункты 2 – 4 для синусоидального сигнала.
6. Задать два периодических прямоугольных сигнала со скважностью* 30%, смещенных друг относительно друга на полпериода.
7. Рассчитать ВКФ этих сигналов.
8. Объяснить результат с точки зрения теории.
9. Задать синусоидальный сигнал и добавить к нему шум с амплитудой в два раза большей, чем амплитуда синусоиды.
10. Построить АКФ полученного сигнала.
11. Объяснить результат с точки зрения теории.
12. Предложить способы использования наблюдаемого явления.
13. Повторить пункты 11 – 12 для ВКФ зашумленного сигнала и исходной синусоиды.

Приложение

* – Скважность – отношение периода следования (повторения) к длительности импульса прямоугольного импульса.

Описание функции «rectpulse»

rectpuls – прямоугольный импульс вида $\text{rect}(t) = \begin{cases} 1, & |t| \leq 1/2 \\ 0, & |t| \geq 1/2 \end{cases}$

Описание:

$$y = \text{rectpuls}(t, \text{width}),$$

где t – вектор значений времени;, width – ширина (длительность) импульса (по умолчанию $\text{width}=1$, в этом случае функция «rectpuls» производит результат, соответствующий математической функции rect).

Пример. Сформировать прямоугольный импульс:

```
>>t = -0.6:0.01:1;  
>> y = rectpuls(t);  
>>figure,plot(t,y);  
>>ylim([0 1.5])
```

Контрольные вопросы

1. Как определяются корреляционные характеристики сигналов?
2. Что характеризует степень сходства сигналов?
3. Определите область допустимых значений коэффициента корреляции.
4. Какие характеристики сигналов характеризует автокорреляционная функция?

3. ЛАБОРАТОРНАЯ РАБОТА № 3 «СПЕКТРАЛЬНЫЙ АНАЛИЗ»

Теория

Спектральный анализ базируется на выполнении преобразований Фурье и заключается в разложении сигнала на его частотные или спектральные составляющие, а также оценке их спектральных характеристик – амплитуды, фазы, спектральной плотности мощности и др.

3.1. Фурье-преобразование сигнала

При изучении детерминированных процессов применяется гармонический анализ: ряды Фурье – для периодических, интеграл Фурье – для аperiodических процессов.

Функция $F(\omega)$ называется преобразованием Фурье или спектральной плотностью сигнала $f(t)$:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt.$$

где ω имеет смысл частоты соответствующей составляющей сигнала.

Обратное преобразование Фурье переводит спектр $F(\omega)$ в исходный сигнал $f(t)$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

Преобразование Фурье периодической функции $f(t)$ на интервале $[0, T]$, описывающей форму сигнала, использует в качестве базисных функций синус и косинус, представленные комплексной экспонентой $e^{i\omega t} = \cos(\omega t) + i \sin(\omega t)$:

$$f(t) = \sum_{-\infty}^{\infty} C_m e^{im\omega_0 t},$$

где $\omega_0 = \frac{2\pi}{T}$.

Коэффициенты C_m имеют вид [4]:

$$C_m = \frac{1}{2\pi} \int_0^T f(t) e^{im\omega_0 t} dt$$

3.2. Спектральный анализ случайных сигналов

Отметим, что применение классического гармонического анализа к случайным процессам невозможно, так как спектральные плотности, рассчитанные по спектрам Фурье их реализаций, не имеют конечных значений на любых частотах. Однако, можно обобщить гармонический анализ, усредняя спектральные разложения, полученные из отдельных выборок.

Для стационарного случайного процесса $X(t)$ справедлива теорема Винера-Хинчина, которая устанавливает связь между его энергетическим спектром $F(\omega)$ и корреляционной функцией $R(\tau)$ с помощью преобразования Фурье:

$$F(\omega) = 2 \int_{-\infty}^{\infty} R(\tau) e^{i\omega\tau} d\tau = 4 \int_0^{\infty} R(t) \cos(\omega t) dt .$$
$$R(\tau) = \frac{1}{4\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega\tau} d\omega = \frac{1}{2\pi} \int_0^{\infty} F(\omega) \cos(\omega\tau) d\omega .$$

Функцию частоты $F(\omega)$ называют энергетическим спектром стационарного случайного процесса. Этот спектр дает только усредненную картину распределения энергии процесса по частотам элементарных гармонических составляющих, но не учитывает их фазовой структуры.

3.3. Преобразования Фурье тестовых сигналов в MATLAB

Для практической реализации спектрального анализа Фурье используется алгоритм быстрого преобразования Фурье (БПФ) [1]. Этот алгоритм реализован в нескольких встроенных функциях Matlab.

В системе MatLab для вычисления прямого преобразования Фурье одномерных сигналов используется функция `fft`:

Синтаксис

$v = \text{fft}(y)$ – вычисляет вектор прямого преобразования Фурье:

$$v(k) = \sum_{j=1}^n y(j) \omega_n^{(j-1)(k-1)},$$

где $n = \text{length}(y)$ – длина вектора исходных (комплексных или действительных) данных, $\omega_n = e^{-\frac{2\pi i}{n}}$.

если y – матрица, преобразование производится для каждого ее столбца по отдельности;

$v = \text{fft}(y, N)$ – вычисляет прямое дискретное преобразование Фурье (ДПФ); перед вычислением предварительно приводят исходные данные к размеру N , урезая их или дополняя нулями.

Для вычисления обратного преобразования Фурье одномерных сигналов используется функция ifft :

$y = \text{ifft}(v)$ – вычисляет вектор обратного преобразования Фурье:

$$y(j) = \frac{1}{n} \sum_{k=1}^n v(k) \omega_n^{(j-1)(k-1)}.$$

$y = \text{ifft}(v, N)$ – вычисляет обратное ДПФ; предварительно приводит исходные данные к размеру N , урезая их или дополняя нулями.

Для вычисления прямого ДПФ двумерного сигнала используется функция

fft2 , обратного ДПФ – ifft2 :

Синтаксис

$y = \text{fft2}(x, N)$ – вычисляет прямое ДПФ для матрицы x ;

$y = \text{fft2}(x, N)$ – вычисляет прямое ДПФ; перед вычислением предварительно приводит исходные данные к размеру N , урезая их или дополняя нулями.

$x = \text{ifft2}(y)$ – вычисляет обратное ДПФ для матрицы y ;

$x = \text{ifft2}(y, N)$ – вычисляет обратное ДПФ; перед вычислением предварительно приводит исходные данные к размеру N , урезая их или дополняя нулями.

При выводе спектральных графиков иногда желательно, чтобы компонента с нулевой частотой вектора x находилась в центре, а диапазон отображаемых частот простирался от $-\frac{f_D}{2}$ до $\frac{f_D}{2}$. Сделать это позволяет функция « fftshift », которая меняет местами половины переданного ей вектора.

Синтаксис

$y = \text{fftshift}(x)$

В рассчитанном преобразовании Фурье присутствуют наряду с вещественными коэффициентами и мнимые. Поэтому для визуализации

ДПФ необходимо определить модуль ДПФ. Для этого и используют функцию `abs` –определение модуля аргумента.

Синтаксис

$$y = \text{abs}(x)$$

Для оценки энергии на различных частотах применяется спектральная плотность мощности (СПМ), равная произведению вектора ДПФ на сопряженный ему вектор. Для вычисления сопряженного вектора используется функция «`conj`».

Синтаксис

$$y = \text{conj}(x)$$

Поскольку преобразования Фурье переводит сигнал из временной области в частотную для вывода графика спектра надо задать вектор нормализованных частот. Размер этого вектора должен совпадать с размером вектора ДПФ (размер вектора вычисляет функция «`size`») [2].

Приведем пример реализации быстрого преобразования Фурье в MATLAB для дискретной функции, меняющейся по гармоническому закону.

```
clear all % очищаем память
K=1024;k=1:1:1024;
A=2;T=200;dt=1;ksi=pi/4;
s=(2*pi*(k-1)*dt)/T;
X1=A*sin(s+ksi);% задаем функцию сигнала
Y=fft(X1);% прямое БПФ
F=ifft(Y);% обратное БПФ
f=(0:length(Y)-1)/length(Y);% вектор частот
figure(1) % вывод графика
clf
subplot(311),plot(X1)
xlim([0 1024]);
subplot(312),plot(f,abs(Y))
ylim([0 1200]);
subplot(313),plot(F)
xlim([0 1024]);
ylim([-2 2]);
```

Ниже приведены примеры формирования прямоугольных импульсов и импульса вида $\frac{\sin(pt)}{pt}$ и их спектров и СПМ.

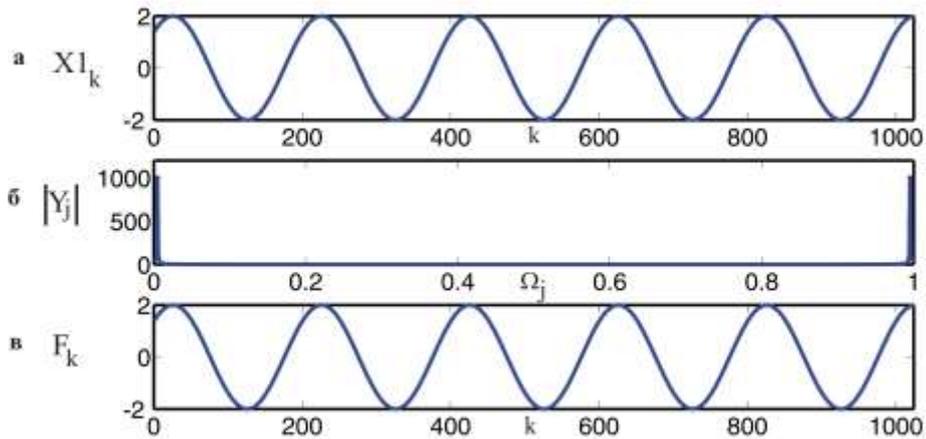


Рис. 8. Графики сигнала (а) и его прямого (б) и обратного (в) преобразований Фурье.

Пример1. Сформировать спектр и СПМ прямоугольного сигнала:

```
>> Fs = 8e3;
>> t = -40e-3:1/Fs:40e-3;
>> T = 20e-3;
>> s = rectpuls(t,T);
>>figure,plot(t,s)
>>ylim([-2 2])
>> Fs = fft(s);
>>Pfs = Fs.*conj(Fs);
>>f = 1000/256*(1:max(size(Fs))); % вектор нормализованных частот
>> F = fftshift(Pfs);
>>figure,plot(f,F),grid % СПМ
>>xlim([1150 1350]);
>> F = fftshift(abs(Fs));
>>figure,plot(f,F),grid % преобразование Фурье
>>xlim([1000 1500]);
```

Пример 2.Сформировать спектр и СПМ импульса вида $\frac{\sin(pt)}{pt}$.

```
>> t = -4:0.1:4;
>> y = sinc(t);
>>figure,plot(t,y);
>> F = fft(y);
>> f = 1000/256*(1:max(size(F)));
>> F1 = fftshift(abs(F));
>>figure,plot(f,F1); % преобразование Фурье
>> F1 = fftshift(F.*conj(F));
>>figure,plot(f,F1); % СПМ
```

Пример 3.Использовать двумерное преобразование Фурье для сигнала вида $z = L2(x) * L2(y)$

```
>> t = -1:0.05:1;
>> y = tripuls(t);
>>plot(t,y)
>> [X,Y] = meshgrid(t);
```

```

>> Z1 = tripuls(X);
>> Z2 = tripuls(Y);
>> Z3 = Z1.*Z2;
>>figure,mesh(X,Y,Z3);
>> F = fft2(Z3);
>> Fa = fftshift(abs(F));
>> f1 = 1000*(1:max(size(F)))/526;
>> [f1,f2] = meshgrid(f1);
>>figure,mesh(f1,f2,Fa).

```

Преобразование Фурье часто используется для выделения основных частот сигнала, зашумленного помехами.

Пример 4. Сформировать сигнал, являющийся суммой двух синусоид с частотами 50 и 130 Гц и случайной компоненты с нулевым средним и с помощью ДПФ определить, какие частоты присутствуют в анализируемом сигнале.

```

>>t = 0:0.001:0.25; % задание вектора времени
>>x = sin(2*pi*50*t)+0.5*sin(2*pi*130*t) % задание сигнала
>> y = x+randn(size(t));
>>figure,plot(t,y);

```

Полученная зависимость не позволяет определить основные частоты сигнала. Для их определения рассчитаем и построим график СПМ, используя ДПФ сигнала:

```

>> Y = fft(y);
>>Pyy = Y.*conj(Y);
>> f = 1000/256*(0:127);
>>figure,plot(f(1:50),Pyy(1:50))
>>title('Power spectral density')
>>xlabel('Frequency(Hz)')

```

3.4. Фильтрация аналоговых сигналов

Под фильтрацией подразумевается выделение полезного сигнала из его смеси с мешающим сигналом – шумом. Наиболее распространенный тип фильтрации – частотная фильтрация. Если известна область частот, занимаемых полезным сигналом, достаточно выделить эту область и подавить те области, которые заняты шумом.

Рассмотрим принцип частотной фильтрации.

Сначала синтезируется исходный сигнал $f(t)$, представленный N отсчетами вектора υ . Затем к этому сигналу присоединяется шум с помощью генератора случайных чисел (функция «rand») и формируется вектор из N отсчетов зашумленного сигнала. Используя прямое БПФ, сигнал с шумом преобразуется из временной области в частотную, что создает век-

тор f_1 из $\frac{N}{2}$ частотных составляющих. Затем выполняется фильтрующее преобразование, эффективность которого оценивается параметром a . Фильтрующее преобразование удобно выполнять с помощью функции Хевисайда (возвращает 1, если $x \geq 0$; иначе 0). Отфильтрованный сигнал (вектор g) подвергается обратному БПФ и создает вектор выходного сигнала h . Сравнение временных зависимостей исходного и выходного сигналов, показывает, что выходной сигнал почти полностью повторяет входной и в значительной мере избавлен от высокочастотных шумовых помех, маскирующих полезный сигнал.

Задания к работе

Для выполнения преобразований Фурье нужно:

1. задать функцию тестового сигнала
2. провести процедуру прямого и обратного преобразования Фурье с использованием алгоритма быстрого преобразования Фурье (БПФ);
3. вывести результаты в виде графиков:
 - функция сигнала (в зависимости от числа отсчетов);
 - прямое преобразование Фурье (в зависимости от частоты);
 - обратное преобразование Фурье (в зависимости от числа отсчетов).

Задание 1

Осуществить прямое и обратное Фурье-преобразование тестовых сигналов

$x_k(t)$, представленных в виде следующих функций:

$$\text{№1 } X_k = A \sin \left(2\pi \frac{(0,3k)^n \Delta t}{T} + \Psi_k \right) + 2,$$

где $A=3$, $T=200$, $\Delta t=1$, $k = 0,1,\dots,1023$, $n=1,45$, $\Psi = \frac{\pi}{6}$.

$$\text{№2 } X_k = A \sin \left(2\pi \frac{(0,3k)^n \Delta t}{T} + \Psi_k \right) + 2,$$

где случайная фаза Ψ_k равномерно распределена на отрезке $\left[0, \frac{\pi}{6} \right]$.

Остальные параметры сигнала такие же, как для сигнала №1.

№3. Случайная функция $X_k(t)$ с экспоненциальным распределением вероятности и параметром $\tau=1$.

Задание 2

Сформировать одиночный прямоугольный импульс и его Фурье-образ. Частота дискретизации равна 10^{-3} , вектор отсчетов времени задать от 0 до $40 \cdot 10^{-3}$ с шагом, равным обратной величине частоты дискретизации, амплитуда равна 5 В, длительность импульса равна $20 \cdot 10^{-3}$. Выдать график прямоугольного импульса, (отлимитировать значения по оси y от -6 до 6 командой `ylim([-6 6])`, задаваемой после команды `plot`). Вывести сетку. Значения частоты для вывода Фурье преобразования задать $f = 1000 \cdot (1:41)/512$. Спектральный график вывести с нулевой частотой, находящейся в центре, для чего использовать соответствующую функцию.

Задание 3

Сформировать одиночный сигнал вида $\sin(x)/x$ и его СПМ. Вектор времени задать от -5 до 5 с шагом, равным 0.01. Значения частоты для вывода СПМ задать равными $f = 1000 \cdot (1:1001)/512$. Спектральный график вывести с нулевой частотой, находящейся в центре, для чего использовать соответствующую функцию. Полученный график отлимитировать от 900 до 1 000.

Задание 4

Построить графики трехмерных функций и их Фурье-преобразований:

а) $Z1 = \text{rect}(x) \cdot \text{rect}(y)$;

б) $Z2 = \text{sinc}(x) \cdot \text{sinc}(y)$,

используя результаты выполнения заданий 2 и 3 соответственно.

Задание 5

Выполнить фильтрацию функции $f(t)$ с помощью БПФ:

- синтезировать функцию $f(t) = |\sin(t)| + |\sin(2t)|$ в виде полезного сигнала, представленного 128 отсчетами вектора v ;
- к полезному сигналу v присоединить шум с помощью функции «`rand(rand - 1)`» и сформировать вектор из 128 отсчетов зашумленного сигнала s ;

- преобразовать сигнал с шумом s из временной области в частотную, используя прямое БПФ (функция «fft»). В результате получится сигнал f из 64 частотных составляющих;
- выполнить фильтрующее преобразование с помощью функции Хевисайда (параметр фильтрации $a = 2$ – в Matlab функция Хевисайда вызывается командой «heaviside(x)», x - параметр фильтрации);
- с помощью функции «ifft» выполнить обратное БПФ и получить вектор выходного сигнала h ;
- построить графики полезного сигнала v и сигнала, полученного фильтрацией зашумленного сигнала s .

Контрольные вопросы

1. Дайте определения прямому и обратному преобразованиям Фурье.
2. Как осуществляется процедура дискретного преобразования Фурье?
3. Какие функции вычисляют ДПФ в MatLab?
5. Какие дополнительные функции используются для того, чтобы получить график преобразования Фурье и спектральную плотность мощности?
6. Для чего применяют БПФ к зашумленному сигналу?

4. ЛАБОРАТОРНАЯ РАБОТА № 4 «ВЕЙВЛЕТ АНАЛИЗ»

Теория

4.1. Разложение сигнала по вейвлетам

Вейвлет-преобразование одномерного сигнала состоит в его разложении по базису, сконструированному из обладающей определенными свойствами функции (вейвлета) с помощью масштабных изменений и переносов. Каждая из функций этого базиса характеризует как определенную пространственную (временную) частоту, так и ее локализацию в физическом пространстве (времени) [1, 8]:

$$s(t) = \sum_k C_k \psi_k(t),$$

где C_k – коэффициенты разложения, несущие информацию о сигнале, $\psi_k(t)$ – базисные функции.

Вейвлеты характеризуются своими временными и частотными образами. Временной образ определяется некоторой $\psi(t)$ – функцией, а частотный задается ее фурье-образом $\psi(\omega)$, который задает огибающую спектра вейвлета. В основе непрерывного вейвлет-преобразования лежит использование двух непрерывных и интегрируемых по всей числовой оси t или x функций: с нулевым значением интеграла 0, определяющая детали сигнала и порождающая детализирующие коэффициенты; масштабирующая (или скейлинг-функция) с единичным значением интеграла, определяющая грубое приближение (аппроксимацию) сигнала и порождающая коэффициенты аппроксимации.

Функция $\Psi(t)$ создается на основе той или иной базисной функции $\psi_0(t)$, которая, как и $\psi(t)$, определяет тип вейвлета [8]:

$$\psi(t) = a^{-\frac{1}{2}} \psi_0\left(\frac{t-b}{a}\right),$$

где b – величина смещения по оси времени (координаты), a – коэффициент масштабирования.

Вейвлеты классифицируются по виду и особенностям базисной функции $\psi_0(t)$, а также по имени ученого, впервые предложившего тот или иной вейвлет. Приведем примеры вейвлет-функций $\psi(t)$ [8].

Пример 1. Вейвлет «мексиканская шляпа» (МНАТ-вейвлет)

$$\psi(t) = \left(\frac{2}{\sqrt{3}} \pi^{\frac{1}{4}} \right) (1-t^2) \exp\left(-\frac{t^2}{2}\right).$$

Пример 2. Вейвлет Хаара (ортогональный вейвлет):

$$\psi(t) = \begin{cases} 1, & 0 \leq t \leq \frac{1}{2} \\ -1, & \frac{1}{2} < t < 1 \\ 0, & t < 0, t > 1 \end{cases}$$

Пример 3. Вейвлет Морле (Morlet):

$$\psi(t) = \exp\left(-\frac{t^2}{2}\right) \cos(5t).$$

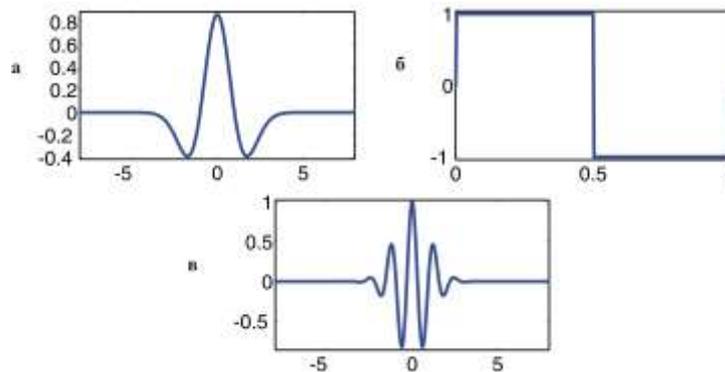


Рис. 9. Примеры вейвлетов: (а) «мексиканская шляпа», (б) вейвлет Хаара, (в) вейвлет-Морле

Прямое непрерывное вейвлет-преобразование сигнала $s(t)$, ограниченного в области \mathbb{R} , задается путем вычисления вейвлет-коэффициентов:

$$W(a,b) = \int s(t) a^{\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) dt.$$

Обратное непрерывное вейвлет-преобразование осуществляется по формуле реконструкции во временной области, которая имеет ряд форм. Приведем эту формулу в виде использованном в пакете расширения системы MATLAB – WaveletToolbox[8]:

$$s(t) = \frac{1}{C_\psi} \iint W(a,b) a^{\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) \frac{dadb}{a^2},$$

где C_ψ – нормализующий коэффициент.

4.2. Энергетические характеристики сигналов

Полная энергия сигнала f может быть записана через амплитуды вейвлет-преобразования в виде:

$$E_f = \int f^2(t) dt = C_\psi^{-1} \iint W^2(a, b) \frac{dabd}{a^2}.$$

Плотность энергии сигнала $E_w(a, b) = W^2(a, b)$ - характеризует энергетические уровни исследуемого сигнала $f(t)$ в пространстве $(a, b) = (\text{масштаб}, \text{время})$.

Локальный спектр энергии позволяет проанализировать временную динамику передачи энергии процесса по масштабам – обмен энергией между составляющими процесс компонентами разного масштаба в любой заданный момент времени. Зная плотность энергии $E_w(a, b)$, можно с помощью окна определить локальную плотность энергии в точке b_0 (или t_0):

$$E_\xi(a, t_0) = \int E_w(a, b) \xi\left(\frac{b-t_0}{a}\right) db.$$

Оконная функция ξ “поддерживает” диапазон около 0 t и удовлетворяет равенству:

$$\int \xi(b) db = 1.$$

Если в качестве ξ выбрать функцию Дирака, то локальный спектр энергии примет вид:

$$E_\delta(a, t_0) = W^2(a, t_0).$$

Глобальный спектр энергии. Полная энергия распределена по масштабам в соответствии с глобальным спектром энергии коэффициентов вейвлет-преобразования:

$$E_w(a) = \int W^2(a, b) db = E_w(a, b) db.$$

Его называют также скалограммой или дисперсией вейвлет-преобразования.

Мера локальной перемежаемости:

$$I_w(a, t) = \frac{E_w(a, t)}{\langle E'_w(a, t) \rangle_t}.$$

– мера локальных отклонений от среднего поля спектров на каждом масштабе: она позволяет определить степень неравномерности распределения энергии по масштабам (угловыми скобками здесь обозначено усреднение).

Мера контрастности:

$$C_w(a,t) = \frac{E_w(a,t)}{E'_w(a,t)},$$
$$E'_w(a,t) = \int_{a'=0}^{a'=a} E_w(a',t) da',$$

позволяет определять даже самые малые изменения в сигнале, когда необходимо, например, выявить структурированность слабого сигнала или слабые вибрации на фоне крупной структуры.

4.3. Вейвлет-анализ сигналов с помощью картин значений коэффициентов вейвлет-преобразований

Картина значений вейвлет-коэффициентов представляет значение коэффициентов вейвлетов в плоскости масштаб (номера коэффициентов) – время (или пространственная координата). Внизу картины значений вейвлет-коэффициентов расположены коэффициенты с малыми номерами, дающие детальную картину сигнала, а сверху – с большими номерами, дающие огрубленную картину сигнала. Результатом вейвлет-анализа заданного сигнала с помощью картин вейвлет-коэффициентов являются графики:

- 1) график функции заданного сигнала;
- 2) картины значений вейвлет-коэффициентов;
- 3) проекции картины значений вейвлет-коэффициентов при фиксированном значении коэффициента a ;
- 4) скелетон (картина линий локальных экстремумов) картины значений коэффициентов вейвлетов.

Для построения картин значений коэффициентов вейвлет-преобразований сигналов используется вычислительный пакет WaveToolbox среды MATLAB [8], который запускается командой `wavemenu`. В результате появится окно со списком разделов вейвлет-преобразований (рис. 10).

Для получения справки по какому-либо типу вейвлета при работе в командном режиме MATLAB достаточно исполнить команду `waveinfo('type')`, указав тип вейвлета. Для просмотра же вейвлетов достаточно исполнить команду `wavemenu` и в появившемся окне со списком разделов ВП нажать кнопку `WaveletDisplay`. Нажатие этой кнопки выводит окно просмотра вейвлетов, в котором имеется возможность просмотра:

общей информации о вейвлетах, выбранного вейвлета (с именем «Name») и информации о нем.



Рис. 10. Главное меню вычислительного пакета WaveToolbox в MATLAB

При вейвлет-анализе заданного сигнала можно использовать набор встроенных вейвлетообразующих функций (на панели справа меню Continuouswavelet 1-D) или задать необходимый вейвлет с помощью инструмента вейвлет-менеджер – wavemngr. Для просмотра встроенных вейвлетов воспользуйтесь кнопкой WaveletDisplay в окне главного меню WaveletToolboxMainMenu.

Для анализа тестовых сигналов с помощью картин значений вейвлет-коэффициентов выберите кнопку главного меню Continuouswavelet 1-D. Появится окно, в котором производится вейвлет-анализ заданного сигнала с помощью непрерывных вейвлетов.

Задания тестовых сигналов производится следующим образом:

Нажмите

- file в меню Continuous wavelet 1-D;
- выберите Loadsignal;
- укажите путь к файлу, в котором задается тестовый сигнал.

Файл с заданным сигналом должен быть сохранен в mat-формате (см Приложение).

В MATLAB вейвлет-коэффициенты $W(a,b)$ обозначены $C(a,b)$.

Пример 1. Проведем вейвлет-анализ с помощью картин вейвлет-коэффициентов для сигнала $X_k = \frac{2}{\sqrt{k+1}} \cos\left(2\pi \frac{k}{T/\sqrt[k+1]} + \frac{\pi}{3}\right)$, где параметры сигнала $k = 0, 1, \dots, 255$ на основе вейвлета Хаара.

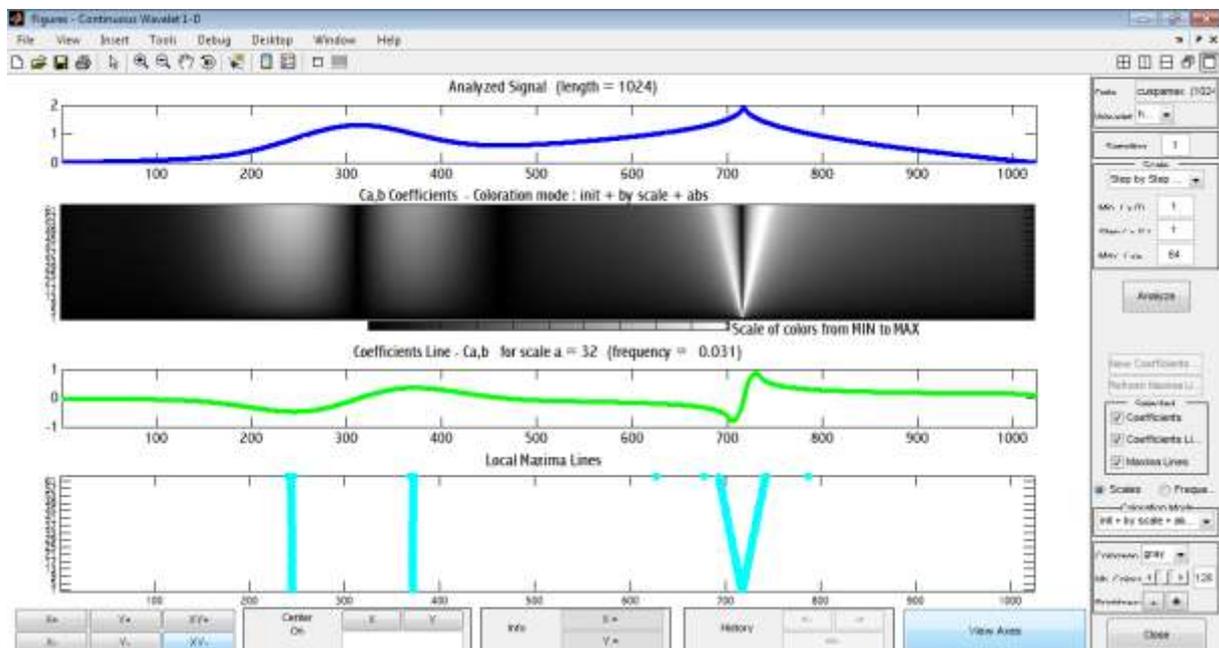


Рис.11. Вейвлет-анализ на основе вейвлетообразующей функции Хаара тестового сигнала.

Локальным особенностям (нарушениям гладкости) на картине вейвлет-коэффициентов отвечают вертикальные полосы, выходящие из точки, где находится особенность (рис. 11). Пикам сигналов соответствует сгущение светлых областей вейвлет-картин значений коэффициентов, а впадинам сгущение темных полос. Чем резче выражена особенность сигнала, тем сильнее она выделяется на картине значений вейвлет-коэффициентов. Анализ тестовых сигналов с использованием комплексных непрерывных вейвлетов производится в окне главного меню:

ComplexContinuouswavelet1-D (рис. 10). Здесь используются два метода анализа заданного сигнала по модулю $(A = \text{Re}^2 + \text{Im}^2)$ и углу $\varphi = \text{arctg}(\text{Im}/\text{Re})$, где Im и Re – соответственно мнимая и действительная часть анализируемой функции.

В приводимых ниже примерах прямое ВП реализуется функцией «cwt», которая может быть представлена в нескольких формах, например: `COEF = cwt(S, SCALES, «wname» PLOTMODE, XLIM)`, где строка S – сигнал, строка SCALES – задание диапазона и шага изменения параметра, а строка «wname» – имя (тип) вейвлета, строка PLOTMODE – настройка цвета: «lvl» – окраска шаг за шагом, «glb» – окраска с учетом всех коэффициентов, «absvil» или «lvlabs» – окраска шаг за шагом с использованием абсолютных значений коэффициентов, строка XLIM – строка переменных настройки.

Пример 2. Гармоническое колебание

```
functiongarm
t = 0:0.00001:0.0004; A1 = 1; F1 = 10000; a1 = 45;
s = A1*soc(2*pi*F1*t+a1);
figure (1); plot(t,s); axis([0 0.0004 -3 3]); grid on;
subplot(211), plot(t,s); title('Сигнал S(t)')
subplot(212), c = cwt(s, 1:2:32, 'mexh', 'abslvl', [0 10]);
title('Вейвлет-спектр '); xlabel('Временной сдвиг, b');
ylabel('Временной масштаб, a');
end
```

И хотя спектрограмма гармонического колебания, представленная на рис. 12, особой выразительностью не отличается, на ней отчетливо видны переходы сигнала через нуль (темный тон) и экстремальные точки (светлый тон).

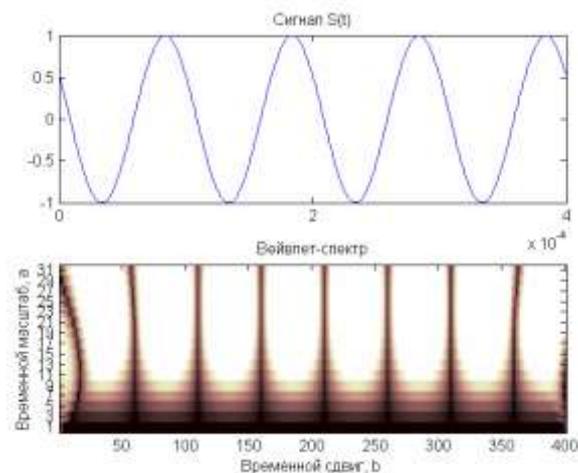


Рис. 12 Спектрограмма гармонического колебания

4.4. Численный расчет энергетических характеристик сигналов

Для вычисления энергетических характеристик заданного сигнала нужно записать формулы энергетических характеристик в виде конечных сумм в вычислительном пакете MATLAB и отобразить результаты в виде графиков.

Запишем интегральные формулы в виде сумм:
глобальный спектр энергии:

$$E_w = \sum_{b=0}^{K-1} |W(a,b)|^2;$$

мера локальной перемежаемости:

$$I_w = \frac{K |W(a,b)|^2}{\sum_{b=0}^{K-1} |W(a,b)|^2};$$

мера контрастности:

$$C_w = \frac{K |W(a=p,b)|^2}{\sum_{a=1}^p |W(a,b)|^2}.$$

Пример 3. Приведем пример построения графиков основных энергетических характеристик (рис 13) для сигнала, заданного функцией Вейерштрасса со следующими параметрами $\sigma = 3,3$; $b = 2,5$; $s = 0,005$; $D = 1,5$; $N = 2$; $K = 255$, случайная фаза Ψ_n распределена равномерно на отрезке $[0, 2\pi]$ (рис. 12). Для нахождения функции $\Psi_{a,n}$, использовалась вейвлет-тообразующая функция «мексиканская шляпа» [4].

Задания к работе:

Задание 1

Сформировать сигнал в виде прямоугольного импульса с шумом. Построить график сигнала и вельвет-спектра. В качестве вейвлета использовать `what`.

Задание 2

Провести вейвлет-анализ с помощью картин вейвлет-коэффициентов для сигналов, заданных функцией Вейерштрасса с параметрами $N = 6$, $D = 1,8$ и $N=6$, $D=1,2$. Построение проекций картин значений вейвлет-

коэффициентов осуществлять при фиксированных значениях коэффициента $a = 8, 12$. Объясните различия в структуре сигналов. Провести анализ на основе вейвлета «мексиканская шляпа».

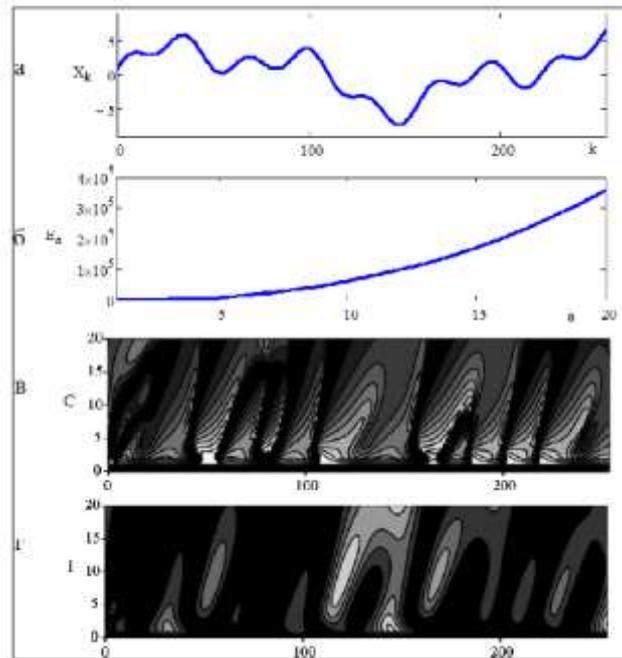


Рис. 13 График основных энергетических характеристик для сигнала, заданного функцией Вейерштрасса

2. Определить глобальный спектр энергии вейвлет-преобразования функции Вейерштрасса с параметрами $N = 6$, $D = 1,2$ и дать ему объяснение. Определить меру локальной перемежаемости для заданного сигнала.

Приложение

Задание сигналов в wavemenu вычислительного пакета WaveToolbox среды MATLAB осуществляется с помощью файлов с расширением mat [8].

Для создания mat-файла нужно воспользоваться следующей процедурой:

```
savefile = 'test001.mat';
Z=математическая формула сигнала;
save(savefile,'Z')
```

В результате создается файл с именем test001 и расширением mat, в который будет записан сигнал Z.

Контрольные вопросы:

1. Как производится разложение сигнала по вейвлетам?
2. Приведите примеры вейвлетообразующих функций.
3. Что позволяют определить меры контрастности и локальной пере-
межаемости?
4. Приведите определение скалограммы.
5. Как определяются локальный и глобальный спектр энергии?
6. Что такое скелетон вейвлет-преобразования сигнала?
7. В каких случаях эффективно применение вейвлет-
преобразований?

5. ЛАБОРАТОРНАЯ РАБОТА № 5 «ФРАКТАЛЬНЫЙ И МУЛЬТИФРАКТАЛЬНЫЙ АНАЛИЗ СИГНАЛОВ»

5.1. Определения фрактала и мультифрактала

Фракталами называются геометрические объекты: линии, поверхности, пространственные тела, имеющие сильно изрезанную форму и обладающие свойством самоподобия [4].

При описании свойств фрактала важную роль играет такая его характеристика как фрактальная размерность. Дадим общее определение этой величины. Пусть d – обычная Евклидова размерность пространства, в котором находится наш фрактальный объект ($d = 1$ – линия, $d = 2$ – плоскость, $d = 3$ – трехмерное пространство). Покроем теперь этот объект целиком d -мерными "шарами" радиуса l . Предположим, что нам потребовалось для этого не менее чем $N(l)$ шаров. Тогда, если при достаточно малых l величина $N(l)$ меняется по степенному закону [4]:

$$N(l) \sim \frac{1}{l^D},$$

то D – называется размерностью Хаусдорфа-Безиковича или фрактальной размерностью этого объекта.

Используя понятие фрактальной размерности, Мандельброт дал более строгое, чем приведенное выше, определение фрактала. Согласно этому определению фрактал представляет собой объект, размерность Хаусдорфа-Безиковича которого больше его топологической размерности (0 – для россыпи точек, 1 – для кривой, 2 – для поверхности и т. д.).

Формулу можно переписать также в виде:

$$D = - \lim_{l \rightarrow \infty} \frac{\ln N(l)}{\ln l}.$$

Это и служит общим определением фрактальной размерности D . В соответствии с ним величина D является локальной характеристикой данного объекта.

Мультифракталы – это неоднородные фрактальные объекты для полного описания которых, в отличие от регулярных фракталов, недостаточно введения всего лишь одной величины фрактальной размерности D , а необходим целый спектр таких размерностей число которых, вообще говоря, бесконечно. Причина этого заключается в том, что наряду с чисто гео-

метрическими характеристиками, определяемыми величиной D , такие фракталы обладают и некоторыми статистическими свойствами.

5.2. Фрактальный анализ сигналов

Фрактальный анализ сигналов проводится в тех случаях, когда необходимо установить, в какой степени в их поведении проявляются фрактальные признаки. При этом в качестве основного критерия наличия у сигналов фрактальных свойств выступает выполнимость зависимости, характеризующей поведение их структурной функции (математическое ожидание приращения). Приведем выражение этой характеристики в рамках модели обобщенного броуновского движения (ОБД) [4]:

$$E[X(t_2) - X(t_1)] = \sqrt{\frac{2}{\pi} \sigma (t_2 - t_1)^{qH}}.$$

Фрактальная размерность сигнала связана с параметром Херста соотношением в рамках ОБД:

$$D = 2 - H. (*)$$

При малых H функция сигнала сильно изрезана, а при больших – H имеет весьма плавный (хотя и не гладкий) характер.

Если построенные в двойном логарифмическом масштабе графики зависимости структурной функции от величины приращения времени хорошо аппроксимируются прямой в большом диапазоне изменения временных приращений, можно говорить о фрактальности исследуемого сигнала. По тангенсу угла наклона прямой можно определить параметр H . По установленному значению параметра Херста легко определяется фрактальная размерность D по формуле (*). Отметим, что при характеристике фрактальности сигнала помимо значения H необходимо указывать область скейлинга – интервал, в пределах которого график структурной функции хорошо аппроксимируется прямой.

Пример 1. Проведем фрактальный анализ сигнала, заданный функцией Вейерштрасса при следующих параметрах: $\sigma = 3,3$; $b = 2,5$; $s = 0,005$; $D = 1,3$; $N = 10$; Ψ_n – случайная фаза, равномерно распределенная (с постоянной вероятностью) на отрезке $[0, 2\pi]$.

Поскольку на практике обработку сигналов в той или иной программной среде удобнее проводить с помощью индексированных (дискретных) функций, перейдем от функции $X(t)$ к функции X_k . Время t связано с индексом $k = 1, 2, 3, \dots$ соотношением $t = k\Delta t$, где Δt – временной интервал между значащими точками сигнала.

Найдем структурную функцию сигнала согласно ее определению:

$$S_m = E[|X_{k+m} - X_k|] = \frac{1}{K-m} \sum_{k=1}^{K-m} |X_{k+m} - X_k|.$$

Усреднение проводится по всем значениям k .

Построенный в двойном логарифмическом масштабе график, характеризующий поведение структурной функции сигнала X_k , приведен на рис 14. По оси ординат отложена величина $L_m = \log_2 S_m$.

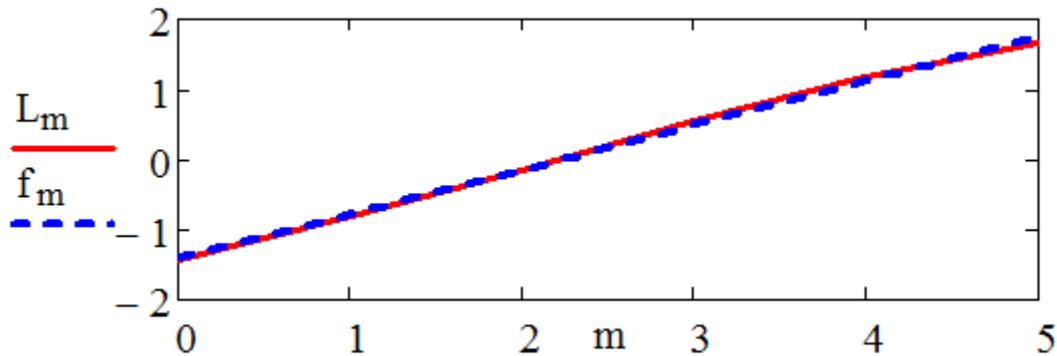


Рис. 14. График структурной функции сигнала и его линейной аппроксимации (пунктир)

На рис. 14 также приведена функция f_m , представляющая линейную аппроксимацию зависимости L_m .

Хорошо видно, что для данного сигнала поведение величины L_m хорошо аппроксимируется линейной зависимостью во всем задаваемом диапазоне изменения $m = 0, \dots, 5$. Этот результат подтверждает факт фрактальности сигналов, заданных с помощью функций Вейерштрасса.

Для оценки степени аппроксимации структурной функции L_m линейной зависимостью используется корреляционный коэффициент r :

$$r = \frac{\sum_m (x_m - \bar{x})(y_m - \bar{y})}{\left[\sum_m (x_m - \bar{x})^2 \sum_m (y_m - \bar{y})^2 \right]^{1/2}},$$

где $x_m = m$, $y_m = L_m$, \bar{x} , \bar{y} – выборочные средние значения. В данном случае значение коэффициента корреляции близко к 1: $r = 0,99$, что соответствует хорошей аппроксимации функции L_m .

Параметр Херста определяется тангенсом угла наклона графика функции f_m (рис. 14) и для заданного сигнала равен $H = 0,63$. Соответст-

вующая ему фрактальная размерность равна $D = 1,37$, что несколько отличается от заданного значения. Это отличие обусловлено конечностью анализируемого модельного сигнала и ограниченностью числа содержащихся в нем гармоник.

5.3. Мультифрактальный анализ сигналов

Идея мультифрактального анализа состоит в разложении исследуемого множества со сложной статистикой по множествам однородных фракталов с четко выраженной фрактальной размерностью. Рассмотрим упрощенный способ мультифрактального анализа, основанный на оценке скейлинговых свойств обобщенной структурной функции [4]:

$$S_{m,q} = E \left[|X_{k+m} - X_k|^q \right] = \frac{1}{K-m} \sum_{k=1}^{K-m} |X_{k+m} - X_k|^q,$$

где q – любое положительное число.

При проведении мультифрактального анализа используется следующее соотношение: $S_{m,q} \propto m^{\tau(q)+1}$, (должен стоять знак пропорции \sim , вместо квадрата) являющиеся обобщением на мультифрактальные сигналы. Величина $\tau(q)$ называется скейлинговой экспонентой. Если сигнал подчиняется модели ОБД, то $\tau(q)$ связана с параметром Херста H соотношением:

$$\tau(q) = H_q - 1.$$

В случае произвольных сигналов $\tau(q)$ находят из угла наклона графиков зависимостей $S_{m,q}$, от m , построенных в двойных логарифмических координатах для различных значений q .

По известной зависимости $\tau(q)$ определяют обобщенные фрактальные размерности $D_q = \frac{\tau(q)}{q-1}$ (размерности Реньи) и спектр сингулярностей

(функция мультифрактального спектра) $D(h) = qh(q) - \tau(q)$, где $h = \frac{d\tau}{dq}$ –

локальный параметр Херста. Величины $D_{q=1} = D_1$ и $D_{q=2} = D_2$ называют информационной и корреляционной размерностью соответственно. Их можно определить по формулам:

$$D_1 = D(h(1)) \text{ и } D_2 = 2h(2) - D(h(2)).$$

Рассматриваемый метод является приближенным, поскольку он позволяет точно определять лишь часть спектра сингулярностей при $q \geq 0$. Вторая часть спектра может быть определена лишь приблизительно, с помощью зеркального отображения рассчитанной кривой относительно вертикальной оси, проходящей через максимум.

Пример 2. Проведем мультифрактальный анализ сигнала X_{1k} , заданного функцией Вейерштрасса с размерностью $D = 1,5 + 0,5 \sin(2\pi * rs * k)$ при следующих параметрах: $\sigma = 3,3$, $b = 2,5$, $s = 0,005$, $r = 2$, $N = 7$, $k = 0,1, \dots, 1023$.

Зададим параметры $q = 0,0.49, \dots, 9.8$ и $m = 0, \dots, 5$. По формуле определяем обобщенную структурную функцию $S_{m,q}$, для заданного сигнала X_{1k} . Для определения зависимости $\tau(q)$ построим вспомогательную скейлинговую экспоненту в двойном логарифмическом масштабе $\zeta_1 = \log_2 S_{1,q} - \log_2 S_{0,q}$ и ее аппроксимацию $\zeta = \zeta_1(1)q^p k_1$, где p и k_1 – положительные коэффициенты. На рис. 15 приведены полученные графики функций ζ_1 и ζ при $p = 0,48$ и $k_1 = 0,98$.

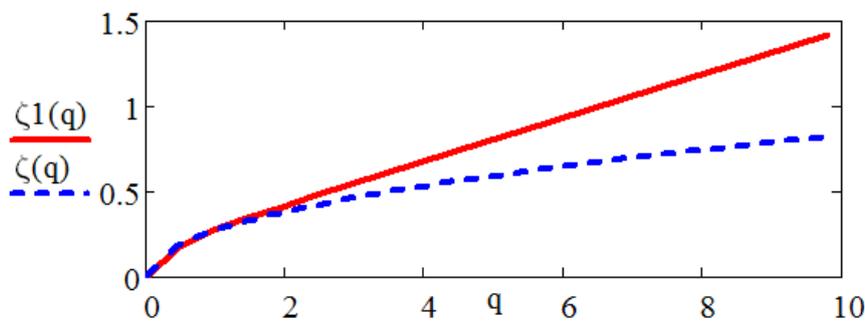


Рис. 15 Графики вспомогательных функций ζ_1 и ζ

По графику на рис. 15 видно, что при $q \leq 2$ достигается хорошая аппроксимация вспомогательной скейлинговой экспоненты ζ_1 . Теперь найдем скейлинговую экспоненту $\tau(q) = \zeta(q) - 1$ (рис. 16).

По зависимости $\tau(q)$ определяем размерности Реньи D_q и спектр сингулярностей $D(h) = qh(q) - \tau(q)$, где локальный параметр Херста вычисля-

ем по формуле: $h = \frac{\tau(q + \Delta q) - \tau(q)}{\Delta q}$. В данном случае $\Delta q = 0,49$. На рис. 17 приведена полученная часть спектра сингулярности сигнала.

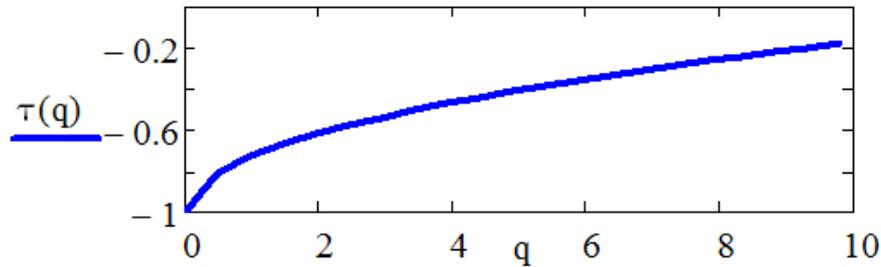


Рис. 16. График скейлинговой экспоненты $\tau(q)$

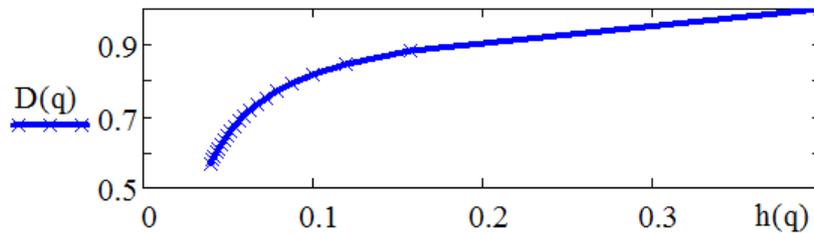


Рис. 17. Часть спектра сингулярности сигнала, рассчитанная на основе структурной функции.

Определим значения информационной D_1 и корреляционной D_2 размерностей по формулам, приведенным выше: $D_1 = 0,93$ и $D_2 = -0,76$ [4].

Задания к работе:

Задание 1

1. Построить сигналы на основе мультифрактальной функции Вейерштрасса с параметрами: $r = 2,50$ и $\sigma = 3,1$, $b = 3$, $s = 0,005$, $N = 7$, $k = 0,1, \dots, 1023$. Случайная фаза $\Psi_n = \frac{\pi}{2}$ распределена равномерно (с постоянной вероятностью) на отрезке $[0, 2\pi]$. Фрактальная размерность изменяется по закону: $D = 1,5 + 0,5 \cos(2\pi srk)$. Выявить различие в их поведении.

2. Определите размерности Реньи и локальные параметры Херста заданных сигналов.

Задание 2

1. Построить сигнал на основе мультифрактальной функции Вейерштрасса с параметрами: $\sigma = 3,1$, $b = 3$, $s = 0,005$, $N = 7$, $k = 0,1, \dots, 1023$, $r = 5$. Случайная фаза Ψ_n распределена равномерно (с постоянной вероятностью) на отрезке $[0, 2\pi]$. Фрактальная размерность изменяется по закону: $D_k = 1,5 + 0,5 \sin(2\pi srk)$. Определить величину скейлинговой экспоненты $\tau(q)$.

2. Определите значения информационной D_1 и корреляционной D_2 размерностей полученного сигнала.

Задание 3

1. Построить фрактальный сигнал на основе функции Вейерштрасса с параметрами $D = 1,5$, $\sigma = 3,2$, $b = 4$, $s = 0,005$, $N = 7$, $k = 0,1, \dots, 1023$. Случайная фаза Ψ_n распределена равномерно (с постоянной вероятностью) на отрезке $[0, 2\pi]$. Определите параметр Херста.

2. Сопоставьте значение заданной фрактальной размерности со значением, получающимся на основе вычислений с помощью структурной функции. Постройте график структурной функции сигнала и его линейной аппроксимации. Оцените степень аппроксимации структурной функции.

Контрольные вопросы

1. Дайте определение фрактала.
2. Сформулируйте определение мультифрактала.
3. В каких пределах меняется фрактальная размерность сигналов?
4. В каких пределах меняется параметр Херста сигналов?
5. В чем состоит физический смысл параметра Херста сигналов?
6. Как определяется структурная функция сигнала?
7. Чем отличаются персистентные и антиперсистентные сигналы?
8. В чем состоит идея мультифрактального анализа?
9. Как определяются размерности Реньи?
10. Как определяются величины информационной и корреляционной размерностей?

6. ЛАБОРАТОРНАЯ РАБОТА № 6 «ОБРАБОТКА ДВУМЕРНЫХ ИЗОБРАЖЕНИЙ В MATLAB»

Теория

6.1. Обработка изображений в MATLAB

Рассмотрим некоторые основы обработки изображений. В качестве примера покажем статистическую обработку изображения, а также предварительную обработку для получения улучшенного результата. Предобработка включает формирование равномерного фона на изображении и преобразование изображения в бинарное. Таким образом, обработка изображения состоит из некоторых основных шагов:

- Шаг 1: Считывание и отображение изображения.
- Шаг 2: Оценка и аппроксимация значений пикселей фона.
- Шаг 3: Аппроксимация и просмотр поверхности фона.
- Шаг 4: Создание изображения с равномерным фоном.
- Шаг 5: Повышение контраста изображения.
- Шаг 6: Создание бинарного изображения.
- Шаг 7: Определение числа объектов на изображении.
- Шаг 8: Анализ изображения.
- Шаг 9: Отображение матрицы меток как псевдоцветового индексного изображения
- Шаг 10: Измерение свойств объектов на изображении.
- Шаг 11: Вычисление статистических данных об объектах на изображении.

Шаг 1: Считывание и отображение изображения

Считаем и визуализируем полутоновое изображение «rice.png» [5].

```
I = imread('rice.png');  
imshow(I)
```

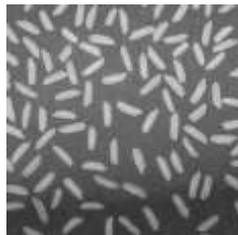


Рис.18. Полутоновое изображение rice.png

Шаг 2: Оценка и аппроксимация значений пикселей фона

Рассматриваемое изображение характеризуется неравномерной за-светкой фона по полю изображения. Поэтому сначала с помощью опера-ции морфологического раскрытия оценим интенсивность фона. При вы-полнении операции морфологического раскрытия используется структур-ный элемент.

В примере используется функция «`imopen`», которая выполняет мор-фологическое раскрытие. При этом также используется функция «`strel`» для создания структурного элемента в виде диска с радиусом 15.

```
background = imopen(I, strel('disk', 15));
```

Для просмотра и оценки фона используются следующие команды

```
figure, imshow(background)
```

Шаг 3: Аппроксимация и просмотр поверхности фона

Используем команду «`surf`» для отображения поверхности фона. С помо-щью команды «`surf`» создается цветная параметрическая поверхность, ко-торая дает возможность просматривать прямоугольную область исследуе-мого изображения. Функция «`surf`» работает с данными, которые представ-лены в формате «`double`». Поэтому, прежде чем применять эту функцию, данные необходимо перевести в формат «`double`».

```
figure, surf(double(background(1:8:end, 1:8:end))), zlim([0 255]);  
set(gca, 'ydir', 'reverse');
```

В данном примере отображается только каждый восьмой пиксель по каждому направлению. Существует также возможность установки мас-штаба и других параметров.

На основе этого можно проводить анализ фона исследуемого изо-бражения.

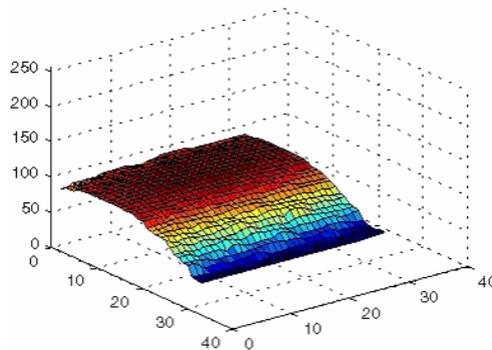


Рис. 19 Отображения поверхности фона

Шаг 4: Создание изображения с равномерным фоном

Для создания изображения с более равномерным фоном, вычтем изображение фона «background» из исходного изображения I.

```
I2 = imsubtract(I,background);
```

Отообразим полученное изображение с более равномерным фоном.

```
figure, imshow(I2)
```

Шаг 5: Улучшение контраста на обрабатываемом изображении

После вычитания полученное изображение будет иметь более равномерный фон и будет более темным. Используем функцию «imadjust» для повышения контраста изображения.

```
I3 = imadjust(I2);
```

Функция «imadjust» увеличивает контраст изображения путем растяжения значений интенсивностей динамического диапазона. Для более детальной информации см. описание функции «imadjust» в системе Matlab. Отообразим улучшенное изображение I3.

```
figure, imshow(I3);
```

Шаг 6: Создание бинарного изображения

Бинарное изображение можно создать, используя функцию «thresholding». Однако функция «graythresh» автоматически определяет подходящий порог, который используется для преобразования полутонового изображения в бинарное. Функция «im2bw» выполняет это преобразование.

```
level = graythresh(I3);
```

```
bw = im2bw(I3,level);
```

```
figure, imshow(bw)
```



Рис 20. Бинарное изображение

Возвращаемое функцией «im2bw» бинарное изображение «bw» представлено в формате «logical». В этом можно убедиться, воспользовав-

шись функцией «whos». Приложение «ImageProcessingToolbox» использует логические массивы для представления бинарных изображений.

```
whos
```

Реакцией системы MATLAB на эту функцию будет

Name	Size	Bytes	Class
I	256x256	65536	uint8 array2
I2	256x256	65536	uint8 array
I3	256x256	65536	uint8 array
background	256x256	65536	uint8 array
bw	256x256	65536	logical array
level	1x1	8	doublearray

В общей сложности используется 327681 элемента, которые занимают 327688 байт.

Шаг 7: Определение числа объектов на изображении

После преобразования изображения в бинарное, можно использовать функцию «bwlabel» для определения числа объектов (зерен риса) на изображении. Функция «bwlabel» отмечает все компоненты на бинарном изображении «bw» и возвращает их число в виде значения «numObjects».

```
[labeled,numObjects] = bwlabel(bw,4);
```

```
numObjects
```

```
ans = 101
```

Точность результата зависит от некоторых факторов, включая

1. размер объектов;
2. соприкасаются ли между собой объекты (в этом случае они могут определяться как один объект);
3. точность аппроксимации фона.
4. выбор связности.

Шаг 8: Анализ матрицы меток

Для проведения анализа формирования матрицы меток, возвращаемой функцией «bwlabel», рассмотрим значения пикселей изображения. Есть несколько путей получения значений пикселей изображения. Например, используя функцию «imshow», можно просматривать небольшие порции изображения. Другой путь состоит в использовании приложения «PixelRegionTool» для просмотра значений пикселей. Отобразим матрицу меток с использованием функции «imshow»,

```
figure, imshow(labeled);
```

Запуск приложения «PixelRegionTool».

```
impixelregion
```

По умолчанию оно автоматически связывает себя с изображением в текущем окне просмотра. Приложение «PixelRegionTool» рисует прямоугольник с центром в видимой части изображения. Расположение этого прямоугольника определяет какие пиксели будут отображаться в «PixelRegionTool». При перемещении прямоугольника значения отображаемых в окне «PixelRegionTool» пикселей обновятся.

На следующем изображении представлен «ImageViewer» с прямоугольником «PixelRegion», размещенным на границе двух рисовых гранул (рис. 21). Отметим, что значения пикселей фона и отдельных гранул являются одинаковыми. Так, в частности, значение фона равно 0.

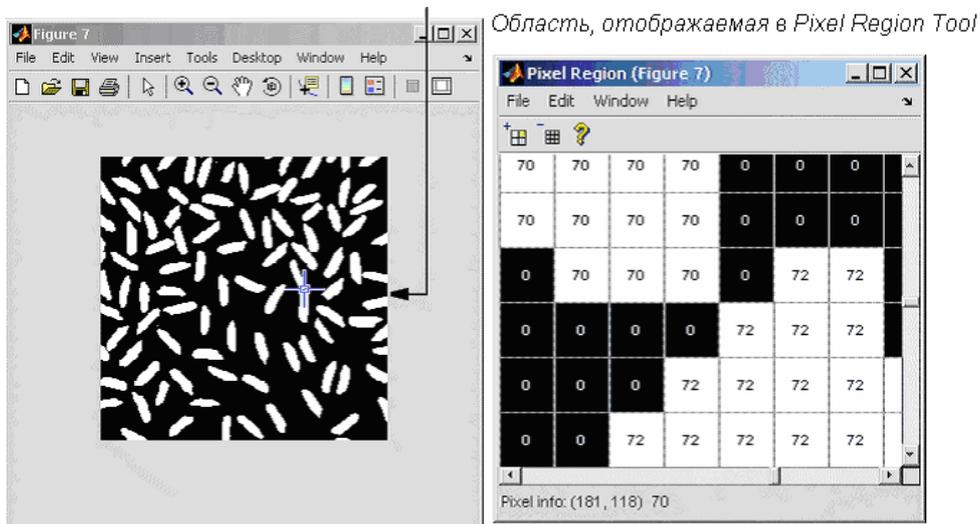


Рис. 21. Просмотр матрицы средствами «PixelRegionTool»

Шаг 9: Отображение матрицы меток в виде псевдоцветного индексного изображения

Один из возможных путей отображения матрицы меток состоит в использовании псевдоцветного индексного изображения. В псевдоцветном изображении числа, которые идентифицируют каждый объект в матрице меток, отображаются разным цветом, который связан с соответствующей палитрой цветов. Такое представление позволяет выделять различные объекты.

Для просмотра матрицы меток используется функция «label2rgb». Эта функция применяется при создании палитры, цвета фона и цвета каждого объекта изображения[5].

```
pseudo_color = label2rgb(labeled, @spring, 'c', 'shuffle');
```

```
imshow(pseudo_color);
```

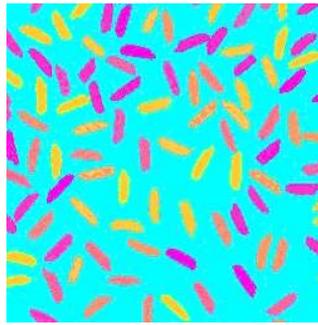


Рис. 22. Представление матрицы меток в виде псевдоцветного изображения

Шаг 10: Измерение свойств объектов на изображении

Функция «regionprops» применяется для измерения свойств объектов в некоторой окрестности изображения и представляет результат в виде массива. Если применить это к изображению с отмеченными компонентами, то создается структура для каждой компоненты[6].

В этом примере используется функция «regionprops» для создания структурированного массива, содержащего некоторые основные свойства. Если установить свойствам параметр «basic», то функция «regionprops» возвращает три измеренных параметра: площадь, центроид (или центр масс) и ограничивающий прямоугольник (локальная окрестность). В данном случае ограничивающий прямоугольник представляет собой локальную окрестность, отображающую гранулы риса.

```
graindata = regionprops(labeled,'basic')
```

Система MATLAB выдает такой результат

```
graindata =  
  
101x1 struct array with fields:  
Area  
Centroid  
BoundingBox
```

Для поиска области с 51-м отмеченным компонентом, необходимо найти поле «Areafield» и его 51 элемент в структуре массива «graindata».

```
graindata(51).Area
```

В этом случае результат будет таким

```
ans = 140
```

Для поиска наименее возможного прямоугольника и центроида (центра масс) некоторых компонентов используется следующий код:

```
graindata(51).BoundingBox, graindata(51).Centroid
ans =
107.5000    4.5000    13.0000    20.0000
ans =
114.5000    15.4500
```

Шаг 11: Вычисление статистических свойств объектов изображения

Функции системы MATLAB могут использоваться для вычисления статистических свойств объектов. Сначала используется функция «max» для поиска наибольшего зерна. (В нашем примере наибольшими являются два зерна риса, которые соприкасаются.)

```
max([graindata.Area])
```

Результат представляется в виде

```
ans =404
```

Далее с помощью функции «find» найдем те компоненты, которыми отмечены зерна риса с этой площадью.

```
biggrain = find([graindata.Area]==404)
```

Результат представляется в виде

```
biggrain =59
```

Найдем средний размер всех зерен риса.

```
mean([graindata.Area])
```

Результат будет представлен в виде

```
ans =175.0396
```

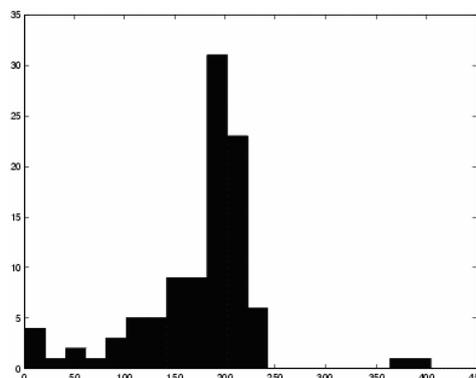


Рис. 23. Гистограмма распределения зерен риса по их размеру

Постоим на Рис. 23. гистограмму распределения зерен риса по их размеру (при этом будем использовать 20 позиций при градации по оси площади). Из гистограммы видно, что основная часть зерен риса имеет площадь от 150 до 250 пикселей [5].

```
hist([graindata.Area],20)
```

Задания к работе

Провести обработку изображений, аналогичную рассмотренной в теории, файл с изображением выбрать в соответствии с вариантом.

№ варианта	Изображение
1	Img1
2	Img2
3	Img3
4	Img4
5	Img5

Контрольные вопросы

1. С использованием какой функции определяется число объектов на изображении? Каковы особенности данной функции?
2. Для чего анализируется матрица меток?
3. Каким образом измеряются объекты на изображении?
4. С помощью какой функции изображение загружается в Matlab? В каком формате изображенное изображение интерпретируется Matlab'ом?

7. ЛАБОРАТОРНАЯ РАБОТА № 7 «АНАЛИЗ ДВУМЕРНЫХ ИЗОБРАЖЕНИЙ В МАТЛАВ. ВЫДЕЛЕНИЕ ГРАНИЦ»

Теория

7.1. Анализ изображений

Применяя технику анализа, можно говорить о структуре изображений. Рассмотрим некоторые функции, которые используются при анализе изображений, а также их текстуры.

Выделение границ

Для определения границ объектов изображения можно использовать функцию «edge». В качестве одного из критериев для определения края используется резкий перепад интенсивностей пикселей изображения. Отметим, что некоторые методы выделения границ, в зависимости от своих особенностей, в большей степени выделяют вертикальные, горизонтальные или все границы одинаково.

Одним из наиболее эффективных методов выделения границ является метод Канни. Метод Канни отличается от других известных методов тем, что при определении границ использует два порога (для слабых и сильных границ). Слабые границы отмечаются в результирующем изображении только тогда, когда они соединены с сильными. Для зашумленных изображений данный метод обеспечивает наилучшее обнаружение границ по сравнению с остальными методами, но требует существенно большего времени.

Рассмотрим пример применения метода выделения границ Канни, а также сравним его с другим известным методом выделения границ:

Считаем и отобразим исходное изображение.

```
I = imread('coins.png');  
imshow(I)
```



Рис. 24. Исходное изображение

Далее это изображение будет обработано операторами выделения границ Собела и Канни с последующей визуализацией.

```
BW1 = edge(I, 'sobel');  
BW2 = edge(I, 'canny');  
imshow(BW1)  
figure, imshow(BW2)
```

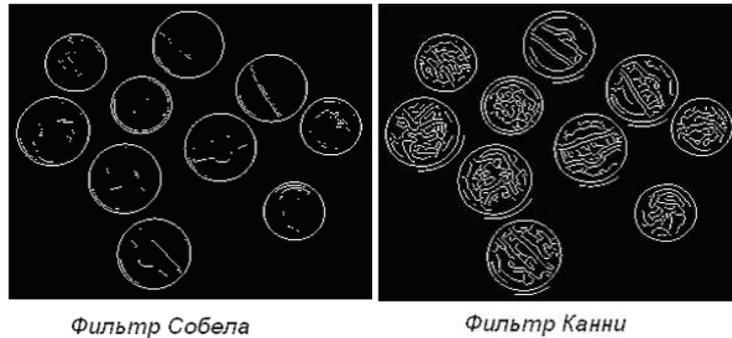


Рис. 25 Результат применения фильтров Собела и Канни к изображению

Для интерактивной демонстрации работы операторов выделения границ в приложении используется функция «*edgedemo*» [5].

Отслеживание границ

Приложение включает две функции, которые могут быть использованы для поиска границ объектов на бинарном изображении. Это функции «*bwtraceboundary*» и «*bwboundaries*».

Функция «*bwtraceboundary*» возвращает координаты строк и столбцов всех пикселей, которые принадлежат границе объектов изображения. Также необходимо описать начальную (стартовую) точку при отслеживании границ объектов изображения.

Функция «*bwboundaries*» также возвращает координаты строк и столбцов всех пикселей, которые принадлежат границе объектов изображений.

Для обеих функций ненулевые пиксели на бинарном изображении принадлежат объекту, а пиксели со значением 0 составляют фон [6].

Рассмотрим пример применения функции «*bwtraceboundary*» для отслеживания границ некоторого объекта бинарного изображения, а затем используем функцию «*bwboundaries*» для отслеживания границ всех объектов изображения.

Считаем и визуализируем изображение.

```
I = imread('coins.png');  
imshow(I)
```

Преобразуем это изображение в бинарное. Функции «bwtraceboundary» и «bwboundaries» работают только с бинарными изображениями.

```
BW = im2bw(I);  
imshow(BW)
```

Определим координаты строки и столбца пикселя на границе объекта. Функция «bwboundary» использует эту точку в качестве начальной при отслеживании границ.

```
dim = size(BW)  
col = round(dim(2)/2)-90;  
row = min(find(BW(:,col)))
```

Функция «bwtraceboundary» выполняет отслеживание границ от начальной точки. В качестве обязательных аргументов необходимо ввести само бинарное изображение, координаты начальной точки и направление первого шага. В примере использовано северное направление («N»).

```
boundary = bwtraceboundary(BW,[row, col], 'N');
```

Отобразим сначала исходное полутоновое изображение, а потом координаты, которые возвращаются функцией «bwtraceboundary».

```
imshow(I)  
hold on;  
plot(boundary(:,2),boundary(:,1), 'g', 'LineWidth', 3);
```

Для отслеживания границ всех монет на изображении используется функция «bwboundaries». По умолчанию функция «bwboundaries» определяет границы всех объектов на изображении, включая вложенные объекты. В нашем примере на бинарном изображении некоторые монеты содержат темные области, которые функция «bwboundaries» интерпретирует как отдельные объекты. Для того чтобы функция «bwboundaries» отслеживала только границы монет, используется функция «imfill» для заполнения областей внутри каждой монеты.

```
BW_filled = imfill(BW, 'holes');  
boundaries = bwboundaries(BW_filled);
```

Функция «bwboundaries» возвращает массив значений, где каждая ячейка содержит координаты пикселей объекта изображения. Отобразим границы всех монет на исходном полутоновом изображении, используя координаты, возвращаемые функцией «bwboundaries».[5]

```
for k=1:10  
    b = boundaries{k};  
    plot(b(:,2),b(:,1), 'g', 'LineWidth', 3);
```

end

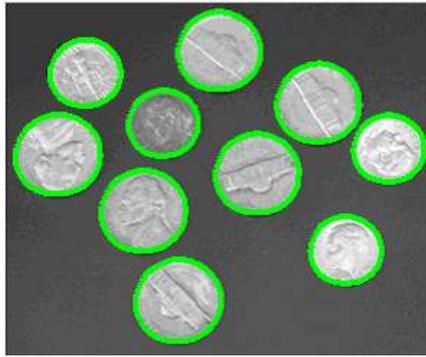


Рис. 26. Выделение границ на изображении

Задания к работе

Выделить границы монет на изображениях аналогично рассмотренному в теории, файл с изображением выбрать в соответствии с вариантом.

№ варианта	Изображение
1	Mon1
2	Mon 2
3	Mon 3
4	Mon 4
5	Mon 5

Контрольные вопросы

1. Каковы особенности метода Канни?
2. Какие функции Matlab используются при поиске границ на изображении?
3. Для чего используется функция «bwboundaries»?
4. Каковы особенности функции «edge»?

Список литературы

1. Вохник О.М., Зотов А.М., Короленко П.В., Рыжикова Ю.В. Моделирование и обработка стохастических сигналов и структур [Режим доступа]: <http://optics.sinp.msu.ru/stud/stud.html>
2. Cross-correlation – MATLAB xcorr [Режим доступа] <http://www.mathworks.com/help/signal/ref/xcorr.html>
3. Борискевич А.А. Предварительная обработка мультимедийной информации в интегрированной среде Matlab/C: метод. пособие по курсу «Цифровая обработка и защита мультимедийной информации» для студ. спец. «Системы распределения мультимедийной информации» всех форм обуч. / А.А. Борискевич, А.Ю. Лагойко, В.Ю. Цветков. – Минск: БГУИР, 2011. – 53 с.
4. В. Колдае. Численные методы и программирование: учеб. Пособие. - М.: Форум, 2016. – 336 с. ISBN 978-5-8199-0333-9
5. Воскобойников Ю. Е. Вейвлет-фильтрации сигналов и изображений (с примерами в пакете MathCAD) / Ю. Е. Воскобойников ; Новосиб. гос. архитектур.-строит. ун-т (Сибстрин). – Новосибирск : НГАСУ (Сибстрин), 2015. –188 с. ISBN 978-5-7795-0755-4
6. Дьяконов В. П. MATLAB. Полный самоучитель. – М.: ДМК Пресс, 2012. – 768 с.: ил. ISBN 978-5-94074-652-2
7. Журавель И.М. Описание ImageProcessingToolbox [Режим доступа]: <http://matlab.exponenta.ru/imageprocess/book5/index.php>
8. Сергиенко А.Б. Signal Processing Toolbox – обзор. [Режим доступа]: <http://matlab.exponenta.ru/signalprocess/book2/index.php>
9. Смоленцев Н. К. Основы теории вейвлетов. Вейвлеты в MATLAB.- М.: ДМК-Пресс, 2014 .- 628 с. ISBN 978-5-94074-955-4

10. Ю. А. Брюханов Основы цифровой обработки сигналов : учеб. пособие / Ю. А. Брюханов, А. Л. Приоров, В. И. Джиган, В. В. Хрящев, Яросл. гос. ун-т им. П. Г. Демидова .— Ярославль : ЯрГУ, 2013 .— Ил. 134. Табл. 4. Библиогр.: 39 назв. — ISBN 978-5-8397-0933-1