

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Емельянов Сергей Геннадьевич
Должность: ректор
Дата подписания: 16.12.2021 20:54:34
Уникальный программный ключ:
9ba7d3e34c012eba476ffd2d064c2781953be730df2374d16f3c0ce538f626

МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра биомедицинской инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Докмонова
« _____ » _____ 2018 г.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ АНАЛИЗА И
КЛАССИФИКАЦИИ СИСТЕМЫ АНАЛИЗА И
КЛАССИФИКАЦИИ КВАЗИПЕРИОДИЧЕСКИХ СИГНАЛОВ

Методические рекомендации по организации и выполнению
самостоятельной работы для аспирантов направления подготовки
09.06.01 и 12.06.01

Курск 2018

УДК 004.93:61

Составители: С.А. Филист., К.Д.А. Кассим

Рецензент

Доктор технических наук, профессор А.Ф. Рыбочкин

Интеллектуальные системы анализа и классификации системы анализа и классификации квазипериодических сигналов: методические рекомендации по организации и выполнению самостоятельной работы

Юго-Зап. гос. ун-т; сост.: С.А. Филист, К.Д.А. Кассим. Курск, 2018. 65 с.

Методические указания по структуре, содержанию и стилю изложения материала соответствуют требованиям, предъявляемым к учебным и методическим пособиям.

Предназначены для аспирантов направлений подготовки 09.06.01 «Информатика и вычислительная техника (Системный анализ, управление и обработка информации (технические и медицинские системы)) и 12.06.01 «Фотоника, приборостроение, оптические и биотехнические системы и технологии (Приборы, системы и изделия медицинского назначения)»

Текст печатается в авторской редакции

Подписано в печать 1.03.18. Формат 60x84 1/16.

Усл.печ.л. 3,3. Уч.-изд.л. 3,1 Тираж 100 экз. Заказ: 1449 Бесплатно.
Юго-Западный государственный университет.

305040, г.Курск, ул. 50 лет Октября, 94.

Самостоятельная работа №1

Статистические характеристики квазипериодических сигналов

Цель работы: Приобретение навыков обработки и обобщения медико-биологических данных, а также навыков работы со статистическими и логическими функциями Excel.

1.1 Теоретические сведения

Отличительной чертой современного этапа развития естествознания является математизация и использование статистических методов анализа. *Статистика* — отрасль знаний (и соответствующие ей учебные дисциплины), в которой излагаются общие вопросы сбора, измерения и анализа массовых статистических (количественных или качественных) данных. практически нет такого метода статистического анализа, который не нашёл бы применения в медицине.

Однако в процессе использования этих методов возникают сложности, обусловленные рядом объективных причин:

- Сильная изменчивость исследуемых признаков ввиду влияния очень большого количества неуправляемых или неконтролируемых факторов;
- Проблемы в формировании выборок (планов экспериментов) требуемого объёма и структуры;
- Влияние психологических установок и воздействий на результаты испытаний;
- Измерение многих важных показателей в неколичественных шкалах (обычно классификации и порядка);
- Трудности в освоении методов статистического анализа медицинскими работниками.

Несмотря на все эти сложности, статистические методы заняли прочные позиции в арсенале современной медицины и фармакологии. В клиниках, например, они служат важным вспомогательным средством для получения информации о различных антропометрических, социальных и экологических факторов на распространённость и течение заболеваний. Также они дают возможность сравнивать эффективность различных методов лечения и т.д. Кроме того, бывают случаи, когда только путём анализа статистически данных можно определить являются ли

некоторые побочные эффекты (случаи летального исхода) следствием применения конкретного препарата.

В статистике при анализе данных приходится сталкиваться со следующими проблемами:

- **Сколько** данных необходимо выбрать и **как** их отбирать;
- **Правомочность** распространения выводов, сделанных на основании выборочных данных на всю генеральную совокупность;
- **Выбор оптимальных** способов оценивания;
- Выбор способов **обобщения, классификации и представления** данных.

Оценки параметров должны отвечать ниже перечисленным требованиям:

1. **Несмещённость.** Это означает, что при проведении очень большого количества испытаний с выборками одинакового размера среднее значение каждой выборки стремится к истинному значению генеральной совокупности. Смещённость обычно обусловлена наличием систематической ошибки.

2. **Состоятельность.** С ростом размера выборки оценка должна стремиться к значению соответствующего параметра генеральной совокупности с вероятностью, стремящейся к 1.

3. **Эффективность.** Выбранная оценка для выборки равного объёма должна иметь минимальную дисперсию.

4. **Достаточность.** оценка должна содержать необходимую информацию и не требовать дополнительной.

1.2 Основные характеристики одномерного распределения

1. **Среднее арифметическое (выборочное).** Характеризует положение центра в распределении. Рассчитывается по формуле:

$$\bar{X} = \frac{\sum_{i=1}^N x_i}{N}.$$

2. **Мода.** Это значение, которое наблюдается наибольшее число раз (наиболее вероятное), рассчитывается по формуле:

$$M_o = X_{M_o} + h \frac{m_{M_o} - m_{M_o-1}}{2m_{M_o} - m_{M_o+1} - m_{M_o-1}}$$

где X_{Mo} – начало модального интервала (такого, которому соответствует наибольшая частота), h – величина модального интервала, m_{Mo} – частота модального интервала, m_{Mo-1} – частота интервала, предшествующего модальному, m_{Mo+1} – частота интервала, следующего за модальным.

3. **Медиана.** Это значение, которое делит ранжированный ряд на две равные по объёму группы. Вариационный ряд ранжируется. Если количество членов ряда нечётное, медианой является значение ряда, которое расположено посередине, то есть элемент с номером $(N+1)/2$. Если число членов ряда чётное, то медиана равна среднему членов ряда с номерами $N/2$ и $N/2+1$. медианным называется интервал, в котором находится значение медианы.

4. **Эмпирическая дисперсия (дисперсия)** – мера разброса данной случайной величины, т. е. её отклонения от математического ожидания, определяемая по формуле:

$$S^2 = \frac{\sum_{i=1}^N (X_i - \bar{X})^2}{N - 1}.$$

5. **Стандартное отклонение (среднеквадратическое отклонение (СКО))** – корень квадратный из дисперсии. Обозначается как S или σ .

6. **Среднее линейное отклонение** – величина, вычисляемая по формуле:

$$\bar{d} = \frac{\sum_{i=1}^N |X_i - \bar{X}|}{N}$$

1.3 Современная технология анализа данных

В основе обработки и анализа данных лежат известные математические методы.

Благодаря использованию информационных технологий, в наше время этап обработки данных стал менее всего трудоемким. На первое место относительно трудоемкости вышли такие этапы, как освоение статистических пакетов, этап подготовки данных к анализу, этап предыдущего анализа данных и этап

интерпретации результатов. Все в целом привело к изменениям в технологии обработки и анализа данных. При этом для выполнения методов обработки медико-биологических данных от пользователя нужно лишь применение статистических методов обработки данных и использования соответствующих пакетов прикладных программ. Врачу, как правило, не нужно углубляться в сложные математические теории, а нужно понимать, для чего и каким образом они используются.

На практике для врача обработка и анализ данных сводятся к **решению следующих задач:**

- Получение представления об основных статистических методах;
- Усвоение пакета прикладных программ;
- Анализа и интерпретация результатов исследований.

Сам анализ данных с использованием статистического пакета (работа с пакетом, сама технология анализа данных) включает в себя такие этапы:

- Планирование исследования;
- Подготовка данных к анализу;
- Предыдущий анализ данных;
- Выбор метода анализа и его реализация;
- Интерпретация результатов;
- Представление результатов.

1.4 Оценка параметров распределения и проверка гипотез

Статистические гипотезы – это предположения, которые относятся к виду распределения случайной величины или отдельных его параметров. Необходимость использования статистических гипотез возникает тогда, когда обстоятельства вынуждают нас делать выбор между двумя способами действия. Для оценивания параметров по эмпирическим законам формулируется нулевая гипотеза (H_0) об “отсутствии разногласий”. Нулевая гипотеза является примером статистического вывода: если нулевую гипотезу отбросить, то вывод заключается в том, что в совокупности, которая рассматривается есть разногласия, то есть принимается альтернативная гипотеза H_1 .

Вероятность с которой может быть отклонена нулевая гипотеза, когда она является верной, называется **уровнем**

значимости (для медико-биологических исследований достаточным является уровень значимости $\alpha=0,05$. Уровень значимости задается предварительно.

Вероятность принятия правильности решения (гипотеза H_0 является верной) называется доверительной вероятностью (для медико-биологических исследований $p = 0,95$). Проверка гипотез как правило сводится к проверке статистических характеристик, которые оценивают параметры законов распределения.

Для проверки гипотез используют **статистический критерий K** – это решающее правило, которое обеспечивает принятие верной гипотезы и отклонения ложной с большой вероятностью. Совокупность значений, при которых основная гипотеза не принимается называется **критической областью**. Точки, что отделяют критическую область от области принятия решений называются **критическими**.

Для определения критической области задается уровень значимости. Для каждого из критериев есть таблицы, по которыми находят значение критических точек. В силу того, что гипотезы не могут быть доведены, а могут быть проверены при принятии гипотезы возможные ошибки.

Например, процесс производства лекарств является сложным. Какое-нибудь отклонение (даже незначительное) от технологии вызывает появление высокотоксичной побочной примеси. Токсичность этой примеси может быть настолько большой, что даже такое ее количество которая не может быть определено при химическом анализе является опасной для пациента. Поэтому перед тем как выпускать в продажу партию счетов ее исследуют на токсичность биологическими методами: небольшие дозы препарата вводятся определенному количеству животных и результаты регистрируются. Количество животных, что погибли является случайной величиной. Как правило инъецируется несколько групп животных. Исследование препарата может привести к одному из двух возможных действий:

- Выпустить партию лекарств в продаже;
- Вернут партию поставщику для переработки или уничтожения.

Выбор между двумя действиями может привести к осуществлению ошибок **двух видов**:

- Признать препарат безопасным для пациентов, когда в действительности препарат опасен. Эта ошибка может стоить жизни пациента;

- Признать препарат опасным для пациентов, когда в действительности он является безопасным. Последствия этой ошибки могут быть выражены и в дополнительных финансовых затратах.

Таким образом, последствия ошибок являются разными за своими значениями, поэтому при испытании гипотез необходимо избегать одной из возможных ошибок, которая есть более важная чем другая. Следовательно, при проверке гипотез возможные ошибки двух видов:

1. ***H₀ отбрасывается***, когда она правильная – ошибка I-го рода.

2. ***H₀ принимается***, когда правильная H₁ – ошибка II-го рода.

Понижая уровень значимости мы уменьшаем вероятность ошибки первого рода, но при этом увеличивается вероятность ошибки второго рода. Отметим, что чем большая прочность критерия, тем меньшая вероятность ошибки второго рода.

1.5 Этапы проверки гипотез

1. Определение статистической модели, что будет использоваться. На этом этапе выдвигают некоторый набор предпосылок относительно закона распределения случайной величины и ее параметров. Например, закон распределения нормален, величины независимые и т.д.

2. Формулировка H₀ и H₁. Выбирают критерий, который подходит к предложенной статистической модели.

3. Выбирают уровень значимости α в зависимости от надежности выводов, которые требуются.

4. Определение критической области для проверки H₀. Если значение критерия попадает в эту область, то H₀ отбрасывается. При условии, что H₀ правильная, вероятность попадания в критическую область равняется α . Вид этой области (односторонняя или двусторонняя) зависит от принятой H₀.

5. Расчет значение выбранного статистического критерия для существующих данных.

6. Сравнение рассчитанного значения критерия с критическим, а затем решают принять или отбросить H₀.

1.6 Критерии проверки гипотез

В медико-биологических исследованиях часто возникает задача оценивания параметров распределения за малыми выборками. Для оценивания параметров распределения таких выборок используют *распределение Стьюдента*. Для случайной величины t , распределенной по закону Стьюдента с n степенями свободы табулировано. Поэтому сравнивают значение рассчитанного коэффициента t_α с табличным.

Сформулируем *критерий Стьюдента*: Проверка равенства средних значений в двух выборках.

$$\begin{cases} \text{Если } t_\alpha < t, \text{ то принимается нулевая гипотеза } H_0 (\bar{X} = \bar{Y}); \\ \text{Если } t_\alpha > t, \text{ то принимается альтернативная гипотеза } H_1 (\bar{X} \neq \bar{Y}); \end{cases}$$

Таблица 1.1 – Граничные значения t в распределении Стьюдента.

	Уровень значимости, α							
	0.02	0.01	0.05	0.002	0.001	0.0005	0.0002	0.00001
1	3.0770	6.3130	12.7060	31.820	63.656	127.656	318.306	636.619
2	1.8850	2.9200	4.3020	6.964	9.924	14.089	22.327	31.599
3	1.6377	2.35340	3.182	4.540	5.840	7.458	10.214	12.924
4	1.5332	2.13180	2.776	3.746	4.604	5.597	7.173	8.610
5	1.4759	2.01500	2.570	3.649	4.0321	4.773	5.893	6.863
6	1.4390	1.943	2.4460	3.1420	3.7070	4.316	5.2070	5.958
7	1.4149	1.8946	2.3646	2.998	3.4995	4.2293	4.785	5.4079
8	1.3968	1.8596	2.3060	2.8965	3.3554	3.832	4.5008	5.0413
9	1.3830	1.8331	2.2622	2.8214	3.2498	3.6897	4.2968	4.780
10	1.3720	1.8125	2.2281	2.7638	3.1693	3.5814	4.1437	4.5869
11	1.363	1.795	2.201	2.718	3.105	3.496	4.024	4.437
12	1.3562	1.7823	2.1788	2.6810	3.0845	3.4284	3.929	4.178
13	1.3502	1.7709	2.1604	2.6503	3.1123	3.3725	3.852	4.220
14	1.3450	1.7613	2.1448	2.6245	2.976	3.3257	3.787	4.140
15	1.3406	1.7530	2.1314	2.6025	2.9467	3.2860	3.732	4.072
16	1.3360	1.7450	2.1190	2.5830	2.9200	3.2520	3.6860	4.0150
17	1.3334	1.7396	2.1098	2.5668	2.8982	3.2224	3.6458	3.965
18	1.3304	1.7341	2.1009	2.5514	2.8784	3.1966	3.6105	3.9216
19	1.3277	1.7291	2.0930	2.5395	2.8609	3.1737	3.5794	3.8834
20	1.3253	1.7247	2.08600	2.5280	2.8453	3.1534	3.5518	3.8495

Примечание к таблице 1.1: v - число степеней свободы.
Число степеней свободы рассчитывается по формуле:

$$v = (n_x + n_y - 2) \left(0,5 + \frac{S_x^2 \cdot S_y^2}{S_x^2 + S_y^2} \right)$$

где n_x и n_y – количество измерений в выборках, S_x и S_y – СКО выборок.

$$S_x^2 = \frac{\sum_{i=1}^{n_x} (X_i - \bar{X})^2}{n_x - 1}$$

$$S_y^2 = \frac{\sum_{i=1}^{n_y} (Y_i - \bar{Y})^2}{n_y - 1}$$

При этом t_α - рассчитывается по формуле:

$$t_\alpha = \frac{\bar{y} - \bar{x}}{\sqrt{\frac{S_x^2}{n_x} + \frac{S_y^2}{n_y}}}$$

Далее по таблице 1.1 находят значение t для рассчитанного количества степеней свободы и делается вывод о том принять нулевую гипотезу или отклонить её.

Сформулируем **критерий Фишера**: Проверка гипотезы о принадлежности двух дисперсий одной генеральной совокупности и следовательно о их равенстве. Гипотеза о равенстве двух дисперсий двух нормальных генеральных совокупностей принимается, если отношение большей дисперсии к меньшей меньше критического значения распределения Фишера.

$$\left\{ \begin{array}{l} \text{Если } \frac{S_x^2}{S_y^2} < F_{\alpha, v_x, v_y}, \text{ то принимается нулевая гипотеза } H_0 (S_x^2 = S_y^2); \\ \text{Если } \frac{S_x^2}{S_y^2} > F_{\alpha, v_x, v_y}, \text{ то принимается альтернативная гипотеза } H_1 (S_x^2 \neq S_y^2); \end{array} \right.$$

При этом число степеней свободы в числителе и знаменателе рассчитывается по формулам:

$$v_x = n_x - 1$$

$$v_y = n_y - 1$$

Уровень значимости как правило используется равным $\alpha=0,05$.

Таблица 1.2 – Значения критерия Фишера для уровня значимости $\alpha = 0,05$.

y	v_x											
	v_y	1										
	161.45	1	161.45	1	161.45	1	161.45	1	161.45	1	161.45	1
	18.51	2	18.51	2	18.51	2	18.51	2	18.51	2	18.51	2
	10.13	3	10.13	3	10.13	3	10.13	3	10.13	3	10.13	3
	7.71	4	7.71	4	7.71	4	7.71	4	7.71	4	7.71	4
	6.61	5	6.61	5	6.61	5	6.61	5	6.61	5	6.61	5
	5.99	6	5.99	6	5.99	6	5.99	6	5.99	6	5.99	6
	5.59	7	5.59	7	5.59	7	5.59	7	5.59	7	5.59	7
	5.32	8	5.32	8	5.32	8	5.32	8	5.32	8	5.32	8
	5.12	9	5.12	9	5.12	9	5.12	9	5.12	9	5.12	9
0	4.96	10	4.96	10	4.96	10	4.96	10	4.96	10	4.96	10
1	4.84	11	4.84	11	4.84	11	4.84	11	4.84	11	4.84	11
2	4.75	12	4.75	12	4.75	12	4.75	12	4.75	12	4.75	12
3	4.67	13	4.67	13	4.67	13	4.67	13	4.67	13	4.67	13
4	4.60	3.74	3.34	3.11	2.96	2.85	2.76	2.70	2.65	2.60	2.46	2.46
5	4.54	3.68	3.29	3.06	2.90	2.79	2.71	2.64	2.59	2.54	2.40	2.40
6	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.49	2.35	2.35
7	4.45	3.59	3.20	2.96	2.81	2.70	2.61	2.55	2.49	2.45	2.31	2.31
8	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	2.41	2.27	2.27
9	4.38	3.52	3.13	2.90	2.74	2.63	2.54	2.48	2.42	2.38	2.23	2.23
0	4.35	3.49	3.10	2.87	2.71	2.60	2.51	2.45	2.39	2.35	2.20	2.20

Примечание к таблице 1.2: ν_x - число степеней свободы большей дисперсии, ν_y - число степеней свободы меньшей дисперсии.

1.7 Основы формальной логики

Логика - наука, изучающая законы и формы мышления; учение о способах рассуждений и доказательств.

Законы мира, сущность предметов, общее в них мы познаем посредством абстрактного мышления. Основными формами абстрактного мышления являются понятия, суждения и умозаключения.

Понятие - форма мышления, в которой отражаются существенные признаки отдельного предмета или класса однородных предметов. Понятия в языке выражаются словами. Содержание понятия – совокупность существенных признаков, отраженных в этом понятии.

Объем понятия – множество предметов, каждому из которых принадлежат признаки, составляющие содержание понятия. Выделяют понятия общие и единичные.

Выделяют следующие отношения понятий по объему:

- тождество или совпадение объемов, означающее, что объем одного понятия равен объему другого понятия;
- подчинение или включение объемов: объем одного из понятий полностью включен в объем другого;
- исключение объемов - случай, в котором нет ни одного признака, который бы находился в двух объемах;
- пересечение или частичное совпадение объемов;
- соподчинение объемов - случай, когда объемы двух понятий, исключаящие друг друга, входят в объем третьего.

Суждение - это форма мышления, в которой что-либо утверждается или отрицается о предметах, признаках или их отношениях.

Умозаключение - форма мышления, посредством которой из одного или нескольких суждений, называемых посылками, мы по определенным правилам вывода получаем суждение-заключение.

Алгебра в широком смысле этого слова наука об общих операциях, аналогичных сложению и умножению, которые могут

выполняться не только над числами, но и над другими математическими объектами.

Примеры алгебр:

- алгебра натуральных чисел;
- алгебра рациональных чисел,;
- алгебра многочленов;
- алгебра векторов;
- алгебра матриц;
- алгебра множеств и т.д.

Объектами алгебры логики или булевой алгебры являются высказывания.

Высказывание - это любое предложение какого-либо языка (утверждение), содержание которого можно определить как истинное или ложное. Всякое высказывание *или истинно, или ложно*; быть одновременно и тем и другим оно не может. В естественном языке высказывания выражаются повествовательными предложениями. Восклицательные и вопросительные предложения высказываниями не являются. Высказывания могут выражаться с помощью математических, физических, химических и прочих знаков. Из двух числовых выражений можно составить *высказывания*, соединив их знаками равенства или неравенства. Высказывание называется *простым (элементарным)*, если никакая его часть сама не является высказыванием.

1.8 Основные операции алгебры высказываний

1. Логическая операция конъюнкция (лат. conjunctio - связываю).

- В естественном языке соответствует союзу «и»;
- Обозначение: &;
- В языках программирования обозначение: and;
- Иное название: логическое умножение.

Конъюнкция - это логическая операция, ставящая в соответствие каждому двум простым высказываниям составное высказывание, являющееся истинным тогда и только тогда, когда оба исходных высказывания истинны.

Таблица 1.3 – Таблица истинности конъюнкции

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

2. Логическая операция дизъюнкция (лат. disjunctio - различаю).

- В естественном языке соответствует союзу «или»;
- Обозначение: \cup ;
- В языках программирования обозначение: or;
- Иное название: логическое сложение.

Дизъюнкция - это логическая операция, которая каждому двум простым высказываниям ставит в соответствие составное высказывание, являющееся ложным тогда и только тогда, когда оба исходных высказывания ложны и истинным, когда хотя бы одно из двух образующих его высказываний истинно.

Таблица 1.4 – Таблица истинности дизъюнкции

A	B	$A \cup B$
0	0	0
0	1	1
1	0	1
1	1	1

3. Логическая операция инверсия (лат. inversio - переворачиваю).

- в естественных языках соответствует частице «не»;
- Обозначение \bar{A} ;
- В языках программирования обозначение: not;
- Иное название: отрицание.

Отрицание - это логическая операция, которая каждому простому высказыванию ставит в соответствие составное высказывание, заключающееся в том, что исходное высказывание отрицается.

Таблица 1.5 – Таблица истинности отрицания

A	\bar{A}
0	0
1	1

4. Логическая операция импликация (лат. *implicatio* - тесно связываю).

- В естественном языке соответствует обороту Если ..., то ...;
- Обозначение: \rightarrow ;
- Иное название: логическое следование.

Импликация - это логическая операция, ставящая в соответствие каждому двум простым высказываниям составное высказывание, являющееся ложным тогда и только тогда, когда условие (первое высказывание) истинно, а следствие (второе высказывание) ложно.

Таблица 1.6 – Таблица истинности импликации

A	B	$A \rightarrow B$
0	0	0
0	1	1
1	0	1
1	1	1

5. Логическая операция эквиваленция (лат. *aequivalens* - равноценное).

- В естественном языке соответствует оборотам речи тогда и только тогда и в том и только в том случае;
- Обозначение: \sim ;
- иное название: равнозначность.

Эквиваленция - это логическая операция, ставящая в соответствие каждому двум простым высказываниям составное высказывание, являющееся истинным тогда и только тогда, когда оба исходных высказывания одновременно истинны или одновременно ложны.

Таблица 1.7 – Таблица истинности эквиваленции

A	B	$A \sim B$
0	0	1
0	1	0
1	0	0
1	1	1

Логические операции имеют следующий приоритет:

1. Действия в скобках;
2. Инверсия;
3. $\&$;
4. \cup ;
5. \rightarrow ;
6. \sim .

Таблицу, показывающую, какие значения принимает составное высказывание при всех сочетаниях (наборах) значений входящих в него простых высказываний, называют *таблицей истинности* составного высказывания. Составные высказывания в алгебре логики записываются с помощью логических выражений. Для любого логического выражения достаточно просто построить таблицу истинности.

Алгоритм построения таблицы истинности:

1. Подсчитать количество переменных n в логическом выражении;
2. Определить число строк в таблице $m = 2^n$;
3. Подсчитать количество логических операций в формуле;
4. Установить последовательность выполнения логических операций с учетом скобок и приоритетов;
5. Определить количество столбцов в таблице: число переменных плюс число операций;
6. Выписать наборы входных переменных с учетом того, что они представляют собой натуральный ряд n -разрядных двоичных чисел от 0 до $2^n - 1$;
7. Провести заполнение таблицы истинности по столбикам, выполняя логические операции в соответствии с установленной в п.4 последовательностью.

Наборы входных переменных, во избежание ошибок, рекомендуют перечислять следующим образом:

- а) Определить количество наборов входных переменных;

- б) Разделить колонку значений первой переменной пополам и заполнить верхнюю часть колонки 0, а нижнюю -1;
- с) Разделить колонку значений второй переменной на четыре части и заполнить каждую четверть чередующимися группами 0 или 1, начиная с группы 0;
- д) Продолжать деление колонок значений последующих переменных на 8, 16 и т.д. частей и заполнение их группами 0 или 1 до тех пор, пока группы 0 и 1 не будут состоять из одного символа.

1.9 Функции. Синтаксис функций Excel

Функции Excel - это специальные, заранее созданные формулы, которые позволяют легко и быстро выполнять сложные вычисления. Их можно сравнить со специальными клавишами на калькуляторах, предназначенных для вычисления квадратных корней, логарифмов и проч. Excel имеет несколько сотен встроенных функций, которые выполняют широкий спектр различных вычислений. Некоторые функции являются эквивалентами длинных математических формул, которые можно сделать самому. А некоторые функции в виде формул реализовать невозможно. Функции состоят из двух частей: *имени функции* и *одного или нескольких аргументов*. *Имя функции*, например СУММ, - описывает операцию, которую эта функция выполняет. *Аргументы* задают значения или ячейки, используемые функцией. В формуле, приведенной ниже: СУММ - имя функции; B1:B5 - аргумент. Данная формула суммирует числа в ячейках B1, B2, B3, B4, B5.

=СУММ(B1:B5)

Знак равенства в начале формулы означает, что введена именно формула, а не текст. Если знак равенства будет отсутствовать, то Excel воспримет ввод просто как текст. *Аргумент функции* заключен в *круглые скобки*. Открывающая скобка отмечает начало аргумента и ставится сразу после имени функции. В случае ввода пробела или другого символа между именем и открывающей скобкой в ячейке будет отображено ошибочное значение #ИМЯ? Некоторые функции не имеют аргументов. Даже в этом случае функция должна содержать круглые скобки:

=С5*ПИ()

При использовании в функции нескольких аргументов они отделяются один от другого *точкой с запятой*. Например, следующая формула указывает, что необходимо перемножить числа в ячейках А1, А3, А6:

=ПРОИЗВЕД(А1;А3;А6)

В функции можно использовать до 30 аргументов, если при этом общая длина формулы не превосходит 1024 символов. Однако любой аргумент может быть *диапазоном*, содержащим произвольное число ячеек листа. Например:

=СУММ(А2:А5;В4:В8)

Указанные в ссылке ячейки в свою очередь могут содержать формулы, которые ссылаются на другие ячейки или диапазоны. Используя аргументы, можно легко создавать длинные цепочки формул для выполнения сложных операций.

В приведенных ранее примерах все аргументы были *ссылками на ячейки или диапазоны*. Но в качестве аргументов можно также использовать *числовые, текстовые и логические значения, имена диапазонов, массивы и ошибочные значения*. Некоторые функции возвращают значения этих типов, и их в дальнейшем можно использовать в качестве аргументов в других функциях. Аргументы функции могут быть числовыми. Например, функция СУММ в следующей формуле суммирует числа 24, 987, 49:

=СУММ(24;987;49)

В качестве аргумента функции могут использоваться текстовые значения. Например:

=ТЕКСТ(ТДАТА();"Д МММ ГГГГ")

В этой формуле второй аргумент функции ТЕКСТ является текстовым и задает шаблон для преобразования десятичного значения даты, возвращаемого функцией ТДАТА(NOW), в строку символов. Текстовый аргумент может быть строкой символов,

заклученной в двойные кавычки, или ссылкой на ячейку, которая содержит текст.

Аргументы ряда функций могут принимать только логические значения **ИСТИНА** или **ЛОЖЬ**. Логическое выражение возвращает значение **ИСТИНА** или **ЛОЖЬ** в ячейку или формулу, содержащую это выражение. Например:

=ЕСЛИ(A1=ИСТИНА;"Повышение";"Понижение")&" цены"

В качестве аргумента функции можно указать имя диапазона. Например, если диапазону ячеек A1:A5 присвоено имя "Дебет" (Вставка – Имя - Присвоить), то для вычисления суммы чисел в ячейках с A1 по A5 можно использовать формулу:

=СУММ(Дебет)

В одной функции можно использовать аргументы различных типов:

=СРЗНАЧ(Дебет;C5;2*8)

1.10 Логические функции Excel

Функция ЕСЛИ Функции И, ИЛИ, НЕ. Вложенные функции ЕСЛИ Функции ИСТИНА и ЛОЖЬ Функция ЕПУСТО.

Логические выражения используются для записи условий, в которых сравниваются числа, функции, формулы, текстовые или логические значения. Любое логическое выражение должно содержать по крайней мере один *оператор сравнения*, который определяет отношение между элементами логического выражения. Ниже представлен список операторов сравнения Excel:

- = Равно;
- > Больше;
- < Меньше;
- >= Больше или равно;
- <= Меньше или равно;
- <> Не равно;

Результатом логического выражения является логическое значение *ИСТИНА* (1) или логическое значение *ЛОЖЬ* (0).

Функция ЕСЛИ (IF) имеет следующий синтаксис:

=ЕСЛИ(логическое_выражение;значение_если_истина;значение_если_ложь)

Следующая формула возвращает значение 10, если значение в ячейке А1 больше 3, а в противном случае - 20:

=ЕСЛИ(А1>3;10;20)

В качестве аргументов функции ЕСЛИ можно использовать другие функции. В функции ЕСЛИ можно использовать текстовые аргументы. Например:

=ЕСЛИ(А1>=4;"Зачет сдал";"Зачет не сдал")

Можно использовать **текстовые аргументы** в функции ЕСЛИ, чтобы при невыполнении условия она возвращала пустую строку вместо 0. Например:

=ЕСЛИ(СУММ(А1:А3)=30;А10;"")

Аргумент логическое_выражение функции ЕСЛИ может содержать текстовое значение. Например:

=ЕСЛИ(А1="Динамо";10;290)

Эта формула возвращает значение 10, если ячейка А1 содержит строку "Динамо", и 290, если в ней находится любое другое значение. Совпадение между сравниваемыми текстовыми значениями должно быть точным, но без учета регистра.

Функции **И (AND)**, **ИЛИ (OR)**, **НЕ (NOT)** - позволяют создавать сложные логические выражения. Эти функции работают в сочетании с простыми операторами сравнения. Функции И и ИЛИ могут иметь до 30 логических аргументов и имеют синтаксис:

=И(логическое_значение1;логическое_значение2...)
=ИЛИ(логическое_значение1;логическое_значение2...)

Функция НЕ имеет только один аргумент и следующий синтаксис:

=НЕ(логическое_значение)

Аргументы функций **И**, **ИЛИ**, **НЕ** могут быть логическими выражениями, массивами или ссылками на ячейки, содержащие логические значения.

Пусть Excel возвращает текст "Прошел", если ученик имеет средний балл более 4 (ячейка A2), и пропуск занятий меньше 3 (ячейка A3). Формула примет вид:

=ЕСЛИ(И(A2>4;A3<3);"Прошел";"Не прошел")

Не смотря на то, что функция **ИЛИ** имеет те же аргументы, что и **И**, результаты получаются совершенно различными. Так, если в предыдущей формуле заменить функцию И на ИЛИ, то ученик будет проходить, если выполняется хотя бы одно из условий (средний балл более 4 или пропуски занятий менее 3). Таким образом, функция ИЛИ возвращает логическое значение ИСТИНА, если хотя бы одно из логических выражений истинно, а функция И возвращает логическое значение ИСТИНА, только если все логические выражения истинны. Функция НЕ меняет значение своего аргумента на противоположное логическое значение и обычно используется в сочетании с другими функциями. Эта функция возвращает логическое значение ИСТИНА, если аргумент имеет значение ЛОЖЬ, и логическое значение ЛОЖЬ, если аргумент имеет значение ИСТИНА.

Иногда бывает очень трудно решить логическую задачу только с помощью операторов сравнения и функций И, ИЛИ, НЕ. В этих случаях можно использовать вложенные функции ЕСЛИ. Например, в следующей формуле используются три функции ЕСЛИ:

=ЕСЛИ(A1=100;"Всегда";ЕСЛИ(И(A1>=80;A1<100);"Обычно";ЕСЛИ(И(И(A1>=60;A1<80);"Иногда";"Никогда"))))

Если значение в ячейке A1 является целым числом, формула читается следующим образом: "Если значение в ячейке A1 равно 100, вернуть строку "Всегда". В противном случае, если значение в ячейке A1 находится между 80 и 100, вернуть "Обычно". В противном случае, если значение в ячейке A1 находится между 60 и 80, вернуть строку "Иногда". И, если ни

одно из этих условий не выполняется, вернуть строку "Никогда". Всего допускается до 7 уровней вложения функций ЕСЛИ.

Функции ИСТИНА (TRUE) и ЛОЖЬ (FALSE) предоставляют альтернативный способ записи логических значений ИСТИНА и ЛОЖЬ. Эти функции не имеют аргументов и выглядят следующим образом:

=ИСТИНА()
=ЛОЖЬ()

Например, ячейка A1 содержит логическое выражение. Тогда следующая функция вернуть значение "Проходите", если выражение в ячейке A1 имеет значение ИСТИНА:

=ЕСЛИ(A1=ИСТИНА();"Проходите";"Стоп")

В противном случае формула возвратит "Стоп."

1.11 Ход работы

1. Изучить теоретический материал, описанный выше в данных методических указаниях.
2. Запустить редактор электронных таблиц MS Excel и создать новый документ.
- 3.



Рисунок 1.1 – Запуск MS Excel.

4. Создать таблицу, показанную ниже и используя статистические функции Excel рассчитать указанные параметры.

Показатели физического развития школьников

	Масса тела, кг	Рост, см	ОКГ, см		
	41	150	80		
	45	145	75		
	39	155	81		
	42	151	76		
	38	161	77		
	41	159	79		
	42	153	73		
	43	152	61		
	41	150	81		
	40	154	80		
Среднее	?	?	?		
Медиана	?	?	?		
Мода	?	?	?		
Линейное отклонение	?	?	?		
Дисперсия	?	?	?		
СКО	?	?	?		
Кол-во детей	?				
Макс. рост	?				
Мин. ОКГ	?				
Кол-во детей, масса которых больше среднего значения	?				

Рисунок 1.2 – Показатели физического развития школьников

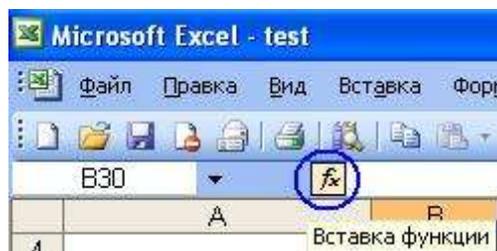


Рисунок 1.3 – Вставка функции

При этом следует воспользоваться такими статистическими функциями:

- СРЗНАЧ;
- МЕДИАНА;
- МОДА;
- СРОТКЛ;
- ДИСПР;
- СТАНДОТКЛОН.

Данные функции доступны в разделе «Статистические». См. рисунок 1.4.

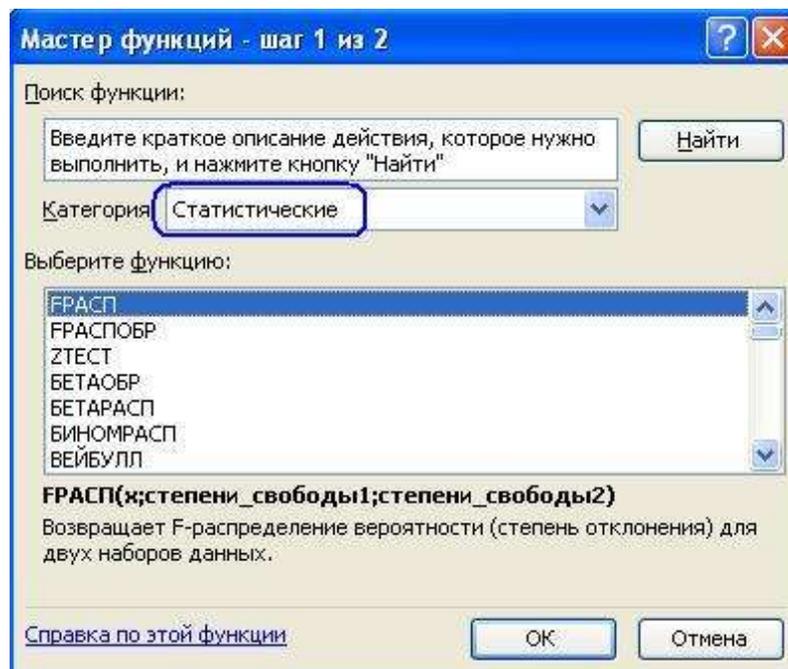


Рисунок 1.4 – Перечень статистических функций

Рисунок 1.5 демонстрирует использование функции СРЗНАЧ:

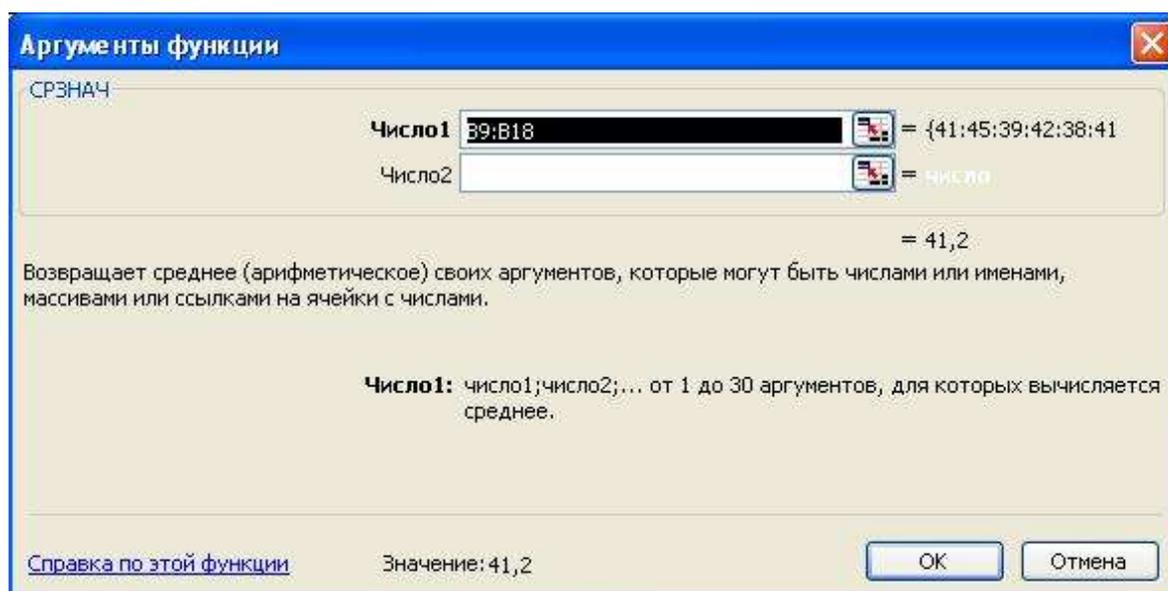


Рисунок 1.5 – Применение функции СРЗНАЧ

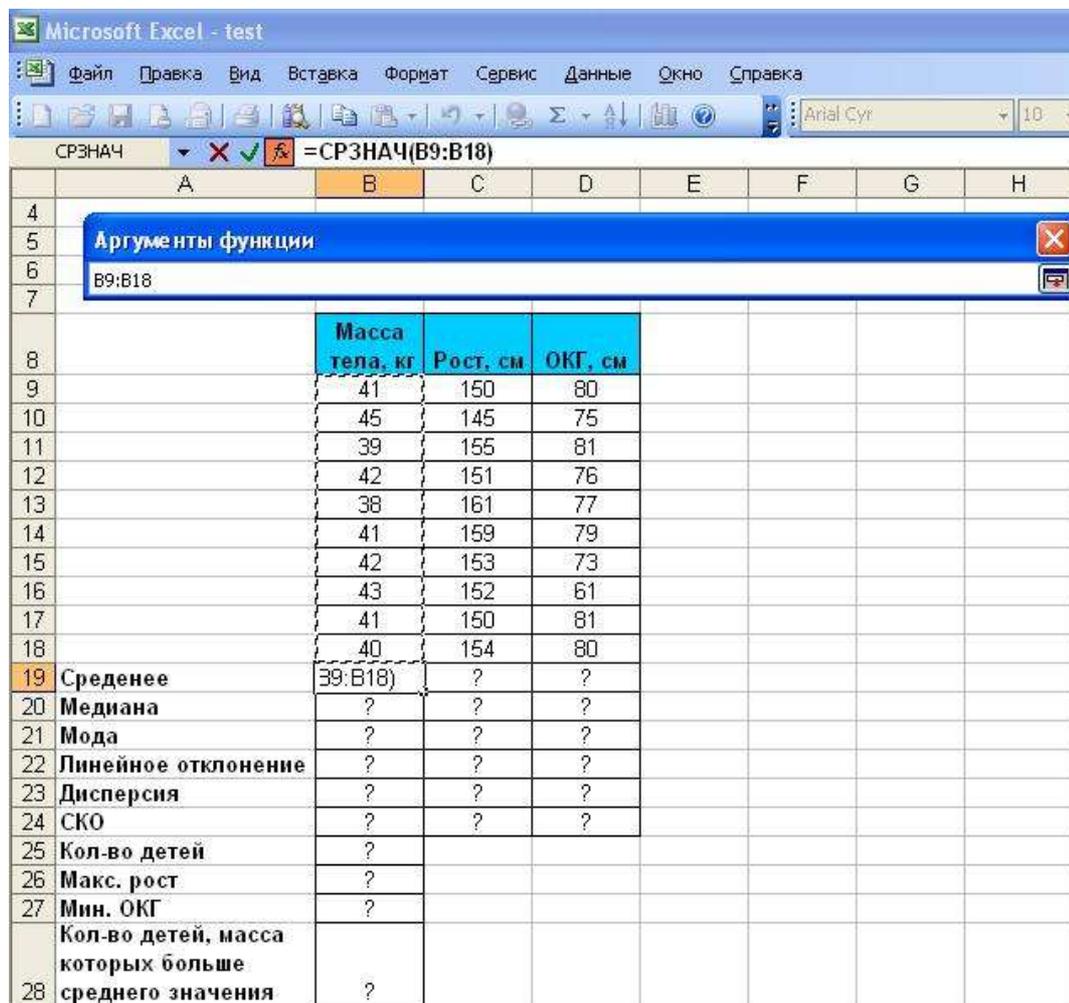


Рисунок 1.6 – Выбор диапазона аргументов функции

Аргументы функции выбираются путём выделения требуемого диапазона левой клавишей мыши. Для расчёта остальных параметров воспользоваться функциями:

- СЧЁТ;
- МАКС;
- МИН;
- СЧЁТЕСЛИ.

4. Используя описанный выше критерий Стьюдента, определить, существует ли зависимость нейролиновой кислоты в эритроцитах при болезнях крови. Ниже приведена таблица концентрации данной кислоты в крови больных и здоровых людей. Задать уровень значимости 0,05.

Концентрация нейролиновой кислоты

	Группа здоровых (X)	Группа больных (Y)
	21	16
	24	18
	18	19
	19	19
	25	22
	17	18
	18	19
	22	
Число степеней свободы		
Значение t_{α}		
Табличное значение t		
Вывод:		

Рисунок 1.7 – Проверка гипотезы о зависимости концентрации нейролиновой кислоты в крови больных и здоровых людей

По результатам расчётов сделать выводы.

5. Используя описанный выше критерий Фишера, определить влияние магнитных полей низкой частоты на размер опухоли карциномы Герена на четвёртый день заболевания.

6.

Размер опухоли Герена

	Размер опухоли интактной группы, см	Размер опухоли после действия ПМ НЧ, см
	0,027	0,075
	0,036	0,4
	0,1	0,08
	0,12	0,105
	0,32	0,075
	0,45	0,12
	0,049	0,06
	0,105	0,075
Значение F (расчётное)		
Значение F (табличное)		
Степень свободы X		
Степень свободы Y		
Вывод:		

Рисунок 1.8 – Проверка гипотезы о влиянии магнитных полей НЧ на размер опухоли Герена

По результатам расчётов сделать выводы.

7. Используя алгоритм построения таблиц истинности, описанный выше, построить таблицу истинности вручную и с использованием логических функций Excel для логической функции:

$$F = A \& (B \cup C)$$

При этом использовать логические функции:

- И;
- ИЛИ.

Логические функции

A	B	C	A(B+C)
0	0	0	ЛОЖЬ
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

Рисунок 1.9 – Таблица истинности логической функции

8. Рассчитать идеальную массу тела, а также индекс массы тела, и на основании расчётных данных поставить диагноз. Идеальная масса тела по Куперу определяется формулой:

$$\begin{cases} m = \left(\frac{\text{Рост} \times 3,5}{2,54} - 108 \right) \times 0,453 - \text{для женщин;} \\ m = \left(\frac{\text{Рост} \times 4,0}{2,54} - 128 \right) \times 0,453 - \text{для мужчин.} \end{cases}$$

Индекс массы тела (индекс Кетле) рассчитывается по формуле:

$$\text{ИМТ} = \frac{\text{Масса}}{\text{Рост}^2}$$

Интерпретация результатов:

$$\left\{ \begin{array}{l} \text{ИМТ} = 18,5 \div 24,9 \left(\frac{\text{кг}}{\text{м}^2} \right) - \text{норма}; \\ \text{ИМТ} = 25,0 \div 29,9 \left(\frac{\text{кг}}{\text{м}^2} \right) - \text{наличие лишнего веса}; \\ \text{ИМТ} = 30,0 \div 34,9 \left(\frac{\text{кг}}{\text{м}^2} \right) - \text{ожирение 1-й степени}; \\ \text{ИМТ} = 35,0 \div 39,9 \left(\frac{\text{кг}}{\text{м}^2} \right) - \text{ожирение 2-й степени}; \\ \text{ИМТ} > 40,0 \left(\frac{\text{кг}}{\text{м}^2} \right) - \text{ожирение 3-й степени}. \end{array} \right.$$

Идеальная масса тела							
№	ФИО	Пол	Масса, кг	Рост, см	ИМТ	Индекс Кетле	Заключение врача
1	Горлова	ж	79	164			
2	Карбышев	м	73	180			
3	Соручан	ж	62	165			
4	Малиновский	м	89	161			
5	Жданов	м	78	179			
6	Пушкарёва	ж	56	162			

Рисунок 1.10 – Диагностика ожирения

Для построение автоматизированного вывода использовать функции:

- ЕСЛИ;
- Вложенные конструкции функции ЕСЛИ;

1.12 Контрольные вопросы

1. Что изучает статистика?
2. Какие статистические пакеты Вы знаете? Привести примеры.
3. Что такое формула в Excel? Синтаксис формул.
4. Перечислить основные характеристики одномерного распределения. Какие функции Excel их рассчитывают?
5. Что такое статистическая гипотеза?
6. Что такое критерий Стьюдента? Для чего он применяется?
7. Что такое критерий Фишера? Для чего он применяется?
8. Что такое логическая функция? Дать определение.

9. Что такое таблица истинности логической функции? Как она создаётся с использованием логических функций Excel?

10. Как реализовываются условные конструкции Excel? Для чего они применяются?

Самостоятельная работа №2

Моделирование и фильтрация сигналов средствами MATLAB. Работа с пакетом Signal

Цель работы: получить навыки моделирования сигналов при помощи пакетов расширения Signal Processing Toolbox и Simulink среды Matlab.

2.1 Теоретические сведения

2.1.1 Моделирование сигналов при помощи пакета Signal Processing Toolbox

Пакет расширения Signal Processing Toolbox среды Matlab предназначен для обработки сигналов и изображений. При помощи этого пакета можно моделировать сигналы и обрабатывать их, в том числе осуществлять Фурье-преобразования, строить спектрограммы, периодограммы и т.д.

Сигнал – это зависимость одной величины от другой (функция). Для обработки на компьютере требуется перевести сигналы в цифровую форму. Один из способов сделать это – равномерно по времени измерить значения сигнала на определенном промежутке времени и ввести полученные значения амплитуд в компьютер. Если делать измерения достаточно часто, то по полученному дискретному сигналу можно будет достаточно точно восстановить вид исходного непрерывного сигнала. Как правило, непрерывные сигналы в системах компьютерной математики являются абстракцией. В них обычно задаются дискретные сигналы, определяемые в некоторые промежутки времени – чаще всего с постоянным шагом. Для описания таких сигналов в Matlab задается вектор времени, например: $t=0:0.1:10$. Здесь задано 11 отсчетов времени от 0 до 10 с шагом 0.1. Одиночные (или одноканальные) сигналы задаются как

$$y1=\sin(t)$$

$$y2=t$$

$$y3=t.^2$$

$$y4=\exp(-t)$$

Каждый из этих сигналов представляется вектором, размер которого равен размеру вектора времени t . В Matlab возможно и задание множественных (многоканальных) сигналов, например:

$$y_m = [y_1 \ y_2 \ y_3 \ y_4] \text{ или } y_m = [\sin(t) \quad t \quad t.^2 \quad \exp(-t)]$$

Такой сигнал представляется матрицей.

Для моделирования шумов используется генератор случайных чисел с тем или иным законом распределения. Наиболее важные сигналы в пакете Signal Processing Toolbox задаются специальными функциями, хотя любой сигнал можно создать средствами Matlab по подобию описанных выше примеров.

Моделирование сигналов – одна из важнейших задач пакета Signal Processing Toolbox среды Matlab. Многие сигналы являются детерминированными, т.е. их временная зависимость $y(t)$ аналитически определена и для любого момента времени t . Детерминированные сигналы удобны для аналитического описания и анализа систем обработки и преобразования сигналов и широко используются в роли тестовых сигналов. Однако большинство сигналов не являются детерминированными. Одной из причин этого является засоренность сигналов шумами и помехами, имеющими случайный характер с тем или иным законом распределения. Моделирование зашумленных сигналов в пакете Signal Processing Toolbox обеспечивается при помощи генератора случайных чисел. Другая причина заключена в самой сути сигналов как носителей информации. Разумным компромиссом между детерминированными и недетерминированными сигналами являются детерминированные сигналы с повторяющейся при каждом запуске моделирования шумовой компонентой. Это позволяет, с одной стороны, обеспечить учет влияния шума, а с другой – обеспечить повторяющиеся в процессе многократного моделирования результаты.

Далее рассмотрим некоторые примеры моделирования детерминированных и недетерминированных сигналов при помощи пакета Signal Processing Toolbox.

2.1.2 Построение зашумленного сигнала

Пусть нужно создать зашумленный сигнал, содержащий две синусоидальные компоненты – первой (основной) и третьей гармоник. Ниже приводится листинг программы для построения зашумленного сигнала.

```
t=(0:.01:2)';  
y=sin(2*pi*1*t)+0.25*sin(2*pi*3*t)  
randn('state',0);  
yn=y+0.5*randn(size(t));  
plot(t,y,t,yn)
```

Рисунок 2.1 – Листинг программы

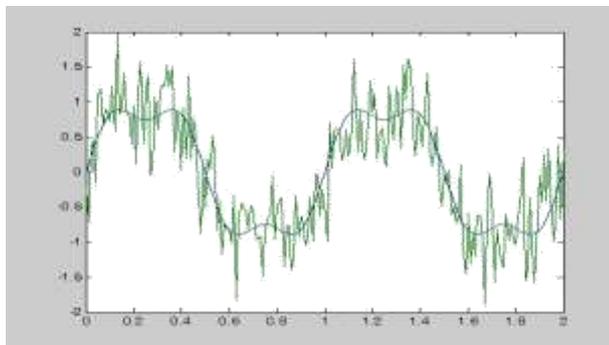


Рисунок 2.2 – Зашумленный сигнал

В первой строке программы задается вектор-столбец времени, что обеспечивает квантование сигнала по времени. Знак апострофа означает транспонирование вектора, т.е. превращение его из вектор-строки в вектор-столбец. Хотя на первых порах такое преобразование не имеет существенного значения, оно принципиально важно – многие функции Matlab требуют однозначного задания векторов либо в виде векторов-строк, либо в виде векторов-столбцов. Во второй строке программы задается вектор некоторой функции $y(t)$, которая представляет собой суммусинусоиды с амплитудой 1 и частотой 1 Гц с синусоидой, имеющей амплитуду 0.25 Гц и частоту 3 Гц. Третья строка предназначена для создания сигнала, на который наложен аддитивно шум, моделируемый с помощью генератора случайных чисел. Четвертая строка предназначена для построения вектора yn зашумленного сигнала. Результат представлен на рисунке 1. Из рисунка видно, насколько шум искажает форму сигнала.

2.1.2 Моделирование косинусоиды с переменной частотой

Для этих целей используется функция `chirp`. Она задается следующим образом:

$$y = \text{chirp}(t, f_0, t_1, f_1, ['\text{method}', \text{phi}]).$$

Функция `chirp` формирует выборку (дискретные значения) косинусоидального сигнала с частотой от f_0 в начальный момент времени t до f_1 в конечный момент времени t_1 . По умолчанию $t = 0$, $f_0 = 0$, $f_1 = 100$. Необязательный параметр `phi` (по умолчанию 0) задает начальную фазу сигнала. Другой необязательный параметр `'method'` задает закон изменения частоты. Этот параметр может принимать следующие значения:

1. `linear` - линейный закон изменения частоты
2. `quadratic` – квадратичный закон изменения частоты
3. `logarithmic` – логарифмический закон изменения частоты f

По умолчанию принято значение `method=linear`. Значения параметров по умолчанию используются, если соответствующая переменная отсутствует или задано пустое значение.

Пусть требуется задать косинусоидальный сигнал, частота которого меняется по полиномиальному закону. Ниже приводится листинг соответствующей программы.

```
t=[0 0.5 1.0 1.5 2.0] % задание вектора времени
f=[0 200 100 150 300]; % задание вектора частот
p=polyfit(t,f,4) % регрессия полиномом 4-го порядка
t=0:0.001:2; % задание вектора времени
y=chirp(t,p) % генерация сигнала и построение графиков
subplot(211); plot(t,polyval(p,t));set(gca,'ylim',[0 500]);
subplot(212); specgram(y,128,1E3,128,120);
```

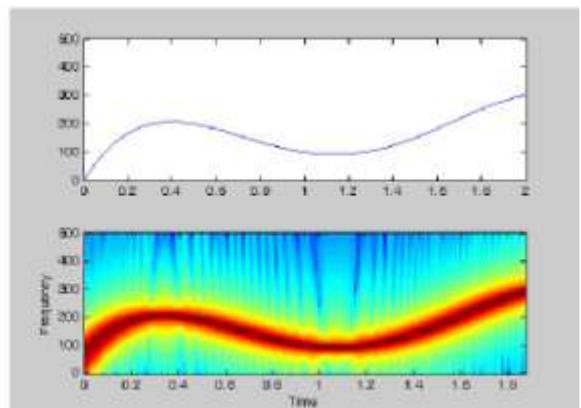


Рисунок 2.3 – Листинг программы

Функция `subplot` позволяет вывести в графическом окне сразу несколько графиков. Применение функции `subplot(m,n,p)` разбивает графическое окно на $m \times n$ подокон, при этом m – число подокон по вертикали, n – число подокон по горизонтали, p – номер текущего подокна.

В первых трех строчках модуля задано построение полинома 4-го порядка, описывающего функцию времени, которая используется для модуляции частоты косинусоидального сигнала – следующие две строки.

На рисунке справа приводятся графики моделирующей полиномиальной функции и спектрограмма сигнала (амплитудная составляющая вектора по времени), модулированного по заданному этой функцией закону. Вообще спектрограммой сигнала называют его мгновенный спектр, зависящий от времени. Из рисунка мы видим, что спектрограмма оконного преобразования Фурье идентифицирует модуляцию синусоидального сигнала, она строит модулирующую функцию. На рисунке приводятся графики моделирующей полиномиальной функции и спектрограмма сигнала (амплитудная составляющая вектора по времени), модулированного по заданному этой функцией закону.

2.1.3 Генерация импульсов

Для создания отсчетов импульсных сигналов разной формы служит функция `y=pulstran(t,d,'func'[p1,p2,...])`. Форма задается параметром `func`, который может иметь значения:

- `gauspuls` – синусоида, модулированная по закону Гаусса ;
- `rectpuls` – прямоугольный импульс;
- `tripuls` – треугольный импульс.

Вектор `y` вычисляется для отсчетов времени, заданных вектором `t`, по формуле $y = \text{func}(t-d(1)) + \text{func}(t-d(2)) + \dots$

Число импульсов в заданном интервале времени задается длиной вектора `d`, т.е. `length(d)`. Необязательные параметры `p1,p2,...` при необходимости позволяют задать дополнительные параметры обращения к `'func'`, например, типа `func(t-d(1), p1,p2,...)`. При записи функции в виде `y=pulstran(t,d,p,[fs])` можно задать частоту дискретизации `fs` (по умолчанию 1 Гц).

Листинг программы, построенной с использованием функции `pulstran`.

```

t=0:.00001:.005;
d=[0:.001:.01;0.5^(0:10)]';
y=pulstran(t,d,@gauspuls,5000,.5);
plot(t,y)

```

Рисунок 2.4 – Листинг программы

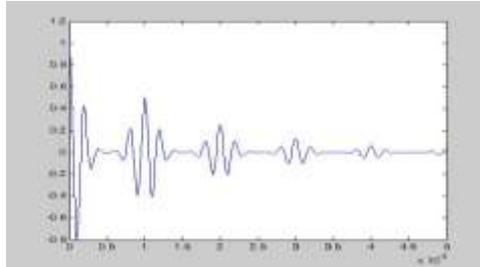


Рисунок 2.5 – График сигнала, построенного функцией pulstran.

2.1.4 Моделирование сигнала при помощи функции Дирихле

Функция Дирихле может рассматриваться как сигнал и описывается так $y=\text{diric}(x,n)$. Здесь n – целое положительное число, влияет на вид функции Дирихле.

```

x=0:0.1:20;n=10;
y=diric(x,n);
plot(x,y)

```

Рисунок 2.6 – Листинг программы

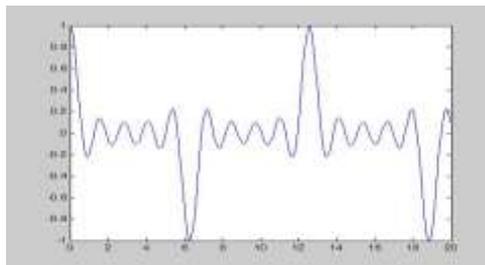


Рисунок 2.7 – Сигнал, построенный при помощи функции Дирихле

При моделировании сигналов этим методом целесообразно рассмотреть влияние параметра n на вид сигнала.

2.1.5 Моделирование трапециевидного импульса

Сформируем симметричный трапециевидный импульс с амплитудой 10 В и размерами верхнего и нижнего оснований 20 и

60 мс соответственно. Частота дискретизации равна 1 кГц. Листинг соответствующей программы приведен ниже.

```
Fs=1e3;  
t=-50e-3:1/Fs:50e-3;  
A=10;  
T1=20e-3;  
T2=60e-3;  
s=A*(T2*tripuls(t,T2)-T1*tripuls(t,T1))/(T2-T1);  
plot(t,s,'k');
```

Рисунок 2.8 – Листинг программы

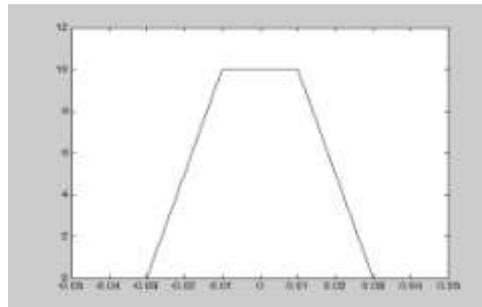


Рисунок 2.9 – Трапециевидный импульс

В результате на экран выводится одиночный трапециевидный импульс.

2.1.6 Моделирование прямоугольного импульса

Последовательность прямоугольных импульсов формируется при помощи функции `square`. Функция `square` принимает в общем виде два входных параметра – вектор значений времени `t` и параметр `duty`, с помощью которого можно регулировать скважность получаемой последовательности:

$$y=\text{square}(t,\text{duty}).$$

Параметр `duty` задает не саму скважность (скважность – это отношение периода к длительности импульса), а обратную ей величину – коэффициент заполнения (в процентах), т.е. отношение длительности импульса к периоду. Сформируем прямоугольный сигнал скважности 50% на интервале времени от 0 до 20 с.

```
t=0:0.1:20;
y=square(t,50);
plot(t,y)
ylim([-1.1 1.1]);
```

Рисунок 2.10 – Листинг программы

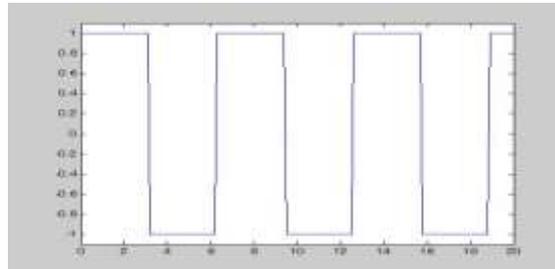


Рисунок 2.11 – Прямоугольный сигнал

2.1.7 Моделирование пилообразного сигнала

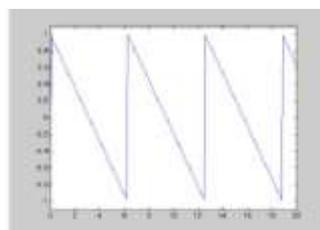
Для моделирования пилообразного сигнала применяется функция `sawtooth`, которая также имеет два входных параметра – вектор значений времени `t` и параметр `width`, при помощи которого можно регулировать длительность «обратного хода» – промежутка, на котором уровень сигнала линейно падает от 1 до -1. Значение параметра `width` должно быть в пределах от 0 до 1.

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

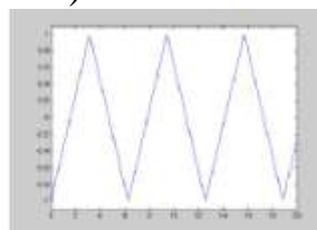
Сформируем треугольный сигнал на интервале времени от 0 до 20 с. Исследуем влияние параметра `width` на форму сигнала.

a)

```
t=0:0.1:20;
y=sawtooth(t,0);
plot(t,y)
ylim([-1.1 1.1]);
```



b)



c)

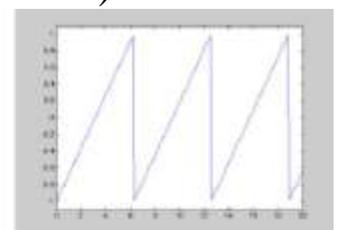


Рисунок 2.12 – Пилообразный сигнал. a) - `width=0`, b) `width=0.5`, c) `width=1`

Задание 1.

1. Используя листинги приведенных выше программ, получите при помощи пакета расширения Signal Processing Toolbox графики, показанные на рис.1-7.

2. Постройте синусоидальный, линейно нарастающий, квадратичный и экспоненциально спадающий сигналы.

2.2 Моделирование автоколебаний при помощи пакета Simulink

2.2.1 Пакет расширения Simulink, его назначение и возможности

Роль математического моделирования трудно переоценить. Математическое моделирование – это область науки и техники, которая обеспечивает выявление закономерностей протекания различных явлений окружающего нас мира или работы систем и устройств путем их математического описания и моделирования без проведения натурных испытаний. При этом используются фундаментальные положения и законы математики, описывающие моделируемые явления, системы или устройства на уровне их идеализации.

В состав системы MATLAB входит пакет моделирования динамических систем – Simulink. Пакет *Simulink* позволяет осуществлять моделирование и исследование поведения динамических нелинейных систем. Ввод характеристик исследуемых систем производится в диалоговом режиме, путем графической сборки схемы соединений элементарных стандартных звеньев. В результате такой сборки образуется модель исследуемой системы. Модель хранится в файле с расширением *.mdl*.

Создание моделей в пакете *Simulink* основано на использовании технологии *Drag-and-Drop*. В качестве "кирпичиков" для построения модели применяются модули (или блоки), хранящиеся в библиотеке *Simulink*. Любая модель может иметь иерархическую структуру, т.е. состоять из моделей более низкого уровня, причем число уровней иерархии практически не ограничено. В ходе моделирования имеется возможность наблюдать за процессами, происходящими в системе. Для этого

используются специальные смотровые окна, входящие в библиотеку *Simulink*.

Каковы основные свойства подсистемы *Simulink*? Подсистема *Simulink*:

- предоставляет возможность моделирования непрерывных, дискретных и гибридных - как линейных, так и нелинейных - систем
- включает в себя обширную библиотеку блоков (непрерывные элементы, дискретные элементы, математические функции, нелинейные элементы, источники сигналов, средства отображения), которые можно использовать для создания новых систем;
- позволяет объединять блок-диаграммы в составные блоки, что обеспечивает иерархическое представление структуры модели;
- содержит средства для создания блоков и библиотек, определяемых пользователем;
- дает возможность проектировать подсистемы, имеющие изменяемую во времени структуру.

Запуск подсистемы *Simulink* выполняется из основного окна, для чего необходимо щелкнуть левой кнопкой мыши на кнопке запуска этой подсистемы, находящейся в верхней части окна, либо набрать в основном окне команду *Simulink*. В том и другом случае откроется окно *Simulink Library Browser* (Система просмотра библиотек *Simulink*). В верхней части этого окна две крайние левые кнопки служат для создания новой и открытия существующей модели соответственно.

Процесс построения модели *Simulink* включает в себя компоновку и задание необходимых параметров. Компоновка заключается в выборе из библиотек *Simulink* необходимых блоков, размещение их в открывшемся окне и задание межблочных связей. Далее для каждого блока устанавливаются соответствующие параметры, отвечающие требованиям моделируемой системы.

2.2.2 Генератор Ван-дер-Поля как автоколебательная система

В природе многие процессы являются колебательными. Колебания – это движения или процессы, обладающие той или иной степенью повторяемости во времени. Колебания свойственны всем явлениям природы: пульсирует излучение звезд, внутри

которых происходят циклические ядерные реакции; с высокой степенью периодичности вращаются планеты Солнечной системы; движение Луны вызывает приливы и отливы на Земле; периодически бьется человеческое сердце. В физике выделяются колебания механические, акустические, электромагнитные и т.д. Все колебания можно подразделить на свободные (осуществляющиеся без воздействия внешней силы) и вынужденные (происходящие под действием некоторой периодически действующей внешней силы).

Можно выделить особый тип колебаний – автоколебания. Автоколебания представляют собой незатухающие колебания, поддерживаемые внешними источниками энергии в нелинейной диссипативной системе, вид и свойства которых определяются самой системой и не зависят от начальных условий. Автоколебания в динамической системе могут быть не только периодическими, но и квазипериодическими и даже стохастическими. Возможность существования периодического асимптотически устойчивого движения, которое изображается изолированной замкнутой траекторией в фазовом пространстве системы, к которой со временем притягиваются траектории из некоторой окрестности независимо от начальных условий, обеспечивается только в нелинейных диссипативных системах. Этот тип динамических систем настолько важен при изучении колебательных процессов, что для его выделения А.А.Андронов предложил специальный термин – автоколебательные системы.

Автоколебания могут быть получены при помощи генератора Ван-дер-Поля.

Автоколебания принципиально отличаются от других колебательных процессов в диссипативных системах тем, что для их поддержания не требуется периодических воздействий извне. Математическим образом автоколебаний является предельный цикл Пуанкаре – замкнутая траектория в фазовом пространстве, отвечающая периодическому движению.

2.2.3 Работа с демонстрационной моделью версии Simulink, описывающей генератор Ван-дер-Поля

Simulink имеет специальный редактор дифференциальных уравнений. С помощью которого можно задать систему

дифференциальных уравнений первого порядка явно в принятой форме Коши и тут же начать ее решение с помощью решателя. Для получения доступа к решателю надо загрузить файл dee, который находится в папке Matlab / toolbox / simulink / dee. Воспользуемся готовой демонстрационной моделью – файлом под именем deedemo1. Он выводит окно всего с двумя блоками – блоком vdp и осциллографом x1:

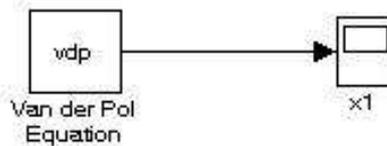


Рисунок 2.13 – Модель для описания колебаний, полученных при помощи генератора Ван-дер-Поля.

Первый блок решает заданное уравнение Ван-дер-Поля, а второй просто отображает решение в виде временной зависимости. Если дважды щелкнуть левой кнопкой мыши по первому блоку, откроется редактор дифференциальных уравнений. В этом окне можно модифицировать решаемые уравнения или ввести новые, менять значения начальных условий. Второй блок (осциллограф) предназначен для визуализации графических данных.

Для начала процесса моделирования нужно войти в меню модели Simulation и выбрать Start. На экране осциллографа будет наблюдаться динамическая картина колебаний, получаемых при помощи генератора Ван-дер-Поля. Чтобы остановить изображение колебаний на экране осциллографа, нужно нажать левой кнопкой мыши на изображении блок-схемы модели, или в меню модели воспользоваться командой Simulation/Pause. Параметры моделирования (время моделирования, погрешность и т.д.) могут быть изменены при помощи команды Simulation parameters в меню Simulation.

Задание 2.

1. Загрузите демонстрационную модель, описывающую колебания, полученные согласно уравнению Ван-дер-Поля. Получите осциллограмму колебаний.
2. Измените параметры моделирования. Получите осциллограмму колебаний.

3. Введите вместо уравнения Ван-дер-Поля другое дифференциальное уравнение и получите его решение.

2.3 Контрольные вопросы

1. Пакет Signal Processing Toolbox, его назначение.
2. Функции пакета расширения Signal Processing Toolbox.
3. Пакет расширения Simulink, его назначение.
4. Основные возможности пакета расширения Simulink.
5. Доступ к пакету расширения Simulink.
6. Автоколебания. Генератор Ван-дер-Поля.
7. Демонстрационная модель, описывающая колебания согласно уравнению Ван-дер-Поля.
8. Назначение блоков в модели и работа с ними.

Самостоятельная работа №3

Спектральный анализ дискретных сигналов. Средства визуализации спектра сигналов

Цель работы: проведение спектрального анализа с использованием быстрого преобразования Фурье (БПФ).

3.1 Теоретические сведения

Анализ – один из ключевых компонентов обработки сигналов. Основной целью анализа является сравнение сигналов друг с другом для выявления их сходства и различия.

Периодический сигнал любой формы можно разложить на синусоиды различных частот, или, наоборот, посредством соответствующего сложения синусоид с разными частотами возможно синтезировать сигнал какой угодно формы. Разложив сигнал на частотные составляющие, можно узнать, как возник первоначальный сигнал, по какому пути следовал, какому внешнему влиянию подвергался, то есть получить информацию для выяснения происхождения сигнала. Подобный метод анализа называется спектральным анализом или анализом Фурье.

Спектральный анализ с использованием преобразования Фурье позволяет охарактеризовать частотный состав исследуемых сигналов. Ниже представлено выражение (1) спектральной плотности сигнала в соответствии с преобразованием Фурье:

$$S(\omega) = \int_{-\infty}^{\infty} s(t)e^{-j\omega t} dt$$

где $s(t)$ – сигнал во временном пространстве;

ω – угловая частота;

$S(\omega)$ – интегральное преобразование Фурье.

Дискретное преобразование Фурье (ДПФ) – разновидность преобразования Фурье, специально предназначенная для работы с дискретными сигналами.

Выражение для ДПФ выглядит как:

$$X(n) = \sum_{k=0}^{N-1} x(k)e^{-j\frac{2\pi nk}{N}},$$

где $x(k)$ – последовательность отсчетов;

N – длина последовательности.

ДПФ – алгоритм, который необходимо знать, когда осуществляется анализ Фурье с помощью компьютера. Но обычно на практике специалисты не используют его из-за большого объема вычислений. Для вычисления одного коэффициента ДПФ по формуле (2) необходимо выполнить N комплексных умножений и сложений. А расчет всего ДПФ, содержащего N коэффициентов, потребует N^2 пар операций «умножение-сложение». Число операций возрастает пропорционально квадрату размерности ДПФ.

Если N не является простым числом и может быть разложено на множители, то можно разделить анализируемый набор отсчетов на части, вычислив их ДПФ и объединив результаты. Этот способ вычислений называется быстрым преобразованием Фурье (БПФ) и часто используется на практике.

При реализации БПФ возможно несколько вариантов организации вычислений: в зависимости от способа деления последовательности отсчетов на части (прореживание по времени или по частоте) и от того, на сколько фрагментов производится разбиение последовательности на каждом шаге (основание БПФ).

Следует отметить, что БПФ не является приближенным алгоритмом и дает тот же результат, что и исходная формула ДПФ. Применение БПФ имеет смысл, если число элементов в анализируемой последовательности является степенью числа 2.

Недостатки преобразования Фурье:

- ограниченная информативность анализа нестационарных сигналов и практически полное отсутствие возможностей анализа их особенностей (сингулярностей), т.к. в частотной области происходит «размазывание» особенностей сигналов (разрывов, ступенек, пиков и т.п.) по всему частотному диапазону спектра;
- появление эффекта Гиббса (пульсаций) на скачках функций, при усечениях сигналов и при вырезке отрезков сигналов для локального детального анализа;
- невозможность представить перепады сигналов с бесконечной крутизной (например, прямоугольные импульсы) плавной базисной функцией (синусоидой)

3.2 Методика проведения быстрого преобразования Фурье

— запустить в Matlab приложение **Signal Processing Tool** набором в командной строке

```
>> sptool;
```

— появится основное окно программы. В трех списках этого окна перечислены загруженные в программу SPTool сигналы (Signals), фильтры (Filters), спектры (Spectra);

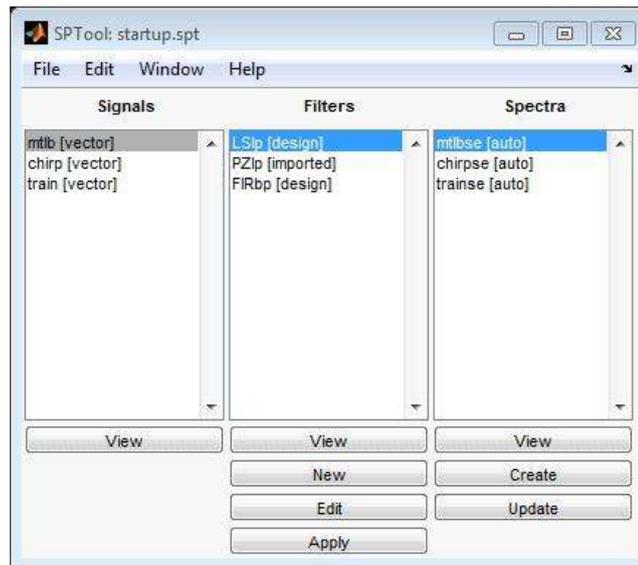


Рисунок 3.1 – Окно запуска Signal Processing Tool

— загрузить сформированный сигнал. Для этого в меню File главного окна программы SPTool выбрать команду Import. Появится окно Import to SPTool, изображенное на рисунке 3.2;

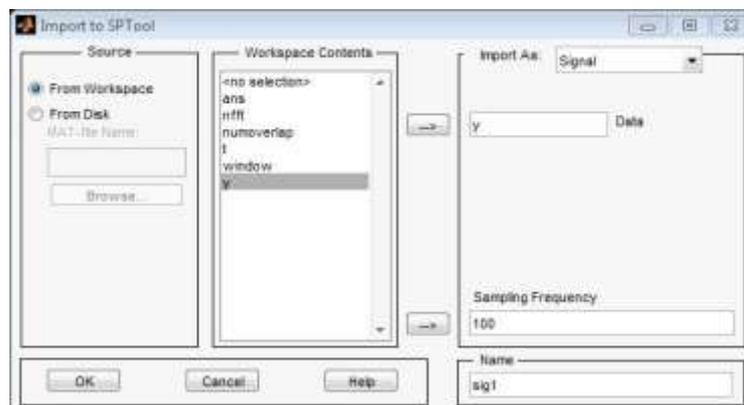


Рисунок 3.2 – Import to SPTool

— в раскрывающемся списке Import As выбрать вариант Signal;

- указать в списке переменную, содержащую отсчеты исследуемого сигнала, и щелкнуть на кнопку;
- в поле ввода Sampling Frequency ввести частоту дискретизации, значение которой обратно пропорционально шагу времени;
- в поле ввода имени Name отредактировать имя сигнала;
- щелкнуть на кнопку Ok, импортированный сигнал появится в списке Signal основного окна программы;
- выбрать нужный сигнал в списке Signals и щелкнуть на кнопке Create, появится окно Spectrum Viewer, показанное на рисунке 3.3;

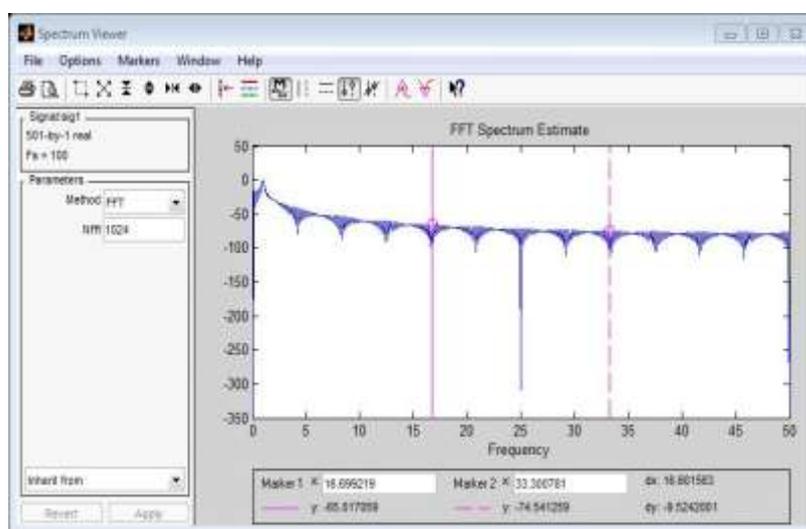


Рисунок 3.3 — Окно Spectrum Viewer

- в левой части окна выбрать метод спектрального анализа, количество точек, щелкнуть на кнопку Apply. Будет рассчитана оценка спектра сигнала и выведен соответствующий график.

3.3 Контрольные вопросы

1. Что такое спектральный анализ?
2. Назовите выражение спектральной плотности сигнала в соответствии с преобразованием Фурье.
3. Дайте определение ДПФ. Когда его применяют.
4. В каких случаях используется быстрое преобразование Фурье?
5. Перечислите недостатки преобразования Фурье.

Самостоятельная работа №4

Работа с пакетом Wavlet Toolbox

Цель: Получить навыки фильтрации сигналов на основе вейвлет-анализа.

4.1 Теоретические сведения

4.1.1 Методика шумоочистки с помощью вейвлетов

Зашумленный сигнал может быть представлен в следующем виде:

$$s(n) = f(n) + \sigma e(n).$$

Процесс шумоочистки заключается в подавлении части сигнала, содержащей шум для восстановления исходного сигнала f .

Данный метод является эффективным для семейства функций f , которые имеют несколько ненулевых вейвлет-коэффициентов. Например, функция гладкая почти всюду имеющая несколько резких изменений.

Со статистической точки зрения, модель данного сигнала представляет собой модель регрессии и шумоочистка может рассматриваться как непараметрическое оценивание функции f используя ортогональный базис.

Процесс шумоочистки может быть разбит на 3 этапа

1. Декомпозиция сигнала.

Выбор вейвлета, выбор уровня N . Вычисление вейвлет-декомпозиции сигнала s до уровня N .

2. Ограничение детализирующих коэффициентов

Для каждого уровня от 1 до N , выбирается порог и выполняется мягкое ограничение детализирующих коэффициентов.

3. Реконструкция

Восстановление сигнала используя исходные коэффициенты аппроксимации уровня N и измененные детализирующие коэффициенты уровней от 1 до N .

Рассмотрим два вопроса: как выбрать ограничительный порог и как выполнить ограничение коэффициентов.

Выбор порога

Для ограничения коэффициентов может быть использована функций Matlab.

`yt = wthresh(y,sorh,thr)`

`y` – массив коэффициентов

`sorh` – способ ограничения: ‘s’ – мягкий (soft), ‘h’ – жесткий (hard)

`thr` – значение порога, коэффициенты которые меньше `thr` приравниваются к 0.

Пусть `t` – значение порога.

Жесткое ограничение коэффициентов:

$$yt = \begin{cases} y, & \text{если } |y| > t \\ 0 & \text{если } |y| \leq t \end{cases}$$

Мягкое ограничение коэффициентов:

$$yt = \begin{cases} \text{sign}(y)(|y| - t), & \text{если } |y| > t \\ 0 & \text{если } |y| \leq t \end{cases}$$

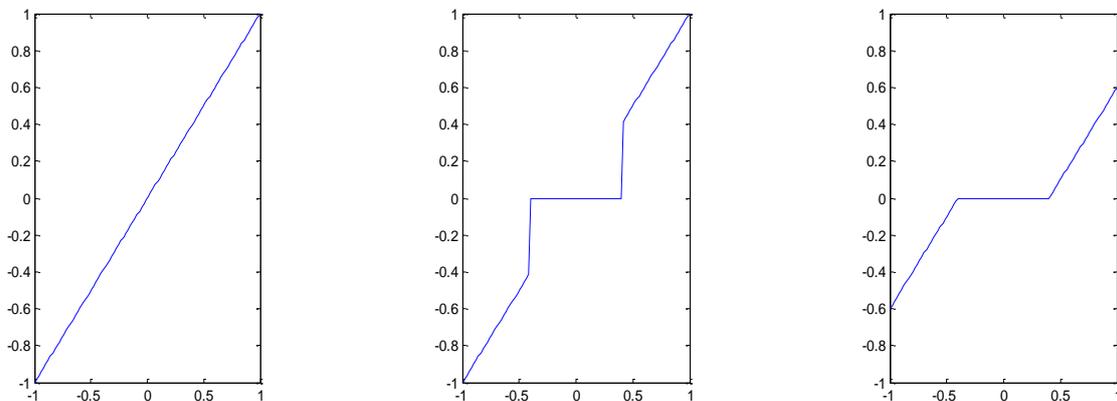


Рисунок 4.1 - Графики

Методы определения порогового значения

В MATLAB реализована функция выбора порога для известной базовой модели шума (белый шум с диапазоном [-1,1])

`thr = thselect(y,tpr)`

Которая позволят выбрать порог на основе одного из 4 правил выбора порогового значения.

Правила задаются параметром `tptr`.

Таблица 4.1 – Правила параметров

Параметр <code>tptr</code>	Правило
<code>'rigrsure'</code>	Установка порога на основе принципа несмещенной оценки риска Штейна (Stein's Unbiased Risk Estimation – SURE)
<code>'sqrtwolog'</code>	Фиксированный порог равный $\sqrt{2 \cdot \log(\text{length}(s))}$
<code>'heursure'</code>	Эвристический вариант метода <code>'rigrsure'</code>
<code>'minimaxi'</code>	Минимаксный принцип выбора порога

Обычно модель шума не может быть установлена точно. В данном случае можно использовать функцию

$$\text{sd} = \text{wden}(s, \text{tptr}, \text{sorh}, \text{scal}, n, \text{wav}).$$

Функция `wden` возвращает очищенный от шума сигнал `sd`, полученный ограничением вейвлет-коэффициентов входного сигнала `s`.

`tptr` – определяет правило выбора порога (см выше);

`sorh` – `'s'` или `'h'` (см выше)

`scale` – определяет модель шума.

Таблица 4.2 – Определение соответствующей модели по параметру `scale`

Параметр <code>scale</code>	Соответствующая модель
<code>'one'</code>	Базовая модель (перемасштабирование отсутствует)
<code>'sln'</code>	Базовая модель с ненормированным шумом.
<code>'mln'</code>	Базовая модель с небелым шумом

где `n` – уровень разложения, `wav` – название вейвлета.

4.1.2 Использование GUI

Для выполнение шумоочистки можно использовать графический интерфейс пользователя GUI.

1. Вызвать GUI Wavelet Toolbox >> wavemenu.
2. Вызвать окно 1D Wavelet (Рисунок 4.2).

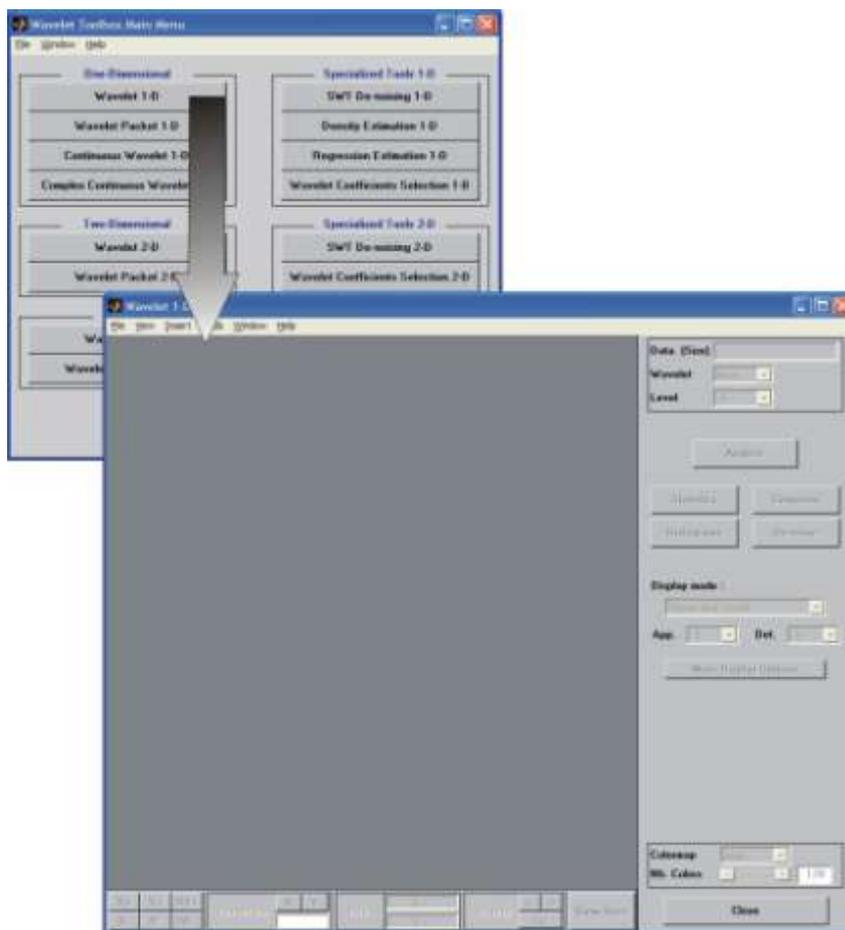


Рисунок 4.2 – Вызов окна 1D Wavelet

3. Загрузить сигнал и выполнить декомпозицию сигнала с выбранными вейвлетом и максимальным уровнем разложения. Далее вызвать окно De-Noise (Рисунок 4.3).

4. Задать метод выбора порога и тип шума. Нажать кнопку De-Noise (Рисунок 4.4).

Порог для каждого из уровней может быть задан вручную как для всего сигнала так и для участков (кнопка Int.(erval) Depended Threshold Settings).

5. Нажать кнопку CLOSE. Далее будет предложено обновить синтезированный сигнал.

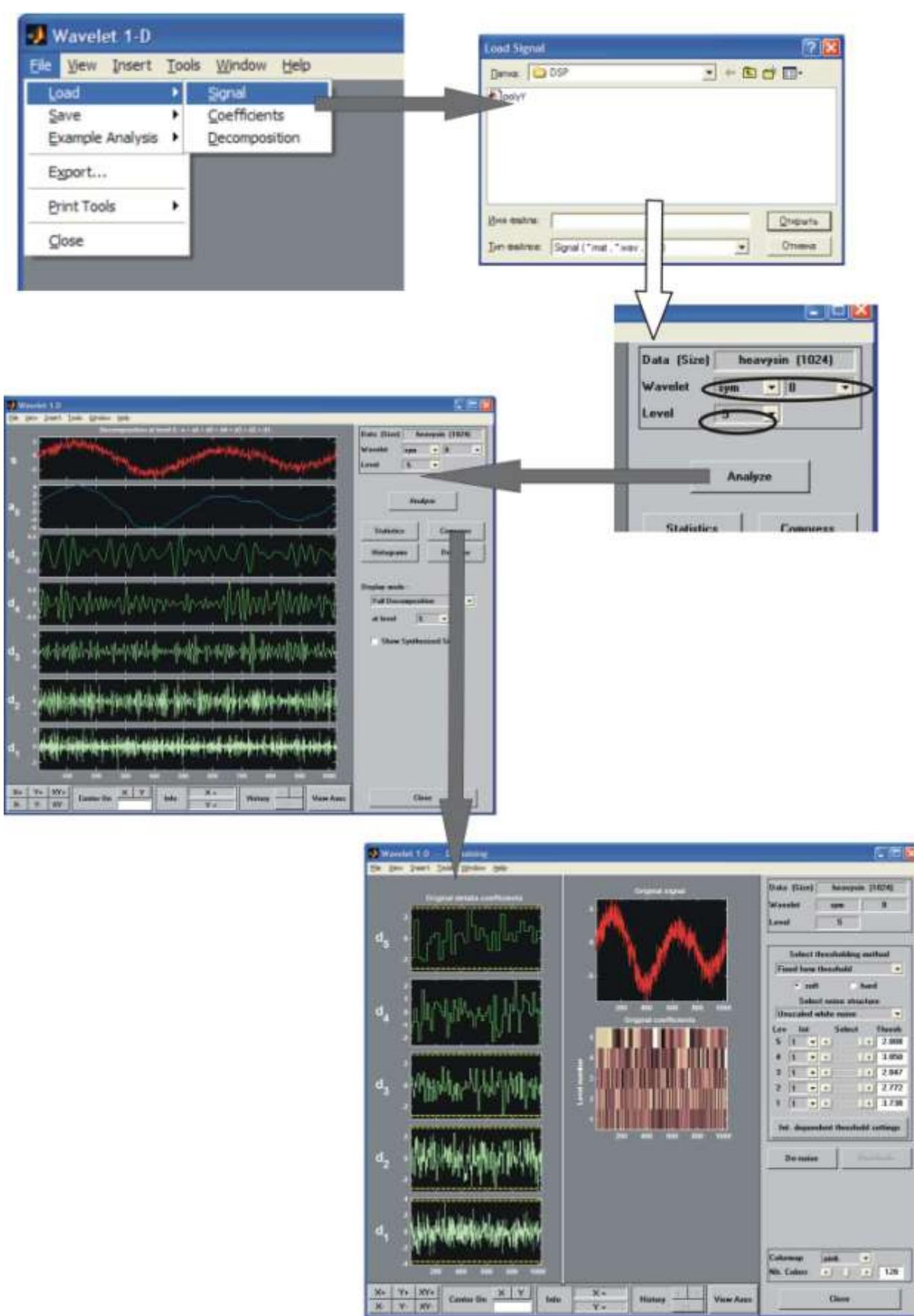


Рисунок 4.3 – Пошаговая выполнение пункта 3

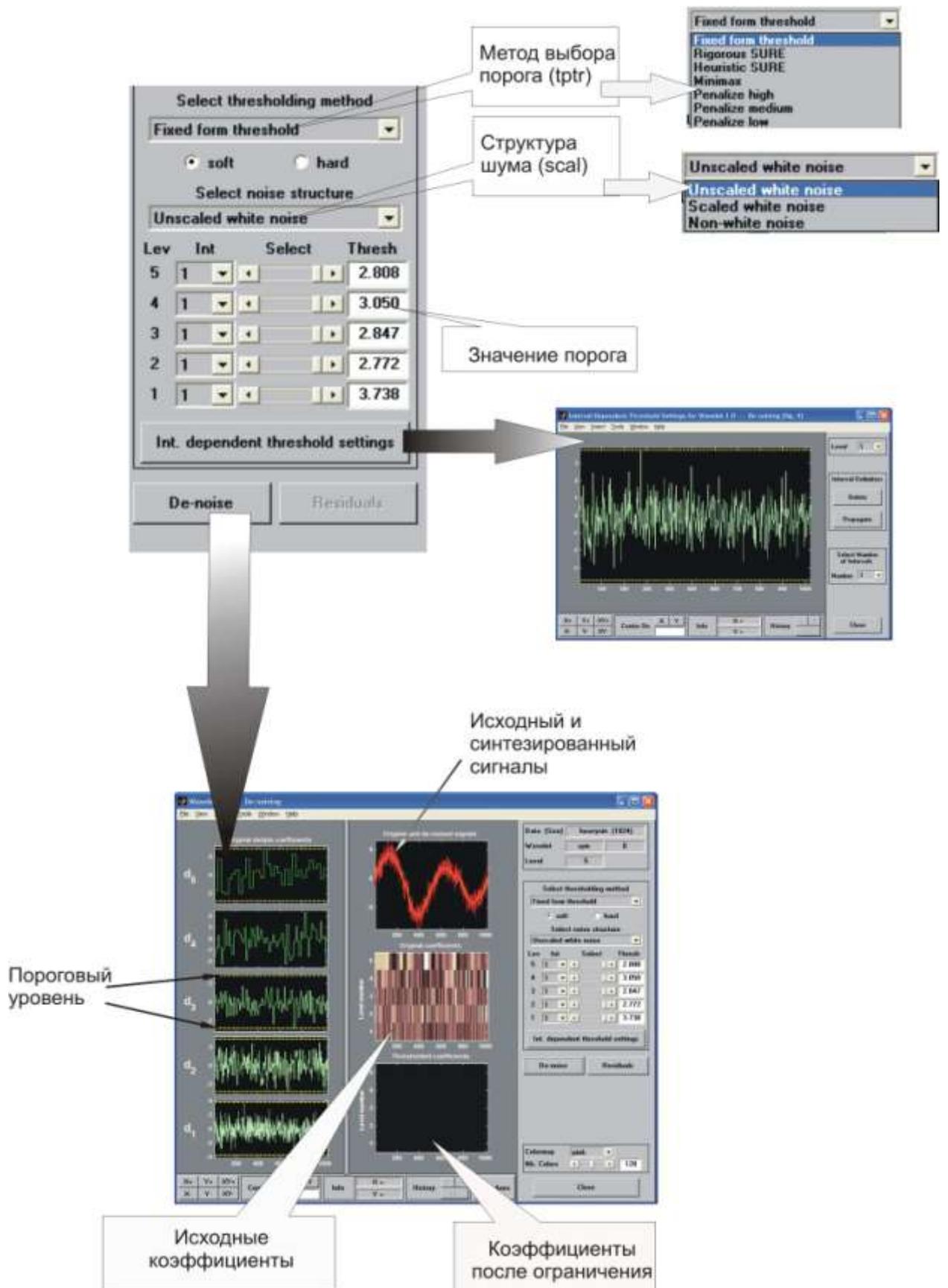


Рисунок 4.4 – Пошаговая задача метода выбора порога и типа шума

4.1.3 Пример шумоочистки сигнала

В качестве примера рассмотрим демонстрационный сигнал из пакета Wavelet Toolbox – Noisy Doppler (noisdopp.m).

Проводим анализ сигнала. Для этого выбираем вейвлет `sym4`. Число уровней разложения – 5 (Рисунок 4.5).

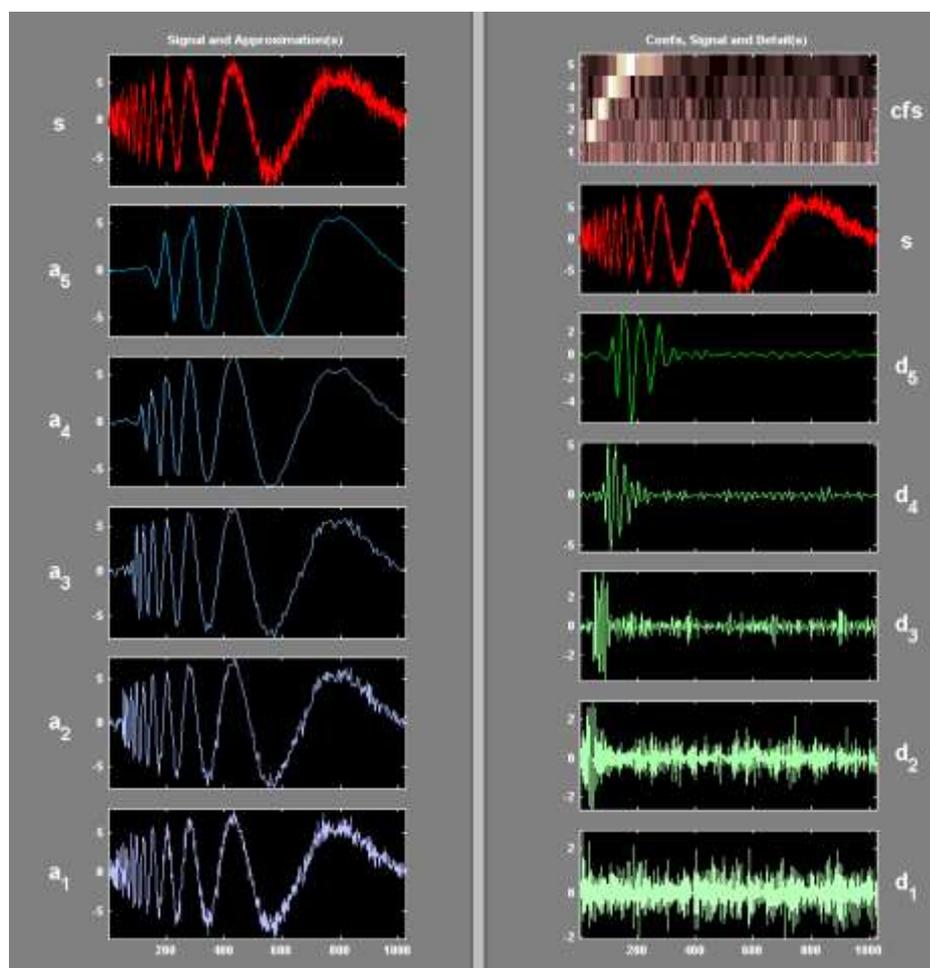


Рисунок 4.5 – Анализ сигнала

Заметим, что высокие частоты появляются в начале исходного сигнала. Каждый уровень аппроксимации содержит все меньше и меньше шума, при этом мы также теряем высокочастотную составляющую сигнала. На уровне аппроксимации A_5 , например, потеряно около 20% сигнала.

Нажав кнопку **De-noise** получаем окно, содержащее детали разложения с автоматически выставленным порогом (рисунок 4.6). В этом окне нажимаем кнопку **De-noise** и получаем синтезированный сигнал с учетом выставленных ограничений на коэффициенты (рисунок 4.7).

Заметим, что сигнал очищенный от шума данным методом в начале близок к 0 (рисунок 4.6). Некоторые высокочастотные составляющие были потеряны в процессе шумоочистки.

Повторите этот пример самостоятельно, используя различные вейвлеты и ограничительные пороги.

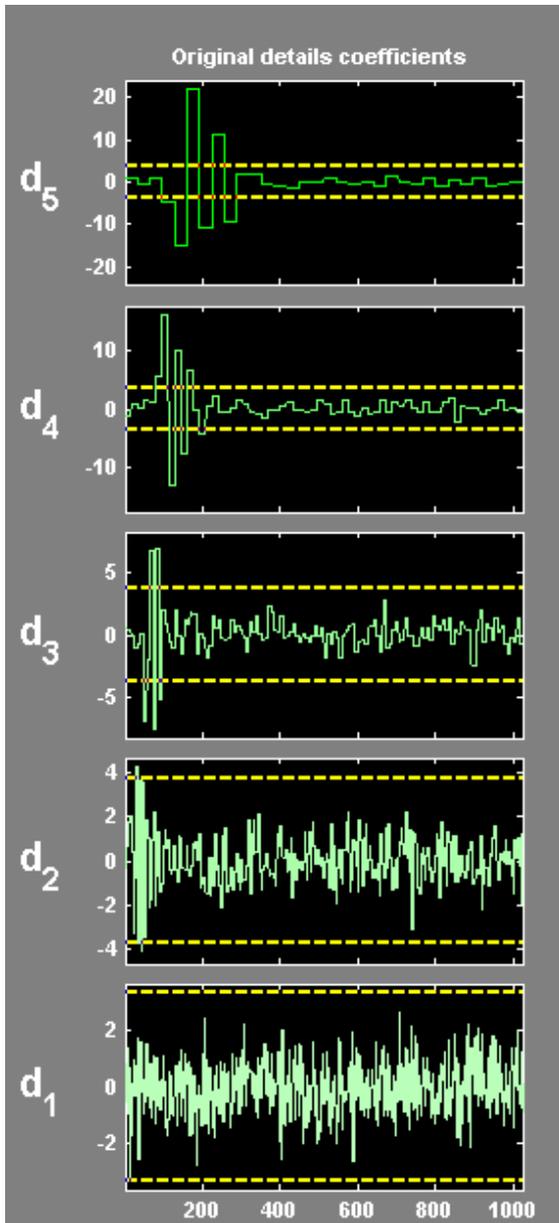


Рисунок 4.6 – Окно, содержащее детали разложения с автоматически выставленным порогом

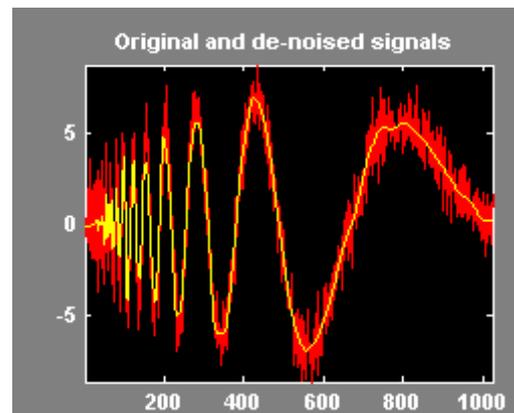


Рисунок 4.7 – Синтезированный сигнал с учетом выставленных ограничений на коэффициенты

4.2 Задание для самостоятельной работы

1. Выполните вейвлет-преобразование шумов с различными характеристиками. Как изменяются вейвлет-коэффициенты с изменением уровня? Какое распределение имеют коэффициенты?

2. Для заданных сигналов вычислить пороговое значение различными способами.

3. В соответствии с заданием выполните подавление заданной компоненты сигнала на основе вейвлет-анализа

4.

СИГНАЛ	Варианты						
	1	2	3	4	5	6	7
	Компоненты для подавления подчеркнуты						
$\sin(\alpha\pi) + \sin(\beta\pi) + \sin(\gamma\pi)$	<u>$\alpha=5$</u> $\beta=10$ $\gamma=15$	$\alpha=5$ <u>$\beta=10$</u> $\gamma=15$	<u>$\alpha=5$</u> $\beta=10$ <u>$\gamma=15$</u>	<u>$\alpha=25$</u> $\beta=35$ $\gamma=55$	<u>$\alpha=25$</u> <u>$\beta=35$</u> $\gamma=55$	$\alpha=25$ <u>$\beta=35$</u> <u>$\gamma=55$</u>	<u>$\alpha=5$</u> $\beta=15$ <u>$\gamma=45$</u>
$y = \begin{cases} \sin(\alpha\pi), & t \leq 10 \\ \sin(\beta\pi) & t > 10 \end{cases}$	<u>$\alpha=5$</u> $\beta=10$	$\alpha=5$ <u>$\beta=10$</u>	<u>$\alpha=25$</u> $\beta=40$	$\alpha=25$ <u>$\beta=40$</u>	<u>$\alpha=55$</u> $\beta=10$ 0	$\alpha=55$ <u>$\beta=10$</u> <u>0</u>	<u>$\alpha=75$</u> $\beta=10$
$y = \begin{cases} \sin(\alpha\pi) + \sin(\beta\pi), & t \leq \\ \sin(\beta\pi) + \sin(\gamma\pi) & t > \end{cases}$	<u>$\alpha=5$</u> $\beta=10$ $\gamma=15$	$\alpha=5$ <u>$\beta=10$</u> $\gamma=15$	<u>$\alpha=5$</u> $\beta=10$ <u>$\gamma=15$</u>	<u>$\alpha=25$</u> $\beta=35$ $\gamma=55$	<u>$\alpha=25$</u> <u>$\beta=35$</u> $\gamma=55$	$\alpha=25$ <u>$\beta=35$</u> <u>$\gamma=55$</u>	<u>$\alpha=5$</u> $\beta=15$ <u>$\gamma=45$</u>

5. Выполнить очистку сигнала от шума.

Var. 1	Var. 2	Var.3	Var.4	Var.5	Var.6	Var.7
Сигналы загрузить из файлов (\MATLAB6p5\toolbox\wavelet\wavedemo\)						
sumsin. mat	noisbloc. mat	noischir. mat	noisbum p. mat	noismim a. mat	noisbloc. mat	noisbloc. mat

4.3 Контрольные вопросы

1. Какими преимуществами обладает вейвлет-фильтрация?
2. Как может быть реализована адаптивная фильтрация?

4.4 Список литературы

1. Геппенер В.В., Черниченко Д.А. Экало С.А. Вейвлет-преобразование в задачах цифровой обработки сигналов.: Учеб. пособие. СПб.:Изд-во СПбГЭТУ "ЛЭТИ", 2002. 64 с.
2. Добеши И. Десять лекций по вейвлетам. – Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001, 464 с.
3. Дьяконов В. Matlab. Обработка сигналов и изображений. Специальный справочник. – СПб.: Питер, 2002. – 608 с.
4. Дьяконов В. Вейвлеты. От теории к практике. – М.: Солон-Р, - 2002. – 448 с.
5. Чуи Ч. Введение в вэйвлеты: Пер. с англ. – М.: Мир, 2001.- 412 с.
6. Справка пакета Wavelet Toolbox (ug_wavelet.pdf)

Самостоятельная работа № 5

Цифровая обработка изображений средствами MATLAB

Цель работы – изучение операций по сегментации изображений, функций, реализующих операции по сегментации изображения, и приобретение практических навыков использования этих функций.

5.1 Теоретические сведения

Сегментация изображения представляет собой разделение изображения на области по сходству свойств (признаков) в их точках. Признаки подразделяются на естественные и искусственные. Естественные признаки устанавливаются простым (визуальным) анализом изображения, а искусственные – в результате специальной обработки различных измерений. Примерами естественных признаков являются структура, текстура, яркость объекта. Примеры искусственных признаков: гистограммы распределения яркости, спектр и др.

К основным видам сегментации изображений относится сегментация по яркости, цветовым координатам, контурам, форме.

5.2 Методы сегментации

5.2.1 Сегментация методом выращивания областей

Для сегментации изображения можно использовать метод выращивания областей – группирование пикселей или подобластей в более крупные области по заранее заданным критериям роста. Берутся «центры кристаллизации», а затем на них наращиваются области путем добавления к каждому центру тех соседних пикселей, которые по своим свойствам близки к центру кристаллизации (например, имеют яркость или цвет в определенном диапазоне). Ниже приведена функция `regiongrow`, которая выполняет выращивание областей.

Синтаксис

[g, NR, SI, TI] = regiongrow(f, S, T),

где f – это сегментируемое изображение, а параметр S – массив (с размерами как у f) или скаляр. Если S – массив, то он содержит 1 в тех позициях, где расположены центры кристаллизации и 0 во всех остальных местах. Если S является скаляром, то он задает значение яркости пикселей, которые становятся центрами кристаллизации. Аналогично, T может быть массивом (с размерами, как у f) или скаляром. Если T – массив, то его элементы являются локальными пороговыми значениями для f . Скаляр T определяет глобальный порог.

Задание 1. Создать функцию `regiongrow`.

```
function [g, NR, SI, TI] = regiongrow(f, S, T)
if numel(S) == 1
SI = f == S;

S1 = S
else
SI = bwmorph(S, 'shrink', Inf);
J = find(SI);
S1 = f(J);
end
TI = false(size(f));
for K = 1:length(S1)
seedvalue = S1(K);
S = abs(f - seedvalue) <= T;
TI = TI | S;
end
[g, NR] = bwlabel(imreconstruct(SI, TI));
```

Задание 2. Выполнить сегментацию наращиванием областей для изображения, хранящегося в файле `Finance.bmp`, используя функцию

```
regiongrow.
[x,map] =
imread('c:\image\Finance.bmp'); I =
im2double(ind2gray(x,map));
figure,imshow(I)
S = 0.9783; T = 0.0651; % эти значения находятся
экспериментально (взяты из изображения)
[g, NR, SI, TI] = regiongrow(I, S, T);
```

figure,imshow(TI)

Также для сегментации используется метод разделения. Функция MatLab, реализующая подобный алгоритм, приведена ниже.

5.2.2 Сегментация методом разделения

Изображение разбивается на непересекающиеся блоки, которые с помощью некоторого критерия проверяются на однородность.

Функция сегментации полутоновых изображений методом разделения qtdecomp.

Синтаксис

$A = \text{qtdecomp}(I, \text{threshold}, \text{mindim})$

Функция qtdecomp осуществляет сегментацию полутоновых изображений методом разделения. В функции qtdecomp каждый блок разбивается на 4 неперекрывающихся блока одинакового размера. На первом шаге алгоритма блоком считается все изображение. Мельчайшим по размерам является блок, который нельзя разделить на 4 блока одинакового размера, т. е. такой блок, у которого число строк или число столбцов нечетное. Таким образом, в функции qtdecomp рекомендуется использовать изображения с размерами, равными степеням двух. Функция $A = \text{qtdecomp}(I, \text{threshold}, \text{mindim})$ осуществляет сегментацию полутонового изображения I методом разделения и помещает результат в разреженный массив A (тип данных sparse MatLab). Элементам матрицы $A(r,c)$, соответствующим координатам левых верхних углов блоков на исходном изображении I , присваиваются значения, определяющие размеры каждого блока. Блок считается однородным, если разница между максимальным и минимальным значением пикселей блока меньше параметра threshold. Параметр mindim определяет минимальный размер блока.

Функция получения блоков из квадродерева результатов сегментации qtgetblk

Синтаксис

$[vals, idx] = qtgetblk(I, A, dim)$

Функция возвращает в массив `vals` все блоки размером `dim`, получившиеся в результате сегментации полутонового изображения `I` с помощью функции `qtdecomp`. В параметре `A` передается разреженный массив, описывающий quadro-дерево с результатами сегментации. Координаты левых верхних углов блоков, помещенных в массив `vals`, находятся в векторе `idx`. Если нет ни одного блока размером `dim`, то всем возвращаемым параметрам присваиваются значения пустых `vals` матриц.

Функция замены блоков – результатов сегментирования `qtsetblk`

Синтаксис

$I_D = qtsetblk(I_S, A, dim, vals)$

Функция создает новое полутоновое изображение `ID`, заменяя в исходном полутоновом изображении все блоки размера `dim`, полученные в результате сегментации с помощью функции `qtdecomp`, на блоки из массива `vals`. В параметре `A` передается разреженный массив, описывающий quadro-дерево с результатами сегментации. Данная функция используется для преобразования изображения в соответствии с результатами сегментации методом разделения.

Рассмотрим работу функции `qtdecomp` совместно с функциями `qtgetblk` и `qtsetblk` для полутонового изображения размера 8×8 пикселей. Формат представления данных – `uint8`. Будем считать, что блок изображения является однородным, если величина разброса яркостей пикселей в блоке не превышает 10 градаций яркости. Установим минимально возможный размер блока. В нашем случае он равен двум.

Будем считать, что к объекту относятся блоки, средняя яркость которых не превышает 50. Требуется изменить исходное изображение так, чтобы пикселям блоков, относящихся к объекту, было присвоено значение 1, а пикселям блоков, не относящихся к объекту, 0.

Задание 3. Выполнить сегментацию небольшого текстового изображения методом разделения.

Исходное изображение:

```
>> I = [ 10 11 10 15 20 25 47 51
        11 14 17 13 27 29 52 55
        12 13 11 10 24 47 56 60
        13 14 11 13 49 54 74 77
        15 16 43 48 79 82 87 86
        17 18 45 50 85 80 80 84
        29 51 50 59 80 83 83 85
        59 1 58 61 81 85 86 88 ];
```

Сегментация методом разделения: размер минимального блока 2×2 ; блок считается однородным, если в его пределах яркость изменяется меньше, чем на 10 градаций.

```
>> A = qtdecomp(I,10,2);
```

Для удобства визуального анализа предварительно преобразуем разреженную матрицу A в обычную матрицу M с помощью функции `full`.

```
>> M =
>> full(A) M=
    4 0 0 0 2 0 2 0
    0 0 0 0 0 0 0 0
    0 0 0 0 2 0 2 0
    0 0 0 0 0 0 0 0
    2 0 2 0 4 0 0 0
    0 0 0 0 0 0 0 0
    2 0 2 0 0 0 0 0
    0 0 0 0 0 0 0 0
```

В результате сегментации получили 2 блока размером 4×4 (левая верхняя и правая нижняя части изображения и 8 блоков размером 2×2).

Переберем в цикле все возможные размеры блоков: 8, 4, 2.

```
>> dim = 8;
>> while dim >= 2
% получить в переменной blocks все блоки
размера dim. [blocks,idx] = qtgetblk(I,A,dim);
[x y n] = size(blocks);
```

```

%     если блоки такого размера есть в квадрате-дереве,
%     if n>0
%     то перебираем все блоки размера dim
%     for j = 1:n
%     если среднее значение яркости пикселей в пределах блока
меньше 50
%     if (mean2(blocks(:,:,j))<50)
%     то заменяем значения всех пикселей
блока на 1, blocks(:,:,j) = ones(dim,dim);
%     else

%     иначе заменяем значения всех
пикселей на 0. blocks(:,:,j) = zeros(dim,dim);
%     end;
%     end % end for

%     устанавливаем новые значения всех пикселей
размера dim I = qtsetblk(I,A,dim,blocks);
%     end; % end if
%     dim = dim/2;
%     end % end while      % ссылка на 1 из списка литературы

%     получившееся
изображение I >> I
I =
1 1 1 1 1 1 0 0
1 1 1 1 1 1 0 0
1 1 1 1 1 1 0 0
1 1 1 1 1 1 0 0
1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

```

Задание 4. Выполнить сегментацию реального изображения из файла cotton3.bmp

```

rgb = imread('c:\Image\cotton3.bmp');
I = im2double(rgb2gray(rgb));
figure,imshow(I)
T=graythresh(I);
A = qtdecomp(I,0.1,2);

```

```

dim = 8;
while dim >= 2
[blocks,idx] = qtgetblk(I,A,dim);
[x y n] = size(blocks);
if n>0
for j = 1:n
if (mean2(blocks(:,:,j))<T)
blocks(:,:,j) = ones(dim,dim);
else
blocks(:,:,j) = zeros(dim,dim);
end
end
I1 = qtsetblk(I,A,dim,blocks);
end
dim = dim/2;
end
figure,imshow(I1)

```

Функция выбора интересующей области по цвету roicolor

Синтаксис

```
BW = roicolor(S,low,high)
```

```
BW = roicolor(S,v)
```

Для любого варианта вызова функции roicolor бинарное изображение формируется по следующему алгоритму: пикселу бинарного изображения BW(r, c) присваивается значение 1, если яркость пиксела S(r, c) исходного полутонового изображения или индекс S(r, c) палитрового изображения принадлежит диапазону [low, high] или любому из значений вектора v. В противном случае BW(r, c) присваивается значение 0.

Задание 5. Выбрать цветовые области из изображения файла chip.bmp, задавая индексы с помощью гистограммы.

```

>> [x,map]=imread('C:\Image\chip.bmp');
>> figure,imhist(x,map),title('histogram')
>> figure,imshow(x,map),
>> bw=roicolor(x,9,12);
>> figure,imshow(bw),title('9 - 12')
>> x1=immultiply(bw,x);
>> figure,imshow(x1,map),title('9 - 12')

```

```

>> bw=roicolor(x,3,8);
>> figure,imshow(bw),title('3 - 8')
>> x1=immultiply(bw,x);
>> figure,imshow(x1,map),title('3 - 8')

```

Яркостный срез

Этот метод помогает выделить определенный диапазон яркости:

$$\forall A_{i,j} \quad B_{i,j} = \begin{cases} 0 & A_B < A_{i,j} < A_H \\ A_{i,j} & A_H \leq A_{i,j} \leq A_B \end{cases} \quad \text{или}$$

$$B_{i,j} = \begin{cases} 0 & A_B < A_{i,j} < A_H \\ K & A_H \leq A_{i,j} \leq A_B \end{cases}$$

где A_H – нижнее граничное значение определяемого диапазона яркости; A_B – верхнее граничное значение определяемого диапазона яркости.

Для выполнения яркостного среза можно использовать функцию `impixel`.

Синтаксис

`R = impixel(S,c,r)`

Функция `impixel` возвращает значения красной, зеленой и синей составляющих цвета для заданных координат – c и r – векторов значений столбцов и строк.

Задание 6. Выполнить яркостный срез полноцветного изображения файла `cotton3.bmp`.

```

[img] = imread('c:\Image\cotton3.bmp');
[m,n,k] = size(img)
img = im2double(img);
R = zeros(m,n,3);
z = [0.1,0.8;0.1,0.8; 0.1,0.9];
for y = 1:m
for x = 1:n
b = impixel(img,x,y);
if ((b(1)>= z(1,1))&(b(1)<= z(1,2)))& ((b(2)>= z(2,1))&(b(2)<=
z(2,2))) ... & ((b(3)>= z(3,1))&(b(3)<= z(3,2)))
R(x,y,1) = b(1); R(x,y,2) = b(2);
R(x,y,3) = b(3); else

```

```
R(x,y,1) = 0; R(x,y,2) = 0; R(x,y,3) = 0;  
end  
end  
end  
figure, imshow(img)  
figure, imshow(R)
```

5.3 Задания для самостоятельного решения

1. Выполнить сегментацию изображений файла Clouds.bmp методом разделения.
2. Выполнить сегментацию изображений файла Construc.bmp методом выращивания областей.
3. Выбрать цветовые области из изображения файла bike.bmp, задавая индексы с помощью гистограммы.
3. Выполнить яркостный срез полноцветного изображения файла bike.bmp, задавая диапазон r от 0.2 до 0.8; g от 0.2 до 0.7; b от 0.1 до 0.7.

5.4 Контрольные вопросы

1. В чем заключается сегментация изображения?
2. Какие признаки используются для сегментации?
3. В чем заключается метод выращивания областей, использующийся для сегментации изображения?
4. В чем заключается метод разделения, использующийся для сегментации изображения?
5. Что является входными параметрами функции сегментации методом разделения?
6. В чем заключается преобразование яркостного среза?
7. Какие параметры возвращает функция `imrixel`?