

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования**
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания по дисциплине

**МЕТОДЫ ИССЛЕДОВАНИЯ И МОДЕЛИРОВАНИЯ ИНФОРМАЦИОННЫХ
ПРОЦЕССОВ И ТЕХНОЛОГИЙ**

Направление подготовки 09.04.02 «Информационные системы и технологии»
Направленность (профиль) «Управление данными»
Квалификация выпускника: магистр

Ставрополь
2026

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА 1 Моделирование процессов и систем в пакете Simulink системы MATLAB	4
ЛАБОРАТОРНАЯ РАБОТА 2 Моделирование объектов, описываемых системами алгебраических уравнений.	17
ЛАБОРАТОРНАЯ РАБОТА 3 Методы определения аппроксимирующего спектра сигнала.....	24
ЛАБОРАТОРНАЯ РАБОТА 4 Исследование поведения динамической системы.....	48
ЛАБОРАТОРНАЯ РАБОТА 5 Разработка структур виртуальных моделей для решения оптимизационных задач.....	59
ЛАБОРАТОРНАЯ РАБОТА 6 Модели оптимизационных задач	65
ЛАБОРАТОРНАЯ РАБОТА 7 Разработка структуры квазианалоговых моделей.....	68
ЛАБОРАТОРНАЯ РАБОТА 8 Моделирование систем, описываемых системами линейных алгебраических уравнений или неравенств.....	73
ЛАБОРАТОРНАЯ РАБОТА 9 Построение моделей задач линейного программирования с помощью функциональных блоков Simulink и SimPowerSystem.....	79
ЛАБОРАТОРНАЯ РАБОТА 10 Построение моделей транспортной задачи средствами Simulink.....	83
ЛАБОРАТОРНАЯ РАБОТА 11 Спектральный анализ и синтез сигналов средствами MATLAB.....	92
ЛАБОРАТОРНАЯ РАБОТА 12 Вейвлет-анализ и синтез сигналов в MATLAB	105
ЛАБОРАТОРНАЯ РАБОТА 13 Вейвлет-обработка и компрессия сигналов в MATLAB	117

ЛАБОРАТОРНАЯ РАБОТА №14 Моделирование случайного процесса на основе реализации случайного вектора с заданной ковариационной матрицей.....	131
ЛАБОРАТОРНАЯ РАБОТА №15 Моделирование случайного процесса на основе алгоритма скользящей суммы	138
ЛАБОРАТОРНАЯ РАБОТА №16 Моделирование случайного процесса на основе алгоритма авторегрессии скользящей суммы.....	146
ЛАБОРАТОРНАЯ РАБОТА №17 Компьютерное моделирование информационных процессов	152
ЛАБОРАТОРНАЯ РАБОТА №18 Компьютерное моделирование скалярных случайных полей	159
ЛАБОРАТОРНАЯ РАБОТА 19 Формирование цифровых латентных изображений.....	178
ЛАБОРАТОРНАЯ РАБОТА 20 Методы выявления скрытой информации в латентных изображениях.....	186
ЛАБОРАТОРНАЯ РАБОТА 21 Разработка систем компьютерного моделирования динамических процессов в жидких дисперсных магнитных наносистемах.....	193
ЛАБОРАТОРНАЯ РАБОТА 22 Компьютерное моделирование динамики магнитных капель	200
ЛАБОРАТОРНАЯ РАБОТА 23 Разработка систем распознавания изолированного слова	207
ЛАБОРАТОРНАЯ РАБОТА 24 Создание системы распознавания слитной речи	231
ЛИТЕРАТУРА И ИСТОЧНИКИ	258

ЛАБОРАТОРНАЯ РАБОТА 1

Моделирование процессов и систем в пакете Simulink системы

MATLAB. Построение простейших моделей (2 ч)

Цель и содержание: изучить основные приемы работы с системой компьютерной математики MATLAB приобрести навыки построения моделей систем в пакете Simulink

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: построение простейших моделей в пакете Simulink системы MATLAB.

Теоретическое обоснование

Система компьютерной математики MATLAB относится к инструментальным средствам, которые предназначены для решения задач в различных областях человеческой деятельности (научных, экономических, инженерных, физических и т. д.), позволяющим производить моделирование, разработку и отладку различных систем и устройств.

Для работы с системой MATLAB необходим IBM – совместимый компьютер с процессором Pentium II и выше, операционная система Windows XP и выше.

Эффективность использования среды MATLAB определяется:

- достаточно простым интерфейсом пользователя;
- большим количеством моделей функциональных устройств (в частности, элементов систем связи);
- возможностью создавать свои модели;
- разнообразием видов анализа функциональных устройств и систем.

Система MATLAB является интерактивной системой для выполнения инженерных и научных расчетов, ориентированной на работу с массивами

данных. Система использует математический сопроцессор и допускает возможность обращения к программам, написанным на языках Fortran, C и C++.

Система MATLAB имеет собственный язык программирования, средства для спектрального анализа и синтеза, быстрого преобразования Фурье (БПФ), обработки изображений, Wavelet-анализа, имеет большие возможности для работы с сигналами, для расчета и проектирования систем связи, цифровых и аналоговых фильтров, различных вычислительных систем, моделирования информационных процессов и систем.

При помощи командного окна (рисунок 1.1) можно осуществлять все вычисления в режиме калькулятора. При этом можно осуществлять присвоения различным переменным значений и далее пользоваться ими в командном окне.

Программирование в среде MATLAB осуществляется путем создания М-файлов с расширением *.m (рисунок 1.2).

Пакет Simulink позволяет осуществлять моделирование поведения динамических нелинейных систем. Для начала работы необходимо создать новый файл с расширением *.mdl (Simulink model), в котором с помощью библиотек Simulink (рисунок 1.3) пользователь осуществляет графическую сборку любой системы из отдельных блоков, хранящихся в библиотеках Simulink.

В результате получаем модель исследуемой системы (S-модель).[3]

Рассмотрим наиболее часто встречающиеся блоки Simulink, которые будут необходимы при моделировании информационных процессов и систем.

На рисунке 1.4 представлены источники сигналов и другие элементы, входящие в состав библиотеки Sources.

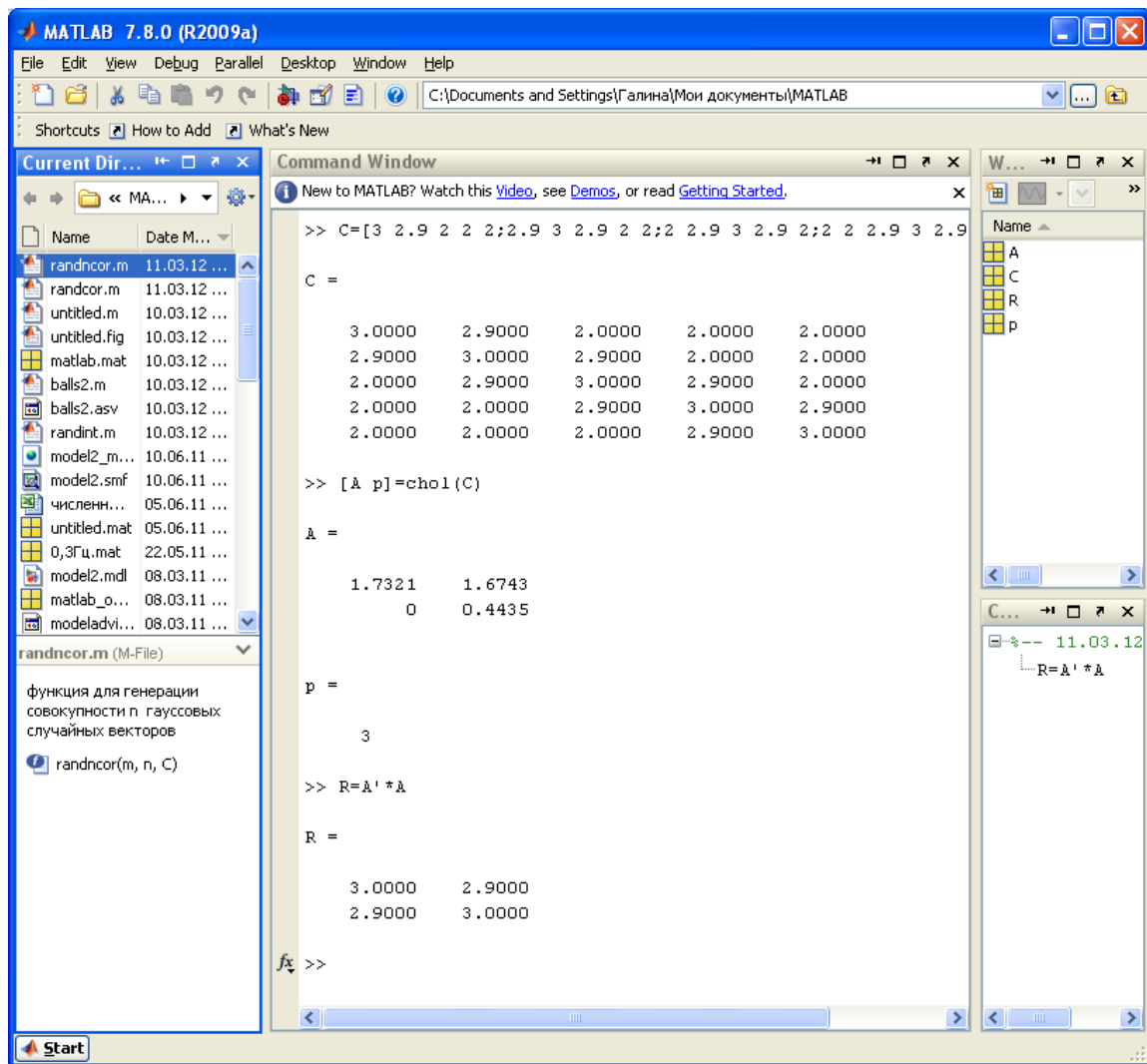


Рисунок 1.1 –Интерфейс системы MATLAB

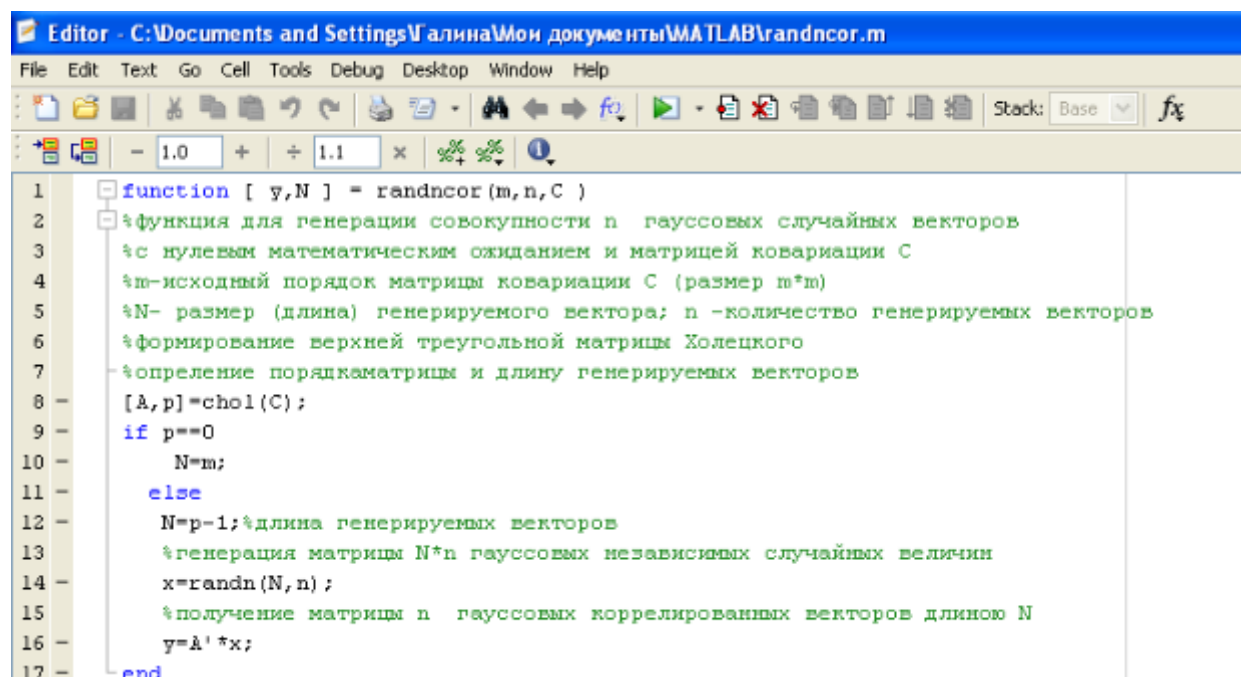


Рисунок 1.2 – Окно М-файла

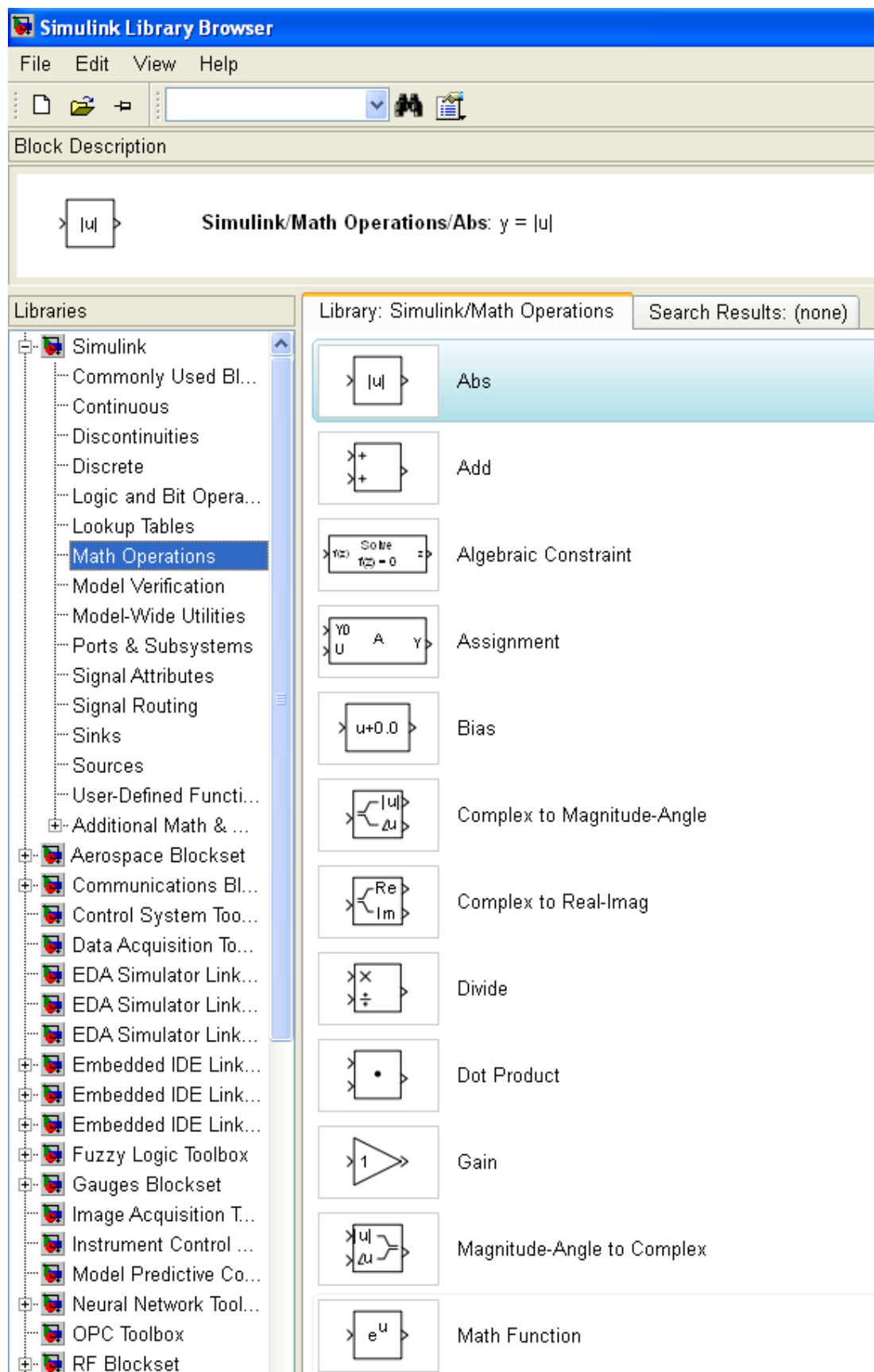


Рисунок 1.3 – Библиотеки Simulink. Блоки математических операций

Рассмотрим некоторые блоки математических операций (рисунок 1.3):

- **Abs** предназначен для вычисления абсолютного значения входного сигнала.
- **Add** предназначен для вычисления алгебраической суммы двух (или более) сигналов.
- **Gain** умножает входной сигнал на заданный коэффициент.
- **Divide** перемножает или делит два (или более) сигналов.
- **MathFunction** преобразует входной сигнал по заданной из списка функции (например, возведение в квадрат, корень квадратный, логарифм и т.д.).

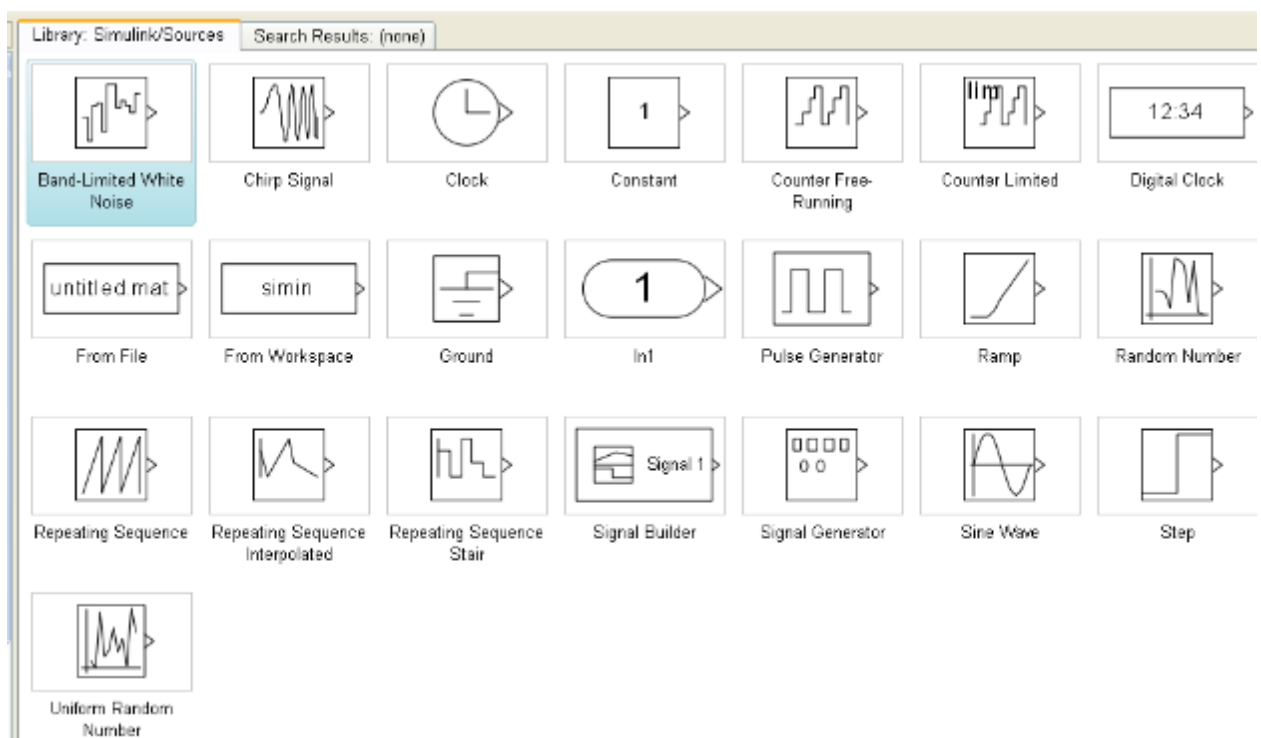


Рисунок 1.4 – Блоки библиотеки Simulink/Sources

Рассмотрим часто встречающиеся блоки (рисунок 1.4):

- **From Workspace** – блок считывания данных из рабочей области;
- **From File** – блок считывания данных из файла;
- **Sine Wave** формирует синусоидальный сигнал с заданной частотой, амплитудой, фазой и смещением;

– **Signal Generator** формирует один из четырех видов периодических сигналов (синусоидальный, прямоугольный, пилообразный, случайный сигналы);

– **Ramp** формирует линейный сигнал;

– **Step** формирует ступенчатый сигнал;

– **Random Number** формирует случайный сигнал с нормальным распределением уровня сигнала.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно рассмотрев приведенные в примеры.

Задание 1.1. Построение простейших моделей

Построить модель сигнала вида $x(t) = A\sin(\omega t + \varphi) + t^m$ на интервале $[a; b]$, и результаты моделирования вывести на виртуальный осциллограф. Параметры сигналов приведены в таблице 1.1. Номер варианта соответствует номеру, под которым студент записан в списке группы.

На рисунке 1.5 приведен один из вариантов построения сигнала: $x(t) = 1,5\sin(\pi t + \pi/3) + t^2$ с помощью функциональных блоков. Сигнал

$1,5\sin(\pi t + \pi/3)$ задан в окне диалога параметров блока Sine Wave – Source Block Parameters: Sine Wave (рисунок 1.6).

Таблица 1.1 – Параметры сигнала

№ варианта	Амплитуда, A	Частота, ω	Фаза, φ	Степень, m	Интервал, $[a; b]$
1.	0,5	2π	$\pi/3$	2	$[0; 3]$
2.	2,5	$3\pi/4$	$\pi/6$	3	$[0; 1]$
3.	1,5	$2\pi/3$	$\pi/8$	2	$[0; 5]$
4.	4,5	$\pi/4$	$5\pi/4$	4	$[0; 6]$
5.	3,5	$3\pi/2$	$3\pi/5$	3	$[0; 7]$
6.	8,5	$\pi/8$	$3\pi/4$	1,5	$[0; 4]$
7.	7,5	$5\pi/4$	$2\pi/3$	3,5	$[0; 1,5]$
8.	9,5	$3\pi/5$	$\pi/4$	2	$[0; 2,5]$
9.	0,9	$\pi/3$	$3\pi/2$	4	$[0; 1,8]$
10.	0,45	$\pi/6$	$\pi/12$	3	$[0; 3,2]$

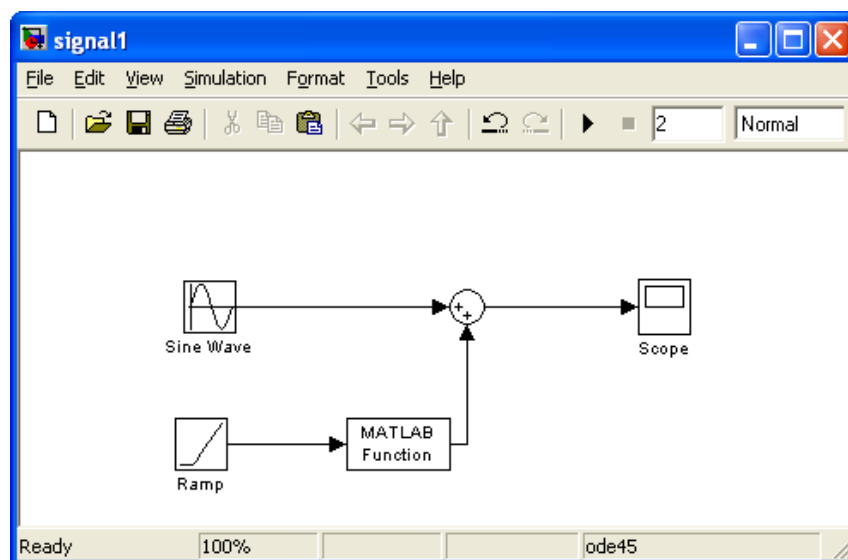


Рисунок 1.5 – Структурная схема модели сигнала $x(t) = 1,5\sin(\pi t + \pi/3) + t^2$

Для сигнала t^2 использовались два блока – блок линейного сигнала и блок математической функции, для которого в соответствующем окне была введена функция в степени 2 (рисунок 1.7).

Интервал моделирования задан в пределах от 0 до 2 в диалоговом окне Configuration Parameters меню Simulation (рисунок 1.8).

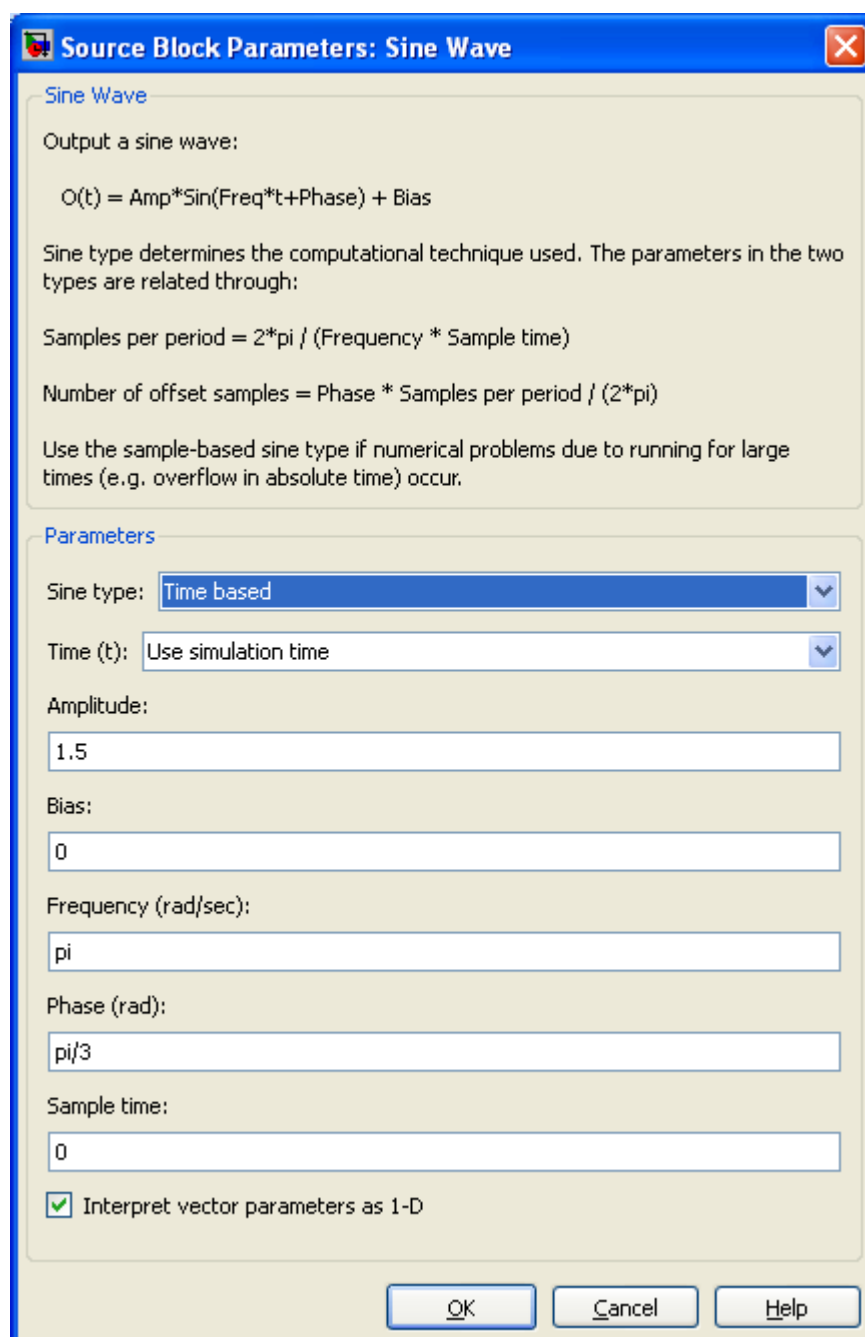


Рисунок 1.6– Окно диалога параметров блока Sine Wave с заданными параметрами сигнала

Результаты моделирования работы выведены на экран виртуального осциллографа (рисунок 1.9).

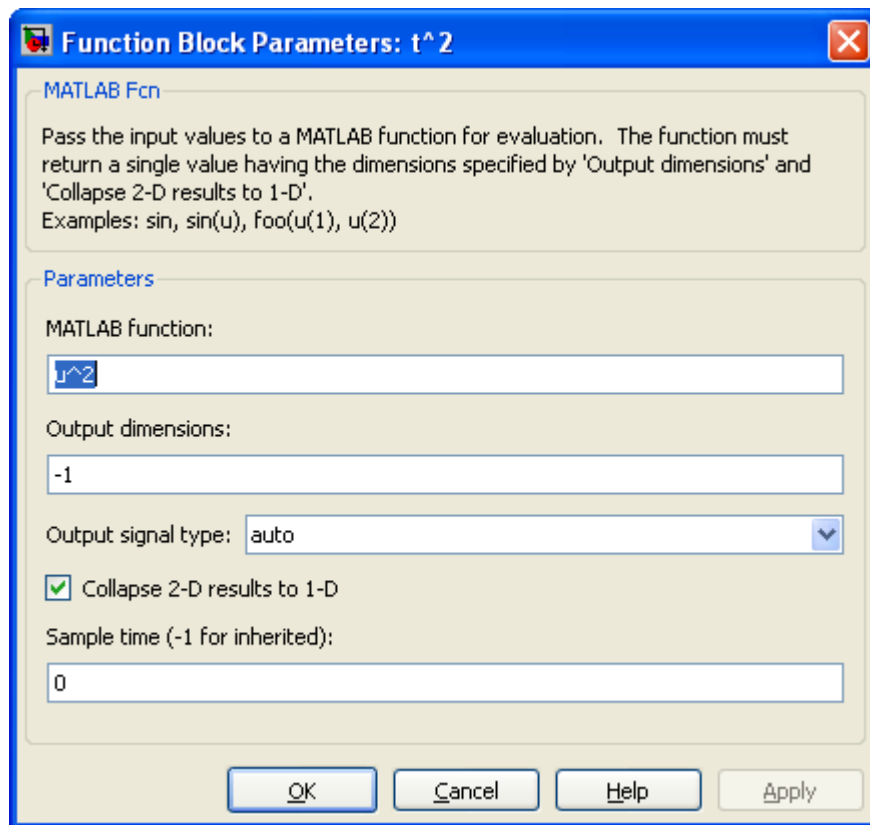


Рисунок 1.7. Параметры функции t^2

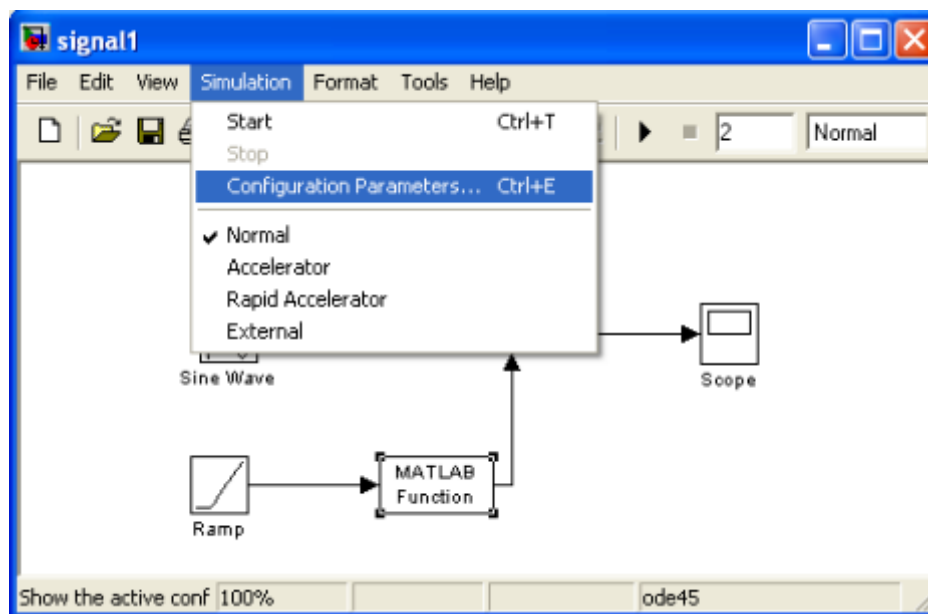


Рисунок 1.8 – Вызов окна диалога Configuration Parameters меню Simulation

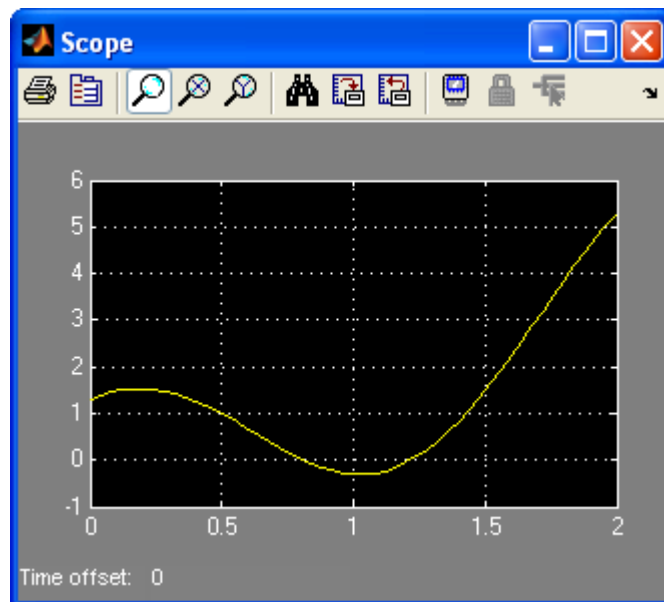


Рисунок 1.9 –График заданного сигнала на экране виртуального осциллографа

Задание 1.2. Моделирование сигнала заданной формы

1. $x(t) = (2 + \sin(2t))^2$.
2. $x(t) = (2 + 0,5\sin(2t))^{1/2}$.
3. $x(t) = te^{-t}\cos(2\pi t + \pi/4)$
4. $x(t) = 1.5 + 0.7t - 0.05t^2 - (e^{-3t} + t^3)^{1/3}$

Задание 1.3. Моделирование интегрированного сигнала

В системах связи помехи и замирания, воздействующие на сигнал при его прохождении по каналу связи, имеют статистический характер и могут быть описаны при помощи различных законов распределений. В частности, замирания в канале связи при отсутствии прямой видимости между абонентом и базовой станцией имеют рэлеевский закон распределения; аддитивные помехи (шумы) часто описываются нормальным (гауссовским) законом распределения; временные интервалы между вызовами в сетях связи обычно имеют экспоненциальный закон распределения и т.д.

Построить модель для имитации смеси сигналов:

- синусоидального сигнала и белого шума;
- синусоидального сигнала и гауссова шума (с математическим ожиданием $m=0$, и дисперсией $\sigma^2=0.01$);
- синусоидального сигнала и шума с равномерным распределением.

Расположение источников сигналов в пакете Simulink:

- 1) Шум с равномерным распределением – Communications Blockset/Comm Sources – рисунок 1.10: Uniform Noise Generator (Noise Lower Bound =0, Noise Upper Bound =1, Seed 0, Sample Time 0.01);
- 2) Гауссов шум – Communications Blockset/Comm Sources – рисунок 1.10: Gaussian Noise Generator (Mean Value $M=0$, Variance $\sigma^2=1$, Initial Seed 0, Sample Time 0.01);
- 3) Белый шум: Simulink/Sources – рисунок 1.4: Band – Limited White Noise;
- 4) Синусоидальный сигнал/Simulink–Sources– рисунок 1.4: Sine Wave (Amplitude 1, Frequency 1, Phase 0, Sample Time 0).

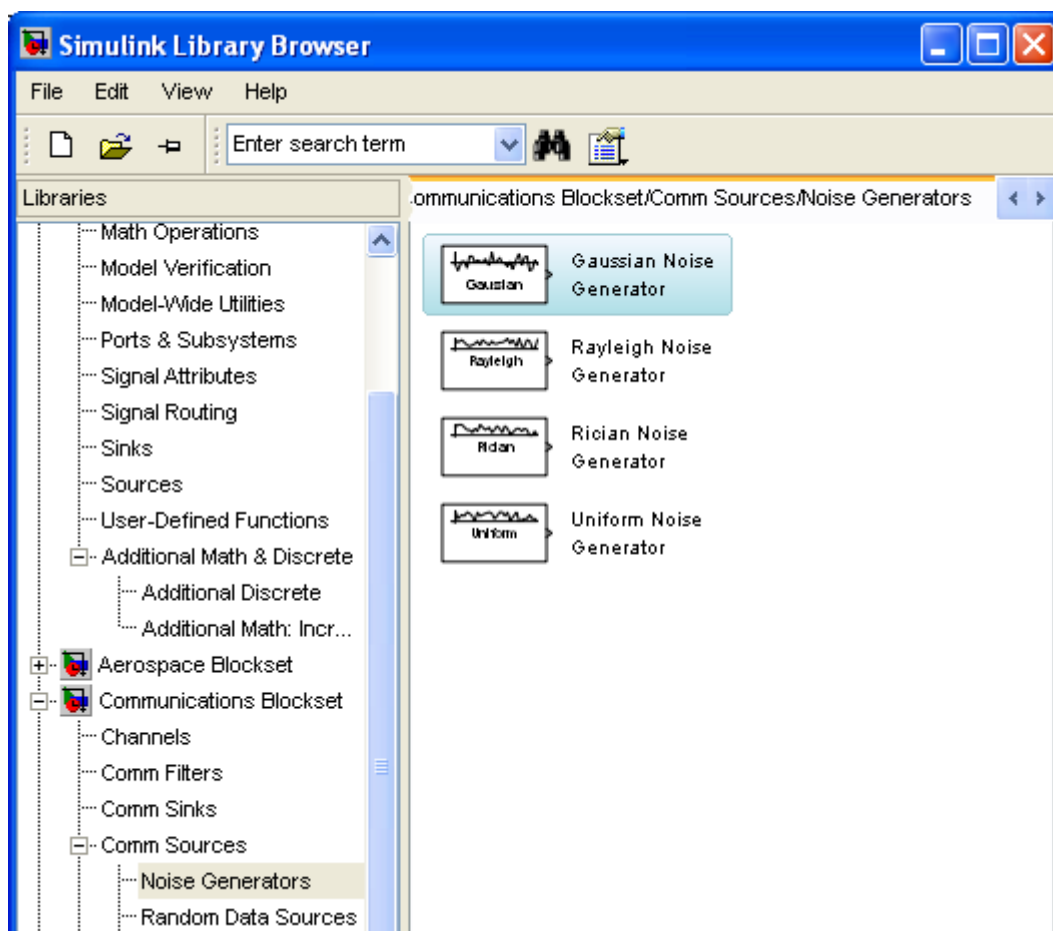


Рисунок 1.10 – Блоки библиотеки Communications Blockset/Comm Sources

Пример построения модели для имитации смеси синусоидального сигнала белого шума и график полученного в результате имитации результирующего сигнала представлены на рисунках 1.11, 1.12.

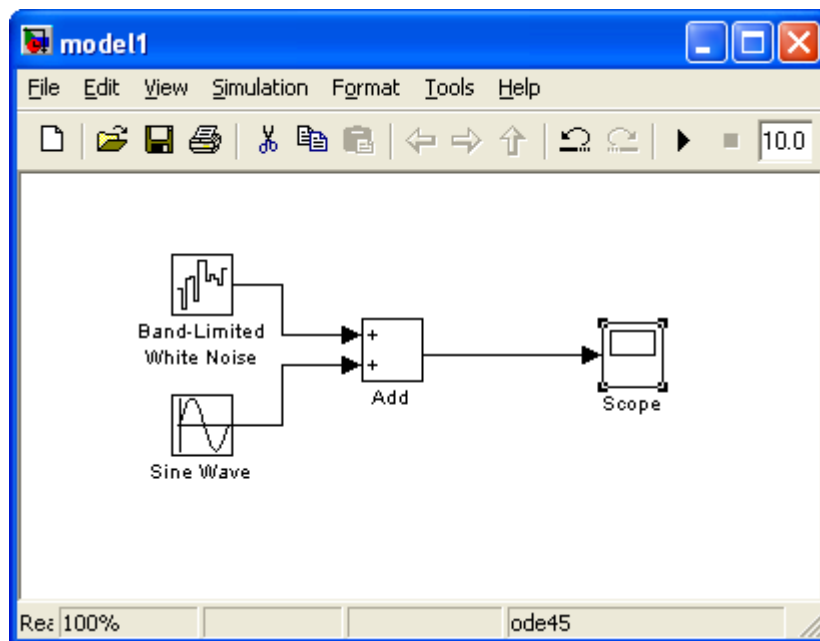


Рисунок 1.11– Модель имитации смеси сигналов синусоидального и белого шума

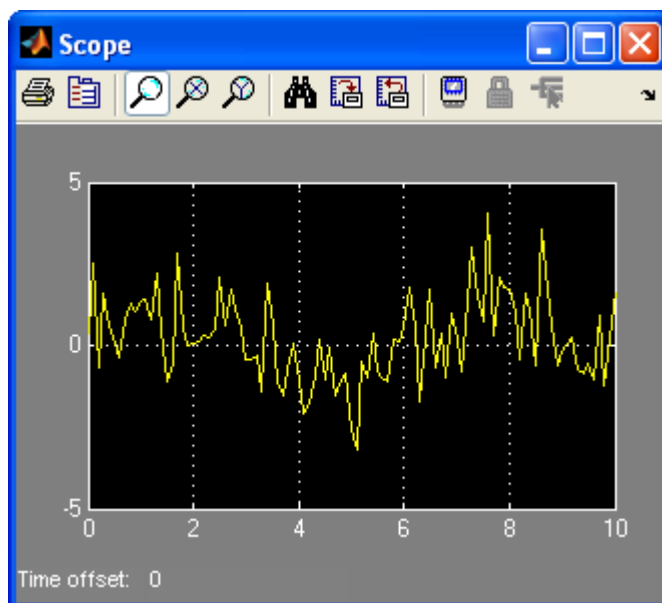


Рисунок 1.12 –Результат имитации смеси сигналов синусоидального и белого шума

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Для чего используется система MATLAB?
2. Как осуществляется программирование в среде MATLAB?
3. Для чего используется пакет Simulink?
4. Как с помощью Simulink можно моделировать поведение сложных систем?
5. На какой технологии основана разработка моделей средствами SIMULINK (S-модели)?
6. Для начала работы необходимо создать новый файл с расширением *.mdl (Simulink model), в котором с помощью библиотек Simulink (рисунок 1.3) пользователь осуществляет графическую сборку любой системы из отдельных блоков, хранящихся в библиотеках Simulink.

ЛАБОРАТОРНАЯ РАБОТА 2

Моделирование объектов, описываемых системами алгебраических уравнений (2 ч)

Цель и содержание: использование классических методов аналоговой вычислительной техники для моделирования объектов, описываемых системами алгебраических уравнений, приобретение навыков построения таких моделей в системе компьютерной математики MATLAB.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: построение моделей объектов, описываемых системами алгебраических уравнений.

Теоретическое обоснование

Моделирование объектов, описываемых системами алгебраических уравнений, можно осуществлять путем сведения системы алгебраических уравнений к эквивалентной системе дифференциальных уравнений.

Пусть исследуемый объект описывается системой уравнений:

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\&\dots \\a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n\end{aligned}\tag{2.1}$$

Заменяем данную систему алгебраических уравнений эквивалентной системой дифференциальных уравнений:

$$\begin{aligned}\frac{dx_1}{dt} + a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n - b_1 &= 0 \\ \frac{dx_2}{dt} + a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n - b_2 &= 0\end{aligned}\tag{2.2}$$

...

$$\frac{dx_n}{dt} + a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n - b_n = 0$$

Для эквивалентности системы уравнений 2.1 и 2.2 необходимо, чтобы решение системы дифференциальных уравнений 2.2 было затухающим, т.е. как только все производные затухнут ($\frac{dx_i}{dt} = 0$), будет получено решение системы уравнений 2.1: $\{x_1, x_2, \dots, x_n\}$.

Достаточным условием, обеспечивающим затухающее решение, является положительная определенность матрицы коэффициентов линейной системы уравнений. Это возможно, в частности, при условии, когда

$$a_{ij} \geq \sum_{j=1}^n a_{ij}, i \neq j.$$

Пример 2.1. Найти решение системы линейных алгебраических уравнений:

$$\begin{cases} 4x_1 + 2x_2 = 14 \\ 2x_1 + 5x_2 = -5 \end{cases} \quad (2.3)$$

Перейдем к эквивалентной системе дифференциальных уравнений:

$$\begin{cases} \frac{dx_1}{dt} = 14 - 4x_1 - 2x_2 \\ \frac{dx_2}{dt} = 5 - 2x_1 - 5x_2 \end{cases} \quad (2.4)$$

Структурная схема модели данной системы приведена на рисунке 2.1. Она построена с использованием классических методов аналоговой вычислительной техники. Переходный процесс установления решения изображен на экране виртуального осциллографа (рисунок 2.2).

На рисунке 2.2 видно, что после $t = 2$ на выходах виртуальных интеграторов устанавливаются сигналы, соответствующие решению системы линейных алгебраических уравнений:

$$x_1 = 5,$$

$$x_2 = -3.$$

Проверьте решение системы уравнений 2.3 с помощью Excel (рисунок 2.3).

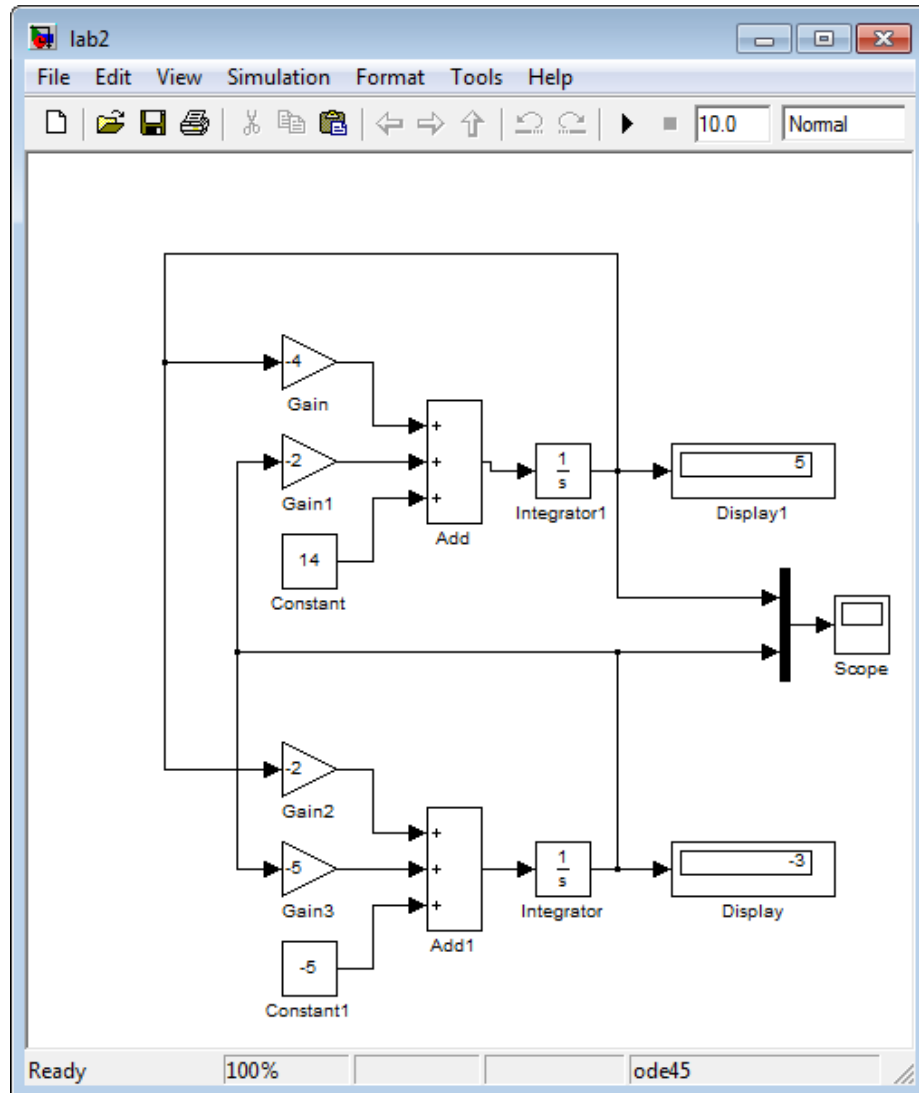


Рисунок 2.1 – Структурная схема модели системы дифференциальных уравнений 2.2

проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 2.1. Построить модель решения задачи. Отряд туристов вышел в поход на нескольких байдарках L , часть из которых A -местные, а часть – B -местные. Сколько A -местных и сколько B -местных байдарок было в походе, если отряд состоит из M человек?

Переходный процесс установления решения выведите на экран виртуального осциллографа. Параметры сигналов приведены в таблице 2.1. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Таблица 2.1– Параметры задачи

№ вар-та	A	B	L	M
1.	1	2	12	20
2.	1	3	7	17
3.	1	4	5	11
4.	1	5	8	16
5.	2	3	8	19
6.	2	4	10	37
7.	2	5	8	16
8.	3	4	8	26
9.	3	5	6	24
10.	4	5	7	33

Задание 2.2. Построить модель системы линейных алгебраических уравнений вида:

$$\begin{aligned} Ax_1 + Bx_2 + Cx_3 &= D \\ Lx_1 + Mx_2 + Nx_3 &= O \\ Qx_1 + Rx_2 + Sx_3 &= T \end{aligned}$$

Переходный процесс установления решения выведите на экран виртуального осциллографа. Параметры сигналов приведены в таблице 2.2. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Таблица 2.2 – Параметры сигнала

№ вар-та	A	B	C	D	L	M	N	O	Q	R	S	T
1.	5	2	1	2.5	3	7	2	-1.5	4	0.5	5	11.5
2.	2	1.5	3.5	-16.25	2	3	4	-19.5	2	-0.5	5	-22.75
3.	0.5	-8	-9	-30.5	4	5	6	25.5	-1.5	2	3	8.5
4.	-1	2.5	9	-35.25	5	6	7	-22	1	2	-2.5	10
5.	3	-2	-7	-8.5	6	7	8	38	1	-3.5	6	32
6.	4.5	5.5	4	-3.25	7	8	9	-12	-4.5	5	1	38
7.	6.5	-5	-6	39	1	2	3	10.5	4	1	-5.5	7.25
8.	5	7.5	9.5	27.75	2	3	4	-4.5	1	-6.5	9	11.75
9.	-4	10	-3	-67.5	3	4	5	15.5	-7.5	8	1	-90.5
10.	8	1	6	64.5	4	5	6	22.5	7	1	-8.5	70

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Как с помощью Simulink можно моделировать объекты, описываемые системами алгебраических уравнений?
2. Какие основные математические блоки используются для моделирования объектов, описываемых системами алгебраических уравнений?
3. Что является достаточным условием, обеспечивающим затухающее решение системы дифференциальных уравнений?

ЛАБОРАТОРНАЯ РАБОТА 3

Методы определения аппроксимирующего спектра сигнала(2 ч)

Цель и содержание: изучить методику определения аппроксимирующего спектра сигнала, приобрести навыки построения таких моделей в системе компьютерной математики MATLAB.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: задача определения аппроксимирующего спектра методом наименьших квадратов; построение структурной схемы виртуального генератора системы базисных функций; метод равных площадей; блочно-импульсная аппроксимация.

Теоретическое обоснование

Пусть непрерывный сигнал $x(t)$ задан на некотором интервале изменения аргумента $[0, T]$. Выбирается некоторая система функций $\bar{S} = \{s_k(t)\}_{k=0}^n$, заданных на том же интервале изменения аргумента. Из теории аппроксимации известно, что сигнал может быть представлен на определенном интервале изменения аргумента обобщенным полиномом

$$x_a = \sum_{k=0}^n X(k)s_k(t). \quad (3.1)$$

Систему коэффициентов $X(k)$ обычно называют аппроксимирующим полиномиальным спектром (АПС) сигнала $x(t)$.

Метод наименьших квадратов

Задача определения аппроксимирующего спектра методом наименьших квадратов формулируется следующим образом: определить систему величин $X(k)$, минимизирующую норму $\|\beta(t)\|$ функции ошибки $\beta(t) = x_a(t) - x(t)$.

В качестве нормы ошибки будем использовать условие

$$\frac{1}{T} \int_0^T r(t) \beta^2(t) dt = \min, \quad (3.2)$$

где $r(t)$ – весовая функция.

Вектор АПС $\bar{X} = \{X(k)\}_{k=0}^n$ сигнала находится по формуле:

$$\bar{X} = W^{-1} \cdot \bar{Q}, \quad (3.3)$$

где элементы операционных матриц W и вектора $\bar{Q}$ вычисляются следующим образом:

$$\begin{aligned} W_{ij} &= \int_0^T r(t) S_{i-1}(t) S_{j-1}(t) dt, \\ q_i &= \int_0^T r(t) S_{i-1}(t) x(t) dt, \\ i, j &= 1, \dots, n+1. \end{aligned} \quad (3.4)$$

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Научиться использовать метод наименьших квадратов, выполнив приведенные примеры.

Пример 3.1. Необходимо аппроксимировать методом наименьших квадратов сигнал $x(t) = \cos \pi t^2$ на интервале $[0, 1]$. В качестве базисных функций взять степенные функции 5-го порядка.

Структурные схемы определения коэффициентов по формулам 3.3–3.4 представлены на рисунках 3.1 и 3.2.

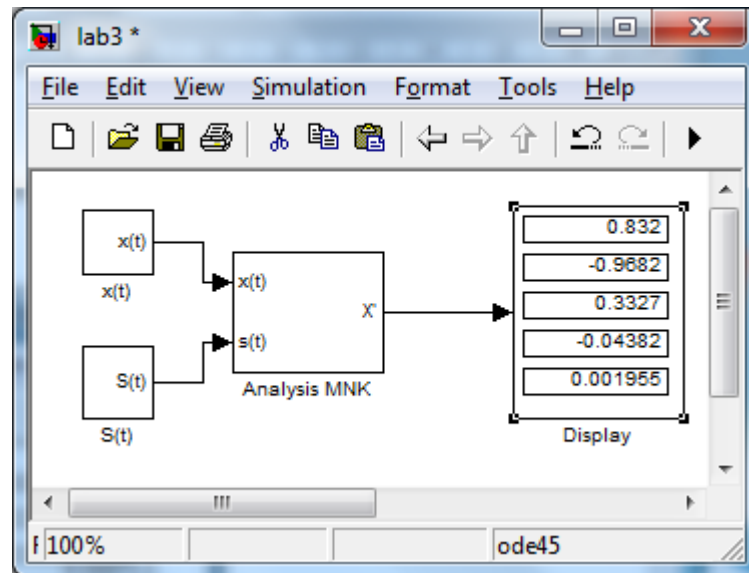


Рисунок 3.1. Структурная схема определения коэффициентов аппроксимирующего полинома

Рассмотрим подробнее блоки, задающие сигнал и базисные функции. На рисунке 3.2 показан блок $X(t)$.

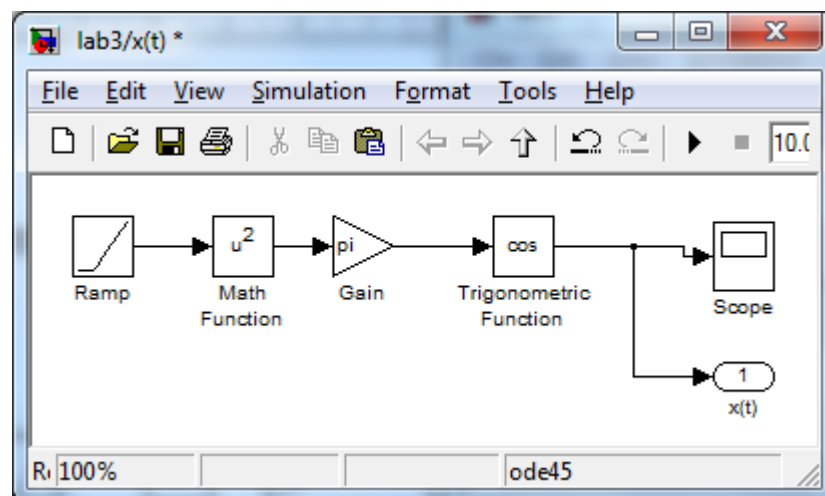


Рисунок 3.2. Блок, генерирующий аппроксимируемый сигнал

Источником является генератор линейно возрастающего сигнала Ramp, который в данном случае ассоциируется со временем t .

Далее, на рисунке 3.2 видно, как с помощью блоков математических операций (возведение в степень, умножение на константу, преобразование входного сигнала с помощью тригонометричной функции), на выходе получаем заданный сигнал $x(t)$. Для наглядности перед выходом подключен «виртуальный осциллограф» (Scope), на котором в процессе моделирования отображается сигнал $x(t) = \cos \pi t^2$. Общий вид окна блока Scope с его панелью инструментов изображен на рисунке 3.3

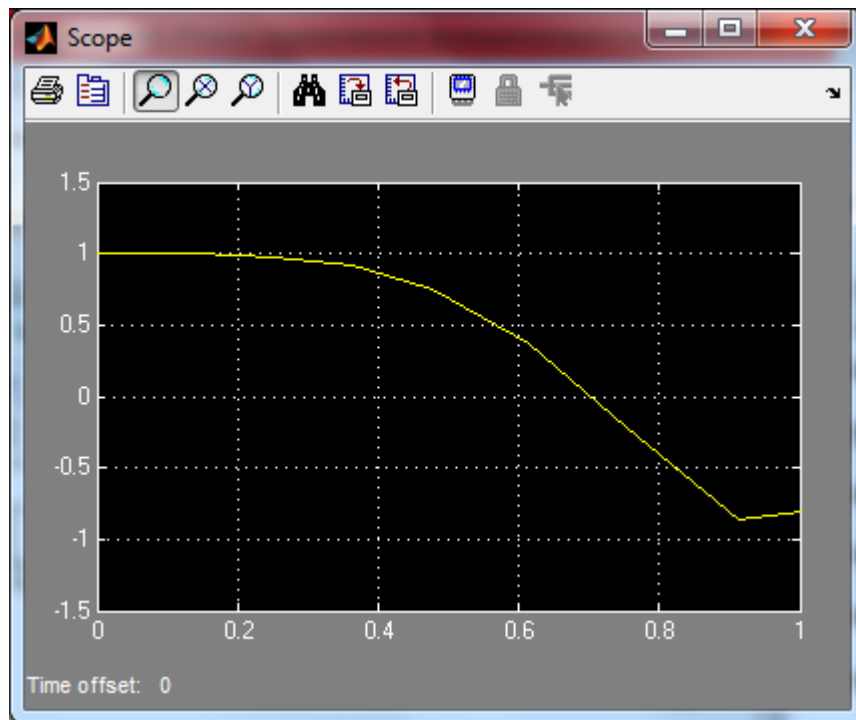


Рисунок 3.3. Панель инструментов виртуального осциллографа и график сигнала, выведенный на осциллограф

Блок $S(t)$, представленный на рисунке 3.1, генерирует систему базисных функций $s_k(t) = \{t^k\}_{k=0}^4$, его структурная схема показана на рисунке 3.4.

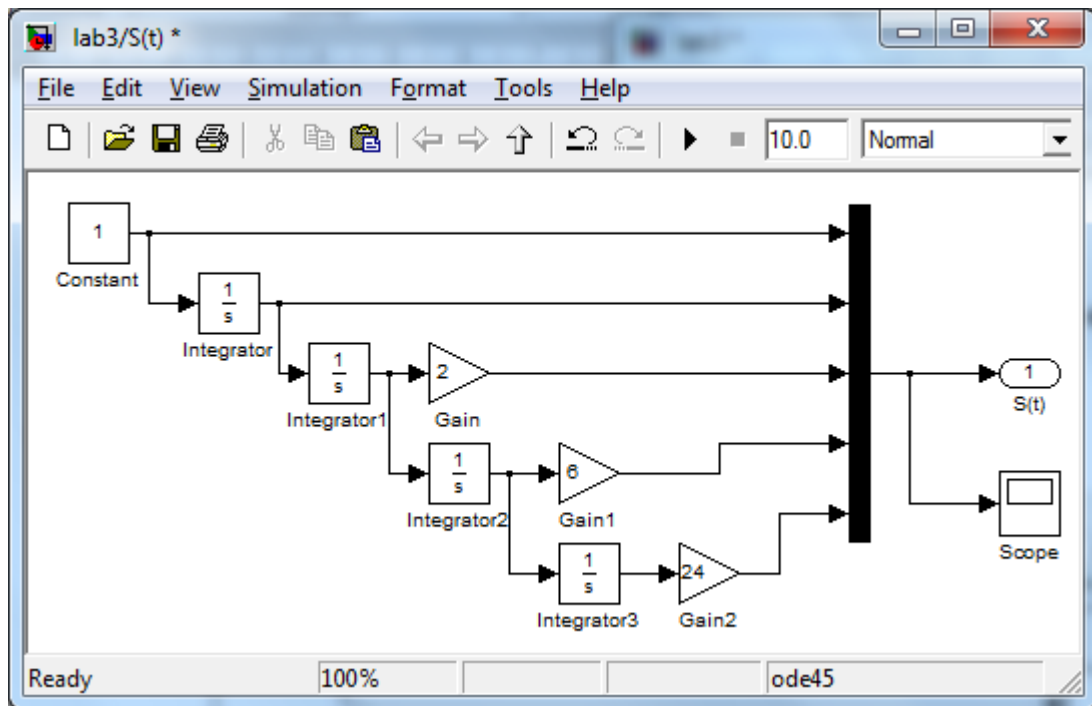


Рисунок 3.4. Структурная схема виртуального генератора системы базисных функций

Поскольку система базисных функций имеет удобный для моделирования вид, можно из первой функции $s(t) = 1$ с помощью интегрирования (блок Integrator) получить следующие. Блоки умножения на константу Gain, Gain1, Gain2 необходимы для нормирования степенных функций.

Полученные функции выводятся на выход этой подсистемы и на виртуальный осциллограф. Изображения базисных функций показаны на рисунке 3.5.

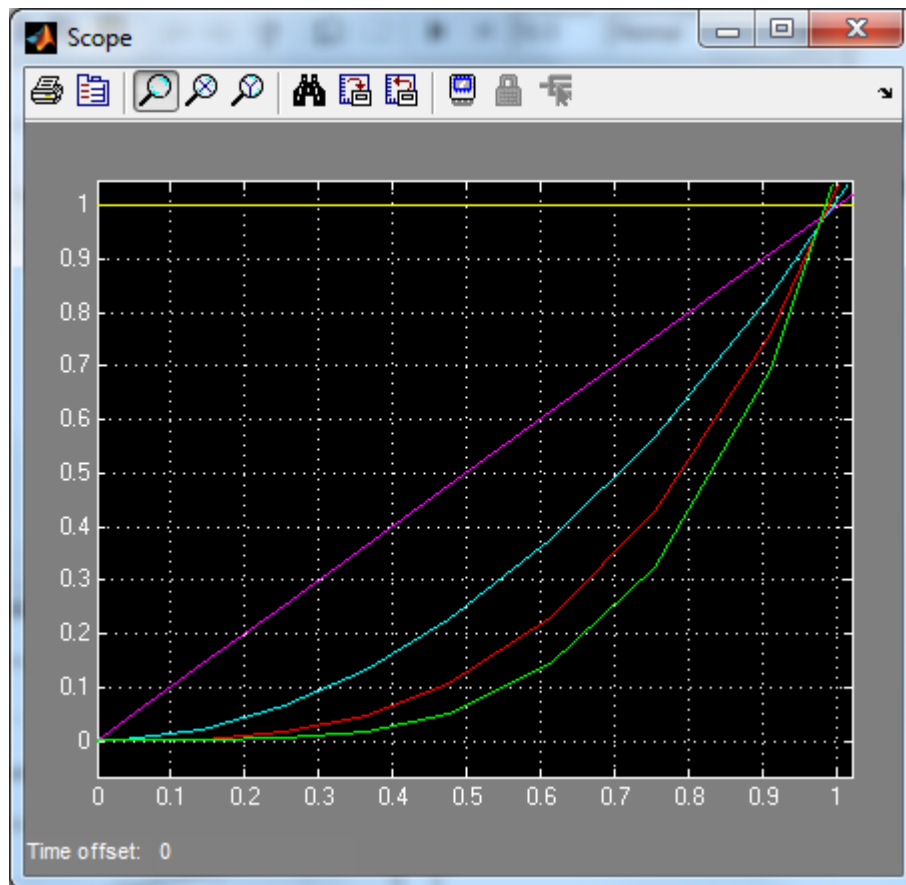


Рисунок 3.5. Система базисных функций

Основу схемы составляет подсхема Analysis MNK, которая в развернутом виде приведена на рисунке 3.6.

Виртуальные индикаторы Score и Display, показанные на этой подсхеме, не являются обязательными для определения коэффициентов аппроксимирующего полинома, а предназначены для контроля правильности построения структурной схемы в процессе отладки виртуальной модели.

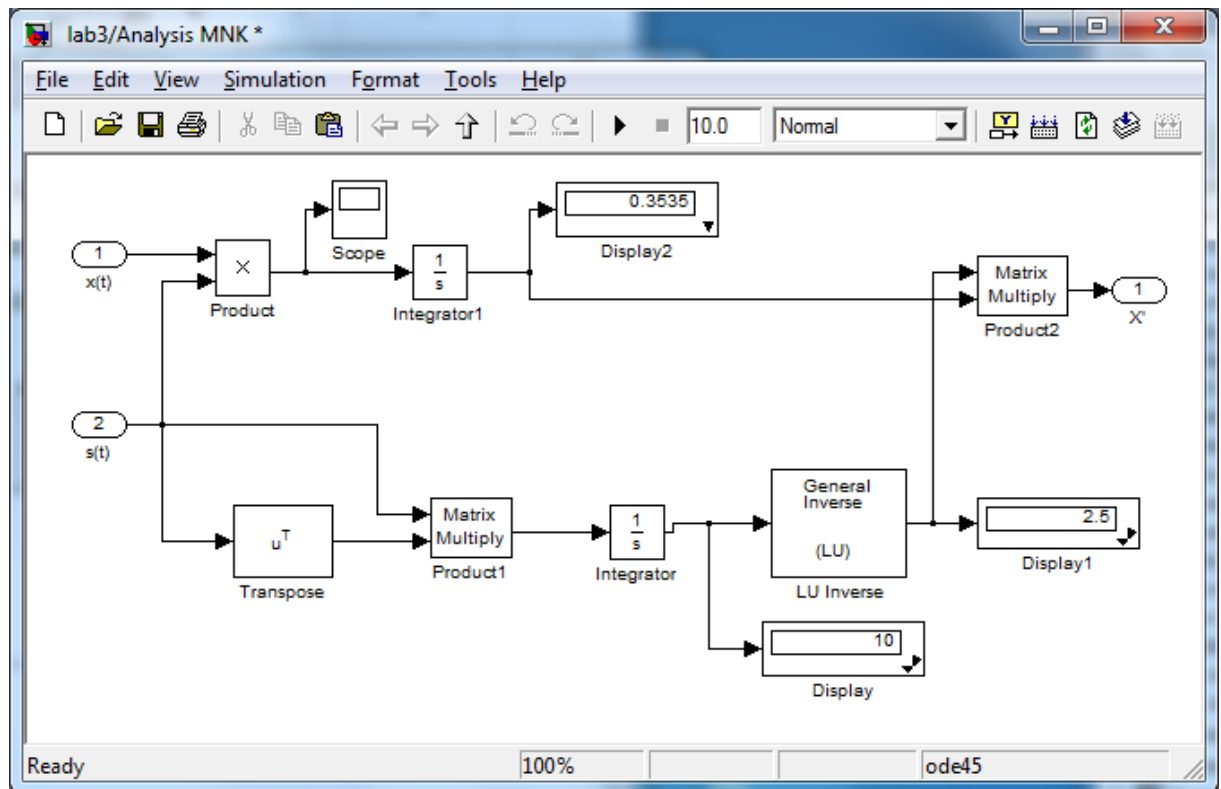


Рисунок 3.6. Расчет коэффициентов по методу наименьших квадратов

Дополним схему расчета коэффициентов блоками для восстановления сигнала. Схема примет следующий вид:

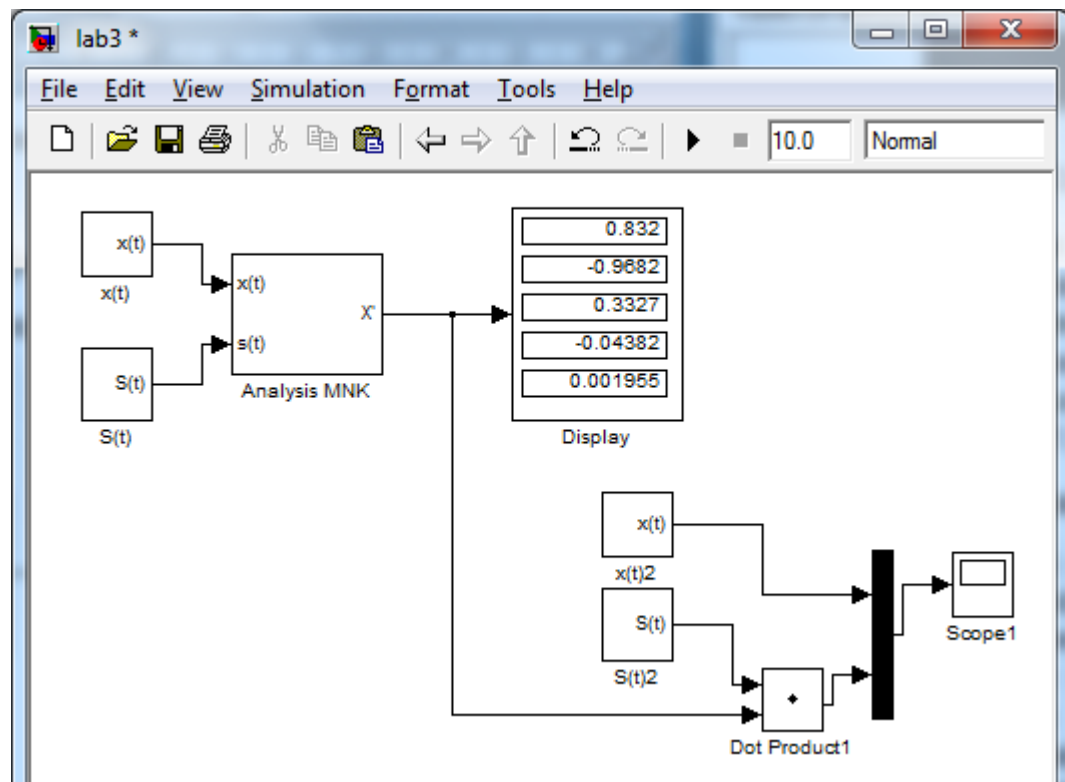


Рисунок 3.7 – Восстановление сигнала по найденному спектру

Результаты работы модели выводятся на экран виртуального осциллографа, и изображены на рисунке 3.8.

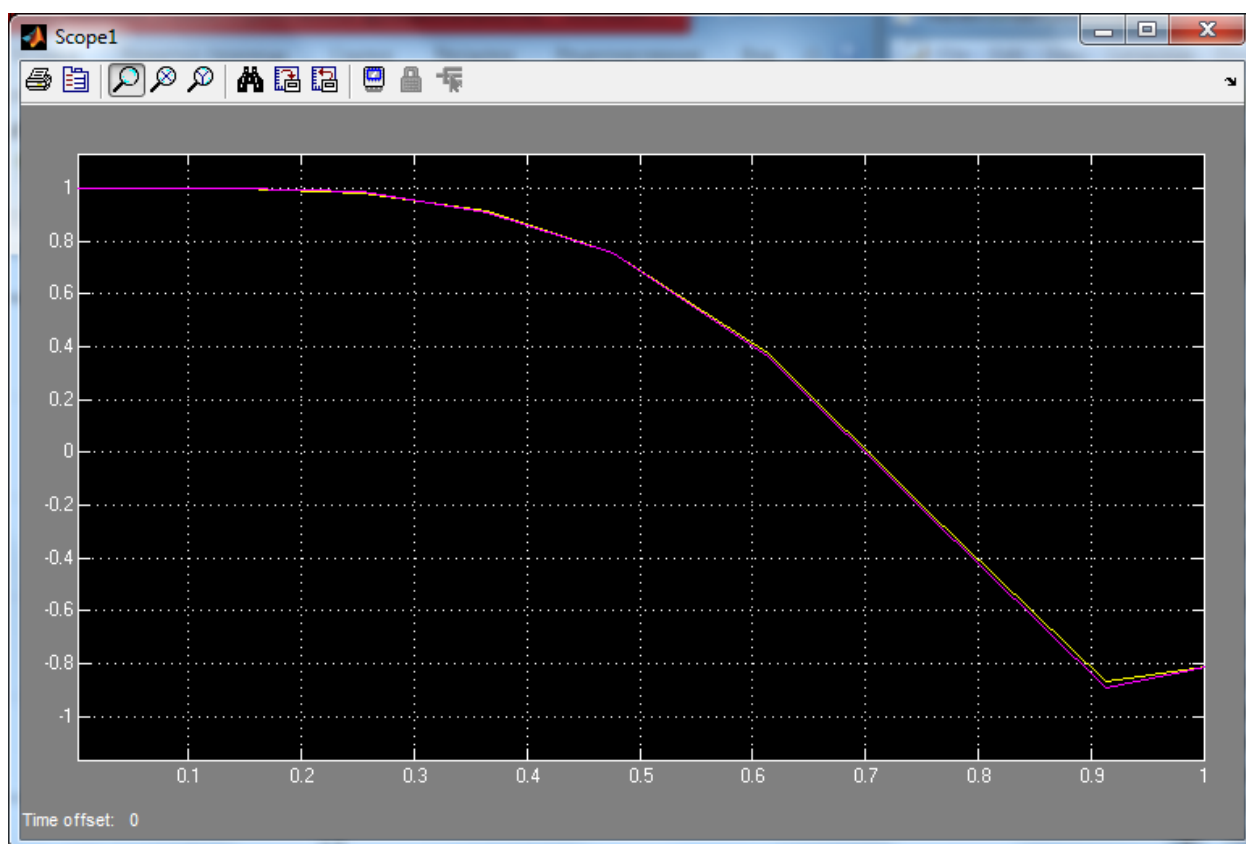


Рисунок 3.8. Исходный сигнал $x(t) = \cos \pi t^2$ и его аппроксимация по методу наименьших квадратов

Модифицированный метод равных площадей

Метод равных площадей является более простым методом аппроксимации, основанный на определении коэффициентов аппроксимирующего полинома, исходя из условия равенства нулю интегралов функций ошибки аппроксимации на некоторой системе интервалов изменения аргумента.

Суть модифицированного метода равных площадей заключается в следующем. Интервал изменения аргумента t разбивается на $(n+1)$ не обязательно равных частей (подинтервалов). На рисунке 3.9 представлены возможные типы систем подинтервалов. Уравнение аппроксимирующей

кривой подбирается так, чтобы площади, ограничиваемые аппроксимируемой и аппроксимирующей функциями, были равны между собой для каждого участка в отдельности. Это приводит к системе линейных алгебраических уравнений, решение которой определяет аппроксимирующий спектр.

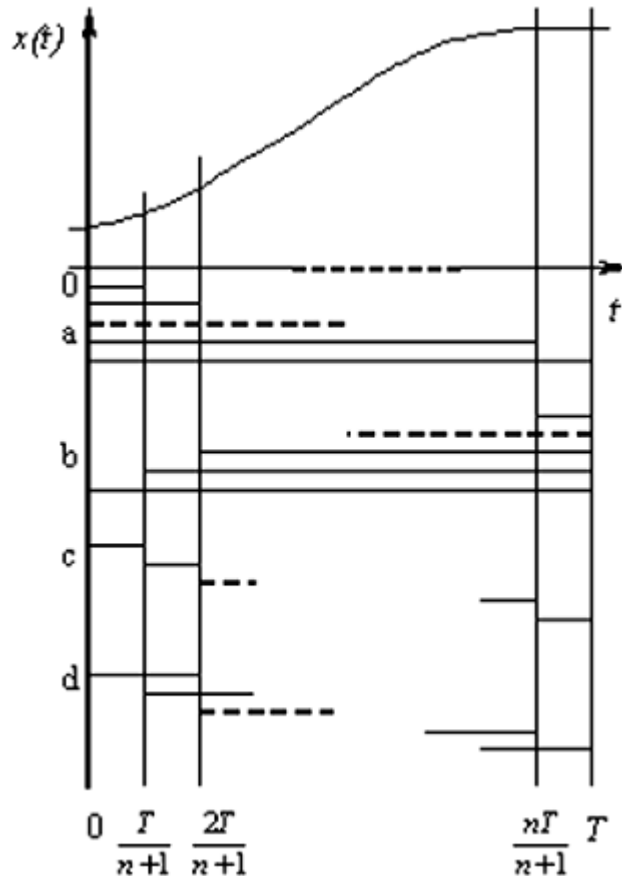


Рисунок 3.9. Системы подинтервалов для метода равных площадей

Условия равенства указанных площадей (интегралов) для некоторой системы линейно-независимых подинтервалов имеет вид:

$$\int_{t_m}^{t_l} r(t) \left[\sum_{k=0}^n X(k) s_k(t) - x(t) \right] dt = 0. \quad (3.5)$$

Например, при выборе системы подинтервалов вида $\left[0, \frac{iT}{n+1}\right], i = 1, \dots, n+1$, матрица $\mathbf{W}$ и вектор $\bar{\mathbf{Q}}$ вычисляются по формулам:

$$W_{ij} = \int_0^{\frac{iT}{n+1}} r(t)s_{j-1}(t)dt,$$

$$q_i = \int_0^{\frac{iT}{n+1}} r(t)x(t)dt,$$

$$i, j = 1, \dots, n + 1.$$
(3.6)

Пример 3.2. Рассмотрим реализацию модифицированного метода равных площадей в среде MATLAB/Simulink. Модель, описывающая модифицированный метод равных площадей, представлена на рисунке 3.10.

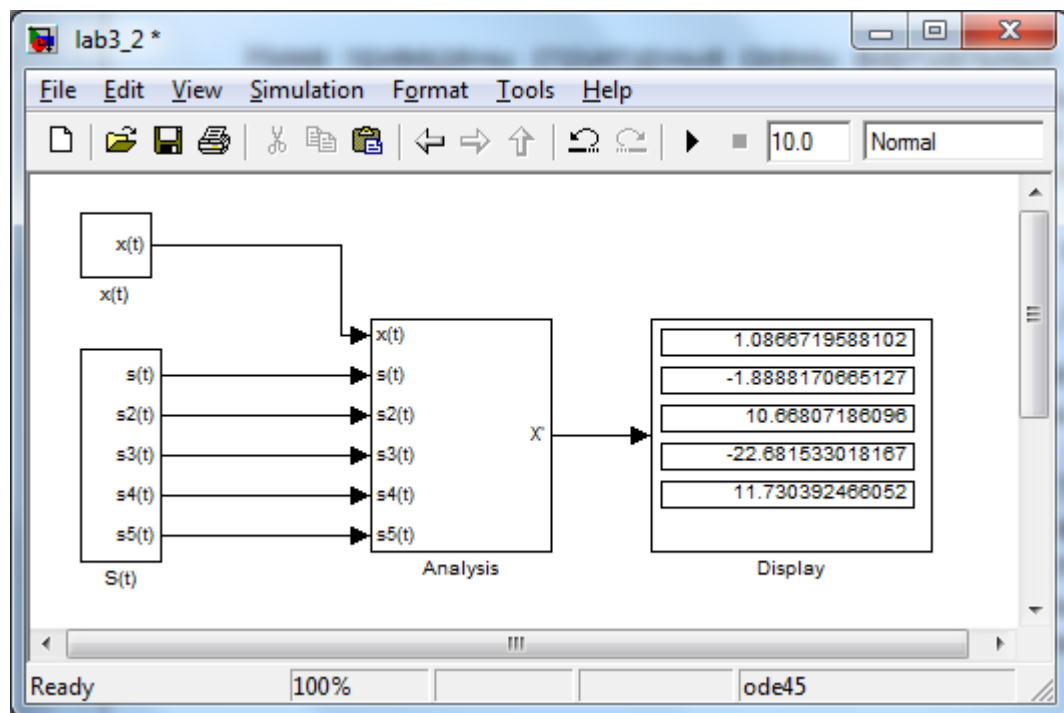


Рисунок 3.10. Структурная схема реализации модифицированного метода равных площадей

Основной частью схемы является блок анализа (Analysis). Его входные параметры – непрерывный сигнал $x(t)$ и система базисных $\bar{\mathbf{S}}(t)$. На выходе находится дисплей, на котором отображается полученный спектр. Рассмотрим пример из предыдущего подраздела. Необходимо определить аппроксимирующий спектр функции на интервале $[0,1]$. Интервал

разбивается на пять подинтервалов $[0; 0,2]$, $[0,2; 0,4]$, $[0,4; 0,6]$, $[0,6; 0,8]$, $[0,8; 1]$. Базисные функции имеют вид $S_k(t) = t^k, k = \overline{0, n}$.

Рассмотрим подробнее блок, задающий базисные функции. Блок S(t) генерирует систему базисных функций $s_k(t) = t^k, k = \overline{0, n}$, его структурная схема показана на рисунке 3.11.

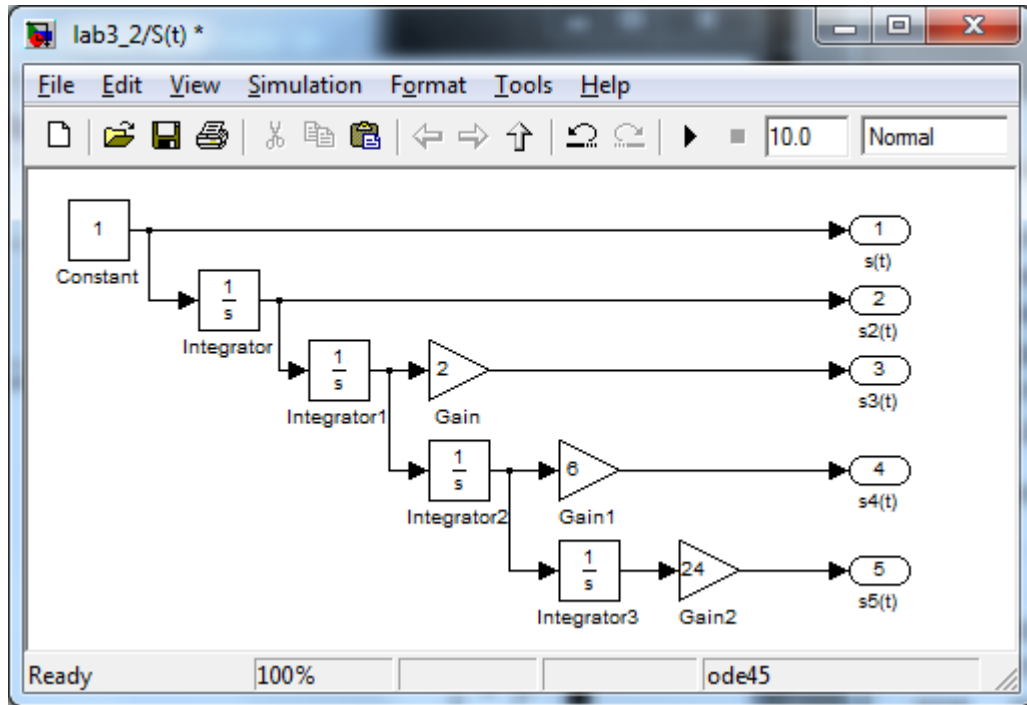


Рисунок 3.11. Структурная схема виртуального генератора системы базисных функций

Поскольку система базисных функций имеет удобный для моделирования вид, можно из первой функции $s(t) = 1$ с помощью интегрирования (блок Integrator) получить следующие. Блоки умножения на константу Gain, Gain1, Gain2 необходимы для нормирования степенных функций.

Перед тем, как перейти к описанию центрального блока модели Analysis, необходимо отметить, что в подсистемах этого блока используется конструкция, которую можно использовать как при расчете вектора $\bar{Q}$, так и при расчете векторов-столбцов матрицы W. Общим является интегрирование функции на одних и тех же интервалах. Таким образом, для получения

вектора $\bar{Q}$ необходимо проинтегрировать на заданной системе подинтервалов сигнал $x(t)$, а для получения столбцов матрицы W – базисные функции, на той же системе подинтервалов. Это позволяет для выполнения указанных операций использовать одну и ту же подсистему, структурная схема которой представлена на рисунке 3.12.

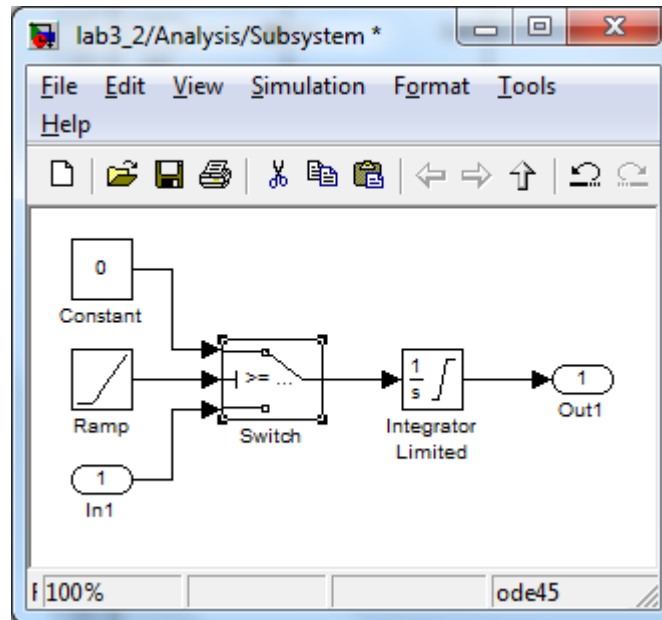


Рисунок 3.12. Блок интегрирования функции на системе подинтервалов

На вход этого блока поступает сигнал – интегрируемая функция. Два других – сигнал, нулевой сигнал и генератор линейно возрастающей функции (в данном случае он описывает время t) являются вспомогательными. Пока аргумент t находится в пределах заданного подинтервала, переключатель Switch подает на интегратор сигнал $x(t)$. Но как только время выходит за пределы подинтервала, сигнал переключается на нулевой и, таким образом, больше не влияет на значение интеграла. Так как MATLAB ориентирован в первую очередь на обработку массивов данных, то имеет смысл задавать в параметрах блоку Switch не один подинтервал, а вектор, состоящий из всех пяти подинтервалов, тогда на выходе также получим вектор значений интеграла. Определение системы подинтервалов с помощью параметров блока Switch показано на рисунке 3.13.

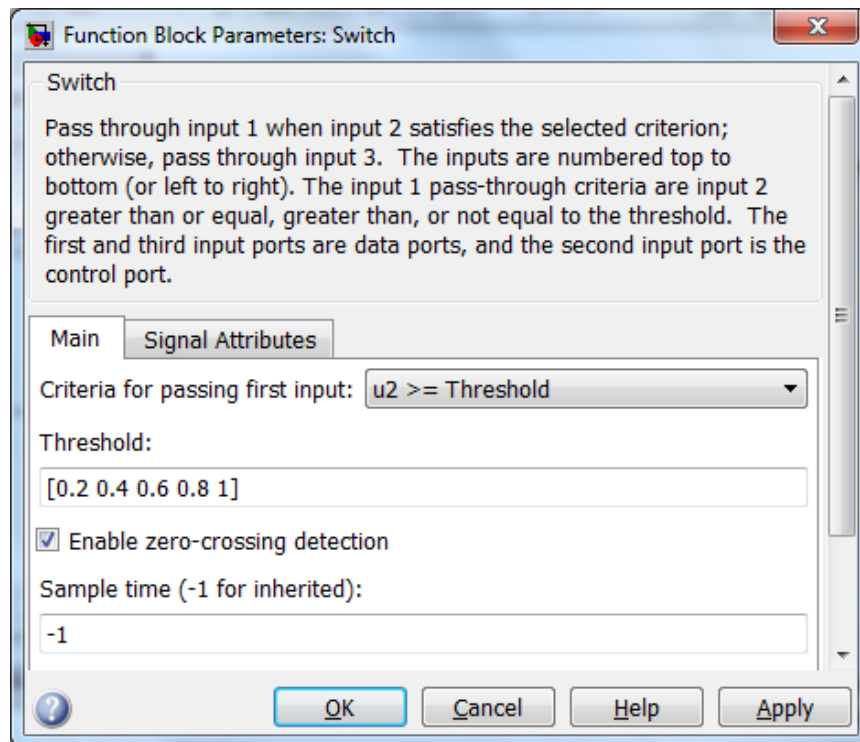


Рисунок 3.13. Окно настройки параметров блока Switch

Теперь рассмотрим структурную схему блока Analysis.

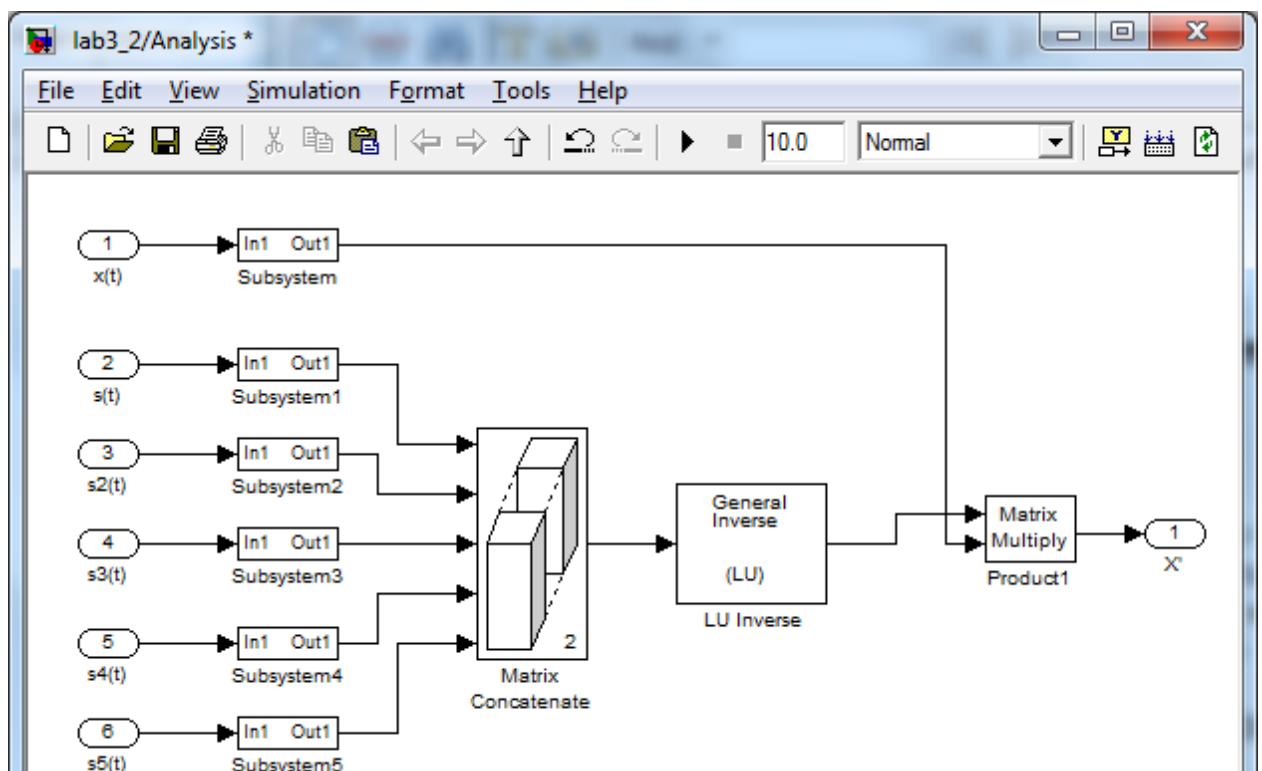


Рисунок 3.14. Структурная схема блока Analysis

На вход блока подается сигнал $x(t)$ и система базисных функций $s(t)$. Подсистема Subsystem интегрирует сигнал и подает на выход вектор $\bar{Q}$, а подсистемы Subsystem 1 – 5 – векторы-столбцы матрицы W . Блок MatrixConcatenate объединяет столбцы в матрицу. Далее размещены блоки расчета обратной матрицы и умножения матриц. В итоге на выходе получаем сигнал аппроксимирующего спектра, который появляется на экране дисплея модели самого высокого уровня (рисунок 3.10). На рисунке 3.15 приведено значение спектра функции, рассматриваемого в качестве примера.

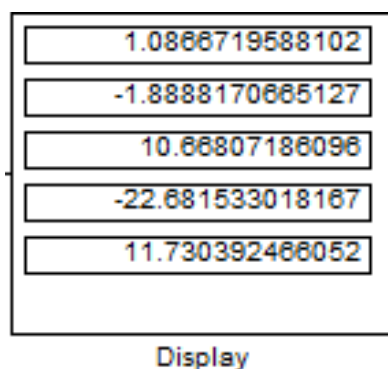


Рисунок 3.15. Значение аппроксимирующего спектра

Можно рассмотреть обратную задачу – построение сигнала по заданному спектру в системе базисных функций. Для наглядности используем спектр, полученный выше, на основе его построим аппроксимацию и сравним ее с реальным сигналом.

Базисные функции умножаются на соответствующие коэффициенты спектра, суммируются и выводятся на виртуальный осциллограф. На этот же осциллограф подается заданный сигнал. Дополненная схема расчета коэффициентов блоками для восстановления сигнала представлена на рисунке 3.16. Графики сигнала и его аппроксимации представлены на рисунке 3.17.

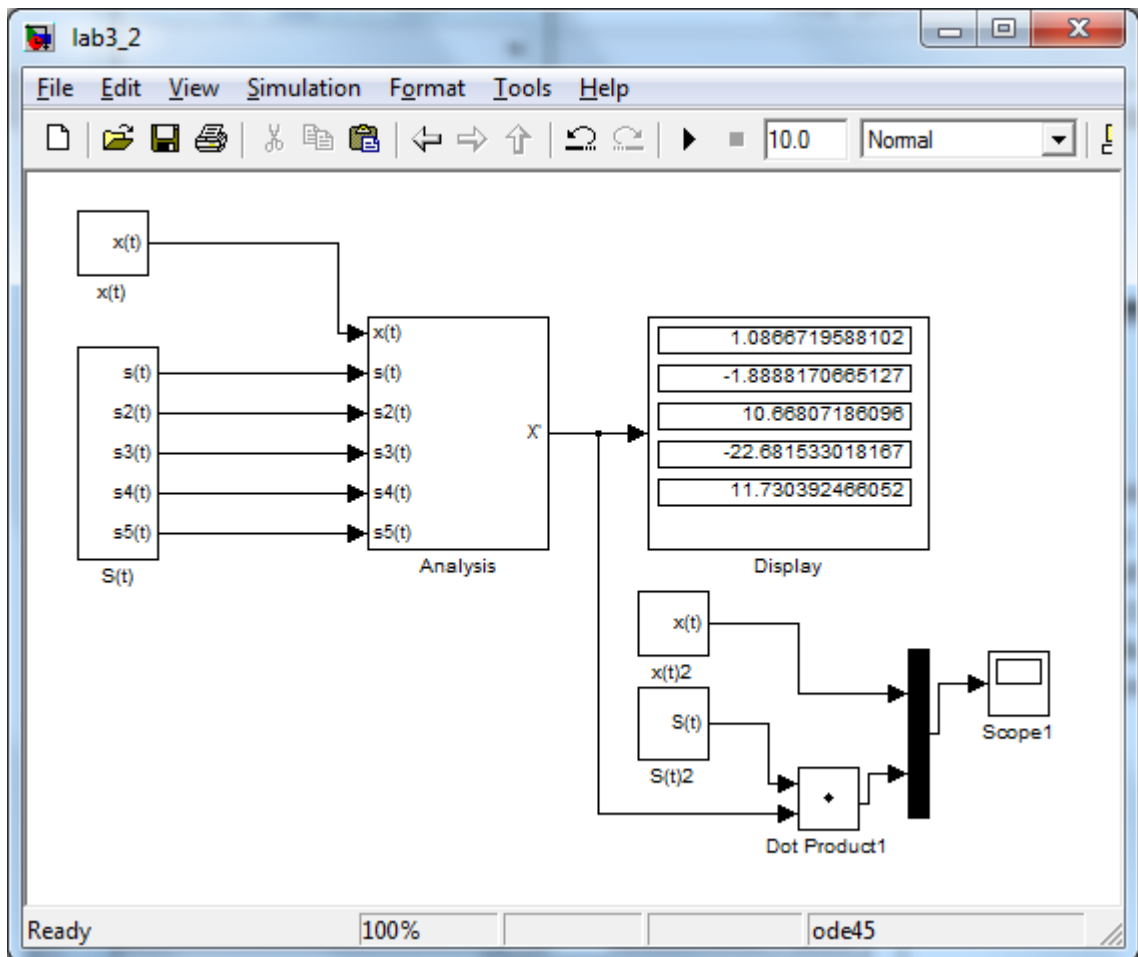


Рисунок 3.16. Структурная схема синтеза сигнала

В модели был рассмотрен случай, когда количество базисных функций равнялось количеству подинтервалов. Но это необязательно. Можно увеличить точность вычислений за счет большего числа подинтервалов. Однако, в этой модели необходимо блок расчета обратной матрицы заменить на блок вычисления псевдо обратной матрицы. Этот блок можно найти в дополнительной библиотеке Simulink – DSP (DigitalSignalProcessing) Blockset.

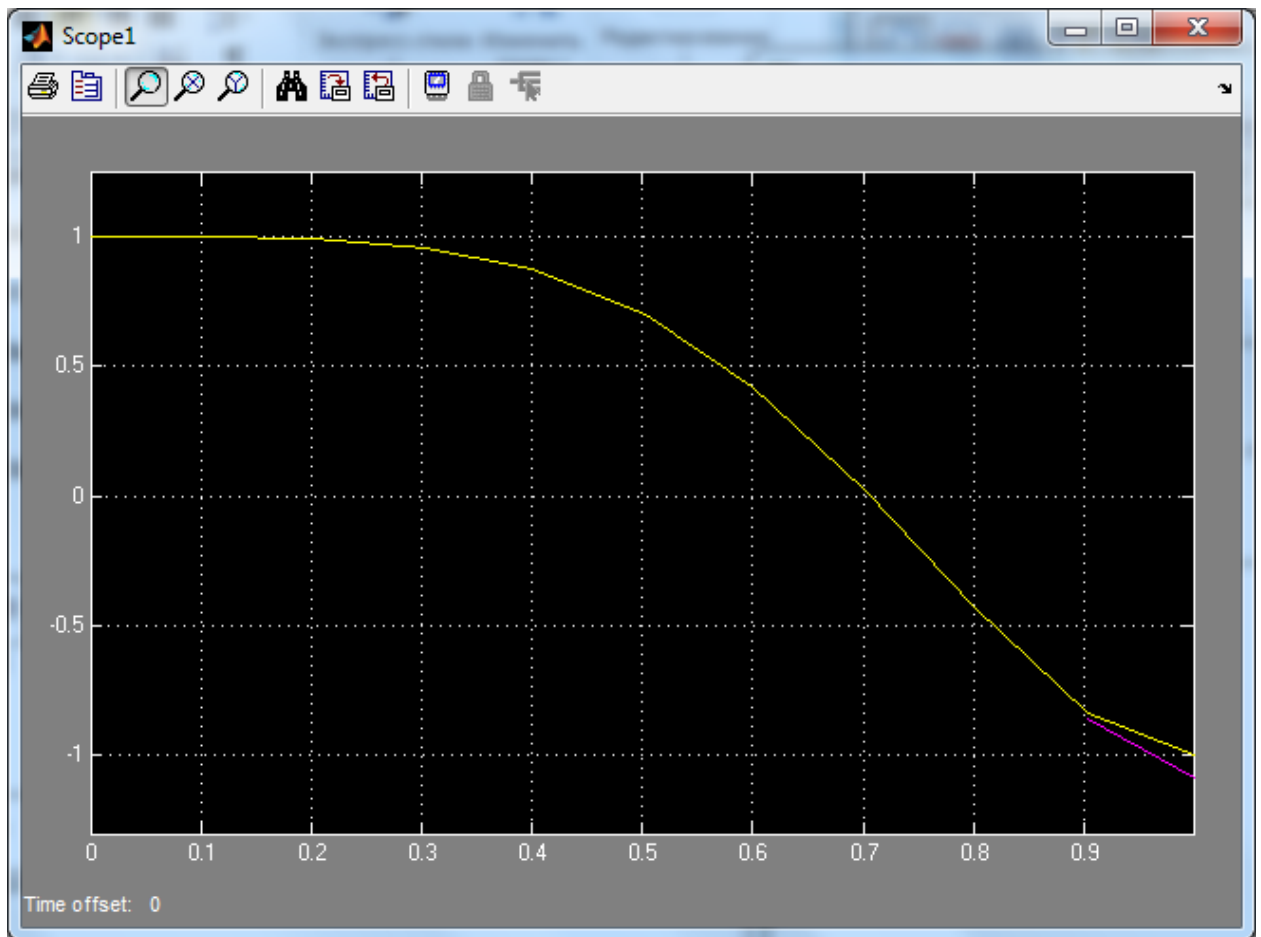


Рисунок 3.17. График сигнала и его аппроксимация

Блочно-импульсная аппроксимация

Одной из наиболее распространенных и удобных для применения в задачах обработки и моделирования сигналов динамических систем оказывается базисная система блочно-импульсных функций. Рассмотрим блочно-импульсную аппроксимацию и некоторые необходимые операции над спектрами.

Пусть отрезок $[0, T]$ разбит на m равных частей, длиной $h = T/m$. На решетке аргумента $\{ih\}_{i=1}^m$ задается система базисных функций

$$v_i(t) = \sigma(t - (i - 1)h) - \sigma(t - ih). \quad (3.7)$$

Здесь $i = 1, \dots, m$; $v_i(t)$ – прямоугольные блочно-импульсные функции; $\sigma(t)$ – функция единичного скачка:

$$\sigma(t) = \begin{cases} 0, & \text{если } t < 0, \\ 1, & \text{если } t \geq 0. \end{cases} \quad (3.8)$$

Аппроксимация сигнала по системе блочно-импульсных функций определяется по формулам (3.3)–(3.4) как

$$\tilde{x}(t) = \sum_{i=1}^m X_0(i) v_i(t), \quad (3.9)$$

где

$$X_0(i) = \frac{1}{h} \int_{(i-1)h}^{ih} x(t) dt. \quad (3.10)$$

Если к блочно-импульсным базисным функциям добавить кусочно-линейные функции:

$$w_i(t) = \left(\frac{2t}{h} + 1 - 2i \right) v_i(t), \quad (3.11)$$

то аппроксимация сигнала по системе локально-импульсных базисных функций первого порядка существенно улучшится и будет определяться по формуле:

$$x_a(t) = \sum_{i=1}^m (X_0(i) v_i(t) + X_1(i) w_i(t)). \quad (3.12)$$

Пример 3.3. Рассмотрим реализацию метода аппроксимации блочно-импульсными функциями нулевого и первого порядков в среде MATLAB/Simulink. В качестве примера используем сигнал $x(t) = \cos \pi t^2$ на интервале $[0, 1]$, $m = 10$. Структурная схема анализа сигнала рассматриваемым методом представлена на рисунке 3.18.

Назначение блоков следующее:

- **BF** генерирует систему базисных функций;
- **X(t)** генерирует анализируемый сигнал;
- **X₀, X₁** рассчитывают элементы спектра.

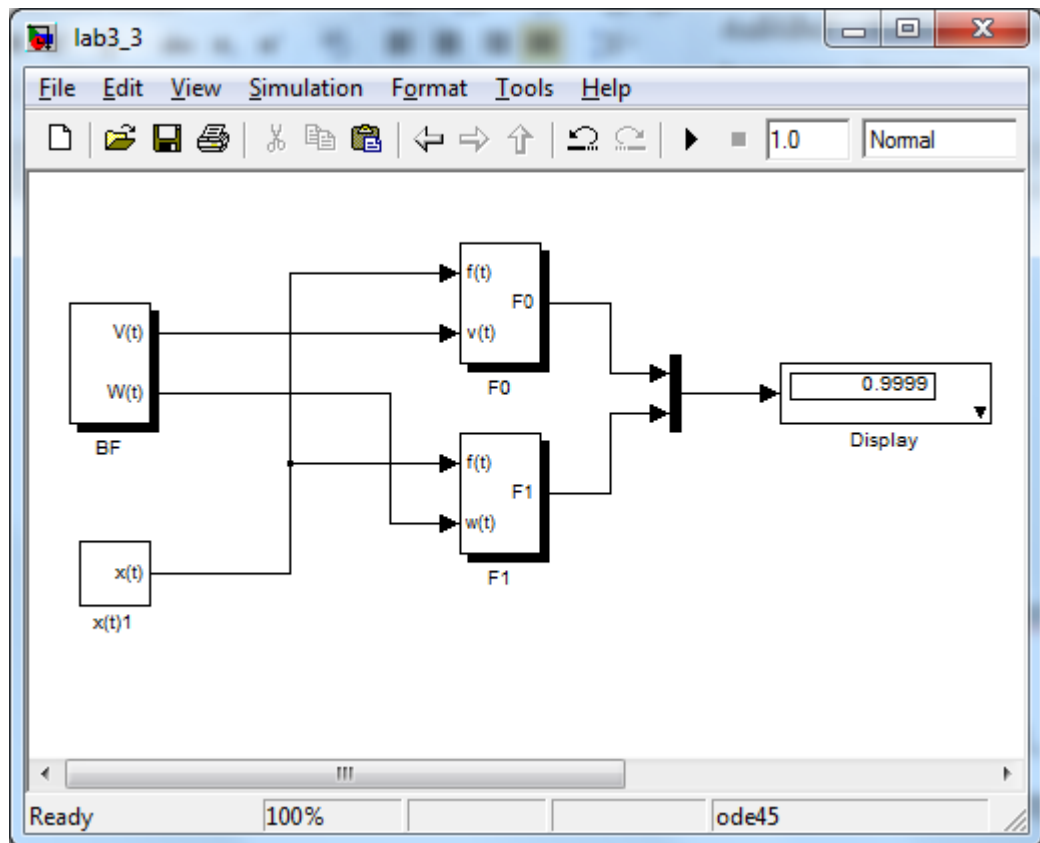


Рисунок 3.18. Структурная схема анализа сигнала блочно-импульсным методом

На рисунках 3.19, 3.20 представлены структурные схемы моделей, генерирующих блочно-импульсные базисные функции нулевого и первого порядков.

На рисунках 3.21, 3.22 представлены схемы для расчета коэффициентов блочно-импульсного и аппроксимирующего импульсного спектров.

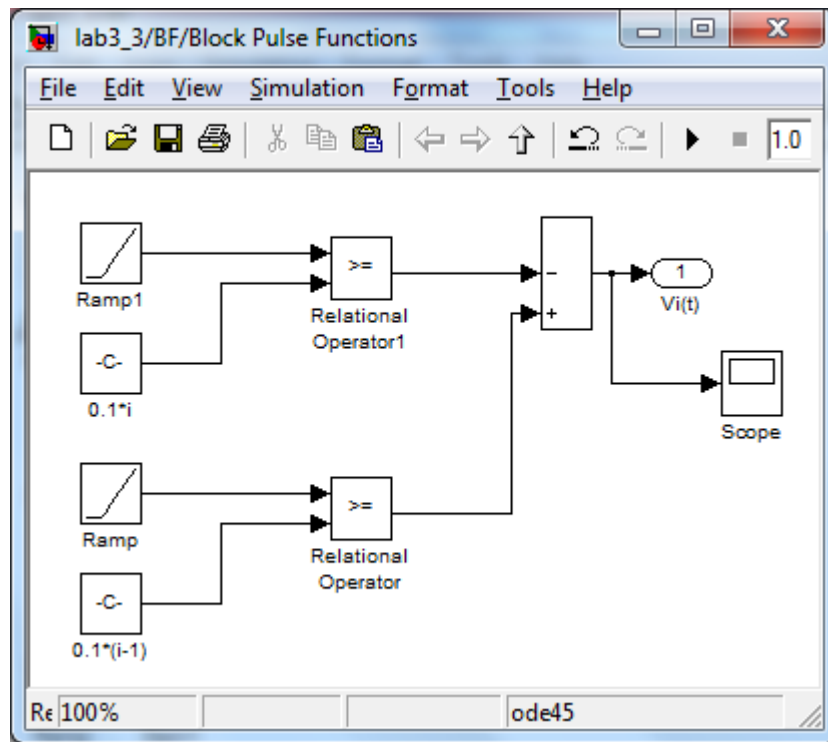


Рисунок 3.19. Структурная схема, генерирующая блочно-импульсные функции нулевого порядка, и график блочно-импульсных функций

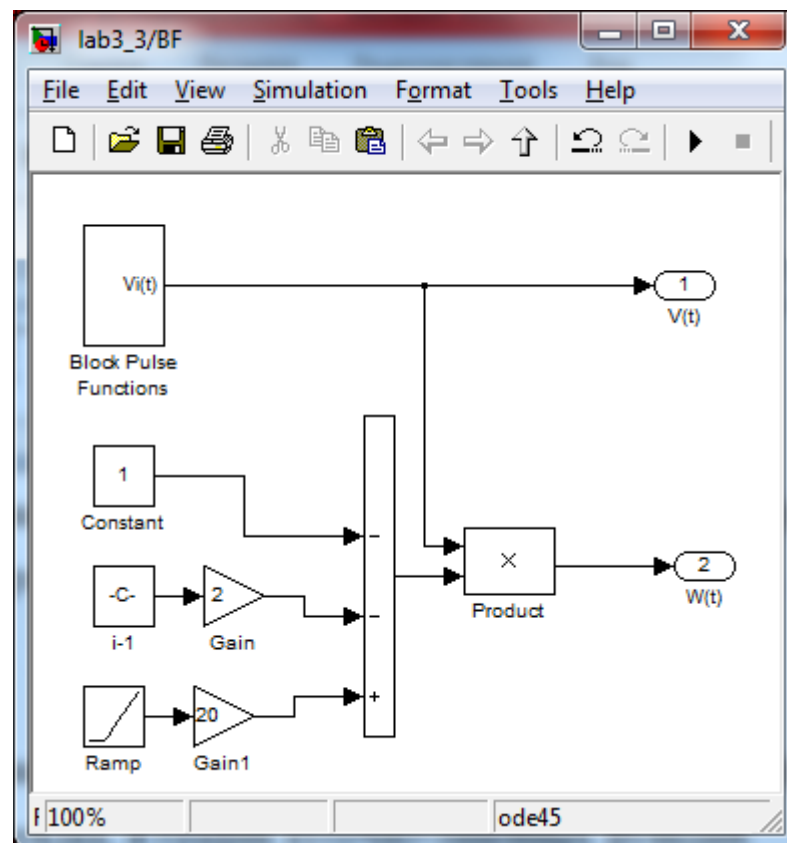


Рисунок 3.20. Структурная схема, генерирующая блочно-импульсные функции первого порядка, и график кусочно-линейных функций

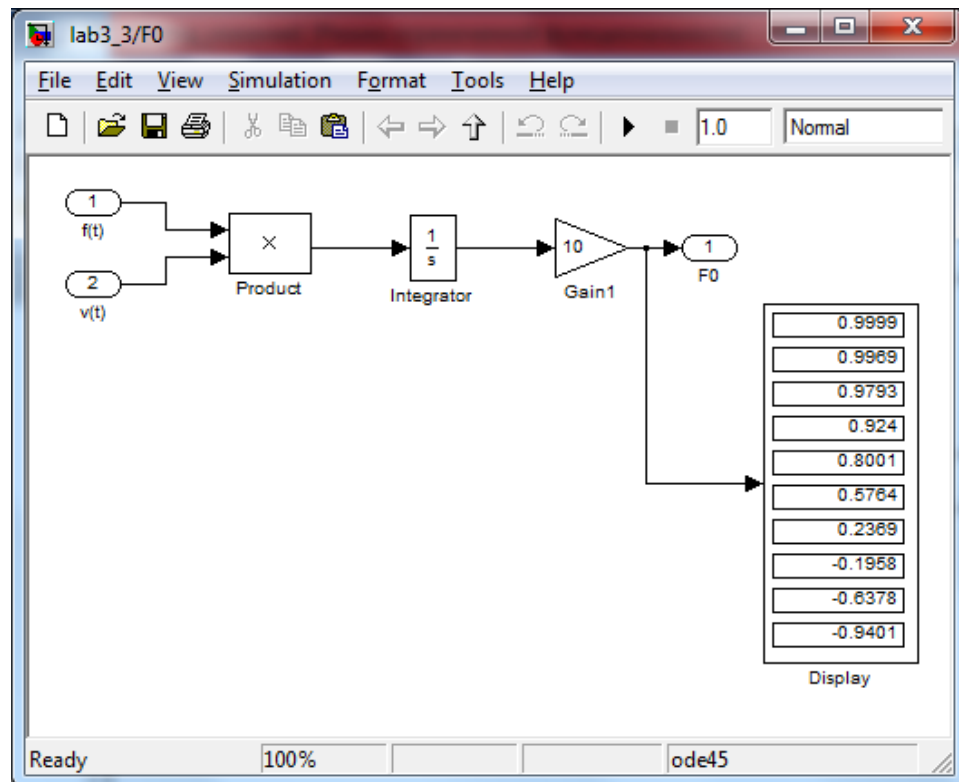


Рисунок 3.21. Расчет коэффициентов блочно-импульсного спектра

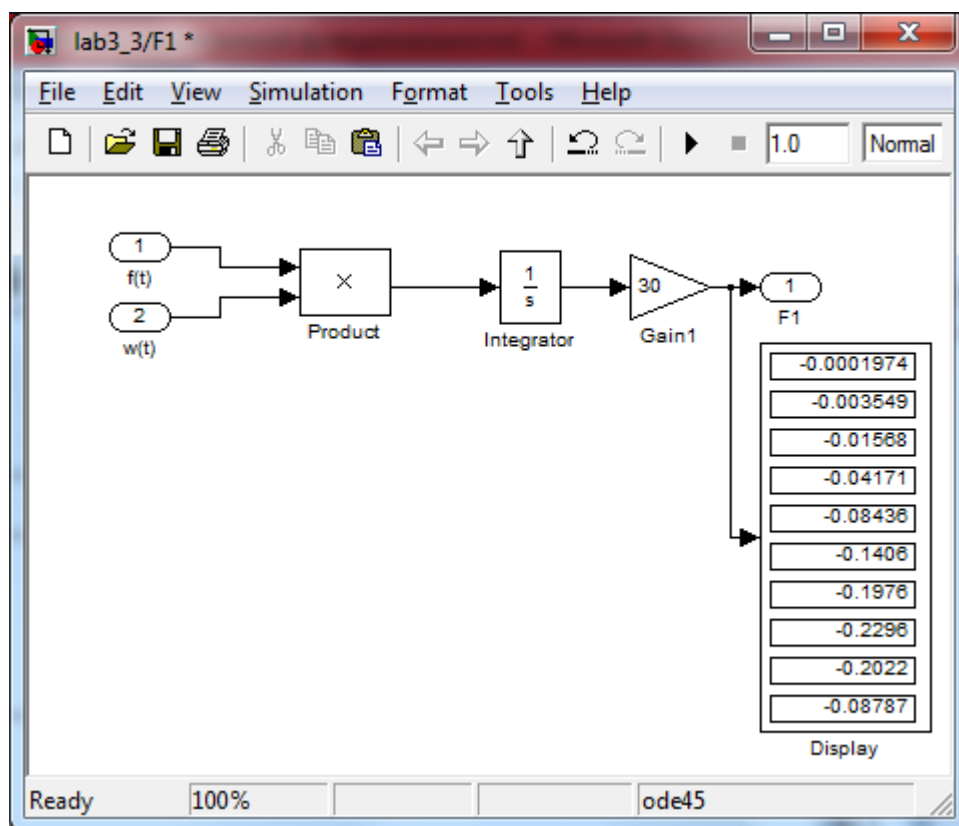


Рисунок 3.22. Расчет коэффициентов аппроксимирующего импульсного спектра

Дополним схему расчета коэффициентов блоками для восстановления сигнала. На вход схемы подаются базисные функции и спектр анализируемого сигнала, на выход – аппроксимация сигнала по соответствующему методу и исходный сигнал. Схема примет следующий вид:

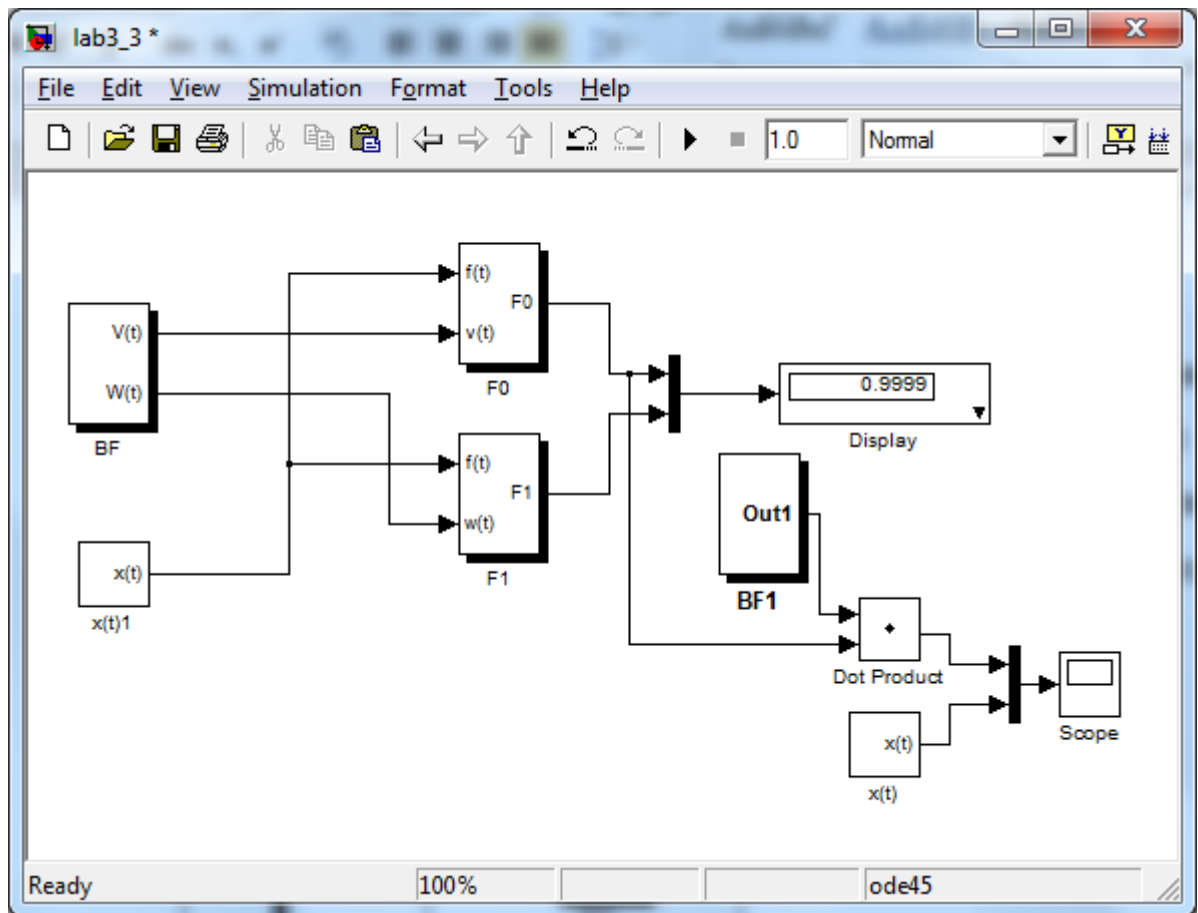


Рисунок 3.23. Структурная схема синтеза сигнала по методу БИФ

На виртуальный осциллограф подаются исходный и восстановленный сигналы. Они представлены на рисунке 3.24 для обоих методов соответственно.

Из графиков видно, что даже при небольшом числе подынтервалов (m) получается довольно точная аппроксимация сигнала.

Точность можно повышать, разбивая интервал определения сигнала на большее количество подынтервалов.

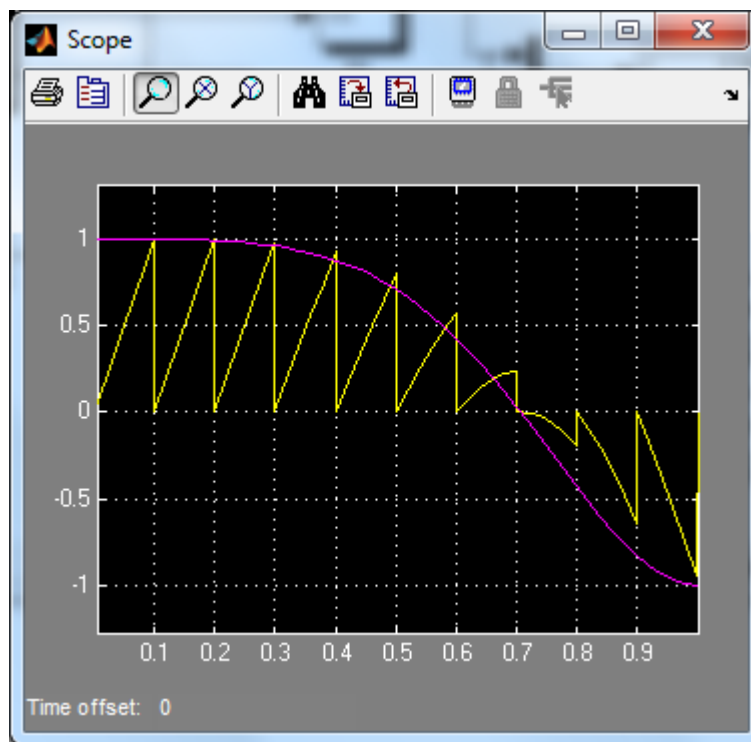


Рисунок 3.24. Результаты аппроксимации сигнала[2]

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью. Параметры сигналов приведены в таблице 3.1. Номер

варианта соответствует номеру, под которым студент записан в списке группы.

Задание 3.1. Необходимо аппроксимировать методом наименьших квадратов сигнал $x(t) = A * func(B) + C$ на интервале $[0,1]$. В качестве базисных функций взять степенные функции 5-го порядка.

Задание 3.2. Необходимо аппроксимировать методом равных площадей сигнал $x(t) = A * func(B) + C$ на интервале $[0,1]$. В качестве базисных функций взять степенные функции 5-го порядка.

Задание 3.3. Необходимо аппроксимировать методом аппроксимации блочно-импульсными функциями нулевого и первого порядков сигнал $x(t) = A * func(B) + C$ на интервале $[0,1]$.

Таблица 3.1. Параметры задачи

№ вар-та	func	A	B	C
1.	cos	1	πt^2	$2t$
2.	sin	0.5	πt	t
3.	tg	0.2	πt^2	0
4.	sin	1	πt^3	t^2
5.	cos	0.1	$\frac{\pi t}{2}$	$2t$
6.	tg	0.6	πt^3	t
7.	sin	1	$2\pi t$	$2t$
8.	cos	0.9	πt^2	t
9.	tg	0.7	$3\pi t$	0
10.	sin	1	3π	t^2

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Какие основные методы используются для определения аппроксимирующего спектра сигнала?
2. В чем заключается суть метода наименьших квадратов для определения аппроксимирующего спектра?
3. В чем заключается суть метода равных площадей для определения аппроксимирующего спектра?
4. В чем заключается суть метода блочно-импульсной аппроксимации?

ЛАБОРАТОРНАЯ РАБОТА 4

Исследование поведения динамической системы (2 ч)

Цель и содержание: изучить методику исследования поведения динамических систем, приобрести навыки реализации таких моделей в системе компьютерной математики MATLAB.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: исследование поведения динамической системы

Теоретическое обоснование

Распространенным способом описания поведения динамической системы является система дифференциальных или интегро-дифференциальных уравнений.

Реализацию таких математических моделей в Simulink рассмотрим на ряде примеров.

Пример 4.1. Модель физического маятника, находящегося под воздействием экспоненциально-затухающего косинусоидального возмущения.

Уравнение движения такого маятника имеет вид:

$$\begin{aligned}\frac{d^2 y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_2 y(t) &= a_3 e^{-a_4 t} \cdot \cos a_5 t, \\ y(t)|_{t=0} &= y_0, \\ \frac{dy(t)}{dt} \Big|_{t=0} &= y'_0,\end{aligned}\tag{4.1}$$

Выбрав числовые значения параметров, например: $a_1 = a_2 = 0,1$, $a_3 = -5$, $a_4 = 1$, $a_5 = 0,1$, получим следующее уравнение:

$$\begin{aligned}
 y' + 0,1y' + 0,1y &= -5e^{-t} \cdot \cos(0,1t), \\
 y(0) &= -1,5, \\
 y'(0) &= 2.
 \end{aligned}
 \tag{4.2}$$

Структурная схема модели будет иметь вид, показанный на рисунке 4.1.

Результаты работы модели показан на экране виртуального осциллографа (рисунок 4.2), а параметрический график зависимости производной сигнала от сигнала (фазовый портрет маятника) изображен на рисунке 4.3.

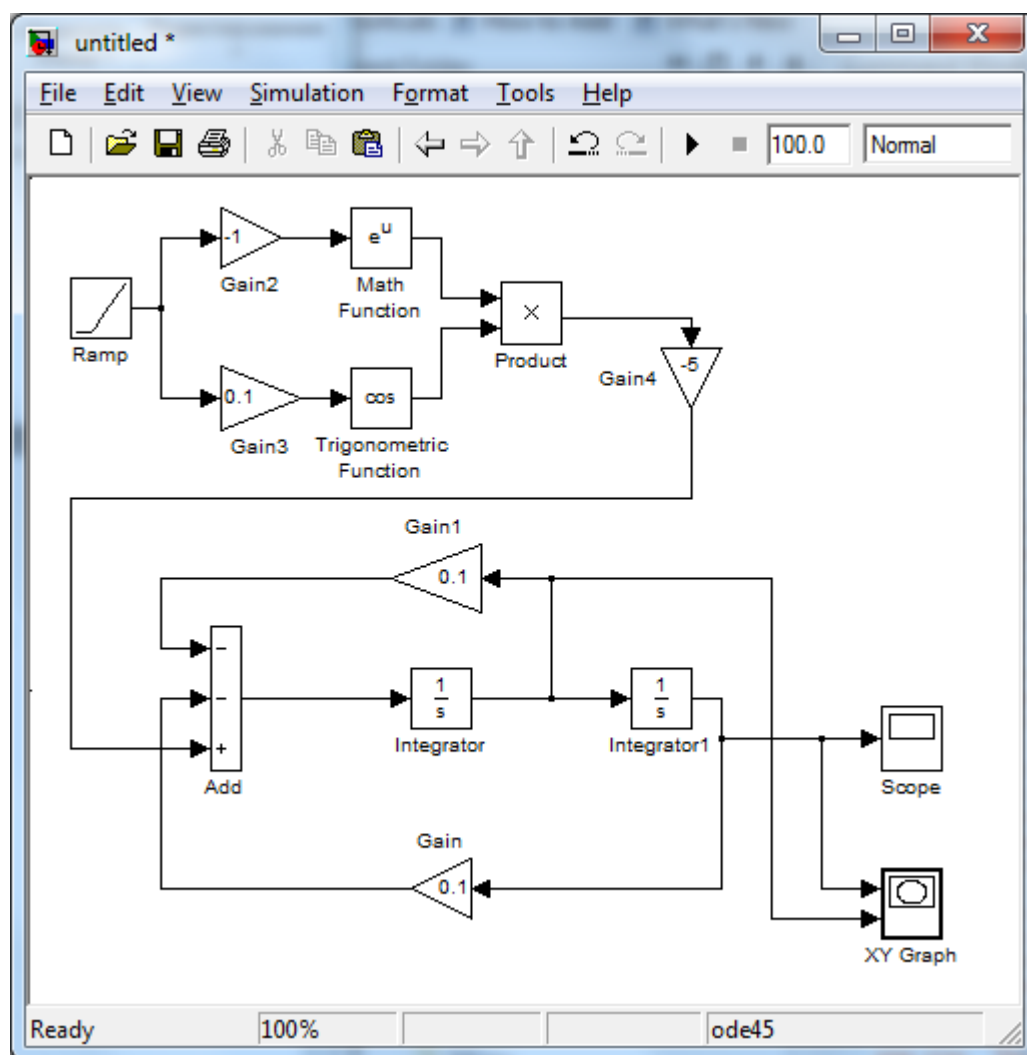


Рисунок 4.1. Модель физического маятника

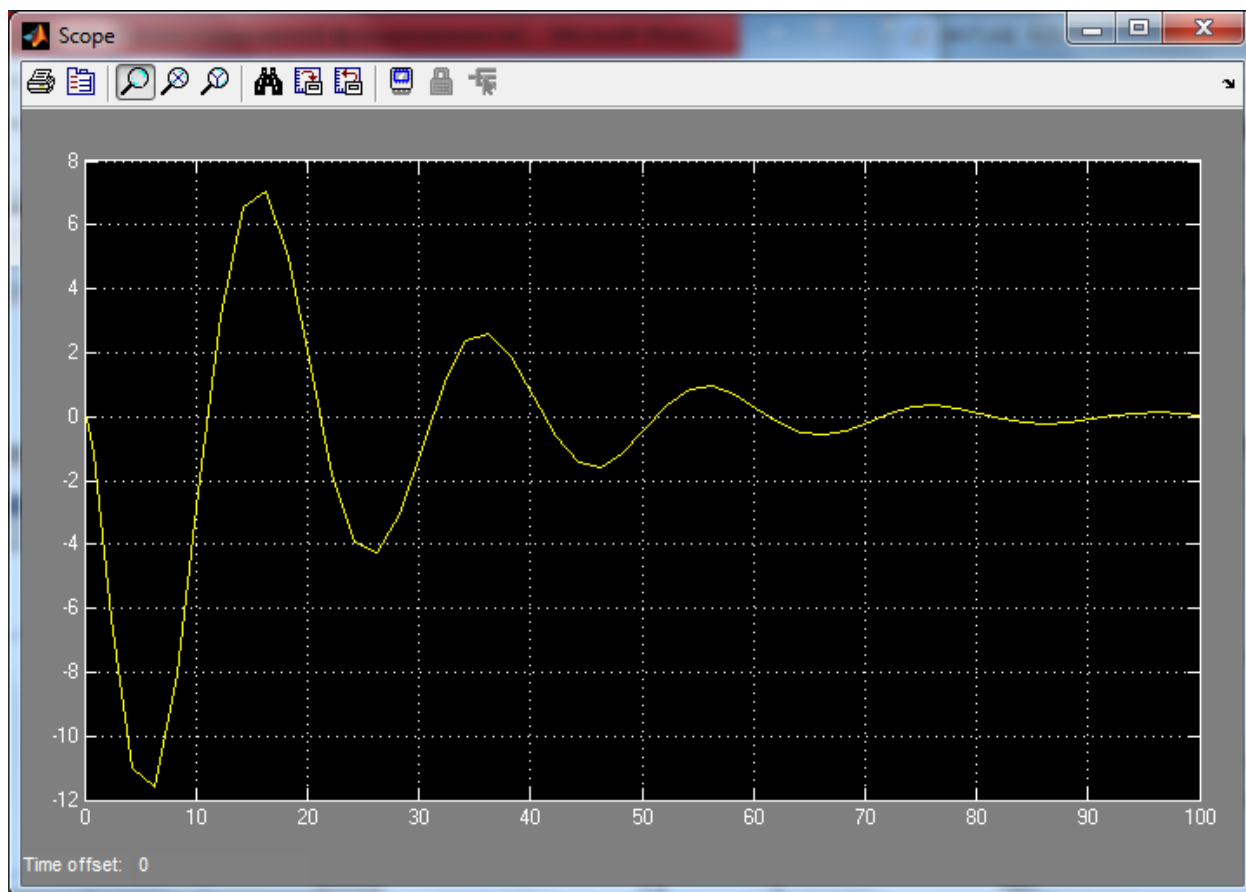


Рисунок 4.2. Движение физического маятника

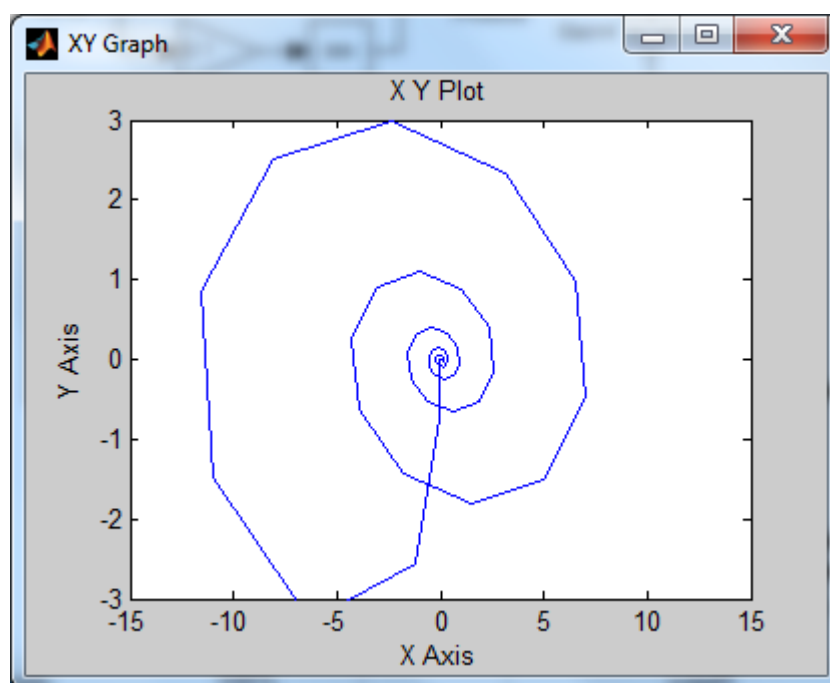


Рисунок 4.3. Фазовый портрет физического маятника

Пример 4.2. Модель динамической системы, описываемой дифференциальным уравнением 3-го порядка.

Задано дифференциальное уравнение:

$$\frac{d^3 y(t)}{dt^3} + 2,5 \frac{d^2 y(t)}{dt^2} + 6 \frac{dy(t)}{dt} + 2,5 y(t) = e^{-t},$$

$$y(0) = 1; y'(0) = -1; y''(0) = 2. \quad (4.3)$$

Структурная схема модели динамической системы, построенная по обычным правилам аналоговой вычислительной техники, приведена на рисунке 4.4.

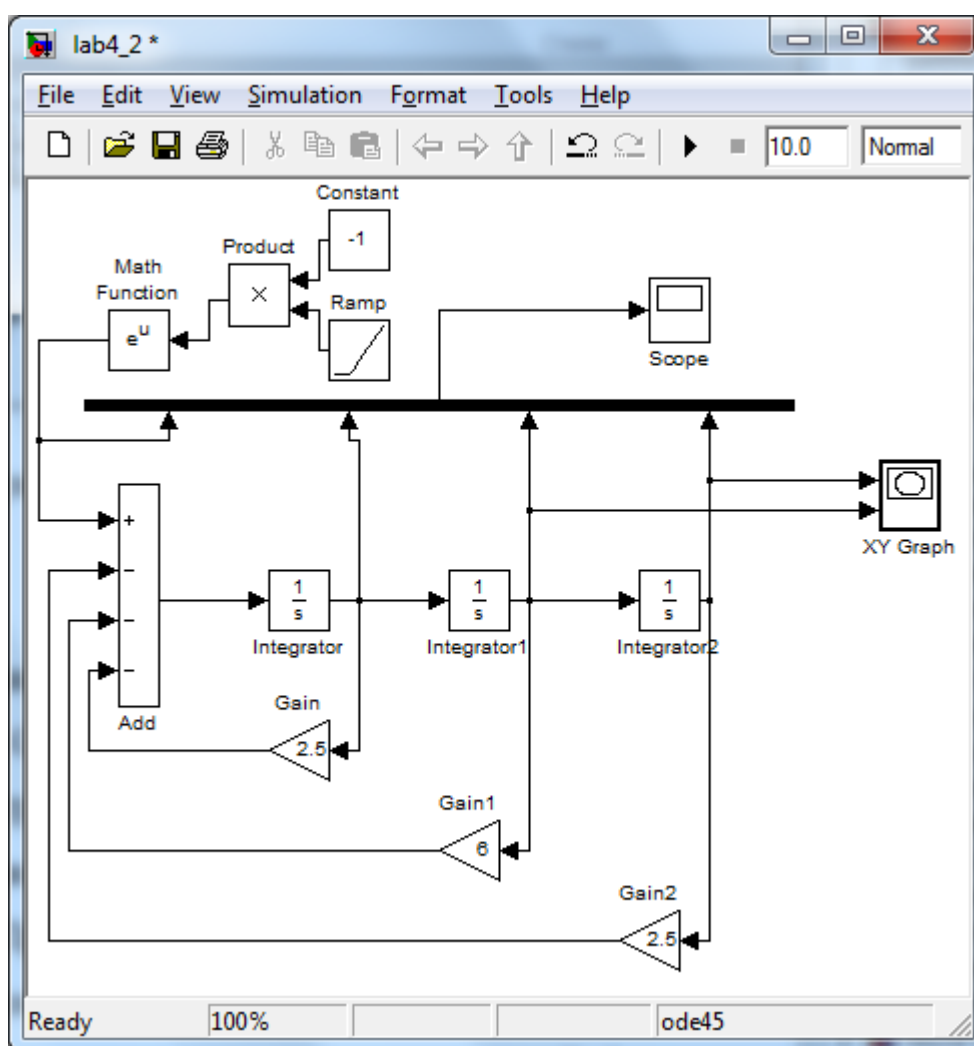


Рисунок 4.4. Структурная схема модели динамической системы 3 порядка

Коэффициенты уравнения устанавливаются в окнах параметров масштабных блоков Gain – Gain2. Начальные условия для функций и

производных – в окнах параметров интеграторов Integrator – Integrator2. На рисунке 4.5 представлено окно параметров для интегратора Integrator, соответствующего $y''(0) = 2$, начальные условия для интеграторов Integrator1 – Integrator2, соответствующие $y'(0) = -1$ и $y(0) = 1$ соответственно задаются аналогично. Правая часть дифференциального уравнения сформирована с помощью блоков Ramp (генератор аргумента – t) и блока Math Function, настроенного на реализацию экспоненциальной функции. Визуализация переходного процесса показана на экране виртуального осциллографа (рисунок 4.6). Фазовый портрет системы на экране виртуального двух координатного регистратора (XY Graph) приведен на рисунок 4.7.

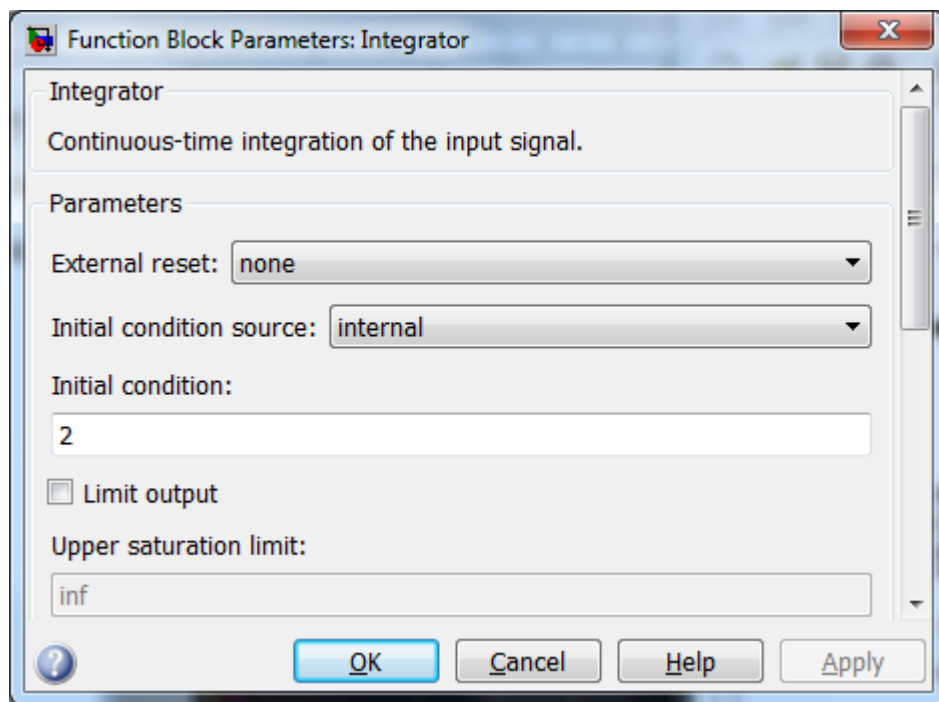


Рисунок 4.5. Окно параметров функционального блока Integrator

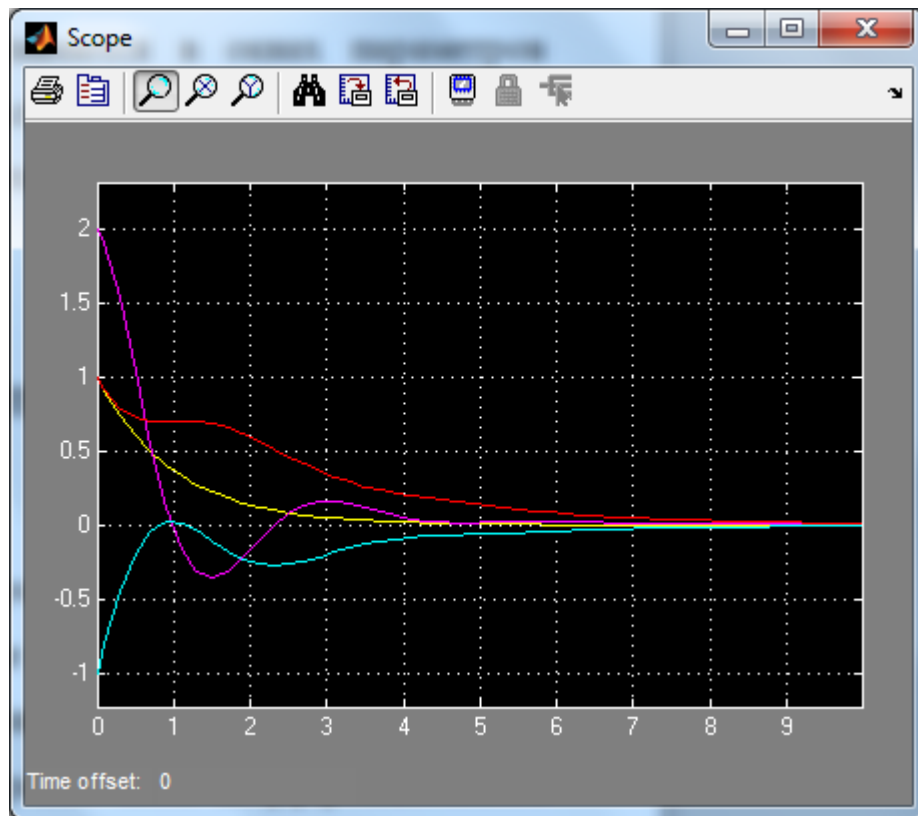


Рисунок 4.6. Переходный процесс системы

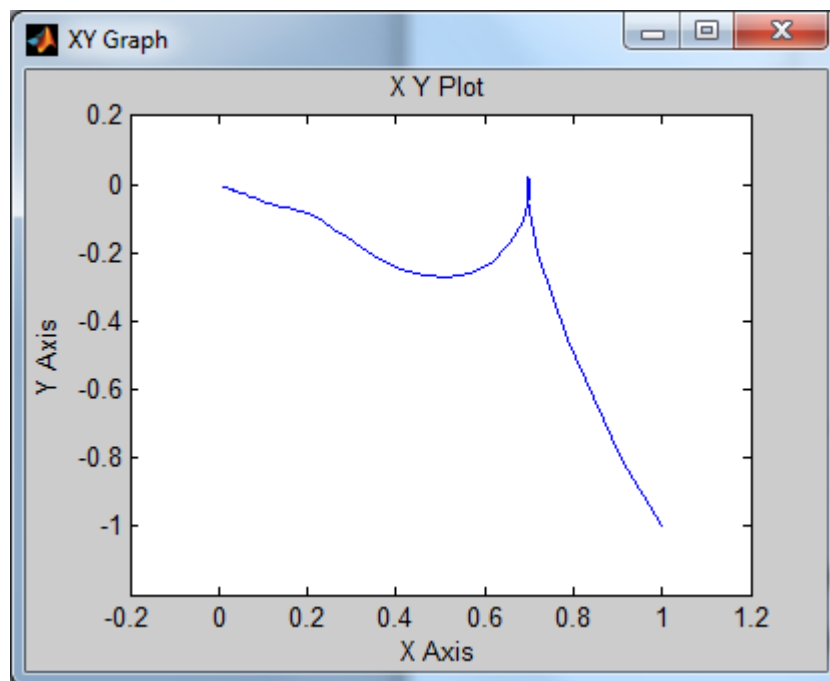


Рисунок 4.7. Фазовый портрет системы

Пример 4.3. Модель траектории полета тела, брошенного с начальной скоростью под углом к горизонту. Предположим, что наблюдатель, находясь

над уровнем земли на высоте 1м, бросил камень под углом 30 градусов к горизонту с начальной скоростью 20 м/сек. Необходимо реализовать модель траектории полета камня под действием силы тяжести и определить расстояние от наблюдателя до точки падения камня. Влиянием атмосферы на полет камня пренебречь. Уравнения движения камня имеют вид:

$$\begin{aligned} y &= y_0 + v \cdot \sin(\alpha) \cdot t - g \cdot \frac{t^2}{2}, \\ x &= v \cdot \cos(\alpha) \cdot t, \end{aligned} \quad (4.4)$$

где: $y_0 = 1\text{м}$, $v = 20\text{ м/сек}$, $\alpha = 30^\circ$, $g = 9,81\text{ м/сек}$.

Структурная схема модели траектории камня приведена на рисунке 4.7.

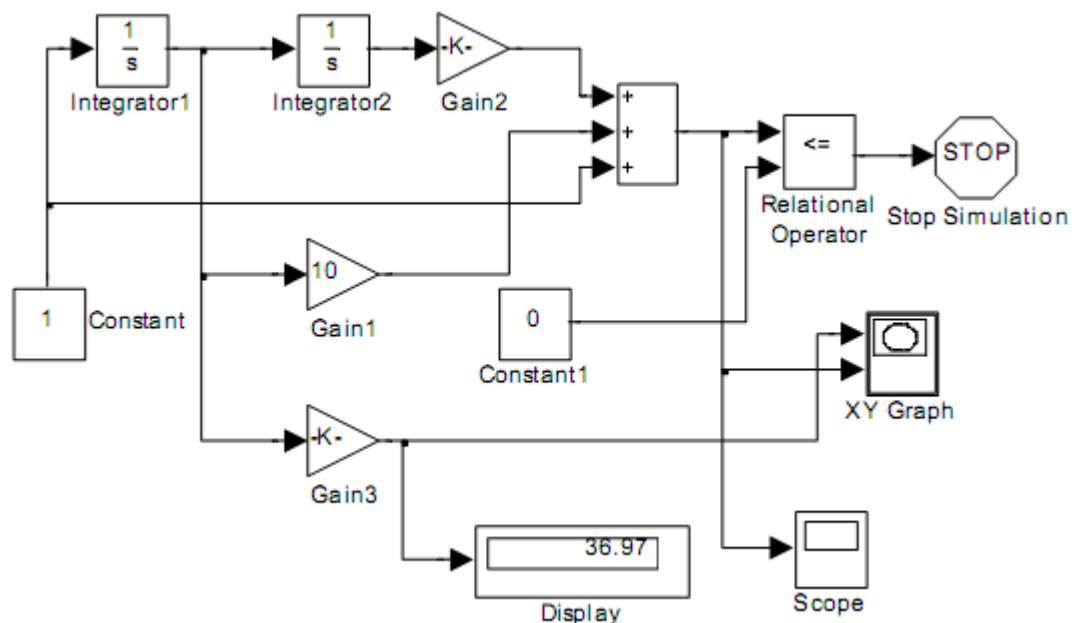


Рисунок 4.7. Структурная схема модели траектории камня

Вертикальная составляющая начальной скорости камня задается блоком Gain1, горизонтальная составляющая – блоком Gain3, ускорение земного тяготения – блоком Gain2. Значение текущей высоты полета камня как функции времени формируется на выходе сумматора. Сигнал окончания моделирования формируется блоками Relational Operator и Stop Simulation в момент времени, когда высота сравнивается с нулем. Показания цифрового регистратора соответствует длине пути по горизонтали, пройденной камнем до момента соприкосновения с землей. На рисунках 4.8 и 4.9 показаны соответственно траектория камня на экране виртуального двух

координатного регистратора и график изменения во времени высоты полета камня.

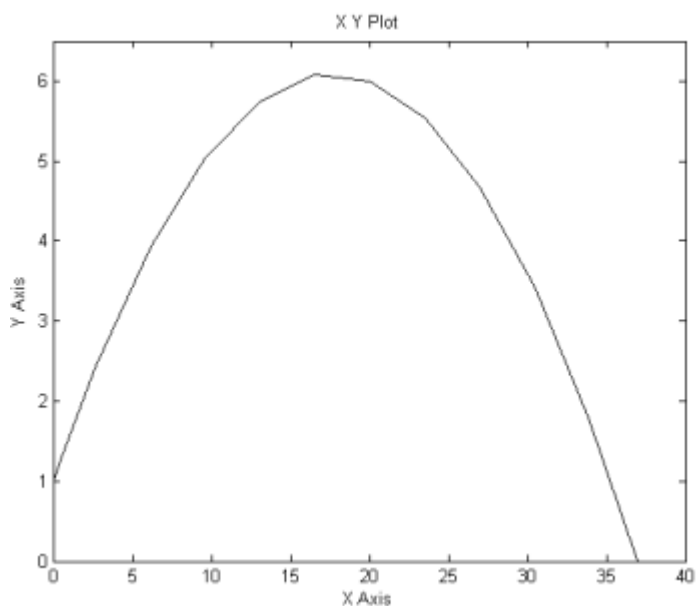


Рисунок 4.8. Траектория полета камня, брошенного под углом 30 градусов к горизонту с начальной высоты 1м, с начальной скоростью 20 м/сек

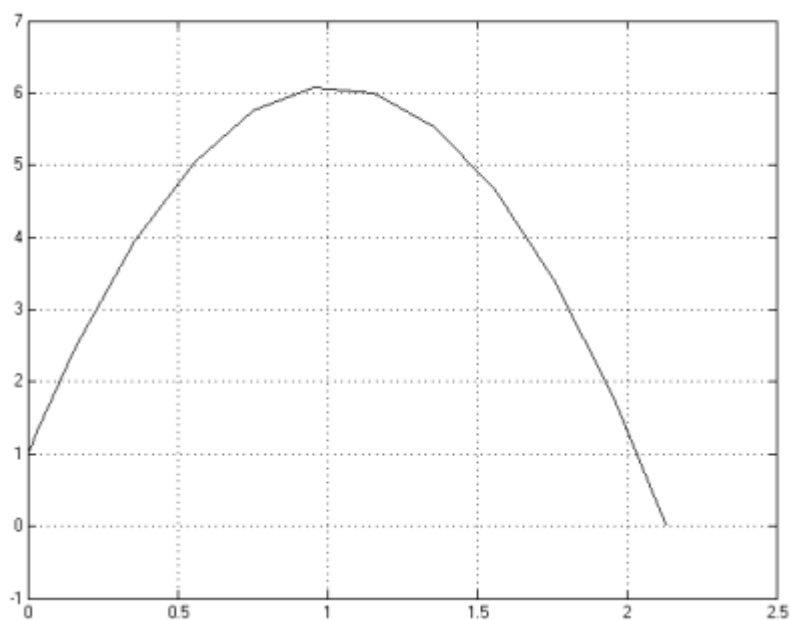


Рисунок 4.9. График изменения во времени высоты полета камня [2]

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение:

операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 4.1. Синтезировать структурную схему модели дифференциального уравнения 3 порядка:

$$\frac{d^2y}{dx^2} + A \frac{dy}{dx} + By = 0 \quad (4.5)$$

С начальными условиями: $y(0), y'(0), y''(0)$. Построить фазовый портрет системы. Параметры сигналов приведены в таблице 4.1. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Задание 4.2. Синтезировать структурную схему модели дифференциального уравнения 3 порядка:

$$\frac{d^3y}{dt^3} + C \frac{d^2y}{dx^2} + D \frac{dy}{dx} + Ey = 0 \quad (4.6)$$

С начальными условиями: $y(0), y'(0), y''(0)$. Построить фазовый портрет системы. Параметры сигналов приведены в таблице 4.1. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Задание 4.3. Построить модель траектории полета пушечного ядра под действием силы тяжести и определить расстояние от пушки до точки падения

ядра, выпущенного с начальной скоростью V_0 под углом α к горизонту. Высота от уровня земли до дульного утолщения равна h . Влиянием атмосферы на полет ядра пренебречь.

Параметры задачи приведены в таблице 4.2. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Таблица 4.1. Параметры задач 4.1 и 4.2

№ вар-та	A	B	C	D	E	$y(0)$	$y'(0)$	$y''(0)$
1.	0.2	0.4	0.5	2	0.3	1	2	0
2.	1	0.8	3.2	1.4	0.6	-1	1	2
3.	3	4	4.2	1	0.4	0	2	1
4.	4.2	3	1	2	0.5	1	-1	-1
5.	4.2	1	0.4	4	4.2	2	1	0
6.	1	2	0.5	3	1	-1	0	1
7.	4	4.2	1	0.8	3.2	1	2	0
8.	3	1	2	4	4.2	-1	1	2
9.	1	0.4	4	3	1	-2	2	1
10.	2	0.5	3	1	0.4	1	0	1

Таблица 4.2. Параметры задачи 4.3

№ вар-та	V_0 , м/с	α , градусов	h , метров
1.	32	30	1
2.	38	35	1,2
3.	34	32	2,2
4.	42	36	1,4
5.	37	34	0,4
6.	29	33	0,5
7.	52	30	2,8
8.	44	36	2
9.	47	33	4
10.	49	31	3

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;

- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Приведите как минимум 3 примера динамических систем?
2. Какие основные математические блоки используются для моделирования динамических систем?
3. Дифференциальное уравнение какого порядка потребуется для описания модели физического маятника? Приведите общий вид такого уравнения.

ЛАБОРАТОРНАЯ РАБОТА 5

Разработка структур виртуальных моделей для решения оптимизационных задач (2 ч)

Цель и содержание: изучить методику разработки структур виртуальных моделей, приобрести навыки построения модели линейного преобразователя.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: структура виртуальных моделей; модели линейного преобразователя.

Теоретическое обоснование

Библиотека виртуальных решающих элементов графических приложений системы MATLAB включает в себя чрезвычайно большое количество элементов. Для удобства рассмотрения приведем и прокомментируем свойства элементов, которые потребуются для решения поставленных в работе задач.

На рисунке 5.1 приведена совокупность используемых элементов. Необходимо отметить, что форматы задания сигналов в приложениях Simulink и Sim Power System отличаются друг от друга, что не позволяет соединять блоки различных приложений друг с другом без согласования. Пиктограммы содержат обозначения входных сигналов блоков Simulink в виде стрелки «>», тогда как полюса блоков приложения Sim Power System обозначены в виде квадратов «□».

Для согласования сигналов этих приложений некоторые блоки приложения Sim Power System содержат полюса, характерные для каждого из приложений в отдельности: «>» и «□».

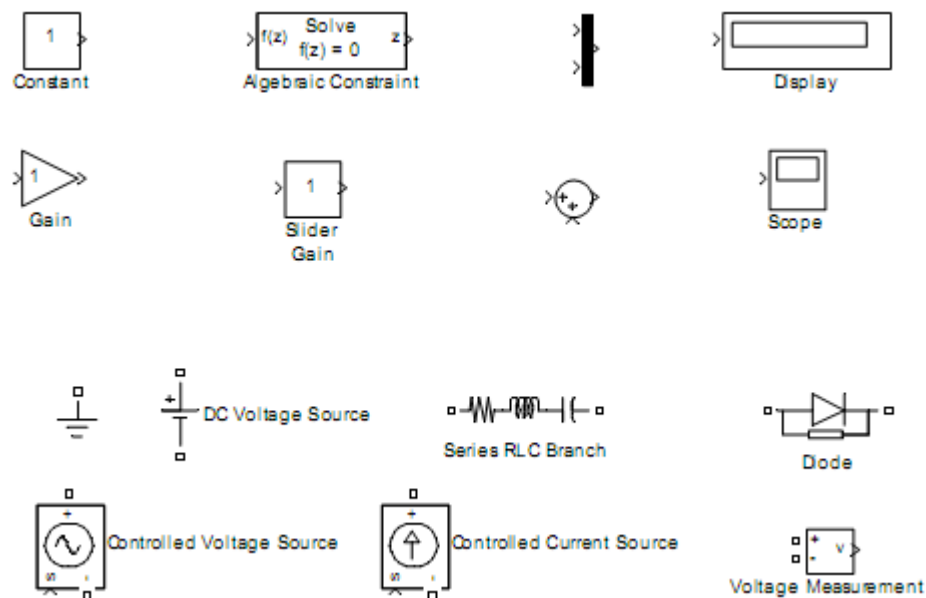


Рисунок 5.1. Совокупность используемых элементов

Рассмотрим элементы приложения Sim Power System. Слева направо:

- Виртуальный источник электродвижущей силы (ЭДС) (DC Voltage Source) – задает значение и полярность э.д.с. постоянного тока.
- Виртуальная ветвь электрической цепи в виде последовательного соединения резистора, конденсатора и катушки индуктивности. Задание параметров ветви ее топологии (частных случаев) обеспечивается с помощью меню (окна параметров).
- Виртуальный диод (Diode) – аналог диода. Падение напряжения в проводящем направлении и обратное сопротивление в запертом состоянии задаются с помощью меню элемента и могут изменяться в широких пределах. По умолчанию падение напряжения на диоде в проводящем состоянии равно 800 мВ (аналог кремниевого полупроводникового диода).
- Управляемый источник напряжения (Controlled Voltage Source) – Виртуальный источник электродвижущей силы, значение э.д.с. которого определяется сигналом формата Simulink. Подходит для обеспечения связи элементов приложений Simulink и Sim Power System.
- Управляемый источник тока (Controlled Current Source) – виртуальный источник тока, величина тока которого определяется сигналом формата

Simulink. Подходит для обеспечения связи элементов приложений Simulink и Sim Power System.

– Измеритель напряжения (Voltage Measurement) – виртуальный вольтметр приложения Sim Power System. Кроме измерения сигналов в системе Sim Power System может быть использован для обеспечения связи элементов приложений Simulink и Sim PowerSystem.

Необратимый линейный преобразователь в системе Simulink

Ограничимся линейным преобразованием с 4 сигналами: $a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = 0$. Обобщение на большее число сигналов является тривиальной задачей. Структурная схема преобразователя в среде Simulink приведена на рисунке 5.2.

Схема состоит из 4 масштабных преобразователей, сумматора и элемента алгебраического ограничения. Параметры масштабных преобразователей соответствуют коэффициентам линейного преобразования. Для корректной работы преобразователя необходимо определить, какой из 4 сигналов является искомым. Пусть для определенности это будет x_3 . Следующим этапом необходимо соединить выход (Out1) со входом (In3). На остальные три входа должны быть поданы значения сигналов x_1, x_2, x_4 . В результате работы системы будет реализовано следующее линейное преобразование: $x_4 = -\frac{1}{a_4}(a_1x_1 + a_2x_2 + a_3x_3)$. В этой структурной схеме элемент алгебраического ограничения исполняет роль виртуального операционного усилителя, формирующего такой выходной сигнал, который обеспечивает на его входе нулевой сигнал. Если такую схему свернуть в подсхему, то из таких подсхем можно строить необратимые модели систем линейных алгебраических уравнений (по классификации квазианалоговых моделей – α -аналоги системы линейных алгебраических уравнений).

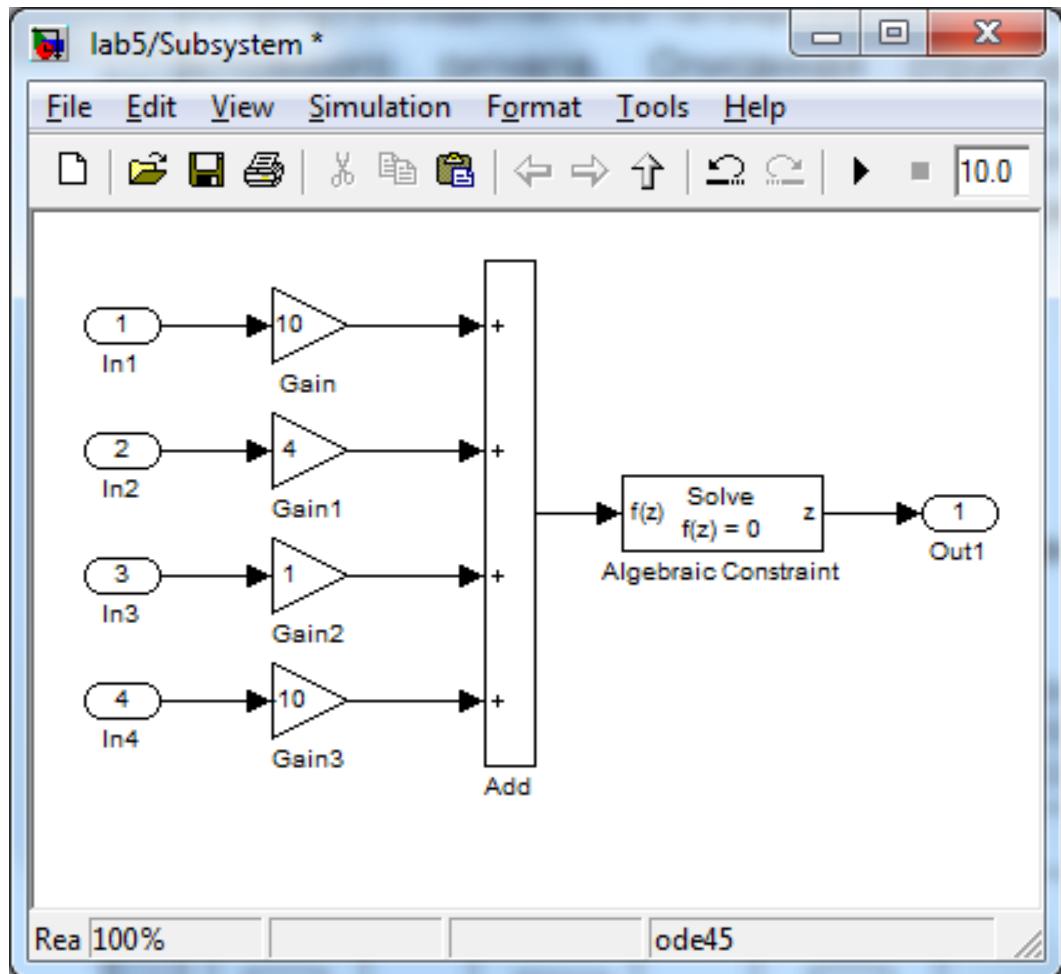


Рисунок 5.2. Структурная схема модели линейного преобразователя [2]

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае

обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью. Параметры сигналов приведены в таблице 5.1. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Задание 5.1. Синтезировать структурную схему модели линейного преобразователя в блоке Subsystem для следующего линейного уравнения:

$$a_1x_1 + a_2x_2 + a_3x_3 = a_4$$

Задание 5.2. Синтезировать структурную схему модели линейного преобразователя в блоке Subsystem1 для следующего линейного уравнения:

$$a_5x_1 + a_6x_2 + a_7x_3 = a_8$$

Задание 5.3. Синтезировать структурную схему модели линейного преобразователя в блоке Subsystem2 для следующего линейного уравнения:

$$a_9x_1 + a_{10}x_2 + a_{11}x_3 = a_{12}$$

Таблица 5.1. Параметры задач

№ вар- та	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}
1.	5	2	1	2.5	3	7	2	-1.5	4	0.5	5	11.5
2.	2	1.5	3.5	-16.25	2	3	4	-19.5	2	-0.5	5	-22.75
3.	0.5	-8	-9	-30.5	4	5	6	25.5	-1.5	2	3	8.5
4.	-1	2.5	9	-35.25	5	6	7	-22	1	2	-2.5	10
5.	3	-2	-7	-8.5	6	7	8	38	1	-3.5	6	32
6.	4.5	5.5	4	-3.25	7	8	9	-12	-4.5	5	1	38
7.	6.5	-5	-6	39	1	2	3	10.5	4	1	-5.5	7.25
8.	5	7.5	9.5	27.75	2	3	4	-4.5	1	-6.5	9	11.75
9.	-4	10	-3	-67.5	3	4	5	15.5	-7.5	8	1	-90.5
10.	8	1	6	64.5	4	5	6	22.5	7	1	-8.5	70

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Как обозначаются входные сигналы блоков Simulink?
2. Как обозначаются входные сигналы блоков приложения Sim Power System?
3. Приведите как минимум 3 примера блоков, которые могут быть использованы для согласования сигналов приложений Simulink и Sim Power System. Пример должен содержать пиктограмму соответствующего блока и его описание.
4. Опишите подробно методику изменения характеристик блока «виртуальный диод» приложения Sim Power System.

ЛАБОРАТОРНАЯ РАБОТА 6

Модели системы алгебраических уравнений (2 ч)

Цель и содержание: изучить методику разработки модели системы алгебраических уравнений, приобрести навыки построения таких моделей.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: разработки модели системы алгебраических уравнений

Теоретическое обоснование

Из подсхем типа представленного на рисунке 5.2 можно построить модель системы линейных алгебраических уравнений 3 порядка, поскольку один из входов будет использован для задания правой части соответствующего уравнения. Структурная схема модели системы линейных алгебраических уравнений с 3 неизвестными приведена на рисунке 6.1. В ее состав входят три подсхемы рисунок 5.2, источник постоянного сигнала, мультиплексор и цифровой регистратор (дисплей). Одноименные полюсы подсхем объединены и подключены к выходам соответствующих подсхем, формирующих значения неизвестных x_1, x_2, x_3 , объединенные четвертые полюса подключены к источнику постоянного сигнала. Описанная структурная схема позволяет определять решения системы линейных алгебраических уравнений лишь в случае, когда матрица коэффициентов системы является положительно определенной. В данном примере параметры схемы соответствовали следующей системе уравнений:

$$10x_1 + 4x_2 + x_3 = 10,$$

$$4x_1 + 10x_2 + x_3 = -29,$$

$$x_1 + 4x_2 + 10x_3 = -3,5$$

На цифровом регистраторе рисунка 6.1 показано решение системы уравнений ($x_1 = 2,5$, $x_2 = -4$, $x_3 = 1$).

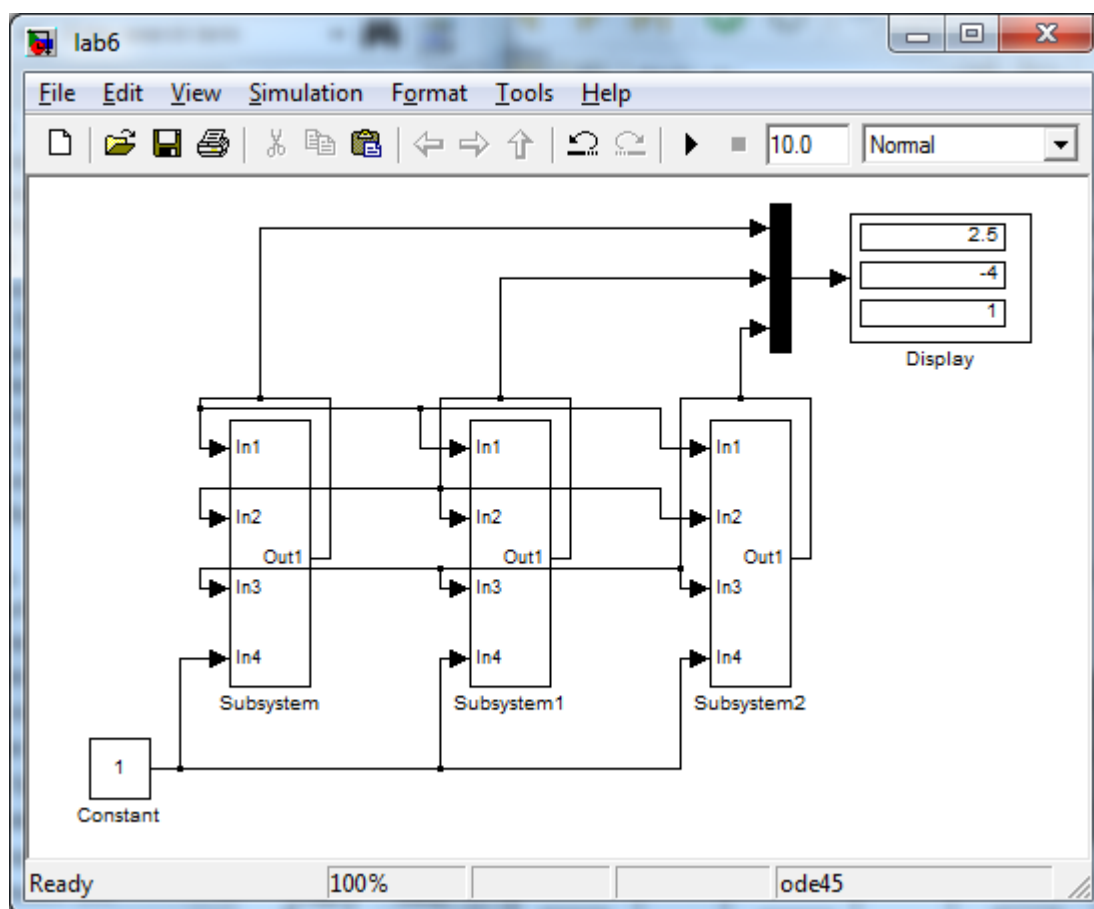


Рисунок 6.1. Структурная схема α -аналоговой модели системы линейных алгебраических уравнений 3 порядка [2]

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае

обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 6.1. Синтезировать структурную схему модели системы линейных алгебраических уравнений используя схемы линейных преобразователей синтезированных в лабораторной работе 5.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. К решению какой математической модели сводится решение оптимизационной задачи?

2. Приведите пример оптимизационной задачи из жизни. Составьте систему уравнений для ее решения.

ЛАБОРАТОРНАЯ РАБОТА 7

Разработка структуры квазианалоговых моделей (2 ч)

Цель и содержание: изучить методику разработки структур квазианалоговых моделей, приобрести навыки построения таких моделей.

Теоретическое обоснование

Виртуальный аналог операционного усилителя

В библиотеках приложений Simulink и Sim Power System, к сожалению, нет виртуального аналога операционного усилителя, без которого нельзя сконструировать обратимые квазианалоговые модели. Такой виртуальный аналог можно построить, используя элемент алгебраического ограничения приложения Simulink и виртуальные измеритель напряжения и управляемый источник напряжения приложения Sim Power System. Структурная схема такого виртуального операционного усилителя приведена на рисунке 7.1. Измеритель напряжения и управляемый источник напряжения фактически являются согласующими элементами, обеспечивающими совместимость решающих блоков Simulink в среде Sim Power System. При включении структурной схемы рисунка 7.1 с соответствующими цепями обратной связи на выходном полюсе Conn2 формируется такой сигнал, при котором сигнал на входном полюсе Conn1 равен нулю. Структурная схема рисунка 7.1 будет свернута в подсхему, используемую в последующих структурах обратимых квазианалоговых моделей. Рассмотрим структуру обратимого квазианалогового преобразователя p - типа.

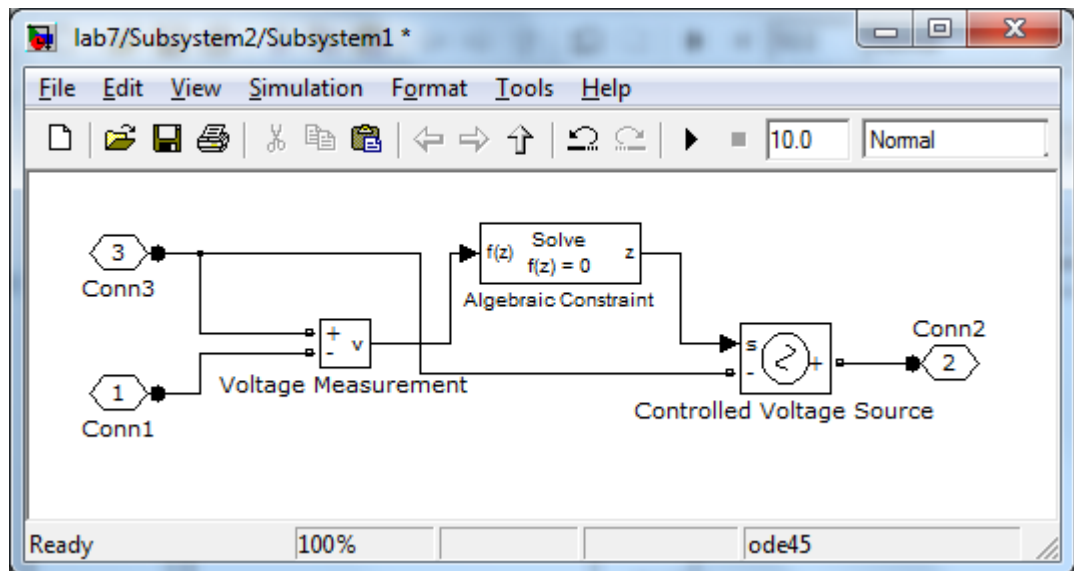


Рисунок 7.1. Виртуальный аналог операционного усилителя

Обратимый линейный преобразователь

Структурар-аналогового обратимого линейного преобразователя приведена на рисунке 7.3. Подсистема Subsystem1 является виртуальным аналогом операционного усилителя, рассмотренного выше.[2]

Резистор в данной схеме представлен блоком «Series RLC Branch» из библиотеки приложения Sim Power System с типом «R». Настройка параметров данного блока представлена на рисунке 7.2

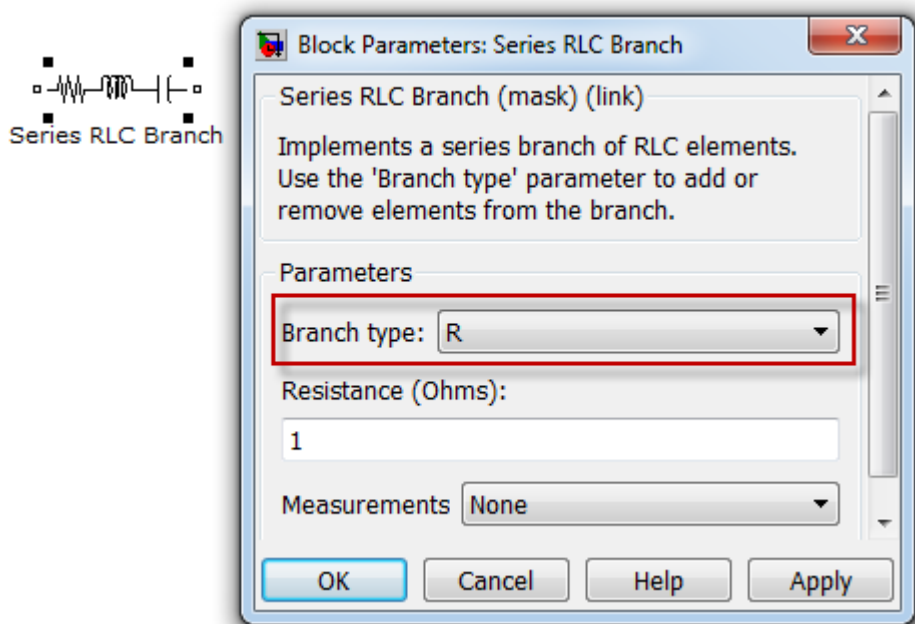


Рисунок 7.1. Окно параметров блока «SeriesRLCBranch»

Полюс Conn1 должен быть заземлен, полюса Conn2-Conn5 соответствуют сигналам, ограниченным соотношением:

$$a_2 x_2 + a_3 x_3 + a_5 x_5 = 0.$$

Коэффициенты указанного соотношения устанавливаются заданием проводимостей верхнего ряда резисторов схемы.

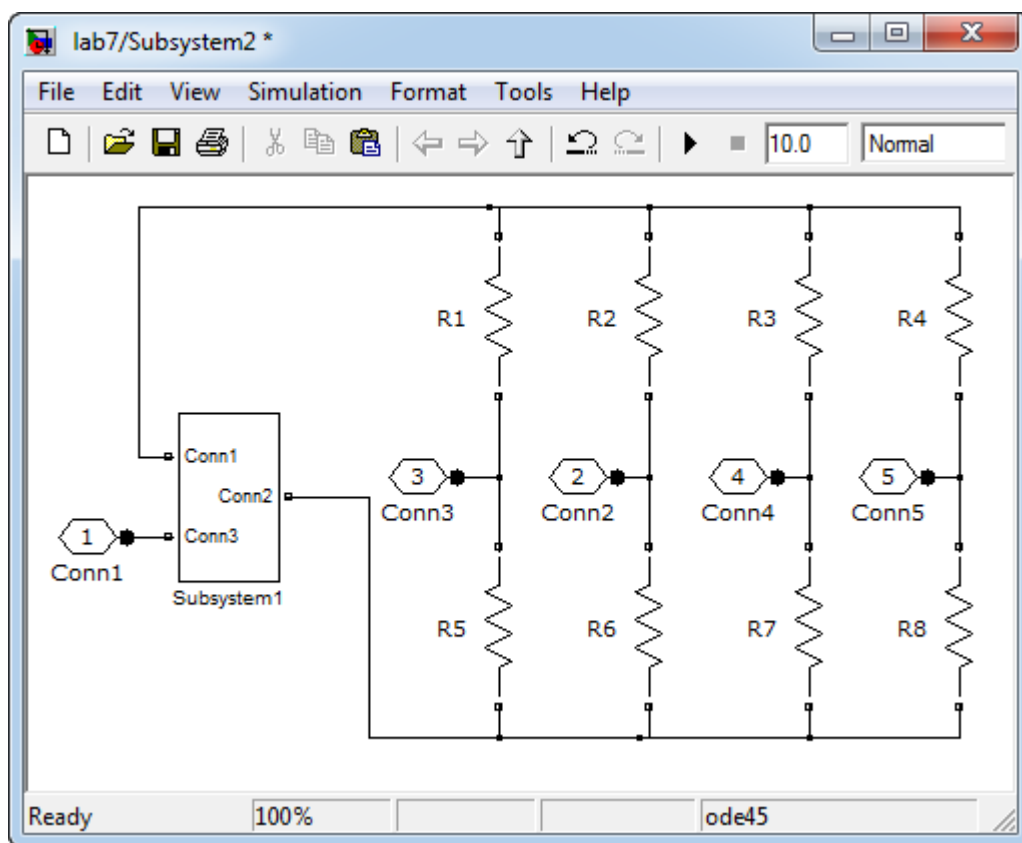


Рисунок 7.2. Структурная схема обратимого линейного преобразователя [2]

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных

компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 7.1. Построить модель виртуального аналога операционного усилителя.

Задание 7.2. Построить модель обратимого линейного преобразователя.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Назовите основные подсистемы квазианалоговых моделей.
2. Какие блоки в модели виртуального аналога операционного усилителя обеспечивают совместимость решающих блоков Simulink в среде SimPowerSystem.
3. Какими блоками библиотеки SimPowerSystem задаются коэффициенты при переменных. Опишите механизм определения данных коэффициентов.

ЛАБОРАТОРНАЯ РАБОТА 8

Моделирование систем, описываемых системами линейных алгебраических уравнений или неравенств(2 ч)

Цель и содержание: изучить методику разработки моделей системы линейных алгебраических уравнений и системы линейных алгебраических неравенств. Приобрести навыки построения таких моделей.

Теоретическое обоснование

Рассматривая структурную схему обратимого линейного преобразователя как модель линейного алгебраического уравнения, можно построить модель системы линейных алгебраических уравнений путем объединения нескольких обратимых линейных преобразователей.

На рисунке 8.1 представлена модель системы линейных алгебраических уравнений 3 порядка. Параметры обратимых линейных преобразователей(ОЛП), обозначенные на рисунке 8.1, как Subsystem1, Subsystem2, Subsystem3, устанавливаются пропорциональными коэффициентам уравнений и их правым частям. Одноименные полюса обратимых линейных преобразователей объединяются, чтобы обеспечить совместность ограничений, накладываемых каждым ОЛП. Полюса Conn1 заземляются, а виртуальный источник э.д.с. подключается к объединенному полюсу Conn5. Виртуальные вольтметры, мультиплексор и цифровой регистратор обеспечивают измерение и вывод значений неизвестных системы уравнений. В данном примере решалась следующая система уравнений:

$$\begin{aligned}2x_1 + 5x_2 + x_3 &= 11, \\4x_1 + x_2 + 3x_3 &= 7, \\x_1 + x_2 + x_3 &= 4\end{aligned}$$

Неизвестные системы уравнений, указанные на цифровом регистраторе, равны: $x_1 = -1, x_2 = 2, x_3 = 3$.

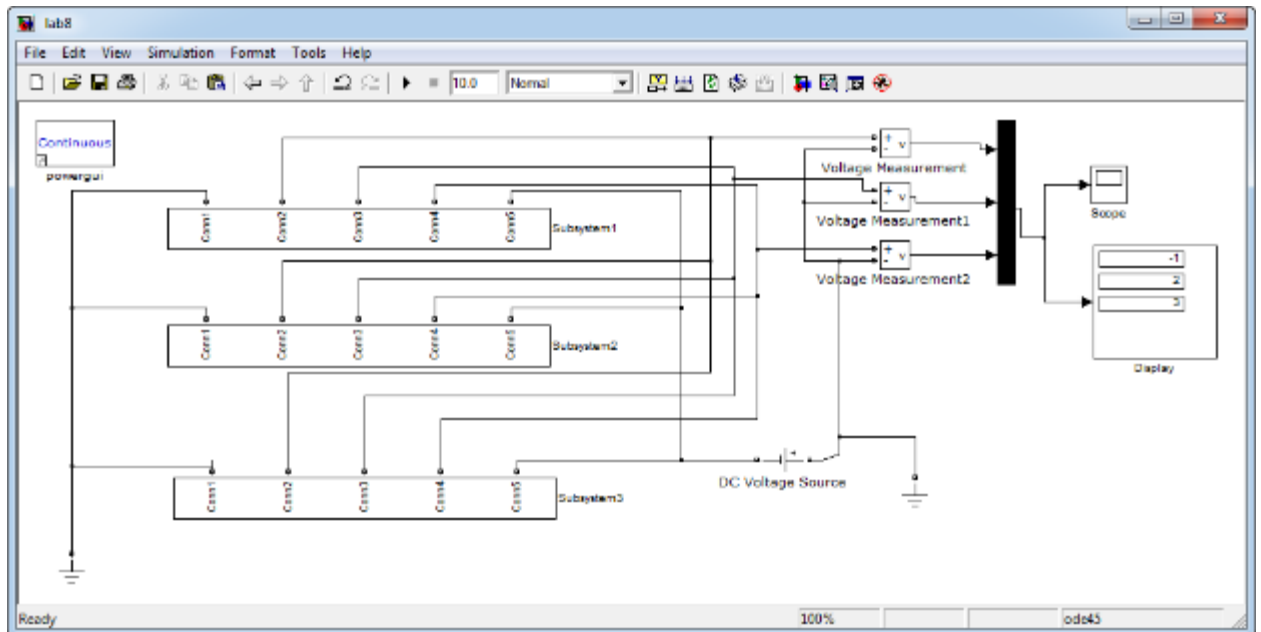


Рисунок 8.1. Структурная схема обратимой модели системы линейных алгебраических уравнений

Рассмотренные выше структуры обратимых линейных преобразователей могут быть положены в основу построения обратимой модели системы линейных неравенств. Ограничимся случаем системы трех линейных неравенств, зависящих от двух переменных вида:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &\leq b_1, \\ a_{21}x_1 + a_{22}x_2 &\leq b_2, \\ a_{31}x_1 + a_{32}x_2 &\geq b_3. \end{aligned} \quad (8.1)$$

Известно, что любая система линейных неравенств может быть сведена к эквивалентной системе линейных уравнений путем введения дополнительных переменных, на которые накладываются условия неотрицательности (не положительности). Преобразования приведенной системы неравенств приводят к следующим выражениям:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + x_3 &= b_1, \\ a_{21}x_1 + a_{22}x_2 + x_4 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + x_5 &= b_3, \\ x_3 &\geq 0, x_4 \geq 0, x_5 \leq 0. \end{aligned} \quad (8.2)$$

Пример 8.1. Структурная схема обратимой модели системы линейных неравенств, приведенная на рисунке 8.2, содержит три подсистемы обратимых линейных преобразователей (Subsystem, Subsystem1, Subsystem2), объединенных одноименными полюсами (Conn1, Conn2, Conn3, Conn5), которые реализуют модель системы трех линейных алгебраических уравнений с 5 переменными. Полюса подсистем Conn5 подключены к виртуальным диодам (Diode1, Diode2, Diode4), обеспечивающим не отрицательность дополнительных переменных x_3, x_4 и неположительность x_5 .

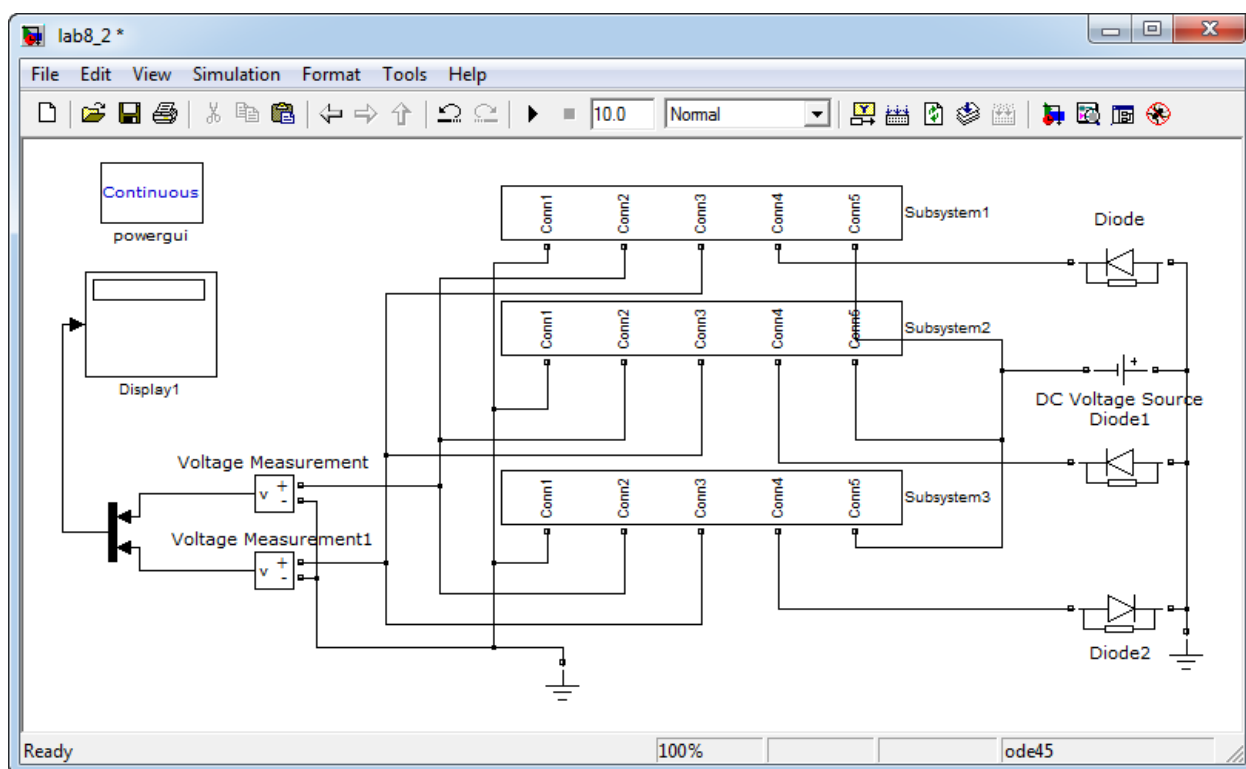


Рисунок 8.2. Структурная схема обратимой модели системы линейных неравенств [2]

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 8.1. Построить обратимую модель системы линейных алгебраических уравнений вида:

$$\begin{aligned} Ax_1 + Bx_2 + Cx_3 &= D \\ Lx_1 + Mx_2 + Nx_3 &= O \\ Qx_1 + Rx_2 + Sx_3 &= T \end{aligned}$$

Параметры сигналов приведены в таблице 8.1. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Таблица 8.1. Параметры сигналов

№ вар-та	A	B	C	D	L	M	N	O	Q	R	S	T
1.	5	2	1	2.5	3	7	2	-1.5	4	0.5	5	11.5
2.	2	1.5	3.5	-16.25	2	3	4	-19.5	2	-0.5	5	-22.75
3.	0.5	-8	-9	-30.5	4	5	6	25.5	-1.5	2	3	8.5
4.	-1	2.5	9	-35.25	5	6	7	-22	1	2	-2.5	10
5.	3	-2	-7	-8.5	6	7	8	38	1	-3.5	6	32
6.	4.5	5.5	4	-3.25	7	8	9	-12	-4.5	5	1	38
7.	6.5	-5	-6	39	1	2	3	10.5	4	1	-5.5	7.25
8.	5	7.5	9.5	27.75	2	3	4	-4.5	1	-6.5	9	11.75
9.	-4	10	-3	-67.5	3	4	5	15.5	-7.5	8	1	-90.5
10.	8	1	6	64.5	4	5	6	22.5	7	1	-8.5	70

Задание 8.2. Построить обратимую модель системы линейных алгебраических неравенств вида:

$$a_{11}x_1 + a_{12}x_2 \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 \leq b_2,$$

$$a_{31}x_1 + a_{32}x_2 \geq b_3.$$

Параметры сигналов приведены в таблице 8.2. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Таблица 8.2. Параметры сигналов

№ вар-та	a_{11}	a_{12}	a_{21}	a_{22}	a_{31}	a_{32}	b_1	b_2	b_3
1.	5	2	1	2.5	3	7	26	4	41
2.	2	1.5	3.5	-2	2	3	14	1	20
3.	6	2	1	2.5	3	7	29	4	41
4.	1	1.5	3.5	-2	2	3	11	1	20
3.	6	3	1	2.5	3	7	34	4	41
4.	1	0.5	3.5	-2	2	3	6	1	20
7.	6	3	2	2.5	3	7	34	7	41
8.	1	0.5	2.5	-2	2	3	6	-2	20
9.	6	3	2	3.5	3	7	34	12	41
10.	1	0.5	2.5	-3	2	3	6	-7	20

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Перечислите основные структурные подсистемы и блоки обратимой модели системы линейных алгебраических уравнений.
2. Перечислите основные структурные подсистемы и блоки обратимой модели системы линейных алгебраических неравенств.

3. Как в обратной модели системы линейных алгебраических уравнений задаются коэффициенты при переменных?

4. Как в обратной модели системы линейных алгебраических неравенств задаются коэффициенты при переменных?

ЛАБОРАТОРНАЯ РАБОТА 9

Построение моделей задач линейного программирования с помощью функциональных блоков Simulink и SimPowerSystem(2 ч)

Цель и содержание: изучить методику разработки моделей задач линейного программирования. Приобрести навыки построения таких моделей.

Теоретическое обоснование

Рассмотрим следующую задачу линейного программирования: максимизировать целевую функцию $\mu = c_1 x_1 + c_2 x_2$, зависящую от переменных x_1, x_2 , удовлетворяющих следующей системе линейных неравенств:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &\leq b_1, \\ a_{21}x_1 + a_{22}x_2 &\leq b_2, \\ x_1 &\geq 0, x_2 \geq 0. \end{aligned} \quad (9.1)$$

Структурная схема обратимой модели задачи линейного программирования изображена на рисунке 9.1. Она содержит три обратимых линейных преобразователя, два из которых вместе с виртуальными диодами реализуют первые два неравенства задачи линейного программирования. Два последних неравенства (не отрицательности переменных) реализуются еще двумя виртуальными диодами (Diode2, Diode3), подключенными к полюсам ОЛП Conn2, Conn3. Максимизация целевой функции обеспечивается с помощью виртуального источника тока, управляемого сигналом константы.

Параметры структурной схемы, которые задавались в данном примере, соответствовали следующим значениям коэффициентов задачи линейного программирования: $a_{11} = 3, a_{12} = 2, a_{21} = 1, a_{22} = 4, b_1 = 6, b_2 = 4, c_1 = c_2 = 1$. Значения переменных, соответствующих оптимальному решению задачи линейного программирования ($x_1 = 1.6, x_2 = 0.6, -\mu = -2.2$), указаны на цифровом регистраторе. Для решения задачи минимизации целевой функции при тех же ограничениях достаточно изменить знак

константы в параметрах источника постоянного сигнала (Constant), подключенного к управляющему входу виртуального источника тока (Controlled Current Source).

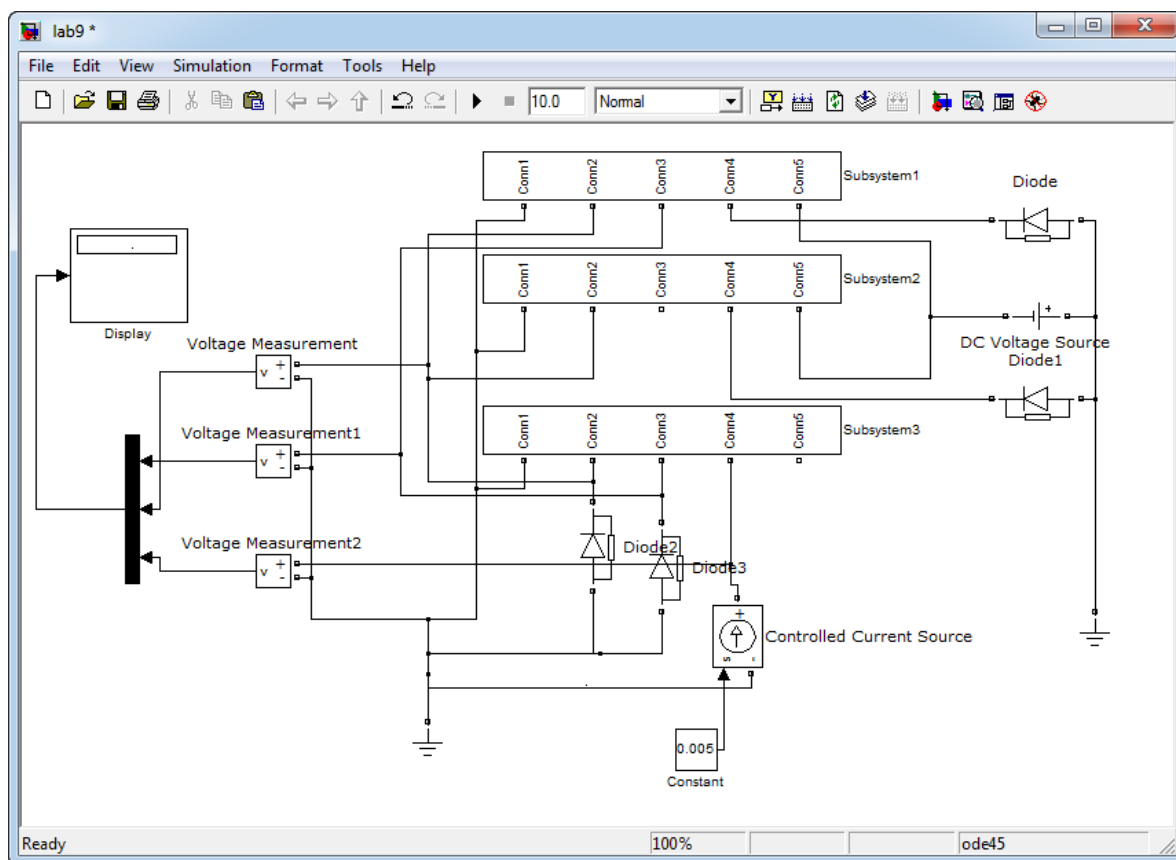


Рисунок 9.1. Структурная схема обратимой модели задачи линейного программирования [2]

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае

обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 9.1. Решить задачу линейного программирования: максимизировать целевую функцию $\mu = c_1x_1 + c_2x_2$, зависящую от переменных x_1, x_2 , удовлетворяющих следующей системе линейных неравенств:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &\leq b_1, \\ a_{21}x_1 + a_{22}x_2 &\leq b_2, \\ x_1 &\geq 0, x_2 \geq 0. \end{aligned} \quad (9.2)$$

$c_1 = c_2 = 1$. Параметры сигналов приведены в таблице 9.1. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Таблица 9.1. Параметры сигналов

№ вар-та	a_{11}	a_{12}	a_{21}	a_{22}	b_1	b_2
1.	5	2	1	2.5	26	4
2.	2	1.5	3.5	-2	14	1
3.	6	2	1	2.5	29	4
4.	1	1.5	3.5	-2	11	1
3.	6	3	1	2.5	34	4
4.	1	0.5	3.5	-2	6	1
7.	6	3	2	2.5	34	7
8.	1	0.5	2.5	-2	6	-2
9.	6	3	2	3.5	34	12
10.	1	0.5	2.5	-3	6	-7

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Перечислите основные структурные подсистемы и блоки обратной модели задач линейного программирования.
2. Как в обратной модели задач линейного программирования задаются коэффициенты при переменных?

ЛАБОРАТОРНАЯ РАБОТА 10

Построение моделей транспортной задачи средствами Simulink(2 ч)

Цель и содержание: изучить методику разработки моделей транспортной задачи. Приобрести навыки построения такой модели.

Теоретическое обоснование

Частным, но весьма распространенным видом задач линейного программирования, является транспортная задача. Классическая транспортная задача линейного программирования предусматривает нахождение оптимального плана перевозки однородного продукта в транспортной сети между пунктами производства и потребления продукта, при котором минимизируется стоимость транспортных издержек на перевозку. Математическая модель транспортной задачи имеет следующий вид:

Пусть имеются m пунктов производства однородного продукта с объемами производства $a_i, i := 1, \dots, m$ и n пунктов потребления этого продукта с объемами потребления $b_j, j := 1, \dots, n$. Пункты производства связаны пунктами потребления транспортной сетью, образующей, в общем случае $m \times n$ маршрутов с заданными стоимостями перевозки единицы продукта c_{ij} по маршруту $i - j$. Если обозначить неизвестные объемы перевозок по маршруту $i - j$ как x_{ij} , математическая модель транспортной задачи будет иметь вид:

$$\sum_{j=1}^n x_{ij} = a_i, i := 1, \dots, m, \quad (10.1)$$

$$\sum_{i=1}^m x_{ij} = b_j, j := 1, \dots, n, \quad (10.2)$$

$$x_{ij} \geq 0, \quad (10.3)$$

$$\mu = \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \rightarrow \min. \quad (10.4)$$

Выражение (10.1) является условием баланса объемов производства и перевозок продукта. Аналогично, выражение (10.2) является условием

баланса объемов перевозок и потребления. Условие (10.3) требует не отрицательности перевозок (нецелесообразность перевозок из пунктов потребления в пункты производства). Общие транспортные издержки (стоимость перевозки) определяется выражением (10.4) и должна минимизироваться выбором объемов перевозок по различным маршрутам.

Известна аналоговая электрическая модель транспортной задачи линейного программирования в виде цепи постоянного тока, содержащей идеальные источники напряжения, тока и диоды. Топология цепи повторяет топологию транспортной сети, моделями маршрутов являются ветви цепи из последовательно соединенных диодов и источников напряжения, источники тока используются для моделирования пунктов производства и потребления продукта. Уравнения (10.1) и (10.2) воспроизводятся как уравнения первого закона Кирхгофа электрической цепи-модели, условия не отрицательности (10.3) реализуются односторонней проводимостью диодов. В цепи устанавливается такое распределение токов, моделирующих объемы перевозок, при котором минимальна мощность, рассеиваемая в модели транспортной сети. Таким образом реализуется минимум выражения (10.4) и получается оптимальное решение транспортной задачи.

Для построения виртуальной модели транспортной задачи в среде визуального программирования необходимо сконструировать структурные схемы источников тока, моделирующих пункты производства и потребления продукта, а также структурную схему модели ветви. Далее, можно используя эти структурные схемы в качестве подсистем, синтезировать структурную схему виртуальной модели транспортной задачи в целом.

Структурная схема виртуальной модели пункта производства показана на рисунке 10.1

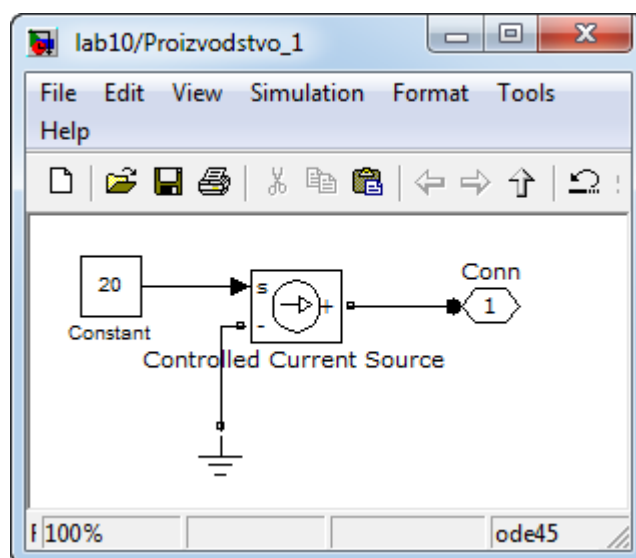


Рисунок 10.1. Модель пункта производства

Схема состоит из управляемого источника тока (Controlled Current Source) приложения Sim Power Systems, значение тока которого задается блоком константы (Constant) приложения Simulink. Отрицательный полюс источника тока заземлен, а положительный подключен к выходному полюсу (Connection Port). Значение константы, устанавливаемое в окне параметров блока Constant, определяется величиной объема производства пункта производства, моделью которого является Proizvodstvo_1.

Структурная схема виртуальной модели пункта потребления строится аналогично и приведена на рисунке 10.2.

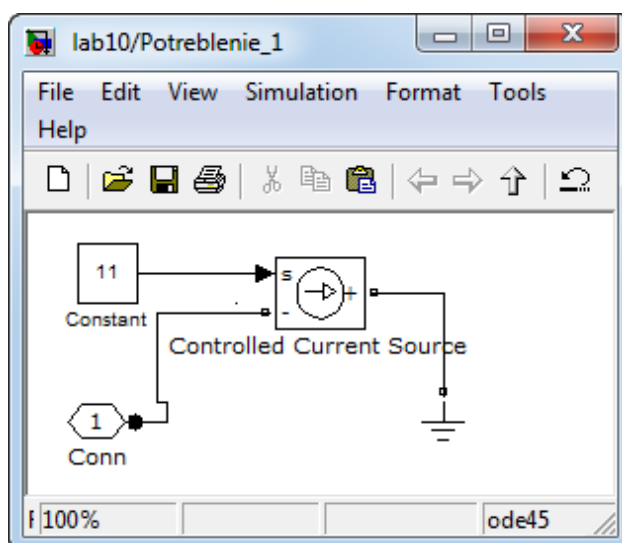


Рисунок 10.2. Модель пункта потребления

Структурная схема виртуальной модели ветви транспортной сети приведена на рисунке 10.3.

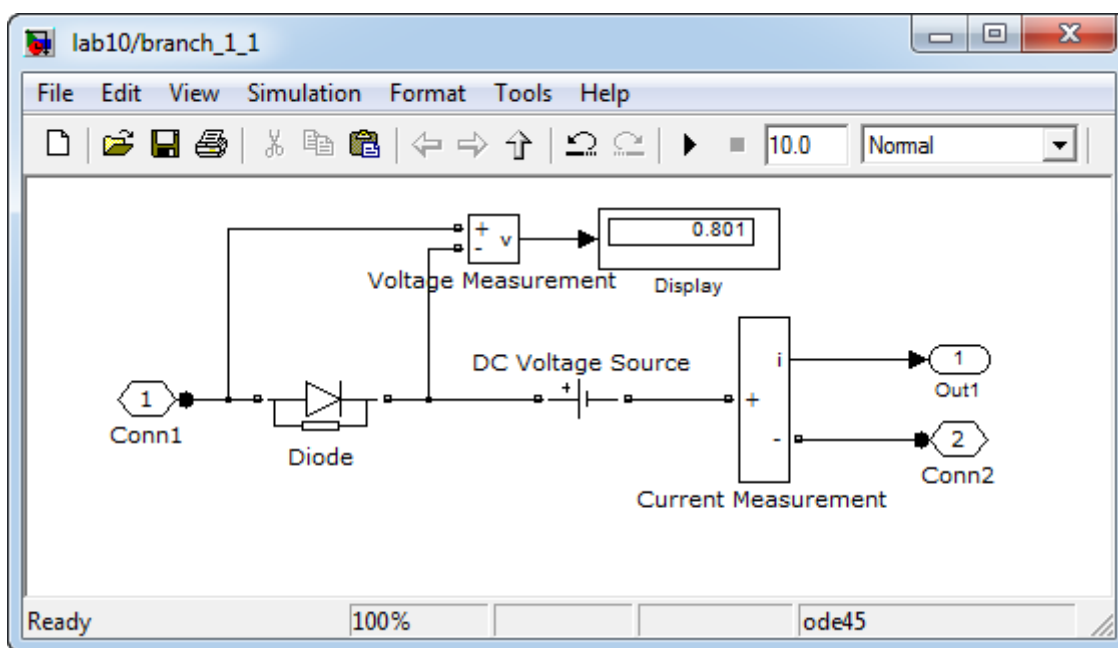


Рисунок 10.3. Модель ветви транспортной сети

Основу модели ветви составляют последовательное соединение виртуального диода (Diode) и виртуального источника электродвижущей силы постоянного тока (DC Voltage Source). Анод виртуального диода соединен с входным полюсом виртуальной модели ветви (Connection Port1), а отрицательный полюс виртуального источника э.д.с. через виртуальный измеритель тока (Current Measurement) соединен с выходным полюсом модели ветви (Connection Port2). Выходной полюс Out1 виртуального измерителя тока формирует сигнал стандарта Simulink и может подключаться к цифровым регистраторам этого приложения не посредственно или через мультиплексор Mux. Виртуальный вольтметр, образованный виртуальным измерителем напряжения Voltage Measurement и цифровым регистратором Display, не являются обязательными элементами модели ветви и предназначены для оценивания состояния виртуального диода (открытого или закрытого). Порт измерений для виртуального диода

можно отключить снятием галочки «Show Measurement port» в параметрах (Рисунок 10.4).

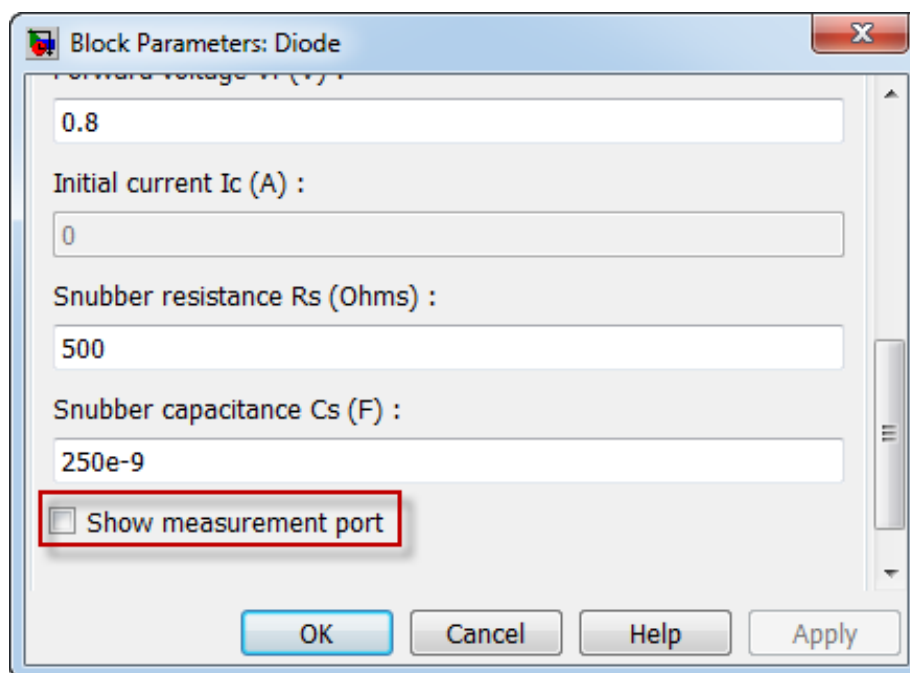


Рисунок 10.4. Отключение порта для вывода измерений

Входные полюса моделей ветвей транспортной сети (Connection Port1) подключаются к выходным полюсам (Connection Port1) моделей пунктов производства, а выходные полюса моделей ветвей транспортной сети (Connection Port2) подключаются к входным полюсам (Connection Port2) моделей пунктов потребления. Порядок построения структурной схемы виртуальной модели транспортной задачи рассмотрим на примере.

Пример 10.1. Дана следующая транспортная задача линейного программирования. Найти оптимальный вариант перевозки цемента, производимого двумя цементными заводами на три стройки, расположенными в том же районе. Производительности цементных заводов составляют соответственно 200 и 100 тонн в сутки. Потребности строящихся объектов составляют 110, 140 и 50 тонн в сутки. Стоимости перевозки 1 тонны цемента для возможных 6 маршрутов в условных единицах, приведена в таблице 10.1.

Таблица 10.1. Стоимость перевозки

	Об-кт № 1	Об-кт № 2	Об-кт № 3
З-д № 1	30	10	30
З-д № 2	30	50	60

На рисунке 10.5 приведена структурная схема виртуальной модели транспортной задачи, построенная из элементов, рассмотренных выше (рисунки 10.1 – 10.2). Виртуальные модели заводов №1 и №2 представлены элементами Proizvodstvo_1 и Proizvodstvo_2. Виртуальные модели строительных объектов №1, №3, представлены элементами Potreblenie_1 и Potreblenie_3. Так как задача является сбалансированной (суммарный объем производства цемента равен суммарной потребности строительных объектов), модели одного из объектов воспроизводить не требуется. В нашем случае выбран объект №2. Модели различных маршрутов представлены на схеме следующими подсистемами:

Для завода № 1 – branch_1_1, branch_1_2 и branch_1_3,

Для завода № 2 – branch_2_1, branch_2_2 и branch_2_3.

Виртуальные токи, изображающие объемы перевозок в оптимальном варианте, оцениваются цифровым регистратором Display, на который поступают сигналы от виртуальных моделей ветвей через мультиплексор.

Значения объемов производства и потребления продукта задавались в окнах параметров блоков Constant виртуальных моделей пунктов производства и потребления продукта (подсистемы Proizvodstvo_x и potreblenie_y) в масштабе 0.1. Величины стоимости перевозки единицы продукта по различным маршрутам задавались в окнах параметров блоков DCVoltage Source виртуальных моделей ветвей транспортной сети (подсистемы branch_x_y) также в масштабе 0.1. Результаты моделирования, приведенные на экране цифрового регистратора, имеют вид:

$$x_{11} = 1 \quad x_{12} = 14 \quad x_{13} = 5$$

$$x_{21} = 10 \quad x_{22} \approx 0 \quad x_{23} \approx 0$$

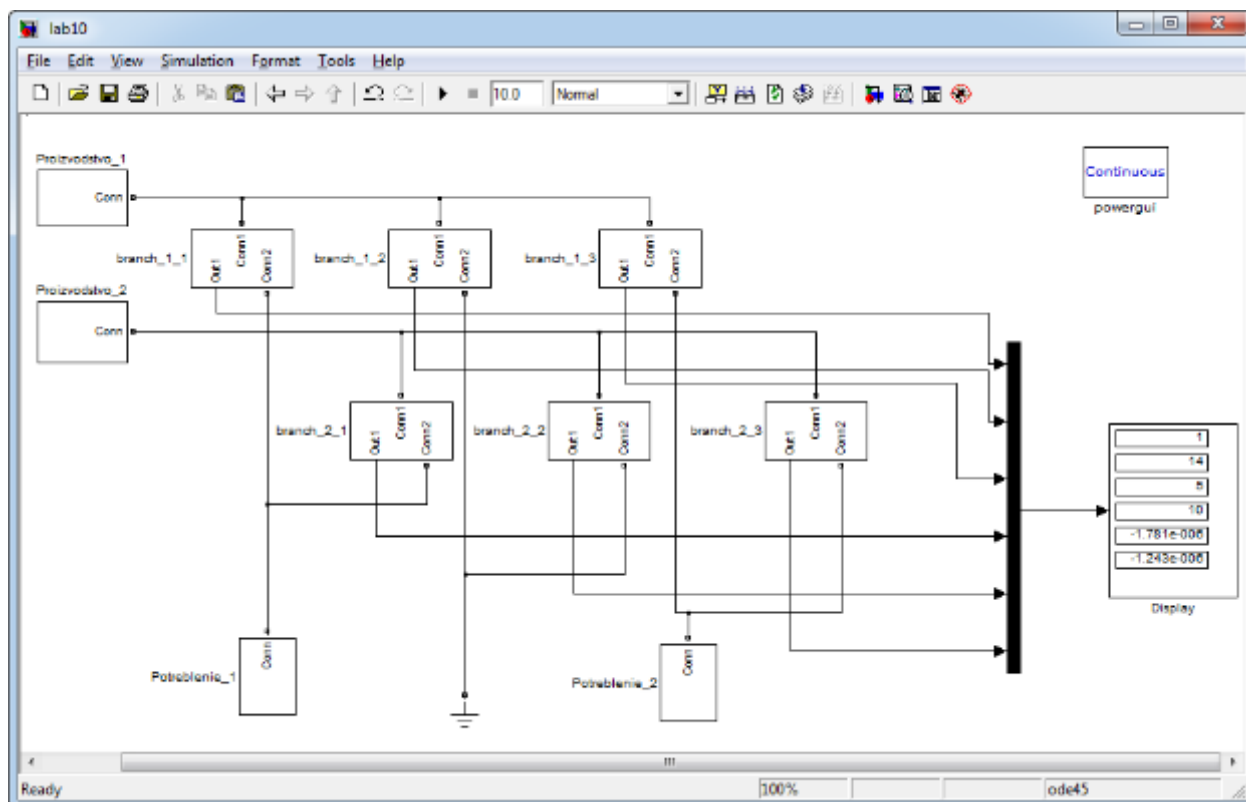


Рисунок 10.5. Структурная схема модели транспортной задачи[2]

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 10.1. Найти оптимальный вариант перевозки молока, производимого двумя филиалами молочного комбината в три детских сада. Производительности филиалов составляют соответственно А и В литров в сутки. Потребности детских садов составляют X, Y и Z литров в сутки. Стоимости перевозки 1 литра молока для возможных 6 маршрутов в условных единицах составляют соответственно $A_x, A_y, A_z, B_x, B_y, B_z$.

Параметры задачи приведены в таблице 10.2. Номер варианта соответствует номеру, под которым студент записан в списке группы.

Таблица 10.2. Параметры задачи 10.1

№	A	B	X	Y	Z	A_x	A_y	A_z	B_x	B_y	B_z
1.	100	190	20	180	90	1	2	3	2	3	1
2.	110	240	100	100	150	2	1	2	3	2	3
3.	120	300	140	70	210	1	2	1	2	3	2
4.	140	210	85	85	180	2	1	2	1	2	3
5.	115	90	45	100	60	1	2	1	2	1	2
6.	135	70	75	70	60	2	1	2	2	3	1
7.	145	85	80	50	100	1	2	2	3	1	3
8.	140	180	105	85	130	2	2	3	1	3	1
9.	90	315	200	115	90	2	3	1	3	2	1
10.	70	200	100	140	30	3	1	3	1	1	2

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Опишите математическую модель транспортной задачи.
2. Из каких подсистем состоит аналоговая электрическая модель транспортной задачи?
3. Какие результаты ожидаются в ходе решения классической транспортной задачи?

ЛАБОРАТОРНАЯ РАБОТА 11

Спектральный анализ и синтез сигналов средствами MATLAB (2 ч)

Цель и содержание: практическое изучение методов спектрального анализа и синтеза сигналов средствами MATLAB

Теоретическое обоснование

Основная задача математического моделирования процесса прохождения сигналов через линейные искажающие (или преобразующие) их устройства (фильтры, усилители, каналы связи и т. д.) может решаться различными методами. При спектральном подходе к ее решению она может быть представлена следующей диаграммой:

$$y_i(t) \rightarrow \text{ППФ} \rightarrow A_i(\omega), \varphi_i(\omega) \rightarrow \text{ИУ} \rightarrow A_o(\omega), \varphi_o(\omega) \rightarrow \rightarrow \text{ОПФ} \rightarrow y_o(t) \quad (11.1)$$

Исходный входной сигнал, представленный произвольной временной зависимостью $y_i(t)$ с помощью прямого преобразования Фурье (ППФ, или DFT), преобразуется в свое частотное представление. Если сигнал периодический, то он представляется суммой дискретных гармоник с амплитудами $A_i(\omega)$ и фазами $\varphi_i(\omega)$ где i - номер гармоники от 0 (постоянная составляющая) до N , и $\omega=2\pi f$ - круговая частота (f - частота в герцах). Первая гармоника имеет частоту повторения сигнала ω_1 , а высшие - частоты, кратные основной гармонике $i\omega_1$. Для непериодических сигналов спектр будет сплошным, понятие гармоник уже не существует, но можно говорить о плотности мощности сигнала в определенной полосе частот.

В общем виде ППФ записывается следующим образом:

$$A(\omega) = \int_{-\infty}^{\infty} y(t)e^{-i\omega t} dt \quad (11.2)$$

В результате ППФ мы получаем комплексную амплитуду на заданной круговой частоте ω . После тривиальных преобразований можно получить модуль $A(\omega)$ - частотный спектр сигнала $A_i(\omega)$ и фазу $\varphi_i(\omega)$ - фазовый спектр сигнала.

Моделирование сигналов – одна из важнейших задач пакета Signal Processing Toolbox. Многие сигналы являются детерминированными, то есть их временная зависимость $y(t)$ аналитически определена и для любого момента времени t при неоднократном моделировании мы будем получать одни и те же значения $y(t)$. Детерминированные сигналы удобны для аналитического описания и анализа систем обработки и преобразования сигналов и широко используются в роли тестовых сигналов.

Однако на практике большинство реальных сигналов не являются детерминированными. Одной из причин этого является засоренность сигналов шумами и помехами, имеющими случайный характер с тем или иным законом распределения вероятности.

Моделирование зашумленных сигналов в пакете SignalProcessingToolbox обеспечивается с помощью генератора случайных чисел. Разумным компромиссом между детерминированными и недетерминированными сигналами являются детерминированные сигналы с повторяющейся при каждом запуске моделирования шумовой компонентой. Это позволяет обеспечить, с одной стороны, учет влияния шума, а с другой — обеспечить повторяющиеся в процессе многократного моделирования результаты.

Спектральный анализ заключается в определении частотного состава сигнала и оценивании его спектральных характеристик, являющихся функциями частоты.

С помощью спектрального анализа решаются задачи обнаружения, разрешения и оценки параметров сигналов, сжатия данных, идентификации объектов, распознавания образов – речи, изображений и т.д. Спектральный анализ случайных сигналов направлен на выявление скрытых периодичностей и статистических (корреляционных) связей.

Спектральный анализ сигналов в пакете Signal Processing Toolbox реализуется большим числом методов. Все они вычисляют спектральную плотность мощности (СПМ) сигналов и имеют одинаковые (как по смыслу, так и по обозначению) параметры.

Названия функций начинаются с буквы **p** (от слова «power» - мощность).

Входные параметры функций спектрального анализа следующие:

x– вектор значений сигнала;

p– порядок модели;

nfft– число отсчетов сигнала, используемого при быстром преобразовании Фурье – БПФ (минимальное значение по умолчанию равно 256);

f– частота (в Гц);

w– круговая частота;

fs– частота дискретизации;

'range'– параметр указания диапазона частот;

'squared'– параметр, отменяющий вывод графика СПМ в децибелах и дающий график в прямоугольной системе координат.

Некоторые входные параметры могут отсутствовать. Их следует задавать как пустую матрицу [], что служит указанием для задания соответствующего параметра по умолчанию.

Выходные параметры функций этого раздела следующие:

freq– вектор частот, для которых производится оценка СПМ;

Pxx– вектор-столбец оценки СПМ, размер которого при четном **nfft** равен $nfft/2+1$, а при нечетном $(nfft+1)/2$. Если **x** содержит комплексные данные, то СПМ оценивается для всех частот, и число элементов вектора **Pxx** равно **nfft**.

Все описанные ниже функции имеют внутреннее обращение к функции **plot(f, Pxx)** и строят графики зависимости СПМ от частоты, используя векторы частот **f** и СПМ на них **Pxx**.

В пакете Signal Processing Toolbox существуют следующие методы спектрального анализа: метод Бурга (pburg), ковариационный метод (pcov), модифицированный ковариационный метод (pmscov), многооконный метод (pmtm), метод Уэлча (pwelch) и метод периодограмм (periodogram)[1].

Таблица 11.1. Сравнение основных методов спектрального анализа

Метод	Функция	Достоинства	Недостатки
Периодограмм, непараметрический	periodogram	Хорошее разрешение по частоте	Изрезанность при длинных сигналах. Флюктуации при большом числе отсчетов
Уэлча, непараметрический	pwelch	Хорошее сглаживание сигналов	Возможная потеря острых пиков спектра
Берга, на основе авторегрессии	pburg	Высокое разрешение и стабильность при анализе коротких сигналов. Хорошее устранение шума. Использование окон (весовых функций) и частично перекрывающихся сегментов сигнала. Дисперсия оценки спектра мощности меньше, чем при методе Бартлетта	Начальные фазы синусоид сильно влияют на положение спектральных пиков; появляется смещение спектральных пиков при анализе суммы синусоид с шумом; при большом порядке модели может наблюдаться расщепление спектральных пиков
Ковариационный	pcov	Высокая разрешающая способность при спектральной оценке коротких сигналов. Минимизация погрешностей прямого предсказания. Хорошая оценка сигналов в виде суммы чистых синусоид	Рассчитанный формирующий фильтр может оказаться нестабильным. При анализе суммы синусоид с шумом возможно смещение спектральных пиков
Модифицированный ковариационный	pmscov	Высокое разрешение при анализе коротких сигналов, отсутствие расщепления пиков	Зависимость положения спектральных линий от начальных фаз синусоид
Многооконный, на основе последовательностей Слепиана	pmtm	Высокое разрешение	Работа только с вещественными данными

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение:

персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 11.1. Вычислим СПМ методом Бурга.

Вычисление реализовано следующими вариантами записи функции pburg:

$P_{xx} = \text{pburg}(x, p)$

$[P_{xx}, w] = \text{pburg}(x, p)$

$[P_{xx}, w] = \text{pburg}(x, p, \text{nfft})$

$[P_{xx}, f] = \text{pburg}(x, p, \text{nfft}, fs)$

$[P_{xx}, f] = \text{pburg}(x, p, \text{nfft}, fs, 'range')$

$[P_{xx}, w] = \text{pburg}(x, p, \text{nfft}, 'range')$

Если в качестве выходного параметра указан только вектор P_{xx} , то вычисляется только СПМ. При указании в качестве выходных параметров списка $[P_{xx}, w]$ или $[P_{xx}, f]$ дополнительно вычисляются векторы частот — обычных f (в Гц) или круговых w , на которых выполнено вычисление P_{xx} . По умолчанию могут задаваться параметры $\text{nfft} = 256$, $f = 1$ Гц.

При использовании параметра `range` рекомендуется уточнять диапазон частот — f или w — по справке заданной функции. Обычно он выбирается

разным для вещественных и комплексных данных и зависит от выбора частоты дискретизации f_s .

Для иллюстрации оценки СКМ методом Бурга воспользуемся наглядным примером из справки по этой функции. Вначале построим АЧХ и ФЧХ автокорреляционной AR-системы:

```
a=[1 -2.2137 2.9403 -2.1697 0.9606]; %Коэффициенты AR-фильтра  
freqz(1,a) % АЧХ и ФЧХ AR-фильтра  
title('АЧХ AR-системы')
```

Их графики представлены на рисунке 11.1.

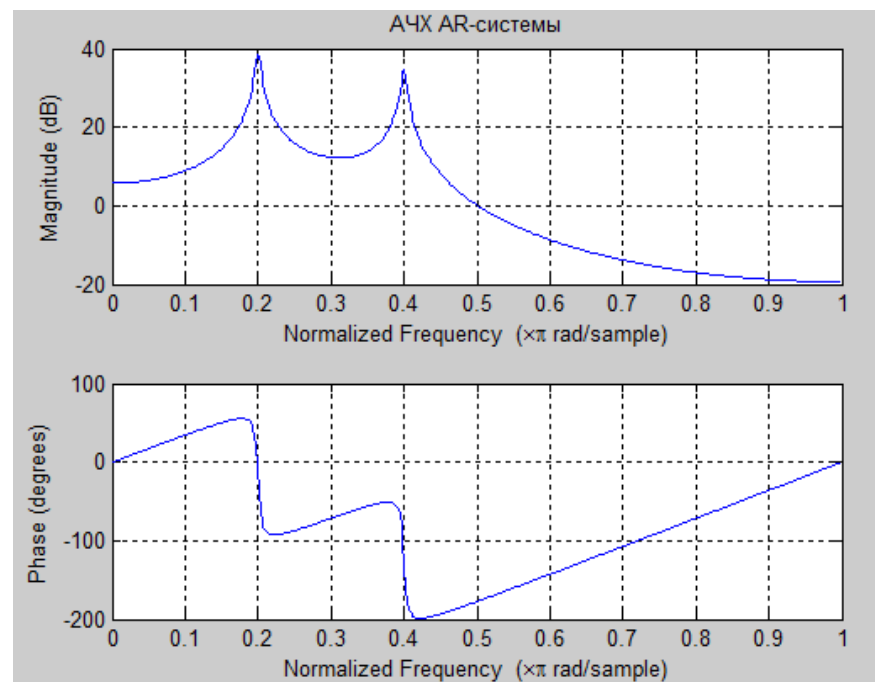


Рисунок 11.1. Графики АЧХ и ФЧХ AR-фильтра

Теперь построим АЧХ для сигнала из белого шума, используя AR-систему 4-го порядка:

```
randn('state',1);  
x=filter(1,a,randn(256,1)); % Выход AR-системы  
pburg(x,4) % Оценка СКМ 4-го порядка
```

Построенный график АЧХ для этого случая показан на рис. 11.2.

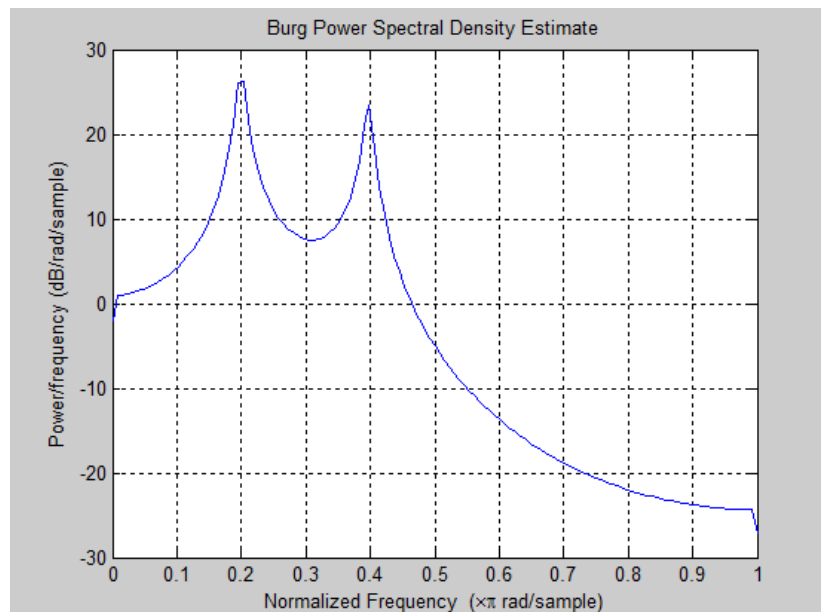


Рисунок 11.2. АЧХ белого шума, построенная с помощью функции `rburg`

Задание 11.2. Вычислим СПМ модифицированным ковариационным методом.

Ковариационный метод реализует функция `pcov`, имеющая следующие варианты записи:

`Pxx=pcov(x,p)`

`[Pxx,w]=pcov(x,p)`

`[Pxx,w]=pcov(x,p,nfft)`

`[Pxx,f]=pcov(x,p,nfft,fs)`

`[Pxx,f]=pcov(x,p,nfft,fs,'range')` `[Pxx,w]=pcov(x,p,nfft,'range')`

Назначение параметров этой функции уже описано. С ее работой можно ознакомиться, выполнив приведенный ниже пример:

```
a=[1 -2.2137 2.9403 -2.1697 0.9606];
freqz(1,a)
title('AR System Frequency Response')
randn('state',1);
x=filter(1,a,randn(256,1));
pcov(x,4)
```

Проверьте насколько графики, построенные при выполнении этого примера, отличаются от графиков предыдущего примера.

Задание 11.3. Вычислим СПМ ковариационным методом.

Модифицированный ковариационный метод реализован функцией `pmcov`:

```
Pxx=pmcov(x,p) [xx,w]=pmcov(x,p)
```

```
[Pxx,w]=pmcov(x,p,nfft)
```

```
[Pxx,f]=pmcov(x,p,nfft,fs)
```

```
[Pxx,f]=pmcov(x,p,nfft,fs,'range')
```

```
Pxx,w]=pmcov(x,p,nfft,'range')
```

Параметры и техника применения этой функции подобны уже описанным. Аналогично предыдущим примерам постройте график и сравните с уже построенными.

Задание 11.4. Вычислим СПМ многооконным методом.

Для реализации многооконного метода служит функция `pmtm`.

Варианты ее записи представлены ниже:

```
[Pxx,w]=pmtm(x,nw)
```

```
[Pxx,w]=pmtm(x,nw,nfft)
```

```
[Pxx,f]=pmtm(x,nw,nfft,fs)
```

```
[Pxx,Pxxc,f]=pmtm(x,nw,nfft,fs)
```

```
[Pxx,Pxxc,f]=pmtm(x,nw,nfft,fs,p)
```

```
[Pxx,Pxxc,f]=pmtm(x,e,v,nfft,fs,p)
```

```
[Pxx,Pxxc,f]=pmtm(x,dpss_params,nfft,fs,p)
```

```
[..]=pmtm(... 'method')
```

```
[..]=pmtm(...'range')
```

Эта функция работает с вещественными данными и использует совокупность ортогональных окон. В качестве их используются дискретные сфероидальные последовательности Слепиана. Параметр `nw` задает спектрально-временное разрешение. По умолчанию `nw = 4`, рекомендуется выбирать его значения равными 2, 5/2, 3, 7/2. Число последовательностей Слепиана равно $2*nw-1$.

Функция имеет параметр `method`, позволяющий задать метод вычисления СПМ:

adapt– адаптивный нелинейный алгоритм Томсона комбинации индивидуальных оценок (используется по умолчанию);

unity– линейная комбинация индивидуальных оценок с весами, равными 1;

eigen– линейная комбинация индивидуальных оценок с весами, задаваемыми собственными значениями.

Ниже представлен пример оценки СПМ синусоиды с частотой 250 Гц, на которую наложен небольшой шум.

```
randn('state', 0); fs=1000; t=0:1/fs:0.3;  
x=cos(2*pi*t*250)+0.05*randn(size(t));  
[Pxx, Pxxc, f]=pmtm(x, 3.5, 512, fs, 0.99);  
hpsd = dspdata.psd([Pxx Pxxc], 'Fs', fs);  
plot(hpsd)
```

Этот пример создает графики, представленные на рисунке 11.3.

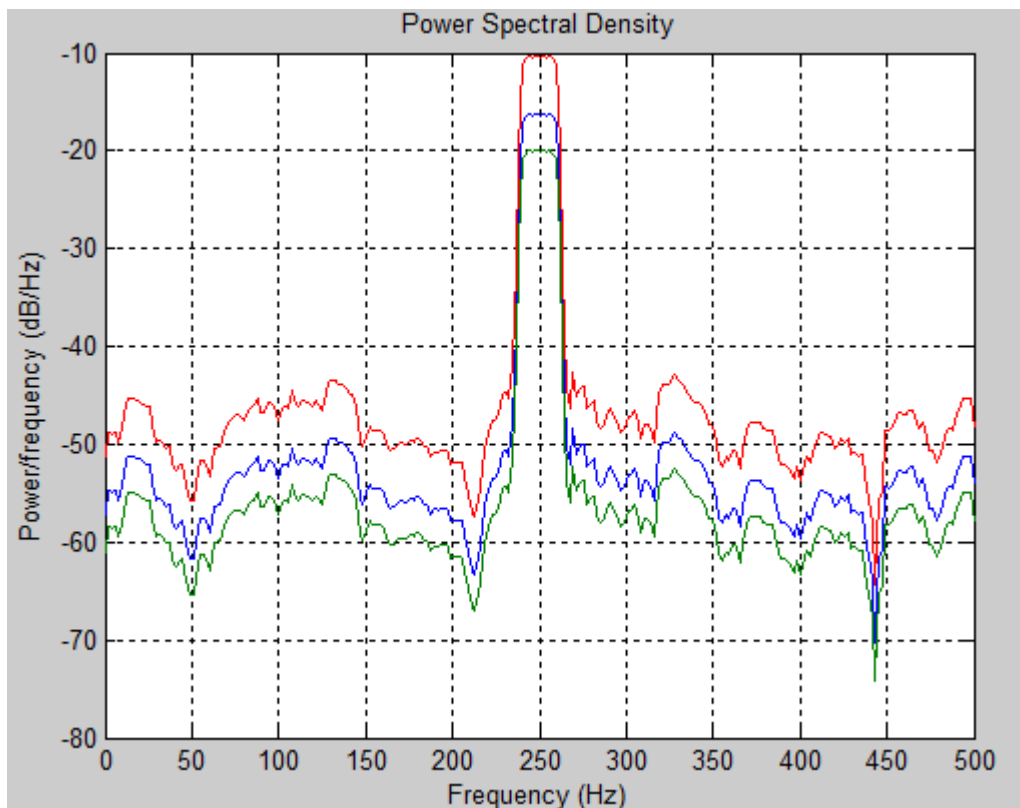


Рисунок 11.3. Пример оценки СПМ с помощью функции `pmtm`

Задание 11.5. Вычислим СПМ методом Уэлча.

Метод Уэлча вычисления СКМ реализуется следующей функцией:

```
[Pxx,w]=pwelch(x) [Pxx,w]=pwelch(x,nwin)
```

```
[Pxx,w]=pwelch(x,nwin, noverlap)
```

```
[Pxx,w]=pwelch(x,nwin,noverlap,nfft)
```

```
[Pxx,f]=pwelch(x,nwin,noverlap,nfft,fs)
```

```
[..]=pwelch(x, nwin, noverlap,...,'range')
```

Здесь целочисленное положительное значение параметра `nwin` задает длину окна Хэмминга. Если `nwin` задан как двухэлементный вектор, он задает размеры прямоугольного окна. Остальные параметры этой функции подобны уже описанным. Для расширения знакомства с этой функцией используйте ее демонстрационный пример:

```
randn('state',0); Fs=1000; t=0:1/Fs:.3;  
x=cos(2*pi*t*200)+randn(size(t));  
pwelch(x,33,32,[],Fs,'twosided')
```

Просмотрите получаемый при этом график (рисунок 11.4) и попробуйте модифицировать этот пример. Описание алгоритма реализации этого метода можно найти в справочной системе.

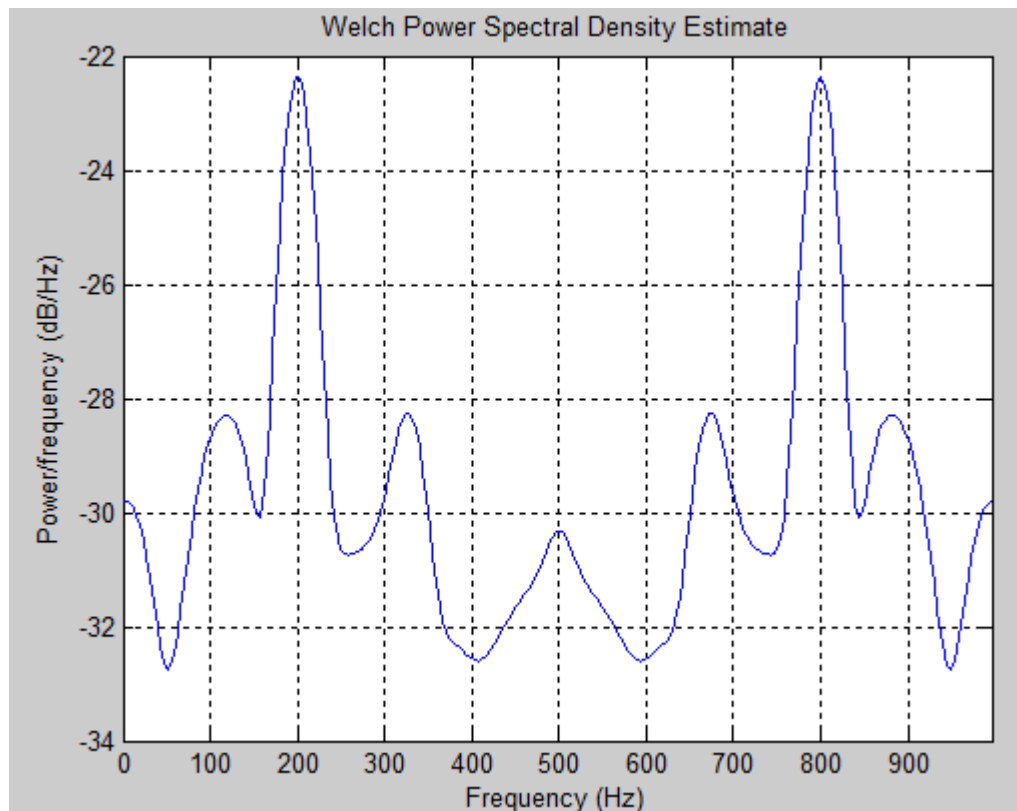


Рисунок 11.4. Пример оценки СПМ с помощью функции pwelch

Задание 11.6 Вычислим СПМ методом периодограмм.

Для построения периодограмм как зависимостей спектральной мощности в дБ/Вт от частоты служит функция `periodogram`:

```
[Pxx,w]=periodogram(x)
```

```
[Pxx,w]=periodogram(x>window)
```

```
[Pxx,w]=periodogram(x>window,nfft)
```

```
[Pxx,f]=periodogram(x>window,nfft,fs)
```

```
[Pxx,...]=periodogram(x,...,'range')
```

Частота может задаваться как угловая w или в герцах f . Параметр `range` может иметь два значения:

twosided – вычисляет двухполосную спектральную мощность в частотном диапазоне $[0,fs)$. При задании вместо этой опции пустого вектора `[]` частотный диапазон определяется как $[0,1)$. Если не используется спецификация f , то этот диапазон выбирается равным $[0,2)$.

onesided – вычисляется однополосная спектральная мощность в диапазоне частот, заданном для вещественных компонент вектора x . Для x с вещественными компонентами эта опция используется по умолчанию.

Для построения периодограмм используется функция `periodogram`, что иллюстрирует следующий пример:

```
Fs=1000; t=0:1/Fs:.3;  
x=sin(2*pi*t*200)+cos(2*pi*t*350)+0.01*randn(size(t));  
periodogram(x,[],'onesided',512, Fs)
```

В результате получим график (рисунок 11.5).

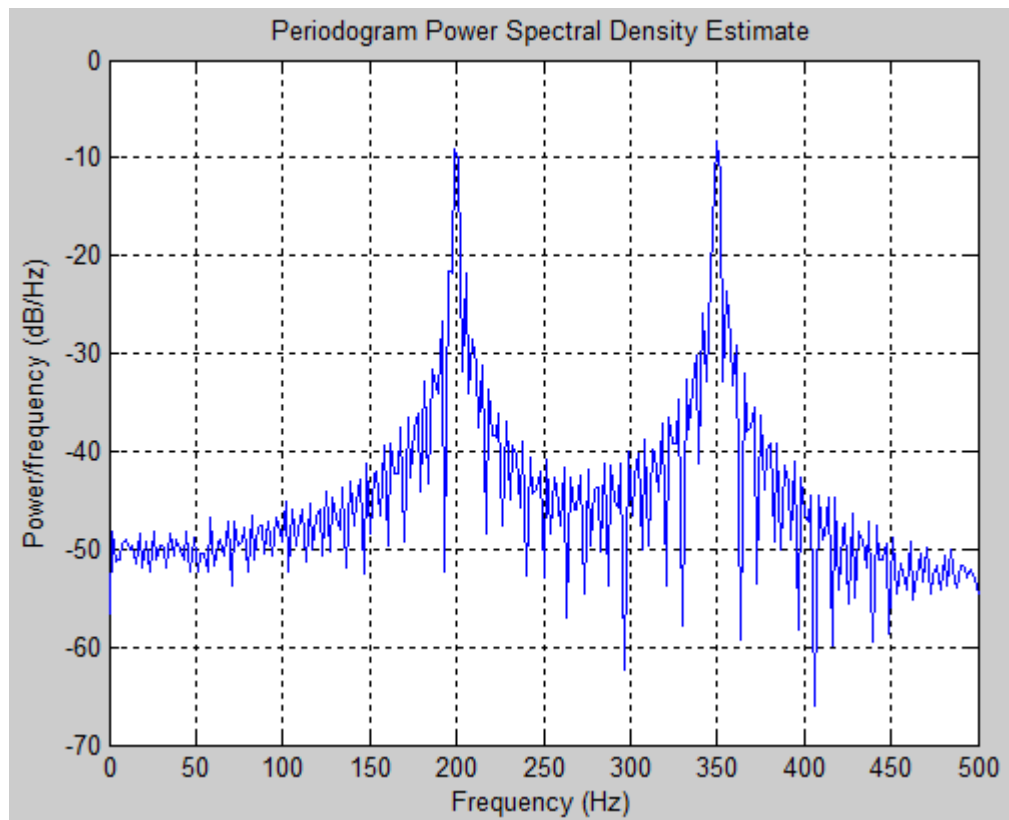


Рисунок 11.5. Пример оценки СПМ с помощью функции periodogram

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Дайте содержательную характеристику спектральному анализу
2. Какой пакет MATLAB содержит методы спектрального анализа сигналов?

3. Какие основные методы представлены в этом пакете?
4. Дайте краткую характеристику каждому из представленных методов.
5. Опишите достоинства и недостатки представленных методов.

ЛАБОРАТОРНАЯ РАБОТА 12

Вейвлет-анализ и синтез сигналов в MATLAB (2 ч)

Цель и содержание: получить основные навыки применения вейвлет-преобразования сигналов. Освоение GUI WavemenuMATLAB.

Теоретическое обоснование

Вейвлеты и их применение – новейшее научное направление, возникшее на стыке математики, информатики и техники связи. Технология вейвлетов базируется на обобщенном представлении сигналов $s(t)$ в векторном пространстве в виде взвешенной суммы базисных функций $\Psi_k(t)$, помноженных на коэффициенты C_k :

$$s(t) = \sum_k C_k \Psi_k(t) \quad (12.1)$$

Термин *wavelet*, введенный впервые Морле (J. Morlet), означает «короткая или маленькая волна».

Под вейвлет-анализом будем понимать декомпозицию сигнала при помощи прямого вейвлет-преобразования. Прямым вейвлет-преобразованием (ContinuousWaveletTransform — CWT) называют вычисление вейвлет-коэффициентов в области определения R :

$$C(a, b) = \int_{-\infty}^{\infty} s(t) a^{-\frac{1}{2}} \Psi\left(\frac{t-b}{a}\right) dt \quad (12.2)$$

Интегральное преобразование (12.2) называют непрерывным, если параметры масштаба a и сдвига b , в ходе обработки сигнала, принимают любые действительные значения. Оно избыточно по затратам времени на вычисления, но зато позволяет добиться наибольшей детализации при анализе сигналов.

Прямое непрерывное вейвлет-преобразование лежит в основе построения вейвлет-спектрограмм сигналов. В простейшем случае вейвлет-спектрограмма отражает уровень вейвлет-коэффициентов (градиациям и

яркости) в плоскости масштаб-время. На рисунке 12.1 показан пример построения нестационарного сигнала (синусоида, на которую наложен прямоугольный импульс с изменяющимся периодом повторения). Перепады сигнала на спектрограмме, полученной в MATLAB, представлены в виде «зубцов». Уменьшение амплитуды перепадов в сотни-тысячи раз (при этом они сливаются с синусоидой) перепады тем не менее, отчетливо выявляются спектрограммой (подокно снизу). Способность вейвлет-спектрограмм обнаруживать артефакты сигналов не имеет прецедентов в технике спектрального анализа. Это полезно в сейсмографии, медицине и других областях применения вейвлет-анализа.

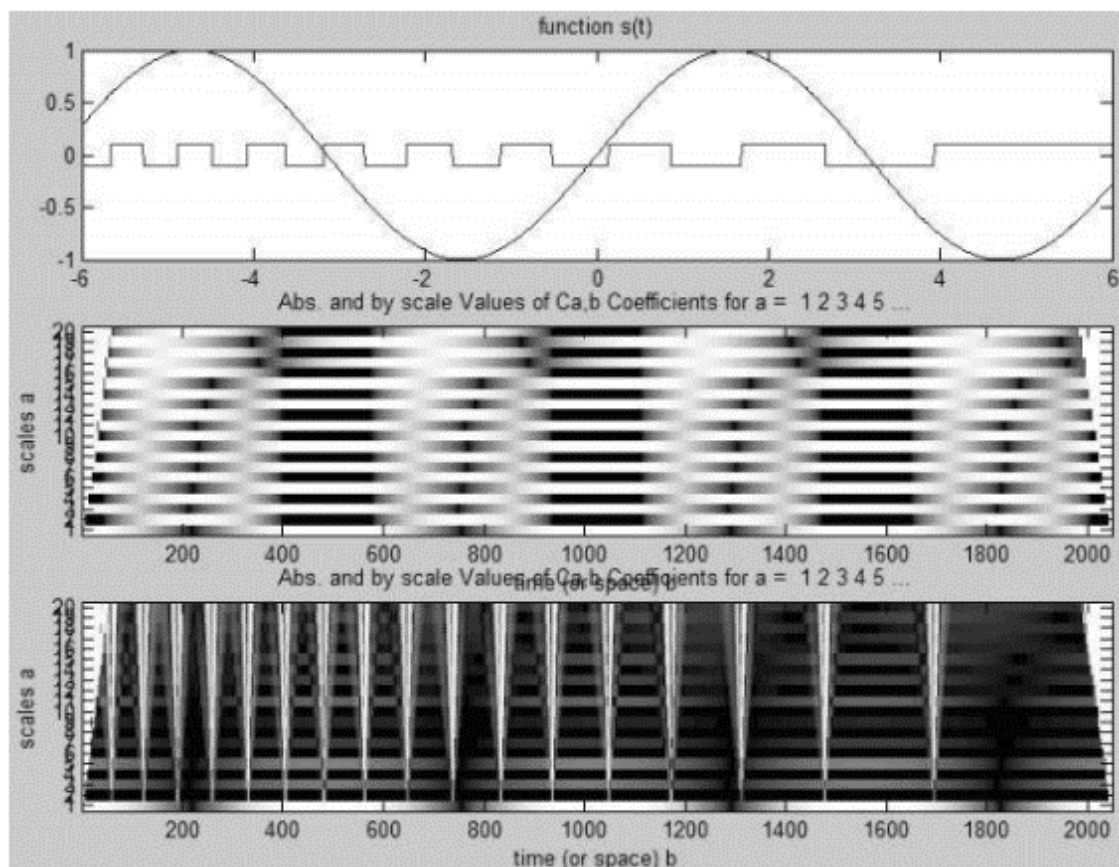


Рисунок 12.1. Вейвлет – спектрограмма сигнала синусоиды, на которую наложен прямоугольный импульс с изменяющимся периодом повторения

Пакет расширения WaveletToolbox системы MATLAB является одним из самых мощных средств для работы с вейвлетами. Эта работа возможна как с помощью окон графического интерфейса пользователя GUI (пример

рисунок12.2), так и из окна командного режима. Будучи установленный (вместе с базовой системой MATLAB) пакет может использоваться для обработки осциллограмм и спектров от современных цифровых измерительных приборов, расширяя области их применения.

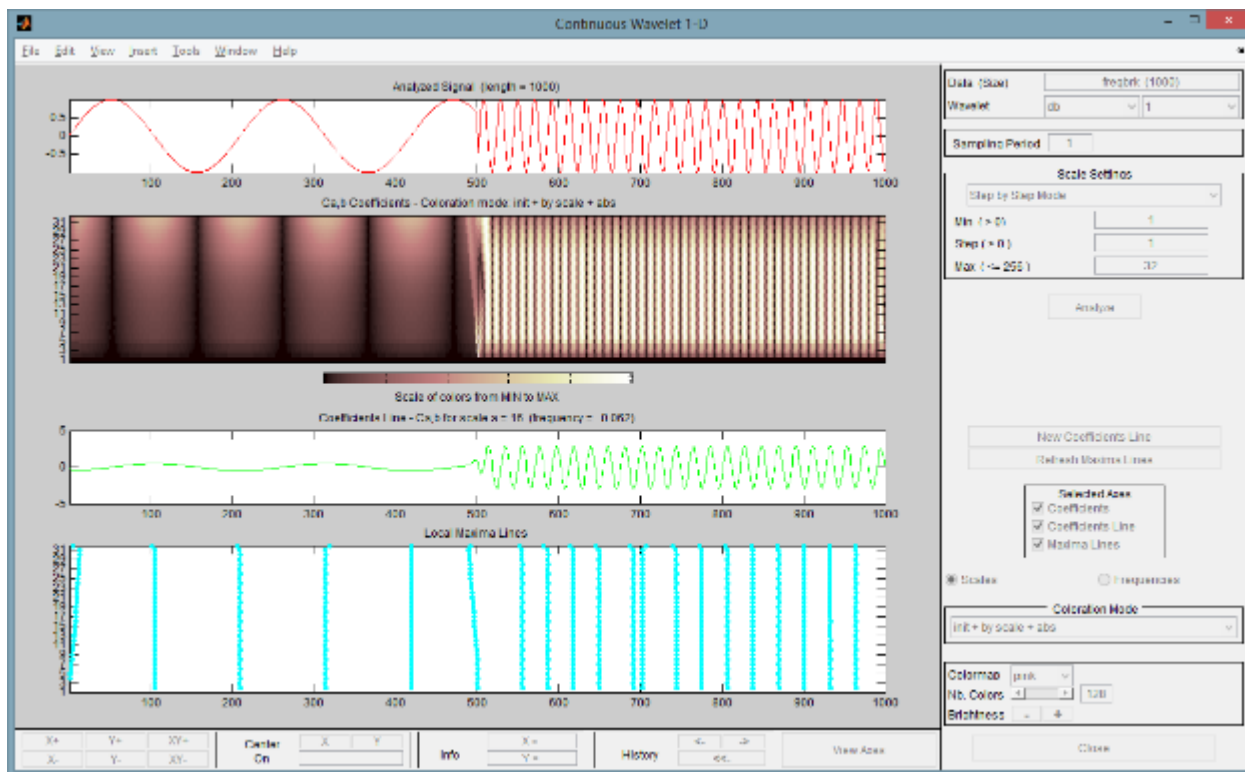


Рисунок 12.2. Вейвлет-спектрограмма двухчастного сигнала, полученная в окне GUI системы MATLAB

Вейвлет-анализ в системе MATLAB реализуется CWT функцией $\text{COEF} = \text{cwt}(\text{ydata}, \text{start}:\text{step}:\text{end}, 'W', 'opt')$, где ydata — вектор сигнала, start и end — начальный и конечный уровни разложения, W — тип вейвлета и opt — опция, задающая тип окраски вейвлет-спектрограммы, которую строит эта функция. Функция возвращает вектор вейвлет-коэффициентов COEF .

Вейвлет-синтез сигнала заключается в изменении вейвлет-коэффициентов исходного сигнала с целью в результате выполнения обратного вейвлет-преобразования получить сигнал с новыми свойствами. В зависимости от способа модификации коэффициентов достигаются различные цели вейвлет-синтеза, например, фильтрация или сжатие.

Обратное непрерывное вейвлет-преобразование осуществляется по формуле реконструкции во временной области:

$$f = C_{\psi}^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{dad b}{a^2} \langle f, \psi^{a,b} \rangle \psi^{a,b} \quad (12.3)$$

В пакете расширения MATLAB — WaveletToolbox используется следующая формула реконструкции сигнала:

$$s(t) = \frac{1}{K_{\psi}} \int_{R+} \int_{R+} C(a,b) a^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) \frac{dad b}{a^2}, \quad (12.4)$$

где K_{ψ} — константа, определяемая функцией ψ .

Если a и b задавать дискретно, что обычно и делается, то можно реализовать дискретное кратно масштабное вейвлет-преобразование. Чаще всего кратность берется равной 2^L , что соответствует диадическому вейвлет-преобразованию, где целое L есть уровень преобразования, задающий разрешение.

Средства вейвлет-преобразований пока не используются в серийных цифровых осциллографов и анализаторах спектров. Но они уже представлены в ряде пакетов расширений по вейвлетам СКМ Mathcad, Mathematica и MATLAB. При установке этих систем приборы открытой архитектуры эти средства уже можно полноценно использовать в практике исследований и измерений. Это возможно так же пристыковке измерительных приборов с закрытой архитектурой с ПК, на которых установлены СКМ. Вейвлет-преобразования наиболее эффективны при решении следующих наиболее типовых задач:

- построение вейвлет-спектрограмм и анализ тонких особенностей сигналов;
- фильтрация сигнала путем ограничения числа используемых вейвлет-коэффициентов;
- очистка и сглаживание сигналов (в том числе нестационарных) и изображений от шума и помех;
- компрессия (сжатие) сигналов с малыми их искажениями;

- интерполяция, экстраполяция и аппроксимация сигналов [4];

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории)..

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Пакет расширения WaveletToolbox системы MATLAB позволяет использовать вейвлет анализ и преобразование данных в самых различных областях науки и техники.

Программное обеспечение пакета позволяет выполнять вейвлет-преобразования как в командном режиме (и готовить специализированные программы), так и в диалоговом режиме по интерфейсу GUI, который включается командой «wavemenu» (рисунок 12.3).

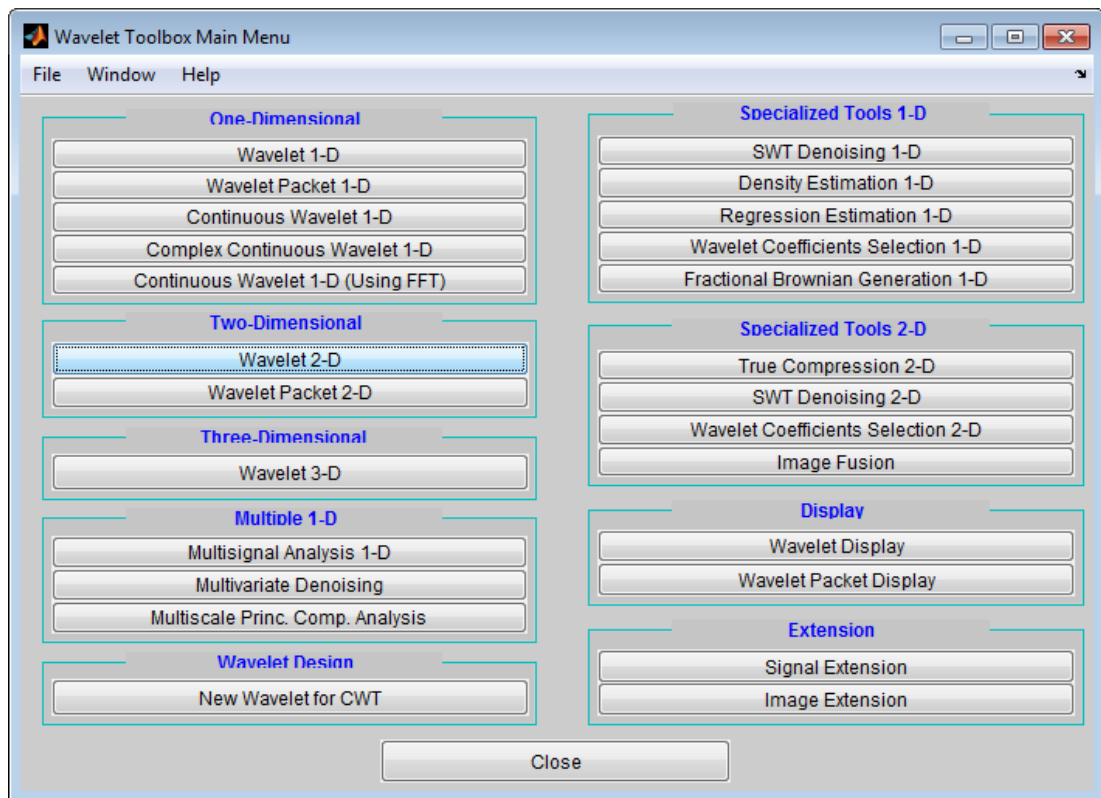


Рисунок 12.3. Интерфейс WaveletToolbox

Пакет имеет демонстрационные примеры вейвлет-преобразований, окно которых включается командой «wavedemo» (рисунок 12.4).

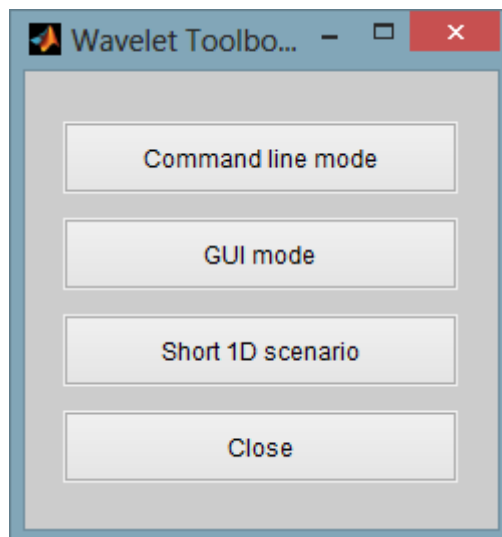


Рисунок 12.4. Интерфейс WaveletToolboxDemo

Первой кнопкой окна (Commandlinemode) включается довольно обширное меню примеров работы в командном режиме с одно- и

двумерными вейвлетами всех типов в обычном и в пакетном исполнении (непрерывные и дискретные вейвлет-преобразования с декомпозицией и реконструкцией сигналов, сжатие сигналов, очистка от шумов и пр.). Слайды примеров сопровождаются соответствующими листингами программных фрагментов.

Аналогично второй и третьей кнопками включается доступ к демонстрационным примерам в интерфейсе GUI.

При изучении настоящей темы рекомендуется самостоятельно просматривать соответствующие демонстрационные примеры по ходу изложения материала.

Задание 12.1. Произвести анализ спектра непрерывного вейвлет-преобразования сигнала, параметры и аналитическое выражение которого, приведены ниже:

$$s(t) = \sin(2\pi \cdot 10^4 \cdot t + 45).$$

Построить вид рассматриваемого сигнала во временной области, спектр его непрерывного вейвлет-преобразования, тип вейвлет-функции, применяемой для анализа.

Алгоритм вычисления непрерывного вейвлет-преобразования сигнала является реализованным в пакете MATLAB с использованием функции `cwt`, листинг которой приведен ниже.

```
COEFS = cwt(S, SCALES, 'wname', PLOTMODE, XLIM)
```

```
% функция для вычисления непрерывного вейвлет-преобразования
% COEFS -- коэффициенты непрерывного вейвлет-преобразования
% S -- анализируемый сигнал
% SCALES -- параметр, изменяющий временной масштаб вейвлет-функции
% 'wname' -- тип вейвлет-функции, применяемый
% для вычисления спектра непрерывного вейвлет-преобразования
% PLOTMODE -- параметр, изменяющий настройки цвета, для отображения набора
% вейвлет-коэффициентов. PLOTMODE может принимать следующие значения:
% 'lvl' -- окраска шаг за шагом
% 'glb' -- окраска с учетом всех коэффициентов
% 'abslvl' или 'lvlabs' -- окраска шаг за шагом с использованием
% абсолютных значений вейвлет-коэффициентов
```

```

% 'absglb' или 'glbabs' -- окраска с масштабированием и использованием
% абсолютных значений вейвлет-коэффициентов
% Использование в настройках PLOTMODE значений
% '3Dlvl', '3Dglb', '3Dabslvl' ('3Dlvlabs'),
% '3Dabsglb' ('3Dglbabs'), взамен 'lvl', 'glb',
% 'abslvl' ('lvlabs'), 'absglb' ('glbabs'),
% соответственно, позволяет отображать спектр вейвлет-коэффициентов
% в виде поверхности взамен плоскости
% XLIM - параметр, изменяющий настройки цвета, для отображения набора
% вейвлет-коэффициентов.
% Представляет собой вектор [x1 x2], 1<=x1<x2<=length
% Программа для вычисления и построения спектра
% непрерывного вейвлет-преобразования

t = 0:0.000001:0.0004;
A = 1; F = 10000; phi = 45;
signal = A*sin(2*pi*F*t + phi);
subplot(211); plot(t,signal);
axis([0 0.0004 -1 1]);
title('Исходный сигнал'); grid
xlabel('Время, t'); ylabel('Значения сигнала')
subplot(212);
c = cwt(signal,1:1:16,'mexh','3Dglbabs',[0 15]);
title('Вейвлет-спектр');
xlabel('Временной сдвиг, b');
ylabel('Временной масштаб, a')
lb = -5; ub = 5; n = 1000;
[psi,x] = gauwavf(lb,ub,n,2);
figure; plot(x,psi);
title('Гауссов вейвлет'); grid
xlabel('t'); ylabel('\psi(t)')

```

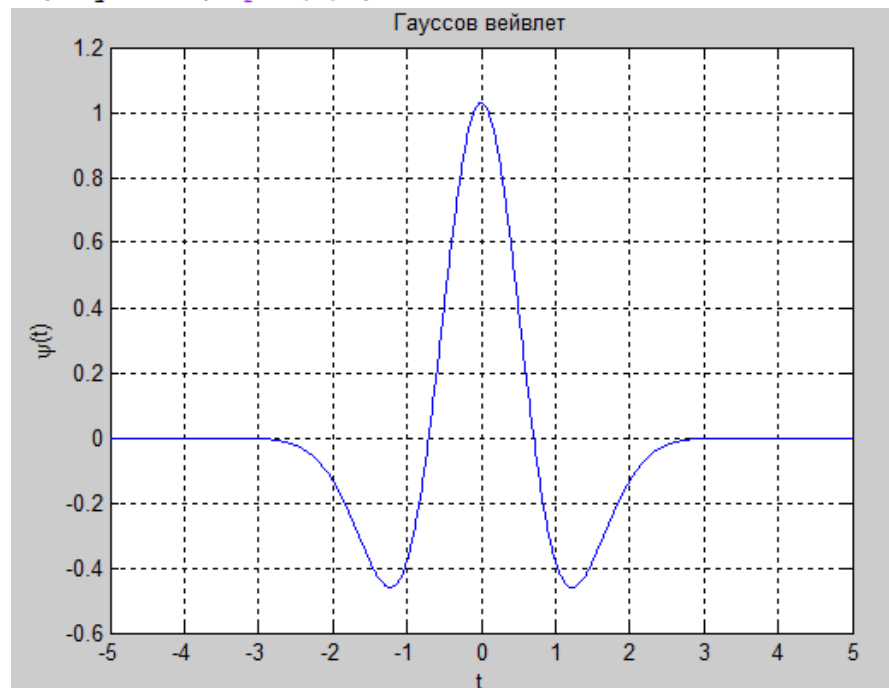


Рисунок 12.5 – Пример вейвлет-функции используемой для построения спектра приведенного на рисунке 12.6

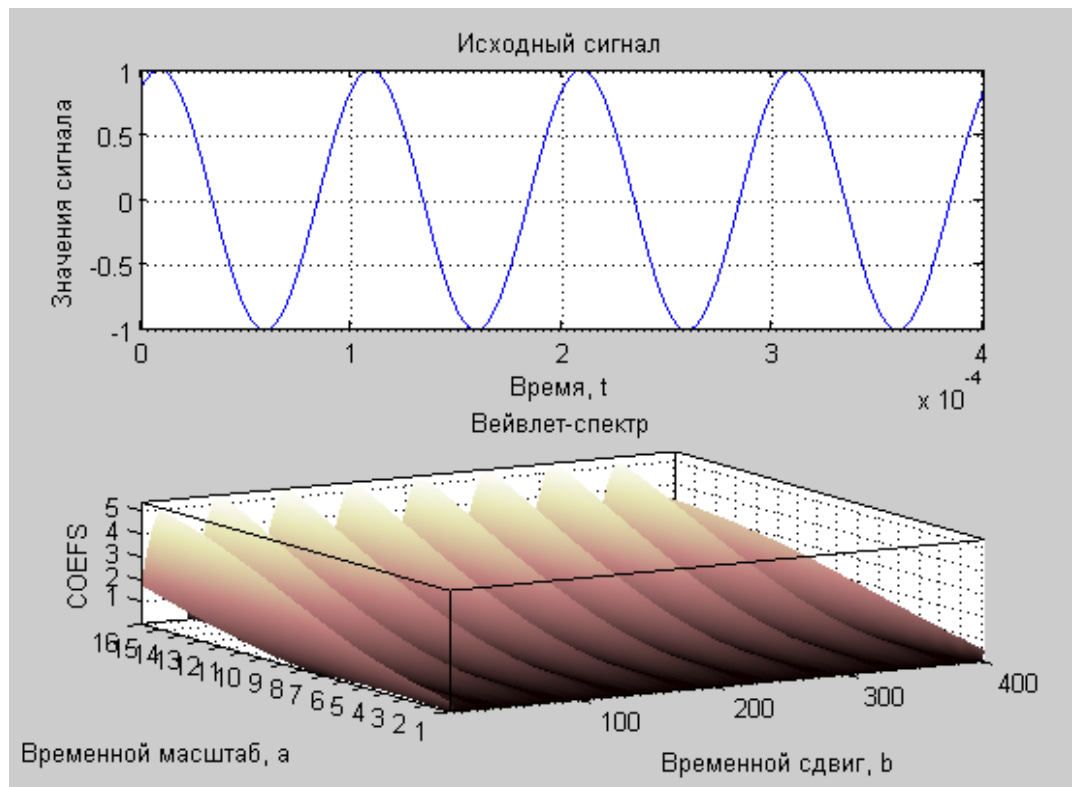


Рисунок 12.6 – Пример анализируемого сигнала и его непрерывного вейвлет-спектра

Задание 12.2. Выполнить реконструкцию сигнал используя коэффициенты детализации.

```
% Исходный сигнал
x = 0: pi/10: 10*pi;
l = length (x);
s = sin (x) + rand (1, l);
% Выбор вейвлета
wname = 'coif2';
% Дискретное преобразование
[sa, sd] = dwt (s, wname, 'mode', 'ppd');

% Графическая поддержка разложения
subplot (411)
plot (s), title ('Исходный сигнал')
subplot (412)
plot (sa), title ('Коэффициенты аппроксимации')
```

```

subplot (413)
plot (sd), title ('Коэффициенты детализации')
% Восстановление сигнала
s1 = idwt (sa, sd, wname, 'mode', 'ppd');
% Рисование восстановленного сигнала
subplot (414)
plot (s1), title ('Восстановленный сигнал')

```

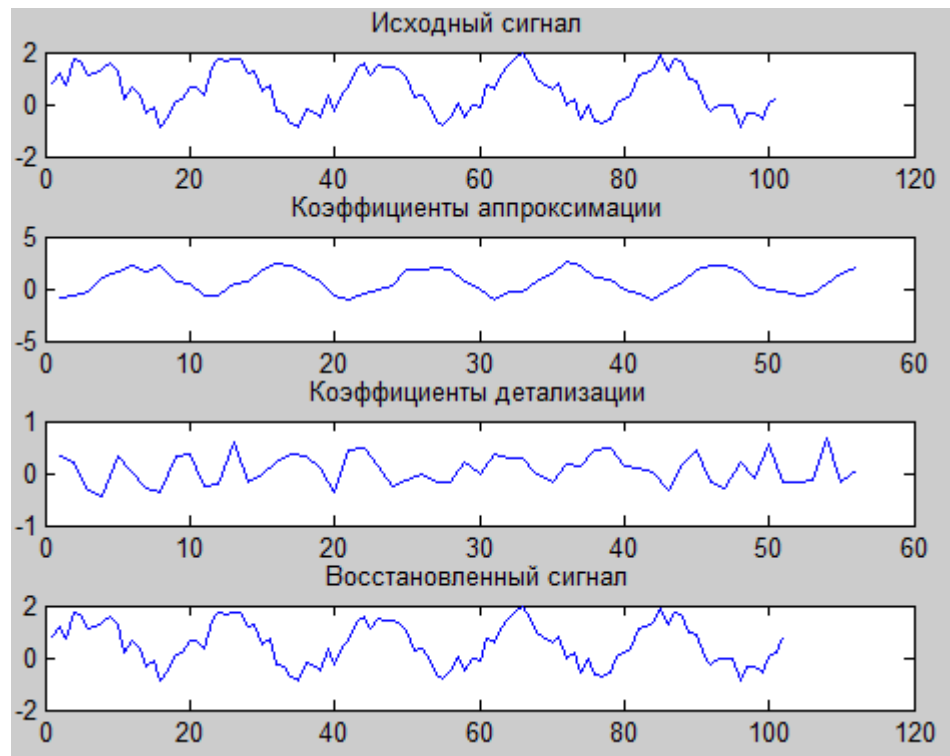


Рисунок 12.7 – Восстановление сигнала с использованием коэффициентов детализации

Задание 12.3. Выполнить реконструкцию сигнал без коэффициентов детализации.

```

% Исходная последовательность данных
x = 0: pi/10: 10*pi;
l = length (x);
s = sin (x) + rand (1, l);
% Вейвлет-функция как базис преобразования
wname = 'coif2';

```

```

% Дискретный анализ
[sa, sd] = dwt (s, wname, 'mode', 'ppd');
% Графическая поддержка
subplot (211)
plot (s), title ('Исходный сигнал')

% Восстановление без коэффициентов детализации
% играет роль низкочастотного фильтра
s1 = idwt (sa, [], wname, 1);
% "Восстановленный" сигнал
subplot (212)
plot (s1), title ('Восстановленный сигнал')

```

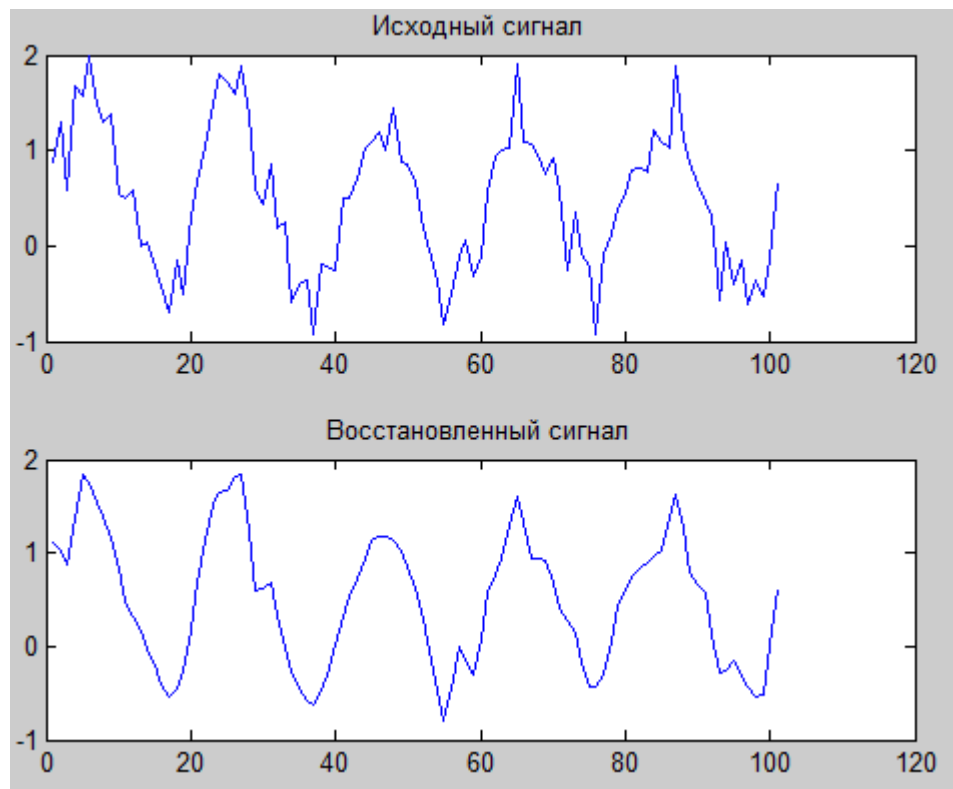


Рисунок 12.8 – Восстановление сигнала без использования коэффициентов детализации

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;

- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что такое вейвлет?
2. Что понимаю под вейвлет-синтезом?
3. Что такое непрерывное вейвлет-преобразование?
4. Какая функция MATLAB реализует вейвлет-преобразование?
5. В чем заключается вейвлет-синтез сигнала?

ЛАБОРАТОРНАЯ РАБОТА 13

Вейвлет-обработка и компрессия сигналов в MATLAB(2 ч)

Цель и содержание: получить основные навыки вейвлет-обработки и компрессии сигналов.

Теоретическое обоснование

Важным применением вейвлетов стала очистка сигналов от шума и их сжатие (компрессия). Они основаны на ограничении уровня используемых коэффициентов.

В простейшей модели предполагается, что зашумленный сигнал имеет вид $s(n) = f(n) + \sigma e(n)$, где $f(n)$ – полезный сигнал, σ – уровень шума и $e(n)$ – гауссов белый шум, т. е. стационарная случайная последовательность с нулевым математическим ожиданием, абсолютно некоррелированная и с дисперсией, равной единице. Название «белый шум» отражает тот факт, что такой сигнал $e(n)$ имеет постоянный спектр на всех частотах по аналогии с белым светом, имеющим в видимой части равномерный сплошной спектр. Цель состоит в том, чтобы подавить шумовую часть $e(n)$ сигнала и восстановить f .

Основу методики удаления шумов из сигнала с использованием вейвлет-преобразования составляет использование пороговых функций различной формы, на основе которых происходит ограничение уровня детализирующих коэффициентов. Задав определенный порог для их уровня и «отсекая» коэффициенты ниже этого порога, можно значительно снизить уровень шума и сжать сигнал. Пример пороговых функций, рассматриваемых в данной работе и зачастую используемых в современных алгоритмах фильтрации, основанных на кратномасштабных схемах преобразования, представлен на рисунке 13.1.

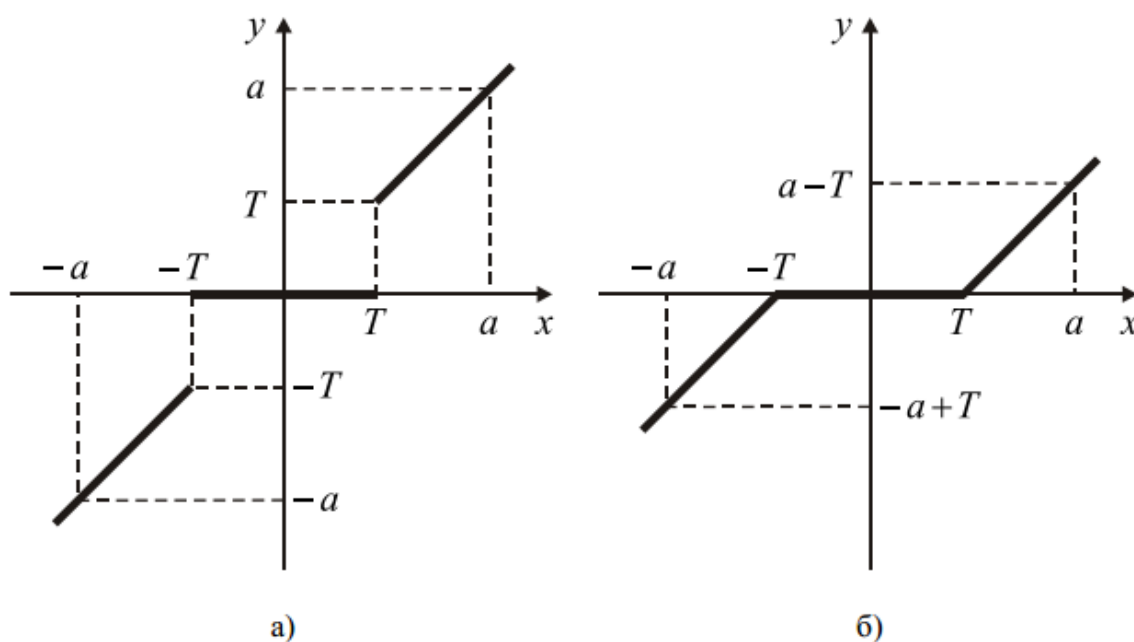


Рисунок 13.1. Пороговые функции обработки коэффициентов преобразования: а) жесткая пороговая функция; б) мягкая пороговая функция

На рисунке 13.1а представлена жесткая пороговая функция (жесткая пороговая оценка), описываемая следующим выражением

$$y(x) = \begin{cases} x, & \text{если } |x| \geq T, \\ 0, & \text{если } |x| < T \end{cases} \quad (13.1)$$

Здесь в качестве величины T выступает некоторое пороговое значение, о возможности определения которого будет сказано ниже, в качестве x и y выступают входное и выходное значения коэффициентов преобразования. На рисунке 13.1 б представлена мягкая пороговая функция (мягкая пороговая оценка), описываемая следующим выражением:

$$y(x) = \begin{cases} \text{sign}(x)(|x| - T), & \text{если } |x| \geq T, \\ 0, & \text{если } |x| < T \end{cases} \quad (13.2)$$

Обозначения, представленные в выражение (13.2) те же, что и описанные выше, в качестве функции $\text{sign}(x)$ определяющей знак коэффициента x выступает функция вида:

$$\text{sign}(x) = \begin{cases} 1, & \text{если } x > 0, \\ 0, & \text{если } x = 0, \\ -1, & \text{если } x < 0. \end{cases} \quad (13.3)$$

Основное отличие мягкой пороговой функции от жесткой заключается в том, что первая из них (мягкая пороговая функция) не содержит разрыва в точке определяемой пороговым значением T . Иными словами мягкая пороговая функция, в отличие от жесткой, является непрерывной. Последнее обстоятельство, в случае мягкой пороговой обработки позволяет лучшим образом обрабатывать зашумленный сигнал (изображение) в окрестности точки его разрыва. Нужно отметить, что понижение значения коэффициентов разложения на величину порогового значения, в случае мягкой пороговой обработки, в целом для большого количества сигналов оказывает неблагоприятное воздействие на итоговую оценку качества восстановленного сигнала. Поэтому в целом, как показывают эксперименты, более лучшей, с точки зрения численной оценки качества восстановленного сигнала, является жесткая пороговая оценка.

Как показывают эксперименты для того, чтобы в случае мягкой пороговой обработки численная оценка качества восстановленного сигнала приближалась к последней в случае жесткой пороговой обработки, необходимо выбирать значение порога T равным половине значения порога жесткой обработки сигнала. Основной проблематикой использования пороговых методов обработки является выбор порогового значения. В большом количестве научных работ выбор порога осуществлялся с использованием следующего выражения:

$$T = \sigma \sqrt{2 \ln(N)} \quad (13.4)$$

Формула (13.4) записана для случая одномерного сигнала. В формуле использованы следующие обозначения: N - общее количество отсчетов обрабатываемого сигнала, σ - среднеквадратическое отклонение шума, оцениваемое в данной работе с использованием следующего выражения

$$\sigma = \frac{M_X}{0.6745} \quad (13.5)$$

где M_X - медиана абсолютного значения множества коэффициентов $\{X\}$

Выражение(13.5) очень хорошо подходит для оценки среднеквадратического отклонения белого гауссова шума с нулевым математическим ожиданием, который является аддитивно подмешанным в некоторый полезный сигнал[5].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

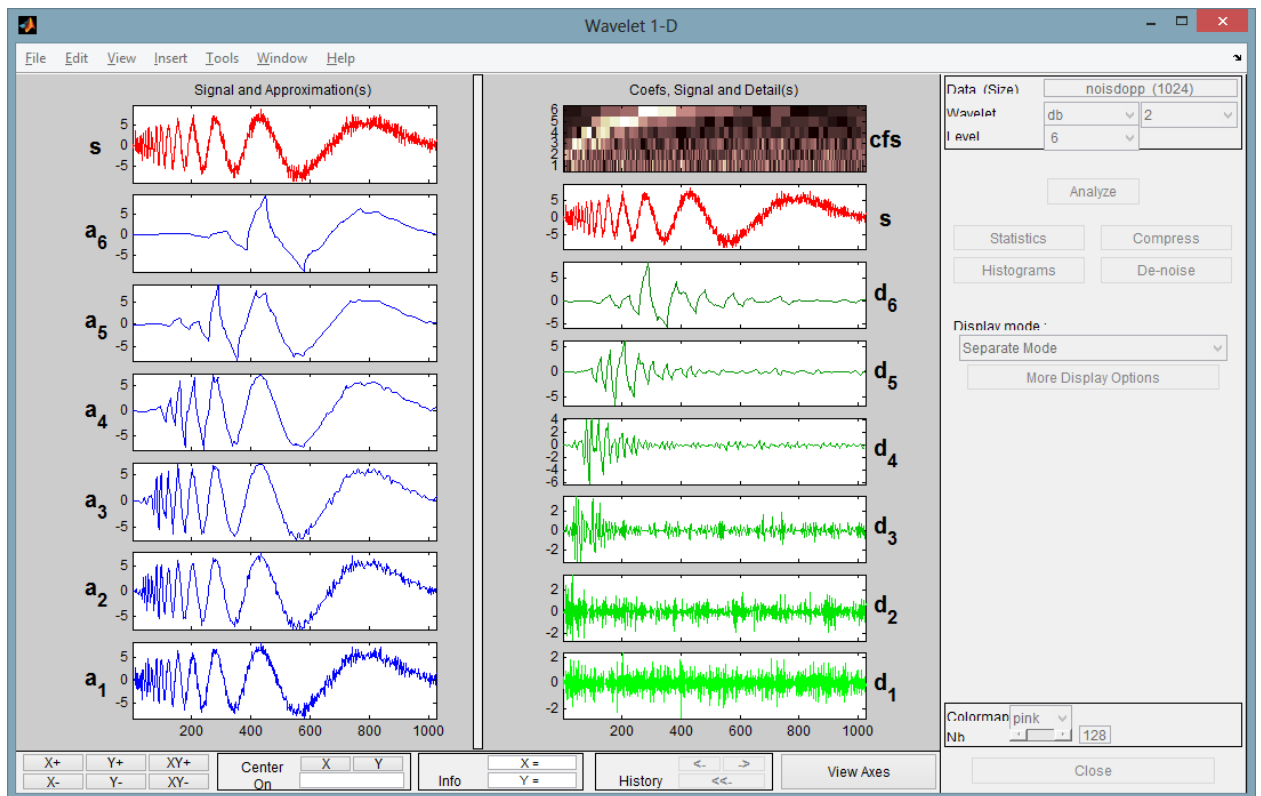


Рисунок 13.1.Окно преобразования в режиме графического вывода всех коэффициентов разложения (SeparateMode)

Окно преобразования включается из 'Wavelet Toolbox Main Menu' кнопкой "Wavelet 1-D" и имеет несколько большие функциональные возможности. Пример окна приведен на рисунке 13.1 в режиме графического вывода всех коэффициентов разложения (Separate Mode). Переключателем "Display mode" режимы вывода можно изменять, детали вывода можно устанавливать в отдельном подокне (кнопка "More Display Options").

Под кнопкой "Analyze" окно имеет 4 кнопки включения окон выполнения специальных операций над результатами разложения сигнала.

Окна "Statistics" и "Histograms" предназначены для анализа и графического вывода статистических характеристик сигнала и всех коэффициентов его разложения. В окнах "Compress" и "De-noise" устанавливаются режимы компрессии (сжатия) сигналов и очистки сигналов от шумов, и выполняются эти операции. Их применение будет рассмотрено в дальнейшем.

Подробности использования функций для проведения очистки сигнала от шума, а также компрессии можно найти в руководстве пользователя «De-noising» в Chapter 6, «Advanced Concepts» (Wavelet Toolbox\Usingthe Wavelet Toolbox\Advanced Concepts\Wavelet Applications\De-noising).

Основные функции очистки сигналов от шума:

- **thselect** – выбор порога для удаления шума 1-D;
- **wthresh** – мягкая или жесткая пороговая обработка (1-D, 2-D);
- **wthcoef** – пороговая обработка;
- **wthcoef2** – двумерная пороговая обработка;
- **wpthcoef** – пороговая обработка пакетных коэффициентов (1-D, 2-D);
- **wden** – автоматическое удаление шума 1-D;
- **wdencmp** – удаление шума и сжатие при помощи вейвлетов (1-D, 2-D);

Функция **thselect** – выбор порога для удаления шума 1D.

$THR = thselect(X, TPTR).$

Находит значение порога для сигнала X , используя правило, определенное строкой $TPTR$. Этот параметр может принимать следующие значения:

- $TPTR = 'rigrsure'$ – адаптивный выбор порога, используя принцип Штейна несмещенной оценки риска;
- $TPTR = 'heursure'$ – эвристический вариант первого выбора;
- $TPTR = 'sqtwolog'$ – $porogsqrt(2*\log(\text{length}(X)))$;
- $TPTR = 'minimaxi'$ – минимаксный порог.

Правила выбора порога базируются на основной модели $y = f(t) + e$, где e является белым шумом $N(0,1)$. Если шум не масштабированный или небелый, можно его обработать, используя перемасштабирование выходного порога THR .

Функция **wden** – автоматическое удаление шума (1D). Функция **wden** автоматически удаляет шум одномерного сигнала X , используя вейвлеты. Применяется в виде:

$[XD, CXD, LXD] = wden(X, TPTR, SORH, SCAL, N, 'wname'),$

$[XD, CXD, LXD] = wden(C, L, TPTR, SORH, SCAL, N, 'wname')$.

Первая версия функции вычисляет очищенную от шума версию XD входного сигнала X, используя пороговую обработку вейвлет-коэффициентов. Дополнительно находит вейвлет-коэффициенты [CXD, LXD] разложения очищенного сигнала XD.

Исходные параметры:

- X – сигнал;
- TPTR – строка, определяющая выбор порога:
 - 'rigsure' – адаптивный выбор порога, используя принцип Штейна несмещенной оценки риска (SURE);
 - 'heursure' – эвристический вариант первого выбора;
 - 'qtwolog' – универсальный порог $\sqrt{2 \cdot \log(\text{length}(X))}$;
 - 'minimaxi' – минимаксный порог;
- SORH = 's' или 'h' – выбор мягкого или жесткого порога;
- SCAL – строка, которая определяет мультипликативное пороговое перемасштабирование. Если шум вне пределов [0,1] или небелый, то порог должен быть перемасштабирован с использованием оценки уровня шума. Возможны следующие варианты:

- 'one' – без перемасштабирования, когда используется базовый метод;
- 'sln' – для перемасштабирования порога с использованием оценки уровня шума на базе коэффициентов первого уровня;
- 'mln' – для перемасштабирования порога с использованием оценки уровня шума, зависящей от уровня разложения;
- N – уровень вейвлет-разложения и 'wname' – имя ортогонального вейвлета.

Во втором случае функция wden производит то же самое, но использует уже полученное вейвлет-разложение [C,L] исходного сигнала до уровня N и ортогональный вейвлет 'wname'.

Функция wdenstr – удаление шума и сжатие (компрессия) при помощи вейвлетов (1D, 2D).

[XC,CXC,LXC, PR0, PRL2] =

wdencmp('gbl',X,'wname',N,THR,SORH,KEEPAPP),

[XC,CXC,LXC, PR0, PRL2] = wdencmp('lvd',X,'wname',N,THR,SORH),

[XC,CXC,LXC, PR0, PRL2] = wdencmp('lvd',C,L,'wname',N,THR,SORH).

Описание. Случай параметра 'gbl'. Функция производит:

- XC – очищенную от шума версию входного сигнала X (1-D или 2-D);
- [CXC,LXC] – структуру вейвлет-разложения сигнала XC;
- PR0 – оценку числа обнуленных коэффициентов сигнала X в

процентах;

• PRL2 – отношение L^2 -норм сжатого и первоначального сигналов в процентах. $PRL2 = 100 \cdot (\text{norm}(CXC) / \text{norm}(C))^2$, где [C,L] – вейвлет-разложение X. Если X – одномерный сигнал и 'wname' – ортогональный вейвлет, то $PRL2 = 100 \cdot \|XC\|^2 / \|X\|^2$.

Функция использует:

- X – сигнал, или изображение;
- wname – имя вейвлета и N – уровень вейвлет-разложения;
- THR – значение порога;
- SORH = 's' или 'h' – выбор мягкого или жесткого порогового метода;
- KEEPAPP = 1, то коэффициенты аппроксимации не подвергаются

пороговой обработке, иначе это возможно.

Функция wdencmp('gbl',C,L,'wname',N,THR,SORH,KEEPAPP) выполняет тоже самое, используя вейвлет-разложение [C,L] исходного сигнала до уровня N.

В одномерном случае и опции 'lvd'. Функция выполняет то же самое, но используя зависимые от уровня пороги, указанные в векторе THR (THR должны иметь длину N). Кроме того, коэффициенты аппроксимации сохраняются.

Для двумерного случая и опции 'lvd'. В этом случае THR должен быть матрицей $3 \times N$, содержащей зависимые от уровня пороги в трех направлениях: по горизонтали, по диагонали и по вертикали.

Отметим, что сравнительно с `wden` (автоматическое удаление шума), `wdenctr` дает больше возможностей и можно осуществить свою стратегию удаления шума. Подробности можно найти в Chapter 2, «Using Wavelets».

Задание 13.1. Используя средства написания программного кода системы Matlab произвести построение и исследование алгоритма фильтрации аддитивного белого гауссова шума с нулевым математическим ожиданием и произвольно выбранным значением дисперсии из некоторого полезного сигнала на основе вейвлет-преобразования.

Основные этапы реализуемого алгоритма должны включать в себя:

1. Вычисление прямого вейвлет-преобразования сигнала (выбор типа вейвлет-функции и числа уровней вейвлет-разложения).

2. Изменение полученных коэффициентов преобразования по определенному правилу (выбор алгоритма нахождения порогового значения, выбор пороговой функции, выбор стратегии обработки детализирующий коэффициентов вейвлет-разложения).

3. Вычисление обратного вейвлет-преобразования от множества модифицированных (обработанных) вейвлет-коэффициентов.

Основные пункты исследования работы алгоритма должны включать:

1. Сравнение использования мягкой и жесткой пороговых функций внутри рассматриваемого алгоритма фильтрации. Подобную процедуру необходимо проводить при пяти различных степенях зашумления сигнала (дисперсии шума), произвольно выбранных настроек алгоритма, включающих в себя: тип вейвлет-функции, число уровней вейвлет-разложения, алгоритм нахождения порогового значения, стратегию обработки детализирующий коэффициентов вейвлет-разложения. Сравнение качества работы алгоритма осуществить на основе критерия среднеквадратической ошибки (СКО), вычисляемой с использованием следующего математического выражения:

$$\text{СКО} = \frac{1}{N} \sum_{k=0}^{N-1} \left(f(k) - \hat{f}(k) \right)^2 \quad (13.6)$$

где $f(k)$ – эталонный сигнал, $\hat{f}(k)$ – восстановленный сигнал

2. Сравнение использования различных типов вейвлет-функций внутри рассматриваемого алгоритма фильтрации. В качестве вейвлет-функций рекомендуется выбирать вейвлеты из класса Добеши.

Подобную процедуру необходимо проводить при пяти различных степенях зашумления сигнала (дисперсии шума), трех различных типов вейвлетов, произвольно выбранных настроек алгоритма, включающих в себя: число уровней вейвлет-разложения, алгоритм нахождения порогового значения, тип пороговой функции, стратегию обработки детализирующий коэффициентов вейвлет-разложения. Сравнение качества работы алгоритма осуществить на основе критерия среднеквадратической ошибки (СКО).

3. Сравнение использования различного числа уровней вейвлет-разложения внутри рассматриваемого алгоритма фильтрации. Подобную процедуру необходимо проводить при пяти различных степенях зашумления сигнала (дисперсии шума), трех различных значений числа уровней вейвлет-разложения, произвольно выбранных настроек алгоритма, включающих в себя: тип вейвлет-функции, алгоритм нахождения порогового значения, тип пороговой функции, стратегию обработки детализирующий коэффициентов вейвлет-разложения. Сравнение качества работы алгоритма осуществить на основе критерия среднеквадратической ошибки (СКО).

Создадим файл MSE.m и добавим в него следующую функцию:

```
function mse = MSE(signal,rsignal)
% функция для оценки качества восстановленного
% сигнала по критерию среднеквадратической ошибки
%
% mse -- среднеквадратическая ошибка
% signal -- эталонный сигнал
% rsignal -- восстановленный сигнал
%
```

```

if isequal(signal,rsignal)
    disp('Сигналы идентичны; СКО равно нулю')
else
    N = length(signal);
    mse = sum((double(signal) - double(rsignal)).^ 2)/N;
    disp(sprintf('СКО = %5.4f',mse));
end

```

В дальнейшем мы будем вызывать эту функцию для оценки качества восстановленного сигнала.

```

% Программа для обработки (фильтрации) сигнала
% на основе дискретного вейвлет-преобразования
%
t = 0:0.000001:0.0004;
A = 1; F = 10000; phi = 45;
signal = A*sin(2*pi*F*t + phi);
subplot(311); plot(t,signal)

title('Исходный не зашумленный сигнал'); grid
xlabel('Время, t'); ylabel('Значения сигнала')
signal_noise = signal + ...
random('norm',0,0.2,[1 length(t)]);
subplot(312); plot(t,signal_noise)
title('Зашумленный сигнал'); grid
axis([min(t) max(t) min(signal_noise) max(signal_noise)])
xlabel('Время, t'); ylabel('Значения сигнала')

tptr = 'sqtwolog';
sorh = 'h';
scal = 'one';
lev = 4;
wavefunc = 'db8';

rsignal = wden(signal_noise,tptr,sorh,scal,lev,wavefunc);
mse = MSE(signal,rsignal);
subplot(313); plot(t,rsignal)
title(['Восстановленный сигнал' ',' ' ' 'СКО' '=' num2str(mse)]); grid
axis([min(t) max(t) min(rsignal) max(rsignal)])
xlabel('Время, t'); ylabel('Значения сигнала')

```

Результат выполнения программы представлен на рисунке 13.2.

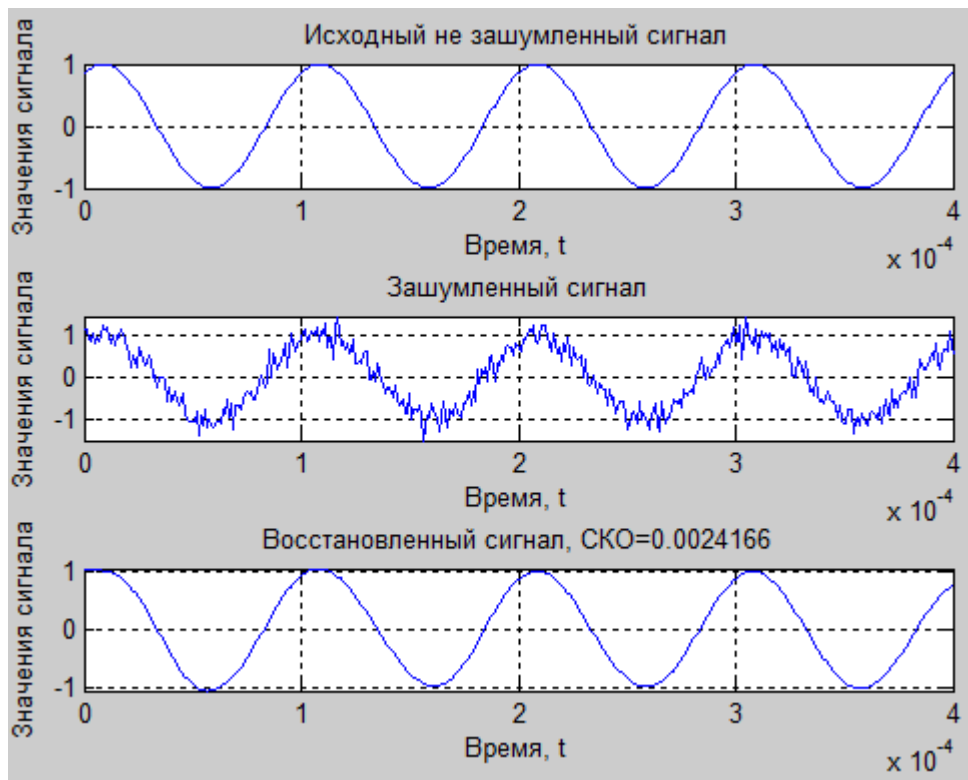


Рисунок 13.3. Окно обработки (фильтрации) сигнала на основе дискретного вейвлет-анализа

Задание 13.2. Удаление шума, компрессия. Функции вейвлет-анализа в MATLAB. Создать зашумленный сигнал функцией `wnoise`. Для удаления шума используем мягкий эвристический метод SURE. Разложение x берем до уровня 5 по вейвлету `sym8`.

```
snr = 3; init = 2055615866;
[xref,x] = wnoise(3,11,snr,init);
xd = wden(x, 'heursure', 's', 'one', 5, 'sym8');
subplot(311), plot(xref), axis([1 2048 -10 10]);
title('Original signal');

subplot(312), plot(x), axis([1 2048 -10 10]);
title('Noisy signal');
subplot(313), plot(xd), axis([1 2048 -10 10]);
title('De-noised signal - heuristic SURE')
```

Результат представлен на рисунке 13.3.

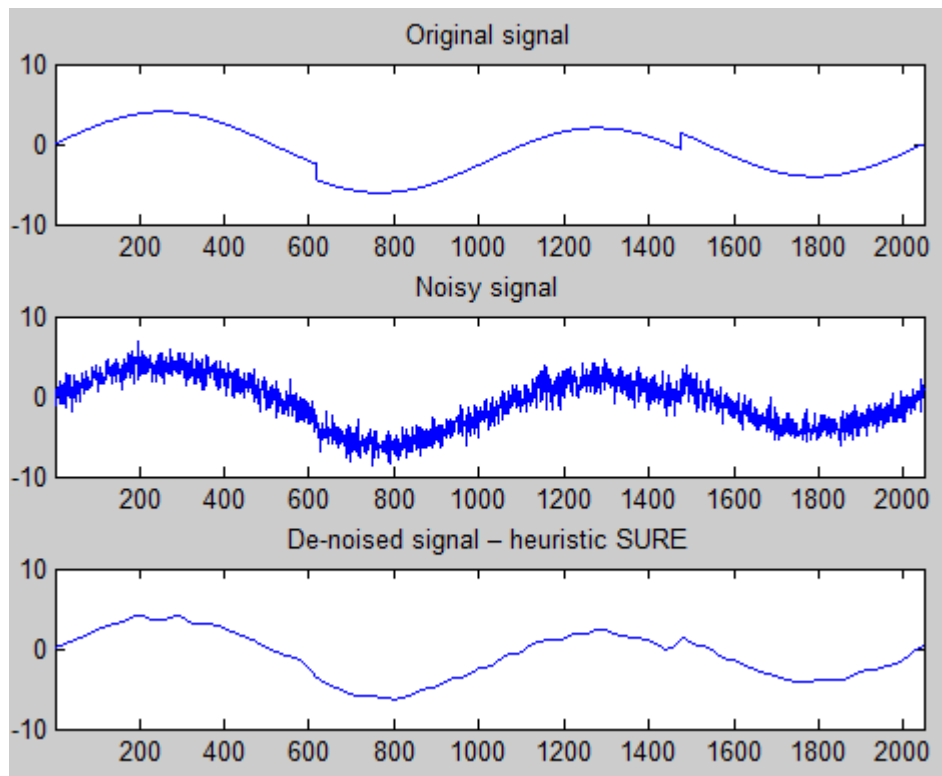


Рисунок 13.3.Окно преобразования в режиме графического вывода всех коэффициентов разложения (Separate Mode)

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. На чем основана идеология построения алгоритма фильтрации на основе вейвлет-преобразования?

2. Какие функции Matlab используются для создания зашумленного сигнала?
3. Назовите примерные этапы проведения алгоритма фильтрации зашумленного сигнала на основе вейвлет-преобразования.
4. Расскажите о функциях MATLAB, обеспечивающих удаление шума и сжатие (компрессию) при помощи вейвлетов.

ЛАБОРАТОРНАЯ РАБОТА №14

Моделирование случайного процесса на основе реализации случайного вектора с заданной ковариационной матрицей

Цель и содержание: изучение основных принципов компьютерного моделирования случайного процесса на основе реализации случайного вектора с заданной ковариационной матрицей.

Теоретическое обоснование

В основе моделирования случайных процессов различных видов лежит использование модели **гауссового (нормального) стационарного процесса (ГСП)** $\xi(t)$ с математическим ожиданием $m=M[\xi(t)]=0$ и с известной корреляционной функцией $R(\tau) = M[\xi(t) \xi(t+\tau)] = \sigma^2 \rho(\tau)$, где σ^2 - дисперсия, а $\rho(\tau)$ – нормированная функция корреляции процесса. Вместо функции $R(\tau)$ может быть изначально задана однозначно связанная с ней спектральная плотность мощности процесса $G(\omega)$.

Реализацию процесса $\xi(t)$ будем обозначать символом $\xi_*(t)$, т. е. помечаемую произвольным индексом *. При использовании цифровой ЭВМ разрабатываемая модель ГСП должна обеспечивать генерацию таких реализаций в дискретном времени для случайной последовательности $\xi[n]=\xi(n\Delta t)$, $n=1,2,\dots$ с заданным интервалом дискретизации Δt . Имея реализации ГСП и выполняя над ними различного рода линейные и нелинейные преобразования, можно получить (также в дискретном времени) реализации не гауссового случайного процесса с заданным законом распределения и корреляционной функцией. Таким образом, модель и формируемый на ее основе алгоритм генерации ГСП являются базовыми для получения моделей и алгоритмов генерации случайных процессов других видов.

Модель ГСП почти всегда базируется на выполнении некоторого линейного преобразования совокупности значений независимых гауссовых случайных величин $\{x[k], k=1,2,\dots\}$ в последовательность $\xi(n), n=1,2,\dots$ коррелированную по заданному закону. Рассмотрим возможные варианты применения данного подхода.

Простейшим способом получения реализации ГСП $\xi[n], n = \overline{1, N}$, конечной длины является ее представление как реализации случайного вектора $\xi = (\xi_1, \dots, \xi_N)^T, \xi_n = \xi[n], n = \overline{1, N}$ с матрицей ковариации вида

$$C = M[\xi\xi^T] = \|c_{ij}\|, c_{ij} = R(|i-j|\Delta t), i = \overline{1, N}, j = \overline{1, N} \quad (14.1)$$

Тогда, как известно, вектор ξ может быть получен на основе линейного преобразования вектора x , компоненты которого являются независимыми гауссовыми стандартными случайными величинами

$$\xi = Ax, x = (x_1, \dots, x_N)^T, M[x_i] = 0, M[x_i^2] = 1, i = \overline{1, N}, M[x_i x_j] = 0, \quad (14.2)$$

$$i \neq j$$

Здесь матрица A , выполняющая линейное преобразование, удовлетворяет уравнению

$$AA^T = C \quad (14.3)$$

Матрица A является нижней треугольной матрицей ($A = \|a_{ij}\|, a_{ij} = 0, j > i, i = \overline{1, N}, j = \overline{1, N}$). Подобное представление называется факторизацией ковариационной матрицы (факторизацией Холецкого). При выполнении такого преобразования

$$\xi_1 = a_{11}x_1, \xi_2 = a_{21}x_1 + a_{22}x_2, \dots, \xi_N = a_{N1}x_1 + \dots + a_{NN}x_N \quad (14.4)$$

С учетом того, что

$$c_{11} = M[\xi_1^2] = a_{11}^2, c_{12} = M[\xi_1 \xi_2] = a_{11}a_{21}, c_{22} = M[\xi_2^2] = a_{21}^2 + a_{22}^2, \dots, \quad (14.5)$$

определение ненулевых элементов матрицы A может производиться на основе подобных уравнений, связывающих элементы матрицы ковариации C с элементами матрицы A , последовательно в виде.

$$\begin{aligned}
 a_{11} &= \sqrt{c_{11}}, a_{21} = c_{12}/\sqrt{c_{11}}, a_{22} = \sqrt{c_{22} - c_{12}^2/c_{11}}, a_{31} = c_{13}/\sqrt{c_{11}}, \\
 a_{32} &= (c_{23} - c_{12}c_{13}/c_{11})/\sqrt{c_{22} - c_{12}^2/c_{11}}, a_{33} = \sqrt{c_{33} - a_{31}^2 - a_{32}^2}, \dots
 \end{aligned}
 \tag{14.6}$$

Следует отметить, что при выполнении преобразования (14.2) для случая негауссового вектора x с теми же параметрами вектор ξ будет иметь негауссово распределение с таким же (нулевым) математическим ожиданием и матрицей ковариации C .

Использование рассмотренного подхода к генерации реализаций ГСП как реализаций случайного вектора с заданной ковариационной матрицей обладает одним недостатком: размерность генерируемого вектора и, соответственно, длина получаемой реализации ГСП ограничены. Это связано с тем, что выполнение разложения Холецкого для матриц ковариации больших размеров требует больших вычислительных ресурсов, а в некоторых случаях может происходить утрата свойств положительной определенности за счет потери значимости при выполнении вычислений. В этом случае точное решение для матрицы A при достаточно больших значениях N может быть не найдено, что требует проведения дополнительного анализа[1].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае

обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и ход выполнения работы

Выполним программную реализацию указанного алгоритма моделирования случайного вектора в среде MATLAB. Возможности этой среды предоставляют доступ к стандартной функции для выполнения факторизации Холецкого обычных и разреженных матриц. Разложение Холецкого возможно как для действительных положительно определенных, так и для комплексных эрмитовых матриц. Оператор $A = \text{chol}(C)$ для квадратной симметричной матрицы C возвращает верхнюю треугольную матрицу A такую, что $A^*A=C$. Если матрица C не является положительно определенной матрицей, выдается сообщение об ошибке. При выполнении обращения $[A \ p] = \text{chol}(C)$ с двумя выходными аргументами функция фактически генерирует сообщение об ошибке в ходе выполнения разложения Холецкого квадратной матрицы C . Если C положительно определенная матрица, то $p=0$, а A совпадает с вышеописанным случаем, иначе p - положительное целое число, а A - верхняя треугольная матрица порядка $q=p-1$, такая, что $A^*A = C(1:q,1:q)$. По сути, это означает, что функция $\text{chol}(C)$ самостоятельно выбирает размерность той части входной матрицы, для которой еще выполняются условия положительной определенности. С учетом этого можно предложить для использования следующую функцию, обеспечивающую генерацию реализации случайного вектора $\xi = Ax$:

```

function [y N]=randncor(m,n,C)

%функция для генерации совокупности p гауссовых случайных векторов
%с нулевым математическим ожиданием и матрицей ковариации C
%m-исходный порядок матрицы ковариации C (размер m*m)
%N-размер (длина) генерируемого вектора; n-количество генерируемых векторов
%формирование верхней треугольной матрицы разложения Холецкого
%с определением порядка матрицы и длины генерируемых векторов
[A p]=chol(C);

if p==0 N=m; else N=p-1; end; %длина генерируемых векторов
%генерация матрицы N*n гауссовых независимых случайных величин
x=randn(N,n);

%получение матрицы p гауссовых коррелированных векторов длиной N
y=A'*x;

```

Рассмотрим иллюстративный пример, позволяющий оценить работу алгоритма, если необходимые условия для проведения факторизации в полном объеме не выполняются:

```

%генерация реализаций случайного вектора
m=5; %m-исходный порядок матрицы ковариации
n=100000; %количество генерируемых векторов

C=[3 2.9 2 2 2; 2.9 3 2.9 2 2; 2 2.9 3 2.9 2; 2 2 2.9 3 2.9; 2 2 2 2.9 3];
[A p]=chol(C);
R=A'*A; %отображение той части матрицы, для которой проводится факторизация
[y N]=randncor(m,n,C);
%N - порядок фрагмента матрицы ковариации,
%для которого проводится факторизация

% выборочная оценка фрагмента матрицы ковариации
s=zeros(N,N);
for i=1:n
    s=s+y(:,i)*y(:,i)';
end;
s=s/n;

```

В данном m-файле реализуется генерация 100 тыс. реализаций гауссовых векторов и проводится оценка выборочной матрицы ковариации. При этом их длина оказывается меньше требуемой, так как задаваемая матрица C размера 5x5 не является положительно определенной и решение оказывается возможным только для левого верхнего фрагмента C размера 2^2 :

$$C =$$

3.0000	2.9000	2.0000	2.0000	2.0000
2.9000	3.0000	2.9000	2.0000	2.0000
2.0000	2.9000	3.0000	2.9000	2.0000
2.0000	2.0000	2.9000	3.0000	2.9000

$$R =$$

3.0000	2.9000
2.9000	3.0000

$$s =$$

3.0021	2.9003

2.0000	2.0000	2.0000	2.9000	3.0000

p = 3

2.9003	2.9979

N = 2

Тестирование алгоритма генерации ГСП с использованием функции randncor проводилось для процессов с функциями корреляции следующих видов:

$$R(\tau) = \sigma^2 \exp(-\alpha\tau); R(\tau) = \sigma^2(1 - \alpha|\tau|), \alpha|\tau| \leq 1; R(\tau) = \sigma^2 \exp(-\alpha^2\tau^2) \quad (14.7)$$

При этом для процессов первого и второго типа были получены реализации длиной $N > 5000$ при практически любых значениях нормированного коэффициента корреляции соседних значений процесса $r = \rho(\Delta t) : r = \exp(-\alpha\Delta t); r = (1 - \alpha\Delta t)$. Для процесса третьего типа аналогичные результаты получены для всех значений $r = \exp(-\alpha^2\Delta t^2) \leq 0,1$, однако уже при значениях $r = 0,05$ порядок используемой матрицы и, соответственно, длина генерируемого вектора составили $N = 48$, а при значениях $r = 0,01$ - только $N = 11$. Программа генерации ГСП представлена ниже[1].

```
%генерация реализаций процесса
m=100; % m-исходный порядок матрицы ковариации
n=100000; % количество генерируемых векторов
D=1; % дисперсия процесса
alfa=0.25; % параметр корреляционной функции процесса
dt=1; % интервал дискретизации по времени
alfdt=alfa*dt;
r=exp(-(alfdt)^2);
% r=exp(-alfdt);
% r=(1-alfdt);
%формирование матрицы ковариации элементов процесса
C=zeros(m,m);
%задание вектора коэффициентов ковариации (верхней строки матрицы ковариации)
R=zeros(1,m);
for i=1 :m,
    R(i)=D*exp(-(i-1)*alfdt)^2;
    % R(i)=D*exp(-(i-1)*alfdt);
    % h=1-(i-1)*alfdt;
    % if h>0 R(i)=D*h; end;
end;
%формирование матрицы ковариации на основе функции получения теплицевой матрицы
C=toeplitz(R);
%получение матрицы n гауссовых коррелированных векторов длиной N каждый
[y N]=randncor(m,n,C);
```

Вопросы для защиты работы:

1. Дайте краткую характеристику моделированию случайных процессов
2. Каким недостатком обладает рассматриваемый подход? С чем это связано?

ЛАБОРАТОРНАЯ РАБОТА №15

Моделирование случайного процесса на основе алгоритма скользящей суммы

Цель и содержание: изучение основных принципов компьютерного моделирования случайного процесса на основе алгоритма скользящей суммы

Теоретическое обоснование

Алгоритм скользящей суммы (СС) предполагает, что по отношению к совокупности независимых значений стандартной гауссовой случайной величины $\{x[k]\}$ при получении последовательности $\xi[n]$ коррелированной по заданному закону, выполняется линейное преобразование, которое записывается в виде СС:

$$\xi[n] = \sum_{k=1}^N a_k x[n-k], \quad (15.1)$$

где $a_k, k = \overline{1, N}$ - совокупность коэффициентов, которые при построении алгоритма нужно найти исходя из вида корреляционной функции $R(x)$. Уравнение (15.1) описывает реакцию линейного фильтра с передаточной в смысле дискретного преобразования Лапласа (z-преобразования) характеристикой

$$K(z) = \sum_{k=1}^N a_k z^{-k} \quad (15.2)$$

Для решения указанной задачи возможно применение различных подходов и методов, как аналитических, так и численных.

Рассмотрим сначала метод разложения спектральной плотности, использование которого позволяет в ряде случаев получить аналитические соотношения для расчета коэффициентов $a_k, k = \overline{1, N}$. Пусть вместо функции $R(\tau)$ изначально дан энергетический спектр процесса $\xi(t)$

$$G(\omega) = 2 \int_0^{\infty} R(\tau) \cos \omega \tau d\tau \quad (15.3)$$

Пусть также расположение $G(\omega)$ на оси частот ограничено интервалом $[-\omega_r, \omega_r]$, т. е. $G(\omega) \approx 0$, при $|\omega| > \omega_r$. Тогда $\xi(t)$ можно рассматривать как результат воздействия ГСП $x(t)$ с энергетическим спектром $G_x(\omega) = G_0, |\omega| \leq \omega_r, G_x(\omega) = 0, |\omega| > \omega_r$ на линейную систему с импульсной характеристикой

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} K_0(j\omega) e^{j\omega t} d\omega = \frac{1}{\pi} \int_0^{\omega_r} \left[\frac{1}{G_0} G(\omega) \right]^{1/2} \cos \omega t d\omega \quad (15.4)$$

где $K_0(j\omega)$ - частотная передаточная функция, удовлетворяющая соотношению $G_0 |K_0(j\omega)|^2 = G(\omega)$. Тогда, используя разложение $\xi(t)$ и $x(t)$ в ряд Котельникова с шагом $\Delta t = \frac{\pi}{\omega_r}$, можно получить

$$\begin{aligned} \xi[n] = \xi(n\Delta t) &= \int_{-\infty}^{+\infty} h(\tau) x(n\Delta t - \tau) d\tau = \sum_{k=-\infty}^{\infty} \tilde{a}_k x[n - k], \tilde{a}_k \\ &= \frac{1}{\omega_r} \int_0^{\omega_r} \left[\frac{1}{G_0} G(\omega) \right]^{1/2} \cos \frac{k\pi\omega}{\omega_r} d\omega, k = \dots, -1, 0, 1, \dots \end{aligned} \quad (15.5)$$

С учетом условия нормировки для мощности входного процесса

$$\sigma_x^2 = \frac{1}{\pi} \int_0^{\omega_r} G_0 d\omega = 1, G_0 = \pi/\omega_r \quad (15.6)$$

и быстрой сходимости значений $\tilde{a}_k$ к нулю при $k \rightarrow \infty$ окончательно можно получить искомую форму алгоритма моделирования

$$\xi[n] = \sum_{k=1}^N a_k x[n - k], a_k = \tilde{a}_{k-p-1}, k = \overline{1, N}, N = 2p + 1 \quad (15.7)$$

Параметр p здесь ограничивает число весовых коэффициентов ряда (15.5) с методической погрешностью ε :

$$\left| 1 - \frac{1}{\sigma^2} \sum_{-p}^p \tilde{a}_k \right| < \varepsilon \quad (15.8)$$

где σ^2 дисперсия $\xi(t)$.

Пример 15.1. Пусть $R(\tau) = \sigma^2 e^{-a^2 \tau^2}$. Тогда

$$G(\omega) = 2 \int_0^{\infty} R(\tau) \cos \omega \tau d\tau = \frac{\sqrt{\pi} \sigma^2}{\alpha} e^{-\frac{\omega^2}{\alpha^2 4}} \quad (15.9)$$

$$\tilde{a}_k = \frac{1}{\omega_r} \int_0^{\omega_r} \left[\frac{\omega_r \sqrt{\pi} \sigma^2}{\pi \alpha} e^{-\frac{\omega^2}{\alpha^2 4}} \right]^{1/2} \cos \frac{k \pi \omega}{\omega_r} d\omega = \int_0^1 S_0(x) \cos k \pi x dx, \quad (15.10)$$

$$S_0(x) = \left[\frac{\omega_r}{\pi} G(\omega_r x) \right]^{1/2}. \quad (15.11)$$

Используя табличный интеграл

$$\int_0^{\infty} e^{-\beta x^2} \cos b x dx = 1/2 \sqrt{\frac{\pi}{\beta}} e^{-\frac{b^2}{4\beta}}, \quad (15.12)$$

Окончательно можно получить

$$\tilde{a}_k = \frac{\sigma \sqrt{2\alpha \Delta t}}{\sqrt[4]{\pi}} e^{-2k^2 \alpha^2 \Delta t^2}, k = -p, \dots, -1, 0, 1, \dots, p. \quad (15.13)$$

Рассмотренный метод приводит к приближенным формулам для определения коэффициентов скользящего суммирования и требует анализа уровня погрешности исходя из соотношения параметров корреляционной функции, интервала дискретизации и количества коэффициентов скользящей суммы. Кроме того, полученные при интегрировании выражения могут быть использованы не для всех ситуаций, что также требует проведения дополнительного анализа.

Более общим представляется численный метод получения коэффициентов скользящей суммы путем решения системы нелинейных уравнений, связывающих значения корреляционной функции и коэффициенты СС. Очевидно, что перемножая $\xi[n] \xi[n - k + 1]$, $k = \overline{1, N + 1}$, и выполняя соответствующие операции усреднения $c_{1k} = M\{\xi[n] \xi[n - k + 1]\}$, можно записать

$$\begin{aligned} c_{11} &= R(0) = a_1^2 + a_2^2 + \dots + a_N^2, \\ c_{12} &= R(\Delta t) = a_1 a_2 + a_2 a_3 + \dots + a_{N-1} a_N, \end{aligned} \quad (15.14)$$

$$\begin{aligned} & \dots \\ c_{1N} &= R((N-1)\Delta t) = a_1 a_N, \\ c_{1N+1} &= R(N\Delta t) = 0. \end{aligned}$$

В матричном виде эту систему можно переписать таким образом:

$$\begin{pmatrix} c_{11} \\ c_{12} \\ \vdots \\ c_{1N} \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & \dots & a_{N-1} & a_N \\ a_2 & a_3 & \dots & a_N & 0 \\ & & \dots & & \\ a_N & 0 & \dots & 0 & 0 \end{pmatrix} \times \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix}. \quad (15.15)$$

Искомые коэффициенты при известной корреляционной функции могут быть получены путем решения матричного нелинейного уравнения вида (15.15). В ставшей уже классической работе [2] указывалось, что решение данной системы связано с определенными трудностями вычислительного характера. Следует также отметить, что уравнения (15.15) однозначно позволяют решить обратную задачу - нахождения значений корреляционной функции по коэффициентам СС[1].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Современная реализация среды MATLAB предоставляет возможности для нахождения значений корреляционной функции по коэффициентам СС. Для решения матричного нелинейного уравнения вида (15.15) в ней можно

использовать стандартную функцию `fsolve`, реализующую численное решение нелинейных уравнений вида $F(x) = 0$, где x - вектор, который требуется найти. Решение осуществляется на основе метода Гаусса-Ньютона или Левенберга-Марквардта, а сама функция имеет большое количество настроек, которые могут быть установлены после изучения «помощи», предоставляемой средой. Оказалось, что использование стандартных средств MATLAB дает необходимый результат для достаточно широкого перечня случаев. Программа, с помощью которой проводилось исследование возможностей получения искоемых коэффициентов СС путем решения уравнений (15.15) и сравнения полученных результатов с результатами, получаемыми на основе аналитического решения вида (15.5), имеет следующий вид:

```
%расчет коэффициентов скользящей суммы путем решения системы нелинейных уравнений
%N - количество коэффициентов скользящей суммы
%R - массив (вектор-строка) значений корреляционной функции длиной N
%a - массив (вектор-столбец) коэффициентов скользящей суммы длиной N
global R N;
N=25;%N=2*p+1, нечетное число
D=1;%дисперсия процесса
alf=0.1;%параметр корреляционной функции
dt=1;%интервал дискретизации по времени
alfdt=alf*dt;
sigma=sqrt(D);
%задание массива коэффициентов ковариации (верхней строки матрицы ковариации)
for i=1:N,
    R(i)=D*exp(-(alfdt*(i-1))^2);%первый вариант
    % R(i)=D/(1+(alfdt*(i-1))^2); %второй вариант
    % R(i)=D*(1-alfdt*(i-1)); %третий вариант
end;
%решение системы нелинейных уравнений с использованием функции fsolve
Nm=(N+1)/2;
a0=zeros(N,1)+0.1*sigma; % задание начального вектора решения
a0(Nm)=sigma;
%задание опций и параметров алгоритма решения методом Гаусса-Ньютона
options = optimset('Display','iter','TolFun',1e-16);
[a, fval] = fsolve('myf',a0, options);
a1=a'; %запись решения в виде строки
```

```

%расчет коэффициентов с помощью соотношений (4.2)
for k=1:N,
a2(k)=sigma*(sqrt(2*alfdt)/(pi^0.25))*exp(-2*(alfdt*(k-Nm))^2);
% a2(k)=2*sigma*alfdt*1.5*(1-(-1)^abs(k-Nm)*exp(-alfdt/2))/(alfdt^2+4*pi^2*(k-Nm)^2);
% a2(k)=sigma*sqrt(alfdt); %alfdt=1/N
end;
E1=a1*a1'; %дисперсия генерируемого процесса на основе (4.3)
E2=a2*a2'; %дисперсия генерируемого процесса на основе (4.2)
x=-(N-1)/2:(N-1)/2; %визуализация значений полученных коэффициентов на графике
figure;
plot(x,a1,'--rs',x,a2,'b'); xlabel('k = -(N-1)/2,...,(N-1)/2'); ylabel('a[k]');
title(['N=',num2str(N), ' alf=',num2str(alf), ' E1 =',num2str(E1), ' E2=',num2str(E2)]);

```

Для выполнения решения с помощью функции `fsolve` должна быть описана функция, определяющая запись нелинейной системы уравнений в виде $F(x) = 0$. В данном случае такая функция имеет вид

```

function F=myf(x)
% функция, определяющая запись системы нелинейных уравнений (4.3)
% R - массив значений корреляционной функции длиной N
% x - массив (вектор-столбец) коэффициентов скользящей суммы длиной N
global R N;
A=zeros(N);
for i=1:N,
    for j=1:N-i+1,
        A(i,j)=x(i+j-1);
    end;
end;
F=R'-A*x;

```

При выполнении основной программы выводится не только вектор решения a , но и результирующие значения функции $F(a)$ - $fval$, которые дают погрешность решения, значения дисперсии генерируемых процессов, получаемых на основе (15.15) - $E1$ и на основе (15.5) - $E2$, а также графики, отображающие полученные коэффициенты в зависимости от номера.

При проведении расчетов рассматривались процессы с функциями корреляции следующих видов:

$$R(\tau) = \sigma^2 \exp(-\alpha^2 \tau^2), R(\tau) = \sigma^2 / (1 + \alpha^2 \tau^2).$$

Для них используются следующие аналитические соотношения:

$$\tilde{a}_k = \frac{\sigma \sqrt{2\alpha \Delta t}}{\sqrt[4]{\pi}} e^{-2k^2 \alpha^2 \Delta t^2}, \tilde{a}_k = \frac{2\sigma(\alpha \Delta t)^{\frac{3}{2}}}{(\alpha \Delta t)^2 + 4\pi^2 k^2} \left(1 - (-1)^k e^{-\frac{\alpha \Delta t}{2}}\right),$$

$$\tilde{a}_k = \frac{\sigma}{N^{\frac{1}{2}}}, \quad k = -p, \dots, -1, 0, 1, \dots, p.$$

Полученные таким образом решения в программе отображаются в массиве $a2$. На рисунке 15.1 представлены результаты расчетов для первого варианта функции корреляции. Эти результаты показывают, что при больших значениях величины $\alpha\Delta t$ решения на основе (15.5) и (15.15) совпадают по форме зависимости; при этом соотношения (15.5) имеют некоторую методическую погрешность, отражающуюся в увеличении дисперсии процесса. В тоже время, при малых $\alpha\Delta t$ численные решения отличаются по форме от аналитической зависимости; при этом аналитические соотношения (15.5) дают потери дисперсии, а результаты численного решения (15.15) – ее увеличение.

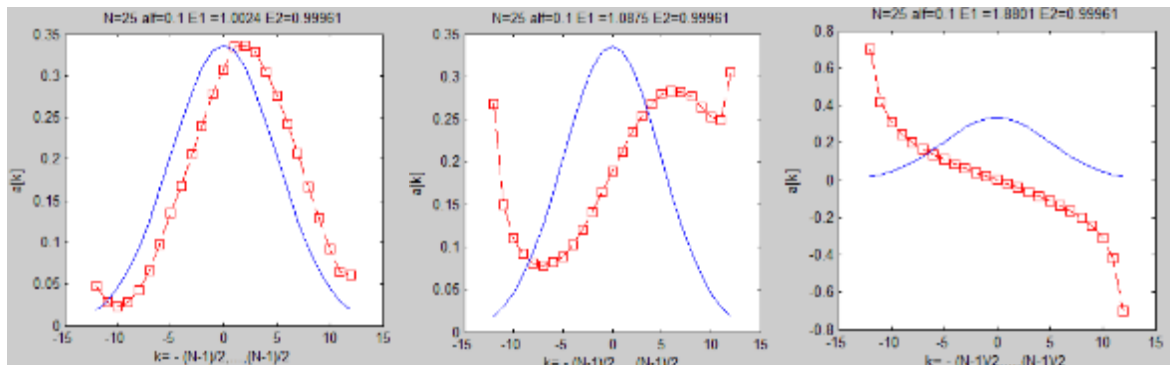


Рисунок 15.1. Результаты расчетов коэффициентов скользящей суммы

После получения коэффициентов СС тем или иным методом в виде массива a , генерация реализаций процесса в цикле по p реализациям получается достаточно просто с использованием стандартной функции среды MATLAB раздела SignalProcessingToolbox - filter (могут также использоваться функции conv, convmtx), реализующей соотношение (15.1) при следующем обращении к ней[1]:

```
%генерация реализаций процесса
m=100; %длина генерируемой реализации
n=10000; %количество генерируемых реализаций
zi=sqrt(D)*randn(N-1,1);%задание вектора начальных значений
yz=zeros(m,n);%предварительное выделение памяти
for k=1:n
    x=randn(m,1);% генерация массива гауссовых независимых случайных величин
    [yz(:,k) zf]=filter(a,1,x,zi);%генерация реализации для вектора коэффициентов a длиной N
    zi=zf; % вектор конечных значений используется как начальный для следующей реализации
end;
```


Вопросы для защиты работы:

1. Что из себя представляет алгоритм скользящей суммы?
2. Какие подходы используются в этом алгоритме? В чем их особенности?

ЛАБОРАТОРНАЯ РАБОТА №16

Моделирование случайного процесса на основе алгоритма авторегрессии скользящей суммы

Цель и содержание: практическое изучение методов моделирования случайного процесса на основе алгоритма авторегрессии скользящей суммы в среде MATLAB.

Теоретическое обоснование

Алгоритм авторегрессии - скользящей суммы (АР-СС). Данный алгоритм при генерации ГСП в общем случае реализует рекуррентное уравнение вида

$$\xi[n] = - \sum_{k=1}^p b_k \xi[n-k] + \sum_{k=0}^q a_k x[n-k], \quad n = 1, 2, \dots \quad (16.1)$$

Модель процесса в форме (16.1) называется моделью авторегрессии - скользящей суммы, так как в ней комбинируется первая часть, являющаяся авторегрессией для $\xi[n]$, и вторая часть, реализующая вычисление СС для входной последовательности независимых гауссовых стандартных величин $\{x[k]\}$. В итоге, уравнение (16.1) описывает реакцию линейного фильтра с передаточной в смысле дискретного преобразования Лапласа (z - преобразования) характеристикой

$$K_*(z) = \frac{\sum_{k=0}^q a_k z^{-k}}{1 + \sum_{k=1}^p b_k z^{-k}} \quad (16.1)$$

Задача построения модели ГСП состоит в нахождении коэффициентов a_k и b_k по виду корреляционной функции $R(\tau)$. Для получения алгоритмов данного класса также возможно применение как аналитических, так и численных методов.

Рекуррентные алгоритмы (16.1) пригодны для моделирования процессов $\xi(t)$ с рациональной спектральной плотностью $G(\omega) = G_1(\omega)/G_2(\omega)$, где $G_1(\omega)$,

$G_2(\omega)$ - полиномы степени q' и p' относительно ω . Такой процесс получается при прохождении гауссового белого шума через линейную систему с сосредоточенными параметрами и передаточной функцией $K(j\omega) = K_1(j\omega) / K_2(j\omega)$, где $K_1(j\omega)$, $K_2(j\omega)$ полиномы относительно ω со степени q и $p > 1$.

Рассмотрим получение алгоритма вида (18.1) методом факторизации. В этом методе для нахождения $K_*(z) = A(z)/B(z)$ передаточной функции, формирующей соответствующий дискретный процесс $\xi[n]$, необходимо сначала найти его спектральную плотность:

$$F(z) = \sum_{n=-\infty}^{+\infty} R[n]z^{-n}, R[n] = R(n\Delta t), \quad (16.3)$$

а затем с учетом того, что $F(z) = |K_*(z)|^2$, провести факторизацию, т. е. представить с помощью алгебраического разложения полиномов в виде сомножителей

$$F(z) = |K_*(z)|^2 = \frac{A(z)}{B(z)} \cdot \frac{A(z^{-1})}{B(z^{-1})} \quad (16.4)$$

$$A(z) = a_0 + \dots + a_q z^{-q}, \quad B(z) = 1 + b_1 z + \dots + b_p z^{-p}.$$

Для получения $F(z)$ удобно воспользоваться соотношением

$$F(z) = F^+(z) + F^+(z^{-1}) - R(0), \quad F^+(z) = \sum_{n=0}^{\infty} R[n]z^{-n}.$$

Пример 16.1. Пусть $R(\tau) = \sigma^2 e^{-\alpha|\tau|}$, $G(\omega) = 2\sigma^2\alpha/(\alpha^2 + \omega^2)$. Тогда

$$F^+(z) = \sigma^2 \sum_{n=0}^{\infty} z^{-n} e^{-\alpha\Delta t n} = \frac{\sigma^2}{1 - rz^{-1}}, \quad r = e^{-\alpha\Delta t}; \quad (16.5)$$

$$F(z) = \sigma^2 \frac{1 - r^2}{(1 - rz)(1 - rz^{-1})} = \frac{\sigma\sqrt{1 - r^2}}{1 - rz} \cdot \frac{\sigma\sqrt{1 - r^2}}{1 - rz^{-1}}; \quad (16.6)$$

$$K_*(z) = \frac{\sigma\sqrt{1 - r^2}}{1 - rz^{-1}}; \quad a_0 = \sigma\sqrt{1 - r^2}; \quad b_1 = -r. \quad (16.7)$$

В итоге получен простейший рекуррентный алгоритм моделирования реализации $\xi[n]$:

$$\xi[n] = r\xi[n - 1] + \sigma\sqrt{1 - r^2}x[n] \quad (16.8)$$

Рассмотрим теперь возможности нахождения коэффициентов в уравнении АР-СС численным методом. Для этого так же, как и в случае модели СС, выполняя всевозможные усреднения вида $M\{\xi[n]\xi[n-k+1]\}$, можно связать значения корреляционной функции процесса с этими коэффициентами. Интересно, что при определенных значениях временного сдвига параметры авторегрессионной части можно найти отдельно и независимо от параметров скользящей суммы как решение системы линейных уравнений. Такие уравнения называются нормальными уравнениями Юла-Уокера и имеют следующий вид:

$$\begin{pmatrix} R[q+1] \\ R[q+2] \\ \vdots \\ R[q+p] \end{pmatrix} = \begin{pmatrix} R[q] & R[q-1] & \dots & R[q-p+2] & R[q-p+1] \\ R[q+1] & R[q] & \dots & R[q-p+3] & R[q-p+2] \\ \vdots & \vdots & \dots & \vdots & \vdots \\ R[q+p-1] & R[q+p-2] & \dots & R[q+1] & R[q] \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} \quad (16.9)$$

где, как и ранее, $R[n] = R(n\Delta t)$, $R[n] = R[-n]$. Отметим, что матрица в (16.9), составленная из коэффициентов ковариации процесса, является теплоцевой. Одновременно необходимо отметить, что значения параметров СС части модели не являются решениями линейных уравнений и связаны со значениями корреляционной функции, как и в (6.4), нелинейными уравнениями. Вид этих уравнений по аналогии при известном векторе коэффициентов $b = (1, b_1, b_2, \dots, b_p)^T$ можно получить, если ввести

$$\begin{aligned} \eta[n] &= \xi[n] + \sum_{k=1}^p b_k \xi[n-k] = b^T \xi^n = \\ &= \sum_{k=0}^q a_k x[n-k], \quad \xi^n = (\xi[n], \xi[n-1], \dots, \xi[n-p])^T \end{aligned} \quad (16.10)$$

и решать уравнения вида (16.9) для коэффициентов ковариации процесса $\eta[n]$, которые определяются как соответствующие элементы

$$c_{\eta 1k} = M\{\eta[n]\eta[n-k+1]\} = b^T M[\xi^n (\xi^{n-k-1})^T] b \quad (16.11)$$

Тем не менее, можно избежать сложностей, связанных с решением нелинейных уравнений, если ограничиться использованием только АР-модели процесса

$$\xi[n] = - \sum_{k=1}^p b_k \xi[n-k] + a_0 x[n], \quad n = 1, 2, \dots \quad (16.12)$$

Для АР-модели нормальные уравнения Юла-Уокера имеют следующий, более упрощенный, вид:

$$\begin{pmatrix} a_0^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} R[0] & R[-1] & \dots & R[-2] & R[-p] \\ R[1] & R[0] & \dots & R[-p+2] & R[-p+1] \\ \vdots & \vdots & \dots & \vdots & \vdots \\ R[p] & R[p-1] & \dots & R[1] & R[0] \end{pmatrix} \times \begin{pmatrix} 1 \\ b_1 \\ \vdots \\ b_p \end{pmatrix} \quad (16.13)$$

Отметим, что при $p = 1$ решение (16.13) однозначно дает уравнение вида (16.9), где $r = \rho(\Delta t)$ – коэффициент корреляции процесса. Аналогично, следует отметить, что с использованием уравнений вида (16.13) может быть решена обратная задача – нахождения значений корреляционной функции по коэффициентам АР-модели[1].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

В общем случае решение, позволяющее одновременно определить коэффициенты $b_1, b_2, \dots, b_p, a_0$ с учетом того, что матрица является треугольной, может быть получено на основе алгоритма Левинсона. В среде MATLAB для этого имеется стандартная функция `levinson`. Эта функция выполняет решение уравнений Юла-Уокера вида (18.9) при $q=0$. Решение выдается в

виде вектора-строки $b = (1, b_1, b_2, \dots, b_p)$. Получив такое решение, можно рассчитать значение

$$a_0^2 = R[0] + \sum_{k=1}^p R[k]b_k.$$

Пример выполнения подобных вычислений с последующим проведением генерации реализации процесса на основе функции filter представлен ниже.

```
%нахождение параметров АР-модели
p=24;%параметр авторегрессии
D=1;%дисперсия процесса
alf=0.3;%параметр корреляционной функции
dt=1;%интервал дискретизации по времени
alfdt=alf*dt;
%задание массива коэффициентов ковариации (верхней строки матрицы ковариации)
for i=1:p+1,
    R(i)=D*exp(-(alfdt*(i-1))^2);%первый вариант
    %R(i)=D/(1+(alfdt*(i-1))^2); %второй вариант
    %R(i)=D*exp(-alfdt*(i-1)); %третий вариант
end;
b = levinson(R,p); %расчет вектора коэффициентов авторегрессии
a0=sqrt(R*b');
%генерация реализации процесса
m=100;%длина генерируемой реализации
n=10000;%количество реализаций
yz=zeros(m,n);%предварительное выделение памяти
zi=sqrt(D)*randn(p,1);%задание вектора начальных значений
for k=1:n
    x=randn(m,1);% генерация массива гауссовых независимых случайных величин
    [yz(:,k) zf]=filter(a0,b,x,zi);%генерация реализации для коэффициентов a0,b
    zi=zf;% вектор конечных значений используется как начальный для следующей реализации
end;
```

При использовании функции filter в этой лабораторной работе, так же как и в предыдущей, особое значение играет задание начальных условий генерации. Анализ показывает, что начальный участок получаемой при первом обращении реализации искажает статистические характеристики генерируемого процесса. Это приводит к необходимости использования достаточно длинной реализации или, как в рассмотренных примерах, использование специальных установок для задания начальных значений при генерации нескольких реализаций. Следует также отметить, что АР-алгоритм при определенных (достаточно больших) значениях коэффициента корреляции процесса для достижения приемлемой точности воспроизведения

процессов с заданными характеристиками требует увеличения порядка вектора b , что, в свою очередь, в ряде случаев ведет к потере устойчивости уравнения АР. Поэтому для алгоритмов этого класса также необходимо проводить дополнительный анализ получаемых решений[1].

Вопросы для защиты работы:

1. Что из себя представляет алгоритм авторегрессии скользящей суммы?
2. Опишите получение АР-алгоритма методом факторизации.

ЛАБОРАТОРНАЯ РАБОТА №17

Компьютерное моделирование информационных процессов

Цель работы: изучение основных принципов компьютерного моделирования случайных процессов; проведение сравнительного анализа алгоритмов.

Теоретическое обоснование

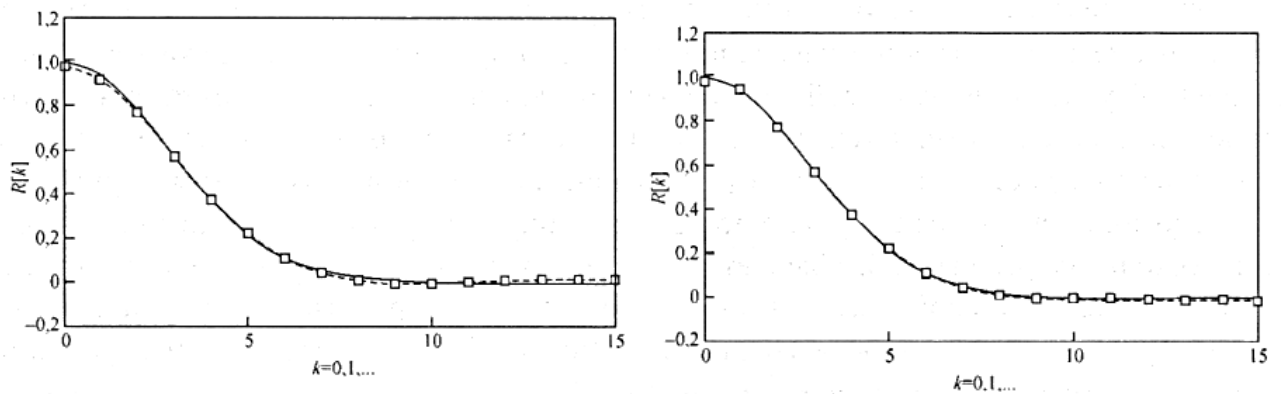
Учитывая многообразие возможных реализаций алгоритмов генерации реализаций ГСП возникает задача их сравнения и выработки рекомендаций по их применению. При проведении сравнительного анализа следует учитывать различные факторы: диапазон применяемых параметров процесса; наличие методических погрешностей; необходимость проведения дополнительного анализа получаемых алгоритмов; затрачиваемый вычислительный ресурс.

В плане качественных характеристик алгоритм, основанный на проведении факторизации матрицы ковариации, обладает ограничениями по длине генерируемой реализации, что является его главным недостатком. В то же время в нем фактически отсутствует методическая погрешность, он инвариантен к виду корреляционной функции процесса, а главное при его использовании практически не требуется проведение дополнительного анализа: допустимое решение формируется автоматически при определении порядка матрицы, для которой еще возможна факторизация.

Алгоритм СС не ограничивает длину формируемой реализации, что является его безусловным преимуществом. При этом он требует решения нелинейных уравнений или использования аналитических соотношений для коэффициентов СС с проведением последующего анализа получаемой методической погрешности. Алгоритм также имеет ограничения по диапазону применяемых параметров генерируемого процесса.

Алгоритм АР также не ограничивает длину формируемой реализации. Алгоритм в наибольшей степени ориентирован на работу с процессами, имеющими рациональную спектральную плотность, и со значительно меньшей точностью обеспечивает получение реализаций процессов других классов. При увеличении порядка АР требуется проводить исследование его устойчивости.

На рисунке 20.1 представлены результаты статистического анализа функции корреляции реализаций одного и того же процесса, генерируемых с использованием этих алгоритмов. Сплошной линией на рисунках показана зависимость истинной функции $= \exp(-0,25k^2)$, $\Delta t = 1$, $k = 0, 1, 2, \dots$, а штрих-пунктирной — ее оценка, полученная путем моделирования 10 тыс. реализаций длиной 100 отсчетов каждая. Для первого алгоритма при этом порядок получаемой после факторизации матрицы также равнялся 100 (т. е. ограничения не присутствовали), для второго алгоритма количество используемых коэффициентов СС составило $N = 21$, для третьего алгоритма порядок авторегрессии (количество коэффициентов вектора b) задавалось равным $p = 8$.



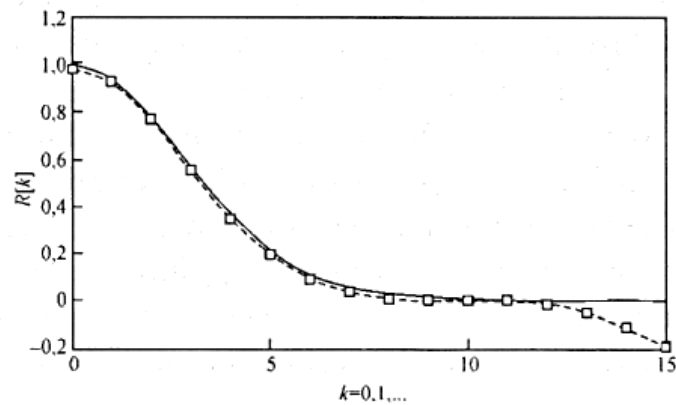


Рисунок 20.1. Результаты статистического анализа функции корреляции для реализаций процесса, генерируемых с использованием различных алгоритмов

Анализ представленных зависимостей показывает, что первые два алгоритма обеспечивают примерно одинаковую погрешность при генерации процесса, тогда как второй имеет явную погрешность, что и не удивительно, ведь его спектральная плотность не является рациональной.

С точки зрения затрачиваемого времени вычислений при сравнении алгоритмов требуется анализировать время, затрачиваемое на выполнение этапа подготовительных вычислений, и время, затрачиваемое на этапе непосредственной генерации заданного количества реализаций.

Существует ряд функций MATLAB, позволяющих оценить затрачиваемое время вычисления того или иного алгоритма. Особый интерес представляет пара функций *tic* и *toc*, позволяющих вычислить время выполнения системой той или иной операции. Если в командной строке набрать приведенную ниже последовательность команд:

tic, операция, toc

где операция - это команда или набор команд, то будет отображен не только результат выполнения заданной операции, но и приблизительное время ее выполнения (в секундах). При этом функция *tic* запускает секундомер, а функция *toc* выводит вычисленное время. Это время зависит от нескольких различных факторов и может изменяться, поэтому для получения

более точных значений времени следует провести несколько измерений, а результаты усреднить.

Функции *tic* и *toc* очень полезны, когда нужно выбрать наиболее эффективную реализацию решения одной и той же задачи.

Также в MATLAB есть возможность измерить затраченное время центрального процессора. Для этого используется функция *cputime*:

```
Start = cputime;  
... (операции);  
Elapsed = cputime - Start
```

затрачиваемое на вычисление каждой строки.

Для анализа сложных алгоритмов огромную пользу оказывает инструмент MATLAB–профилировщик.

Профилирование - это процедура измерения затрат времени на выполнение строк программы. Результат таких измерений часто оказывается неожиданным, поскольку затраты оказываются максимальными вовсе не там, где предполагалось. Поэтому первоначальную реализацию программы надо делать настолько простой, насколько это возможно, а затем использовать профилировщик, чтобы выявить критические участки программы, если быстрое действие действительно является главным показателем эффективности создаваемой программы. Преждевременная оптимизация может усложнить код, не обеспечив реального повышения эффективности.

Профилировщик целесообразно использовать, чтобы выявить m-функции, которые требуют большого времени, затем определить, почему и как они вызываются и поискать способы минимизации их использования. Часто бывает полезно задаться вопросом, а требуется ли вызывать m-функцию столько раз. Поскольку программы часто имеют несколько уровней, может оказаться, что созданный код вызывает наиболее трудоемкие функции неявно. Более точно, функции внутри вашего кода могут вызывать другие функции, которые требуют большого времени и могут находиться на

более низких уровнях. В этом случае важно определить, какие из функций высшего уровня являются ответственным за такие обращения.

Часто профилировщик помогает выявить проблемы, которые могут быть решены:

- отказом от лишних вычислений, которые могут быть следствием невнимательности;
- корректировкой алгоритма, чтобы избежать вызова неэффективных m-функций;
- отказом от многократных повторных вычислений путем хранения результатов для последующего использования.

Конечная цель профилирования состоит в том, чтобы повысить быстродействие программы. Как только достигается состояние, когда наибольшее время тратится на обращения к малому числу встроенных функций, то это означает, что достигнута оптимизация кода.

Команда *profile* позволяет обратиться к тем М-функциям, которые предполагается профилировать. При каждом обращении можно профилировать только одну m-функцию. В процессе профилирования ведется отсчет интервалов в 0.01 секунду, необходимых для вычисления каждой строки.

Оператор *profile*. Эта команда позволяет запустить профилировщик из командной строки, можно использовать одну из следующих форм обращения:

profile on;

... (операции);

profile off;

profileviewer;

Опции *on* и *off* позволяют запускать и приостанавливать профилировщик[10].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение:

персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и ход выполнения работы

Выполним программную оценку времени исполнения алгоритма моделирования случайного вектора с заданной ковариационной матрицей в среде MATLAB.

В таблице 17.1 представлены результаты оценки (на ПЭВМ класса IntelCorei3 с тактовой частотой 2,13 ГГц) этих величин для рассматриваемого примера при генерации в цикле 10 тыс. реализаций с использованием представленных выше m-файлов и с предварительным выделением памяти для массива, используемого для записи реализаций.

Таблица 17.1. Затрачиваемое на выполнение вычислений время, с

Этап	Функции MATLAB	
	tic...toc	cputime
Подготовительный	0.0502	0.1563
Генерации реализаций	0.4972	0.7813

Следует также отметить, что этот алгоритм предоставляет возможность (это позволяет описанная ранее функция `randncor`) получать все реализации

сразу в матричной форме, что, собственно. При этом время генерации 10 тыс. реализаций с его использованием немного сокращается.

Если запустить профилировщик на этом алгоритме, то в результате получим данные представленные на рисунке 17.1

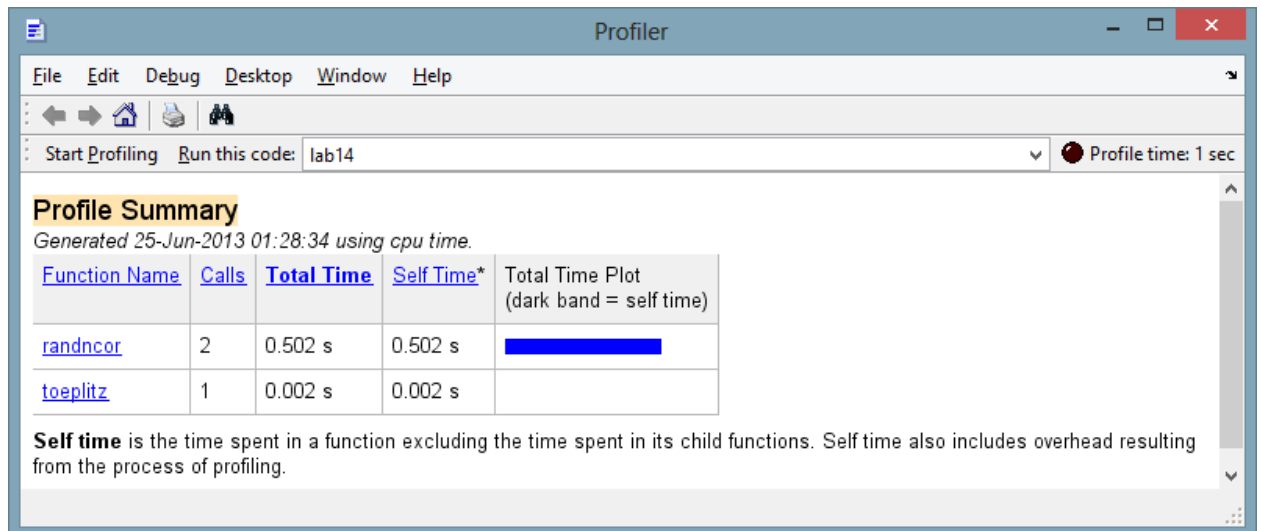


Рисунок 17.1. Результат выполнения оператора profiler на алгоритме случайного вектора с заданной ковариационной матрицей

Выполните оценку времени выполнения и затраты процессорного времени для остальных алгоритмов генерации реализаций ГСП. Проанализируйте с помощью профилировщика эти алгоритмы.

Контрольные вопросы:

1. Назовите основные алгоритмы генерации реализаций ГСП.
2. Опишите особенности каждого алгоритма
3. Какой из алгоритмов показывает наибольшую статистическую погрешность?
4. Какой из алгоритмов показывает наибольшую производительность на определенных этапах?

Какие функции требуют больше времени на выполнения в каждом из алгоритмов?

ЛАБОРАТОРНАЯ РАБОТА №18

Компьютерное моделирование скалярных случайных полей

Цель и содержание работы: изучение основных принципов компьютерного моделирования скалярных случайных полей; практическая реализация случайного поля на основе развертки случайного вектора.

Теоретическое обоснование

Скалярным случайным полем называется скалярная случайная функция нескольких переменных. Далее для определенности будем рассматривать случайные поля, заданные относительно двух переменных - координат x и y : $\xi(x, y)$. Подобные модели наиболее часто используются при описании сигналов и помех применительно к задачам обработки изображений. В основе моделирования случайных полей также лежит использование модели гауссового (нормального) поля $\xi(x, y)$ (любая совокупность его отсчетов подчиняется многомерному гауссовому распределению), имеющему математическое ожидание $m = M[\xi(x, y)] = 0$ и корреляционную функцию вида $R(x, x', y, y') = M[\xi(x, y)\xi(x', y')] = \sigma^2 \rho(x - x', y - y') = \sigma^2 \rho(x'', y'')$, где σ^2 - дисперсия, а ρ - нормированная функция корреляции. Вместо функции корреляции $R(x'', y'')$ может быть изначально задана спектральная плотность мощности $G(\lambda, \mu)$ случайного поля.

Случайное поле, функция корреляции которого имеет подобный вид, т. е. зависит только от сдвига точек, для которых определяется корреляция случайных функций, называется однородным (аналог свойства стационарности случайного процесса). Однородное случайное поле, для которого корреляционная функция имеет вид

$$R(x - x', y - y') = \sigma^2 \rho(r), r = \sqrt{(x - x')^2 + (y - y')^2}, \quad \text{называется}$$

изотропным.

Реализацию поля $\xi(x, y)$ будем обозначать символом с произвольным индексом, например $\xi^*(x, y)$. Цифровая модель гауссового однородного поля (ГОП) должна обеспечивать генерацию этих реализаций в дискретном времени в виде случайной последовательности чисел $\xi[n, m] = \xi(n\Delta x, m\Delta y)$, $n = 1, 2, \dots, m = 1, 2, \dots$, с заданными интервалами дискретизации Δx , Δy и корреляционной функцией $R[n, m] = R(n\Delta x, m\Delta y)$. Модель ГОП и реализуемый на ее основе алгоритм генерации подобных реализаций так же, как и алгоритм ГСП при моделировании случайных процессов, может использоваться для получения моделей и алгоритмов генерации случайных полей других видов. Алгоритм генерации ГОП аналогично реализуется путем выполнения некоторого линейного преобразования совокупности независимых значений стандартной гауссовой случайной величины.

Алгоритмы генерации реализации случайного поля на основе развертки случайного вектора. Данный класс алгоритмов базируется на очевидной идее представления совокупности $\xi[n, m]$, $n = \overline{1, N}$; $m = \overline{1, M}$, в пределах заданной области (рисунок 18.1) конечного размера в виде случайного вектора

$$\begin{aligned} \xi &= (\xi_{11}, \dots, \xi_{1M}, \xi_{21}, \dots, \xi_{2M}, \dots, \xi_{N1}, \dots, \xi_{NM})^T, \\ \xi_{nm} &= \xi[n, m], n = \overline{1, N}; m = \overline{1, M} \end{aligned} \quad (18.1)$$

Матрица ковариации этого вектора рассчитывается следующим образом:

$$\begin{aligned} C_\xi &= M[\xi \xi^T] = \|c_{ij}\|, c_{ij} = R[n - l, m - k], \\ i &= M(n - 1) + m, \quad n = \overline{1, N}, \quad m = \overline{1, M}; \\ j &= M(l - 1) + k, \quad l = \overline{1, N}, \quad k = \overline{1, M} \end{aligned} \quad (18.2)$$

При использовании подобной развертки для генерации реализации случайного поля можно воспользоваться одним алгоритмов из рассмотренных в предыдущих лабораторных работах.

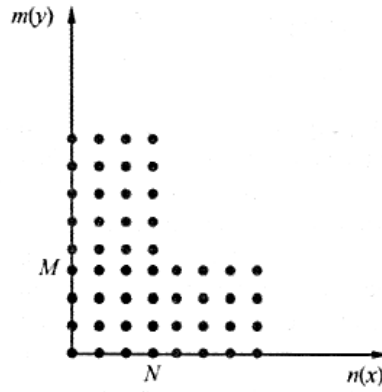


Рисунок 18.1. Иллюстрация процесса формирования фрагментов поля

Алгоритм авторегрессии. Данный класс алгоритмов базируется на использовании модели пространственной авторегрессии, которая в общем случае задается уравнением вида

$$\xi(n, m) = \sum_{\substack{k, r \in D_p \\ (k, r) \neq (m, n)}} b_{k-n, r-m} \xi[k, r] + a_0 x[n, m], \quad (18.3)$$

где $b_{k-n, r-m}$ - коэффициенты АР; $x[n, m]$ - независимые гауссовы величины, образующие случайное поле на двумерной координатной сетке; a_0 - коэффициент, определяющий дисперсию возмущения, формируемого $x[n, m]$; D_p - область уже определенных значений случайного поля, задающая окрестность точки с координатами (n, m) . Такая модель, как и аналогичная АР-модель для ГСП, является частным случаем модели АР-СС для случайных полей.

При переходе к обработке двумерных случайных полей (изображений) зачастую не удастся представить изображение эквивалентным рекуррентным разностным уравнением. Это следствие того, что в общем случае для двумерных полей отсутствует естественно определенное свойство каузальности. Такое свойство можно ввести только искусственно, указав определенный порядок просмотра элементов координатной сетки. Поэтому в зависимости от конфигурации области D_p разделяют каузальные, полукаузальные и некаузальные авторегрессионные модели случайных

полей, описываемых общим соотношением (18.3). АР-модель называется каузальной, если значения индексов $k \leq n, r \leq m$; при этом одновременно не может выполняться $k = n, r = m$.

Другими словами, это означает, что элементы поля, определенные в области D_p не упреждают вновь определяемый в точке (n, m) элемент поля. На рисунке 18.2, а и б представлены соответствующие конфигурации области D_p для каузальной модели. Для некаузальной модели ранее определенные элементы поля в области D_p упреждают по обеим координатам вновь определяемый в точке (n, m) элемент поля (область D_p как бы окружает точку (n, m) со всех сторон). Данная ситуация представлена на рисунке 18.2, в. Для полукаузальной модели такое «упреждение» происходит только по одной из координат (рисунок 18.2, г).

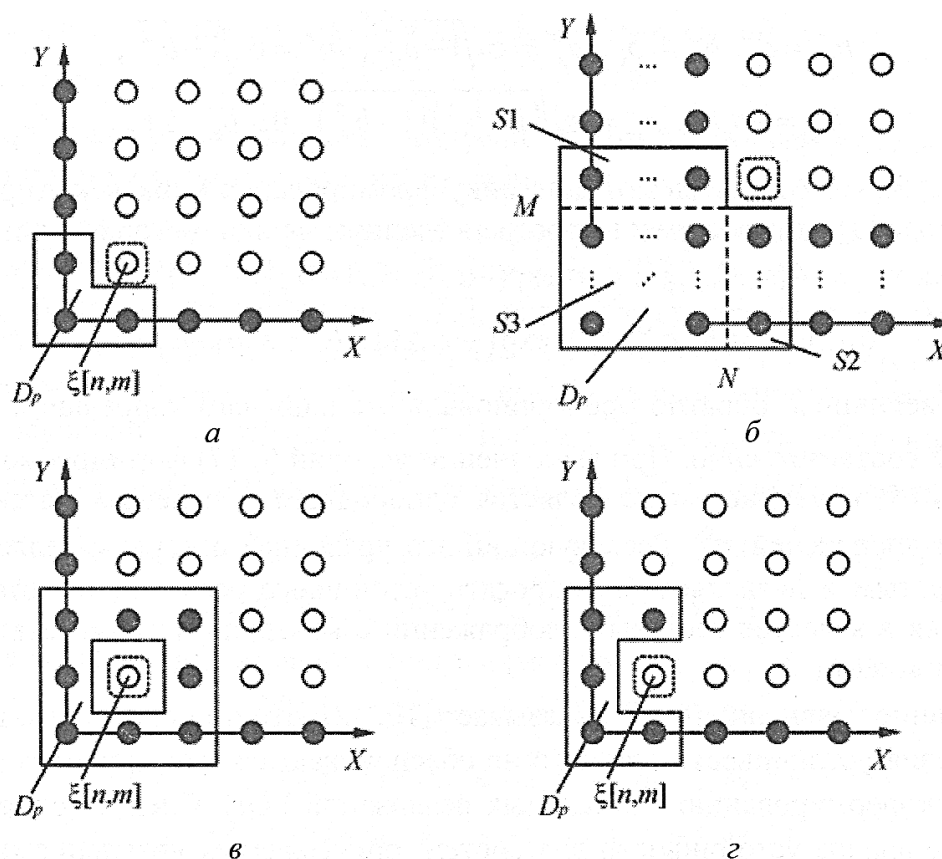


Рисунок 18.2. Конфигурация области D_p для каузальных (а, б), некаузальной (в) и полукаузальной (г) моделей

Каузальная модель формирования случайного поля в простейшем случае описывается рекуррентными уравнениями пространственной динамики следующего вида:

$$\begin{aligned} \xi(n, m) = & b_x \xi[n-1, m] + b_y \xi[n, m-1] + b_{xy} \xi[n-1, m-1] + \\ & a_{xy} x[n, m], \quad n = 2, 3, \dots, m = 2, 3, \dots, \end{aligned} \quad (18.4)$$

для которых предварительно определяются граничные условия и генерируются граничные значения на основе следующих уравнений

$$\begin{cases} \xi[1, 1] = \sigma x[1, 1] \\ \xi[n, 1] = b_x^r \xi[n-1, 1] + a_x^r x[n, 1], n = 2, 3, \dots \\ \xi[1, m] = b_y^r \xi[1, m-1] + a_y^r x[1, m], m = 2, 3, \dots \end{cases} \quad (18.5)$$

В (18.4), (18.6) $b_x^r, b_y^r, a_x^r, a_y^r$ - параметры, определяющие уравнения для задания граничных условий; b_x, b_y, b_{xy}, a_{xy} - параметры пространственной АР, используемой при формировании случайного поля. Будем называть подобную модель случайного поля трехточечной в том смысле, что упреждающая область D_p при определении очередного значения $\xi[n, m]$ состоит из трех точек (рис. 18.2, а). Выполнение алгоритма генерации предполагает сначала задание начальных (осевых) граничных значений в точках, расположенных на осях OX и OY . Затем к ним последовательно добавляются точки, располагающиеся в центральной части координатной сетки (добавление происходит при помощи уравнения (21.4)). В случае, если выполняется

$$b_x^r = b_x, b_y^r = b_y, a_x^r = \sqrt{1 - b_x^2}, a_y^r = \sqrt{1 - b_y^2} \quad (18.6)$$

$$b_{xy} = -b_x b_y, a_{xy} = \sigma \sqrt{(1 - b_x^2)(1 - b_y^2)}, 0 \leq b_{x,y} \leq 1 \quad (18.7)$$

уравнения (18.4), (18.5) описывают динамику формирования марковского разделимого случайного поля, которое имеет пространственную функцию корреляции вида

$$R(x, x', y, y') = \sigma^2 \exp(-\alpha_x |x - x'| - \alpha_y |y - y'|) \quad (18.8)$$

$$R[n, m] = \sigma^2 b_x^n b_y^m, b_x = \exp(-\alpha_x \Delta x), b_y = \exp(-\alpha_y \Delta y) \quad (18.9)$$

где α_x, α_y - величины, обратно пропорциональные радиусам корреляции поля вдоль осей OX и OY соответственно. При выполнении условий (18.9) формируемое на основе уравнения (18.4) случайное поле является однородным с нулевым математическим ожиданием и дисперсией σ^2 . Реализующий это уравнение алгоритм является исключительно простым и позволяет генерировать устойчивые реализации, которые могут использоваться в качестве тестовых изображений с контролируемыми статистическими характеристиками.

Исследование уравнения (18.5) показывает, что при произвольных коэффициентах условия устойчивости для него не обеспечиваются, что приводит при его использовании к формированию аномальных реализаций ГОП. Следует отметить, что в общем случае анализ устойчивости для систем, описываемых уравнениями пространственной авторегрессии, представляет определенные трудности. В некоторых работах для решения этой задачи предлагается проводить анализ однородности поля на основе получения выражений для условных и безусловных моментов плотностей распределения значений поля в произвольных точках координатной сетки.

Некаузальная модель при условии, что получены точки области D_p , окружающей точку с координатами (n, m) , описывается в простейшем случае уравнениями следующего вида:

$$\begin{aligned} \xi[n, m] = & b_{-1-1}\xi[n-1, m-1] + b_{0-1}\xi[n, m-1] + b_{1-1}\xi[n+1, m-1] + \\ & + b_{-10}\xi[n-1, m] + b_{10}\xi[n+1, m] + b_{-11}\xi[n-1, m+1] + \\ & + b_{01}\xi[n, m+1] + b_{11}\xi[n+1, m+1] + a_0x[n, m] \end{aligned} \quad (18.10)$$

Здесь в область D_p попадет восемь точек, а в область определения нового значения, как и в предыдущем случае, - одна точка. Применение некаузальных алгоритмов генерации случайного поля имеет специфику, связанную с заданием для них граничных значений. Эти значения должны быть заранее predeterminedены на всей области значения случайного поля, что вызывает определенные трудности.

Рассмотрим теперь возможности получения алгоритма генерации ГОП на основе каузальной АР-модели для общего случая конфигурации D_p , представленного на рисунке 21.2, б. Такую модель будем называть S -точечной, имея в виду, что количество элементов, определяемое в D_p , равно теперь $S = N \times M + N + M$. При использовании каузальной модели авторегрессии соотношение (21.11) можно переписать следующим образом:

$$\xi[n, m] = \sum_{r=0}^M \sum_{k=0}^N b_{kr} \xi[n - k, m - r] + a_0 x[n, m] \quad (18.11)$$

Рассмотрим алгоритм получения коэффициентов АР-модели исходя из заданного вида функции пространственной корреляции поля $R[n, m] = R(n\Delta x, m\Delta y)$. Для этой цели могут использоваться нормальные уравнения Юла-Уокера, общий вид которых для пространственной авторегрессии представлен в виде уравнений с блочными матрицами. Решение базируется на реализации быстрого алгоритма Левинсона для блочно-теплицевых матриц. Мы ограничимся более простой формой записи и решения этих уравнений, исходя из следующего. Представим вектор элементов случайного поля, лежащий на рисунке 21.2, б в области, объединяющей D_p и точку определения нового значения $\xi[n = N + 1, m = M + 1]$, в виде составного вектора:

$$\begin{aligned} \xi &= (\xi_{11}, \dots, \xi_{1M+1}, \xi_{21}, \dots, \xi_{2M+1}, \dots, \xi_{N+11}, \dots, \xi_{N+1M+1})^T \\ &= (\zeta^T, \xi_{N+1M+1})^T \\ \xi_{nm} &= \xi[n, m], n = \overline{1, N+1}; m = \overline{1, M+1} \end{aligned} \quad (18.12)$$

Матрица ковариации составного вектора рассчитывается следующим образом:

$$\begin{aligned} C_\xi &= M[\xi \xi^T] = \|c_{ij}\|, c_{ij} = R[n - 1, m - k], \\ i &= (M + 1)(n - 1) + m, n = \overline{1, N+1}; m = \overline{1, M+1} \\ j &= (M + 1)(l - 1) + k, l = \overline{1, N+1}; k = \overline{1, M+1} \end{aligned} \quad (18.13)$$

Найдем теперь матрицу, связывающую элемент $\xi_{N+1M+1} = \xi[N + 1, M + 1]$ с вектором ζ , включающим все элементы поля, лежащие в области D_p .

Повторяя, по существу, ранее приведенные соотношения для корреляционного согласования фрагментов поля, получим, что

$$\xi_{N+1M+1} = B\zeta + v \quad (18.14)$$

где B - матрица-строка линейного преобразования, состоящая из коэффициентов АР, которую надо найти; v - гауссова случайная величина, соответствующая возмущению (18.11) с параметрами $M[v] = 0, M[v^2] = a_0^2$. Умножив левую и правую части уравнения для ξ_{N+1M+1} на ζ^T и выполнив усреднение, получим сначала матричное уравнение для определения матрицы B , а затем и уравнение для расчета дисперсии возмущения a_0^2 :

$$(M[\xi_{N+1M+1}\xi_{11}]), (M[\xi_{N+1M+1}\xi_{12}]), \dots, (M[\xi_{N+1M+1}\xi_{N+1M}]) = c_{\xi\zeta} = BC_{\zeta} \quad (18.15)$$

$$B = c_{\xi\zeta}C_{\zeta}^{-1}, a_0^2 = \sigma^2 - BC_{\zeta}B^T$$

где $c_{\xi\zeta} = (c_{(N+1)(M+1)}, c_{(N+1)(M+1),2}, \dots, c_{(N+1)(M+1),(N+1)(M+1)-1})$ - нижняя строка матрицы ковариации составного вектора $C_{\zeta} = M[\xi\xi^T]$.

Таким образом, получены все коэффициенты АР-модели, позволяющие проводить генерацию реализаций ГОП. Выполнение алгоритма генерации предполагает сначала задание начальных (осевых) граничных значений поля в виде «дорожек», расположенных вдоль осей OX и OY (как это показано на рисунке 21.2, б). Эти граничные значения могут быть получены на основе ранее описанного рекуррентного алгоритма (21.2) путем последовательной генерации блоков размера $M \times N$ и их присоединения к ранее сформированным фрагментам поля с выполнением условий корреляционного согласования. Затем к полученным граничным значениям последовательно добавляются точки, располагающиеся в центральной части координатной сетки [1].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение:

операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

1. Реализовать алгоритм, основанный на выполнении факторизации матрицы ковариации (факторизацией Холецкого) полученного случайного вектора.

Ниже приводится фрагмент *m*-файла, реализующего подготовительный этап моделирования ГОП с функцией корреляции вида $R(x, x', y, y') = \sigma^2 \exp(-\alpha r)$, $r = \sqrt{1(x - x')^2 + (y - y')^2}$.

```
%Моделирование поля путем факторизации матрицы ковариации случайного вектора
%задание размера моделируемого фрагмента поля, параметров корреляционной функции,
N=50; %число точек по оси OX (число столбцов)
M=25; %число точек по оси OY (число строк)
D=1; alf=0.25;
dx=1; dy=1;

% 1. Подготовительные вычисления
%формирование матрицы ковариации элементов поля с учетом развертки в вектор-столбец
PX=zeros(N*M,N*M);
for m=1:M,
    for n=1:N,
        i=M*(n-1)+m;
        for k=1:M,
            for l=1:N,
                j=M*(l-1)+k;
                if j>i,
                    PX(i,j)=D*exp(-alf*sqrt(((n-l)*dx)^2+((m-k)*dy)^2));
                end;
            end;
        end;
    end;
end;
PX=PX+PX'+D*eye(N*M,N*M);
```

```

% 2. Генерация реализации случайного поля
[x p]=randncor(N*M,1,PX);
%восстановление матрицы реализации случайного поля размера M*N
X=reshape(x,M,N);
%отображение результата в виде полутонового изображения (формат double от 0 до 1)
figure(1);
imshow(mat2gray(X));

```

На рис. 21.3, а представлен результат выполнения данной программы в виде реализации случайного поля в пределах указанной области. Рассмотренный подход обладает одним недостатком: ограниченный объем моделируемого фрагмента случайного поля (реально достигаемые размеры не более 100x100). Естественным выходом из этой ситуации является реализация последовательного моделирования отдельных фрагментов и их присоединение к ранее полученным реализациям с выполнением соответствующего корреляционного согласования. Подобное согласование выполняется следующим образом.

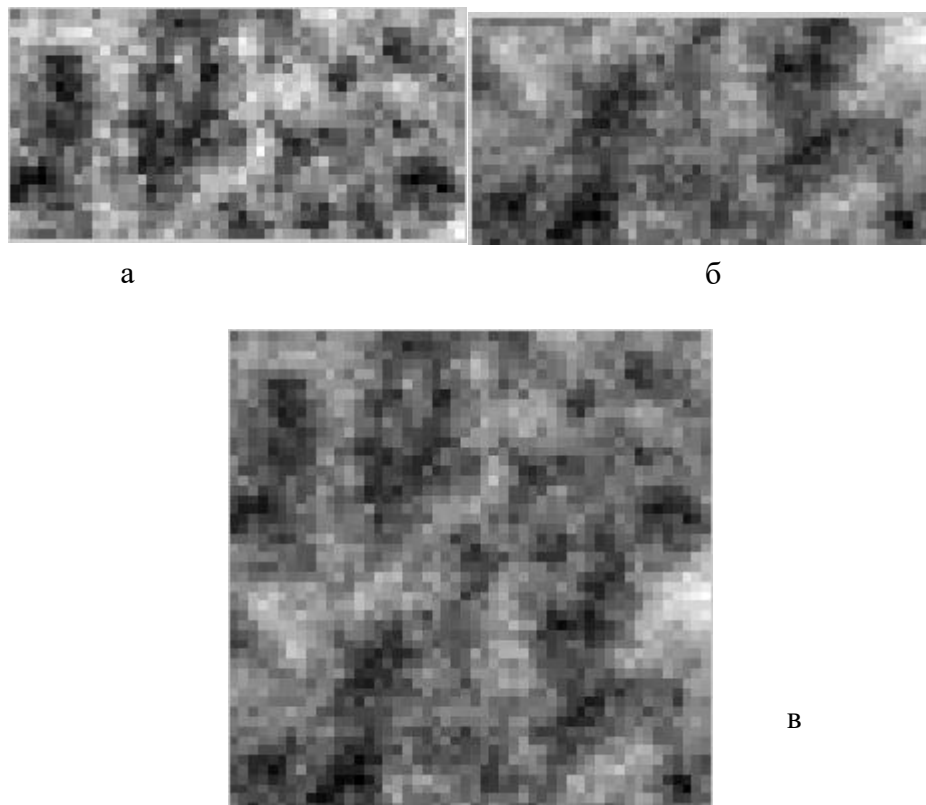


Рисунок 21.3. Пример генерации реализации случайного поля на основе выполнения процедуры корреляционного согласования

Рассмотрим гауссовый случайный вектор

$$\eta = (\xi_{1M+1}, \dots, \xi_{1M+L}, \xi_{2M}, \dots, \xi_{2M+L}, \dots, \xi_{NM+1}, \dots, \xi_{NM+L})^T,$$

$$\xi_{nm} = \xi[n, m], n = \overline{1, N}, m = \overline{M, M+L},$$

описывающий присоединяемый вдоль оси OY фрагмент случайного поля. Исходя из свойств гауссовых случайных векторов и их условных плотностей распределения вероятностей (плотностей вида $P(\eta/\xi)$, где ξ , вектор, описывающий ранее полученный фрагмент поля) будем искать η в виде

$$\eta = H\xi + \nu,$$

где H – матрица линейного преобразования, которую надо найти; ν – гауссовый случайный вектор с параметрами $M[\nu] = 0$, $M[\nu\nu^T] = P_\nu = P_\nu$. Матрица ковариации P_ν условной плотности $P(\eta/\xi)$ также неизвестна. Умножив левую и правую части уравнения для η на ξ^T и выполнив усреднение, получим сначала матричное уравнение для определения матрицы H , а затем и уравнение для определения P_ν :

$$P_{\eta\xi} = M[\eta\xi^T] = HC_\xi, \quad H = P_{\eta\xi}C_\xi^{-1} = (C_\xi^{-1}P_{\xi\eta})^T, \quad P_{\xi\eta} = P_{\eta\xi}^T, \quad (18.16)$$

$$P_\nu = M[(\eta - H\xi)(\eta - H\xi)^T] = C_\eta - P_{\eta\xi}C_\xi^{-1}P_{\xi\eta} = c - HCH^T.$$

Матрицы $C_\eta, P_{\xi\eta}$ в (7.1) определяются следующим образом:

$$C_\eta = |||c_{ij}|||, \quad c_{ij} = R[n-l, m-k],$$

$$i = L(n-1) + m, \quad n = \overline{1, N},$$

$$m = \overline{1, L}; j = L(l-1) + k, l = \overline{1, N}, k = \overline{1, L}. \quad (18.17)$$

$$P_{\xi\eta} = |||c_{ij}|||, c_{ij} = R[n-l, m-k],$$

$$i = M(n-1) + m, \quad n = \overline{1, N}, \quad m = \overline{1, M};$$

$$j = L(l-1) + k - M, \quad l = \overline{1, N}, k = \overline{M+1, M+L}.$$

Соответствующее продолжение m -файла, реализующее эти вычисления при $M = L$, представлено ниже.

```

% 3. Увеличение размеров генерируемой реализации путем корреляционного согласования
% формирование матрицы взаимной ковариации элементов поля в соседних
% (верхнем и нижнем) кадрах с учетом развертки в вектор-столбец
PYX=zeros(N*M,N*M);

for m=1:M,
    for n=1:N,
        i=M*(n-1)+m;
        for k=M+1:2*M,
            for l=1:N,
                j=M*(l-1)+k-M;
                PYX(i,j)=D*exp(-alf*sqrt(((n-l)*dx)^2+((m-k)*dy)^2));
            end;
        end;
    end;
end;

% расчет матрицы линейного преобразования для условной плотности P(Y/X)
H=(inv(PX)*PYX)';
% расчет матрицы ковариации PV условной плотности P(Y/X)
PY=PX;
PV=PY-H*PX*H';

% 4. Генерация присоединяемой реализации
[v p]=randncor(N*M,1,PV);
y=H*x+v;
% восстановление матрицы присоединяемой реализации и отображение результата
Y=reshape(y,M,N);
figure(2);
imshow(mat2gray(Y));
% присоединение матриц X и Y в направлении оси OY и отображение результата
Z=cat(1,X,Y);
figure(3);
imshow(mat2gray(Z));

```

На рис. 18.3, б и в приведены получаемые результаты, из которых видно, что данная процедура обеспечивает формирование реализации увеличенного в два раза размера без наличия видимых разрывов на границе присоединения реализаций. Очевидно, что последовательное выполнение данной процедуры позволяет получить реализации случайного поля любого размера и произвольной конфигурации, при этом заполнение фрагментов поля может осуществляться в различных направлениях (см. рисунок 18.1).

Отметим, что получаемый по этой методике алгоритм может быть в общем случае (с учетом необходимости и возможности получения нестационарных реализаций) описан на основе векторного разностного уравнения вида

$$\xi_k = H_k \xi_{k-1} + v_k, \quad k = 1, 2, \dots \quad (18.18)$$

Для исследования качества алгоритма и определения рекомендуемых параметров случайных полей при его применении, как в прочем и в любом другом случае, должно проводиться статистическое моделирование большого количества реализаций с соответствующей обработкой. Следует также контролировать порядок производимой факторизации и устойчивость алгоритма, реализующего присоединение реализаций в рекуррентном режиме.

2. Реализовать алгоритм авторегрессии

Здесь сначала выполняются следующие подготовительные вычисления.

```

%Моделирование реализации случайного поля размера (K*N)*(K*M)
%с использованием алгоритма авторегрессии.
%Задание размеров поля, граничных блоков, дисперсии случайного поля,
%параметров функции корреляции, интервалов дискретизации
K=10; %количество граничных блоков, располагающихся вдоль осей
N=10;%размеры граничного блока по оси OX (число столбцов)
M=N;%размеры граничного блока по оси OY (число строк)
kr=1; %количество генерируемых реализаций случайного поля
D=1; alf=0.25;
dx=1; dy=1;
KM2=round(K*M/2); %координаты срединной точки поля
KN2=round(K*N/2);

%1. Подготовительные вычисления
%1.1. Формирование матрицы ковариации поля
%с учетом развертки в вектор-столбец
PX=zeros(N*M);
for i=1:N*M,
    for j=1:N*M,
        [n m]=coor(M,i);
        [I k]=coor(M,j);
        PX(i,j)=D*exp(-alf*((n-I)*dx)^2+((m-k)*dy)^2);
    end;
end;

%1.2. Формирование матрицы взаимной ковариации элементов поля в соседних
%(верхнем и нижнем) кадрах с учетом развертки в вектор-столбец
PYX=zeros(N*M);
for i=1:N*M,
    for j=1:N*M,
        [n m]=coor(M,i);
        [I k]=coor(M,j);
        PYX(i,j)=D*exp(-alf*((n-I)*dx)^2+((m-k-M)*dy)^2);
    end;
end;

%расчет матрицы линейного преобразования для условной плотности P(Y/X)
HY=(inv(PX)*PYX)';
%расчет матрицы ковариации для условной плотности P(Y/X)
PVY=PX-HY*PX*HY';

%1.3. Формирование матрицы взаимной ковариации элементов поля в соседних
%(левом и правом) кадрах с учетом развертки в вектор-столбец
PXY=zeros(N*M);
for i=1:N*M,
    for j=1:N*M,
        [n m]=coor(M,i);
        [I k]=coor(M,j);
        PXY(i,j)=D*exp(-alf*((n-I-N)*dx)^2+((m-k)*dy)^2);
    end;
end;

%расчет матрицы линейного преобразования для условной плотности P(X/Y)
HX=(inv(PX)*PXY)';
%расчет матрицы ковариации для условной плотности P(X/Y)
PVX=PX-HX*PX*HX';

```

```

%1.4.Расчет коэффициентов авторегрессии
N1=N+1;
M1=M+1;
PN=zeros(N1*M1);
for m=1:M1,
    for n=1:N1,
        i=M1*(n-1)+m;
        for k=1:M1,
            for I=1:N1,
                j=M1*(I-1)+k;
                if j>i,
                    PN(i,j)=D*exp(-alf*((n-I)*dx)^2+((m-k)*dy)^2);
                end;
            end;
        end;
    end;
end;
PN=PN+PN'+D*eye(N1*M1);
PRX=PN(1:N1*M1-1,1:N1*M1-1);
pn=PN(N1*M1,1:N1*M1-1);
HR=pn*(inv(PRX));%расчет матрицы коэффициентов авторегрессии
PVR=D-HR*PRX*HR';%расчет дисперсии возмущения

```

При проведении вычислений использовалась функция `соог`, позволяющая восстанавливать координаты элемента поля на плоскости по его номеру после развертки блока в столбец:

```

function [x y]=соог(m,nn)
%функция преобразования координаты элемента случайного поля после развертки
%в вектор-столбец в координаты x, y при матричном представлении
%nn-номер элемента при вертикальной развертке
%m-длина столбца исходной матрицы*п, развернутой в столбец длиной m*n
tx=rem(nn,m); % вычисление остатка от деления nn на m
if tx==0, t=nn/m; e=m;
else t=floor(nn/m)+1; e=tx;
end;
x=t; y=e; %при матричном представлении: t- номер столбца; e - номер строки

```

При продолжении *m*-сценария осуществляется процесс формирования реализаций поля на основе авторегрессионной модели, который выглядит следующим образом. Сначала проводится формирование граничных

```

%2. Генерация реализации случайного поля
AB=zeros(1,2*K*N-1);
s=zeros(M*N,M*N);
R=zeros(K*M,K*N);

```

```

for kkk=1:kr,
    %2.1.Задание начального фрагмента в виде реализации гауссового вектора
    [x0 p]=randncor(N*M,1,PX);
    X0=reshape(x0,M,N);

%2.2.Задание граничных значений вдоль оси OY
x=x0;
X=X0;
for kk=1:K-1,
    [v p]=randncor(N*M,1,PVY);
    y=HY*x+v;%получение присоединяемой реализации
    Y=reshape(y,M,N);
    ZY=cat(1,X,Y);%присоединение реализаций X и Y в направлении оси OY
    x=y;
    X=ZY;
end;

%2.3.Задание граничных значений вдоль оси OX
x=x0;
X=X0;
for kk=1:K-1,
    [v p]=randncor(N*M,1,PVX);%получ. присоединяемой реализ. y=HX*x+v
    Y=reshape(y,M,N);
    ZX=cat(2,X,Y);%присоединение реализаций X и Y в направлении оси OX
    x=y;
    X=ZX;
end;

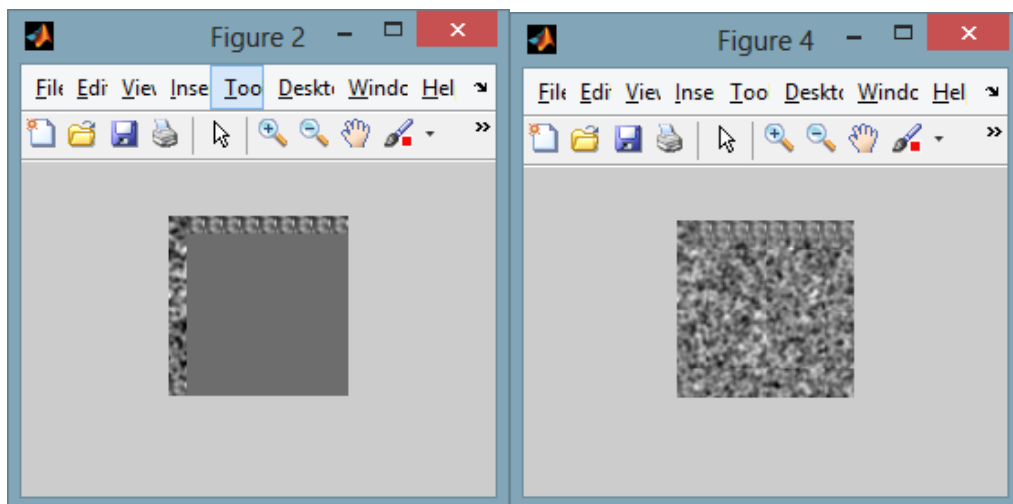
%формирование граничных значений генерируемой реализации поля
R(1:K*M,1:N)=ZY(1:K*M,1:N);
R(1:M,N:K*N)=ZX(1:M,N:K*N);
figure(1);
mesh(R);
figure(2);
imshow(mat2gray(R));

%2.4.Генерация реализации случайного поля
for j=M+1:K*M
    for i=N+1:K*N,
        vr =randn*sqrt(PVR);
        xrl(1:M1,1:N1)=R(j-M:j,i-N:i);
        xr=xrl(:);
        yr=HR*xr(1:N1*M1-1)+vr;
        R(j,i)=yr;
    end;
end;

%2.5. Статистическая обработка реализаций
AB=AB+xcorr(R(KM2,:), 'unbiased'); %оценка корреляционной ф-ии сечения
f=R(KM2+1:KM2+M,KN2+1:KN2+N);
s=s+f(:)*f(:)'; %оценка матрицы ковариации срединного фрагмента поля
end;

```

Результат однократного (значение переменной kr , определяющей количество формируемых реализаций поля, равно единице) выполнения этой части программы представлен на рисунке 18.4,а.



а

б

Рисунок 18.4.Формирование начальных условий в виде граничных фрагментов и результат формирования реализации поля в целом

В предыдущем m-сценарии была представлена часть программы, реализующая собственно авторегрессионный алгоритм (18.10), а также накопление и статистический анализ в случае, если проводится формирование нескольких реализаций в цикле по переменной $kkk= 1:kr$ при $kr > 1$:

```
%2.4.Генерация реализации случайного поля
for j=M+1:K*M
    for i=N+1:K*N,
        vr =randn*sqrt(PVR);
        xr1(1:M1,1:N1)=R(j-M:j,i-N:i);
        xr=xr1(:);
        yr=HR*xr(1:N1*M1-1)+vr;
        R(j,i)=yr;
    end;
end;
%2.5. Статистическая обработка реализаций
AB=AB+xcorr(R(KM2,:), 'unbiased'); %оценка коррелиционной ф-ии сечения
f=R(KM2+1:KM2+M,KN2+1:KN2+N);
s=s+f(:)*f(:)'; %оценка матрицы ковариации срединного фрагмента поля
```

Результат однократного (значение переменной kr , определяющей количество формируемых реализаций поля, равно единице) выполнения этой части программы представлен на рисунке 18.4, б.

Наконец, в заключительной части m -файла осуществляется отображение результатов статистического анализа получаемых реализаций в виде оценки корреляционной функции сечения поля и оценки строки матрицы ковариации срединного фрагмента поля, развернутого в вектор, и их сравнение с соответствующими теоретически рассчитанными характеристиками поля.

```
%отображение оценки корреляционной функции сечения поля оси OX
for j=-N+1:N-1,
    RX(N-j)=D*exp(-alf*(j*dx)^2);
end;
figure(5);
xx=-N+1:N-1;
plot(xx,AB(2*KN2-N+1:2*KN2+N-1),'--rs',xx,RX(:),'b');
xlabel('k= -N+1.....N-1'); ylabel('AB');
%отображение выборочной матрицы ковариации
figure(6);
X=0:M*N-1;
plot(X,s(round(N*M/2),:),'--rs',X,PX(round(N*M/2),:),'b');
xlabel('k= 1,2,...,M*N'); ylabel('s(round(N*M/2),:)');
```

Эти результаты представлены на рисунке 18.5, а и б.

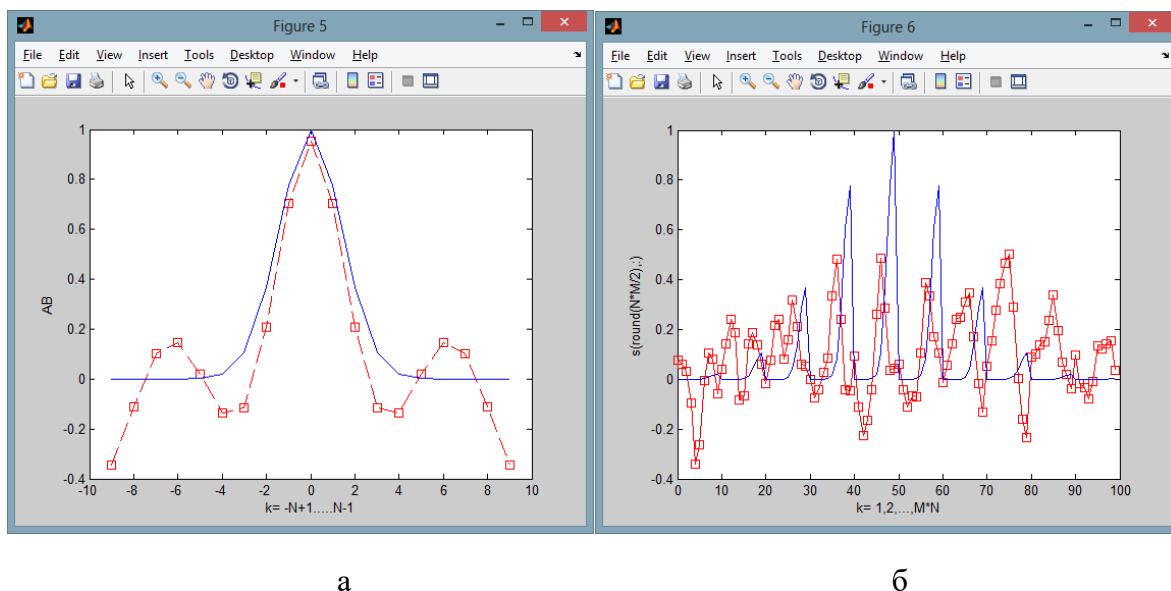


Рисунок 18.5. Экспериментальные оценки функции корреляции сечения и матрицы ковариации фрагмента случайного поля в сравнении с их теоретическими зависимостями

ЛАБОРАТОРНАЯ РАБОТА 19

Формирование цифровых латентных изображений. (2 ч)

Цель и содержание: изучить методику формирования цифровых латентных изображений, приобрести навыки создания цифровых латентных изображений в системе компьютерной математики MATLAB.

Теоретическое обоснование

Латентные изображения это изображения, которые обладают свойством изменения видимости элементов изображения при изменении условий наблюдения или способа регистрации основного изображения.

В настоящее время латентные изображения широко применяются в различных сферах для защиты документов от копирования. Вместе с тем не существует универсального способа выявления латентных изображений, в связи, с чем затруднен и, как следствие, является дорогостоящим процесс отличия копии от подделки. Поэтому разработка и автоматизация методов контроля скрытой информации в латентных изображениях является актуальной задачей, решение которой позволит сделать более дешевыми, точными и качественными процессы контроля подлинности документов, использующих в качестве защиты латентные изображения[6].

К современным методам формирования цифровых латентных изображений относятся методики Шевелева А.А.[18] и Топчиева И.Н.[16]

Способ формирования латентного изображения, предложенный Шевелевым А.А., заключается в том, что основное изображение дважды растривается с использованием двух растровых структур. Из растриванных основных изображений удаляются части изображения с помощью позитивной и негативной масок, которые получают из внедряемого изображения. Затем фрагменты полученных изображений объединяют в единое целое. Для того чтобы качество латентного изображения, полученного этим способом, не отличалось от основного изображения

необходимо, чтобы это изображение имело высокое разрешение, не менее 8000×6000 dpi.

Методика, предложенная Топчиевым И.Н., в отличие от способа Шевелева предполагает формировать латентное изображение путем растривания не основного изображения, а внедряемого:

$$\begin{cases} L(x, y) = k_1 \cdot I(x, y) + k_2 \cdot S_{adj}(x, y); \\ S_{adj}(x, y) = a \cdot S(x, y) + b; \\ S(x, y) = k_1 \cdot G_{inv}(x, y) + k_2 \cdot G_r(x, y); \\ G_{inv}(x, y) = (2^L - 1) - G(x, y); \\ G_r(x, y) = k_1 \cdot G(x, y) + k_2 \cdot R_i(x, y), \end{cases} \quad (19.1)$$

где $L(x, y)$, $I(x, y)$, $G(x, y)$ – числовые массивы, соответствующие латентному, основному и внедряемому изображениям соответственно;

$G_r(x, y)$, $G_{inv}(x, y)$ – числовые массивы растриванного внедряемого и инвертированного внедряемого изображений;

$S(x, y)$ – числовой массив, полученный в результате объединения инвертированного и растриванного изображений;

$S_{adj}(x, y)$ – числовой массив, полученный после яркостных преобразований изображения $S(x, y)$;

k_1 , k_2 – неотрицательные константы; L – число разрядов квантования, $L = 8$; a и b – определяются желаемыми значениями яркости $S_{adj}(x, y)$:

$$a = (S_{adj}(x, y)_{\max} - S_{adj}(x, y)_{\min}) / (S(x, y)_{\max} - S(x, y)_{\min});$$

$$b = (S_{adj}(x, y)_{\min} \cdot S(x, y)_{\max} - S_{adj}(x, y)_{\max} \cdot S(x, y)_{\min}) / (S(x, y)_{\max} - S(x, y)_{\min})$$

$$\text{и} \quad \text{при} \quad S_{adj}(x, y)_{\min} = 0, \quad S_{adj}(x, y)_{\max} = 255,$$

$$a = 2^L - 1 / S(x, y)_{\max} - S(x, y)_{\min} \quad \text{и} \quad b = (1 - 2^L) \cdot S(x, y)_{\min} / S(x, y)_{\max} - S(x, y)_{\min}$$

.

Основное, скрываемое и изображение растровых структур – это числовые массивы одного класса и одинаковых размеров. Предложенная

модель позволяет формировать латентные изображения с помощью стандартного программного обеспечения. Формирование латентных изображений с помощью модели (19.1) реализовано в системе компьютерной математики Matlab и графическом редакторе AdobePhotoshop. Методика формирования латентного изображения заключается в выполнении последовательности действий в соответствии с приведенной схемой (рисунок 3.3):



Рисунок 19.1 – Схема процесса формирования латентного изображения

- 1) Инвертирование скрываемого изображения.
- 2) Растривание дублированного скрываемого изображения растровыми структурами.
- 3) Объединение инвертированного скрываемого и растриванного изображений.
- 4) Яркостные преобразования изображения $S_{adj}(x, y)$.
- 5) Получение латентного изображения путем объединения основного изображения и изображения с измененным контрастом.

Рассмотрим способ реализации математической модели формирования латентного изображения с помощью системы компьютерной математики Matlab.

Для формирования латентного изображения необходимо выполнить следующую последовательность действий:

1. Импортировать в рабочую область Matlab необходимые для формирования латентного изображения данные: « I » – основное изображение $I(x, y)$; « G » – скрываемое изображение $G(x, y)$; « R_i » – растровые структуры $R_i(x, y)$, где $i = \overline{1, n}$, с помощью функции: $imread('filename')$, где: « $filename$ » – строка символов имени загружаемого изображения (рисунок 19.2). В рассматриваемом примере скрываемое изображение растрируется только одной растровой структурой $R_1(x, y)$, которая изображена на рисунке 19.2в. Основное и скрываемое изображение – это числовые массивы одного класса и одинакового размера. К общему размеру изображения можно привести с помощью функции $imresize(I, m, method)$, которая создает новое изображение, с размерами в m раз отличающихся от размеров исходного изображения « I ». Параметр « $method$ » предназначен для изменения размеров по одному из predetermined методов интерполяции (билинейная или бикубическая).

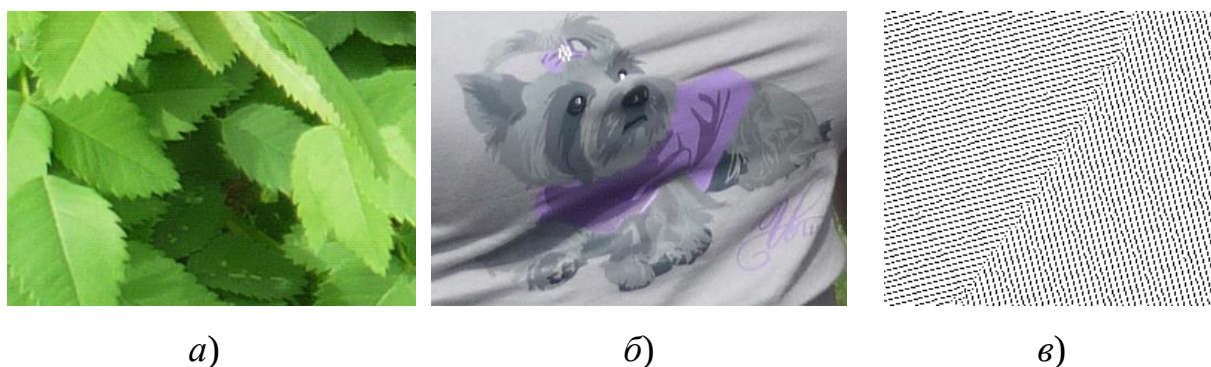


Рисунок 19.2 – Исходные изображения: а) – основное « I », б) – скрываемое « G », в) – растровая структура « R_i »

2. Сформировать на основе скрываемого изображения « G » его дубликат, присвоив новому массиву дублированного изображения « $GSору$ »

значения массива скрываемого изображения $G(x, y)$, с помощью операции присваивания $GCopy = G$.

3. Сформировать на основе массива скрываемого изображения «G» новый числовой массив, соответствующий его инвертированной копии «GInvert», с помощью операции: $GInvert = 255 - G$.

4. Провести растривание дублированного скрываемого изображения «GCopy» растровой структурой $R_1 = R_1(x, y)$ с помощью функции вычисления линейной комбинации «imlincomb», для получения числового массива растриванного дублированного скрываемого изображения «GCopy_R» = $G_r(x, y)$, с помощью операции: $GCopy\_R = imlincomb(0.5, GCopy, 0.5, R_1)$.

5. Объединить числовые матрицы «GInvert» и «GCopy_R», соответствующие инвертированной копии и растриванному изображению в массив «SCR_SN» = $S(x, y)$, с помощью функции вычисления линейной комбинации изображений: $SCR\_SN = imlincomb(0.5, GCopy\_R, 0.5, GInvert)$.

6. Изменить параметры яркости изображения «SCR_SN_Lightness» = $S_{adj}(x, y)$, с помощью функции преобразования яркости изображения $SCR\_SN\_Lightness = imadjust(SCR\_SN, [0.4 \ 0.5], [0.4 \ 0.5])$, которая формирует новое изображение, со значениями яркости в интервале 0,4 – 0,5.

7. Объединить основное изображение «I» и изображение с измененными значениями контраста «SCR_SN_Lightness» в новый массив, соответствующий изображению «Latent» = $L(x, y)$ с помощью функции линейной комбинации изображений: $Latent = imlincomb(0.25, SCR\_SN\_Lightness, 1, I)$.

На рисунке 19.3 представлено полученное изображение $L(x, y)$, содержащее скрытую информацию.



а)



б)

Рисунок 19.3 – Полученные латентные изображения с помощью методики Топчиева (а) и методики Шевелева (б)

Как видно из представленного рисунка 19.3 изображения характеризуются незаметностью внедренной информации.

Для оценки визуального качества латентного изображения использован известный эмпирический подход, согласно которому в качестве параметров изображений использовались: среднеарифметическое значение $\bar{L}$ яркостей, полнота использования градаций яркостей, резкость изображения и его обобщенный контраст. Проведенный анализ изображений показал, что для оценки визуального качества достаточными являются два параметра оценки уровня адаптации зрительной системы и оценка степени использования возможных градаций яркостей, поэтому расчет интегрального критерия визуального качества осуществлялся с помощью выражения:

$$Q = k \cdot LQ \cdot KQ, \quad (19.2)$$

где k – нормирующий коэффициент;

LQ – оценка уровня адаптации зрительной системы;

KQ – аналитическая оценка степени использования возможных градаций яркостей.

$$LQ = \frac{\bar{L} - LMAX / 2}{LMAX / 2}, \quad (19.3)$$

где $LMAX / 2$ – половина максимально возможного диапазона яркостей.

$$KQ = \frac{S}{LMAX}, \quad (19.4)$$

где S – количество уровней яркостей, для каждого из которых на данном изображении присутствуют большее чем $b \times N \times M$ количество элементов с данной яркостью (N и M – размеры изображения, b – некоторая константа).

Полученные результаты расчета интегрального критерия для исходного, а также для сформированных латентных изображений известным способом и предложенной методикой соответственно равен: 0,79; 0,89; 0,77, что позволяет сделать вывод о том, что качество полученного латентного изображения с помощью методики Топчиева лучше, чем качество изображения, полученного с помощью способа Шевелева.

Для того чтобы качество латентного изображения, полученного способом Шевелева, не отличалось от оригинала необходимо использовать основное изображение высокого разрешения: $8000 \times 6000 \text{ dpi}$, в то время как для формирования латентного изображения на основе модели Топчиева достаточно разрешения $400 \times 300 \text{ dpi}$.

Предложенный способ позволяет формировать латентные изображения с высокой степенью защиты от подделки с помощью стандартного, промышленно используемого программного обеспечения. Высокая степень защиты обусловлена тем, что часть кодированной информации латентного изображения теряется при сканировании и обработке (изменение геометрических параметров изображений, например, разрешения, размера)[16].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 19.1. Сформировать латентное изображение из изображений предложенных преподавателем.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Какие современные методы формирования цифровых латентных изображений существуют?
2. В чем заключается методика предложенная Шевелевым А.А.?
3. В чем заключается методика предложенная Топчиевым И.Н.?

ЛАБОРАТОРНАЯ РАБОТА 20

Методы выявления скрытой информации в латентных изображениях

(2 ч)

Цель и содержание: изучить методику выявления скрытой информации в латентных изображениях, приобрести навыки выявления скрытого изображения в системе компьютерной математики MATLAB.

Теоретическое обоснование

Для выявления латентного изображения используются оптические методы неразрушающего контроля, основанные на способности скрытого изображения становится видимым в косопадающем свете, ультрафиолетовых, инфракрасных, лазерных лучах, а также методы, основанные на цифровой фильтрации. Выявление латентного изображения на основе Фурье-преобразования и фильтрации, дает положительные результаты, если в обрабатываемом изображении сохранилась растровая структура, для чего необходимо иметь разрешение изображения не менее 1200 dpi. При таком разрешении изображения операции прямого и обратного преобразования Фурье требуют больших вычислительных ресурсов компьютера.

Одним из перспективных методов выявления скрытого изображения в латентном является вейвлет-разложение.

В основе вейвлет-анализа сигнала лежит идея разложения сигнала на низкочастотную и высокочастотную составляющие, поэтому вейвлет-преобразование латентного изображения можно использовать для выявления скрытой информации.

Средства среды Matlab, представленные пакетом WaveletToolbox, позволяют анализировать изображения с помощью вейвлетов Хаара (haar), Добеши (db), Симлета (sym), Коифлета (coif), биортогональных вейвлетов (bior) и обратно-биортогональных (rbio) и вейвлетов Мейера в дискретном варианте (dmey).

Вейвлет-разложение латентного изображения и его анализ удобней всего производить в графической среде **Wavelet 2-D** (рисунок 20.1). Эта среда предназначена для проведения двумерного дискретного вейвлет-анализа изображений, которые представляют собой двумерный сигнал, задаваемый матрицей. Латентное изображение загружается с помощью меню **File/LoadImage**. После загрузки изображения в поле **Original Image** необходимо выбрать вид вейвлета в поле **Wavelet**(1, рисунок 20.1) и уровень вейвлет-разложения (2).

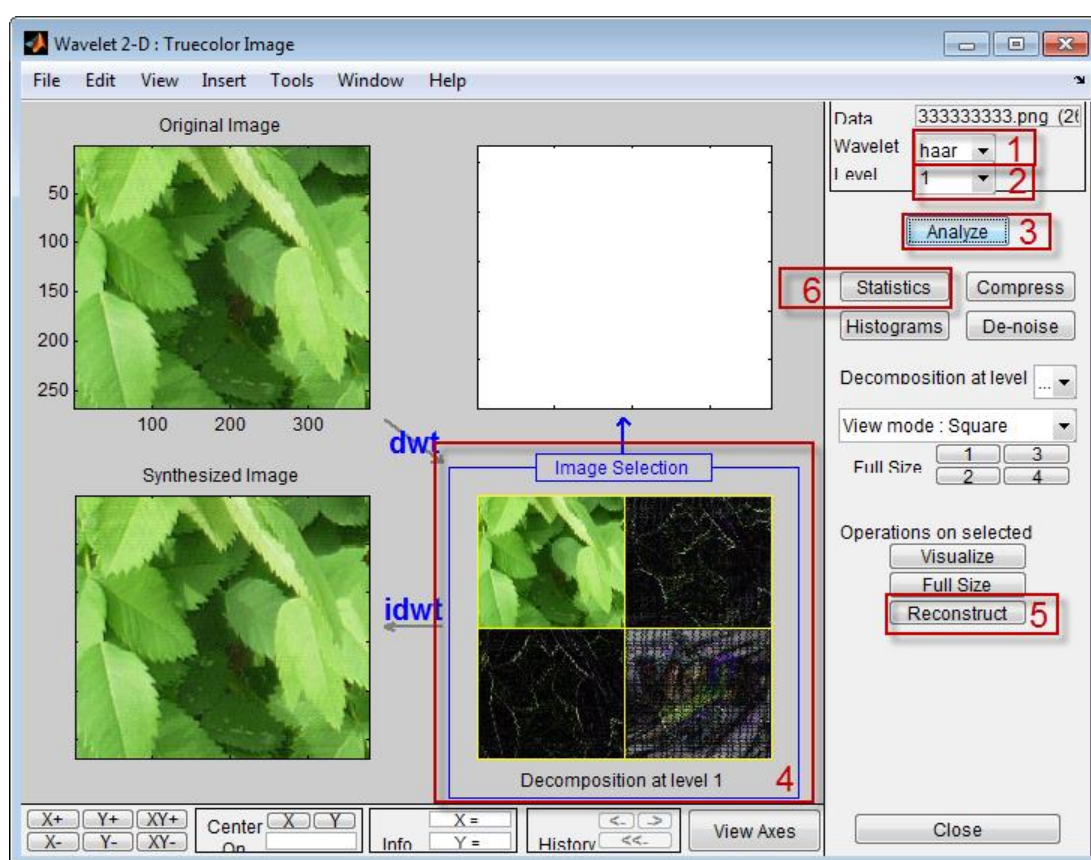


Рисунок 20.1. Графическая среда по анализу изображений

Для выполнения вейвлет-разложения необходимо нажать кнопку **Analyze**(3). Результаты разложения будут выведены в виде блочной матрицы (4) в поле **Decomposition at level 1** графического окна. В верхнем левом блоке представлены аппроксимирующие коэффициенты, в правом верхнем – горизонтальные детализирующие коэффициенты, в левом нижнем –

вертикальные детализирующие коэффициенты, в правом нижнем – диагональные детализирующие коэффициенты. Все полученные коэффициенты образуют матрицы размером в два раза меньше, чем исходная матрица изображения. На рисунке 20.1 видно, что диагональные коэффициенты (правый нижний угол) выявили скрытое изображение.

Восстановление изображения осуществляется по коэффициентам выбранного блока после нажатия кнопки **Reconstruct** (5). Результат выводится в поле **SynthesizedImage**.

Для получения статистических характеристик используется кнопка **Statistic**(6, рисунок 20.1). В окне статистики **Wavelet 2-DStatistics** (рисунок 20.2) содержатся статистические данные о сигнале (исходном, синтезированном) или коэффициентах разложения. По умолчанию выводятся статистические данные об исходном сигнале. Для указания коэффициентов аппроксимации или детализирующих коэффициентов необходимо с помощью переключателя выбрать **Approximation** или **Detail** соответственно.

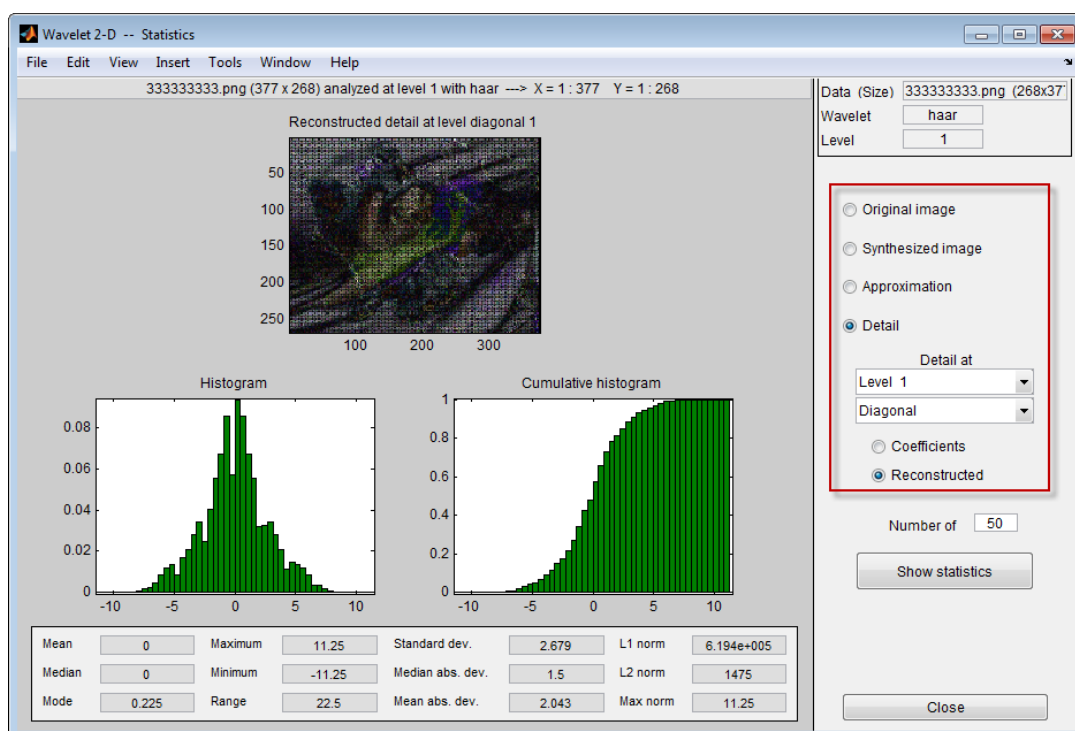


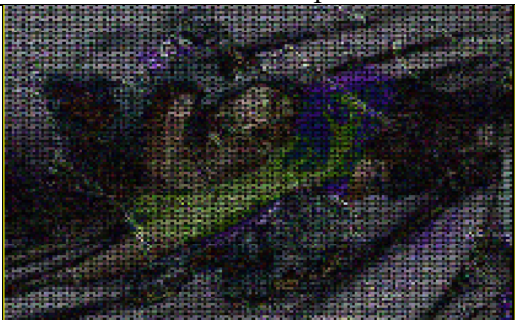
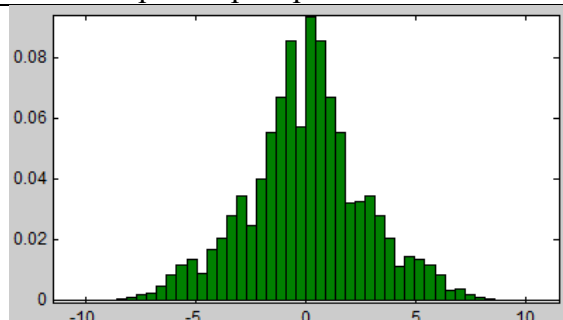
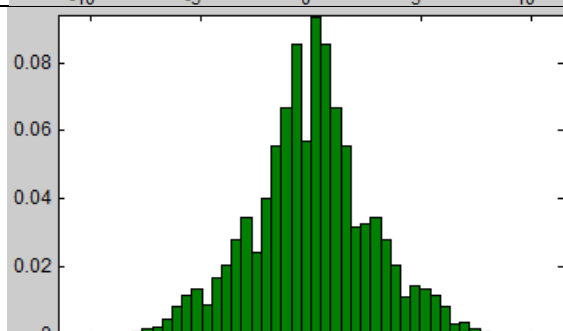
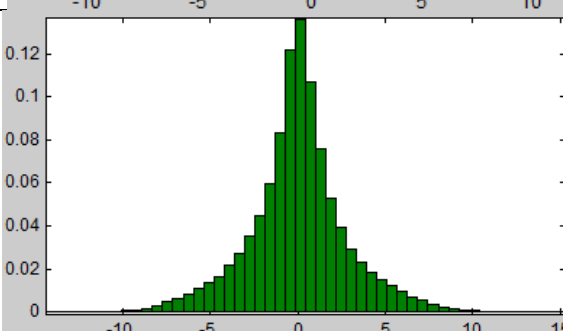
Рисунок 20.2. Окно статистики

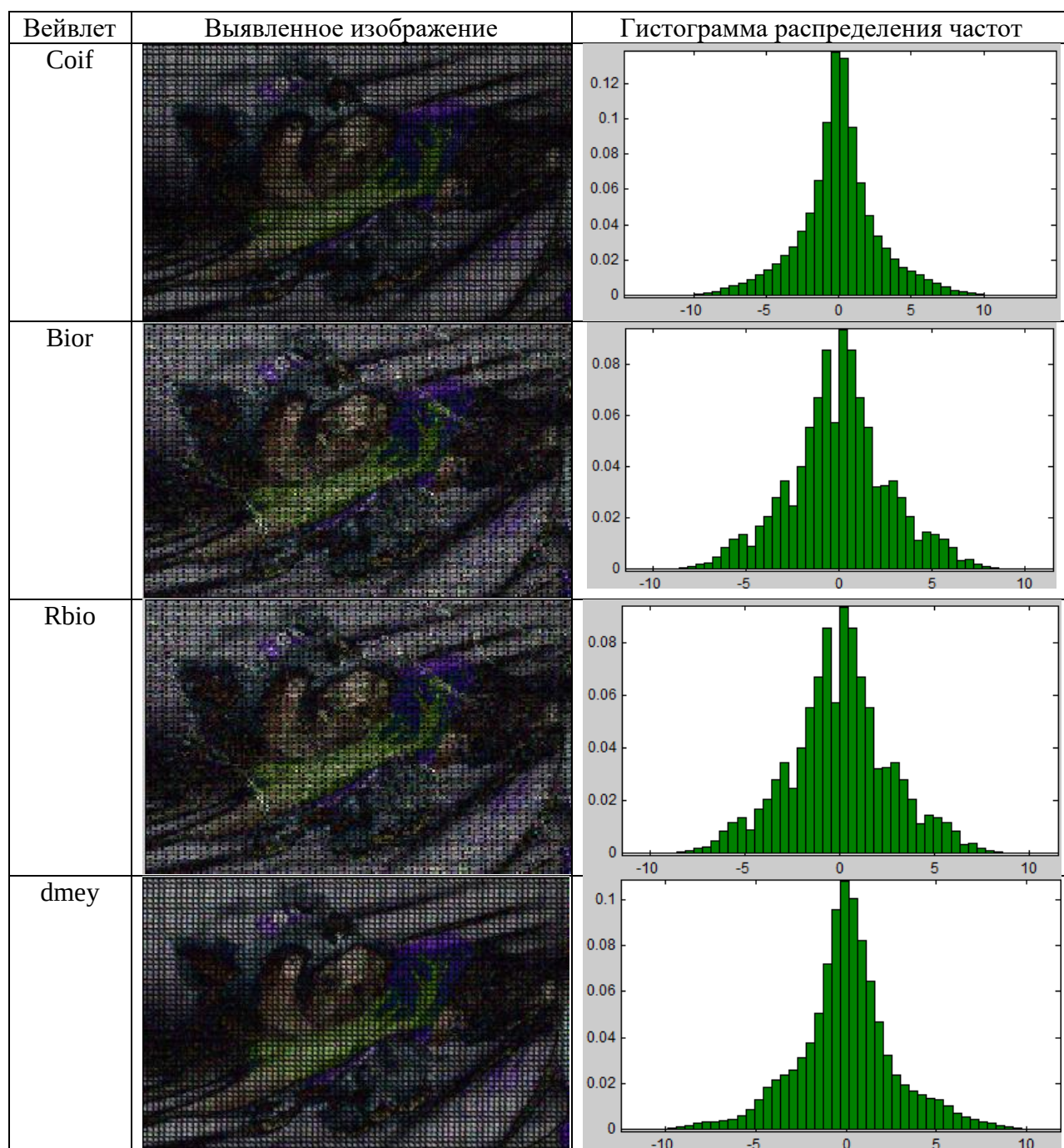
Если выбраны детализирующие коэффициенты, то необходимо указать уровень разложения, например, **Level 1** и для каких коэффициентов (горизонтальных, вертикальных или диагональных) необходимо рассчитать статистические характеристики.

Как видно из рисунка 20.2 система Matlab позволяет получить статистические характеристики для детализирующих коэффициентов.

В таблице 20.1 представлены скрытые изображения и гистограммы относительных частот. Вид гистограммы отражает плотность распределения двумерной случайной величины, характеризующей выявленное изображение. Визуально установлено, что лучший результат был получен с использованием вейвлета Мейера. Неплохие результаты были получены с помощью симлетов и коифлетов.

Таблица 20.1. Результаты анализа

Вейвлет	Выявленное изображение	Гистограмма распределения частот
Haar		
Db		
Sym		



В таблице 20.2 приведены статистические характеристики анализируемых изображений.

Таблица 20.2. Статистические характеристики изображений

	haar	db	sym	coif	bior	rbio	dmey
Медиана	0	0	0.016	0.016	0	0	0.009
Мода	0.225	0.225	0.144	-0.236	0.225	0.225	-0.074
Макс.	11.25	11.25	15.03	14.68	11.25	11.25	11.61
Мин.	-11.25	-11.25	-14.16	-14.57	-11.25	-11.25	-12.24
Диапазон	22.5	22.5	29.19	29.25	22.5	22.5	23.85

Из таблицы 2 видно, что медиана у более качественных изображений (0,009; 0,016; 0,016) отлична от нуля, а такая числовая характеристика как мода меньше (-0,074; -0,236; 0,144), чем у остальных (0,225), в тоже время частотный диапазон таких изображений шире. Таким образом, по виду гистограммы и полученным статистическим характеристикам можно сделать вывод о качестве выявленного изображения [6].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 20.1.С помощью вейвлет-разложения выявите скрытую информацию из изображения, сформированного в лабораторной работе №19, с применением различных вейвлетов.

Задание 20.2.Соберите статистические данные по диагональным коэффициентам полученным при вейвлет-разложении изображения различными вейвлетами.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Какие виды вейвлетов доступны для вейвлет-анализа изображений в системе matlab?
2. Какие статистические характеристики полученных коэффициентов можно получить в системе matlab?
3. На чем основаны методики выявления латентных изображений?

ЛАБОРАТОРНАЯ РАБОТА 21

Разработка систем компьютерного моделирования динамических процессов в жидких дисперсных магнитных наносистемах (2 ч)

Цель и содержание: изучить основные приемы разработки систем компьютерного моделирования динамических процессов в пакете Simulink

Теоретическое обоснование

Для разработки системы компьютерного моделирования динамических процессов в жидких дисперсных магнитных наносистемах использована система дифференциальных уравнений, описывающая процесс восстановления равновесной формы капли магнитной жидкости после выключения внешнего магнитного поля.

$$\left\{ \begin{aligned} \frac{dV_{\lambda}(t)}{dt} &= \frac{2}{3} \cdot \frac{\lambda(t)^2 + 2}{\lambda(t) \cdot (2\lambda(t)^2 + 1)} (V_{\lambda}(t))^2 - \frac{60 \cdot \eta}{\left(\rho + \frac{\rho_1}{2}\right) \cdot R^2} \cdot \frac{\lambda(t)^{2/3}}{(2\lambda(t)^2 + 1)} \cdot V_{\lambda}(t) + \\ &- \frac{45 \cdot \sigma}{4R^3 \left(\rho + \frac{\rho_1}{2}\right)} \cdot \frac{\lambda(t)^3}{(2\lambda(t)^2 + 1) \cdot (\lambda(t)^2 - 1)} \left(\frac{\lambda(t)^2 - 4}{\sqrt{\lambda(t)^2 - 1}} \arcsin\left(\frac{\sqrt{\lambda(t)^2 - 1}}{\lambda(t)}\right) + \frac{2}{\lambda(t)^2} + 1 \right), \\ \frac{d\lambda(t)}{dt} &= V_{\lambda}(t). \end{aligned} \right. \quad (21.1)$$

Состояние динамической системы характеризуется парой величин: координатой $\lambda(t)$ и скоростью $\dot{\lambda}(t) = V_{\lambda}(t)$, являющихся компонентами вектора состояний системы, принадлежащего пространству состояний (иначе, фазовому пространству).

Для решения системы (21.1) необходимо составить блок-схему, позволяющую исследовать поведение изучаемой динамической системы в пакете расширения Simulink системы Matlab. В основе построения блок-схемы лежит последовательный итерационный процесс. Алгоритм итерационного процесса заключается в следующем:

1. Формируется правая часть первого уравнения системы (21.1), которая зависит от относительного удлинения капли $\lambda(t)$ и скорости изменения удлинения $V_\lambda(t)$. Считается, что $\lambda(t)$ и скорость изменения $\lambda(t)$ – $V_\lambda(t) = \dot{\lambda}(t)$ известны, следовательно, известно и ускорение $a_\lambda(t) = \ddot{\lambda}(t)$.

2. Определяется скорость $V_\lambda(t) = \dot{\lambda}(t)$ путем интегрирования ускорения $a_\lambda(t)$.

3. Определяется относительное удлинение капли $\lambda(t)$ путем последующего интегрирования скорости $V_\lambda(t)$.

4. Выполняется п.1 – полученные значения $\lambda(t)$ и $V_\lambda(t)$ используются для формирования правой части первого уравнения системы (21.1)

Блок схема (рисунок 21.1) содержит два последовательно соединенных интегратора с внешне задаваемыми величинами $\rho, \rho_1, \sigma, \eta, R$ в соответствии с системой (21.1) и начальными условиями $\lambda(0) = \lambda_0$ и $V_\lambda(0) = \dot{\lambda}_0$.

На вход первого интегратора подается ускорение $\dot{V}_\lambda(t)$, в качестве начального условия используется начальное значение скорости $V_\lambda(0) = \dot{\lambda}_0$.

На выходе этого интегратора будет текущая скорость изменения относительного удлинения капли $V_\lambda(t)$. Эта величина подается на вход второго интегратора с начальным условием в виде начального удлинения $\lambda(0) = \lambda_0$.

Выходом со второго интегратора будет зависимость относительного удлинения капли от времени.

Отдельным блоком в схеме приведена подсистема – субмодель (блок Sybsystem), представляющая собой функцию правой части первого уравнения системы (21.1) – рисунок 21.2. При построении блок схемы подсистемы ее связь с основной системой осуществляется путем ввода в подсистему стандартных блоков In (Вход) и Out (Выход). Все величины, которые формируются в основной модели системы, а используются в подсистеме (субмодели) поступают в подсистему через блоки In.

Величины, сформированные в подсистеме, а используемые в основной системе, поступают в основную модель из подсистемы через блоки Out. При этом на изображении блока подсистемы (блок Sybsystem) в блок-схеме основной модели появляется количество входов, равное числу введенных в подсистему блоков In (рисунок 21.1).

Входными параметрами подсистемы являются $\lambda(t)$ и $\ddot{\lambda}(t)$, а входными величинами ρ , ρ_1 , σ , η , R .

Для отображения результатов моделирования использован блок Scope, подключенный к сформированному сигналу. Для отображения фазового портрета использован блок XYGraph.

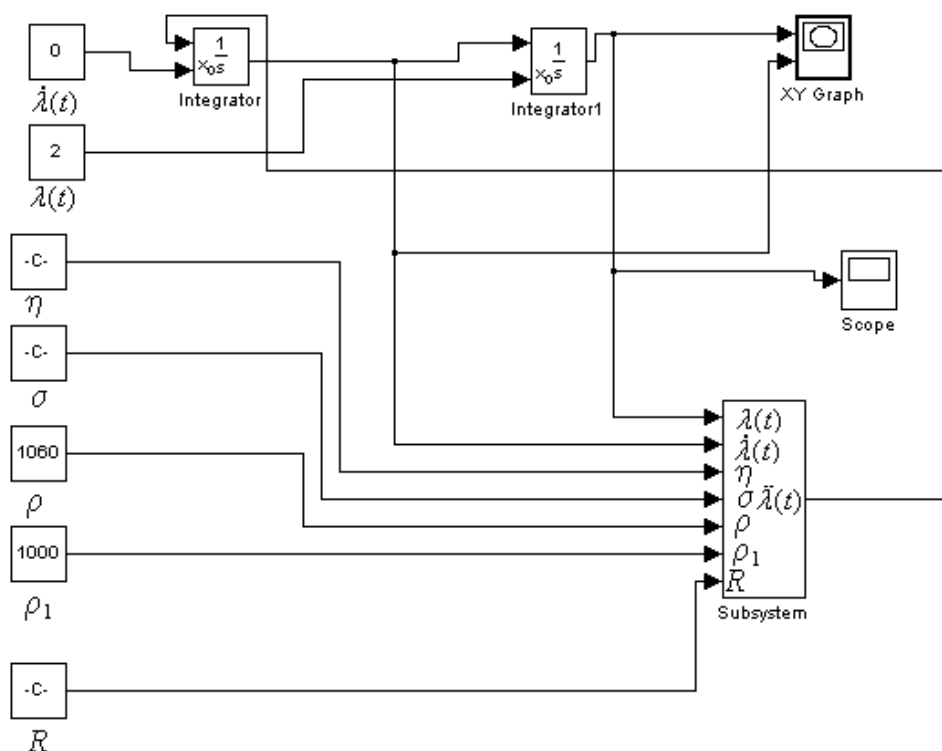


Рисунок 21.1. Блок-схема основной модели динамической системы, описываемой системой уравнений (21.1)

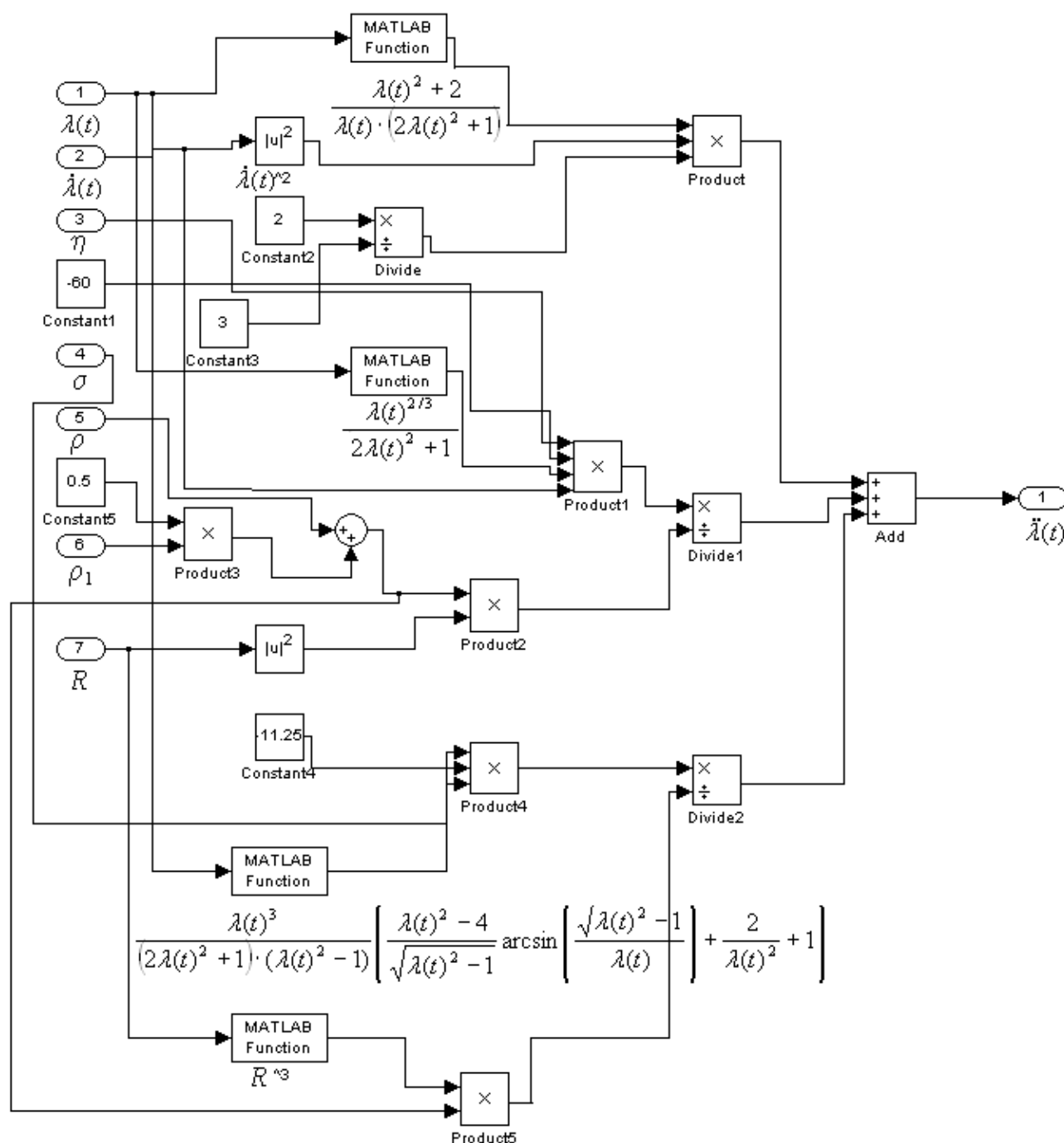


Рисунок 21.2. Блок-схема субмодели динамической системы (21.1)[11]

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение:

операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно изучив материал, представленный в теоретических указаниях.

Задание 21.1. Разработка структурной схемы модели, описывающей динамические процессы – рисунок 21.1

Задание 21.2. Разработка структурной схемы субмодели, описывающей динамические процессы – рисунок 21.2

Задание 21.3. Выполните моделирование, используя следующие параметры: $R = 3$ мм, $\rho = 1060$ кг/м³, $\rho_1 = 1000$ кг/м³, $\sigma = 2 \cdot 10^{-3}$ Н/м, $\eta = 0,0043$ Па·с.

Результаты моделирования представлены на рисунке 21.3, 21.4.

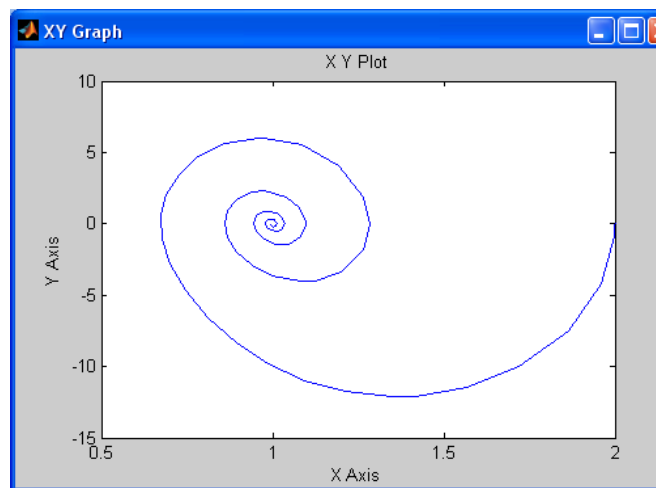
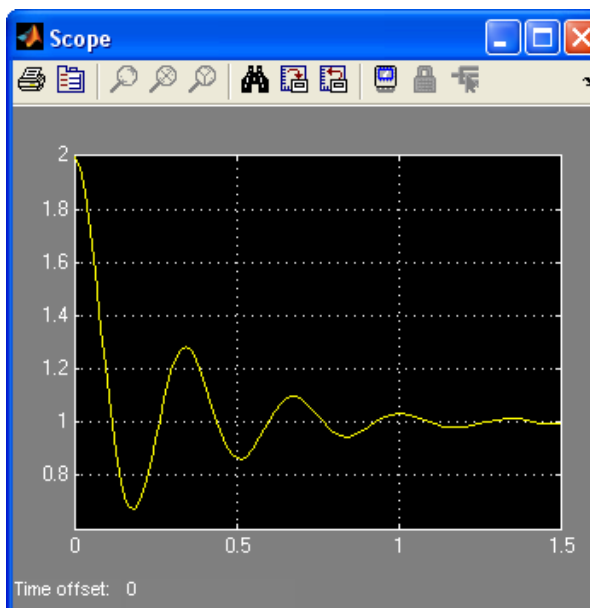


Рисунок 21.3 – Результат решения системы уравнений (21.1) в пакете расширения Simulink системы Matlab

Рисунок 21.4 – Фазовый портрет, полученный в результате решения системы уравнений (21.1) в пакете расширения Simulink системы Matlab

Фазовый портрет, полученный с помощью блока XYGraph пакета Simulink, представлен на рисунке 21.4. Точка с координатами (1,0) на рисунке является асимптотически устойчивым фокусом. Таким образом, эта точка является аттрактором нулевой размерности. Наличие на фазовой траектории фокуса свидетельствует о том, что рассматриваемая динамическая система является диссипативной.

Задание 4. Индивидуальное задание (таблица 21.1)

Вариант	R , мм	ρ , кг/м ³	ρ_1 , кг/м ³	$\sigma \cdot 10^{-3}$, Н/м	η , Па·с
1	3	1060	1000	2	0,0033
2	5	1100	1160	3	0,0043
3	4	1020	1160	5	0,0053
4	7	1080	1030	1	0,0023
5	8	1040	1005	8	0,093
6	2	1050	1040	10	0,0073
7	10	1090	1060	4	0,0013
8	6	1010	1000	2	0,0063
9	9	1070	1020	6	0,0083
10	15	1030	1000	7	0,0039

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Как осуществляется разработка структурных схем?
2. Как с помощью Simulink можно моделировать поведение сложных систем?
3. На какой технологии основана разработка моделей средствами SIMULINK (S-модели)?

ЛАБОРАТОРНАЯ РАБОТА 22

Компьютерное моделирование динамики магнитных капель (2 ч)

Цель и содержание: разработка приложения, описывающего динамику магнитных капель с помощью программного комплекса Simulink Coder системы MATLAB и среды разработки Microsoft Visual Studio 2010.

Теоретическое обоснование

Моделирование динамики магнитных капель с помощью, разработанной в *Simulink* системы компьютерного, невозможно проводить вне среды MATLAB, а так как выполнение программ в MATLAB осуществляется в интерпретирующем режиме, то это ведет к относительно медленной работе системы.

Разработка приложения с использованием любого языка высокого уровня, обеспечивающего компиляцию, для работы вне системы MATLAB обеспечивает более высокую скорость обработки информации за счет повышения скорости вычислений. При компиляции этапы интерпретации и компоновки при исполнении файлов отсутствуют, так как они проводятся в процессе создания исполняемых файлов, в результате скорость вычислений откомпилированных файлов существенно возрастает.

Рассмотрим технологию разработки приложения, описывающего динамику намагничивающихся микрокапель с помощью программного комплекса Simulink Coder (ранее пакет расширения Real-Time Workshop – мастерская реального времени), полностью интегрируемого с MATLAB и Simulink и среды разработки Microsoft Visual Studio 2010.

Программный комплекс Simulink Coder позволяет создавать код любых моделей Simulink, в том числе линейных, нелинейных, непрерывных, дискретных и гибридных. Созданный исходный код может быть использован в приложениях реального времени или в приложениях, которые не требуют выполнения в реальном времени. Этот код также может использоваться для

ускорения моделирования, быстрого прототипирования и аппаратно-программного тестирования. Применим, если необходимо настраивать параметры и наблюдать за сигналами в созданном коде, используя Simulink или взаимодействовать с кодом вне среды MATLAB и Simulink.

BSimulink Coder создан оптимизированный и полностью прокомментированный код на языке высокого уровня C из модели Simulink, представленной системой уравнений (22.1).

$$\left\{ \begin{array}{l} \frac{d\lambda(t)}{dt} = \frac{3\sigma\lambda(t)^{7/3}}{16R\eta(\lambda(t)^2 - 1)} \left(\frac{\lambda(t)^2 - 4}{\sqrt{\lambda(t)^2 - 1}} \arcsin\left(\frac{\sqrt{\lambda(t)^2 - 1}}{\lambda(t)}\right) + 2\lambda(t)^{-2} + 1 \right); \\ \eta(T) = 4,841 \cdot e^{-0,113(T-273)} + 0,063; \\ \sigma(T) = \left(0,563 \cdot e^{-0,028(T-273)} + 0,117 \right) \cdot 10^{-6}. \end{array} \right. \quad (22.1)$$

Температурный диапазон исследования динамики микрокапель определяют аппроксимационные зависимости поверхностного натяжения $\sigma(T)$ и вязкости $\eta(T)$, которые получены при изменении температуры от 293К до 333К. Входными данным для системы являются радиус R капли, температура T и начальное условие – $\lambda(0)$. Выходные данные – зависимость отношения полуосей эллипсоида вращения $\lambda(t)$ от времени.

Построение кода начинается с создания файла model.rtw, который является промежуточным представлением блок-схемы Simulink – модели, которое содержит информацию о значениях параметров элементов системы. Эта информация хранится в независимом от языка формате. После создания model.rtw процедура сборки вызывает компонент «Target Compiler Language», чтобы превратить модель в так называемый Target – зависимый, адаптированный код. «Target Compiler Language» начинает с чтения model.rtw. Затем он компилирует и выполняет команды в target-файлах. Файлы, создаваемые «Target Compiler Language» (TCL-файлы) указывают, как преобразовать model.rtw в предметно-ориентированный код. Затем он загружает отдельные блоки файлов для преобразования информации блоков

модели. На выходе компилятора языка Target является версия исходного кода блок-схемы Simulink.

Следующим этапом процедуры сборки заключается в создании системы Makefile (system.mk) на основе шаблона Makefile (system.tmf) этот шаблон Makefile предназначен для target - системы, что позволяет указать компилятор, опции компилятора а также дополнительную информацию для назначения (цели) созданного исполняемого файла. Model.mk файл создается путем копирования содержимого system.tmf и расширением множества значений, описывающих конфигурацию модели. Вы можете полностью настроить процесс сборки, изменив существующий шаблон Makefile или создать собственный Makefile-шаблон. После model.mk дополнительно загружается исполняемый файл целевого оборудования.

В конце моделирования, исполняемый файл производит model.mat - файл, который содержит переменные, которые были зарегистрированы в Simulink - модели. Этот МАТ-файл используется для анализа в MATLAB.

Основные файлы, создаваемые в процедуре сборки:

- Mikrokaplya.mdl - созданных Simulink, аналогично программирования высокого уровня исходного файла языка.
- Mikrokaplya.rtw - созданные Real-Time Workshop построить процесс, аналогичный объект файла, созданного с языка высокого уровня.
- Mikrokaplya.c - созданные инструментом «Target Compiler Language», это Исходный код на C соответствующей model.mdf
- Mikrokaplya.h - созданные «Target Compiler Language», представляет собой файл заголовка, который отображает взаимосвязи между блоками модели.
- Mikrokaplya _export.h - созданные «Target Compiler Language», это заголовок файла, который содержит экспортируемый сигнал, параметры и функции символов.

– Mikrokaplya.prm - созданные «Target Compiler Language», содержит настройки параметров блоков модели.

– Mikrokaplya.reg - созданные «Target Compiler Language», содержит функцию модель регистрации ответственность за модель инициализации.

В результате генерации кода на языке C были сформированы файлы, часть которых представлена на рисунке 22.1.

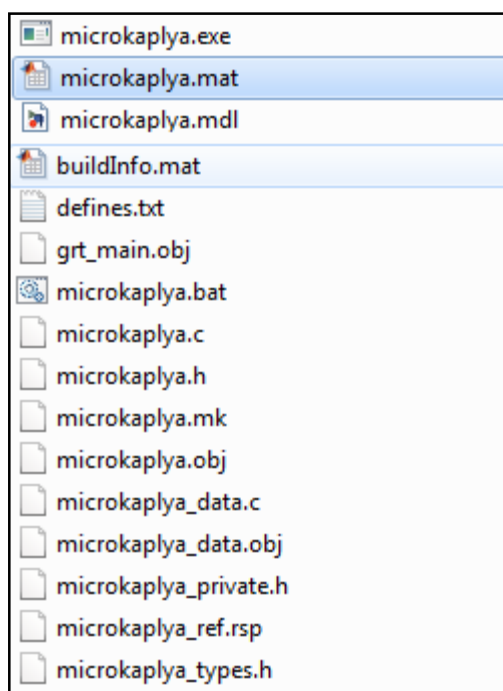


Рисунок 22.1. Файлы исходного кода на языке C, сгенерированные с помощью SimulinkCoder

С помощью среды разработки Microsoft Visual Studio 2010 был разработан интерфейс приложения для моделирования динамики намагничивающихся микрокапель (рисунок 22.2). Входные данные: радиус микрокапли R , температура T и начальное условие $\lambda^{(0)}$ вводятся в главном окне программы. После нажатия на кнопку «Моделировать» в соответствующем поле выводится результат решения системы уравнений (1) в виде зависимости $\lambda^{(t)}$ от времени t .

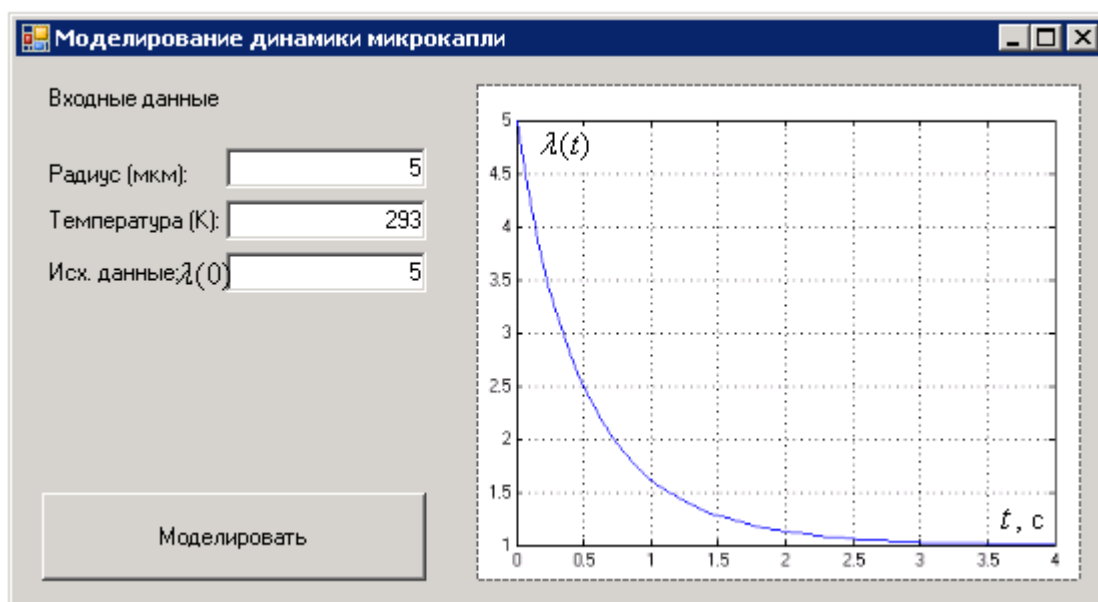


Рисунок 22.2. Окно приложения для моделирования динамики намагничивающихся микрокапель с результатами решения системы уравнений с заданными параметрами

Таким образом, с помощью системы SimulinkCoder сгенерирован исходный код, который использован для разработки приложения на языке C .

Для работы данного приложения не требуется наличия установленного MATLAB – все библиотеки внедрены в конечное приложение.

Использование разработанного приложения для моделирования динамики намагничивающихся микрокапель с учетом их теплофизических характеристик значительно облегчает процесс проведения вычислительного эксперимента[8].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office, систему компьютерной математики MATLAB R2011b и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных

компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно изучив материал, представленный в теоретических указаниях.

Задание 22.1. Разработка структурной схемы модели, описывающей динамические процессы – первое уравнение системы 22.1.

Задание 22.2. Разработка структурной схемы субмодели, описывающей динамические процессы – второе и третье уравнения системы 21.2.

Задание 22.3. Разработка приложения.

Задание 22.4. Выполнение вычислительного эксперимента.

Вариант	R , мкм	T , К	λ (0)
1	5	293	2
2	8	325	3
3	10	320	2
4	15	315	3
5	2	298	2,5
6	4	310	1,5
7	20	300	1,8
8	9	295	1,2
9	14	331	1,1
10	3	333	2

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;

- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Как создается код моделей Simulink?
2. С чего начинается построение кода?
3. Какая информация содержится в файле model.rtw, который является промежуточным представлением блок-схемы?
4. В каком формате хранится информация, представленная в файле model.rtw?
5. Какой компонент вызывает процедура сборки после создания model.rtw?
6. В какой код компонент «Target Compiler Language» превращает модель?
7. В каких файлах после чтения model.rtw и компиляции компонент «Target Compiler Language» выполняет команды?
8. Что указывают файлы, создаваемые «Target Compiler Language» (TCL-файлы)?
9. Куда загружает «Target Compiler Language» предметно-ориентированный код?
10. Создание какой системы, и на основе какого шаблона, является заключительным этапом процедуры сборки?
11. Для чего предназначен шаблон Makefile?

ЛАБОРАТОРНАЯ РАБОТА 23

Разработка систем распознавания изолированного слова (2 ч)

Цель и содержание: изучить методику моделирования систем распознавания изолированного слова, приобрести навыки построения таких моделей в системе НТК.

Теоретическое обоснование

НТК - инструментарий для построения Скрытых Марковских Моделей (СММ) в первую очередь предназначен для построения систем распознавания речи. НТК распространяется свободно. Скачать программное обеспечение можно с официального сайта НТК (НТК: [сайт]. URL: <http://htk.eng.cam.ac.uk/>). Помимо этого, на данном сайте можно найти множество полезной информации по установке, настройке и работе с НТК.

Произнесённое слово можно представить последовательностью векторов или наблюдений O (рисунок 1), определенных как

$$O = o_1, o_2, \dots, o_T \quad (23.1)$$

где o_T – наблюдение в момент времени T .

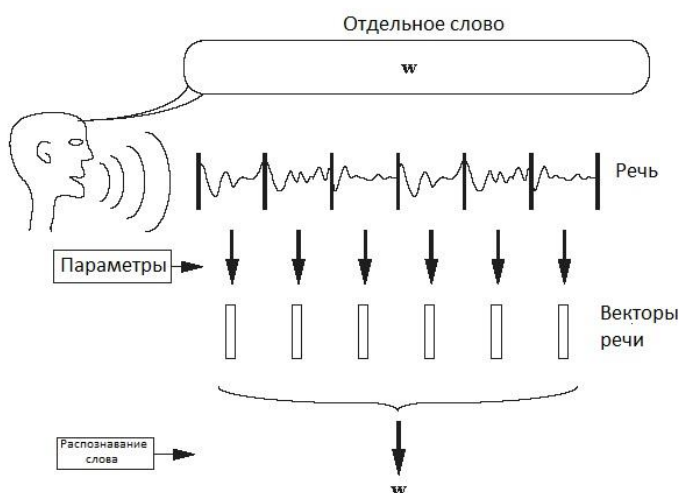


Рисунок 23.1. Задача распознавания изолированных слов

При распознавании речи, основывающемся на СММ, предполагается, что последовательность наблюдаемых векторов речи, соответствующих

какому-то слову, порождена Марковской моделью, как показано на рисунке 23.2. Марковская модель представляет собой автомат с конечным числом состояний, изменяющий свое состояние один раз в каждую единицу времени, и в каждый момент времени t , когда модель находится в состоянии j , вектор речи O_t генерируется исходя из плотности вероятностей $b_j(O_t)$. Кроме того, переход от состояния i к состоянию j также является вероятностным и совершается под управлением отдельной вероятности a_{ij} .

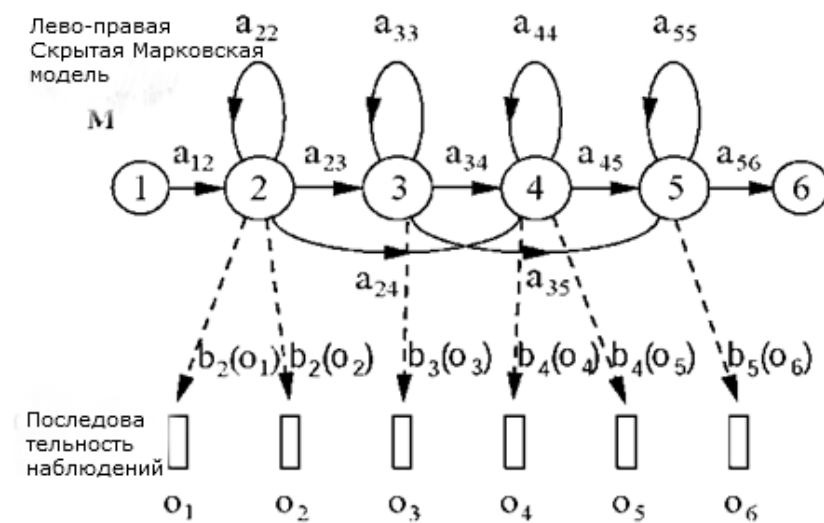


Рисунок 23.2. Левая-правая Скрытая Марковская модель используемая для распознавания речи

Сначала СММ обучают на каждом слове, взятом из словаря, используя множество экземпляров одного и того же слова. В случае представленном на рисунке 23.3, словарь состоит только из трех слов: "один", "два" и "три". Затем, при распознавании некоторого неизвестного слова, вычисляется правдоподобие каждой модели генерирующей данное слово, так что искомое слово определяется по наиболее вероятной модели.

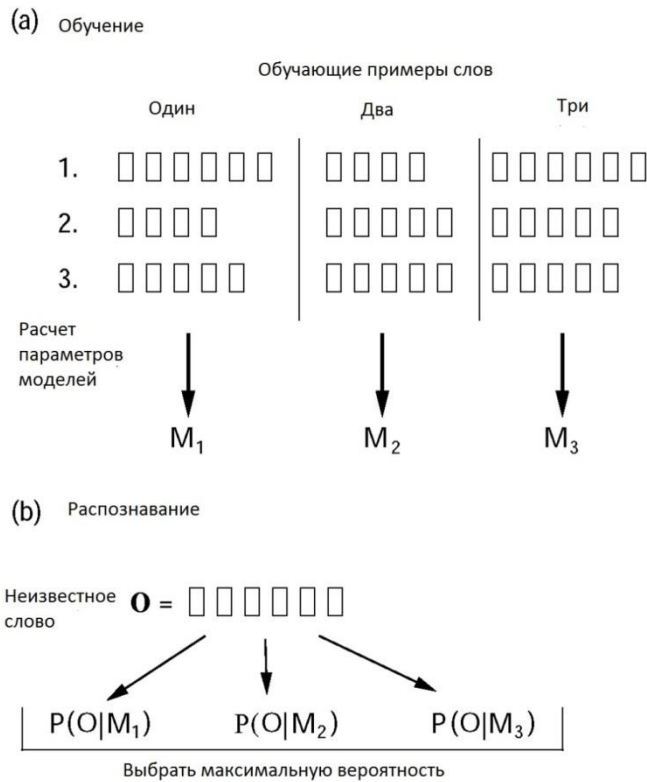


Рисунок 23.3. Применение СММ для распознавания изолированных слов

С помощью алгоритма Витерби производится первое грубое определение параметров СММ (матрица переходов между состояниями $\{a_{ij}\}$ среднюю арифметическую $\hat{\mu}_j$ (23.2) и матрицу ковариации (меру разброса случайной величины) $\hat{\Sigma}_j$ (23.3) для каждого состояния СММ).

$$\hat{\mu}_j = \frac{1}{T} \sum_{t=1}^T o_t \quad (23.2)$$

$$\hat{\Sigma}_j = \frac{1}{T} \sum_{t=1}^T (o_t - \mu_j)(o_t - \mu_j)' \quad (23.3)$$

где o_t – векторы речи.

Данная процедура реализовано в инструменте НТК, именуемом HInit.

Алгоритм Витерби так же используется и при распознавании:

Предположим, что $\phi_j(t)$, для данной модели M , представляет максимальное правдоподобие наблюдения последовательности векторов речи от O_1 до O_t и пребывания в состоянии j в момент времени t . Это частное

правдоподобие можно эффективно вычислить с использованием следующего рекуррентного соотношения:

$$\phi_j(t) = \max_i \{ \phi_i(t-1) a_{ij} \} b_j(O_t) \quad (23.4)$$

где

$$\phi_1(1) = 1 \quad (23.5)$$

$$\phi_j(1) = a_{1j} b_j(O_1) \quad (23.6)$$

Вероятность наблюдений $b_j(O_t)$ находится следующим образом:

$$b_j(O_t) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} e^{-\frac{1}{2}(O_t - \mu_j)' \Sigma_j^{-1} (O_t - \mu_j)} \quad (23.7)$$

Где n – размерность векторов наблюдений O_t ;

Для $1 < j < N$. Тогда максимальное правдоподобие $P(O/M)$ имеет вид:

$$\phi_N(T) = \max_i \{ \phi_i(T) a_{iN} \} \quad (23.8)$$

Что касается рекуррентного оценивания, прямое вычисление правдоподобия ведет к потере значащих разрядов, поэтому вместо этого вычисляют логарифм правдоподобия. При этом вместо уравнения (4) получаем:

$$\psi_j(t) = \max_i \{ \psi_j(t-1) + \log(a_{ij}) \} + \log(b_j(O_t)) \quad (23.9)$$

Далее производится уточнение параметров для каждого состояния модели (23.10) и (23.11) используя алгоритм Баума-Велша:

$$\hat{\mu}_j = \frac{\sum_{t=1}^T L_j(t) O_t}{\sum_{t=1}^T L_j(t)} \quad (23.10)$$

$$\hat{\Sigma}_j = \frac{\sum_{t=1}^T L_j(t) (O_t - \mu_j)(O_t - \mu_j)'}{\sum_{t=1}^T L_j(t)} \quad (23.11)$$

где $L_j(t)$ обозначает вероятность пребывания вектора O_t в состоянии j в момент времени t .

Конечно, чтобы применить соотношения (23.10) и (23.11), нужно рассчитать вероятность состояния $L_j(t)$. Это эффективно делается с

использованием так называемого алгоритма «вперед-назад» (Forward-Backward algorithm) или Баума-Велша. Пусть прямая вероятность $a_j(t)$ для некоторой модели M N состояниями определена в виде:

$$a_j(t) = P(O_1, \dots, O_t, x(t) = j/M) \quad (23.12)$$

То есть, $a_j(t)$ – совместная вероятность наблюдения первых t векторов речи для состояния j в момент времени t . Эта прямая вероятность может быть эффективно рассчитана по следующей рекуррентной формуле:

$$a_j(t) = \left[\sum_{i=2}^{N-1} a_i(t-1) a_{ij} \right] b_j(O_t) \quad (23.13)$$

Начальные условия для вышеупомянутого рекуррентного соотношения имеют вид:

$$a_1(1) = 1 \quad (23.14)$$

$$a_j(1) = a_{1j} b_j(O_1) \quad (23.15)$$

для $1 < j < N$, а конечные условия задаются так:

$$a_N(T) = \sum_{i=2}^{N-1} a_i(T) a_{iN} \quad (23.16)$$

Заметим, что из определения $a_j(t)$ следует:

$$P(O/M) = a_N(T) \quad (23.17)$$

Следовательно, вычисление прямой вероятности позволяет получить полное правдоподобие $P(O/M)$. Обратная вероятность $\beta_j(t)$ определяется следующим образом:

$$\beta_j(t) = P(O_{t+1}, \dots, O_T / x(t) = j, M) \quad (23.18)$$

Как и в случае прямой вероятности, эта обратная вероятность может быть эффективно вычислена с использованием следующего рекуррентного соотношения:

$$\beta_j(t) = \sum_{j=2}^{N-1} a_{ij} b_j(O_{t+1}) \beta_j(t+1) \quad (23.19)$$

с начальным условием

$$\beta_j(T) = a_{iN} \quad (23.20)$$

для $1 < j < N$, и конечным условием

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(O_1) \beta_j(1) \quad (23.21)$$

Заметим, что в приведенных выше определениях прямая вероятность есть совместная вероятность, тогда как обратная вероятность есть условная вероятность. Это несколько асимметричное определение является предумышленным, поскольку позволяет определить вероятность нахождения в состоянии как произведение этих двух вероятностей. По определению,

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(O_1) \beta_j(1) \quad (23.22)$$

$$a_j(t) \beta_j(t) = P(O, x(t) = j/M) \quad (23.23)$$

Отсюда

$$L_j(t) = P(x(t) = j/O, M) = \frac{P(O, x(t) = j/M)}{P(O/M)} = \frac{1}{P} a_j(t) \beta_j(t) \quad (23.24)$$

где $P = P(O/M)$

Теперь мы располагаем всей информацией для осуществления рекуррентного оценивания параметров СММ с использованием алгоритма Баума-Уэлча. Шаги этого алгоритма можно представить следующим образом.

1. Для каждого рекуррентно оцениваемого векторного/матричного параметра, следует выделить место в памяти для организации суммирования в числителе и знаменателе выражений (23.10) и (23.11). Эти места памяти представляют собой накапливающие сумматоры.

2. Вычисляют прямые и обратные вероятности для всех состояний j и моментов времени t .

3. Для каждого состояния j и времени t обновляют содержимое накапливающих сумматоров, используя вероятность $L_j(t)$ и текущий вектор наблюдения O_t .

4. Конечные значения накапливающего сумматора используют для вычисления новых значений параметров.

5. Если значение $P = P(O/M)$ для данной итерации не выше такового для предыдущей итерации, тогда производится останов, в противном случае следует повторить вышеупомянутые шаги, используя новые рекуррентно оцененные значения параметров[19].

На этом закончим обсуждение распознавания изолированного слова с использованием СММ. Не существует инструмента НТК, который непосредственно реализует вышеупомянутый алгоритм Витерби. Вместо этого используют инструмент HVite, который, вместе с его библиотеками поддержки, HNet и HRes.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

В данной лабораторной работе будет разрабатываться система распознавания русских цифр. Для этого необходимо предварительно скачать и установить НТК в папку Windows с официального сайта НТК (НТК: [сайт]. URL:<http://htk.eng.cam.ac.uk/>).

Все системы распознавания речи состоят из следующих компонентов:

А) Языковая модель или грамматика – Языковая модель содержит в себе огромный перечень слов и вероятности их появления в данной последовательности. Грамматика – значительно меньший файл содержащий набор комбинаций слов.

Б) Акустическая модель – содержит статистическое представление звуков и фоном из которых состоит каждое слово в языковой модели или грамматике.

Шаг 1 – Система задания грамматики (Task Grammar)

Назначение создаваемой нами системы – обеспечить управляемый голосом интерфейс для набора цифр. Примеры типичного ввода:

ноль

один

два

...

девять

Сначала создайте папку со своей фамилией на английском языке (например Ivanov). Для удобства вызова функций НТКскопируйте в свою папку все файлы из папки c:/windows/htk/bin.win32/ с расширением .exe. (HVite.exe, HCompV.exe, HDMan.exeи т.д.). Для работы с кириллицей необходимо создать текстовый файл конфигурации Confсодержащий в себе следующее:

NONUMESCAPES = TRUE

Далее в данной папке создайте текстовый файл и назовите его gram. Данный текстовый файл должен содержать в себе следующее:

\$digit = один | два | три | четыре | пять |

шесть | семь | восемь | девять | ноль;

(SENT-START (\$digit) SENT-END)

Словесную сеть можно создать автоматически, используя инструмент HParse и используя в качестве исходных данных описанную выше грамматику, так что если файл gram содержит вышеупомянутую грамматику.

Для этого необходимо вызвать командную строку windows (Пуск>выполнить>cmd). В командной строке перейти в созданную папку:

```
Cd ../Ivanov/
```

Далее необходимо выполнить следующую команду

```
HParse -CConfggramwdnet
```

создаст эквивалентную сеть слов в файле wdnet (wordnet).

Шаг 2 – Словарь произношений (Pronunciation Dictionary)

В силу того что в данной лабораторной работе разрабатывается система распознавания изолированного слова то в качестве фонем будут выступать не звуки, а целое слово.

Для этого необходимо создать текстовый файл dict рисунок 23.4.

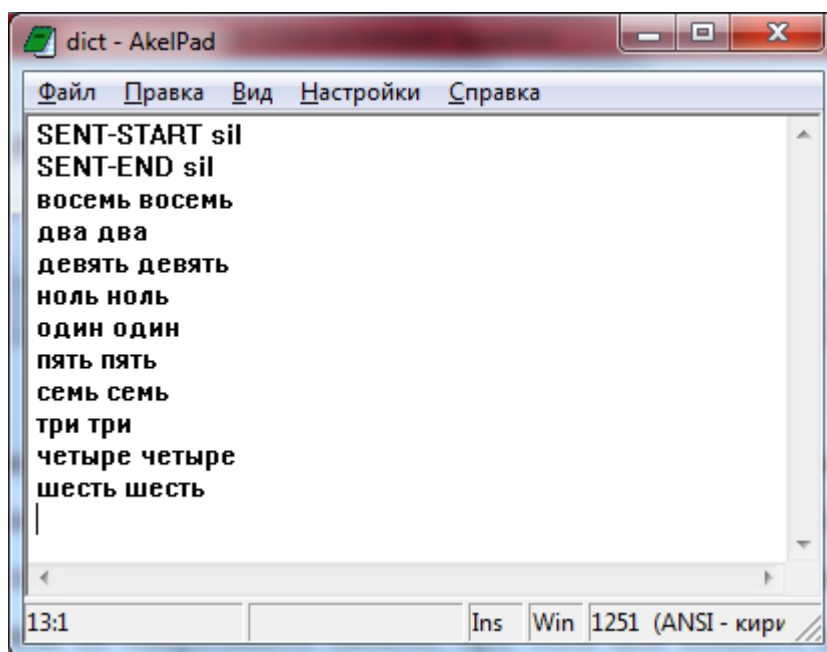


Рисунок 23.4.Содержание словаря

Данный файл является словарем. Слова в нем обязательно должны быть расположены в алфавитном порядке. Слова SENT-ENDи SENT-START –

являются обозначением тишины вначале и конце произношения слова. Им соответствует специальная фонема sil.

Помимо словаря необходимо создание файла Monophones содержащего только фонемы. В нашем случае перечень слов плюс фонема тишины sil (рисунок 23.5).

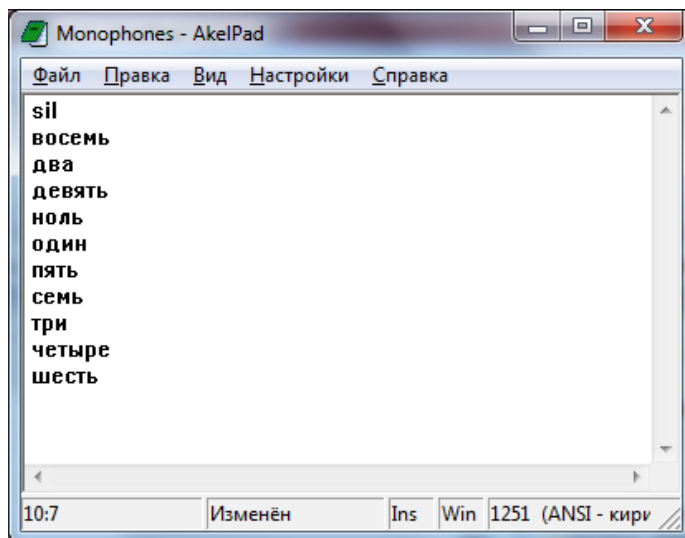


Рисунок 23.5. Перечень фонем в файле Monophones

Шаг 3 – Запись данных

Для записи аудиофайлов вашего голоса в первую очередь убедитесь что выполняются все основные условия для получения качественных данных:

1. В комнате в который вы находитесь тихо насколько возможно
2. Колонки вашего компьютера отключены во избежание наложения голоса
3. Микрофон расположен на расстоянии 1-2 см от рта.

Для записи файлов есть специальная функция HTKHSLab.

Вызвать ее можно следующим образом:

HSLab nome

Где nome имя записываемого файла. Выбор программы для записи полностью за вами. Рекомендую пользоваться бесплатно распространяемой программой Audacity скачать можно на сайте производителя (audacity: [сайт]. URL: <http://audacity.sourceforge.net/download/?lang=ru>).

Обязательные условия при записи файлов: Моно, 16 kHz, 16 bit(рисунок 23.6).

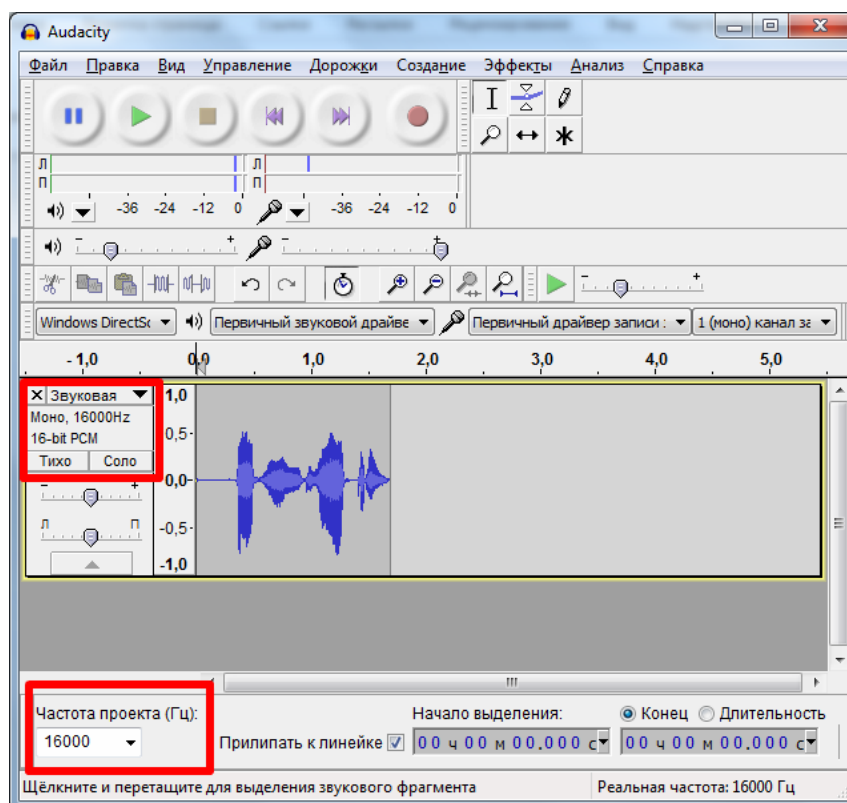


Рисунок 23.6.Интерфейс программы Audacity

Запишите звучание цифр от 0 до 9 по 5 раз для каждой цифры. От записываемых вами данных будет зависеть качество распознавания речи. Старайтесь, чтобы уровень записи находился между 0.5 и -0.5, средний уровень был около 0.3 до -0.3. Это нормально, если иногда уровень записи будет превышать границы 0.5 и -0.5, но не допускайте пересечения границ 1.0 и -1.0.

Создайте внутри своей папки папку wav и сохраните туда записанные файлы. Имена файлов должны быть следующими (nol, odin, dva, tri, chetire, piat, schest, sem, vosem, devat):

- 1.nol.wav
- 2.nol.wav
- 3.nol.wav
- 4.nol.wav

5.nol.wav

...

1.devat.wav

2.devat.wav

3.devat.wav

4.devat.wav

5.devat.wav

Цифра в начале каждого файла необходима для короткой записи при создании файлов транскрипций слов и фонем (MasterLabelFile). Так для 5 примеров звучания цифры будет соответствовать одна транскрипция. Создание данных транскрипций рассмотрено в следующем пункте.

Шаг 4 – Создание файлов транскрипций

Отправной точкой для обоих наборов фонетических транскрипций является орфографическая транскрипция в формате разметки НТК (НТК label format). Это можно сделать довольно просто, используя текстовый редактор

Создаем файл words.mlf рисунок 23.7.

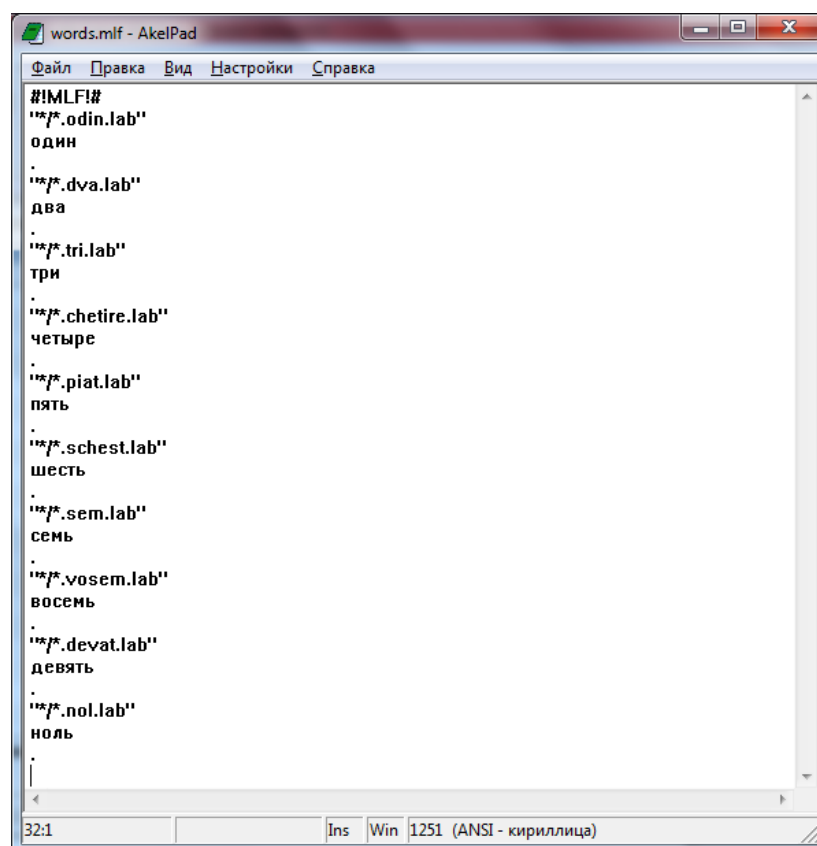


Рисунок 23.7. Транскрипция слов words.mlf

В силу того что мы строим систем изолированного распознавания слова то транскрипция фонем будет идентична файлу words.mlf.

Шаг 5 – Кодирование акустических данных

Для построения системы распознавания речи, акустические данные необходимо обработать методами спектрального анализа. В случае с распознаванием речи используются мел-частотные кепстральные коэффициенты (MFCC). Мел – единица высоты звука, основанная на восприятии этого звука нашими органами слуха. Как известно, амплитудно-частотная характеристика (АЧХ) человеческого уха даже отдаленно не напоминает прямую, и амплитуда – не совсем точная мера громкости звука (рисунок23.8).

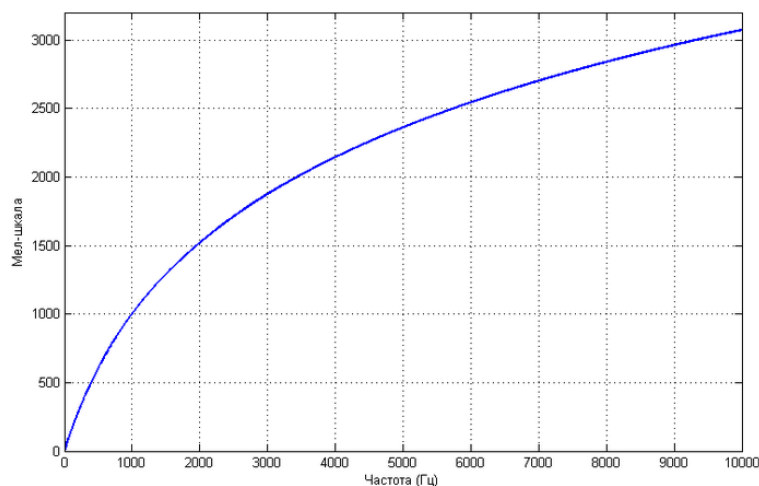


Рисунок 23.8. Логарифмическая зависимость восприятия звука человеком

Такая зависимость описывается простой формулой:

$$m = 1125 \ln(1 + f/700) \quad (23.25)$$

В НТК реализована система кодирования данных. Для задачи параметров кодирования данных необходимо создать файл конфигурации config_wav содержащий следующее:

```
SOURCEFORMAT = WAV
TARGETKIND = MFCC_0_D_A
TARGETRATE = 100000.0
SAVECOMPRESSED = T
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = F
```

Говоря коротко, они определяют, что результатом преобразования должны быть MFCC, при использовании C_0 в качестве энергетического

компонента, протяженность кадра 10 мс (НТК использует единицы по 100 нс), результат должен быть сохранен в сжатом виде, также должна быть добавлена контрольная сумма. БПФ должно использовать окно Хеминга, а сигнала предварительно предусажается (preemphasis) фильтром первого порядка с коэффициентом 0.97. Гребенка фильтров должна иметь 26 каналов, а результатом являются 12 коэффициентов MFCC. Переменная ENORMALISE обеспечивает нормализацию энергии в записанных аудио файлах и по умолчанию имеет значение true. Нормализация невозможна при работе с живым звуком, и поскольку система в конечном счете предназначена для работы с живым звуком, этой переменной должно быть присвоено значение false.

Далее создаем папку mfccsv которую будут скопированы перекодированные аудио файлы.

Последним этапом для обработки данных необходимо создать файл codedata.scr в котором будет прописан путь до **всех** аудиофайлов и до папки mfcc(рисунк 23.9).



Рисунок 23.9. Содержание файла traindata.scp

Теперь осталось воспользоваться функцией HСоруv командной строке windows:

```
HCopy -T 1 -Cconfig_wav -Scodedata.scp
```

Функция -T 1 вызывает информацию о ходе кодирования, -Cconfig_wav активирует файл конфигурации, -Scodedata.scpдает ссылки на папки с данными и на папку с результирующими файлами.

Проверьте по завершению данной операции в папке mfccдолжны быть данные вформате .mfcc.

Шаг 6 – Создание прототипа СММ

Первым шагом в обучении НММ является определение модели-прототипа. Параметры этой модели не важны, она предназначена для определения топологии модели.

Создаем текстовый файл с именем protocодержащий текст представленный на рисунке 23.10.

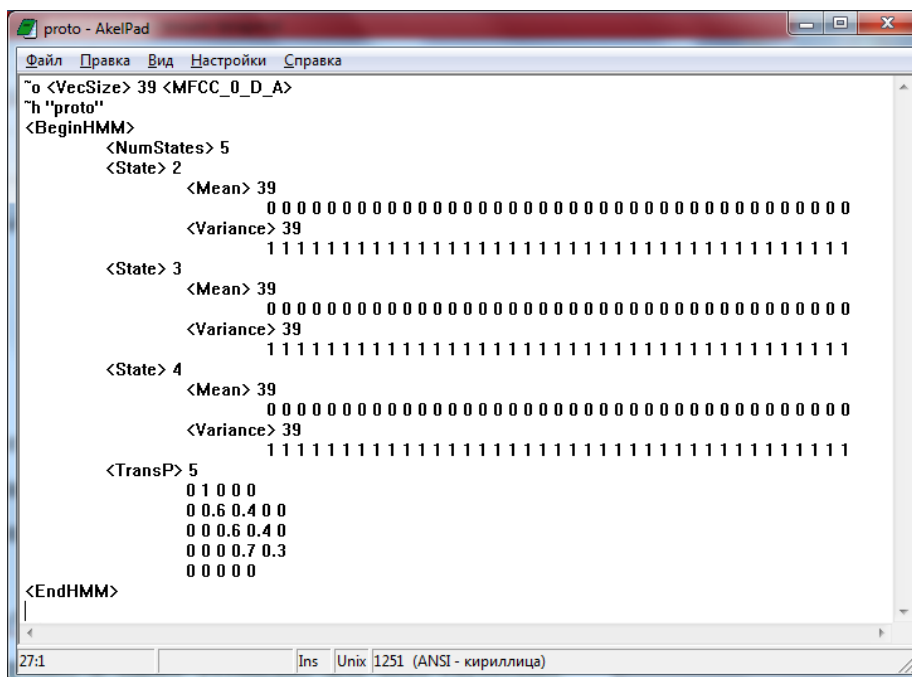


Рисунок 23.10. Прототип СММ

Каждый вектор (ellipsed vector) имеет длину 39. Число 39 рассчитано из суммы длины параметризованного статического вектора (MFCC_0=13) плюс коэффициенты дельта (+13) и плюс коэффициенты ускорения (+13).

НТК инструмент HCompV просканирует набор файлов данных, вычислит глобальные среднее и дисперсию, а также установит для всех гауссовых величин в СММ одинаковые среднее и дисперсию.

Создаем файл `train.scr` содержащий путь до `.mfss` файлов (правая часть файла `codedata.scr`) рисунок 23.11.

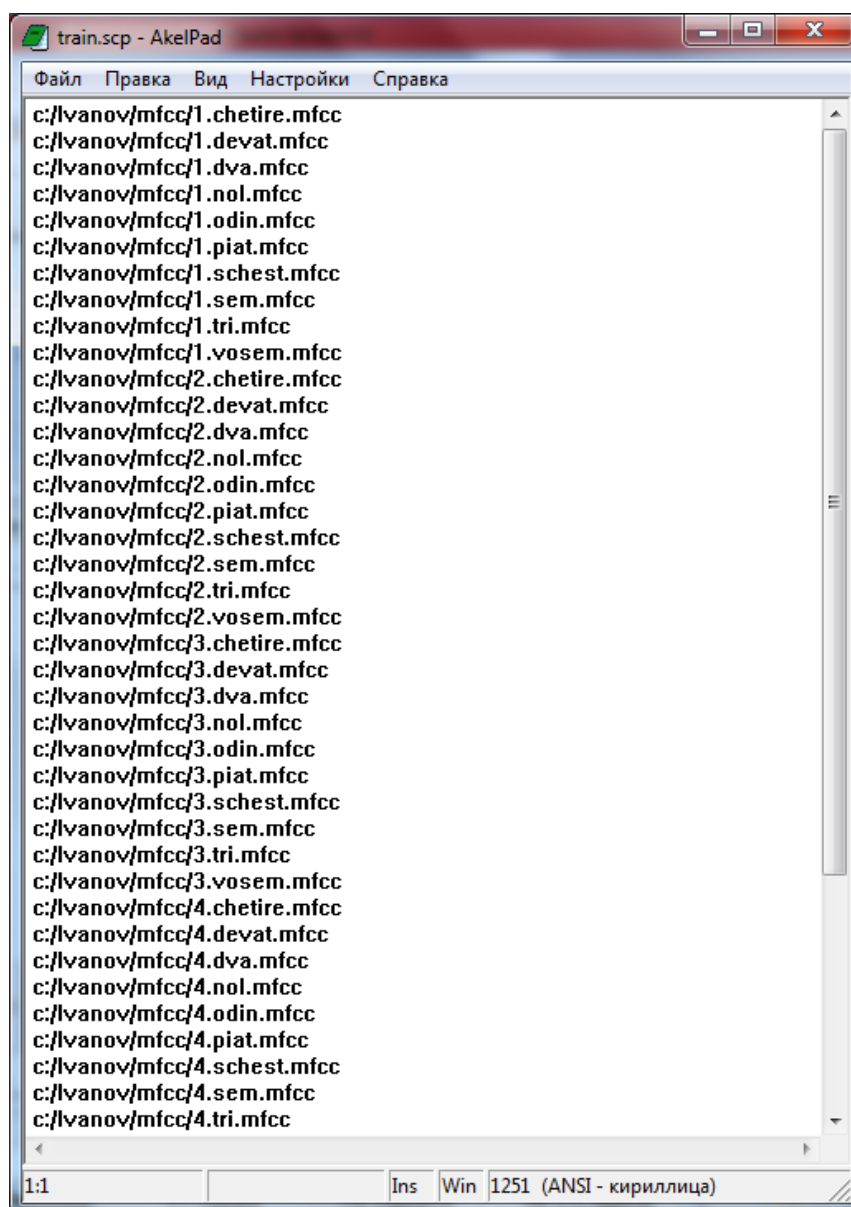


Рисунок 23.11. Содержимое файла train.scp

Создаем файл конфигурации configсодержащий следующее:

TARGETKIND = MFCC_0_D_A

TARGETRATE = 100000.0

WINDOWSIZE = 250000.

0USEHAMMING = T

PREEMCOEF = 0.97

NUMCHANS = 20

CEPLIFTER = 22

NUMCEPS = 12

Создаем папку hmm0 и в командной строке windows активируем:

```
HCompV - C config -f 0.01 -m -S train.scp -M hmm0 proto
```

Будет создана новая версия proto в директории hmm0, в которой нулевые средние и единичные дисперсии заменятся глобальными средними и дисперсиями. Отметим, что прототип НММ определяет тип параметра MFCC_0_D_A. Это значит, что коэффициенты дельта и ускорения будут вычислены и прибавлены к статическим MFCC коэффициентам, вычисленным и сохраненным во время вышеописанного процесса кодирования. . Опцией -f задается макрос округления дисперсии (вызывается vFloors) до одной сотой от общей дисперсии. Это вектор значений, который будет использован для установления минимального уровня дисперсий, оцененных на последующих этапах. Опция -m аналогична и проделываются все те же операции, но уже для среднего.

Необходимо вручную отредактировать результат в hmm0. Переименовать файл vFloors -> macros. Переименовать файл proto -> hmmdefs. В первые две строки файла macros нужно вручную вставить

```
~o
```

```
<STREAMINFO> 1 39
```

```
<VECSIZE> 39<NULLD><MFCC_D_A_0><DIAGC>
```

Так что файл macros должен иметь вид представленный на рисунке 23.12.

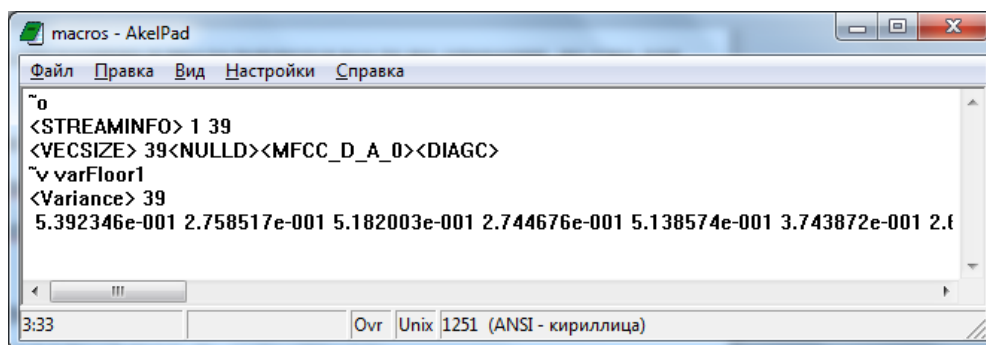


Рисунок 23.12. Содержимое файла macros

В файле hmmdefs прототип фонемы начинается символами

```
~h "proto"
```

и заканчивается

<ENDHMM>

В редакторе взять этот участок в карман и повторить его столько раз, сколько слов в файле monophones включая sil. Затем заменить сочетание "proto" на название слов из monophones рисунок 23.13.

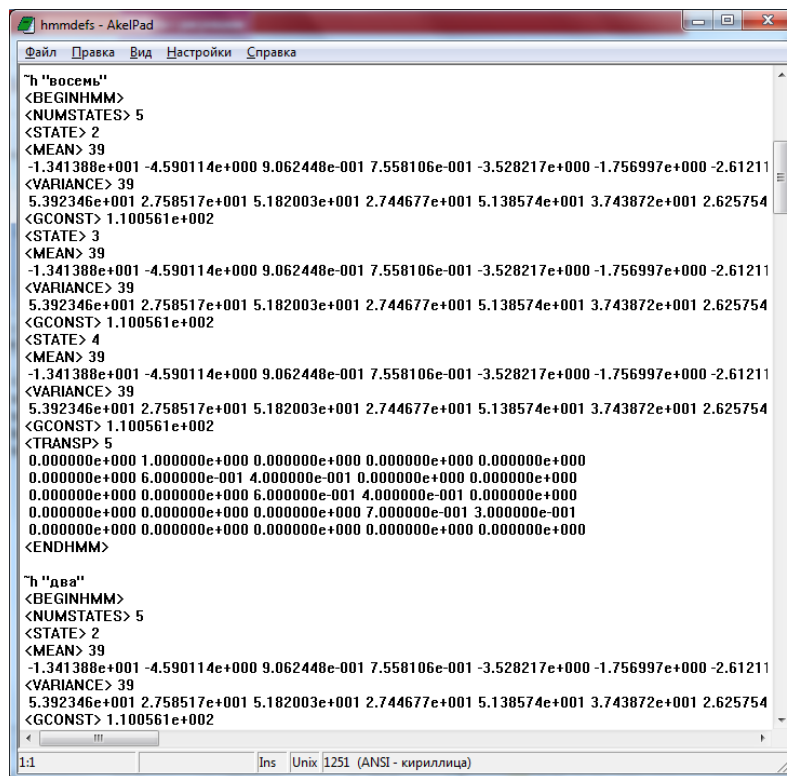


Рисунок 23.13. Содержимое файла hmmdefs

Далее создайте еще 5 папок hmm1, hmm2, hmm3, hmm4, hmm5.

Спомощью функции HERestпересчитайте 5 раз СММ следующим образом:

```
HERest -C Conf -C config -I words.mlf -t 250.0 150.0 1000.0  
-S train.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1  
monophones
```

Эта команда создает файлы hmmdefsи macrosв папке hmm1.

```
HERest -C Conf -C config -I words.mlf -t 250.0 150.0 1000.0  
-S train.scp -H hmm1/macros -H hmm1/hmmdefs -M hmm2  
monophones
```

...

```
HERest -C Conf -C config -I words.mlf -t 250.0 150.0 1000.0  
-S train.scp -H hmm4/macros -H hmm4/hmmdefs -M hmm5  
monophones
```

Эта команда создает файлы `hmmdefs` и `macros` в папке `hmm5`.

Шаг 7 – запуск распознавания голоса с микрофона

Систему распознаванию можно также запустить с реальной входной информацией. Для этого необходимо лишь установить конфигурационные переменные так, чтобы входной аудио сигнал подвергался правильной параметризации. В частности, для создания нового файла конфигурации `config2` содержащий следующее

```
# Codingparameters  
TARGETKIND = MFCC_0_D_A  
TARGETRATE = 100000.0  
SAVECOMPRESSED = T  
SAVEWITHCRC = T  
WINDOWSIZE = 250000.0  
USEHAMMING = T  
PREEMCOEF = 0.97  
NUMCHANS = 26  
CEPLIFTER = 22  
NUMCEPS = 12  
ENORMALISE = F  
# Waveform capture  
SOURCERATE=625.0  
SOURCEKIND=HAUDIO  
SOURCEFORMAT=HTK  
ENORMALISE=F  
USESILDET=T  
MEASURESIL=F
```

```
OUTSILWARN=T
FORCECXTEXP=T
ALLOWXWRDEXP=F
```

Здесь видно, что источником является непрерывный аудио сигнал с периодом дискретизации 62.5 мкс. Детектор тишины включен, поэтому при старте должно быть выполнено измерение отношения уровней речь/тишина. Последняя строка обеспечивает вывод предупреждения при измерении данного параметра тишины.

Когда файл конфигурации настроен на входные данные в виде непрерывного аудио сигнала, HVITE может быть запущен так же, как в предыдущем шаге, с тем лишь исключением, что никакие файлы не нужно указывать в качестве параметров

```
HVite -H hmm5/macros -H hmm5/hmmdefs -C config2
      -w wdnnet -p 0.0 -s 5.0 dict monophones
```

При старте HVITE пригласит пользователя произнести произвольное предложение (приблизительно 4 сек), необходимое для измерения уровней фоновой тишины и речи. Затем предложение будет многократно распознано и, если установлен 1-битовый уровень трассировки, каждое высказывание будет выведено на терминал. Для корректного отображения результатов распознавание кириллицы необходимо запускать систему через MATLAB.

Для корректного отображения символов распознавания речи в командной строке кириллицу следует заменить на латиницу следующим образом:

```
Поменять gramфайл:
$digit = odin | dva | tri | chetire | piat |
schest | sem | vosem | devat | nol;

( SENT-START ( $digit ) SENT-END )

Изменитьсловарьdict:
```

SENT-START sil

SENT-END sil

vosemвосемь

dvaдва

devatдевять

polноль

odinедин

piat пять

sem семь

tri три

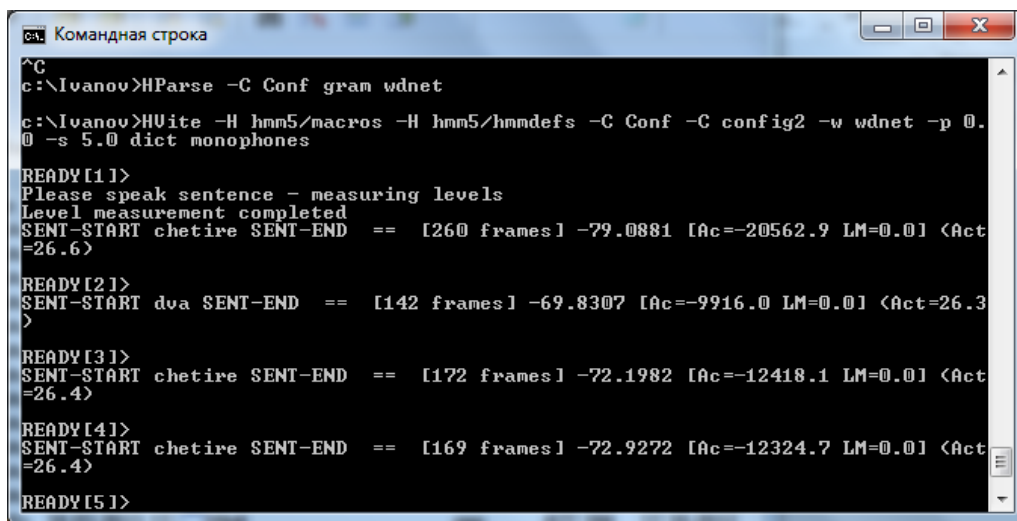
chetire четыре

schest шесть

Запустить функцию HParsec новым gramфайлом:

```
HParse -C Confgramwdnet
```

После этого можно запустить распознавание через командную строку используя функцию HVite (рисунок 23.14)



```
^C
c:\Ivanov>HParse -C Conf gram wdnet

c:\Ivanov>HVite -H hmm5/macros -H hmm5/hmmdefs -C Conf -C config2 -w wdnet -p 0.
0 -s 5.0 dict monophones

READY[1]>
Please speak sentence - measuring levels
Level measurement completed
SENT-START chetire SENT-END == [260 frames] -79.0881 [Ac=-20562.9 LM=0.0] <Act
=26.6>

READY[2]>
SENT-START dva SENT-END == [142 frames] -69.8307 [Ac=-9916.0 LM=0.0] <Act=26.3
>

READY[3]>
SENT-START chetire SENT-END == [172 frames] -72.1982 [Ac=-12418.1 LM=0.0] <Act
=26.4>

READY[4]>
SENT-START chetire SENT-END == [169 frames] -72.9272 [Ac=-12324.7 LM=0.0] <Act
=26.4>

READY[5]>
```

Рисунок 23.14. Распознавание при помощи HViteв cmd

Во время загрузки будет выведена информация, касающаяся разных компонентов системы распознавания. Информация о сети характеризует действительную сеть распознавания, построенную путем расширения решетки с применением текущего набора СММ, словаря и любых заданных

правилах контекстного расширения. В конце каждого высказывания предоставляется количественная информация об общем количестве фреймов, среднем логарифме правдоподобия, приходящимся на один фрейм, общая оценка языковой модели, а также среднее число задействованных моделей.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что из себя представляет Марковская модель?
2. Какие алгоритмы применяются при обучении и распознавании изолированных слов?
3. Опишите процесс обучения модели распознавания слитной речи.

ЛАБОРАТОРНАЯ РАБОТА 24

Создание системы распознавания слитной речи (2 ч)

Цель и содержание: изучить методику моделирования систем распознавания слитной речи, приобрести навыки построения таких моделей средствами НТК.

Теоретическое обоснование

Случай слитной речи получается простым последовательным соединением нескольких СММ. Каждая модель в этой последовательности прямо соответствует предполагаемому базовому символу. Это могут быть либо целые слова при так называемом распознавании связной речи, либо фрагменты слов, такие как фонемы, при распознавании слитной речи. Теперь становится очевидной причина включения входного и выходного непорождающих состояний (начальный и конечный узлы) - эти состояния объединяют модели (рисунок 24.1).

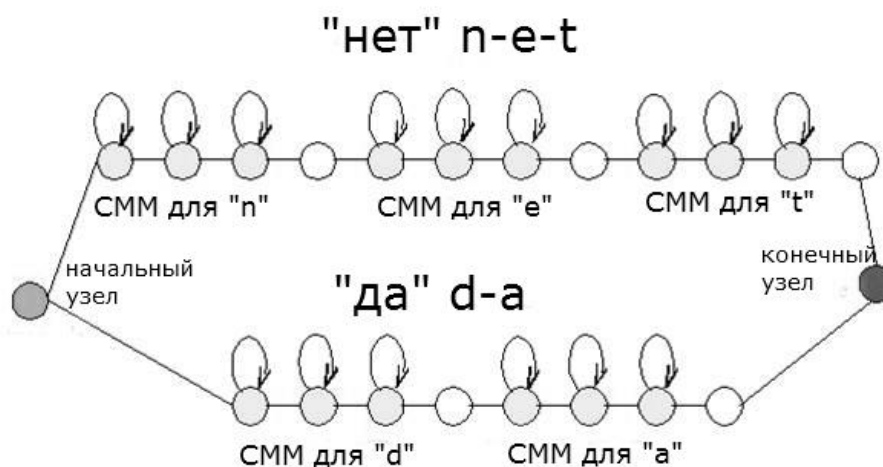


Рисунок 24.1. Пример композитной СММ для фонемного распознавания слов «нет» и «да»

Есть, однако, несколько практических трудностей, которые нужно преодолеть. При слитной речи обучающие данные должны представлять собой слитно произносимые слова и, в общем случае, границы, разделяющие

сегменты речи, соответствующие каждой базовой модели под-слова, не известны. На практике, при малом объеме данных, обычно удается пометить границы вручную. После этого все сегменты, соответствующие данной модели можно извлечь, и можно использовать описанный выше способ обучения на изолированных словах. Однако при таком способе можно обработать весьма ограниченный объем данных, в результате полученные модели будут плохо оценены. Кроме того, даже если бы было большое количество данных, границы, полученные ручной маркировкой, не могут быть оптимальными, как это нужно для СММ. Таким образом, применение в НТК инструментов HInit и HRest для инициализации моделей частей слов (под-слов) рассматривается как процедура начальной загрузки (bootstrap). На практике нужно хорошенько дополнительно поработать, чтобы достигнуть эффективного результата на больших базах обучающих данных. Например, инструмент HERest располагает возможностями для отсечения (pruning) при прямом и обратном проходах, а также для параллельной работы с сетью компьютеров.

На основной стадии обучения используется инструмент HERest, реализующий встроенное обучение (embedded training).

При встроенном обучении используется та же процедура Баума-Уэлча, что и для случая изолированных слов, с тем отличием, что вместо того, чтобы обучать каждую модель индивидуально, все модели обучаются параллельно. Этот процесс состоит из следующих шагов:

1. Выделить память и обнулить сумматоры для всех параметров всех СММ.
 2. Взять следующий обучающий образец произнесения.
 3. Построить составную СММ путем соединения отдельных СММ в последовательность в соответствии с символьной транскрипцией обучающего произнесения.
 4. Вычислить прямые и обратные вероятности для составной СММ.
- Включение промежуточных непорождающих состояний в составную модель

требует некоторых изменений в вычислении прямых и обратных вероятностей, однако они незначительны.

5. Использовать прямые и обратные вероятности для вычисления вероятностей состояния каждого временного фрейма, и произвести обновление содержания сумматоров.

6. Повторять все пункты, начиная с 2, до тех пор, пока все обучающие произнесения не будут обработаны.

7. Использовать сумматоры для вычисления новых оценок параметров для всех СММ.

Эти шаги в дальнейшем можно повторять столько раз, сколько нужно для достижения нужной сходимости. Заметим, что хотя в данной процедуре локализация границ символа в обучающих данных не обязательна (хотя и желательна), символьная транскрипция каждого обучающего произнесения необходима.

Хотя необходимые для осуществления процедуры Баума-Уэлча доработки, обеспечивающие обучение моделей частей слов, относительно невелики, соответствующие доработки алгоритма Витерби более существенны[20].

Используемая в НТК альтернативная формулировка алгоритма Витерби названа моделью пересылки маркера (Token Passing Model). Коротко говоря, модель пересылки маркера делает явным понятие пути состояний для выравнивания. Представим, что каждое состояние j СММ в момент t владеет единственным передвигаемым маркером, который содержит, помимо другой информации, частичный логарифм вероятности $\psi_j(t)$. Этот маркер тогда представляет частичное соответствие между последовательностью наблюдений от O_1 до O_t и моделью, при условии, что модель находится в состоянии j в момент времени t . Представленный рекуррентным уравнением (31) алгоритм продолжения пути затем заменяется эквивалентным алгоритмом пересылки маркера, выполняемым для каждого фрейма,

соответствующего моменту времени. Ключевые шаги этого алгоритма таковы.

Перешлите копию каждого маркера состояния i всем связанным состояниям j , увеличивая логарифм вероятности копии на $\log|a_{ij}| + \log|b_i(O_t)|$.

Проверьте маркеры каждого состояния и удалите все, за исключением маркера с наивысшей вероятностью.

На практике, при работе с непорождающими состояниями, необходимы некоторые модификации, однако они простые, если предположить, что маркеры входных состояний представляют пути, простирающиеся до момента времени $t - \delta t$, а маркеры выходных состояний представляют пути, простирающиеся до момента времени $t + \delta t$.

Обучение и распознавание в случае слитной речи производится благодаря все тем же алгоритмам Баума-Уэлча и Витерби.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер MDT750/ i75930K/ 4 D8192D42133; мультимедиа-проектор Epson; магнитно-маркерная доска и программное обеспечение: операционную систему Windows 7 и выше; Microsoft Office.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

В папке со своей фамилией создайте новую папку ContSpeech и скопируйте все .exe НТК-файлы. Скопируйте файлы config_wav, Conf, config,

config2. Из папки C:\Windows\HTK\samples\HTK Tutorial скрипты prompts2wlist, prompts2mlf, maketrihed.

Шаг 1 – Система задания грамматики (TaskGrammar)

Назначение создаваемой нами системы – обеспечить управляемый голосом интерфейс для набора телефонных номеров. Таким образом, система распознавания должна обрабатывать числовые последовательности.

Примеры типичного ввода:

Один два три пять четыре пять

Один восемь девять ноль два пять шесть

Один три пять

Один четыре четыре пять

...

В HTK имеется язык задания грамматики, используемый для подробного описания таких простых грамматик как эта (мы применяли его при создании грамматики распознавания изолированного слова). Он состоит из перечня объявлений переменных, за которыми следуют стандартные выражения, определяющие слова, подлежащие распознаванию. Для приложения голосового набора номеров может подойти грамматика записанная в файл cont_gram:

```
$digit = один | два | три | четыре | пять |  
шесть | семь | восемь | девять | ноль;
```

```
([SENT-START] (<$digit> ) [SENT-END] )
```

Здесь вертикальные полосы (|) обозначают альтернативы, квадратные скобки ([]) обозначают необязательные элементы (в данном случае не обязательными элементами является тишина перед и после произнесения набора цифр SENT-START и SENT-END), а угловые фигурные скобки(< >) обозначают одно или более повторений. Законченная грамматика может быть изображена в виде сети, как показано на рисунке 24.2.

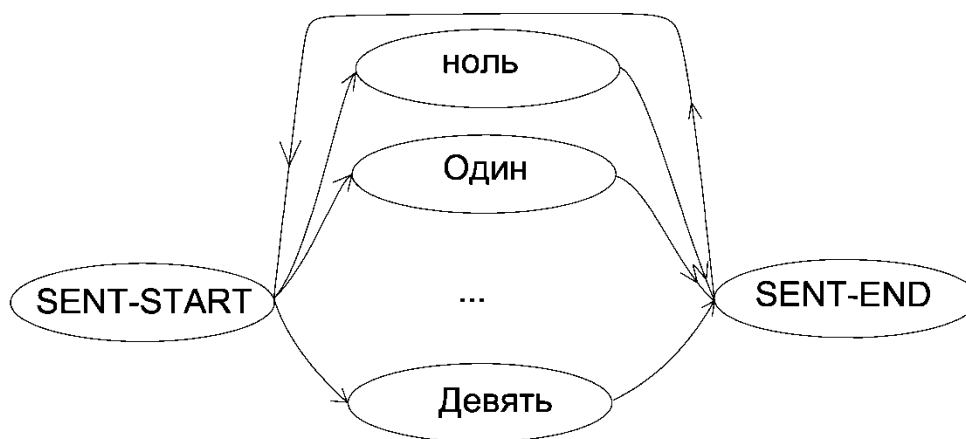


Рисунок 24.2. Грамматика системы распознавания

Рассмотренное выше высокоуровневое представление грамматики удобно в использовании.

Далее применяем функцию HParse:

```
HParse -CConf cont_gram cont_wdnet
```

Получаем эквивалентную сеть слов в файле cont_wdnet.

Шаг 2 – Словарь произношения (Pronunciation Dictionary)

Обычно первым шагом при формировании словаря произношений является создание текстового файла содержащего в первом столбце слова из грамфайла а во втором соответствующее им фонемное представление. Такой словарь должен быть фонетически сбалансирован т.е. каждая фонема находящаяся в нем должна прозвучать как минимум в трех разных словах, для обучения модели. Более того, для создания устойчивых акустических моделей необходимо при обучении использовать большой набор предложений, содержащих много слов и, желательно, фонетически сбалансированных.

Создаем в ручную файл cont_dict рисунок24.3. Обратите внимание что слова должны быть сортированы в алфавитном порядке.

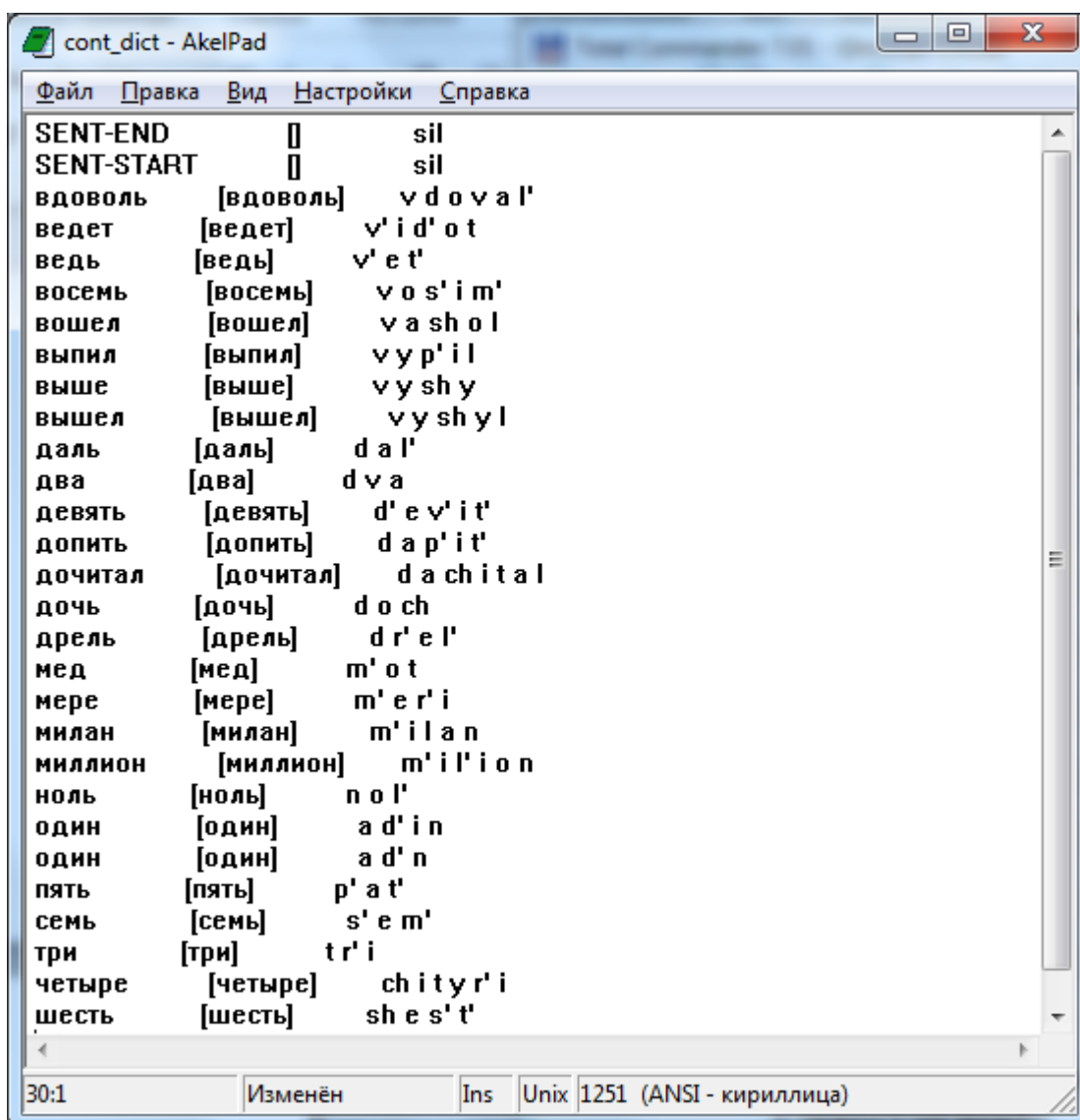


Рисунок 24.3. Словарь произношений

Помимо чисел, которые включены в cont_gramфайл в словаре находятся дополнительные слова необходимы для создания акустически сбалансированной системы распознавания.

Далее создайте текстовый файл promptsв котором будут содержаться:

*/S01 восемь восемь пять ноль четыре миллион дочь мере дочь

*/S02 вышел мед вошел дрель дрель

*/S03 даль

*/S04 четыре четыре три ноль пять восемь один четыре два семь четыре
четыре

*/S05 дочь милан мед

*/S06 мед даль дочь вдоволь дрель мере миллион миллион дрель ведь
мед вдоволь ведет дрель выпил вдоволь дочь ведь мед даль дочь дрель ведет
милан

*/S07 допить мед милан дочь допить мере вошел ведь

*/S08 выше дрель ведет милан дочитал дрель даль вышел дочь вошел
мере мере выпил выше ведет дочитал вдоволь мед

*/S09 мед даль милан миллион ведь выше ведет выпил даль вошел
допить ведь вдоволь вышел допить вдоволь допить миллион дрель даль дочь
вошел мед

*/S10 ведет вошел ведет дочь милан вдоволь ведет мере допить мед

*/S11 вышел ведет мере

*/S12 мед выше выпил милан милан допить вышел выпил даль мед
миллион миллион

*/S13 мед выше мере вошел милан выпил миллион мере выше выпил
вдоволь вошел мере

*/S14 даль ведь миллион

*/S15 вдоволь выпил ведь милан дочитал ведь

*/S16 милан милан

*/S17 два пять шесть восемь один два четыре восемь девять один два
ноль семь три

*/S18 пять один три три один шесть восемь пять два ноль ноль

*/S19 шесть пять ноль один три шесть семь пять восемь шесть один
четыре ноль пять ноль четыре три

*/S20 ведь вошел мед дочитал выпил мере дрель мере вышел мед милан
даль дрель ведет дрель вошел мере выпил даль выпил вошел выше

*/S21 допить дочь ведь дрель ведь выпил мед дочитал даль допить
вышел милан милан дочитал мед вышел ведь ведь

*/S22 семь три два четыре ноль три ноль семь семь три семь

*/S23 ведет даль мере ведет дочитал вошел дочитал ведет вышел вышел
ведь миллион милан

*/S24 даль дрель миллион выпил ведет вышел выпил ведет мед мед миллион мере вдоволь вошел вышел выпил милан мед дрель ведет

*/S25 вдоволь мере мед даль допить вдоволь даль дочитал дочитал выше даль дрель

*/S26 три один шесть три девять три семь четыре один

*/S27 даль даль мере ведь дрель вдоволь выше вышел дочитал ведь мере мере

*/S28 мере вышел вышел мере дочь дочь ведь ведет дрель допить допить мере дочитал дочь вышел мед дочь милан милан допить дочитал

*/S29 четыре шесть пять пять ноль один пять пять восемь три два пять девять шесть пять ноль семь пять девять

*/S30 четыре три два пять

Первая колонка promptsфайла содержит названия аудиофайлов (например S01) последующий текст на русском языке, это текст необходимый записать в аудио файл.

Список эталонных обучающих слов wlist может быть получен из prompts автоматически. Используем скрипт prompts2wlist который мы предварительно скопировали из папки НТК:

```
Perlprompts2wlistprompts2wlist
```

Данный скрипт создает wlistфайл содержащий перечень всех слов из prompts. Откройте файл wlistи добавьте слова SENT-ENDи SENT-STARTрисунок 24.4.

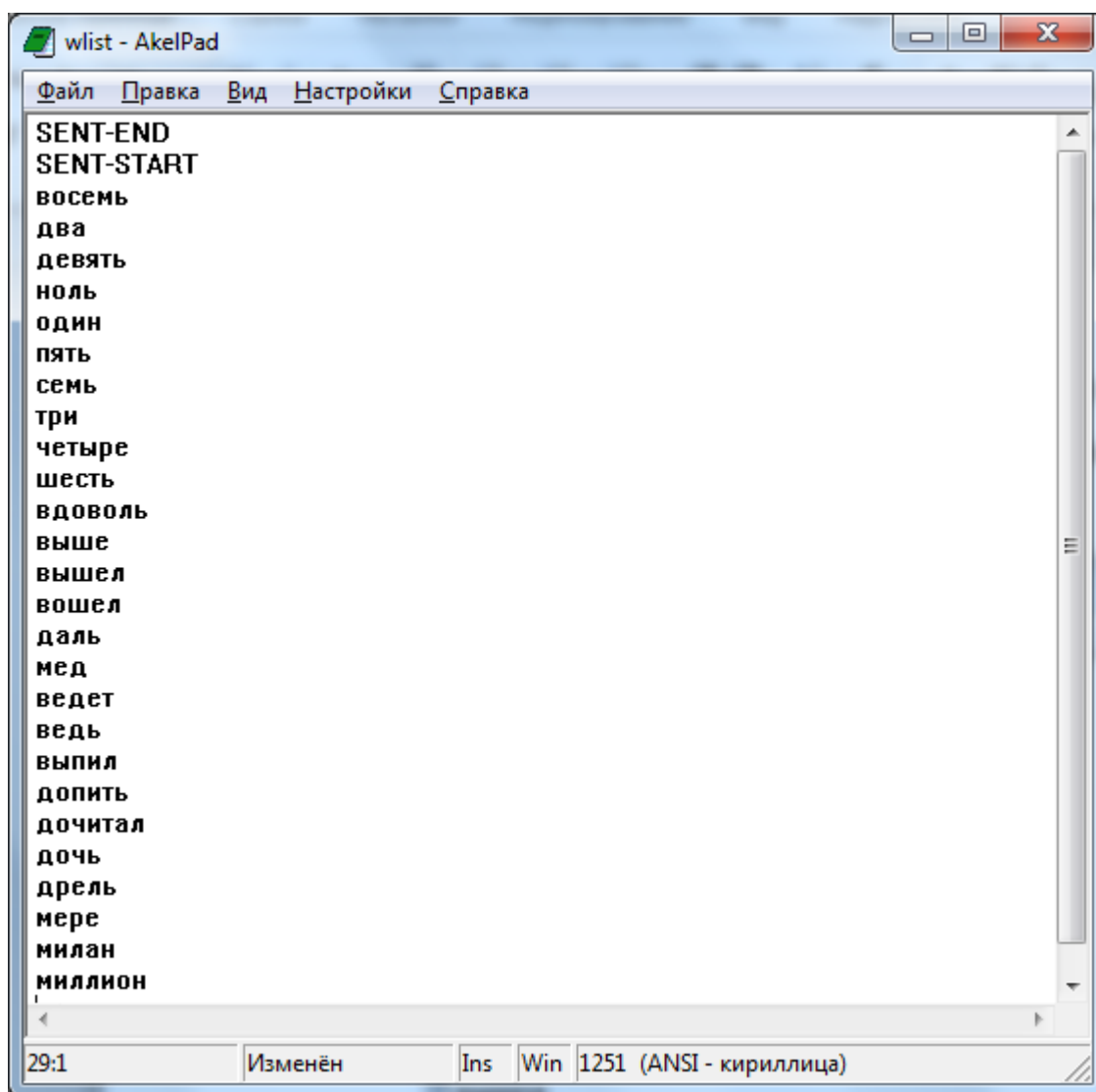


Рисунок 24.4. Содержание файла wlist

Создаем файл global.ded содержащий в себе следующее:

SspMPsil silsp (не забудьте включить пустую строку в конце)

Словарь произношений мы будем формировать за счет функции HDMan используя за основу файл cont_dict. Его фонемный набор будет принят без изменений, лишь будут удалены метки ударений и будут добавлены короткие паузы (sp) в конце каждой фонетической транскрипции за счет команды ASv файле global.ded. Если словарь содержит какие-то метки паузы, команда MP объединит фонемы sil и sp в одну sil.

Далее используем НТК:

```
HDMan -A -D -T 1 -C Conf -m -w wlist -n monophones1 -i -l  
dlog dict cont_dict
```


Будет создан новый словарь `dict` и список фонем `monophones 1`, путем поиска в исходном слова `recont_dict` фонетических транскрипций для каждого слова в `wlist` (рисунок 24.4). В данной ситуации `wlist` должен быть всего лишь сортированным списком слов, которые встречаются в приведенной выше грамматике.

Опция `-l` дает инструкцию `HDMan` вывести файл протокол `dlog` который содержит различную статистику о построении словаря. В частности, тут отмечаются пропуски слов. `HDMan` может также вывести список используемых фонем, названный в данном случае `monophones1`. Для однажды записанных данных обучения и тестирования, будут оценены СММ для всех входящих фонем.

Общий формат каждой записи словаря:

`WORD [outsym] p1 p2 p3 ....`

что означает, что слово `WORD` произносится как последовательность фонем `p1 p2 p3 ....`. Строка в квадратных скобках определяет выходную строку, когда слово будет распознано. Если она пропущена, то будет выведено само слово. Если скобки пустые, тогда ничего не выводится. Файл `dict` должен выглядеть как представлено на рисунке 19. Заметьте что в конце каждого фонемного представления слова находится фонема `sp` (`shortpause`). Она необходима для создания системы слитной речи. Более подробно значение данной фонемы будет описано ниже.

Скопируйте файл `monophones1` и переименуйте его в `monophones0` откройте файл `monophones0` и удалите из перечня фонем фонему `sp`.

Шаг 3 – Запись данных

Создайте папку `train(wav)` и запишите предложения из файла `prompts`.

При этом названия аудиофайлов должны быть озаглавлены в соответствии с первым столбцом файла `prompts` (`S01.wav`, `S02.wav`, `S03.wav...S30.wav`). В результате должны быть получены 30 аудиофайлов для обучения модели.

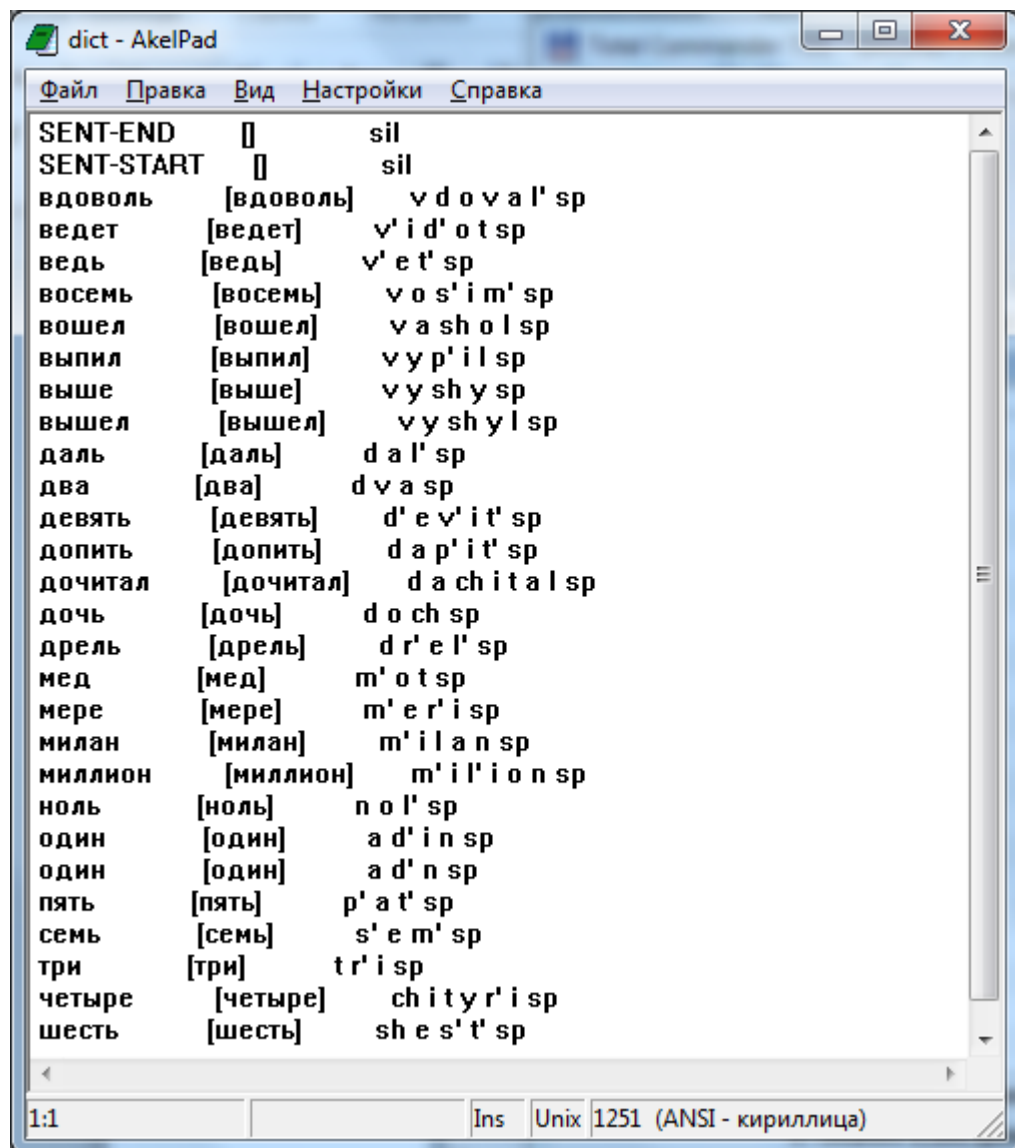


Рисунок 24.5. Содержимое файла dict

Запись производится в соответствии с рекомендациями представленными в лабораторной работе по созданию системы распознавания изолированного слова, параметры аудио должны быть: Mono, 16 kHz, 16 bit.

Далее используйте следующую команду:

```
HSGen -l -n 10 cont_wdnetdict>testprompts
```

Данная команда создаст testprompts файл содержащий 10 предложений из слов которые включены в cont_gram. Команда -l создает нумерацию перед каждым предложением, команда -n 10 задает число предложений в создаст

testprompts равным 10. Откройте файл testprompts уберите все точки после цифр, и поставьте вначале каждой строки символы */ (рисунок 24.6).

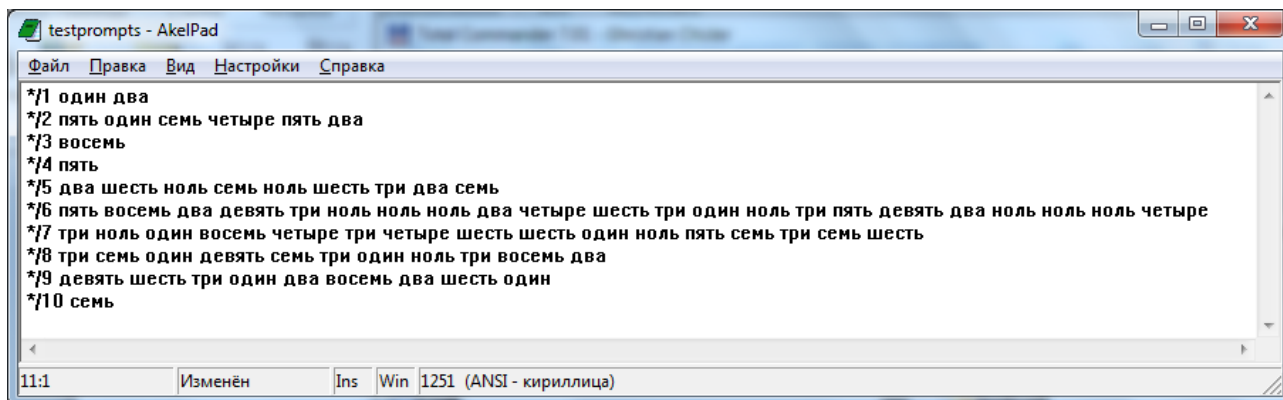


Рисунок 24.6.Пример testprompts файла

Создайте папку test(wav) и запишите в нее 10 аудиофайлов в соответствии с testprompts. Имена файлов должны быть 1.wav, 2.wav ... 10.wav. Запись производится с теми же параметрами и рекомендациями, как и в случае с обучающими данными. Эти 10 файлов будут использоваться для тестирования системы.

Шаг 4 – Создание файлов транскрипций

Чтобы обучить набор моделей СММ, каждый файл обучения должен иметь связанную с ним фонетическую транскрипцию (phone level transcription). Ввиду отсутствия вручную размеченных данных для начальной загрузки множества моделей, используем схему плоского старта (flat-start scheme). Для этой цели понадобится два набора фонетических транскрипций. Используемый вначале набор не будет содержать моделей короткой паузы (sp) между словами monophones0. Как только фонетические модели будут созданы, между словами будет вставлена модель sp, чтобы учесть любые паузы, которые может делать диктор monophones1.

Отправной точкой для обоих наборов фонетических транскрипций является орфографическая транскрипция в формате разметки НТК (НТК

label format). Это можно сделать довольно просто, используя текстовый редактор или язык сценариев.

Создадим Master Label File (MLF) используя скрипт prompts2mlf:

```
Perl prompts2mlf words.mlf prompts
```

Создадим аналогичный файл для тестовых данных:

```
Perl prompts2mlf test_words.mlf testprompts
```

В результате мы получим два файла words.mlf для обучающей выборки и test_words.mlf для проверочной выборки.

MLF-файлы должны выглядеть следующим образом:

```
#!MLF!#
"/S01.lab"
восемь
восемь
пять
ноль
четыре
миллион
дочь
мере
дочь
.
"/S02.lab"
вышел
мед
вошел
дрель
дрель
.
"/S03.lab"
даль
.
И т.д.
```

Как можно заметить, метки сценария должны быть преобразованы в названия путей, каждое слово должно быть записано в отдельной строке, и каждое высказывание должно завершаться единственной точкой. Первая строка файла идентифицирует его как главный файл меток (Master Label File (MLF)). Это отдельный файл, содержащий полный набор транскрипций. В НТК разрешается каждую отдельную транскрипцию хранить в своем файле, однако более эффективно использовать MLF.

Используемая в MLF форма названия пути нуждается в некотором пояснении, поскольку в действительности это не имя, а шаблон (pattern). Когда НТК обрабатывает речевые файлы, он ожидает найти транскрипцию (или файл разметки (label file)) с тем же именем, но с другим расширением. Так, при обработке файла /root/data/S0001.wav НТК искал бы заголовочный файл с именем /root/data/S0001.lab. При использовании файла MLF, НТК ищет в этом файле шаблон, соответствующий требуемому названию файла разметки. Однако, поскольку звездочка соответствует любой последовательности символов, используемый в данном примере шаблон является на самом деле независимым путем. Это позволяет использовать одинаковые транскрипции с разными версиями речевых данных, записанных в разных местах.

Далее создаем файл mkphones0.led содержащий:

EX

IS sil sil

DE sp

(не забудьте включить пустую строку в конце)

и файл mkphones1.led содержащий:

EX

IS sil sil

(не забудьте включить пустую строку в конце)

Команды в данных сценариях редактирования служат для настройки работы функции HLEd. Команда EX заменяет каждое слово в words.mlf

соответствующей фонетической транскрипцией из словаря dict. Команда IS вставляет модель паузы sil в начале и конце каждого высказывания. И, наконец, команда удаления DE удаляет все ненужные метки короткой паузы sr.

После того как создан MLF на уровне слов, можно генерировать MLF на уровне фонем, используя редактор разметки HLEd. Вышеупомянутый MLF на уровне слов был сохранен в файле words.mlf, командой

```
HLEd -l * -d dict -i phones0.mlf mkphones0.led words.mlf
```

будет сгенерирована транскрипция phones0.mlf на уровне фонем в следующем виде, где опция -l нужна, чтобы сгенерировать путь * в результирующих шаблонах(рисунок 24.7).

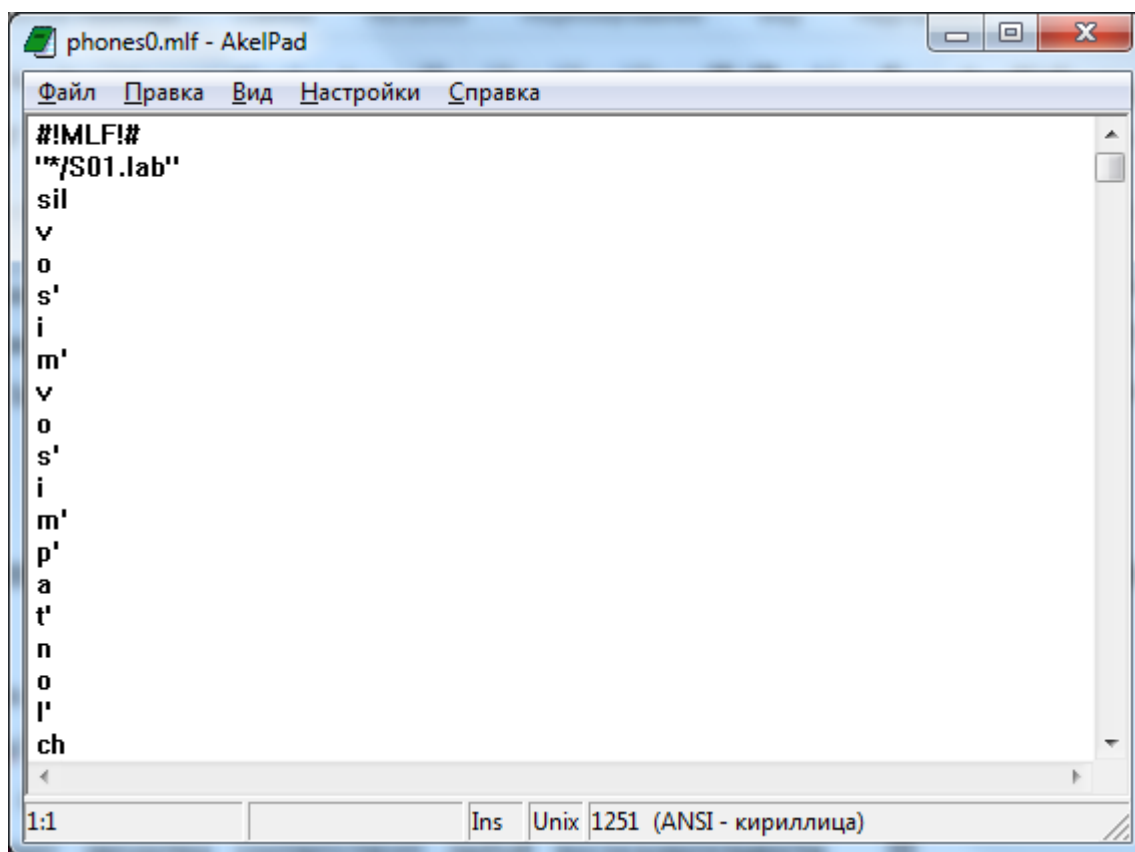


Рисунок 24.7. Содержимое файла phones0.mlf (без sr между словами)

Далее выполняем команду:

```
HLEd -l * -ddict -i phones1.mlf mkphones1.led words.mlf,
```

которая создает файл phones1.mlf (рисунок 24.8).

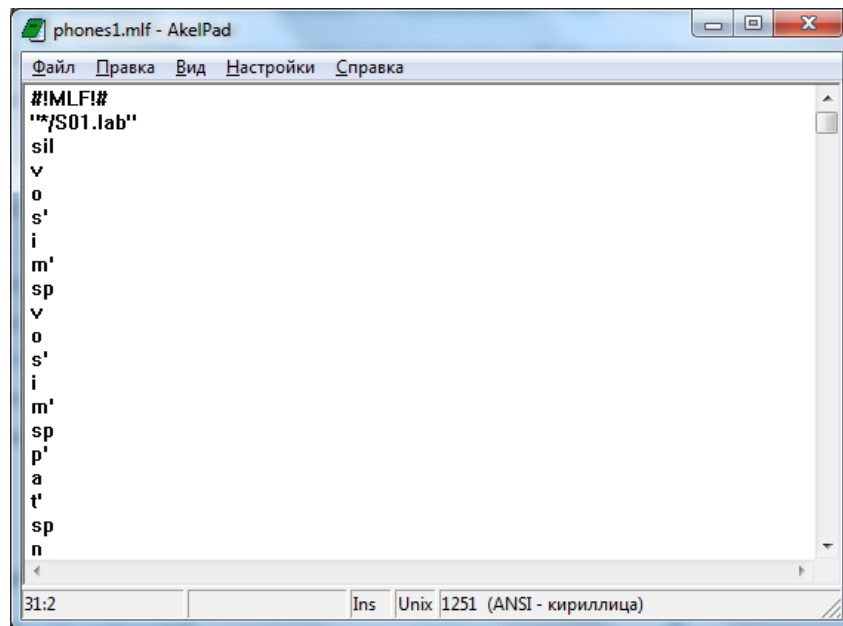


Рисунок 24.8. Содержимое файла phones1.mlf (csp между словами)

Шаг 5 – Кодирование данных

Создайте папку train(mfcc) и test(mfcc). Создайте файл codetrain.scp и пропишите в нем путь до данных обучающей выборки в папке train(wav) и путь до только что созданной папки train(mfcc) так же как было сделано в лабораторной работе №23 по созданию системы распознавания изолированного слова (рисунок 23.7). Пример codetrain.scp:

```
C:/Ivanov/ContSpeech/train(wav)/S01.wav C:/Ivanov/ContSpeech/train(mfcc)/S01.mfcc
C:/Ivanov/ContSpeech/train(wav)/S02.wav C:/Ivanov/ContSpeech/train(mfcc)/S02.mfcc
C:/Ivanov/ContSpeech/train(wav)/S03.wav C:/Ivanov/ContSpeech/train(mfcc)/S03.mfcc
C:/Ivanov/ContSpeech/train(wav)/S04.wav C:/Ivanov/ContSpeech/train(mfcc)/S04.mfcc
C:/Ivanov/ContSpeech/train(wav)/S05.wav C:/Ivanov/ContSpeech/train(mfcc)/S05.mfcc
...
C:/Ivanov/ContSpeech/train(wav)/S30.wav C:/Ivanov/ContSpeech/train(mfcc)/S30.mfcc
```

Аналогично создайте файл codetest.scp, который будет содержать пути для папки test(wav) и test(mfcc).

Далее используйте команду:

```
HCOPY -T 1 -C config_wav -S codetrain.scp
```

Для кодирования обучающих данных в папку train(mfcc) и

```
HCOPY -T 1 -C config_wav -S codetest.scp
```

Кодирующий проверочные данные в папку test(mfcc).

Создайте файлы `train.scr` и `test.scr` содержащие путь до файлов с расширением `*.mfcc` (правый столбец файлов `codetrain.scr` и `codetest.scr`).

Например, для файла `train.scr`:

```
C:/Ivanov/ContSpeech/train(mfcc)/S01.mfcc
C:/Ivanov/ContSpeech/train(mfcc)/S02.mfcc
C:/Ivanov/ContSpeech/train(mfcc)/S03.mfcc
...
C:/Ivanov/ContSpeech/train(mfcc)/S30.mfcc
```

Шаг 6 – Создание монофонов (плоский-старт (flat-start))

В этом разделе будет рассмотрено создание хорошо обученного набора единично-гауссовых (single-Gaussian) монофонов СММ. Отправной точкой является набор идентичных монофонов СММ, у которых все средние и дисперсии равны.

Первым шагом в обучении НММ является определение модели-прототипа. Параметры этой модели не важны, она предназначена для определения топологии модели. Для систем, основанных на звуках (phone-based systems), хорошей является топология с тремя состояниями слева-направо без пропусков. В качестве прототипа берем файл `proto` из лабораторной работы №23 по созданию систем распознавания изолированного слова (рисунок 23.8).

Создаем папку `hmm0` и используем функцию `HCompV`:

```
HCompV -C config -f 0.01 -m -S train.scr -M hmm0 proto
```

Далее создаем `hmmdefs` и `macros` файлы в папке `hmm0` и производим все те же операции что указаны в шаге 6 лабораторной работы №23 по созданию системы распознавания изолированного слова, с тем лишь различием что в файле `hmmdefs` слово `proto` в строке `~h "proto"` заменяется на фонемы из `monophones0` (рисунок 24.9).

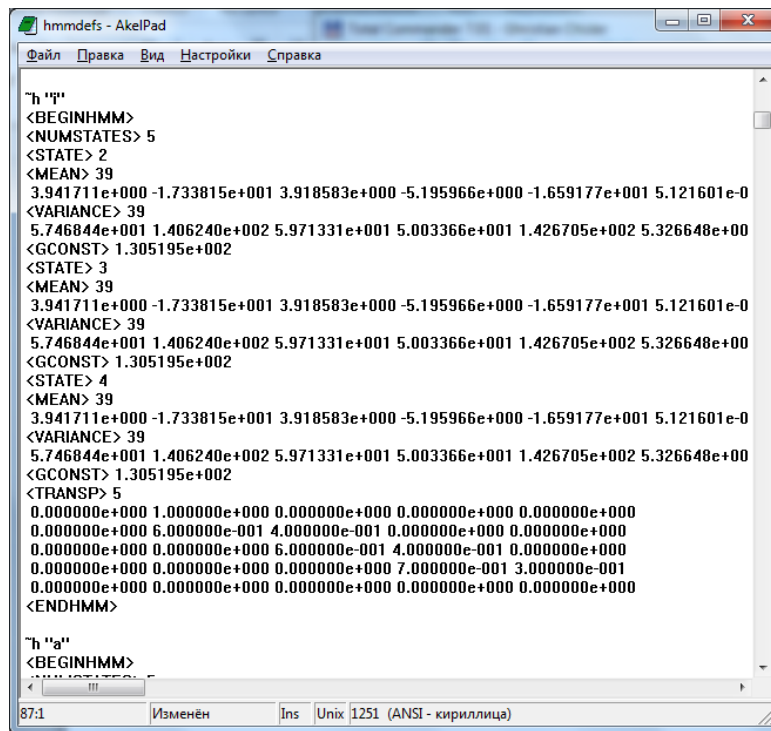


Рисунок 24.9. Содержание файла `hmmdefs`

Создаем папки `hmm1`, `hmm2`, `hmm3`, ... `hmm15`.

Монофоны, полученные с помощью плоского старта (`theflatstartmonophones`) и хранящиеся в директории `hmm0`, переоцениваются с помощью встроенного инструмента для переоценки `HEREST`, вызываемого следующим образом:

```
HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0
-S train.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1
monophones0
```

В результате все модели, которые перечислены в списке моделей `monophones0`, должны загрузиться в `hmm0` (`monophones0` – это модель с более короткими паузами (`sp`)). Затем, используя данные, перечисленные в файле `train.scp`, набор моделей переоценивается, и новый переоцененный набор моделей сохраняется в директории `hmm1`. При каждом запуске `HEREST` осуществляется единственная переоценка. Каждый новый набор НММ запоминается в новой директории.

Аналогично пересчитываем СММ и сохраняем в `hmm2`:

```
HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0
```

```

-S train.scp -H hmm1/macros -H hmm1/hmmdefs -M hmm2
monophones0
Иvhmm3
HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0
-S train.scp -H hmm2/macros -H hmm2/hmmdefs -M hmm3
monophones0

```

Опция `-t` устанавливает пороги отсечения, используемые при обучении. Отсечение ограничивает диапазон регулировки состояния и содержится в процедуре суммирования алгоритма прямого-обратного хода, что позволяет на порядок сократить объем вычислений. Для большинства обучающих файлов можно установить очень близкие отсекающие границы, однако некоторые обучающие файлы акустически плохо согласованы, так что требуется увеличение ширины зоны усечения. Эту задачу решает HEREST, автоматически расширяя границы усечения. В вышеприведенном примере обычная ширина зоны усечения равна 250.0. Если переоценка не удастся для любого из файлов, граница увеличится на 150.0 и файл будет повторно обработан. Это повторяется до тех пор, пока каждый из файлов не будет успешно обработан или не будет превышен пороговый лимит в 1000.0. На данном этапе можно безопасно допустить, что имеется проблема с обучающим файлом и, следовательно, ошибку необходимо устранить (обычно это бывает при неправильной транскрипции), или же обучающий файл должен быть отбракован.

Шаг 7 – создание СММ скороткими паузами между словами

На предыдущем шаге генерировалась СММ с тремя состояниями, слева направо, для каждого звука, а так же СММ для модели тишины `sil`. На следующем шаге добавляются дополнительные переходы из состояния 2 в 4 и из состояния 4 в 2 в модели тишины. Идея заключается в создании более точной модели путем разрешения отдельным состояниям поглощать разнообразные случайные помехи в обучающих данных. Использование

обратного скачка позволяет это сделать без привязки модели к переходу к следующему слову.

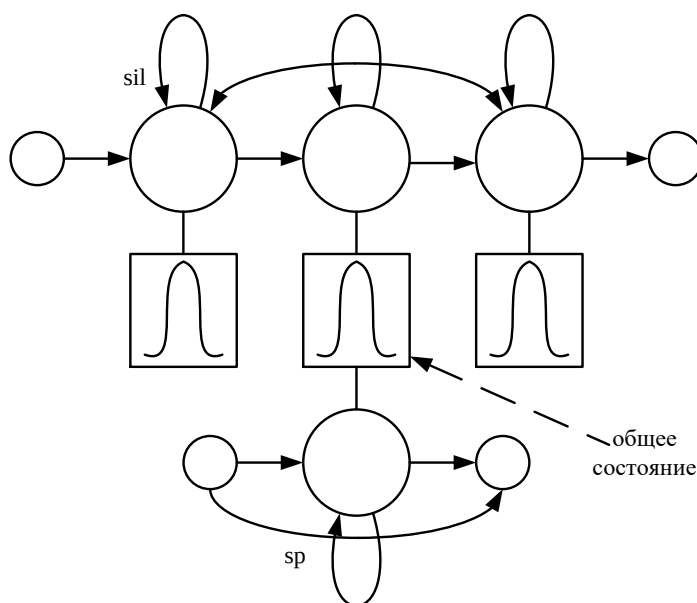


Рисунок 24.10. Модель тишины

Кроме того, на данном этапе должна быть создана модель короткой паузы *sp*, состоящая из 1 состояния. Это должна быть так называемая T-модель (*tee-model*), которая имеет прямой переход от начального до конечного узла. Эта *sp* содержит порождающее состояние, связанное с центральным состоянием модели тишины. Такая топология для двух моделей тишины показана на рисунке 24.10.

Эти модели тишины могут быть созданы в два этапа:

Копируем файлы *hmmdefs* и *macros* в папку *hmm4*, открываем файл *hmm4/hmmdefs* копируем и вставляем модель *~h "sil"* и переименовываем её в *~h "sp"*. Удаляем состояния 2 и 4 (*state 2*, *state 4*) в модели *sp*, меняем *<NUMSTATES>* на 3, *<STATE>* на 2, *<TRANSP>* на 3, меняем матрицу в конце модели на:

```
0.0 1.0 0.0
0.0 0.9 0.1
0.0 0.0 0.0
```

*Sp*модель должна выглядеть как показано на рисунке 24.11.

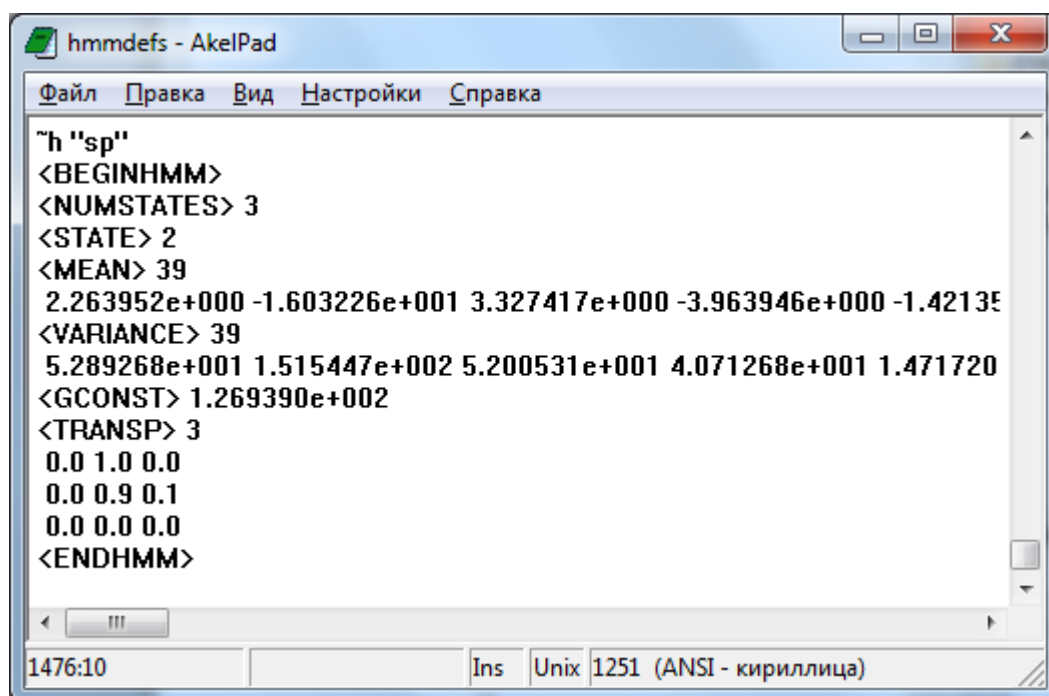


Рисунок 24.11. SpCMM

Создаем файл sil.hed содержащий следующие команды:

AT 2 4 0.2 {sil.transp}

AT 4 2 0.2 {sil.transp}

AT 1 1 0.1 {sp.transp}

TI slist {sil.state[3], sp.state[2]}

Команды AT добавляют переходы к данным матрицам переходов, а последняя команда TI создает состояние-связь (tied-state), называемое silst. Параметры этого состояния-связи состояния записаны в файле hmmdefs и, в пределах каждой модели тишины, первоначальные параметры состояния заменяются именем этого макроса. Ниже макросы описаны детально. Сейчас же достаточно их рассматривать как механизм, с помощью которого НТК осуществляет совместное использование параметра (parameter sharing). Заметим, что используемый здесь список звуков изменен, потому что первоначальный список monophones0 расширен новыми моделями sp. Далее будет использоваться файл monophones1 в команде NHED.

Запускаем редактор NHED, чтобы добавить необходимые дополнительные переходы и привязать состояние sp к центру состояния sil.

HNED работает аналогично HLED. Он применяет набор команд скрипта, чтобы модифицировать набор моделей HMM. В данном случае это выполняется следующим образом:

```
HNEd -Hhmm4/macros -Hhmm4/hmmdefs -Mhmm5 sil.hedmonophones1
```

Пересчитываем полученную модель 2 раза при помощи функции HERest:

```
HERest -A -D -T 1 -C config -I phones1.mlf -t 250.0 150.0  
3000.0 -S train.scp -H hmm5/macros -H hmm5/hmmdefs -M hmm6  
monophones1
```

Создаются файлы в папке `hmm6` `hmmdefs` и `macros`

```
HERest -A -D -T 1 -C config -I phones1.mlf -t 250.0 150.0  
3000.0 -S train.scp -H hmm6/macros -H hmm6/hmmdefs -M hmm7  
monophones1
```

Создаются файлы в папке `hmm7` `hmmdefs` и `macros`

В результате мы получили модель основанную на монофонах с моделью тишины между словами.

Шаг 8 – Преобразование (realigning) обучающих данных

В словаре содержатся разные произношения некоторых слов, в особенности функциональных слов (function words). Созданные к настоящему моменту времени модели звуков могут быть использованы для преобразования (realigning) обучающих данных и создания новых транскрипций. Это можно сделать путем вызова инструмента распознавания НТК по имени HVITE:

```
HVite -l * -o SWT -b SENT-END -C Conf -C config -a -H  
hmm7/macros-H hmm7/hmmdefs -i aligned.mlf -m -t 250.0 -y lab  
-I word.mlf -S train.scp dict monophones1>HVite_log
```

По этой команде хранящиеся в `hmm7` модели СММ используются для преобразования входных данных в виде транскрипций на уровне слова `words.mlf` в новые транскрипции на уровне звука `aligned.mlf`, с использованием произношений из словаря `dict`. Основным отличием этой

процедуры от первичного преобразования слово-звук, осуществляемого HLED на 4-м шаге, состоит в том, что система распознавания рассматривает все произношения для каждого слова, и в качестве выходных данных формирует произношения, наилучшим образом соответствующие акустическим данным.

Примененная выше опция `-b` использована для вставки модели тишины в начале и в конце каждого высказывания. Имя `SENT-END` использовано в предположении, что словарь содержит компонент `SENT-END sil`.

Заметим, что словарь должен быть отсортирован, во-первых, по регистру букв (заглавные буквы идут вначале), а во-вторых, по алфавиту. Опция `-t` устанавливает уровень отсечения `250.0`, а опция `-o` использована для подавления печати меток, имен слов и временных границ в выходных данных MLF.

Файл `HVite_log` содержит информацию о проведении преобразования данных, убедитесь что функция `HVite` распознала все слова каждой строки вашего `prompts` файла.

По окончании создания новых звуков, для переоценки набора параметров СММ необходимо осуществить еще два прохода инструментом `HEREST`.

```
HERest -A -D -T 1 -C config -I aligned.mlf -t 250.0 150.0 3000.0  
-S train.scp -H hmm7/macros -H hmm7/hmmdefs -M hmm8 monophones1
```

Создает файлы `hmmdefs` и `macros` в папке `hmm8`

```
HERest -A -D -T 1 -C config -I aligned.mlf -t 250.0 150.0 3000.0  
-S train.scp -H hmm8/macros -H hmm8/hmmdefs -M hmm9 monophones1
```

Если все это проделано, в директории `hmm9` будет сохранен конечный набор монофонов СММ.

СММ в папке `hmm9` модель которую можно запускать для распознавания голоса с микрофона командой `HVite`.

```
HVite -H hmm9/macros -H hmm9/hmmdefs -C Conf -C config2 -w  
cont_wdnet -p 0.0 -s 5.0 dict monophones1
```

При старте HVITE пригласит пользователя произнести произвольное предложение (приблизительно 4 сек), необходимое для измерения уровней фоновой тишины и речи.

В случае проблем отображение кириллицы в cmd замените кириллицу на латиницу в словаре dict по аналогии с шагом 7 лабораторной работы по построению системы распознавания изолированного слова.

Шаг 9 – Оценка системы распознавания

Теперь система распознавания готова и ее эффективность может быть оценена. Распознающая сеть и словарь уже сконструированы, данные для тестирования записаны. Поэтому, остается лишь запустить систему распознавания и затем оценить результаты, используя НТК инструментарий HRESULTS.

Файл test.scp содержит список кодированных файлов, тогда каждый тестовый файл будет распознан и результат в виде транскрипции будет занесен в MLF по имени recount.mlf, если выполнить следующую команду

```
HVite -C Conf -H hmm9/macros -H hmm9/hmmdefs -S test.scp -I *  
-I recount.mlf -w cont_wdnet -p 0.0 -s 5.0 dict monophones1
```

Где опции `-p` и `-s` устанавливают штраф вставки слова (word insertion penalty) и множитель шкалы грамматики (grammar scale factor), соответственно. Штраф вставки слова – это постоянная величина, добавляющаяся к каждому маркеру при его перемещении от конца одного слова до начала другого. Множитель шкалы грамматики – это величина, с помощью которой масштабируется вероятностная модель языка перед тем, как она будет добавлена к каждому маркеру при его перемещении от конца одного слова до начала другого. Эти параметры могут оказывать значительное влияние на эффективность распознавания, поэтому заслуживают особого внимания процесс некоторой регулировки при подготовке обучающих данных.

Словарь содержит транскрипции монофонов. HVITE выполнит все необходимые преобразования при загрузке словесной сети wdnet.

Файл MLF test_words.mlf содержит транскрипции на уровне слов для каждого тестового файла, тогда действительную производительность можно определить, запустив инструмент HRESULTS следующим образом:

```
HResults -C Conf -I testref.mlf monophones1 recout.mlf
===== HTK Results Analysis=====
Date : Sun may 22 16:14:45 2013
Ref  : test_words.mlf
Rec  : recount.mlf
----- Overall Results -----
SENT: %Correct=90.00 [H=9, S=1, N=10]
WORD: %Corr=99.77, Acc=99.65 [H=85, D=1, S=1, I=1, N=87]
=====
```

Строка, начинающаяся с SENT: указывает на то, что из 10 тестовых высказываний, 9 (90.00%) были правильно распознаны. Следующая строка, начинающаяся словом WORD:, представляет статистику на уровне слов и показывает, что всего было 87 слов, из них верно распознано 85 (99.70%). Была одна ошибка удаления (D), 1 ошибка замены (S) и 1 ошибка вставки (I). Коэффициент точности (Acc) 99.65% ниже процента правильности (Cor), поскольку в нем учитываются ошибки вставки, которые в последнем не учитываются.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;

- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

- 1. Каким образом определяются границы, разделяющие сегменты речи?
- 2. Опишите кратко процесс обучения модели распознавания слитной речи.
- 3. Какие алгоритмы применяются при обучении и распознавании слитной речи?

ЛИТЕРАТУРА И ИСТОЧНИКИ

Основная литература:

1. Шагрова, Г.В. Методы исследования и моделирования информационных процессов и технологий : учебное пособие / Г.В. Шагрова, И.Н. Топчиев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь : СКФУ, 2016. - 180 с. : ил. - Библиогр.: с. 178 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=458289> (13.05.2019).
2. Шкундин, С.З. Теория информационных процессов и систем : учебное пособие / С.З. Шкундин, В.Ш. Берикашвили. - М. : Горная книга, 2012. - 475 с. - ISBN 978-5-98672-285-6 [Электронный ресурс].

URL: <http://biblioclub.ru/index.php?page=book&id=229031>

Дополнительная литература:

1. Буканова, Т. С. Моделирование систем управления: учебное пособие / Т.С. Буканова, М.Т. Алиев ; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 144 с. : ил., граф. - <http://biblioclub.ru/>. - Библиогр. в кн. - ISBN 978-5-8158-1899-6, экземпляров неограниченно. Режим доступа: https://biblioclub.ru/index.php?page=book_red&id=483694&razdel=259
2. Методы исследования и моделирования информационных процессов и технологий : лабораторный практикум / авт.-сост. Г.В. Шагрова, М.Г. Романенко, И.Н. Топчиев ; Министерство образования и науки Российской Федерации и др. - Ставрополь : СКФУ, 2016. - 241 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=458081>

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению курсовой работы по дисциплине
«Методы исследования и моделирования информационных процессов и
технологий»
для студентов направления подготовки 09.04.02 «Информационные системы
и технологии»
Направленность (профиль) «Управление данными»
Квалификация выпускника: магистр

Ставрополь, 2025

Методические указания к выполнению курсовой работы по дисциплине «Методы исследования и моделирования информационных процессов и технологий» составлены в соответствии с программой дисциплины для студентов направления подготовки 09.04.02 «Информационные системы и технологии», программа магистратуры «Управление данными».

Они предназначены для организации самостоятельной работы при выполнении курсовой работы и содержат краткие сведения необходимые для выполнения задания.

СОДЕРЖАНИЕ

1. Введение.....	4
2. Цель, задачи и реализуемые компетенции	5
3. Тематика курсовой работы. Формулировка задания.....	8
4. Содержание и структура курсовой работы	12
5. Общие требования к написанию курсовой работы	13
6. Рекомендации по организации выполнения курсовой работы, примерный календарный план ее выполнения	15
7. Критерии оценивания работы	16
8. Организация и руководство курсовыми работами	18
9. Порядок защиты работы	20
10.Список рекомендуемой литературы	22
10.1 Основная литература	22
10.2 Дополнительная литература	22

1. Введение

Методические указания к выполнению курсовой работы составлены в соответствии с программой дисциплины «Методы исследования и моделирования информационных процессов и технологий» для студентов направления подготовки 09.04.02 «Информационные системы и технологии» и Положением об организации и выполнении курсовых работ (проектов) в федеральном государственном автономном образовательном учреждении высшего профессионального образования «Северо-Кавказский федеральный университет» и определяют требования к выполнению и защите курсовых работ (проектов) в ФГАОУ ВПО «СКФУ» (далее Положение).

Положение разработано в соответствии с нормативно-правовыми документами:

- действующим законодательством Российской Федерации в области образования, нормативными документами Министерства образования и науки РФ;
- Типовым положением об образовательном учреждении высшего профессионального образования (вышем учебном заведении), утвержденным постановлением Правительства Российской Федерации;
- Государственными образовательными стандартами высшего профессионального образования и Федеральными государственными образовательными стандартами высшего профессионального образования;
- образовательными стандартами, устанавливаемыми СКФУ;
- Уставом федерального государственного автономного образовательного учреждения высшего профессионального образования «Северо-Кавказский федеральный университет»;
- другими нормативными и правовыми актами в области профессионального образования;
- решениями Ученого совета СКФУ.

Требования Положения являются обязательными для всех сотрудников Университета, участвующих в организации выполнения курсовых работ, и для студентов всех форм обучения.

2. Цель, задачи и реализуемые компетенции

Программой дисциплины «Методы исследования и моделирования информационных процессов и технологий» предусмотрено выполнение курсовой работы, которая является обязательным видом учебной работы, выполняется студентом в течение семестра

Курсовая работа это самостоятельная исследовательская работа студента, она представляет собой логически завершенное и оформленное научное исследование.

Целью выполнения курсовой работы является:

- формирование у студента навыков научно-исследовательской работы;
- повышение уровня его профессиональной (теоретической и практической) подготовки путем углубления и закрепления теоретических знаний, а также практических навыков, приобретенных в результате изучения дисциплины «Методы исследования и моделирования информационных процессов и технологий»;
- развитие интереса и навыков самостоятельной работы с научной и справочной литературой;
- подготовка к самостоятельному решению инженерных задач;
- подготовка к выполнению магистерской диссертации.

В процессе выполнения курсовой работы необходимо:

1. Приобрести новые теоретические знания в соответствии с темой работы и заданием руководителя.
2. Систематизировать, обобщить и логично изложить концепции, альтернативные точки зрения по исследуемой проблеме.

3. Развить учебно-исследовательские и методические навыки, необходимые для системного научного анализа изучаемого объекта, процесса, явления.

Процесс изучения дисциплины направлен на формирование у студентов следующих:

Код	Формулировка
УК-6	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки
ОПК-4	Способен применять на практике новые научные принципы и методы исследований;
ОПК-5	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;
ПК-1	Способен осуществлять математическое моделирование и исследование информационных процессов, систем и технологий, объектов, и устройств вычислительной техники на базе современных пакетов компьютерного моделирования
ПК-2	Способен разрабатывать и исследовать теоретические и экспериментальные модели объектов профессиональной деятельности в различных областях
ПК-3	Способен планировать и проводить эксперименты, обрабатывать и анализировать результаты с помощью современных информационных технологий

В результате освоения дисциплины обучающийся получает знания: методов анализа и синтеза информационных систем; формальных моделей систем; средств структурного анализа; методологии структурного системного анализа и проектирования; моделей предметных областей информационных систем; объектно-ориентированного подхода; моделей дискретных объектов и явлений реального и виртуальных миров; математических (аналитико-имитационные) моделей информационных процессов; методов анализа структур информационных систем.

Умения: разрабатывать аналитические и имитационные модели предметных областей; руководить процессом исследования и проектирования информационных систем; применять на практике методы и средства проектирования информационных систем; проводить исследования характеристик компонентов и информационных систем в целом.

Навыки владения методами анализа и синтеза информационных систем, методами разработки математических моделей информационных систем, методами проектирования информационных систем.

При выполнении работы студент должен продемонстрировать:

знания:

- методов системного анализа для описания предметной области объектов проектирования и поставленных задач;
- формального аппарата для анализа организационной, функциональной, информационной, алгоритмической и технической структур информационных процессов и систем;
- методов системного моделирования информационных систем.

умения:

- формировать основные технико-экономические требования к информационным процессам, системам и технологиям;
- формировать и решать задачи применительно к информационным процессам и системам, к их компонентам с использованием информационных технологий;
- количественно оценивать производительность и информационную надежность информационных процессов и систем;
- проектировать информационные системы для сферы высшего образования;
- разрабатывать структуры информационно-логических, функциональных, алгоритмических моделей информационных систем и их компонентов.

3. Тематика курсовой работы. Формулировка задания

Тематика курсовой работы должна быть актуальной и отвечать как учебным задачам дисциплины, так и потребностям науки и практики. Актуальность тематики курсовых работ обусловлена научностью, современностью и направленностью к получению студентами навыков самостоятельной творческой работы.

Темы курсовых работ являются комплексными. Они содержат ряд взаимосвязанных между собой проблем науки и практики, опираясь:

- на фактический материал научных исследований, выполняемых в образовательных учреждениях;
- на использование новейших достижений отечественной и зарубежной науки;
- на актуальные прикладные аспекты образовательной деятельности;
- на итоги учебной и производственной практики студентов;
- на научные работы членов кафедры, студенческих научных кружков и проблемных групп.

Курсовая работа является составной частью выпускной квалификационной работы и ориентирована на практическое использование знаний и умений по дисциплине с применением вычислительной техники. В зависимости от вида объекта исследований и предмета разработки при выполнении курсовой работы используются методы системного анализа, методы исследования и моделирования информационных процессов, систем и технологий.

В курсовой работе должны быть отражены следующие аспекты:

- экономические;
- этические;
- социально-политические;
- экологические;

– безопасности труда, а также вопросы устойчивого развития, касающиеся предмета исследования.

Темы курсовых работ утверждаются на заседании кафедры информационных систем и технологий в течение двух недель после начала семестра. Выписка из протокола заседания кафедры передается в дирекцию института, где формируется распоряжение об утверждении тем курсовых работ. В течение 10 дней после выхода распоряжения об утверждении тем курсовых работ названия тем вносятся в автоматизированную базу университета.

Примерный перечень тем курсовой работы:

1. Технология постановки, формализации и решения задач обработки и анализа изображений.
2. Технология исследования многомерных нестационарных явлений с помощью вейвлетов в пакете Wavelet Toolbox.
3. Методы исследования и оптимизации мультимедиа ресурсов дистанционного образования.
4. Методы исследования и моделирования, реализованные в пакете Image Processing Toolbox для задач обработки изображений.
5. Методы исследования и моделирования, реализованные в пакете ComputerVisionSystemToolbox для задач обработки видеоизображений.
6. Методы исследования и моделирования многомерных нестационарных явлений, реализованные в пакете WaveletToolbox.
7. Методы спектрального анализа и синтеза сигналов средствами пакета Signal Processing Toolbox.
8. Моделирование систем автоматического управления средствами пакета Control System Toolbox.
9. Моделирование систем управления с помощью пакета Nonlinear Control Desing (NCD) Blockset.
10. Математическое моделирование и исследование системы управления на базе нечёткой логики средствами SimInTech и Simulink

11. Исследование нелинейных систем автоматического регулирования с помощью методов анализа устойчивости средствами SimInTech и Simulink
12. Исследование и оптимизация параметров модели системы автоматического регулирования средствами SimInTech и Simulink
13. Математическое моделирование нелинейной системы автоматического регулирования ядерного реактора средствами SimInTech и Simulink
14. Анализ динамических систем с запаздыванием в среде динамического моделирования SimInTech и пакете Simulink
15. Исследование известных динамических задач методами структурного моделирования средствами SimInTech и Simulink
16. Анализ динамических систем, заданных нелинейными дифференциальными уравнениями средствами SimInTech и Simulink
17. Моделирование и исследование динамических процессов в управляемых технических системах средствами SimInTech и Simulink
18. Анализ динамики линейных и линеаризованных, описываемых в переменных «Вход – Выход» средствами SimInTech и Simulink
19. Разработка системы компьютерного моделирования динамики магнитожидкостных капель средствами Simulink и SimInTech
20. Компьютерное моделирование динамики магнитных капель в среде динамического моделирования SimInTech и пакете Simulink
21. Исследование технологии решения обыкновенных дифференциальных и дифференциально-алгебраических уравнений средствами SimInTech и Simulink
22. Моделирование цифровой системы управления опорно-поворотным устройством в среде динамического моделирования SimInTech и пакете Simulink

23. Исследование влияния типовых законов управления на систему стабилизации в среде динамического моделирования SimInTech и пакете Simulink

24. Исследование контура стабилизации спутника дистанционного зондирования в среде динамического моделирования SimInTech и пакете Simulink

25. Технология исследования цифровых фильтров средствами Simulink и SimInTech

26. Методы исследования и моделирования объектов профессиональной деятельности (процессов, систем, технологий, программного и аппаратного обеспечения информационных и автоматизированных систем) в соответствии с выбранной предметной областью

27. Методы исследования и моделирование информационных процессов, систем и технологий в соответствии с темой выпускной квалификационной работы (формулируется в окончательном виде в зависимости от темы выпускной квалификационной работы).

Основные содержательные и процессуальные аспекты, необходимые для выполнения работы, оформляются в заданиях по курсовой работе в соответствии с приложением 1.

В задании на курсовую работу формулируется инженерная проблема, требующая решения известным методом или проведения исследования и выбора метода для своего решения в соответствии с Государственным образовательным стандартом и рабочей программой дисциплины. Тематика курсовой работы может охватывать проблемы разработки образовательных ресурсов для сферы высшего образования, аппаратные, программно-аппаратные и программные проблемы, связанные с работой или применением ЭВМ в образовательных, организационных и технических системах.

В задании формулируется название темы работы и требования, определяющие его объем, содержание, а также исходные данные для выполнения количественных расчетов.

4. Содержание и структура курсовой работы

Курсовая работа должна содержать элементы новизны, наряду с фундаментальным аспектом должен быть проведен анализ современного состояния изучаемой проблемы, а также отражены региональные особенности. Задание по курсовой работе формулируется индивидуально с учетом интересов и способностей студента.

Курсовая работа состоит из введения, теоретической части, практической части, заключения, списка литературы и приложения.

Титульный лист оформляется в соответствии с приложением 2.

Во введении обосновывается актуальность выбранной темы исследования, отражаются объект, предмет, задачи, цели, методы, новизна, теоретическая и практическая значимость исследования.

Теоретическая часть должна содержать анализ состояния изучаемой проблемы на основе обзора научной, научно-информационной, учебно-методической, справочной литературы. Представленный материал должен быть логически связан с целью исследования. В параграфах теоретической части необходимо отражать отдельные компоненты проблемы и завершать их выводами.

Практическая часть включает обоснование методов исследования и моделирования информационных процессов и технологий в соответствии с темой курсовой работы, обоснование выбора инструментальных средств разработки, проектирование и разработку приложения, анализ результатов экспериментального исследования, схемы, графические и математические способы интерпретации полученных данных, выводы.

Закключение содержит выводы, подтверждающие или опровергающие первоначальные предположения (гипотезы), перспективы дальнейшего изучения проблемы, связь с практикой, анализ реализации целей и задач исследования.

Список литературы должен быть составлен в соответствии с требованиями ГОСТ к оформлению библиографии.

Приложение содержит весь фактический материал практической части исследований (схемы, чертежи, расчетные материалы, рисунки и т.д.).

5. Общие требования к написанию курсовой работы

Курсовая работа должна оформляться в соответствии с требованиями государственных стандартов по инженерным специальностям и направлениям подготовки:

- ГОСТ 2.105-95. Общие требования к текстовым документам, ГОСТ 2.106-96. Текстовые документы;
- ГОСТ 2.104-68. Основные надписи; Единая система конструкторской документации; Единая система технологической документации; Единая система программной документации;
- ГОСТ 7.32-2001. Отчет по научно-исследовательской работе. Структура и правила оформления;
- ГОСТ Р 7.0.11–2011. Система стандартов по информации, библиотечному и издательскому делу. Диссертация и автореферат диссертации. Структура и правила оформления.

Содержание курсовой работы рекомендуется представлять в объеме 1-2 печатных листа. Текст работы должен быть напечатан через 1,5 интервала на одной стороне стандартного листа белой бумаги (А-4). Текст и другие отпечатанные элементы работы должны быть черными, контуры букв и знаков – четкими, без ореола и затенения. Шрифт TimesNewRoman, кегель 14. Курсив и подчеркивание в работе не допускаются. Названия глав и

параграфов выделяются полужирным шрифтом. Лист с текстом должен иметь поля:

- слева – 30 мм;
- справа – 10 мм;
- сверху – 20 мм;
- снизу – 20 мм.

Нумерация страниц текста делается в правом нижнем углу листа. Проставлять номер страницы необходимо со страницы, где печатается «Введение», на которой ставится цифра «3». После этого нумеруются все страницы, включая приложения.

Список литературы должен быть составлен в соответствии с требованиями ГОСТа к оформлению библиографии:

- ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание;
- ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов;
- ГОСТ Р 7.0.5-2008. Библиографическая ссылка. Общие требования и правила составления;
- ГОСТ 7.80-2000. Библиографическая запись. Заголовок. Общие требования и правила составления.

Между названием главы и названием параграфа этой главы ставится пробел равный двум интервалам, а название параграфа не должно отделяться от текста этого параграфа пробелом. Названия параграфов отделяются от текста предыдущего параграфа пробелом, равным двум интервалам. Каждая глава, а также введение, выводы, предложения и список использованной литературы начинаются с новой страницы. Слово «Глава» не пишется. Главы имеют порядковые номера в пределах всей работы, обозначаемые арабскими цифрами (например: 1,2,3), после которых ставится точка. Слово «параграф» или значок параграфа в названии не ставятся. Параграфы имеют порядковые

номера в пределах глав, обозначаемые арабскими цифрами (например: 1.1. и 1.2.). Заголовки глав и параграфов в тексте работы должны располагаться по центру, точку в конце названия главы и параграфа не ставят. Не допускается переносить часть слова в заголовке.

Нумерация таблиц и рисунков может быть сквозной или соотноситься с номером главы и параграфа. Например, если таблица или рисунок включены в текст первого параграфа второй главы, нумерация следующая: Таблица 2.1.1., рис. 2.1.1. Последняя цифра означает порядковый номер таблицы (или рисунка) в данном параграфе. Таблица помещается в качестве следующей страницы после первого упоминания о ней в тексте.

Листы отчета по курсовой работе должны быть пронумерованы и скреплены.

6. Рекомендации по организации выполнения курсовой работы, примерный календарный план ее выполнения

Примерный календарный план выполнения курсовой работы представлен в таблице.6.1.

Таблица 6.1 – Календарный план

Наименование этапа работы	Срок выполнения
Выдача задания	2 неделя семестра
Выполнение курсовой работы: – подбор информации по теоретической части; – оформление теоретической части работы; – выполнение практической части; – оформление отчета по курсовой работе.	3 –12 недели семестра
Сдача работы на проверку: 1. Теоретической части; 2. Практической части.	6 неделя семестра 9 неделя семестра

Наименование этапа работы	Срок выполнения
Проверка курсовой работы	7, 10 недели семестра
Доработка (исправление) работы	8, 11 недели семестра
Проверка курсовой работы после доработки	12 неделя семестра
Рецензирование работы	13 неделя семестра
Защита курсовой работы	14 неделя семестра

Выполнение курсовой работы рекомендуется выполнять в соответствии с предложенным календарным планом.

7. Критерии оценивания работы

При оценке уровня выполнения курсовой работы, в соответствии с поставленными целями для данного вида учебной деятельности контролируются умения, навыки и компетенции, перечисленные в п. 2 «Цель, задачи и реализуемые компетенции».

В ходе защиты проводится широкое обсуждение работы, позволяющее оценить качество компетенций, сформированных у студента.

Критерии целостной оценки содержания работы представлены в таблице 7.1.

Таблица 7.1 – Критерии оценки содержания работы

№ п/п	Описание	Оценка
1.	Четкое, полное понимание исследуемой проблемы. Задание выполнено в полном объеме. Программный продукт позволяет решать все поставленные задачи в полном объеме.	5
2.	Значительное понимание проблемы. Все требования, предъявляемые к заданию, выполнены. Программный продукт позволяет решать поставленные задачи.	4
3.	Частичное понимание проблемы. Большинство требований, предъявляемых к заданию, выполнено. Программный продукт не позволяет решать все поставленные задачи.	3
4.	Непонимание проблемы.	2

	Многие требования, предъявляемые к заданию, не выполнены. Программный продукт не позволяет решать поставленные задачи.	

В таблице 7.2 представлены критерии оценки качества презентации, программного продукта и защиты работы.

Таблица 7.2 – Критерии оценки презентации, программного продукта и защиты работы

Дескрипторы	Минимальный ответ 2	Изложенный, раскрытый ответ 3	Законченный, полный ответ 4	Образцовый, примерный; достойный подражания ответ 5
Оформление	Презентация оформлена без учета требований к представлению информации. Нет иллюстративного материала поясняющего представляемый материал.	Презентация оформлена без учета требований к представлению информации. Иллюстративного материала не достаточно.	Презентация оформлена с учетом требований к представлению информации. Иллюстративного материала не достаточно.	Использованы для представления информации мультимедиа средства. Материал оформлен с учетом требований и достаточным количеством иллюстративного материала
Формулировка проблемы	Проблема не раскрыта. Отсутствуют выводы.	Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы.	Проблема раскрыта. Не все выводы сделаны и/или обоснованы.	Проблема раскрыта полностью. Выводы обоснованы.
Представление	Представляемая информация логически не связана, не систематизирована и/или не последовательна	Представляемая информация не систематизирована или не последовательна	Представляемая информация систематизирована и последовательна.	Представляемая информация систематизирована, последовательна и логически

	на			связана.
Дескрипторы	Минимальный ответ 2	Изложенный, раскрытый ответ 3	Законченный, полный ответ 4	Образцовый, примерный; достойный подражания ответ 5
Демонстрация программно-го продукта	Работа программы не продемонстрирована	Ошибки в работе программы	Работа программы продемонстрирована не в полном объеме	Работа программы продемонстрирована в полном объеме
Ответы на дополнительные вопросы	Нет ответов на вопросы.	Только ответы на элементарные вопросы.	Ответы на вопросы полные и/или частично полные.	Ответы на вопросы полные с приведением примеров и/или пояснений.
Отзыв на курсовую работу	Отзыв отрицательный	Отзыв положительный с существенными замечаниями	Отзыв положительный с не существенными замечаниями	Отзыв положительный

В соответствии с критериями, представленными в таблицах 7.1 и 7.2, комиссия оценивает результат выполнения работы, степень сформированности компетенций и выставляет итоговую оценку курсовой работы

8. Организация и руководство курсовыми работами

Руководство курсовыми работами поручается квалифицированным преподавателям кафедры, обладающим научно-методическим опытом.

Руководитель магистерской программы на специально организованном кафедрой занятии разъясняет значение выполнения работы для дисциплины, требования, предъявляемые к работе, предлагает выбрать тему в

соответствии темой выпускной квалификационной работы и с предлагаемым примерным перечнем. Окончательно тема формулируется руководителем курсовой работы в соответствии с темой магистерской диссертацией и целями и задачами дисциплины.

Руководство курсовыми работами начинается с выдачи задания студентам после индивидуальной беседы руководителя со студентом по заданию. В ходе беседы выясняется степень подготовленности студента к выполнению данного задания, рекомендуется необходимая литература и сообщается порядок выполнения задания. В результате индивидуальной беседы может быть уточнена или выбрана студентом другая тема работы.

Задание выдается за подписью руководителя работы, датируется днем выдачи, утверждается заведующим кафедрой и регистрируется в кафедральном журнале.

В рамках отведенного для руководства курсовыми работами времени регулярно проводятся индивидуальные консультации, раз в две недели по установленному графику и контролируются этапы выполнения работы. Руководитель, прежде всего, проводит экспертизу практической части, а затем указывает все ошибки, неточности по работе в целом. После проверки руководителем выполнения одного этапа работы студенту (в случае положительного заключения) разрешается перейти к следующему этапу.

Заведующая кафедрой периодически проверяет состояние работы, контролирует направленность и методику деятельности отдельных руководителей, давая на заседаниях кафедры соответствующие методические указания.

Работа перед сдачей руководителю подписывается студентом. Если работа удовлетворяет требованиям, предъявляемым к курсовым работам, она допускается к защите, о чем руководитель делает надпись на титульном листе.

9. Порядок защиты работы

Защита курсовой работы является обязательной формой проверки выполнения работы. Защита производится на заседании научной проблемной группы специальной комиссией, утверждаемой директором института, состоящей обычно из 3 преподавателей кафедры, при непосредственном участии руководителя, в присутствии студентов. Результаты наиболее интересных курсовых работ рекомендуются к докладу на научных конференциях. Публичная защита направлена на стимулирование научного интереса, творчества, ответственности студентов.

Защита состоит в коротком докладе студента по выполненной работе и в ответах на вопросы присутствующих на защите. Научный руководитель зачитывает отзыв на курсовую работу студента, оформленном в соответствии с приложением 3.

Результаты защиты курсовой работы, согласно действующему Положению о текущем контроле и промежуточной аттестации в СКФУ, оцениваются дифференцированной отметкой по пятибалльной системе. Оценка курсовой работы заносится в зачетную книжку студента и зачетно-экзаменационную ведомость, составляемую в 2-х экземплярах, один из которых хранится на кафедре в течение всего срока обучения студента, другой представляется в дирекцию института информационных систем и телекоммуникаций

Защита курсовых работ (проектов), предусмотренных учебным планом, проводится не позднее, чем за две недели до начала зачетно-экзаменационной сессии.

Студент, не представивший в установленный срок курсовую работу или не защитивший ее по неуважительной причине, считается имеющим академическую задолженность.

Курсовые работы, представляющие теоретический и практический интерес, представляются на конкурс в студенческие научные общества, конференции, отмечаются приказом по университету.

Выполненные работы после их защиты должны храниться на кафедре в течение 2 лет, не считая года написания; затем работы, не представляющие для кафедры интерес, уничтожаются по акту.

Итоги выполнения курсовых работ ежегодно обсуждаются на кафедре информационных систем и технологий, и по мере необходимости на Ученом совете института информационных технологий и телекоммуникаций, а в отдельных случаях и на заседаниях Ученого Совета университета.

10.Список рекомендуемой литературы

10.1 Основная литература

1. ПОЛОЖЕНИЕ О ВЫПОЛНЕНИИ И ЗАЩИТЕ КУРСОВЫХ РАБОТ (ПРОЕКТОВ) В ФЕДЕРАЛЬНОМ ГОСУДАРСТВЕННОМ АВТОНОМНОМ ОБРАЗОВАТЕЛЬНОМ УЧРЕЖДЕНИИ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
2. Советов, Б. Я. Теория информационных процессов и систем: учебник/ [Б. Я. Советов, В. А. Дубенецкий, В.В. Цехановский и др.; под ред. Б. Я. Советова. Издательский центр «Академия», 2010. – 432с.
3. Алгазинов, Э.К. Анализ и компьютерное моделирование информационных процессов и систем / Э.К. Алгазинов, А.А. Сирота. – М.: Диалог-МИФИ, 2009. – 416с.

10.2 Дополнительная литература

1. Васильев, В.В.Математическое и компьютерное моделирование процессов и систем в среде MATLAB/SIMULINK. Учебное пособие для студентов и аспирантов / В.В. Васильев, Л.А. Симак, А.М. Рыбникова. - К.: НАН Украины, 2008. – 91 с.
2. Васильев, К.К. Математическое моделирование систем связи: методические указания к лабораторным работам по дисциплине «Математическое моделирование каналов и систем телекоммуникаций» /К. К. Васильев, М. Н. Служивый. - Ульяновск: УлГТУ, 2007. – 24 с.
3. Потемкин, В.Г.Введение в MATLAB./В.Г. Потемкин - М.: Диалог-МИФИ, 2000. – 256 с.
4. <http://MATLAB.exponenta.ru/Simulink/default.php>– раздел сайта посвященный интерактивному инструменту для моделирования Simulink

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра Информационных систем и технологий

Направление подготовки 09.04.02 «Информационные системы и технологии»

Программа магистратуры «Управление данными»

ЗАДАНИЕ
на курсовую работу (проект)

студента _____

(фамилия, имя, отчество)

по дисциплине «Методы исследования и моделирования информационных процессов и технологий» _____

1. Тема работы _____

2. Цель: повышение уровня профессиональной подготовки путем углубления и закрепления теоретических знаний и практических навыков, приобретенных в результате изучения дисциплины «Методы исследования и моделирования информационных процессов и технологий»; подготовка к самостоятельному решению инженерных задач в соответствии с темой выпускной квалификационной работы. _____

3. Задачи _____

3.1 Анализ инженерной проблемы и методов ее решения. Обоснование выбора средств решения проблемы. _____

3.2 Моделирование информационных процессов и технологий в соответствии с темой выпускной квалификационной работы. _____

4. Перечень подлежащих разработке вопросов:

а) по теоретической части: анализ методов исследования и моделирования информационных процессов и технологий в соответствии с темой курсовой работы _____

б) по практической части: разработка моделей информационно-логических, функциональных, алгоритмических моделей проектируемой информационной системы и их компонентов _____

5. Исходные данные:

а) по литературным источникам _____

б) по вариантам, разработанным преподавателем _____

6. Список рекомендуемой литературы _____

7. Контрольные сроки представления отдельных разделов курсовой работы:

25 % - анализ литературы по теме «___» _____ 202_ г.

50 % - предоставление первого раздела «___» _____ 202_ г.

75 % - предоставление второго раздела «___» _____ 202_ г.

100 % - предоставление работы на отзыв «___» _____ 202_ г.

8. Срок защиты студентом курсовой работы «___» _____ 202_ г.

Дата выдачи задания «___» _____ 202_ г.

Руководитель курсовой работы

(ученая степень, звание)

(личная подпись)

(инициалы, фамилия)

Задание принял(а) к исполнению студент(ка) _____ формы
обучения

___ курса ___ группы

(личная подпись)

(инициалы, фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра информационных систем и технологий

КУРСОВАЯ РАБОТА

по дисциплине
«Методы исследования и моделирования информационных процессов и
технологий»
на тему:

«

»

Выполнил:

студент 1 курса группы ____

направления подготовки
09.04.02 «Информационные
системы и технологии»
очной формы обучения

(подпись)

Руководитель работы:

(ФИО, должность, кафедра)

Работа допущена к защите _____
(подпись руководителя)

(дата)

Работа выполнена и
защищена с оценкой _____

Дата защиты _____

Члены комиссии: зав. кафедрой информационных систем
и технологий (ИСТ)

(должность)

(подпись)

(И.О. Фамилия)

профессор кафедры ИСТ
(должность)

(подпись)

(И.О. Фамилия)

доцент кафедры ИСТ
(должность)

(подпись)

(И.О. Фамилия)

Ставрополь, 202_ г.

Отзыв

на курсовую работу (проект) студента/ки ____ курса

Ф.И.О., тема

Актуальность: курсовая работа посвящена

В первой главе

Вторая глава

Выводы, сделанные в Заключение, соответствуют целям, поставленным во Введении.

Проанализирован объем литературы...

За время работы студент/ка проявил/а себя как.....

Таким образом, работа выполнена на ... уровне, соответствует требованиям, предъявляемым к курсовым работам, и заслуживает ... оценки.

Научный руководитель
степень, звание, должность
место работы

Ф.И.О.

Печать

«.....».....202...г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт цифрового развития

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
К САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ
по дисциплине
**«Методы исследования и моделирования информационных
процессов и технологий»**
Направление подготовки 09.04.02
«Информационные системы и технологии»
Направленность (профиль) «Управление данными»
Квалификация выпускника: магистр

Ставрополь, 2023 г.

СОДЕРЖАНИЕ

Введение.....	29
1. Цель и задачи освоения дисциплины	29
2. Место дисциплины в структуре ОП ВО магистратуры	29
3. Связь с предшествующими дисциплинами	29
4. Связь с последующими дисциплинами	29
5. Компетенции обучающегося, формируемые в результате освоения дисциплины.....	Ошибка! Закладка не определена.
6. Структура и содержание дисциплины ..	Ошибка! Закладка не определена.
7. Организационно-методические рекомендации по освоению дисциплины	36
8. Методические указания по выполнению самостоятельной работы студентов.....	36
9. Содержание самостоятельной работы	37
9.1. Примерная тематика заданий для самостоятельной работы студентов	39
10. План-график выполнения СРС	Ошибка! Закладка не определена.
11. Организация контроля знаний студентов	40
11.1 Формы контроля знаний студентов.....	40
11.2. Рекомендации по подготовке к экзамену	41
12. Рекомендации по работе с литературой и источниками.....	41
12.2. Перечень рекомендуемой литературы	43
12.1.1. Основная литература:	Ошибка! Закладка не определена.
12.1.2. Дополнительная литература:	Ошибка! Закладка не определена.
12.1.3. Методическая литература:	43
12.1.4. Интернет-ресурсы:	43
13 Материально-техническое обеспечение дисциплины.....	45

Введение

1. Цель и задачи освоения дисциплины

Цель освоения дисциплины «Методы исследования и моделирования информационных процессов и технологий» формирование профессиональных компетенций будущего магистра по направлению подготовки 09.04.02 – Информационные системы и технологии и профилю подготовки «Управление данными».

Задачи дисциплины: использование методологии системного анализа для исследования информационных процессов, систем и технологий; моделирование информационных процессов и технологий; изучение математического аппарата и имитационного подхода к формализации информационных систем; использование информационных технологий при анализе и синтезе информационных систем и инструментальных средств моделирования информационных систем.

2. Место дисциплины в структуре ОП ВО магистратуры

Дисциплина «Методы исследования и моделирования информационных процессов и технологий» относится к дисциплинам (модулям) Б1, базовой части Б1.О.01.02 и изучается в 1 и 2 семестрах.

3. Связь с предшествующими дисциплинами

Для успешного освоения дисциплины студенты используют знания, умения, навыки, способы деятельности, полученные и сформированные на предыдущих ступенях обучения и при изучении дисциплины «Математическое обеспечение систем поддержки принятия решений».

4. Связь с последующими дисциплинами

Освоение данной дисциплины является необходимой основой для изучения дисциплин: Системы компьютерного моделирования бизнес-процессов; Научно-исследовательская работа; Подготовка к государственному экзамену; Государственный экзамен, а также для

выполнения научно-исследовательской работы и подготовки магистерской диссертации.

В результате освоения дисциплины обучающийся должен:

Перечень компонентов		Технологии формирования компетенции	Средства и технологии оценки
Знать: - основы планирования профессиональной траектории с учетом особенностей как профессиональной, так и других видов деятельности и требований рынка труда;	УК-6	традиционные лекции, самостоятельная работа, лабораторные работы	собеседование, письменный отчет
Знать: - Основные методы научного исследования и моделирования информационных процессов и технологий; - методы сбора, анализ научно-технической информации, отечественного и зарубежного опыта по тематике исследования	ОПК-4	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Знать: - методы разработки и модернизации ПО и автоматизированных систем; - современное оборудование и приборы (в соответствии с целями магистерской программы)	ОПК-5	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Знать: - методологию моделирования процессов и объектов на базе стандартных пакетов	ПК-1	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет

автоматизированного проектирования и исследований;			
Знать: -теоретические модели предметных областей и методы исследования теоретических и экспериментальных моделей информационных процессов и систем; - методы постановки и проведения экспериментов по заданной методике, методы анализа результатов эксперимента - методы анализа экспериментальных результатов; - критерии принятия оптимальных решений; - нормативные документы и требования к подготовке отчетов	ПК-2	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Знать: - методы планирования эксперимента обрабатывать и анализировать результаты с помощью современных информационных технологий	ПК-3	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Знать: - методики обучения персонала, методики преподавания в области исследований и моделирования информационных процессов и технологий	ПК-11	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет

Уметь: - расставлять приоритеты профессиональной деятельности и способы ее совершенствования на основе самооценки; - планировать самостоятельную деятельность в решении профессиональных задач; - подвергать критическому анализу проделанную работу; — находить и творчески использовать имеющийся опыт в соответствии с задачами саморазвития	УК-6	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Уметь: - осуществлять методологическое обоснование научного исследования; - применять современные методы научных исследований для формирования суждений и выводов по проблемам моделирования информационных процессов, технологий и систем - осуществлять сбор, анализ научно-технической информации, отечественного и зарубежного опыта по тематике исследования	ОПК-4	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Уметь: - исследовать	ОПК-5	традиционные лекции,	Собеседование,

теоретические и экспериментальные модели объектов профессиональной деятельности в различных областях науки и техники с целью модернизации ПО или разработки. - выделять в профессиональной информации главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями		самостоятельная работа, лабораторные работы	письменный отчет
Уметь: - осуществлять моделирование процессов и объектов на базе стандартных пакетов автоматизированного проектирования и исследований и применять на практике методы и средства проектирования процессов и объектов	ПК-1	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Уметь: -осуществлять постановку и проведение экспериментов по заданной методике; - проводить анализ результатов эксперимента - осуществлять выбор оптимальных решений -подготавливать и составлять отчеты.	ПК-2	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет

Уметь: - составить план эксперимента, провести эксперименты, обрабатывать и анализировать результаты с помощью современных информационных технологий	ПК-3	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Владеть: - навыками выявления стимулов для саморазвития; - навыками определения реалистических целей профессионального роста.	УК-6	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Владеть: -навыками анализа научного исследования и его результатов; - методами научного поиска и интеллектуального анализа научной информации при решении новых задач - методами сбора, анализ научно-технической информации, отечественного и зарубежного опыта по тематике исследования	ОПК-4	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Владеть: - методами анализа и синтеза информационных систем и методами разработки и исследования теоретических и экспериментальных моделей информационных процессов систем и	ОПК-5	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет

технологий - методами проектирования информационных процессов и объектов; - методами анализа профессиональной информации; - методами разработки аналитических обзоров с обоснованными выводами и рекомендациями.			
Владеть: - программными инструментами проведения эксперимента, обработки и анализа результатов с помощью современных информационных технологий	ПК-1	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Владеть: - методами постановки и проведение экспериментов по заданной методике; - методами анализа результатов эксперимента	ПК-2	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет
Владеть: - методами анализа экспериментальных результатов; - методами принятия оптимальных решений; - методологией подготовки отчетов.	ПК-3	традиционные лекции, самостоятельная работа, лабораторные работы	Собеседование, письменный отчет

5. Организационно-методические рекомендации по освоению дисциплины

Самостоятельная работа студентов является важнейшим условием формирования научного способа познания. Она проводится накануне каждого лабораторного занятия и включает изучение необходимого для выполнения лабораторной работы теоретического материала, а также подготовку отчета по выполненным на лабораторном занятии заданиям.

Подготовленный материал оформляется в виде отчета.

Самостоятельные занятия (СЗ) являются одной из активных форм обучения.

Самостоятельные занятия по дисциплине «Методы исследования и моделирования информационных процессов и технологий» имеют целью:

- закрепить и углубить знания, полученные студентами на лекциях и в процессе лабораторных занятий;
- привить практические навыки при разработке, эксплуатации и исследовании информационных процессов, систем и технологий.

Самостоятельные занятия проводятся в специализированных аудиториях, оборудованных СВТ и возможностью пользования Интернет-ресурсами, в библиотеке, а также дома.

Предлагаемые методические рекомендации содержат информацию для студентов, необходимую при подготовке и проведении лабораторных занятий по дисциплине «Методы исследования и моделирования информационных процессов и технологий».

6. Методические указания по выполнению самостоятельной работы студентов

Методические указания должны включать следующие разделы:

- цель работы;

- задание, которое должно быть выполнено студентом в результате проведения самостоятельной работы;
- варианты индивидуальных заданий;
- основные теоретические положения, необходимые для выполнения задания, они должны быть краткими и содержать ссылки на литературу, в которой эти положения изложены в объеме, достаточном для выполнения самостоятельной работы;
- этапы выполнения задания с указанием конкретных сроков выполнения каждого из этапов и всего задания в целом;
- требования к оформлению графической и текстовой части самостоятельной работы;
- пример выполнения одного из вариантов задания и оформления отчета;
- библиографический список использованных источников
-

7. Содержание самостоятельной работы

Наименование разделов и тем дисциплины, их краткое содержание; вид самостоятельной работы представлено в таблице 2.

Таблица 2 – Наименование разделов и тем дисциплины, их краткое содержание; вид самостоятельной работы

№	Наименование разделов и тем дисциплины, их краткое содержание; вид самостоятельной работы	Форма контроля
	1 семестр	
1	подготовка к лабораторным занятиям	защита
3	Самостоятельное изучение тем Раздел 1. Общая характеристика информационных процессов, систем и технологий Тема 1. Информационные процессы – основа информационной технологии. Взаимосвязь информационных процессов, систем и технологий. Информационная технология – составная часть информатики и основа проектирования информационных систем. Информационные процессы – основа функционирования информационных систем	опрос

	Тема 2. Характеристика и модели базовых информационных процессов. Понятие и структура информационного процесса . Раздел 3. Применение методов системного анализа и информационно-аналитических технологий при проектировании информационных систем Тема 6. Морфологические методы и генерация альтернативных вариантов системы. Раздел 4. Компьютерное и имитационное моделирование процессов и систем Тема 9. Типовые математические схемы элементов сложной системы систем и взаимодействия элементов системы..	
2 семестр		
1	подготовка к лабораторным занятиям	защита
2	Выполнение курсовой работы. Подготовка презентации	защита

Наименование тем дисциплины, цель, форма контроля, задания, требования к оформлению , перечень литературных источников представлен в таблице3.

Таблица 3 – Наименование тем дисциплины, цель, форма контроля, задания и требования к оформлению самостоятельной работы

Название темы	Цель	Форма контроля	Задания для СРС	Требования
Подготовка к лекциям и лабораторным занятиям	изучить процессы и поколения управления данными, а также тенденции их развития	индивидуальное собеседование	Сформулировать ответы на контрольные вопросы к лабораторным работам	Устно ответить на контрольные вопросы к лабораторным работам
Характеристики и модели базовых информационных процессов.	изучить модели базовых информационных процессов.	индивидуальное собеседование	сформулировать и письменно ответить на вопросы для контроля	письменно ответить на вопросы для контроля
Задачи анализа и синтеза систем.	Изучить задачи анализа и синтеза	индивидуальное собеседование	владения компетенциями данного раздела	компетенциями вопросы данного раздела программы

	систем.		программы	
Моделирование и проектирование сложных систем.	изучить методологию моделирования и проектирования сложных систем.			
Морфологические методы; типовые	изучить морфологические методы			
Математические схемы элементов сложной системы	изучить математические схемы элементов сложной системы			

8. Примерная тематика заданий для самостоятельной работы студентов

Примерные темы для самостоятельного изучения по дисциплине «Методы исследования и моделирования информационных процессов и технологий»:

1. Методы анализа качества информационных процессов и систем.
2. Современные средства вычислительной техники, коммуникаций и связи.
3. Применение методологии системного анализа и информационных технологий при проектировании и исследовании информационных систем.
4. Перспективы и тенденции развития информационных технологий;
5. Порядок, методы и средства защиты интеллектуальной собственности.
6. Основные требования к организации труда при исследовании и разработке информационных процессов и систем.

9. Организация контроля знаний студентов

9. 1 Формы контроля знаний студентов

Контроль и оценка знаний, умений и навыков студентов осуществляется на лабораторных занятиях, консультациях, при сдаче экзамена. В ходе контроля знаний преподаватель оценивает понимание студентом содержания дисциплины «Методы исследования и моделирования информационных процессов и технологий», его способность анализировать развитие информационных систем и технологий.

Контроль знаний студентов может осуществляться в следующих формах:

- текущий контроль знаний;
- итоговый контроль знаний.

Текущий контроль знаний студентов имеет целью: дать оценку работы каждого студента по усвоению им учебного материала, выявить недостатки в его подготовке и оказать практическую помощь в их устранении;

Основными формами текущего контроля знаний студентов являются:

- устный контрольный опрос;
- защита лабораторной работы;
- проверка конспектов лекций.

Устный контрольный опрос студентов проводится на лекциях (и лабораторных занятиях). По его результатам преподаватель оценивает качество подготовки студента к занятию.

На лабораторных занятиях знания и практические навыки студентов оцениваются по 5-балльной системе. Полученные оценки выставляются в журнале.

При проверке конспектов лекций дается анализ качества их ведения. Отмечаются допущенные ошибки, в рецензии преподавателя оценивается

качество конспектирования учебного материала, даются рекомендации по улучшению качества конспектирования лекционного материала.

9.2. Рекомендации по подготовке к экзамену

Подготовка к зачету начинается с начала изучения дисциплины.

Необходимо посещать все лекции и лабораторные занятия.

Экзамен, как итоговый контроль знаний студентов имеет целью проверить и оценить учебную работу студентов, уровень полученных знаний и практических навыков.

Экзамен проводится во 2 семестре после защиты всех лабораторных работ в объеме учебной программы.

10. Рекомендации по работе с литературой и источниками

Изучение литературы и источников необходимо начинать с прочтения соответствующих глав учебных изданий, учебных пособий или литературы, рекомендованной в качестве основной или дополнительной по дисциплине «Методы исследования и моделирования информационных процессов и технологий», которые прямо или косвенно относятся к изучаемой теме.

При изучении литературы и источников студенту рекомендуется вести краткий конспект. Однако не следует переписывать все содержание изучаемой темы, нужно выписывать лишь основные идеи и главные на ваш взгляд мысли. В отдельных случаях, когда встречаются важные определения, понятия, необходимый фактический материал и примеры, статистическая информация, имеющие отношение к изучаемой теме, студенту следует выписать их в виде цитат с полным указанием библиографических источников.

Конспектирование рекомендуемой литературы и источников необходимо вести с распределением собранных материалов по отдельным главам и параграфам согласно учебно-тематическому плану. Необходимо

выписывать все выходные данные по используемой литературе и источникам.

Основой технологии интенсификации обучения на платформе цифровых образовательных технологий являются учебно-иллюстрационные материалы (опорный конспект) по дисциплине «Методы исследования и моделирования информационных процессов и технологий».

Работа с учебно-иллюстрационными материалами имеет следующие этапы.

1. Изучение теоретических основ учебного материала в аудитории: изложение преподавателем изучаемого материала студентам с объяснением по опорному конспекту;

2. Самостоятельная работа: индивидуальная работа студентов по опорному конспекту; фронтальное закрепление по блокам опорного конспекта.

3. Первое повторение - воспроизведение содержания заданной темы опорного конспекта по памяти.

4. Устное проговаривание материала опорного конспекта – необходимый этап внешне речевой деятельности при усвоении учебного материала.

5. Второе повторение – взаимопрос и взаимопомощь студентов друг другу.

Применение учебно-иллюстрационных материалов позволяет обобщить сложный по содержанию материал, активизировать мыслительную деятельность студентов.

Необходимо помнить, что главное для студента в самостоятельной работе с рекомендуемой литературой и источниками - это формирование своего индивидуального стиля, который может стать основой в будущей профессиональной деятельности.

11 Перечень рекомендуемой литературы

11.1. Основная литература

Основная литература:

1. Шагрова, Г.В. Методы исследования и моделирования информационных процессов и технологий : учебное пособие / Г.В. Шагрова, И.Н. Топчиев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь : СКФУ, 2016. - 180 с. : ил. - Библиогр.: с. 178 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=458289> (13.05.2019).

2. Шкундин, С.З. Теория информационных процессов и систем : учебное пособие / С.З. Шкундин, В.Ш. Берикашвили. - М. : Горная книга, 2012. - 475 с. - ISBN 978-5-98672-285-6 [Электронный ресурс].

URL: <http://biblioclub.ru/index.php?page=book&id=229031>

Дополнительная литература:

1. Буканова, Т. С. Моделирование систем управления: учебное пособие / Т.С. Буканова, М.Т. Алиев ; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 144 с. : ил., граф. - <http://biblioclub.ru/>. - Библиогр. в кн. - ISBN 978-5-8158-1899-6, экземпляров неограниченно. Режим доступа: https://biblioclub.ru/index.php?page=book_red&id=483694&razdel=259

2. Методы исследования и моделирования информационных процессов и технологий : лабораторный практикум / авт.-сост. Г.В. Шагрова, М.Г. Романенко, И.Н. Топчиев ; Министерство образования и науки Российской Федерации и др. - Ставрополь : СКФУ, 2016. - 241 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=458081>

11.3. Методическая литература

– Шагрова Г.В. Методы исследования и моделирования информационных процессов и технологий: лабораторный практикум / [Г.В. Шагрова, М.Г. Романенко, И.Н. Топчиев] – Ставрополь: Изд-во СКФУ, 2016. – 241 с.

– УМК по дисциплине «Методы исследования и моделирования информационных процессов и технологий» (электронных вариант).

11.4. Интернет-ресурсы

4 Каныгин, Г.И. Учебный курс «Методы исследования и моделирования информационных процессов и технологий», <http://moodle.dstu.edu.ru/course/view.php?id=1649>

5 Тарасик В.П.», Учебный курс «Математическое моделирование технических систем, <http://bookfi.net/book/759430>

6. Самарский А. А., Михайлов А. П. / Математическое моделирование: Идеи. Методы. Примеры. <http://www.vixri.ru/?p=11793>

7. Хачатурова С.М. Математические модели системного анализа: Электронный учебник. <http://ermak.cs.nstu.ru/mmsa/main/Proba.htm>

8. Учебный курс «Моделирование систем», <http://www.intuit.ru/speciality/intuitdpo/modelsys/>

9. Учебный курс «Компьютерное моделирование», <http://www.intuit.ru/department/calculate/compmodel/lit.html>

10 <http://www.intuit.ru> – Сайт национального открытого университета «Интуит»

11 <http://www.citforum.ru> – Сайт центра информационных технологий

12. Материально-техническое обеспечение дисциплины

- индивидуальное взаимодействие со студентами по электронной почте для предварительного ознакомления с их разработками при подготовке к аудиторным занятиям;
- использование на лекциях и лабораторных занятиях мультимедийного оборудования для демонстрации листингов программного кода, скриншотов выполненных запросов, работы программ и пр.;
- включение в лабораторные работы индивидуального поиска, систематизации и анализа информации через Интернет;
- авторские презентации к лекциям;
- лекционная аудитория должна быть оборудована мультимедиа проектором;
- Лабораторные работы рассчитаны на компьютеры, имеющие доступ к Internet.

12.1. Программное обеспечение:

Office (полная версия), система компьютерной математики «Matlab», методические указания к лабораторным работам в электронном виде.

12.2. Материально-техническое обеспечение дисциплины (модуля)

Мультимедиа аудитория: проектор, акустическая система, аудио-усилитель, ноутбук, Internet.

Компьютерный класс. Высокопроизводительная ЭВМ для разработки и тестирования в среде MathWorks Matlab и MathWorks Simulink, SimInTech, создания отчетов по выполненным лабораторным работам в пакете Microsoft Office, Internet, локальная вычислительная сеть.