

ИЗ ФОНДОВ РОССИЙСКОЙ ГОСУДАРСТВЕННОЙ БИБЛИОТЕКИ

Кисляков, Сергей Викторович

1. Разработка и исследование метода распознавания
фонем русского языка на основе аппарата линейного
предсказания

1.1. Российская государственная Библиотека

diss .rsl. ти
2005

Кисляков, Сергей Викторович

Разработка и исследование метода
распознавания фонем русского языка на основе
аппарата линейного предсказания [Электронный
ресурс]: Дис. ... канд. техн. наук : 05.12.13
.-М.: РГБ, 2005 (Из фондов Российской
Государственной Библиотеки)

Электроакустика — Речевые сигналы и их
преобразование — Распознавание речи.
Вычислительные машины электронные цифровые —
Программирование — Диалоговое
программирование. Системы, сети и
устройства телекоммуникаций

Полный текст:

<http://diss.rsl.ru/diss/05/0237/050237003.pdf>

Текст воспроизводится по экземпляру,
находящемуся в фонде РГБ:

Кисляков, Сергей Викторович

Разработка и исследование метода распознавания
фонем русского языка на основе аппарата
линейного предсказания

Санкт-Петербург 2004

Российская государственная библиотека, 2005
год (электронный текст).

0 / : (95- 5/535-

Санкт-Петербургский государственный университет
телекоммуникаций им. проф. М.А. Бонч-Бруевича

Кисляков

Сергей Викторович

**РАЗРАБОТКА И ИССЛЕДОВАНИЕ МЕТОДА РАСПОЗНАВАНИЯ
ФОНЕМ РУССКОГО ЯЗЫКА
НА ОСНОВЕ АППАРАТА ЛИНЕЙНОГО ПРЕДСКАЗАНИЯ**

05.12.13 - Системы, сети и устройства телекоммуникаций

Диссертация на соискание ученой степени

кандидата технических наук



Санкт-Петербург 2004

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 АНАЛИЗ ПРОБЛЕМЫ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ РЕЧИ.....И	
1.1 ОСНОВНЫЕ ЗАДАЧИ ПОСТРОЕНИЯ СИСТЕМ РАСПОЗНАВАНИЯ	11
<i>i</i> 1.2 ВОСПРИЯТИЕ УСТНОЙ РЕЧИ	14
1.3 ОБЩАЯ СТРУКТУРА РАСПОЗНАЮЩЕЙ СИСТЕМЫ	17
1.4 АНАЛИЗ СОСТОЯНИЯ ИССЛЕДОВАНИЙ ПО ПРОБЛЕМЕ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ РЕЧИ.....	20
1.5 ОБЗОР РЫНКА ПРОГРАММНЫХ СРЕДСТВ	21
1.6 СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ВЫДЕЛЕНИЯ ПРИЗНАКОВ РЕЧЕВЫХ СИГНАЛОВ.....	28
1.6.1 АНАЛИЗ СИГНАЛА ВОЗБУЖДЕНИЯ ГОЛОСОВОГО ТРАКТА	29
1.6.2 АНАЛИЗ КЛИППИРОВАННОГО СИГНАЛА.....	30
1.6.3 ФОРМАНТНЫЙ АНАЛИЗ	30
1.6.4 СПЕКТРАЛЬНЫЙ АНАЛИЗ	32
1.6.5 КОРРЕЛЯЦИОННЫЙ АНАЛИЗ	34
1.6.6 СКРЫТОЕ МАРКОВСКОЕ МОДЕЛИРОВАНИЕ	34
1.6.7 ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЕ	35
1.6.8 ЛИНЕЙНОЕ ПРЕДСКАЗАНИЕ	37
1.7 ВЫВОДЫ.....	39
2 ФОРМИРОВАНИЕ РАБОЧЕГО СЛОВАРЯ ПРИЗНАКОВ ФОНЕМ.....	40
2.1 ОЦЕНКА ПАРАМЕТРОВ РЕЧЕВОГО СИГНАЛА	41
2.2 АНАЛИЗ ТОЧНОСТИ МОДЕЛИ ЛИНЕЙНОГО ПРЕДСКАЗАНИЯ	52
2.3 СТАТИСТИЧЕСКИЕ СВОЙСТВА ЛСК	55
2.4 ВЫБОР МЕТОДА ОЦЕНКИ ОШИБКИ РАЗДЕЛЕНИЯ ФОНЕМ В ПОДПРОСТРАНСТВАХ ЛСК.....	64
2.4.1 ИСПОЛЬЗОВАНИЕ СТАТИСТИЧЕСКОГО КРИТЕРИЯ НА ОСНОВЕ ГИСТОГРАММ.....	64
2.4.2 ИСПОЛЬЗОВАНИЕ МЕТОДА ДИНАМИЧЕСКИХ СГУЩЕНИЙ	66
2.5 РАСЧЕТ ОШИБКИ РАЗДЕЛЕНИЯ ФОНЕМ С ИСПОЛЬЗОВАНИЕМ МДС	70
2.5.1 ОЦЕНКА РАЗДЕЛИМОСТИ ФОНЕМ В ДВУМЕРНОМ ПОДПРОСТРАНСТВЕ ЛСК	70
2.5.2 ОЦЕНКА РАЗДЕЛИМОСТИ ФОНЕМ В ТРЕХМЕРНОМ ПОДПРОСТРАНСТВЕ ЛСК.....	74
2.5.3 ОЦЕНКА РАЗМЕРНОСТИ ВЕКТОРА ПРИЗНАКОВ	76
2.6 ПРИМЕНЕНИЕ ПРЕОБРАЗОВАНИЯ КАРУНЕНА-ЛОЭВА ДЛЯ СОКРАЩЕНИЯ РАЗМЕРНОСТИ ВЕКТОРОВ ПРИЗНАКОВ	79
2.7 ВЛИЯНИЕ РАНЖИРОВАНИЯ ПРИЗНАКОВ НА РАЗМЕРНОСТЬ ВЕКТОРОВ ПРИЗНАКОВ.....	93
2.8 ОЦЕНКА ВЛИЯНИЯ ПАРАМЕТРОВ РАСЧЕТА ЛСК НА КАЧЕСТВО РАЗДЕЛЕНИЯ ФОНЕМ.....	96
2.9 ВЫВОДЫ	99
3 ФОРМИРОВАНИЕ КЛАСТЕРОВ ФОНЕМ НА ОСНОВЕ РАБОЧЕГО СЛОВАРЯ ПРИЗНАКОВ	101
3.1 Поиск ОПТИМАЛЬНЫХ ПАРАМЕТРОВ РАСЧЕТА ЛСК	101
3.1.1 Выбор РАЗМЕРА ВРЛМ ИЛИ ДРУГОГО ОКНА.....	101
3.1.2 Выбор ШАГА ВРЕМЕННОГО ОКНА	103
3.2 ФОРМИРОВАНИЕ ПРИЗНАКОВЫХ ПОДПРОСТРАНСТВ	104
3.3 ЗНАЧЕНИЯ ЛСК - ПРИЗНАКИ ПЕРВОГО УРОВНЯ	104
3.3.1 ФОРМИРОВАНИЕ ДЕРЕВА ПРИНЯТИЯ РЕШЕНИЙ (ДПР)	104
3.3.2 Поиск СТАРТОВОГО ПОДПРОСТРАНСТВА ДПР	105
3.3.3 ФОРМИРОВАНИЕ УЗЛОВ(ПЕРЕХОДОВ)ДПР.....	108

3.3.4	Связь РАЗМЕРНОСТИ ПОДПРОСТРАНСТВ И ВЕЛИЧИНЫ ОШИБКИ В УЗЛАХ ДПР	109
3.4	ПРИЗНАКИ ВТОРОГО УРОВНЯ	110
3.4.1	ФОРМИРОВАНИЕ КЛАСТЕРОВ ФОНЕМ НА ОСНОВЕ СКОЛЬЗЯЩИХ СРЕДНИХ ЛСК	110
3.4.2	СКОЛЬЗЯЩИЕ ДИСПЕРСИИ КАК ДОПОЛНИТЕЛЬНЫЕ ПРИЗНАКИ ФОНЕМ	120
3.4.3	СОБСТВЕННЫЕ ВЕКТОРЫ КОВАРИАЦИОННЫХ МАТРИЦ ФОНЕМ - АЛЬТЕРНАТИВНЫЕ ПРИЗНАКОВЫЕ ПОДПРОСТРАНСТВА	122
3.5	Выводы	125
4	<u>РАСПОЗНАВАНИЕ ФОНЕМ НА ОСНОВЕ ЛСК</u>	<u>127</u>
4.1	МЕТОД РАСПОЗНАВАНИЯ НА ОСНОВЕ ЛСК с ИСПОЛЬЗОВАНИЕМ ДПР	128
4.1.1	ПРОЦЕДУРА ОБУЧЕНИЯ	128
4.1.2	ИССЛЕДОВАНИЕ АЛГОРИТМА РАСПОЗНАВАНИЯ	131
4.2	МЕТОД РАСПОЗНАВАНИЯ НА ОСНОВЕ СС ЛСК	134
4.2.1	ПРОЦЕДУРА ОБУЧЕНИЯ	134
4.2.2	ИССЛЕДОВАНИЕ АЛГОРИТМА РАСПОЗНАВАНИЯ	135
4.2.3	НЕЙТРАЛИЗАЦИЯ ВЛИЯНИЯ ДИФТОНГОВ	136
4.3	РАСПОЗНАВАНИЕ НА ОСНОВЕ КОМБИНИРОВАННОГО АЛГОРИТМА С НЕЙТРАЛИЗАЦИЕЙ ВЛИЯНИЯ ДИФТОНГОВ	138
4.4	Выводы	140
	<u>ЗАКЛЮЧЕНИЕ</u>	<u>141</u>
	<u>СПИСОК ЛИТЕРАТУРЫ</u>	<u>143</u>
	<u>ОПУБЛИКОВАННЫЕ СТАТЬИ</u>	<u>147</u>
	<u>ПРИЛОЖЕНИЯ</u>	<u>148</u>
	ПРИЛОЖЕНИЕ 1. ВЫЧИСЛЕНИЕ ЛСК	148
	ПРИЛОЖЕНИЕ 2. ВЫЧИСЛЕНИЕ ЛСК НА СИГНАЛЕ ЛЮБОЙ ДЛИТЕЛЬНОСТИ	151
	ПРИЛОЖЕНИЕ 3. ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АНАЛИЗА РЕЧЕВЫХ СИГНАЛОВ	155
	ПРИЛОЖЕНИЕ 4. ОПИСАНИЕ РЕАЛИЗАЦИИ ПРОГРАММЫ	170
	ПРИЛОЖЕНИЕ 5. ЛИСТИНГ ПРОГРАММЫ	184

ВВЕДЕНИЕ

До недавнего времени процесс речевого общения человека и компьютера был непременным атрибутом научно-фантастических романов и никем не воспринимался всерьез. Несколько лет назад ситуация кардинально изменилась. Сегодня использование речевых технологий в прикладных программах в качестве альтернативного средства взаимодействия в системе «человек-компьютер» приобретает все больший размах. Такой процесс носит вполне обоснованный и объективный характер в силу ряда причин. Во-первых, развитие речевых средств взаимодействия с персональным компьютером лежит в рамках мировой тенденции «очеловечивания» ПК, т.е. позволяет создавать интерфейсы, максимально дружественные пользователю. Во-вторых, миниатюризация современных средств управления и связи требует принципиально новых подходов к осуществлению взаимодействия пользователя с такого типа устройствами. Для современных технологий не представляет большого труда создание, например, мобильного телефона размером с авторучку, однако механический набор номера на таком телефоне будет сопряжен с определенными трудностями. Голосовой набор номера и авторизация в этом случае являются очевидным и наиболее подходящим выходом. В-третьих, для большого круга пользователей речевой способ общения с ПК является единственно возможным в силу ограниченности их физических возможностей (люди с нарушениями опорно-двигательного аппарата, другими физическими недостатками, слепые и т.д.) либо специфики профессии.

Большое значение приобрели также задачи, связанные с быстрым поиском и получением от больших информационно-вычислительных систем («информационных банков») нужных сведений в виде обычных речевых сообщений, передаваемых по телефонным каналам. Все это сделало проблему автоматического распознавания речи разносторонней и актуальной.

Можно указать на следующие, наиболее перспективные области применения автоматического распознавания и синтеза речи:

- сжатие речи для передачи ее по более простым и экономичным узкополосным линиям связи;
- установление прямой речевой связи с ЭВМ специального назначения и с поисково-информационными системами, снабжающими абонентов различной информацией;
- управление голосом (посредством устных команд) различными процессами и машинами (в том числе боевыми) без помощи других управляющих средств или в комбинации с ними;
- идентификация или верификация личности по голосу;
- автоматический синхронный перевод с одного языка на другой;
- создание устройств для понимания устной речи глухими и чтения книг слепыми.

В настоящее время сильно развивается синтетическая телефония. При этом тесно связываются между собой такие области как цифровая обработка сигналов (ЦОС), синтез речи, анализ (распознавание) речи. Теоретические и практические разработки в области ЦОС, кодирования и передачи РС ведутся российскими и зарубежными учеными многие годы. Большой вклад в теоретическом и практическом планах внесли А.А. Пирогов, В.Н. Трунин-Донской, А.А. Ланнэ, Л. Рабинер, Р. Шафер, Д. Макхоул и др.

Исследования по машинному распознаванию речи ведутся сравнительно недавно, однако за это время написано множество теоретических работ и предложен ряд практических реализаций систем распознавания речи (СРР). При этом достаточно большое количество идей взято из области ЦОС. Существенный вклад в развитие СРР внесли советские и российские ученые В.Н. Трунин-Донской, Т.К. Винцюк, Н.Г. Загоруйко, Ю.А. Косарев, Л.Л. Мясников, зарубежные ученые Д.Д. Маркел, А.Х. Грей, Б. Гоулд, Г. Фант и др. Тем не менее задача качественного распознавания русской речи еще не решена.

Восприятие речи есть сложный многоуровневый процесс, в котором тесно переплетены различные уровни - акустический, лингвистический, смысловой. В многоуровневой модели распознавания речи ее качество (надежность) определяется качеством реализации каждого

тельно взятого уровня.

На настоящий момент лингвистическая часть модели проработана удовлетворительно, что подтверждается многими исследователями, а также наличием программ, распознающих написанный или напечатанный текст. Этого нельзя утверждать применительно к акустической части модели распознавания. Объясняется это следующими причинами:

1. Неустойчивостью и случайностью параметров РС, их изменением при смене дикторов, различными внешними факторами (механические перегрузки, пониженные температуры окружающей среды, темнота и т. д.). Основными причинами затруднений при разработке акустической части СРР является отсутствие на настоящий момент системы признаков, позволяющей с высокой точностью определять «что поступает» на вход распознающей системы независимо от обозначенных причин.

2. Недостаточной адекватностью используемых моделей РС, что не позволяет получить качественные признаки для их дальнейшей обработки.

3. Большинство исследователей считает, что нельзя «хорошо понимать» речь, не используя при этом закономерности фонетического уровня.

Объектом исследования являются модели и методы выделения акустических признаков фонем для систем распознавания русской речи.

Предметом исследования являются линейные спектральные корни (ЛСК) применительно к задаче распознавания речи на фонетическом уровне.

Состояние проблемы и задачи исследования. К настоящему времени известно большое число моделей и методов выделения первичных (акустических) признаков РС. Одним из самых перспективных методов многие исследователи (Ю.А. Косарев, С.В. Виноградов, Н.Г. Загоруйко, Д. Макхоул и др.) считают аппарат линейного предсказания (ЛП): полученные на основе ЛП признаки обладают рядом практически полезных свойств - они просто рассчитываются, дают компактное представление РС, позволяют контролировать устойчивость, наименее чувствительны к действиям помех. Наиболее широко при разработке «речевых» сие-

-7-тем используются ЛСК, предложенные Итакурой. С появлением обобщающей теории ЛСК (А.А. Ланнэ «Новая теория линейных спектральных корней») оказалось возможным получать новые (отличные от уже известных) ЛСК, которые еще не исследованы с точки зрения применимости в СРР.

Целью диссертационной работы является исследование ЛСК для их использования в качестве акустических признаков фонем в СР слитной русской речи.

Для достижения поставленной цели необходимо в ходе выполнения теоретических и экспериментальных исследований решить следующие задачи:

1. Провести анализ существующих структур СРР и методов формирования признаков фонем в СРР на акустическом уровне.
2. Разработать математическое и программное обеспечение для проведения исследований и тестирования разработанных моделей.
3. Исследовать свойства ЛСК как признаков фонем:
 - разработать параметрическую модель расчета ЛСК для ПЭВМ;
 - исследовать статистические свойства ЛСК;
 - исследовать влияние параметров расчета ЛСК на качество разделения фонем в пространствах ЛСК;
 - на основании предложенных критериев осуществить выбор оптимальных параметров расчета ЛСК.
4. Разработать компактный рабочий словарь признаков фонем, опираясь на известные свойства ЛСК.
5. Разработать метод формирования кластеров фонем на основе рабочего словаря признаков фонем.
6. Построить тестовую программную модель распознавания фонем для ЭВМ.
7. Осуществить проверку предложенного метода распознавания фонем путем проведения машинных экспериментов на тестовых РС.

-8- Методы исследования. Решение указанных

задач осуществлено на основе применения методов теории вероятности и математической статистики, теории синтеза линейных электрических цепей, цифровой обработки сигналов.

Основные положения, выносимые на защиту:

1. ЛСК - акустические признаки фонем, обладающие высокой компактностью и слабой чувствительностью к смене диктора.
2. Применение преобразования Карунена-Лоэва (на этапе формирования рабочего словаря признаков фонем) позволяет ввести ранжирование признаков по степени их информативности. Исключение из описания фонем «неинформативных» признаков позволяет получить компактный рабочий словарь признаков с размерностью векторов признаков не более трех.
3. Решение задачи повышения надежности распознавания фонем на основе ЛСК может быть решена путем поиска наилучших (в смысле минимума ошибки распознавания) параметров модели линейного предсказания методом динамических сгущений.
4. Методы формирования кластеров фонем на основе признаков «первого уровня» (значений ЛСК) и «второго уровня» (скользящих средних ЛСК) позволяют сформировать признаковые подпространства малых размерностей (2-3), в которых образы фонем хорошо разделяются независимо от диктора. Решена задача поиска оптимальных (в объявленном смысле) подпространств, в которых ошибка разделения фонем минимальна.
5. Алгоритмы распознавания фонем с использованием предлагаемых методов формирования кластеров позволяют выделять фонемы из речевого потока независимо от диктора.

Практическая ценность. На основании результатов исследований сделаны выводы о возможности применения ЛСК в СРР, разработана и доведена до практической реализации на ПЭВМ модель акустического блока СРР, исследована её надежность (качество распознавания).

Результаты внедрения. Разработанные методы, модели и программы являются частью

-9- работ, проводимых в рамках НИР № 190-93-054 по каф. ЦВТИ в 2000-2002 г. СПбГУТ под руководством д.т.н., проф. Е.А. Шульгина.

Результаты научных исследований и практические разработки используются «Центром речевых технологий» Санкт-Петербурга, теоретические результаты внедрены в учебный процесс СПбГУТ им. проф. М.А. Бонч-Бруевича.

Апробация работы. Основные положения, выводы и практические результаты диссертационной работы обсуждались на конференциях:

- 2-я международная научно - техническая конференция студентов, аспирантов и молодых специалистов СПбГУТ в 2000 г;
- Международная научно-практическая конференция «Компьютерные технологии в науке, производстве, социальных и экономических процессах», г. Новочеркасск, 2002 г;
- 52-я, 53-я, 54-я, 55-я научно-технические конференции СПбГУТ.

Часть результатов и выводов представлена в отчете по НИР № 190-93-054, проводившейся в СПбГУТ в 2000-2002 гг. под руководством д.т.н., проф. Е.А. Шульгина.

По теме диссертационной работы опубликовано 10 печатных работ, в том числе отчет по НИР.

Объем и структура диссертации. Диссертационная работа включает введение, четыре главы, заключение, список литературы и приложения. Вся работа изложена на 193 страницах текста, включающих в себя 47 страниц приложений, 67 рисунков, 9 таблиц. Количество библиографических ссылок- 101.

Работа по теме диссертации проводилась с 1998 по 2004 годы на кафедре ЦВТИ в СПб ГУТ им. проф. М.А. Бонч-Бруевича.

Принятые в работе сокращения

КЛП - коэффициенты линейного предсказания

ЛП - линейное предсказание

ЛСК - линейные спектральные корни I

ПП - признаковые подпространства

СВКМ - собственные векторы ковариационных матриц фонем

СММ - скрытые марковские модели

СС ЛСК - скользящие средние ЛСК

ДПР - дерево принятия решений

МДС - метод динамических сгущений

ОТ - основной тон

ПОТ - период основного тона

ЧОТ - частота основного тона

СР - система распознавания

АРСО - автоматическое распознавание слуховых образов

1 АНАЛИЗ ПРОБЛЕМЫ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ РЕЧИ

1.1 Основные задачи построения систем распознавания

Каждая система распознавания приспособлена для распознавания только данного вида объектов или явлений: так, система, предназначенная для диагностики заболеваний, не умеет диагностировать отказы аппаратуры, а система, предназначенная для чтения букв рус-ского алфавита, не умеет читать китайские иероглифы или ноты.

Рассмотрим основные задачи, возникающие в процессе проектирования и построения систем распознавания. При этом необходимо иметь ввиду следующее. Процесс обработки системы распознавания требует построения математической или физико-математической модели системы. Только наличие подобной модели позволяет реализовать итеративный процесс построения прообразов системы распознавания, все более и более приближающихся по своим характеристикам (точностным, временным, габаритным, весовым, стоимостным и т.д.) к требуемым характеристикам, задаваемым на стадии разработки тактико-технических требований к системе. Рассматриваемые ниже задачи в той или иной мере, с одной стороны, обеспечивают построение модели системы, а с другой стороны, могут быть решены только с помощью модели.

Задача 1. Задача заключается в том, чтобы определить полный перечень признаков (параметров), характеризующих объекты или явления, для распознавания которых разрабатывается данная система. Названная совокупность признаков должна быть сформирована без каких-либо ограничений, связанных как с получением априорной информации, необходимой для исходного описания классов объектов, так и с получением апостериорной информации о конкретных объектах, подлежащих распознаванию. Наоборот, первоначально необходимо определить все признаки, хотя бы в малейшей мере характеризующие объекты или явления.

Признаки объектов могут быть подразделены на детерминированные, вероятностные, логические и структурные.

Детерминированные признаки - признаки, принимающее конкретные числовые зна-

-12- чения (например, размах крыла $l_{кр}=25$ м, длина фюзеляжа $l_{ф}=50$ м, масса самолета $G = 70$ т. и т.д.), которые могут рассматриваться в качестве координат точки в признаковом пространстве, соответствующих данному объекту. При рассмотрении детерминированных признаков ошибками измерений пренебрегают.

Вероятностные признаки — признаки, случайные решения которых распределены по всем классам объектов, при этом решение о принадлежности распознаваемого объекта к тому или иному классу может приниматься только на основании конкретных значений признаков данного объекта, определенных в результате поведения соответствующих опытов. Признаки распознаваемых объектов следует рассматривать как вероятностные в случае, если измерение их числовых значений производится с такими ошибками, что по результатам измерений невозможно с полной определенностью сказать, какое числовое значение данная величина приняла.

Логические признаки распознаваемых объектов можно рассматривать как элементарные высказывания, принимающие два значения истинности («да», «нет» или «истина», «ложь») с полной определенностью. К логическим признакам относятся прежде всего признаки, не имеющие количественного выражения. Эти признаки представляют собой суждения качественного характера типа наличия или отсутствия некоторых свойств или некоторых элементов у распознаваемых объектов или явлений. В качестве логических признаков можно рассматривать, например, такие симптомы, используемые при медицинской диагностике, как боль в горле, кашель, насморк и т.д., такие свойства объектов геологической разведки, как растворимость или нерастворимость в определенных кислотах или в некоторых смесях кислот, наличие или отсутствие запаха, цвета и т.д. К логическим можно отнести такие признаки, у которых важна не величина признака у распознаваемого объекта, а лишь факт попадания или непадения ее в заданный интервал. В пределах этих интервалов появление различных значений признаков у распознаваемых объектов предполагается равновероятным. На практике логические признаки подобного рода имеют место в таких ситуациях, когда либо

-13-ошибками измерений можно пренебречь, либо интервалы значений признаков выбраны таким образом, что ошибки измерений практически не оказывают влияния на достоверность принимаемых решений относительно попадания измеряемой величины в заданный интервал. Например, в области технической диагностики решение о выходе из строя технических устройств принимается лишь тогда, когда фактические значения некоторых параметров (признаков) превышают заданные интервалы. Отклонение же значений параметров от номинала, не сопровождающееся выходом за пределы соответствующих интервалов, является информацией о том, что устройство функционирует нормально.

Структурные (лингвистические, синтаксические) признаки представляют собой производные элементы (символы) структуры объекта. Иначе эти элементы (константы) называются терминалами. Каждый объект может рассматриваться как цепочка терминалов или как предложение. Эти предложения и описывают объекты. При этом если предложение, описывающее неизвестный распознаваемый объект, относится к языку данного класса, то этот объект и принимается принадлежащим этому классу. Например, при распознавании букв русского алфавита терминалами являются вертикальная, горизонтальная, наклонная черточки, наличие угла и т.д.

Задача 2. Задача заключается в проведении первоначальной классификации распознаваемых объектов или явлений и в составлении априорного алфавита классов. Основное в данной задаче - выбор надлежащего принципа классификации. Последний определяется требованиями, предъявляемыми к системе распознавания, которые, в свою очередь, зависят от того, какие решения могут приниматься системой управления по результатам распознавания неизвестных объектов или явлений. При решении последующих задач априорный алфавит классов уточняется, в результате чего формируется рабочий алфавит классов системы распознавания.

Задача 3. Эта задача состоит в разработке априорного словаря признаков. Словарь разрабатывается на основе результатов решения первой задачи с учетом того, что в априор-

-14-ный словарь признаков включаются только те признаки, относительно которых может быть получена априорная информация, необходимая для описания классов на языке этих признаков.

Задача 4. Задача состоит в описании всех классов из априорного алфавита классов на языке признаков, включенный в априорный словарь признаков. Эта задача не имеет однозначного решения, и в зависимости от объема исходной информации для ее решения могут быть использованы методы непосредственной обработки исходных данных, обучения или самообучения.

1.2 Восприятие устной речи

Один из вопросов, на который ученые до сих пор не могут дать точного ответа - это вопрос о том, как человек распознает речь и какими единицами распознавания он пользуется: фонемой, слогом, словом или более протяженными отрезками речи.

Существуют три основные теории восприятия устной речи: психоакустическая, лингвистическая и моторная.

Психоакустическая теория видит основу восприятия речи в разрешающей способности механизма слуха и оперирует такими параметрами, как интенсивность и частота сигнала. Сторонники психоакустической теории считают, что восприятие речи, скорее всего, основано на абсолютной классификации акустических сигналов, но, возможно, базируется на относительной различимости речевых образов, т. е. на возможности непосредственно сравнивать смежные участки спектра для выявления их фонематических различий.

Лингвистическая теория стремится установить в речи абсолютно воспринимаемые дискретные информационные элементы, жестко связанные со структурой языка (лингвистикой речи). Согласно лингвистической теории речевой сигнал опознается на основе различения лингвистических единиц, определяемых структурой языка, на котором говорят. При этом главную роль играют процессы, протекающие в центральной нервной системе. Способ-

речевых сигналов на лингвистические категории вырабатывается у человека на основе обучения языку и повседневной разговорной практики. Надежность опознавания этих категорий существенно зависит от объема словаря и от ограничений, имеющих в воспринимаемом сообщении.

Моторная теория распознавания речи обращается к артикуляционным (моторным) непрерывным процессам и к их связям с сенсорными (смысловыми) процессами в высших отделах мозга. Согласно этой теории отображение звука в сознании представляет собой процесс воссоздания фонемы на основе ассоциативных представлений об артикуляционных параметрах (командах), относящихся к услышанному звуковому образу. При этом человек следует определенным правилам (так называемым производящим правилам), которые определяют процесс синтеза внутренних слуховых образов и сравнения их с поступающими. В качестве единицы речи здесь выступает фонема, опознаваемая по ее сегментным артикуляционным признакам.

Было также замечено, что при распознавании фонем большое влияние оказывает акустическое окружение, т. е. восприятие той или иной фонемы зависит от соседних фонем. В частности, было установлено, что восприятие согласных зависит от типа последующих гласных. Это навело на мысль, что акустическими единицами при слуховом восприятии могут служить речевые отрезки, по длительности равные слогу или полуслогу. Но однозначная зависимость между звуками и фонемами не была обнаружена, и это не позволило найти для всех случаев распознавания акустические инварианты (неизменяемые акустические единицы).

Речь распознается лучше, если сообщение является предвиденным, заранее ожидаемым. В свою очередь на предсказываемость сообщения, а следовательно, на разборчивость речи, оказывают влияние семантические и грамматические ограничения.

Грамматика языка заранее определяет допустимую последовательность слов и их морфологическую структуру.

Семантические соображения подсказывают слова, пригодные для выражения смыслового содержания.

Овладение сутью сообщения облегчает поиск подходящих слов и улучшает различимость отдельных слов и целых предложений, так как слуховой анализатор может «повременить» с принятием решения по каждому услышанному слову.

Учитывая все эти обстоятельства, можно сделать вывод, что в устном сообщении, представляющем собой грамматически и семантически правильную структуру, слушатель воспринимает целые фразы или еще более протяженные отрезки речи.

Суммируя приведенные сведения, относящиеся к разным теориям восприятия речи, можно сделать вывод, что наиболее вероятным является представление восприятия в виде многоуровневого процесса, в котором процедура восприятия информации подчинена способам выделения сигнала и цели слушания. Слуховой аппарат может использовать накапливающуюся во времени информацию для принятия дискретных решений об опознании тех или иных законченных звуковых конструкций - фонем, слов фраз. Размеры элементов восприятия речи изменяются в зависимости от целей распознавания, а скорость обработки сигналов изменяется в соответствии с типом поступающей информации.

Фонетический состав русской речи

Основной единицей звукового строя языка является фонема. Звуковой состав различных языков имеет свои особенности. В русском языке насчитывают около 40 фонем [38] (это деление весьма условно и может подвергаться сомнениям, как любое деление сложного процесса на части). Один из возможных вариантов классификации фонем приведен на рис. 1.1. Он включает в себя классы фонем, наиболее привычные в практике автоматической обработки речи.

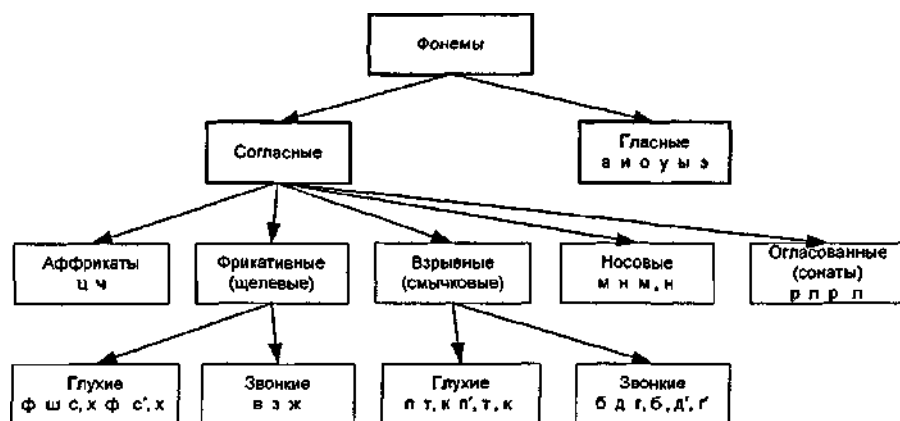


Рис. 1.1. Классификация фонем русского языка

1.3 Общая структура распознающей системы

Из предыдущего параграфа следует, что восприятие речи есть сложный многоуровневый процесс, в котором тесно переплетены различные уровни - от акустического до семантического.

Исходя из этих соображений строится многоуровневая модель восприятия, неопределенность исходного представления речи в которой постоянно снижается за счет закономерностей композиции элементов различных уровней из элементов более низких уровней.

«Перспективная» модель распознавания речи (РР) должна строиться исходя из ряда принципов:

1. Модель должна иметь иерархическую структуру. Это важно для компактности описаний на каждом уровне, для простоты дополнения и развития представлений различных уровней. В общем случае модель может содержать уровни: сигнальный, фонемный, слоговый, словесный, фразовый, смысловой.
2. Взаимодействие уровней обеспечивается за счет системы правил, которые описывают преобразование и сочленение единиц данного уровня в единицы более высокого уровня.
3. Каждый уровень оперирует отрезками речи определенной длительности, т. е. в определенном временном окне анализа. Размер окна увеличивается при переходе к высшим уровням.

4. Процесс многоуровневого восприятия речи состоит в построении иерархического образа речевого сообщения на основе анализа, синтеза и классификации. На начальных уровнях (например, фонемы) классификация ведется путем сравнения параметрических описаний с полным набором эталонных описаний элементов данного класса. На более высоких уровнях используется процедура синтеза гипотез, их оценки, ранжирования и отбора. Иными словами, восприятие речи представляет собой согласование входного сигнала с иерархической системой представлений о речевом сообщении.
5. Необходимая надежность восприятия обеспечивается за счет избыточного представления описаний и результатов на всех промежуточных уровнях. Решения, принятые на этих уровнях, не могут быть окончательными. Запас гипотез на каждом уровне должен быть достаточным для достижения высокой надежности восприятия речи в целом.
6. Для сравнения гипотез должны существовать правила оценки их приемлемости. Число градаций должно быть достаточным для их адекватного ранжирования.
7. Стратегия образования (генерации) гипотез должна сочетать в себе принципы целенаправленного перебора и случайного выбора, поскольку полный перебор иногда фактически невозможен. Фактор случайности связан с неопределенностью исходного описания речи.
8. На каждом уровне классификации оценивается ее качество в виде показателей рассогласования входных и гипотетических описаний; показатели рассогласования различных уровней объединяются, и, объединенный показатель стимулирует генерацию гипотез до тех пор, пока рассогласование не будет снижено до приемлемого уровня. На этой, заключительной стадии совокупность гипотетических описаний всех уровней и является итогом восприятия.
9. Заключительный, прагматический уровень реализуется за счет закономерностей предметной области использования речи. Система восприятия отдает предпочтение той гипотезе о командной фразе, которая лучше согласуется (при прочих равных условиях) с те-

кущей ситуацией. В общем случае гипотеза выбирается на основе взвешенного учета показателей согласованности различных уровней.

Цель, которую ставят в каждом конкретном случае машинного РР, определяет объем распознаваемого словаря (под словарем подразумевается набор фраз, слов, слогов, фонем), необходимую точность опознавания (процент правильно узнанных машиной слов, фонем), скорость выполнения операций (распознавание в реальном или условном масштабе времени).¹

В многоуровневой модели восприятия ее качество (надежность) определяется качеством реализации каждого отдельно взятого уровня, и, чем надежнее работает каждый уровень, тем надежнее работает вся система [1].

Обобщенная структурная схема СРР, включающая в себя все уровни обработки речи, представлена на рис. 1.3.1

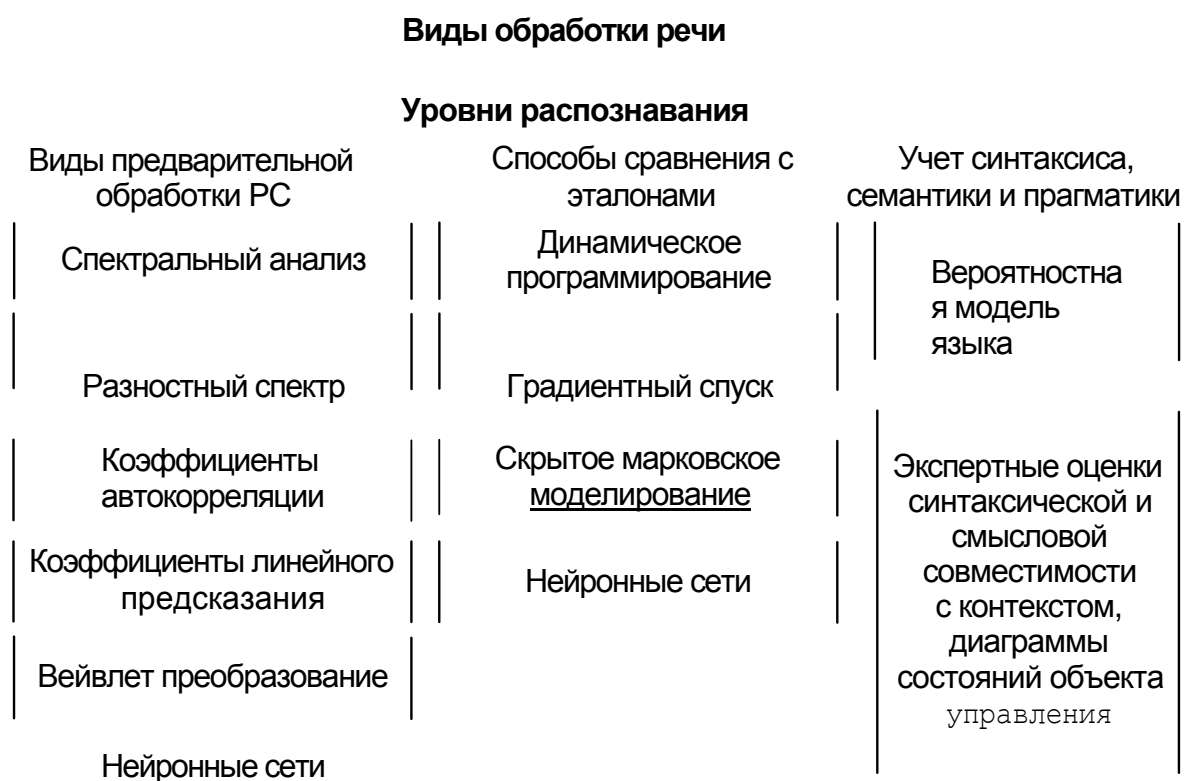


Рис. 1.3.1. Виды обработки речи

1.4 Анализ состояния исследований по проблеме автоматического распознавания речи

Развитие автоматического распознавания речи идет по пути постоянного усложнения методов исследования, и параллельно этому, осознания все новых и новых трудностей, привлечения всех новых сведений из смежных и даже отдаленных областей знаний.

Начальный период исследований был связан в основном с попытками статического описания звуков речи. Предполагалось, что звуки речи являются такими же однородными, стабильными и устойчивыми единицами устной речи, какими в нашем представлении являются буквы письменного текста, и восприятие речи по фонемам можно вести аналогично чтению текста по буквам. Допускалось, в частности, что такое разделение возможно на основе спектральных эталонов отдельных фонем с помощью традиционных методов обработки радиотехнических или других искусственных сигналов. При успешном решении этой проблемы стала бы возможной автоматическая стенография, ввод произвольного текста в ЭВМ, управление различными техническими средствами, многократное сжатие передаваемой по каналам связи речи и многое другое.

Основной упор на этой стадии исследований делали на точность анализа. Число спектральных полос принималось равным, например, 5, 12, 20,..., 120.

Впервые попытка распознавания фонем спектральными методами была сделана Л.Л. Мясниковым. В его приборе речевой сигнал подавался на гребенку полосовых фильтров (18 полос). После детектирования сигналы сравнивались попарно (в определенном порядке) и вырабатывались признаки со значениями +, -, 0. В зависимости от полученной комбинации признаков срабатывал тот или иной решающий элемент (реле), настроенный на определенный звук. Прибор имел два независимых классификатора гласных и согласных звуков. Точность распознавания гласных звуков составила 75 — 80%. Общая точность (по гласным и согласным) оказалась недостаточной для каких-либо практических приложений.

Распознаванию ограниченного подмножества фонем посвящен ряд работ в 1950 — е годы. Из числа наиболее успешных можно назвать работу американца Дж. Форджи. Его ус-

гласных с различных голосов с точностью 93%. Распознавание гласных при этом исследовалось в рамках задачи больших наборов слов. Гласные выделялись по уровню энергии. Их классификация велась с помощью двухступенчатого формантного анализа в сочетании с другими видами анализа (учет основного тона, соотношение энергий в частотных каналах, логический учет нетипичных ситуаций, учет длительностей однородных участков и т. п.).

Более сложная задача была решена Фреем и Динесом. Их установка могла распознавать многие гласные и часть согласных в словах, составленных из этих звуков. Точность распознавания с голоса одного диктора составила 72%. В установке использовался формантный анализ [15].

Кроме спектрального описания стали использовать новые способы кодирования речи, членение речи на фонемы (до этапа классификации), статистические закономерности встречаемости фонем и их сочетаний. Выявилась большая изменчивость фонем, их подверженность влиянию соседних звуков, трудность выделения фонем из потока речи. Выяснилось, что число классов речевых фонем значительно превышает число фонем речи и что эти классы пересекаются.

1.5 Обзор рынка программных средств

В вопросе распознавания речи существует два фундаментальных подхода: распознавание может быть дикторозависимым или дикторонезависимым (диктор — пользователь системы распознавания речи, взаимодействующий с ней в речевом режиме).

Дикторозависимые системы достигают наилучшего качества распознавания речи при настройке на конкретного диктора (пол, акцент, темп речи, интонационные вариации и т.д.) и, в случае смены диктора, требуют перенастройки (обучения), давая, в противном случае, ухудшение качества распознавания. Дикторонезависимые системы способны с хорошим качеством распознавать речь любого пользователя, работающего с ней, без дополнительной настройки (обучения). Первоначальное обучение производится разработчиком такой систе-

-22-мы и, в идеале, в дальнейшем не требует корректировки. Такие системы обладают малой чувствительностью к изменениям параметров речевого тракта, обусловленных индивидуальными характеристиками речи диктора.

Л»

По области применения все программные продукты для распознавания речи делятся на три основных типа: навигаторы, средства речевого ввода текста и средства разработки. **1.1.**

Навигаторы (броузеры)

Подобные программы поставляются на рынок уже более десяти лет. Броузеры позволяют, отдавая устные команды, управлять работой системы, например, запуском и исполнением прикладных программ. Однако в последнее время «чистые» навигаторы превратились не более чем в забавные игрушки, поскольку практически все программы речевого ввода текста содержат в себе и возможности навигации по компьютеру. Опишу одну подобную программу, которая выполняет только функции навигации.

Программа Listen for Windows

Программа Listen for Windows - независимый от диктора речевой навигатор, выпущенный фирмой Verbex Voice System. Теоретически независимость от диктора позволяет исключить предварительное обучение программы распознаванию команд, произносимых конкретными пользователями. Однако уровень точности распознавания системы Listen for Windows, так же как и других речевых навигаторов, при обучении значительно повышается.

В отличие от большинства других навигаторов Listen for Windows может непосредственно работать с множеством распространенных прикладных программ Windows, включая AutoCAD, Microsoft Office и другие, тогда как остальным навигаторам требуется обучение командам, используемым в меню для каждой прикладной программы, которую необходимо

-23-запустить. Кроме того, в отличие от других браузеров, Listen for Windows предоставляет возможность эффективных переходов, позволяющих пропускать несколько уровней вложенных меню.

В пакете Listen for Windows используются две модели речи - для мужчин и для женщин. В исходном состоянии программа распознает мужскую речь лучше женской, но после обучения она начинает распознавать их одинаково хорошо.

1.2. Программы речевого ввода текста

Появление таких программ на рынке программного обеспечения относится примерно в 1993 году. Их возможности позволяют с помощью голоса создавать текстовые документы, вводить числа в электронные таблицы, осуществлять сеансы связи в реальном масштабе времени. Но поскольку системы речевого ввода текста должны оперировать существенно большим объемом слов, чем, например, программы-навигаторы, которым требуется лишь

*

распознать команды меню, продвижение этих систем на рынок происходит медленнее, несмотря на то, что их потенциал очень и очень велик. Задача распознавания речи при диктовке настолько сложна, что предъявляет существенные аппаратные требования к компьютеру. По этой причине разработчики таких пакетов, как правило, требуют обязательного использования плат специализированных сопроцессоров. Кратко опишу некоторые программные продукты этого типа.

1.2.1. Программа Dragon Dictate for Windows

Система Dragon Dictate (далее — DD) является зависимой от диктора и требует раздельного произношения слов.

DD сочетает в себе систему речевого ввода текста, которая может работать с любой

-24-прикладной программой ОС Windows, и вполне приемлемый навигатор, который обеспечивает гибкое речевое управление большинством прикладных программ в рамках Windows, а также может быть обучен выполнению новых команд. DD позволяет непосредственно вводить текст с голоса при работе со многими наиболее распространенными текстовыми процессорами.

При работе с DD обучение обязательно для каждого пользователя (диктора). Процесс обучения требует, чтобы пользователь произнес в ответ на подсказки примерно 750 отдельных слов, что занимает около 20 минут. После завершения этого процесса распознавание осуществляется достаточно точно. Речевые и грамматические модели DD достаточно «умные», чтобы выявлять использование обычных слов в качестве части собственного имени и в соответствии с этим писать их с прописной буквы. Кроме того, программа способна обучаться новым словам с первого предъявления.

Текущее распознаваемое программой DD слово выводится в прямоугольник окна речевого ввода. Можно сообщить системе об ошибке возгласом «OOPS», что приводит к появлению экрана коррекции, на котором отображается пронумерованный список последних пяти произнесенных слов. На неправильно распознанное системой слово надо указать голосом, называя соответствующий ему номер в списке. При этом на экране появится второй список, содержащий варианты предлагаемых исправлений. Можно выбрать правильный вариант замены из этого списка или просто ввести правильное слово вручную с клавиатуры. Такая система позволяет просто и быстро выполнять и ввод, и исправления.

Для программы DD существует русский модуль «Горыныч», при использовании которого появляется возможность диктовать как на английском, так и на русском языке.

1.2.2. Программа Kurzweil Voice for Windows

Фирма Kurzweil Applied Intelligence была пионером современных исследований в об-

начала поставки удачных программных продуктов для распознавания речи. Kurzweil Voice (далее - KV) может воспринимать текст с голоса после очень небольшого обучения, обеспечивая хороший уровень распознавания. Фирма Kurzweil утверждает, что ее пакет обеспечивает достоверность в 85% без предварительного обучения и эта достоверность быстро улучшается со временем, по мере того как программы обучаются индивидуальным особенностям речи каждого пользователя.

В отличие от DD для пакета KV требует специализированная звуковая плата.

Пакет готов к работе сразу после аппаратно-программной инсталляции и указания пола диктора. При продолжительном использовании программы обучаются более точно распознавать голос каждого пользователя, выбирая образцы для исправлений, внесенных во время предыдущих сеансов речевого ввода.

1 2.3. Программа IBM Personal Dictation System

Система IBM Personal Dictation System (далее - IPDS) предназначена для операционной системы OS/2; эта система, зависящая от диктора, который должен произносить слова отдельно, поставляется вместе со специализированной платой речевого процессора. IPDS обеспечивает весьма высокую точность устного ввода текста.

По утверждениям самой IBM, IPDS была разработана для прикладных программ, требующих высокого уровня достоверности. К ее недостаткам можно отнести большие затраты на обучение, требующиеся для каждого нового пользователя. После аппаратно-программной инсталляции пакета необходимо надиктовать два текста - короткий рассказ Марка Твена и описание техники распознавания речи. После этого процесса, занимающего не менее полутора часов, IPDS производит формирование баз данных, на что уходит еще несколько часов интенсивной работы

В отличие от программы DD, которая позволяет устно вводить текст в любой тексто-

-26-ый процессор, полученный от независимых поставщиков, IPDS загружает диктуемый текст в специализированный текстовый редактор, поставляемый вместе с этой программой. Первоначальное редактирование помогает программе IPDS изучить особенности речи каждого говорящего. Однако текстовый редактор IPDS не умеет делать почти ничего, кроме формирования текста и исправления выявляемых ошибок.

Несмотря на трудности, связанные с инсталляцией и обучением, IPDS обладает хорошими возможностями для распознавания речи. Язык IPDS и ее модели контекстов обладают требуемой универсальностью, а словарь объемом в 24000 слов вполне достаточен. В процессе программного контроля речевого ввода можно видеть, как IPDS отслеживает произнесенные слова, строя наилучшие догадки о значении каждого. По мере продолжения речевого ввода IPDS может пересмотреть принятое ею значение предыдущего слова, основываясь на новом контексте. Этим достигается высокий уровень точности.

1.3. Средства разработки

Средства разработки позволяют создавать как универсальные, так и специализированные прикладные программы, в которых используются методы распознавания речи. В основном эти системы ориентируются на программистов, создающих программы на языках C++ и Microsoft Visual Basic, и содержат базовые программы распознавания, а также интерфейсы прикладных программ (API) и необходимые для их использования библиотеки языков программирования.

1.3.1. Пакет программ IBM Continuous Speech Series

IBM Continuous Speech Series (далее - ICSS) представляет собой систему разработки, которая предоставляет инструментарий для создания прикладных программ, управляемых голосом, и позволяет устно запускать прикладные программы в средах Windows и OS/2. Эта система не требует ни предварительного обучения, ни отдельного произнесения слов.

ICSS поставляется со словарем объемом 25000 слов. Распознаваемые ICSS слова подразделяются на контексты, каждый из которых содержит около 1000 слов. Контексты представляют собой совокупности слов, связанных между собой по смыслу, которые, по предположению ICSS, могут быть произнесены вместе. ICSS может сохранять в памяти одновременно несколько контекстов и при необходимости осуществлять переключения между ними. В состав пакета ICSS входят несколько программ на языке C++, предназначенные для Windows или OS/2, которые используют информацию от звуковой платы для распознавания речи.

Средства разработки ICSS совместимы с языками программирования C++, MS Visual Basic и другими, позволяющими работать с библиотеками языка C++. В состав ICSS входят несколько прикладных программ, демонстрирующих возможности пакета, а также исходные тексты этих программ в качестве примеров программирования.

1.3 2. Пакет программ Phonetic Engine 500 Speech Recognition System

Phonetic Engine 500 (далее - PE500) - инструментальный комплект разработки программ, созданный фирмой Speech Systems в расчете на тех, кто создает узкоспециализированные прикладные программы, способные распознавать речь. Подобно системе разработки ICSS, комплект PE500 позволяет создавать прикладные программы, которые не требуют обучения, и не вынуждают пользователя произносить слова отдельно. В целом данный комплект представляет собой достаточно мощный инструмент для подготовки на языках C++ и MS Visual Basic прикладных программ, способных распознавать речь; он совместим с любыми прикладными программами Windows, которые могут обращаться к функциям динамически подключаемых библиотек, включая базы данных, как, например, MS Access.

В основе комплекта лежит независимая от диктора программа распознавания слитной речи, а также мощный и хорошо документированный интерфейс API. В PE500 включены

-28-также простая среда разработки, библиотека динамической компоновки Windows, средства для вызова модулей из языка Visual Basic, примеры исходных текстов программ, которые показывают, как составлять разнообразные запускаемые голосом прикладные программы для среды Windows, а так же модуль для проверки универсальности и пределов возможностей программного блока, служащего для обработки речи, в создаваемых программах.

В комплект PE500 входит плата обработки речи, пакет программ и гарнитура.

1.6 Сравнительный анализ методов выделения признаков речевых сигналов

Основные, традиционные задачи теории распознавания образов - это выбор информативных признаков, выбор решающих функций, предварительная классификация объектов (таксономия).

Выбор признаков в значительной мере влияет на качество алгоритмов распознавания. Из множества различных характеристик объектов (полученных непосредственными измерениями или какими - либо преобразованиями исходных описаний) надо выбрать наиболее существенные, которые позволяют достаточно компактно описать реализации образа и в то же время достаточно точно различать объекты [Косарев ЮА].

Считается, что наиболее перспективны такие описания, которые адекватны процессу речеобразования (например, формантное или по методу коэффициентов линейного предсказания (КЛП)) или слухового восприятия.

Важный и принципиальный вопрос, который явно или неявно присутствует при выборе системы признаков и их реализации в виде решающих функций, - это предельная степень их компактности.

До сих пор элементы речи не поддавались простой многоступенчатой классификации, поскольку до сих пор не удавалось найти наборы признаков, которые бы позволили последовательно классифицировать неизвестный звук, относя его по признакам к той или иной груп-

подгруппе, подподгруппе и т. д.

Авторы идеи дифференциальных признаков [1] исходили из предположения, что существует некоторый набор независимых бинарных признаков фонем, которые позволяют методами строгой бинарной классификации разделить все множество речевых звуков. Порочность этой идеи состоит в ее противоречии с избыточным характером речи на всех ее уровнях, в исходной неопределенности речи. Отсутствие реальных успехов в поисках дифференциальных признаков речи подтверждает бесперспективность данного направления.

1.6.1 Анализ сигнала возбуждения голосового тракта

Голосовой тракт человека может возбудиться шумовым сигналом или квазипериодическими импульсами. При распознавании речи знание параметров сигнала возбуждения очень важно, так как оно позволяет сегментировать звуковой поток на вокализованные и не-вокализованные звуки, т.е. определить, является ли сигнал возбуждения квазипериодическими импульсами (основным тоном) или шумовым сигналом. Более того, анализ, синхронный с частотой основного тона, позволяет получить стабильное описание по сравнению с фиксированной частотой анализа. Частотный диапазон основного тона располагается в интервале от 100 Гц до 1000 Гц, энергия вокализованных звуков намного превосходит энергию невокализованных, кроме того, отсчеты вокализованных звуков обладают гораздо более сильной взаимной корреляцией. Перечисленные свойства лежат в основе методов определения возбуждения голосового тракта.

Простейшим способом определения параметров возбуждения является анализ сигнала с микрофона, укрепленного на горле говорящего (т.н. ларингофон). При наличии основного тона будут зафиксированы импульсы, частота которых совпадает с частотой основного тона. Данный метод, несмотря на простоту, имеет явный недостаток, связанный с необходимостью использования ларингофона.

Существует простой метод обнаружения основного тона, в основе которого лежит

-30-измерение отношения энергии в полосе частот основного тона к полной энергии речевого сигнала. Если энергия достаточно велика и соотношение приближается к единице, то принимается решение о наличии основного тона.

1.6.2 Анализ клипированного сигнала

Операция ограничения сигнала называется клиппированием и является наиболее простым методом формирования признаков речевого сигнала. Метод реализуется путем подсчета числа переходов через нулевой уровень в единицу времени либо анализом распределений интервалов времени между нуль-переходами. В качестве исходного сигнала используется речевой сигнал или его производная. Число нуль-переходов, усредненных за 10-30 мс, характеризует максимум энергетического спектра, а анализ асимметрии интервалов между нуль-переходами позволяет определить степень шумности звука.

Важным достоинством данного метода является простота реализации и удобство анализа в цифровой форме. С другой стороны метод осуществляет довольно грубый анализ формы сигнала, что затрудняет качественное распознавание при значительных словарях (более 100 слов).

1.6.3 Формантный анализ

Речевой сигнал представляет собой реакцию системы голосового тракта на возбуждение звуковыми волнами. Резонаторы образуются полостями рта, глотки и в ряде случаев носовой полостью. При рассмотрении спектральных функций различных звуков речи можно выделить несколько групп составляющих, разделенных частотными интервалами, в пределах которых составляющие спектра или вовсе отсутствуют, или имеют значительно меньшие

составляющих в соседних группах. Области спектральных максимумов носят название формант. Ряд исследователей под формантами понимает только те спектральные максимумы, которые в той или иной степени определяют распознавание конкретных звуков речи.

Основными параметрами форманты являются центральная частота, ширина и ее уровень. В формантных анализаторах речи спектральная огибающая речевого сигнала аппроксимируется путем комбинации нескольких простых резонансных кривых. Формантами в полной мере характеризуются гласные звуки и вокализованные согласные, а для глухих согласных часто используются различного порядка моменты спектра.

Обычно анализируется не более четырех формант. Исследователи по-разному указывают частотный диапазон формант, например имеются такие данные: первая форманта - от 200 до 850 Гц, вторая - от 850 до 2350 Гц, третья - от 2100 до 3000 Гц. Уровни формант, особенно высших, коррелированы в довольно значительной степени. В среднем коэффициент корреляции уровней третьей и четвертой формант составляет 0.96, а первой и второй — 0.77. Это означает, что фактически передача уровней двух формант не обязательна.

Ряд исследований показал неточность оценки большинства согласных звуков по формантным частотам, определяемым как частоты максимумов спектральной функции, поэтому был предложен метод оценки спектров согласных по их моментам первого, второго и третьего порядков, определяемых следующим образом:

$$n \quad \quad \quad n \quad \quad \quad n$$

где f_n - частота n -го фильтра спектрального анализатора; A_n - интенсивность сигнала на входе n -го фильтра.

Несмотря на видимую простоту анализа и хорошее отражение физики речеобразования, формантный анализ не нашел широкого распространения в блоках предварительной обработки. Это, в частности, связано с частой неопределенностью фиксации формантных экстремумов на спектральной функции.

1.6.4 Спектральный анализ

Предварительная обработка методом спектрального анализа получила в настоящее время наиболее широкое распространение. В ходе анализа с периодом 10-30 мс определяется спектральный срез речевого сигнала. Такое описание характеризует и спектр источника возбуждения, и параметры органов артикуляции. Основными характеристиками предварительной обработки методом спектрального анализа являются число фильтров, полоса пропускания и метод сжатия (упаковки), если таковая имеется. Диапазоны анализируемых частот находятся в интервале от 50 до 10000 Гц (наиболее широкая граница) и от 200 до 3000 Гц (узкая граница). Частотный диапазон наиболее часто делят на полосы с равной формантной разборчивостью. Полосами с равной формантной разборчивостью называют полосы частот, в равной степени влияющие на разборчивость речи. Они определяются экспериментально в ходе аудиторных испытаний.

Число каналов фильтрации варьируется в очень широких пределах (от 5 до 120) и обычно уточняется экспериментально.

Существуют различные методы получения спектра речевого сигнала, к которым относятся:

- фильтрация аналоговыми методами;
- цифровая фильтрация;
- вычисление спектра через коэффициенты автокорреляционной функции (АКФ);
- авторегрессионный спектр.

Наиболее простым методом для реализации является аналоговая фильтрация. Для измерения амплитуды сигнала на выходе фильтра осуществляется операция детектирования, включающая в себя определение модуля величины речевого сигнала и последующее сглаживание.

Цифровые методы фильтрации (фильтры, БПФ) чаще используются в исследовательских системах нереального масштаба времени, так как эти методы требуют больших вычислительных ресурсов. В реальном времени они реализуются только при помощи специализированных вычислителей (ЦПОС) или многопроцессорных систем. Кроме метода БПФ, спектр может быть рассчитан через автокорреляционную функцию и коэффициенты авто-регрессии, что также требует больших вычислительных ресурсов.

Вектор параметров, представляющих собой амплитуду сигнала в каналах фильтрации, представляет собой очень информативную систему признаков, но из-за сильной флуктуации амплитуды речевого сигнала значения спектральных составляющих колеблются.

Для уменьшения влияния вариаций речевого сигнала на спектральное описание применяют сглаживание группы отсчетов и нормирование спектра (по средней энергии, площади спектральной функции и т.д.). В настоящее время неясно, по какому критерию лучше всего производить нормировку спектра. Однако существуют преобразования сигнала, равноценные нормализации уровня, например, выделение знака первой и второй производной.

К спектральным методам обработки речи можно отнести метод согласованной фильтрации сигнала, широко используемый в радиолокации при обнаружении объектов определенной формы. Принцип согласованной фильтрации заключается в следующем. Специальными методами создается фильтр, АЧХ которого имеет коэффициент передачи, соответствующий комплексно-сопряженному спектру ожидаемого сигнала (фонеме). Тогда, если на вход фильтра подавать сигналы всевозможных форм, но с равной энергией, то максимальная амплитуда отклика будет наблюдаться только при обработке ожидаемого сигнала.

Результаты исследований показали невозможность различать фонемы непосредственно по их спектрам [14, с. 12]. Опыты по установлению соответствия фонетических символов и спектральных распределений привели к устойчивому отрицательному результату [14, с. 13].

1.6.5 Корреляционный анализ

Автокорреляционная функция (АКФ), наряду со спектральной функцией, несет полную информацию о сигнале. АКФ характеризует взаимосвязь соседних отсчетов речевого сигнала. АКФ обычно рассчитывают не для всего речевого образа, а на отрезке, не превышающем 10-30 мс, на более протяженных интервалах корреляционные свойства речевого сигнала выражены слабо.

Значение АКФ для нулевой величины задержки равно энергии сигнала. АКФ позволяет определить взаимосвязь между соседними отсчетами амплитудно-временной функции. Если амплитудный сигнал является квазипериодическим, например, гласным звуков, то АКФ характеризуется плавным уменьшением по мере увеличения задержки, а при величине задержки, кратной периоду основного тона, АКФ имеет значительные пики амплитуды.

АКФ шумоподобных звуков меняет свою величину от максимума почти до нуля уже при незначительной величине задержки.

Таким образом, АКФ характеризует и шумность звуков, позволяет выделить и измерить величину периода основного тона.

1.6.6 Скрытое марковское моделирование

Марковские модели являются мощным средством моделирования различных процессов и распознавания образов. По своей природе Марковские модели позволяют учитывать непосредственно пространственно-временные характеристики сигналов, и поэтому получили широкое применение в распознавании речи. Каждая модель представляет собой набор N состояний, между которыми возможны переходы. В каждый момент времени система находится в строго определённом состоянии. В наиболее распространённых Марковских моделях первого порядка полагается, что следующее состояние зависит только от текущего состояния. При переходе в каждое состояние генерируется наблюдаемый символ, который соответствует физическому сигналу с выхода моделируемой системы. Выход, генерируемый моде-

-35-лю, может быть так же непрерывным.

Существуют так же модели, в которых набор символов для всех состояний одинаков. Символ в состоянии в момент времени t генерируется с вероятностью p . Набор всех таких вероятностей составляет матрицу. Матрица определяет вероятность перехода из одного состояния в другое состояние. Считается, что A не зависит от времени. Обычно в реальных процессах последовательность состояний является скрытой от наблюдения и остаётся неизвестной, а известен только выход системы, поэтому такие модели называют скрытыми Марковскими моделями.

В распознавании образов скрытые Марковские модели применяются следующим образом. Каждому классу / соответствует своя модель. Распознаваемый речевой сигнал представляется в виде последовательности наблюдений O . Затем для каждой модели вычисляется вероятность того, что эта последовательность могла быть сгенерирована именно этой моделью. Модель, получившая наибольшую вероятность, считается наиболее подходящей, и образ относят к классу j .

Для того, чтобы сократить вычисления, в распознавании речи используются линейные модели. В таких моделях каждое состояние имеет только одно последующее, так же переход возможен обратно в то же состояние. Такие модели учитывают временные характеристики речевого сигнала: определённый порядок следования участков сигнала, их взаимное расположение, возможность локальных растяжений или сжатий.

/. 6.7 Вейвлет — преобразование Применение вейвлетов в задачах обработки и распознавания речи продиктовано особенностями речевого акустического сигнала. Вейвлеты, как средство многомасштабного анализа позволяют выделять, одновременно как основные характеристики сигнала, так и короткоживущие высокочастотные явления в речевом сигнале. Это свойство является существенным преимуществом в задачах обработки речевого сигнала по сравнению с оконным преобразованием Фурье, где, варьируя ширину окна, приходится выбирать масштаб явлений, которые необходимо выделить в сигнале.

Получение дополнительной информации с разных масштабов времени и разных масштабов разрешения сигнала может улучшить точность распознавания речи. Кроме того, считается, что человеческое ухо устроено так, что при обработке звукового сигнала, оно передает мозгу вейвлет-образ сигнала. Колебания амплитуды давления передаются от барабанных перепонки на мембрану и далее распространяются по всей длине завитка внутреннего уха. Завиток скручен в виде спирали во внутреннем ухе. Если представить, что завиток распрямлён в некоторый сегмент, а вместе с ним и распрямлена мембрана, то можно показать, что результирующее преобразование сигнала будет с точностью до константы совпадать с вейвлет-преобразованием.

Вейвлет-анализ - это исследование сигнала $f(t)$ при помощи разложения по системе базисных функций. Сигнал интерпретируется, как функция из $L^2(\mathbb{R})$, а в качестве базиса используется система функций $\psi_{a,b}(t)$, занумерованных не целыми числами, а двумя непрерывными параметрами. Эта система получается из фиксированной функции всевозможными сдвигами и растяжениями. Функция $\psi(t)$ называется вейвлетом, если:

1. $\psi(t)$ непрерывна;
2. $\psi(t)$ интегрируема на всей прямой;
3. $\int_{-\infty}^{\infty} t \psi(t) dt = 0$

Вейвлет-преобразованием называется функция двух переменных:

$$W(f)(a,b) = \int_{-\infty}^{\infty} f(t) \psi_{a,b}(t) dt$$

В отличие от традиционного преобразования Фурье, вейвлет-преобразование определено неоднозначно: каждому вейвлету соответствует свое преобразование. В качестве анализирующих вейвлетов обычно выбираются функции, хорошо локализованные также и в «пространственной области».

На каждом частотном уровне количество центров частотно-временной локализации в два раза меньше, чем на предыдущем уровне, частота которого выше. Существует корреля-

-37-ция между вейвлет коэффициентами как по шкале времени, так и по шкале частот. Такое вейвлет-преобразование, называют непрерывным. Его выполнение требует больших вычислительных затрат. Желательно было бы иметь разложение с вейвлетами в качестве ортогональных базисных функций. Такие вейвлеты существуют, и очень полезны в задачах сжатия информации и подавления шумов. Схема построения таких вейвлетов связана с исчерпанием пространства сигналов системой вложенных подпространств, отличающихся друг от друга только перемасштабированием независимой переменной. Такая система называется многомасштабным анализом.

1.6.8 Линейное предсказание

Речевой сигнал, генерируемый биологической системой, обладает свойствами случайного нестационарного процесса. Для описания процессов такого типа в последнее время часто применяются модели динамических систем, параметры которых оцениваются на каждом из примыкающих друг к другу промежутков времени длительностью 10-30 мс. Положительными факторами описания РС в динамических системах являются:

- хорошее представление априорных сведений о РС и достаточно точная аппроксимация сигнала возбуждения голосового тракта
- возможность описания нестационарными процессами
- алгоритмы обработки, полученные для данной системы, удобны для реализации на ЭВМ
- Параметрами описания являются коэффициенты линейного предсказания или коэффициенты частной корреляции, охватывающие характеристики РС в параметрах голосового тракта.

Существенным отличием методов на основе ЛП является то, что он позволяет четко разделить параметры голосового тракта и сигнала возбуждения.

-38- Таблица 1. Сравнительный анализ методов

распознавания

Основные параметры	Элементная база	Основные виды обработки	Прим.
10 изолированных слов с любого голоса,	ЭВМ, аппаратные средства	Признаки на уровне слов	Обучение
200 изолированных слов, точность 95%, конкретный голос	Тоже	Спектральный анализ, динамическое программирование, признаки на уровне слов	Обучение
200 изолированных слов, точность 99%, конкретный голос	ПЭВМ, ЦПОС	КЛП-анализ, динамическое программирование	Обучение
Слитная речь из словаря 20-30 слов	ЦПОС	Цифровая фильтрация, динамическое программирование	Обучение
1000 изолированных слов с конкретного голоса,	ПЭВМ, специализированные речевые процессоры	То же + фонемный уровень распознавания	Обучение
5000 изолированных слов с конкретного голоса, точность 95%	ПЭВМ, ЦПОС	КЛП-анализ, членение на фонемы, фонемный, синтаксический и смысловой уровни распознавания	Отставание от реального времени в 50 раз, обучение

Большинство существующих на сегодняшний день моделей СРР, ориентированных на работу с большими (неограниченными) словарями, имеют в своей структуре акустико - фонетический уровень распознавания, а анализ речевого сигнала проводится методами линейного предсказания (табл. 1).

1.7 Выводы

1. Произведена классификация современных систем распознавания и выделены основные задачи, возникающие при их построении. Очерчен круг задач, возникающих при построении систем распознавания речи (СРР).
2. Реальный масштаб времени, дикторонезависимость и высокая надежность (минимальный процент ошибки распознавания) определены как ключевые в контексте исследования, проводимого в диссертационной работе.
3. Проведен обзор существующих СРР с оценкой их возможностей, что позволяет утверждать, что ни одна из СРР на сегодняшний день не позволяет распознавать слитную речь в реальном масштабе времени и независимо от диктора. Реализация таких систем требует больших вычислительных ресурсов, что делает актуальной задачу минимизации признаков описаний речевого сигнала.
4. Реализация СРР, работающей с большим (или неограниченным) словарем, требует НаЛИ-
чия в структуре СРР фонетического уровня.
5. Проведен анализ существующих моделей СРР и определены и обоснованы возможные направления исследований, направленных на достижение надежности СРР. Практически каждая модель СРР включает обязательно акустическую и лингвистическую составляющие. Лингвистическая часть задачи на настоящий день решена удовлетворительно, а проблемным участком является часть общей модели СРР между речевым сигналом на входе системы и условным входом лингвистического блока.
6. Акустическая модель вносит наибольший вклад в СРР и определяет дальнейшие алгоритмы обработки и качество результатов распознавания. Входящий речевой поток попадает на блоки предварительной обработки, где определяются признаки РС. Качество признаков имеет значительное влияние на дальнейший ход и результат распознавания.

РАБОЧЕГО СЛОВАРЯ ПРИЗНАКОВ ФОНЕМ

Линейное предсказание - это перспективное направление обработки речевых сигналов, позволяющее создавать системы связи с повышенной эффективностью, устройств распознавания и синтеза речи.

Методы и средства построения систем эффективной передачи телефонных сообщений впервые были широко освещены в монографии М.А. Сапожкова «Речевой сигнал в кибернетике и связи», вслед за которой была издана за рубежом и переведена на русский язык монография Д. Фланагана «Анализ, синтез и восприятие речи» (1968 г). Вскоре была издана еще одна монография группы авторов «Вокодерная телефония. Методы и проблемы» под редакцией А.А. Пирогова (1974 г.). Эти книги явились обобщением результатов плодотворной многолетней работы отечественных и зарубежных ученых по созданию вокодерных систем, систем распознавания РС, устройств автоматического синтеза речи и т. д., в основу которых были положены идеи спектрального анализа сигналов.

Бурный прогресс в технологии изготовления элементов цифровой микросхемотехники привел к необходимости создания новых методов описания речевых сигналов, более удобных для реализации на цифровой элементной базе и, следовательно, допускающих использование широких возможностей цифровой обработки сигналов при построении вокодерных систем, устройств распознавания речи, синтеза речи и т. п. Одним из эффективных подходов к решению этой задачи является применение методов теории фильтрации и предсказания случайных процессов.

Идеи линейной фильтрации и предсказания случайных процессов развивались независимо А.Н. Колмогоровым и Н. Винером. Первоначально эти идеи нашли применение при синтезе систем автоматического управления, но позднее принципы линейного предсказания стали применяться и в системах связи для эффективного кодирования и в задачах анализа временных рядов в экономике, геофизике, биотелеметрии.

Первой работой, в которой была высказана мысль о применении математического ап-

парата фильтрации (предсказания) случайных процессов к речевым сигналам является статья Н.Н. Акинфиева «К вопросу построения теории речевых сообщений» (1957 г.).

Математический анализ поведения динамических систем общего типа стал рассматриваться в качестве самостоятельной области исследований в начале 20-го века. Особенно интенсивно исследования в этой области начали проводиться в 50-х годах с появлением электронных вычислительных машин. Анализ процессов на выходе динамических систем стал центральной проблемой в «анализе временных рядов», методы которого были разработаны главным образом при решении задач статистики, эконометрики и связи. Фактически большая часть исследований в области анализа временных рядов была выполнена специалистами по математической статистике.

Для количественного описания процессов формирования и передачи РС было предложено много различных моделей. Можно, однако, с уверенностью сказать, что ни одна из них не позволяет объяснить все характеристики человеческой речи. Такая модель, в силу своей сложности, вероятно и не потребуется. Во всех практических случаях при выборе математической модели стремятся обеспечить ее минимальную сложность при максимальной точности. Важным достоинством модели на основе линейного предсказания является относительная простота оценки параметров, использующей линейные процедуры обработки сигнала.

2.1 Оценка параметров речевого сигнала

Речь представляет собой непрерывно изменяющийся во времени процесс, кроме того, голосовая щель связана с голосовым трактом, что обуславливает нелинейный характер механизма речеобразования. Однако при некоторых упрощениях можно разработать адекватные линейные на коротких временных интервалах модели для описания речи [11].

На рис.1 представлена часто используемая модель речевого сигнала. Основным элементом здесь является фильтр, возбуждаемый либо квазипериодической последовательностью импульсов, либо случайным процессом. Периодический источник имитирует звонкие

-42-звуки, такие, как гласные и носовые, а

шумовой источник - глухие или фрикативные звуки.

От параметров фильтра зависит идентичность (спектральные характеристики) отдельного звука для каждого из названных способов возбуждения.

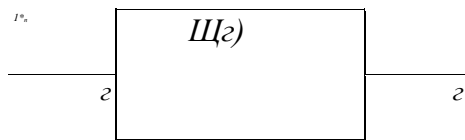


Рис.2.1.1. Дискретная модель речевого сигнала Для применения методов анализа временных рядов к решению задачи распознавания

РС необходимо осуществить дискретизацию рассматриваемых сигналов. Значения каждого сигнала $s(t)$ с непрерывным параметром t отсчитываются в дискретные моменты времени с периодом T . В результате получается сигнал $s(nT)$ с дискретным параметром T , где n — целочисленная переменная. Такие сигналы называются также временными рядами. Частота отсчетов определяется соотношением $f = 1/T$. В дальнейшем с целью упрощения вместо $s(nT)$ будет использоваться s_n .

В настоящее время на первое место вышли задачи моделирования систем. Это связано с тем, что в тех случаях, когда удастся построить достаточно точную параметрическую модель некоторого сигнала, она может быть использована в самых различных случаях, например для предсказания или прогнозирования, управления и сжатия данных.

Одна из наиболее общих моделей сигнала s_n , которая широко используется в настоящее время, может быть представлена с помощью следующего соотношения:

$$s_n = \sum_{i=0}^{p-1} a_i^* s_{n-i} + \sum_{l=0}^{q-1} b_l u_{n-l} + G u_n \quad (1)$$

Такой сигнал обычно интерпретируют как процесс на выходе некоторой системы, на вход которой подается ненаблюдаемая последовательность u_n . Коэффициенты a^* , $0 \leq k < p$, b_l , $0 \leq l < q$, и усиление G — параметры этой гипотетической системы. «Выход» s_n есть линейная функция от прошлых значений выходной последовательности и настоящего и прошлых значений входной последовательности. Это означает, что значение сигнала s_n мо-

-43-жет быть предсказано исходя из линейной комбинации прошлых значений "входа" и "выхода". Такая операция называется линейным предсказанием.

Для левой и правой частей равенства (1) можно вычислить z - преобразование. В результате получим соотношение, являющееся аналогом (1) в частотной области, где $H(z)$ - функция передачи системы, подобной представленной на рис. 1,

$$H(z) = \frac{1 + \sum_{k=1}^p a_k z^{-k}}{1 + \sum_{k=1}^q b_k z^{-k}} \quad (2)$$

где

$$a_k = \lim_{T \rightarrow 0} \Delta t^k \quad (3)$$

есть z - преобразование от s , а $U(z)$ - z - преобразование от u .

Нулями и полюсами этой модели являются соответственно корни полиномов числителя и знаменателя отношения (2).

Наиболее интересными являются два частных случая этой модели:

1. модель, имеющая только нули: $a^* = 0, 1 < k < p$;
2. модель, имеющая только полюсы: $b_k = 0, 1 < l < q$;

В работах по математической статистике первая из этих моделей обычно называется моделью со скользящим усреднением, а вторая - авторегрессионной (АР) моделью.

Наиболее широко применяемой в настоящее время является модель, имеющая только полюсы.

Вычисление коэффициентов линейного предсказания

Сигнал s_n можно представить в виде следующей линейной комбинации прошлых значений этого сигнала и некоторого входного воздействия u_n :

$$s_n = -T \sum_{k=1}^q k^s - k^{+Gu} \quad (4)$$

Где G - коэффициент усиления. Такой сигнал можно сформировать с помощью сле-

темы, изображенной на рис. 2 в частотной и временной областях.

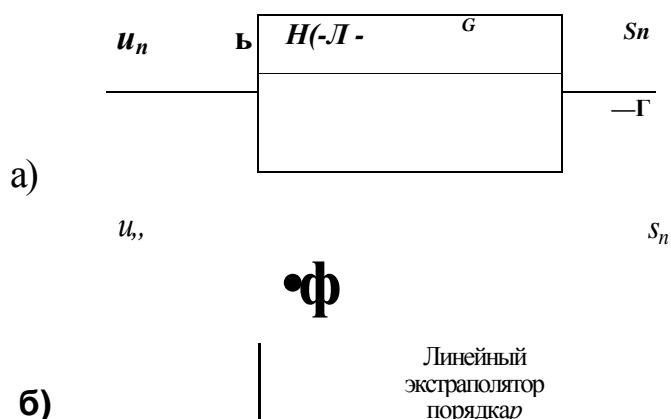


Рис. 2.1.2. представление дискретной модели, имеющей только полюсы: а - в частотной области, б - во временной области

Передаточная функция (2) для рассматриваемого здесь частного случая принимает вид

$$H(z) = \frac{1}{z^p} = z^{-p} \quad (5)$$

и имеет только полюсы. Задаваясь частными значениями сигнала s_n , сводим задачу к определению коэффициентов u_n G модели (4). В частном случае, когда порядок многочлена в числителе равен нулю, определение параметров модели по критерию минимальной средней квадратической ошибки сводится к решению системы линейных уравнений. При этом минимизируется ошибка, возникающая в процессе обратной фильтрации.

Будем считать, что входное воздействие u_n полностью неизвестно (ненаблюдаемо). Следовательно, значение сигнала s_n на основе линейного взвешивания его прошлых значений может быть предсказано только приближенно. Это предсказываемое значение s_n обозначим \hat{s}_n и запишем

$$\hat{s}_n = \sum_{k=1}^p a_k s_{n-k} \quad (6)$$

Очевидно, что ошибка предсказания есть

$$e_n = s_n - \hat{s}_n = s_n - \sum_{k=1}^p a_k s_{n-k} \quad (7)$$

часто называется также остатком. Согласно методу наименьших квадратов, пара-

-45-метры Y/t подбираются таким образом, чтобы минимизировать среднее значение или сумму квадратов ошибок предсказания.

Детерминированный сигнал. Пусть сумма квадратов ошибок предсказания есть

Множество возможных значений индекса суммирования в (8) должно быть задано, однако пока мы этого делать не будем. E как функция параметров принимает минимальное значение в точке, для которой

$$\frac{\partial E}{\partial a_i} = 0, \quad i = 1, \dots, p \quad (9)$$

Из (8) и (9) получаем следующую систему уравнений:

$$\sum_{k=1}^p a_k Z^k V = -Z^J V, \quad |Z| < 1, \quad (10)$$

которые называются *нормальными уравнениями*. При любом определении сигнала s_n система (10) содержит p уравнений с p неизвестными коэффициентами $\{a_k, k = 1, \dots, p\}$. В результате решения этой системы уравнений могут быть найдены те значения неизвестных коэффициентов, которые минимизируют величину ошибки E в (8).

Минимальное значение суммы квадратов ошибок получают из (8) с подстановкой (10). Результат имеет вид

$$Y_p = Z^+ Z^* Z^J Z^* Z^J Z^+ \quad (11)$$

При моделировании речевых сигналов с помощью модели, имеющей только полюсы, исходными данными являются речевые колебания, соответствующие свертке импульсного отклика голосового тракта с возбуждающим сигналом. Для нахождения параметров модели используют два метода: автокорреляционный и ковариационный.

Первый метод используется чаще. Одним из главных его преимуществ является то, что матрица коэффициентов получается *теплицевой*, т. е. все элементы матрицы, расположенные на главной диагонали, равны между собой [12]. Таким образом, при вычислении

-46-элементов матрицы достаточно найти одну ее строку. Кроме того, существуют эффективные методы для решения получающейся системы уравнений. Матрицы Теплица никогда не бывают сингулярными, и, при отсутствии ошибок вычисления искомое решение для фильтра только с полюсами оказывается устойчивым. Ковариационный метод таких гарантий не дает [И].

Выбор метода анализа

Выбор автокорреляционного или ковариационного метода для анализа зависит от разных факторов, например, от того, проводится ли анализ с целью передачи речи или же с целью оценки характеристик речевого тракта. Автокорреляционный метод оценки спектра является глобальным в том смысле, что для получения полезных результатов в пределах анализируемого окна должно содержаться не менее одного периода основного тона (ОТ) вокализованной речи. С другой стороны, ковариационный метод может использоваться либо в качестве локального метода (применяемого к интервалу меньшей длительности, чем период ОТ), либо в качестве глобального. Если анализируется несколько периодов ОТ, то результаты определения спектра обоими методами будут очень близкими. Для невокализованной речи оба метода будут давать близкие результаты на интервалах анализа, превышающих 5-10 мс. Сходство результатов обусловлено тем обстоятельством, что при достаточно большом числе отсчетов ковариационные коэффициенты будут приблизительно равны автокорреляционным коэффициентам. При этом, для автокорреляционного метода требуется меньший объем вычислений (что важно при стремлении к работе в реальном времени), он гарантирует устойчивость и позволяет легко рассчитывать коэффициент усиления. Отсюда следует, что автокорреляционный метод предпочтительнее [11, с. 152].

Автокорреляционный метод. Предположим, что минимизируется значение ошибки на бесконечном интервале наблюдения, т. е. $-\infty < n < +\infty$. Уравнения (10) и (11) запишутся тогда следующим образом

$$2 > 4d(-*) = -e_d, \quad |i| < p \quad (12)$$

$$f_{,,} = L(0) + f_{a4} B_d, \quad (13)$$

где

$$B_d = 2 > A_{+} \cdot \quad (14)$$

$L = -\infty$

есть автокорреляционная функция сигнала $S_{ij}(z)$ - четная функция, т. е.

$$R(-i) = R(i) \quad (15)$$

В прикладных задачах сигнал s_n известен только на конечном интервале или его рассматривают только на конечном интервале.

В таком случае вводится весовая функция (окно) $a >_n$. С помощью этой функции формируется другой сигнал s_n , равный нулю вне некоторого интервала $0 < n < N - 1$

$$s_n = \begin{cases} a_n, & 0 < n < N - 1 \\ 0, & \text{в других случаях} \end{cases} \quad (16)$$

Автокорреляционная функция такого сигнала есть

$$Y(0) = X^* \cdot X, \quad 1/2 \cdot 0 \quad (17)$$

Определение коэффициента G. При выводе нормальных уравнений (метод наименьших квадратов) предполагалось, что входное воздействие u_n неизвестно. Равенство (7) можно записать следующим образом:

$$s_{,,} = \sim 1 X^5 \cdot - * + e_{,,} \quad (18)$$

При сравнении (4) и (18) видно, что $G u_{,,} = e_{,,}$. Это означает, что входной сигнал пропорционален сигналу ошибки. При любом другом возбуждающем воздействии сигнал $u_{,,}$ на выходе фильтра $H(z)$ (см. рис. 2.1.2) будет отличаться от сигнала $s_{,,}$. Однако, если потребовать, чтобы энергия сигнала на выходе фильтра при любом $u_{,,}$ равнялась энергии исходного сигнала $s_{,,}$, то окажется возможным оценить, по крайней мере, энергию входного сигнала.

-48- Так как передаточная функция $H(z)$ фильтра фиксирована, то из приведенных выше рассуждений следует, что энергия входного сигнала $G u_{n,}$ должна быть равна энергии сигнала ошибки, которая в свою очередь определяется выражением (13).

При решении задачи распознавания речи представляют интерес два типа входных воздействий: детерминированная последовательность импульсов и стационарный белый шум. Анализ откликов фильтра $H(z)$ на эти входные воздействия дает дополнительные сведения о свойствах рассматриваемой модели во временной области.

Импульсный входной сигнал Пусть на вход фильтра, функция $H(z)$ которого имеет только полюсы, в момент времени $n = 0$ воздействует единичный импульс $u_{n,} = \delta_{n,0}$, δ_{nm} — дельта - функция Кронекера. Выходным сигналом в этом случае будет импульсный отклик фильтра $h_{n,}$

$$K = -T, a^h \gg -k^{+GS} * o \quad (19)$$

Автокорреляционная функция $R(i)$ импульсного отклика $h_{n,}$ связана с автокорреляционной функцией $R(i)$ сигнала $s_{n,}$. Чтобы показать это, умножим левую и правую части равенства (19) на $h_n i$ и просуммируем по всем значениям n . В результате получим

$$R(i) = -\sum a_k R(i - k), \quad 1 \leq i < \infty \quad (20)$$

$$k(0) = -\sum a_k \Pi_k + O^2 \quad (21)$$

Чтобы выполнялось сформулированное выше условие равенства энергий сигналов h_n и $s_{n,}$, необходимо

$$L(0) = D(0), \quad (22)$$

поскольку значение автокорреляционной функции в нуле равно мощности соответствующего сигнала. Из (21) и идентичности равенств (12) и (20) следует, что

$$L(i) = R(i) \quad 0 < i < p \quad (23)$$

То есть первые $p+1$ коэффициенты автокорреляции импульсного отклика фильтра $H(z)$ равны соответствующим коэффициентам автокорреляции сигнала на выходе этого

-49-фильтра. Уже рассмотренная задача линейного предсказания, решение которой было получено автокорреляционным методом, теперь может быть сформулирована иначе следующим образом. Необходимо найти передаточную функцию фильтра вида (5), такую, чтобы первые $p+1$ значения автокорреляционной функции (АКФ) его импульсного отклика были равны первым $p+1$ значениям функции автокорреляции наблюдаемого сигнала и чтобы выполнялось (31).

Из (21), (23) и (13) следует, что

$$C^2 = \mathbb{E}_p = D(0) + \mathbb{E}_{>}, *(\mathbb{E}), \quad (24)$$

где G — мощность входного воздействия $GS_{,,o}$

Белый шум на входе Предположим теперь, что входной сигнал является случайной последовательностью с некоррелированными значениями (белым шумом); пусть каждый элемент последовательности имеет нулевое среднее значение и единичную дисперсию. Сигнал на выходе рассматриваемого фильтра при таком воздействии на входе обозначим символом $s_{,,}$. Для фильтра $H(z)$ с постоянными параметрами выходной сигнал s_n является стационарным случайным процессом:

$$K = -Y, {}^a k K_{-k} + G u_n \quad (25)$$

Умножим обе части равенства (25) на s_{n-l} и найдем их математические ожидания. Учитывая при усреднении, что величины u_n и s_{n-l} не коррелированы для $i > 0$, вновь приходим к равенствам (20) и (21), в которых теперь $R(i) = \mathbb{E}(i_n s_{n-l})$ является АКФ выходного сигнала s_n [13]. Следовательно, (20) и (21) также полностью определяют случайный процесс на выходе фильтра, имеющего только полюсы.

При случайном сигнале на выходе следует потребовать, чтобы средняя мощность (или дисперсия) процесса s_n на выходе фильтра была равна дисперсии исходного моделируемого сигнала $s_{,,}$, т. е. $R(0) = R(0)$, поскольку значение АКФ в нуле для случайного процесса с ну-

-50-левым математическим ожиданием равно дисперсии этого процесса. Аналогично тому, как это было сделано выше, можно показать, что равенства (23) и (24) остаются справедливыми также и при случайном входном сигнале рассмотренного вида.

Таким образом, соотношения для коэффициентов автокорреляции сигнала на выходе фильтра, имеющего только полюсы, остаются одними и теми же независимо от того, является ли входной сигнал единичным импульсом или случайным белым шумом. Этому и следовало ожидать, поскольку оба эти входные воздействия имеют одинаковые АКФ и, следовательно, одинаковые равномерные спектры. Эта особенность является очень важной при моделировании речи.

Как было отмечено выше, значения коэффициентов экстраполятора a_k могут быть найдены путем решения некоторой системы p уравнений (12) с p неизвестными. Важным достоинством метода линейного предсказания является наличие эффективных приемов решения системы уравнений (12). При автокорреляционном методе можно воспользоваться очень эффективными итеративными алгоритмами [13, 6, 11], первоначально предложенными Левинсоном [14] и усовершенствованными Дурбиным [15].

Действительно, полезными на практике являются представления речи в виде параметрической модели (линейная дискретная динамическая система (ЛДДС) с переменными параметрами) с передаточной функцией:

$$1=0$$

где коэффициенты $\{a_n\}$ несут информацию о состоянии ЛДДС на временном отрезке (кадре) и называются коэффициентами линейного предсказания (КЛП), а фильтр с передаточной функцией

$$1=0$$

называют обратным фильтром или фильтром-предсказателем (ФП). Коэффициенты

фильтра-предсказателя медленно изменяются во времени и могут быть оценены непосредственно с использованием очень эффективных с вычислительной точки зрения алгоритмов. Этот метод обеспечивает на каждом кадре (временном отрезке) стохастического сигнала получение коэффициентов нерекурсивного цифрового фильтра-предсказателя, частотная характеристика которого связана со спектральной функцией данного кадра. На кадре сигнал считается стационарным.

Корни полиномов, специальным образом полученные из полинома $A(z)$, впервые предложенные Итакурой и названные линейными спектральными корнями (ЛСК), очень широко используются в качестве признаков речевого сигнала для построения системы распознавания речи. ЛСК были предложены Итакурой в качестве альтернативы коэффициентам $\{a_i\}$. ЛСК Итакуры оказались не единственно возможными: в работе А.А. Ланнэ «Новая теория спектральных корней» построена общая теория ЛСК, которая "...дает возможность «генерировать» различные варианты ЛСК". В рамках этой теории ЛСК Итакуры являются частным случаем. Различные варианты ЛСК могут быть получены из решения уравнений

$$\operatorname{Re}\{z^N A_N(z)\} = 0, \operatorname{Im}\{z^N A_N(z)\} = 0 \text{ при } R > (N/2), \quad (26)$$

$$A_N(z) = 1 + \sum_{i=1}^N a_i z^i$$

Возможны следующие частные случаи расчета корней в зависимости от значения R :

1. При $R = (N/2)$ решают оба уравнения (26), при этом общее число корней равно N и для полной информации о многочлене $A_N(z)$ необходимо знать значение еще одного параметра, например масштабного множителя при $\operatorname{Im}(a_N)$.
2. При $R = ((N+1)/2)$ - случай Итакуры. Решают оба уравнения (3), число корней N .
3. При $R > ((N+1)/2)$ решают оба уравнения (3), общее число корней больше N (избыточно).
4. При $R = N$ число корней уравнения для $\operatorname{Re}(a_N)$ равно N , а для $\operatorname{Im}(a_N)$ - $N-1$. Для полной информации о многочлене достаточно знать N корней $\operatorname{Re}(a_N)$ или $N-1$ корней $\operatorname{Im}(a_N)$ и масштабный множитель.

-52-Практически во всех исследованиях и разработках

используется случай 2, предложенный Итакурой и на сегодняшний день хорошо изученный. Случай 4 не исследован и представляет интерес не только в приложении к задаче распознавания речи, но и к задачам компрессии речи и экономной передачи параметров ЛП. Различие вариантов 2 и 4 расчета ЛСК в том, что в первом случае ЛСК являются корнями двух специально образованных полиномов,

а во втором случае корнями вещественной и мнимой частей полинома $z \blacksquare A_N(z)$. Это приводит к несовпадению алгоритмов получения ЛСК и их численных значений. Однако для задачи распознавания речевых сигналов важнее другое отличие, которое ни кем из исследователей не отмечалось: гистограммы ЛСК, рассчитанных для случая 4, не перекрываются между собой, если гистограммы рассчитаны в пределах фонем. Подробнее об этом будет написано ниже.

В настоящей работе рассматривается случай 4, когда $R=N=10$. В этом случае вместо традиционного решения двух уравнений порядка $N/2$ для определения ЛСК (одно для вещественной, другое - для мнимой частей) достаточно решить только одно уравнение порядка N , чтобы по его корням найти все коэффициенты исходного многочлена.

2.2 Анализ точности модели линейного предсказания

Выбор меры ошибки и исследование свойств этой меры являются важными моментами решения любой задачи аппроксимации или подбора функции. Выбранная мера ошибки при определении параметров экстраполятора - традиционная мера метода наименьших квадратов. Главное её преимущество состоит в том, что она приводит к простым математическим решениям. Эта мера с большим весом учитывает большие ошибки, в то время как вклад от небольших ошибок оказывается малым. При анализе во временной области часто трудно сказать, является ли такая мера подходящей для каждой конкретной задачи. В задачах линейного предсказания такая мера выбрана потому, что ее можно записать в частотной облас-

-53-ти и можно интерпретировать как меру сходства между заданным спектром сигнала и спектром модели.

Ошибку E_p можно интерпретировать как геометрическое среднее спектра модели. Согласно неравенствам

$$0 < E_p < E_m, \quad E_0 = R(0)$$

ошибка монотонно уменьшается с ростом p . Минимум достигается при $p \rightarrow \infty$. Если $P(a>)$ является ро-полосным спектром, то $E_p - E_{mm}$ для всех $p > p_0$. $E_{mm} = 0$ только в том случае, если спектр $P(\omega)$ равен нулю на бесконечном множестве частот (т.е. на линейном сегменте). Значения такого сигнала могут быть представлены точно, вследствие чего ошибка предсказания равна нулю. Следовательно, абсолютный минимум ошибки зависит только от вида спектра и оказывается равным геометрическому среднему этого спектра, которое всегда положительно для положительно определенных спектров. Аналогичные свойства меры ошибки при использовании критерия максимального правдоподобия отметил Итокава. Этот результат свидетельствует о том, что минимальная ошибка может быть отличной от нуля даже тогда, когда аппроксимирующий спектр полностью идентичен исходному спектру. Следовательно, хотя E_p является мерой близости спектра модели к спектру анализируемого сигнала, ее всегда следует сравнивать с E_{mm} . Отличие от нуля величины E_{mm} становится понятным, если учесть, что при любом p ошибка E_p равна той части энергии сигнала, которая не может быть предсказана экстраполятором порядка p . Например, импульсный отклик фильтра, передаточная функция которого имеет только полюсы, предсказуема точно, за исключением ненулевого начального значения. Именно эта энергия, содержащаяся в начальном значении, обуславливает положительное значение E_p . В ЛП-анализе предпочтительна именно такая мера ошибки. В [13] отмечается, что одним из самых важных этапов построения модели, имеющей только полюсы, состоит в выборе «оптимального» числа полюсов. Очевидно, желательно ограничиться минимальным числом полюсов, при котором еще сохраняется адекватность модели в конкретной рассматриваемой проблеме. Это необходимо как с точки зрения уменьше-

-54-ния объема вычислений, так и для сведения к минимуму возможности плохой обусловленности (которая возрастает с ростом p). Простым критерием для выбора числа полюсов может служить неизменность значения ошибки при дальнейшем увеличении p . Однако, если сигнал наблюдается на выходе l -полюсного фильтра, возбуждаемого белым шумом, то предложенный критерий не подходит, поскольку кривая ошибки не будет плоской при $p > p_0$. На практике ошибки при $p > p_0$, как правило, постоянны. Поэтому целесообразно выбрать следующее правило проверки:

$$\frac{V}{V_p} \pm 8$$

Так следует проверить несколько последовательных значений p , чтобы удостовериться в появлении плоского участка кривой ошибки. В статистической литературе проверка по отношению $V_p + l/V_p$ для определения оптимального значения p является общепринятой. Макхоул [13] предложил использовать теоретико-информационный критерий, статистика которого является оценкой среднего значения логарифма отношения правдоподобия, названный им «финальной ошибкой предсказания». Эта статистика имеет вид

$$I(p) = -2 \log (\text{отн. макс. правдоп.}) + Bp \quad (27)$$

В качестве оптимального выбиралось значение p , соответствующее минимуму функции $I(p)$. Акаике, на основании результатов исследований с использованием критерия (27), было доказано, что оптимальное значение экстраполятора $p = 10$ (Взвешивание сигнала осуществлялось окном Хэмминга). Для точного представления сглаженного спектра невокализованных звуков требуется намного меньше коэффициентов, однако принято использовать фиксированный порядок анализирующего фильтра [11, с. 158].

Линейное предсказание является автокорреляционным анализом. Следовательно такой анализ может быть проведен либо во временной, либо в частотной областях. Критерий наименьших квадратов во временной области переходит в соответствующий критерий аппроксимации спектра в частотной области. В [И, с. 132] приведена оценка нормированной полной квадратической ошибки и минимальной ошибки предсказания (или минимальной

-55- суммарной СКО предсказания) спектра.

График зависимости минимальной суммарной СКО

предсказания от порядка фильтра позволяет определить минимальное число коэффициентов полинома передаточной функции. С увеличением числа коэффициентов с 2 до 10 ошибка существенно уменьшается. Пологий участок кривой начинается с 10 и дальнейшее увеличение числа коэффициентов не приводит к заметному снижению СКО предсказания.

2.3 Статистические свойства ЛСК

В диссертационной работе исследовались статистические свойства ЛСК ((26), случай 4), рассчитанных на отдельных кадрах РС длительностью 15 мс (120 отсчетов) при частоте дискретизации 8 кГц. Такие параметры расчета ЛСК выбраны на основании результатов исследований, приведенных в [6, 11, 13, 28, 38]. Выбор интервала анализа включает два фактора: местоположение этого интервала относительно периода ОТ и его длительность. Хороший выбор длительности интервала зависит от того, анализируется переходный процесс или квазипериодический. Целесообразно проводить спектральный анализ внутри интервала, где изменения речевого тракта незначительны. Для большинства звуков этот интервал составляет 15-20 мс. Такой интервал достаточен для надежных оценок спектра. Таким образом, приемлемое число отсчетов, используемое на интервале анализа, равно частоте дискретизации в кГц, умноженной на 15-20 мс. Это значение представляет собой компромисс между желаниями иметь устойчивые оценки спектра и минимизировать усреднение меняющегося во времени сигнала.

На участках длительностью 15-20 мс речевой сигнал можно считать стационарным. В расчетах использовались только те кадры РС, которые попадали в границы фонем во временной области (рис. 2.3.1). Переходные участки (дифтонги) в расчетах не использовались. Левые и правые границы фонем определялись субъективно.

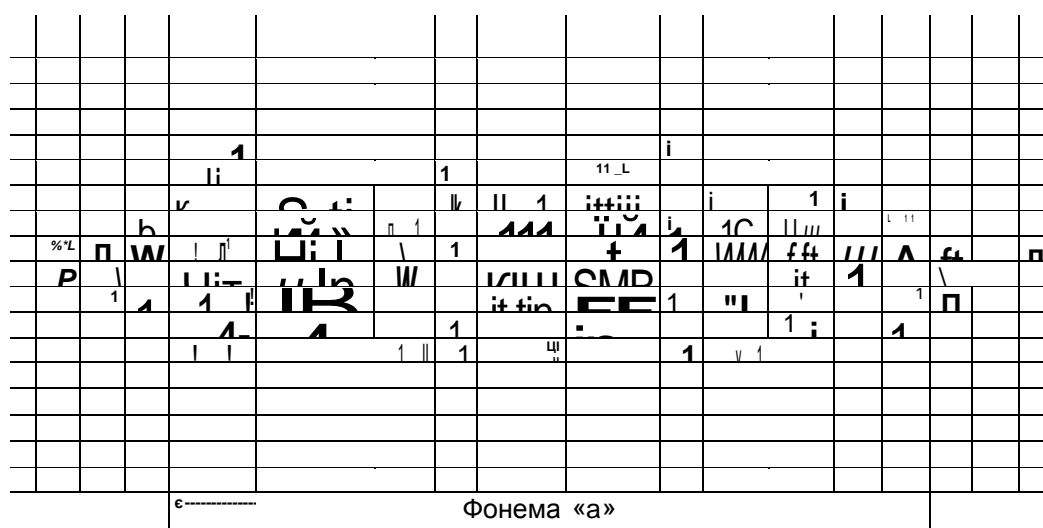
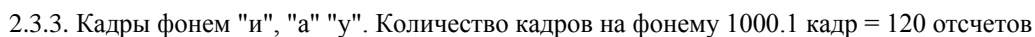
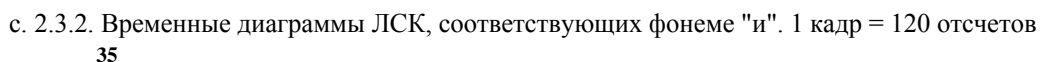


Рис. 2.3.1. Временной отрезок РС, соответствующий фонеме «а».

На рис. 2.3.2 показаны временные диаграммы 10-ти ЛСК одной фонемы. По горизонтальной оси отложены номера кадров. Всего рассчитано 1000 кадров, соответствующих одной фонеме, озвученной 15-ю различными дикторами (мужчинами и женщинами). Каждая точка диаграммы представляет собой значение одного ЛСК со своим порядковым номером, рассчитанное на одном кадре РС. Вертикальный срез дает десятку ЛСК, вычисленных на n -м кадре сигнала. Участки РС, соответствующие времени звучания фонемы, вырезались из непрерывной речи и «пристыковывались» друг к другу. Таким же образом формировался весь банк фонем. Это позволяло быстро рассчитывать ЛСК не по отдельности для каждой фонемы одного диктора, а сразу, обрабатывая весь звуковой файл. На приведенных диаграммах хорошо видно, ЛСК разделяются по оси ординат, т.е. значения каждого ЛСК локализованы вокруг некоторого «своего» среднего. При этом ни значения «соседних» корней, ни остальных не перекрываются между собой по ординате. Можно также указать на различие поведения корней: явно видно, что они имеют различный разброс относительно своих средних значений.



-58-Диаграммы, представленные на рис. 2.3.3,

очень хорошо иллюстрируют изменение

поведения корней при смене фонем. На диаграммах отчетливо видны границы фонем. При переходе от одной фонемы к другой практически все 10 ЛСК «реагируют» - меняются как их средние значения, так и дисперсия. Можно также утверждать, что меняются и значения моментов более высоких порядков.

Проведем количественную оценку статистических параметров ЛСК. Для этого рассчитаем математическое ожидание (МО) и дисперсию ЛСК для гласных и огласованных согласных фонем.

Таблица 2.3.1. Математические ожидания ЛСК фонем

	1	2	3	4	5	6	7	8	9	10
а	0.1279	0.3894	0.6309	0.8906	1.2121	1.6153	1.9202	2.243	2.6031	2.9327
э	0.142	0.3668	0.5815	0.9686	1.3242	1.6363	1.9378	2.2499	2.607	2.9301
и	0.1493	0.2639	0.638	1.0373	1.4296	1.7154	2.0592	2.3669	2.6197	2.9235
о	0.1702	0.3675	0.5671	0.8838	1.2835	1.6433	1.984	2.2939	2.6049	2.9682
у	0.1671	0.2935	0.5793	0.9155	1.3039	1.6356	1.9841	2.3041	2.626	2.9643
ы	0.1495	0.2828	0.6081	0.9823	1.3248	1.6457	1.9437	2.3077	2.6195	2.9508
м	0.1307	0.3211	0.6497	0.9309	1.2758	1.6041	1.9492	2.276	2.6239	2.9673
л	0.1517	0.3416	0.6554	0.9579	1.2806	1.7151	1.9562	2.3561	2.6218	2.9596
в	0.1455	0.2931	0.6166	0.9394	1.2977	1.7898	1.9712	2.2991	2.6193	2.9437
р	0.1559	0.3198	0.6299	0.9445	1.3189	1.6041	1.9916	2.2636	2.6079	2.9588

Для расчета МО и дисперсии ЛСК фонем взято по 1000 кадров указанных в таблице фонем. Анализируя таблицу 2.3.1, можно отметить, что математические ожидания ЛСК фонем четко различаются, при этом можно с уверенностью утверждать, что всегда существуют такие ЛСК каждой отдельной фонемы, по которым можно точно отделить эту фонему от других фонем. МО корней с номерами 1-5 различных фонем, судя по тому, что они сильнее различаются между собой, могут оказаться более информативными, чем корни с номерами 6-10. По табличным данным можно сделать вывод о том, что МО ЛСК сильно различаются, а их дисперсии сравнительно невелики.

Построение гистограмм ЛСК для отдельных фонем позволило выявить очень важное свойство исследуемых в работе ЛСК: гистограммы в рамках каждой отдельной фонемы не перекрываются между собой. Для сравнения на рис. 2.3.4 (а-в) приведены гистограммы ЛСК

-59-фонем, рассчитанных методом Итакуры (а) и методом на основе полинома Гурвица для случаев $R = N/2$ (б) и $R = N$ (в). Хорошо видно, что в первых двух случаях гистограммы корней между собой перекрываются. Это означает, что одно и то же значение могут принимать 3-4 корня вместо 1-го. В этом случае отдельный корень не позволяет определенно позиционировать фонему в ПП ЛСК, и, таким образом вносит неоднозначность на этапе принятия решения о принадлежности фонемы к тому или иному классу.

Отсутствие перекрытий гистограмм выгодно отличает исследуемые в работе ЛСК от других параметров ЛП, в том числе от наиболее широко используемых в настоящее время ЛСК, рассчитанных методом Итакуры [6, 11, 13, 28, 38, 40]. При отсутствии перекрытий гистограмм значение каждого отдельно взятого спектрального корня можно использовать в качестве координаты признакового пространства.

Другим отличительным свойством ЛСК, рассчитанных методом на основе полинома Гурвица для случая $R = N$, от других параметров ЛП, является достаточно высокая их чувствительность к смене фонем. Это еще раз подтверждает адекватность используемой модели РС: при смене фонем меняются параметры речевого тракта - речевой тракт человека перестраивается, следовательно меняются формантные частоты. Это отражается на значениях ЛСК и, естественно, на их статистических характеристиках. На рис. 2.3.5 приведены гистограммы фонем «а» и «у» отдельно для каждого ЛСК. Перекрытия гистограмм говорят о наличии априорной ошибки разделения фонем по значениям того или иного корня. Ввиду перекрытия гистограмм ЛСК (у разных фонем) не следует ожидать очень хорошего различия фонем по значениям одного корня.

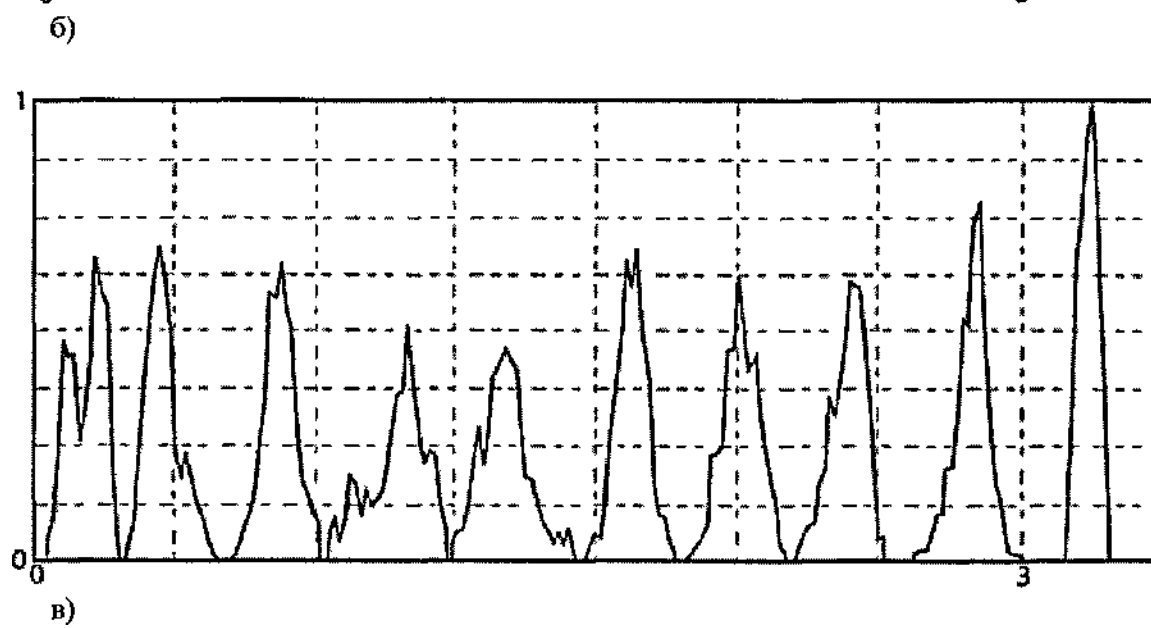
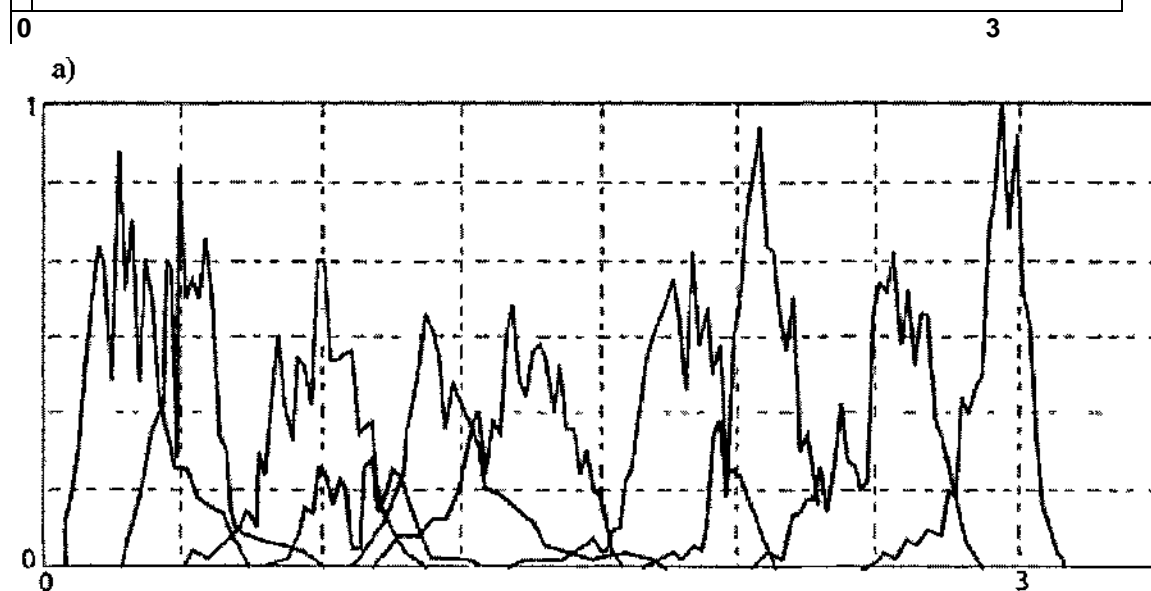
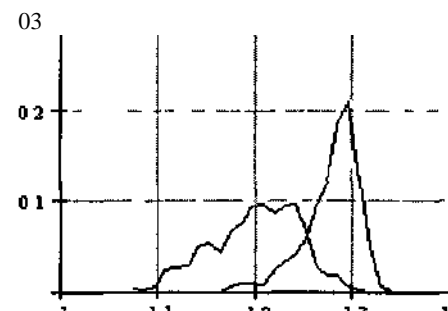
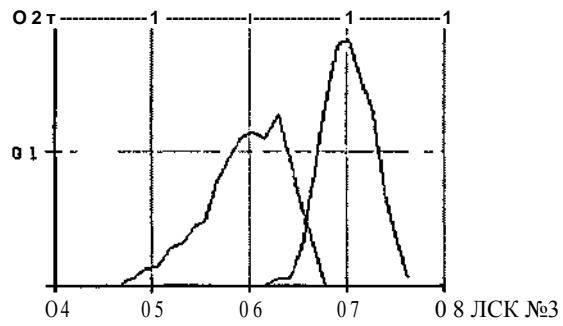
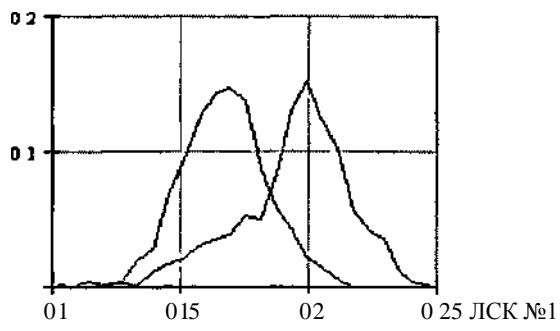


Рис. 2.3.4. Гистограммы распределений ЛСК фонем: а) ЛСК Итакуры; б) для случая $R=N/2$; в) для случая $R=N$

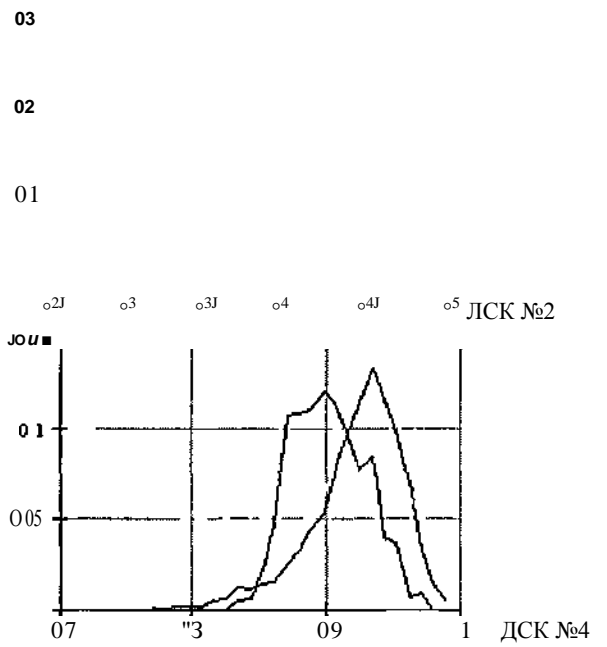
-61- Хорошо видно, что площади перекрытия гистограмм для разных номеров ЛСК отличаются. Сравнительный анализ таких картин (рис. 2.3.5 - 2.3.6) для различных пар фонем привел к следующим выводам: 1) статистические параметры распределений ЛСК фонем различаются, они меняются при переходе от одной фонемы к другой. 2) гистограммы ЛСК фонем, в зависимости от номера корня, имеют различную степень перекрытия, что говорит о необходимости выбора оптимальных номеров ЛСК для процедур распознавания; всегда существуют такие номера ЛСК, по которым разделить фонемы можно с очень маленькой или нулевой априорной ошибкой (соответственно, это те номера ЛСК, которые имеют минимальную или нулевую площадь перекрытия гистограмм) независимо от диктора.

#

¥



	A			
	Λ			
/	/ ^	V		



ЛСК №8

и / -			
0 2-			
0 1"	«ГТМГ»		

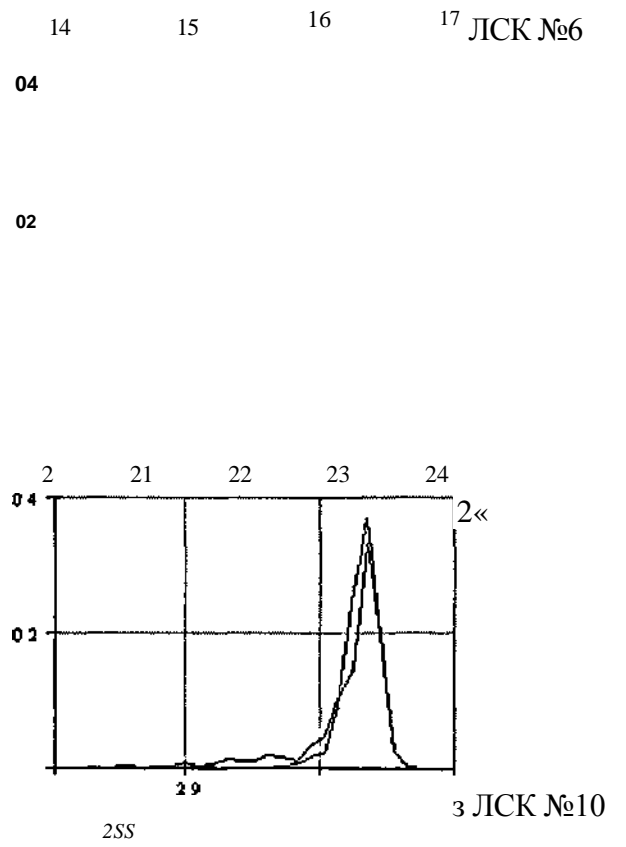
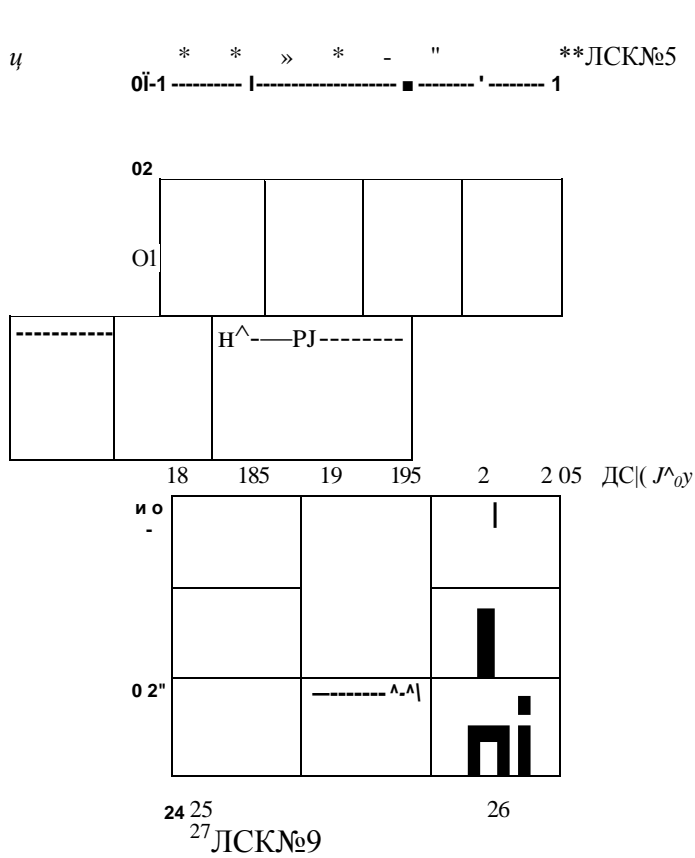
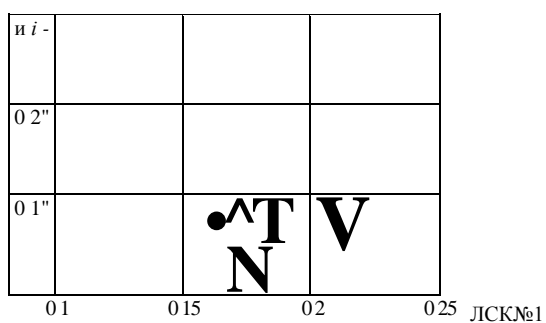


Рис. 2.3.5. Гистограммы ЛСК фонем «а» и «у» , полученных от 15-и дикторов.

63-



03

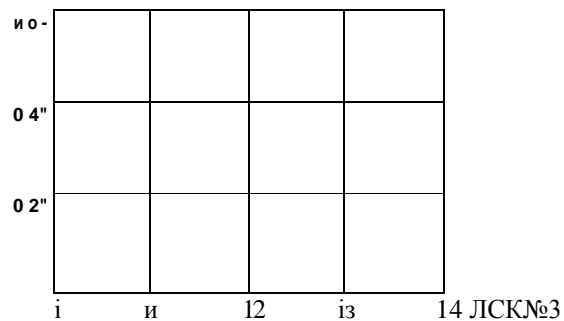
02

01

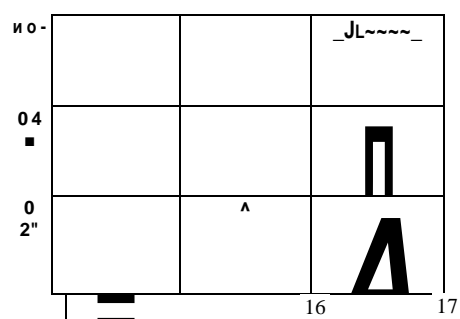
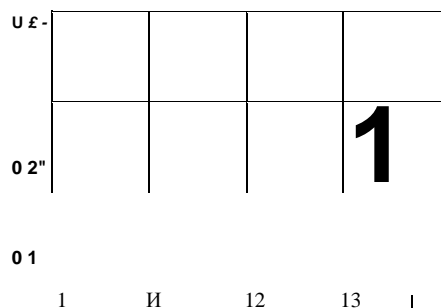
°25 03 035 04 045 05 ЛСК №2

01

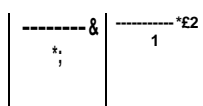
005



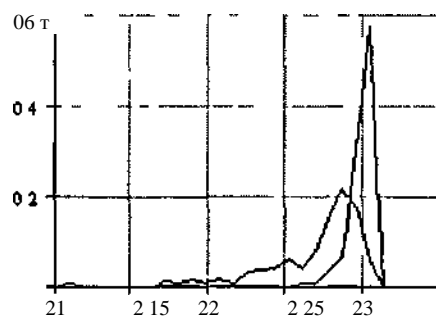
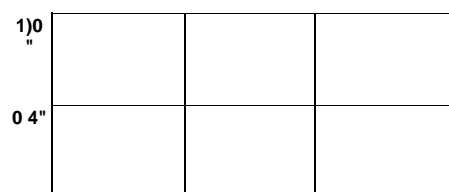
07 08 09 i ЛСК №4



ЛСК №6



i* 15



ЛСК №8

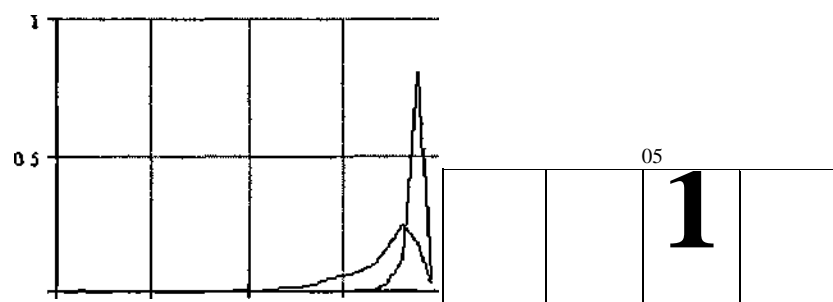
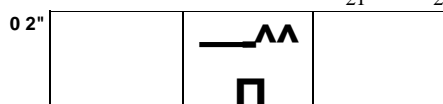


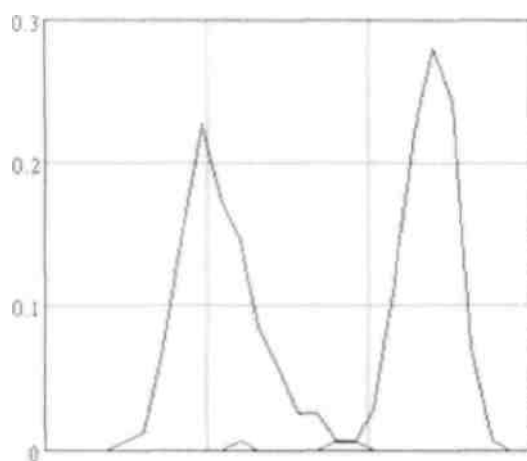
Рис. 2.3.6. Гистограммы ЛСК фонем «а» и «о», полученных от 15-и дикторов.

ЛСК №10

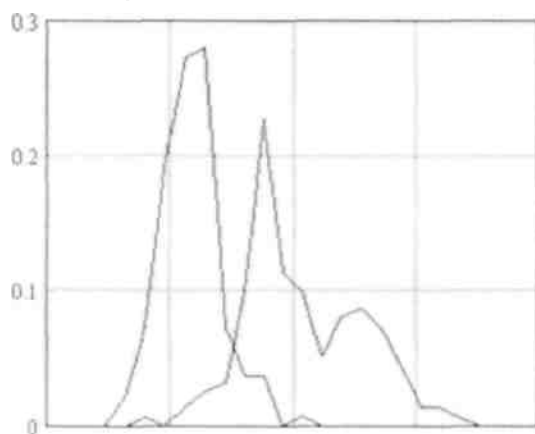
2.4 Выбор метода оценки ошибки разделения фоном в подпространствах ЛСК

Решая задачу оценки качества исследуемых признаков объектов для систем распознавания, необходимо выбрать критерий оценки признаков.

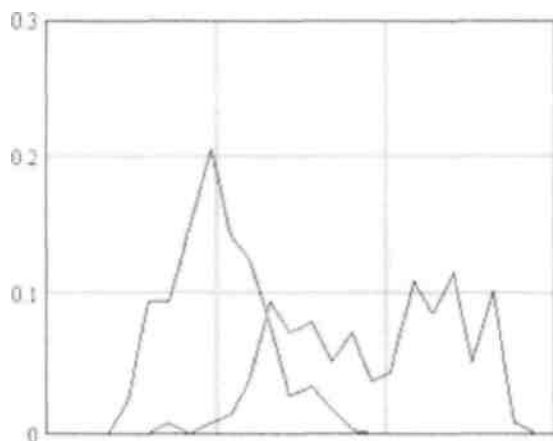
2.4.1 Использование статистического критерия на основе гистограмм Возьмем в качестве координат признаков подпространств значения 10 ЛСК, тогда каждое временное окно РС отобразится точкой в 10-мерном пространстве. Длительность фонемы обычно превышает размер окна анализа, поэтому фонемы в признаковых подпространствах (ПП) отображаются не единственной точкой, а совокупностью точек, которые и формируют образы фонем в ПП ЛСК. Разброс точек, характеризующих фонему, объясняется нестабильностью (случайностью) параметров речевого сигнала, что сказывается на значениях ЛСК (рис. 2.4.1, справа). Это подтверждается статистическими характеристиками ЛСК, приведенными в §2.3. Покажем, что площадь перекрытия гистограмм ЛСК характеризует качество разделения фоном. На рис. 2.4.1 слева показаны гистограммы ЛСК № 2, 3, 7, построенные для пары фонем «а», «и». Хорошо видно, что площади перекрытия гистограмм для различных номеров ЛСК отличаются. На этом же рисунке справа показаны образы фонем в плоскостях ЛСК 2-3 (верхний рисунок) и 3-7. В первом случае (плоскость 2-3) гистограммы имеют относительно небольшие площади перекрытия, образы фонем в этом случае хорошо разделимы. Во втором случае гистограммы ЛСК №7 имеют большую площадь перекрытия. Это, естественно, заметно отражается на разделении образов фонем в плоскости ЛСК 3-7. Разделение фоном в плоскости 2-3 возможно при помощи простой линейной границы с нулевой ошибкой. Расчеты показывают, что метод расчета ошибки через вычисление площадей перекрытия гистограмм всегда дает завышенное значение ошибки, хотя и учитывает статистические особенности ЛСК как случайных величин.



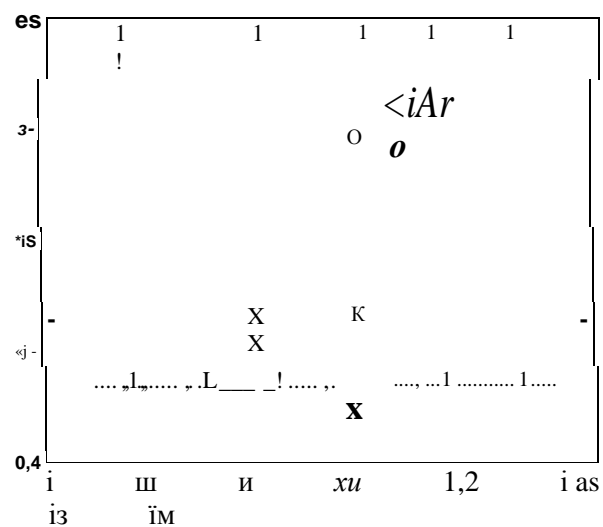
Гистограммы ЛСК №2 фонем «а», «и»



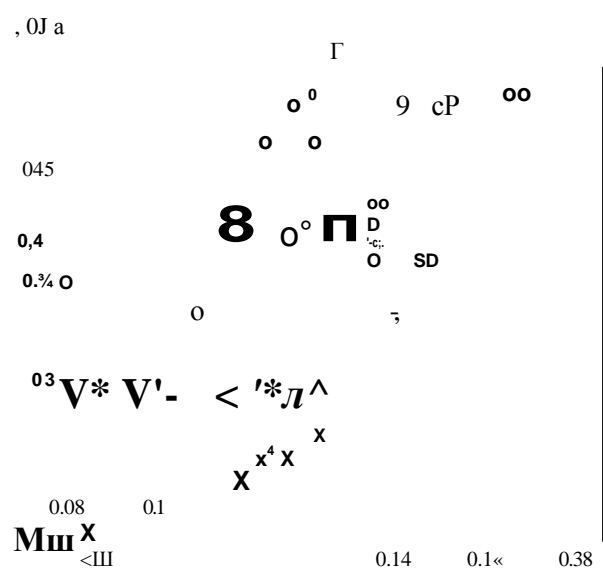
Гистограммы: ЛСК №3
фонем «а», «и»



Гистограммы ЛСК №7
фонем «а», *тш*



Отображение образов фонем «а», «и» в плоскости 2-го и 3-го ЛСК



Отображение образов фонем «а», «и» в плоскости 3-го и 7-го ЛСК

-66-Из этого следует, что, рассчитывая площади перекрытия гистограмм ЛСК, можно получить оценку априорной ошибки разделения по одному корню сверху (для каждого отдельно взятого корня). Рассчитать априорную ошибку по большему числу корней (когда нужно рассчитать ошибку разделения, например, пары фонем, описываемых парой, тройкой и т.д. корней) этим методом оказывается затруднительно в связи со взаимной корреляцией ЛСК.

2.4.2 Использование метода динамических сгущений До последнего времени развитие теории, методологии и практики статистической обработки анализируемых данных шло, по существу, в двух параллельных направлениях. Одно из них представлено методами, предусматривающими возможность вероятностной интерпретации обрабатываемых данных и полученных в результате обработки статистических выводов. Эту систему понятий, определений и методов принято называть математической статистикой. Другое направление представлено методами анализа данных и основано на следующей логической схеме: подлежащие статистической обработке исходные данные не могут интерпретироваться как выборка из генеральной совокупности, и, следовательно, использование вероятностных моделей при построении и выборе наилучших методов статистической обработки и последующая вероятностная интерпретация статистических выводов оказываются неправомерными; из множества возможных методов, реализующих поставленную цель статистической обработки данных, наилучший метод выбирается с помощью оптимизации некоторого заданного критерия (функционала) качества метода [24].

Для анализа ЛСК фонем выбран метод динамических сгущений (МДС), который применялся для решения таких задач, как:

- разбиение обследованной совокупности объектов или признаков на некоторое число (известное заранее или нет) однородных классов - собственно проблема автоматической классификации в различных постановках
- снижение размерности (числа анализируемых показателей) исследуемого массива исходных данных, отбор наиболее информативных показателей и визуализация

- Статистический анализ предпочтений, задача топологизации и агрегирования
- Статистический анализ линейных моделей регрессионного типа и т. д.

Выбор МДС обусловлен тем, что метод является достаточно гибким (может быть адаптирован к конкретной задаче) и предоставляет возможность выбирать тип ядра, вид оптимизируемого функционала качества и связанных с ним параметров, позволяет вкладывать в рамки описываемого метода широкий спектр приемов статистической обработки данных. Ядро класса (т. е. группы точек) может быть подгруппой точек, центром тяжести, осью, случайной переменной и т. д.

Выбор МДС обусловлен его простотой, высокой точностью и удобством реализации на ПЭВМ.

В работе предлагается использовать МДС для расчета ошибок классификации. В частности будем использовать метод центра тяжести - базовый подход в рамках МДС, используемый для подавляющего большинства методов классификации.

Если использовать в качестве признаков отдельного звука речи p ЛСК, то каждое окно РС будет отображено в виде точки в 7-мерном пространстве ЛСК. При $p = 2$ объект характеризуется (описывается) парой ЛСК и отображается точкой на плоскости (рис. 2.4.2).

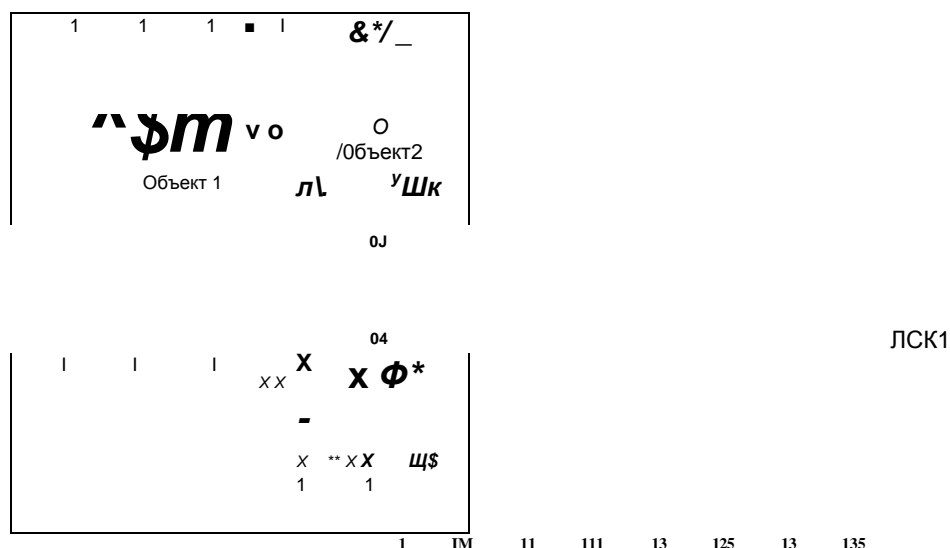
ЛСК2⁰³

Рис. 2.4.2. К описанию расчета ошибки разделения

-68-

Рассмотрим два класса объектов, отображаемых в плоскости ЛСК в виде двух сгущений точек. Каждая точка в плоскости представляет один кадр (окно) РС.

Найдем центры тяжести классов, вычислив для этого средние значения по обеим координатам для каждого класса. Если заранее известно, к какому классу принадлежит каждая точка (известны тестовые выборки), легко подсчитать ошибку классификации. Будем считать, что элемент является неправильно расклассифицированным, если он к центру «чужого» класса ближе, чем к центру «своего». В качестве меры близости используется обычное евклидово расстояние. Рассчитав, таким образом, количество правильно и неправильно классифицированных представителей, можно рассчитать вероятность суммарной априорной ошибки как отношение числа неправильно классифицированных представителей к общему числу представителей обоих классов.

Вероятность суммарной априорной ошибки разделения для случая (рис.2.4.2) рассчитывается по формуле

где J_{Vi} , N_2 - число точек-представителей классов 1 и 2, u_2 - число точек-представителей класса 1, находящихся ближе к центру тяжести класса 2, u_{i1} - число точек-представителей класса 2, находящихся ближе к центру тяжести класса 1.

По сути дела, такой метод предполагает разделение классов при помощи линейной границы, проложенной перпендикулярно отрезку, соединяющему центры тяжести классов (рис. 2.4.2). Такой метод расчета вероятности суммарной априорной ошибки (классификации фонов) вполне может заменить рассмотренный в §2.4.1 метод, требующий расчета гистограмм и их площадей перекрытия.

Последнее оказывается важным по ряду причин. Во-первых, с использованием МДС можно рассчитывать ошибку разделения фонов намного проще - расчет площадей перекрытия гистограмм оказывается более ресурсоемким и сложным, а стало быть медленным. Во-

-69-вторых, МДС позволяет работать в условиях корреляции (ЛСК коррелированы между собой, поэтому рассчитывать вероятность априорной ошибки через произведение площадей перекрытия гистограмм нельзя). В-третьих, МДС наилучшим образом подходит для реализации на ЭВМ: алгоритмически прост, эффективен и требует минимальных вычислительных ресурсов.

0 1 2 3 4 5 6 7 8 9 \

36 0.11 ().121().426 0.257 0.173().14 0.103 0.18 0.279

00

$0_{\text{ILLAlil}}^{0-!!} \quad 9_{\text{JLIZ}}^{4?0-099} 0.085 0.074 0.077 \mathbf{0.11}$

10.121 0.11 0.125 0.121 0.169 0.092 0.088 0.077 0.066

$\text{Er} = \quad \quad \quad 0.118$

10.426 0.074 0.121 J1463_0.2J17 0.125 0.125 0.132 0.202 0.408

0.257 0.173 0.169 0.217 0.261 0.202 0.199 0.202 0.195 0.254

10.173 0.099 0.092 0.125 0.202 0.173 0.165 0.129 0.129 0.162

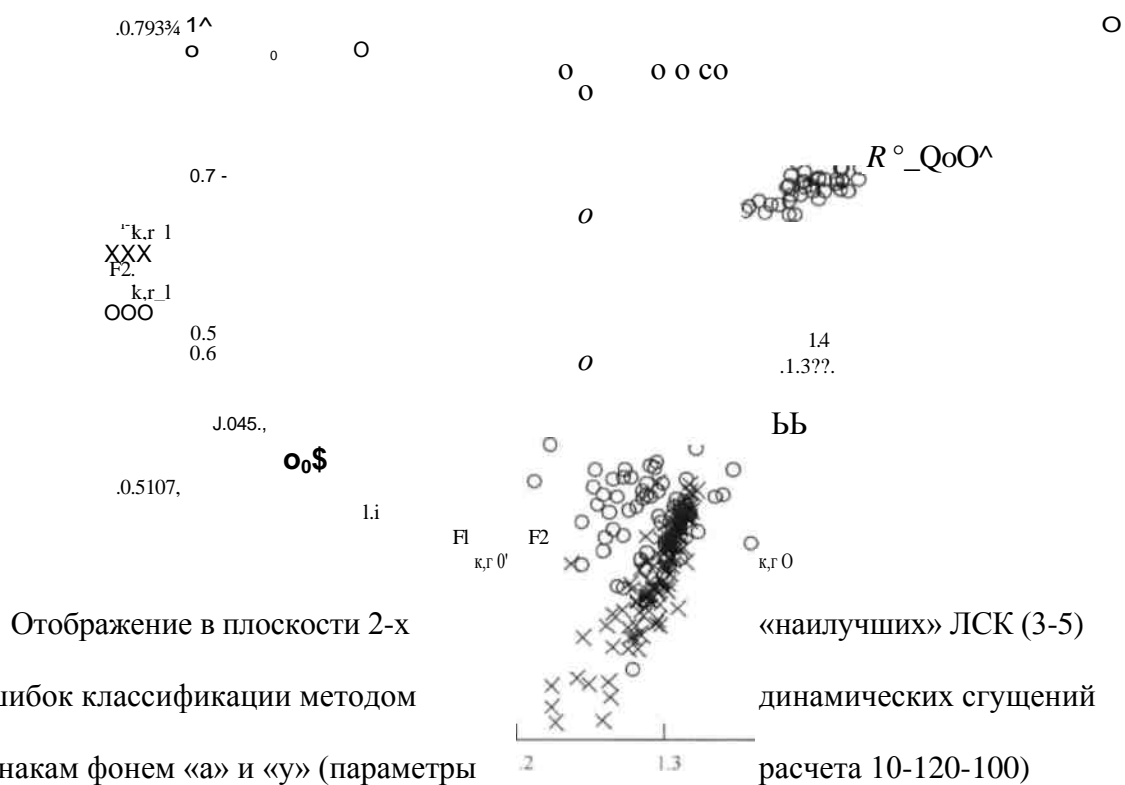
i 0.14 0.085 0.088 0.125 0.199 0.165 0.143 0.125 0.132 0.143

10403 0.074 0.077 0.132 0.202 0.129 0.125 0.1 14 0.1 18 (U 14

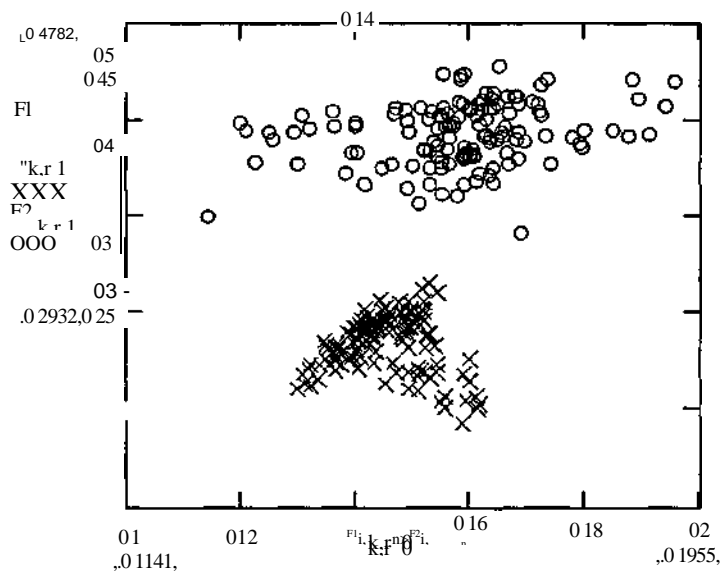
10.18 0.077 0.066 0.202 0.195 0.129 0.132 0.118 0.18 0.188

10.279 0.11 0.118 0.408 0.254 0.162 0.143 0.114 0.188

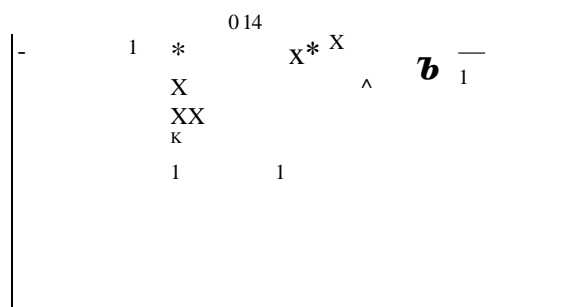
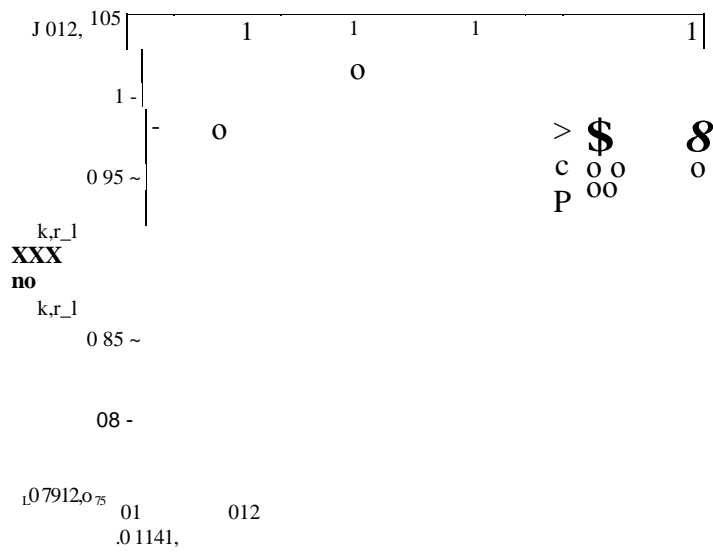
0.235



		i	2	3	4	:	5		6	8	9
		0 0.224	3.39.10 ⁻³	0.014	0.434	0.054	0.098	0.081	0.142	0.105	0.19
En	1	3.39.10 ⁻³	3.39.10 ⁻³	3.39.10 ⁻³	0.02	3.39.10 ⁻³	0	3.39.10 ⁻³	3.39.10 ⁻³	3.39.10 ⁻³	3.39.10 ⁻³
	1 0.014 3	3.39.10 ⁻³	0.014	0.031	0	0	0	0	0.01	0.014	0.014
	0.434	3.39.10 ⁻³	0.031	0.522	0.054	0.088	0.078	0.105	0.068	0.502	0.502
	4 0.054	0.02	0	0.054	0.061	0.068	0.061	0.064	0.061	0.061	0.061
	5 0.098	3.39.10 ⁻³	0	0.088	0.068	0.102	0.105	0.112	0.105	0.105	0.102
	6 0.081	0	0	0.078	0.061	0.105	0.115	0.122	0.115	0.115	0.115
	7 0.142	3.39.10 ⁻³	3.39.10 ⁻³	0.105	0.064	0.112	0.122	0.146	0.136	0.136	0.146
	8 0.105	3.39.10 ⁻³	0.01	0.068	0.061	0.105	0.115	0.136	0.173	0.173	0.173
	■ 0.19	3.39.10 ⁻³	0.014	0.502	0.061	0.102	0.115	0.146	0.173	0.173	0.244



Отображение в плоскости 2-х первых по порядку ЛСК (1-2)



JO 1955, k, r_0

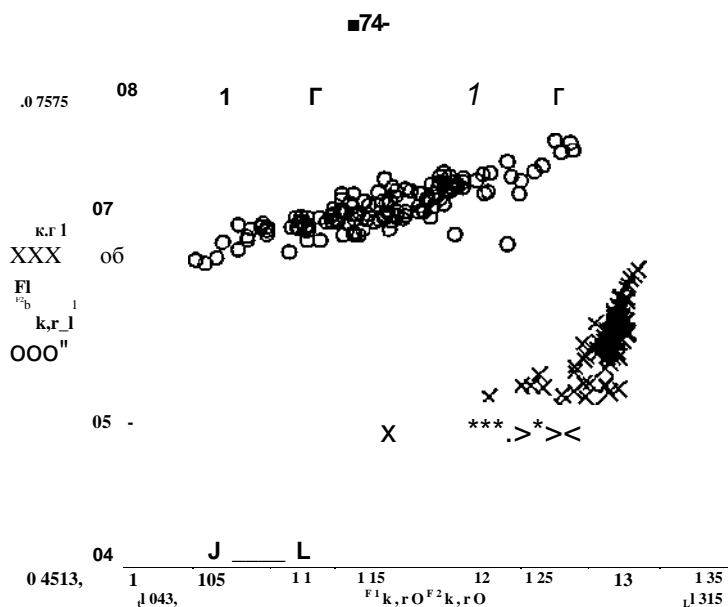
016 018
k, r_0

02

«Плохой»

случай (1-4)

V



Отображение в плоскости 2-х «наилучших» ЛСК (3-5)

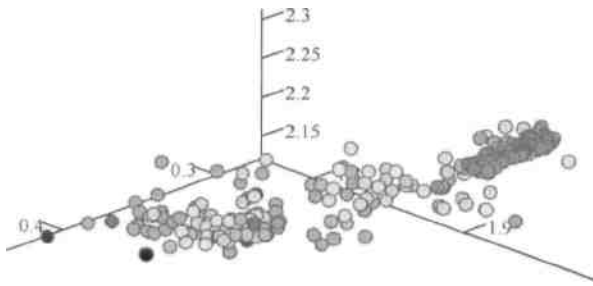
2.5.2 Оценка разделимости фонем в трехмерном подпространстве ЛСК

В таблице 5 приведены вероятности ошибок разделения звуков «а» и «у» на ЛСК (параметры получения шаг и размер 120 отсчетов, порядок модели 10). Корни из файлов a_10-120-120.prn, u_10-120-120.prn.

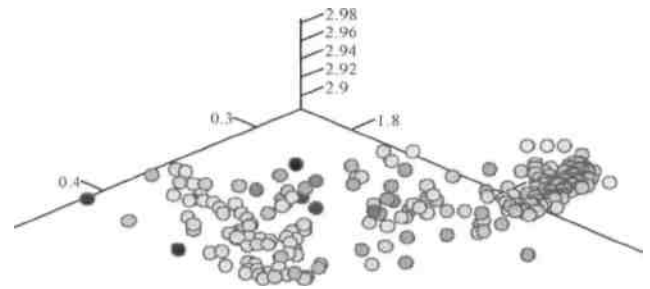
Таблица 1.5.1

a-y	0.199	0.206	0.257	0.625
0.735	0.169	0.279	0.588	0.412
0.235				

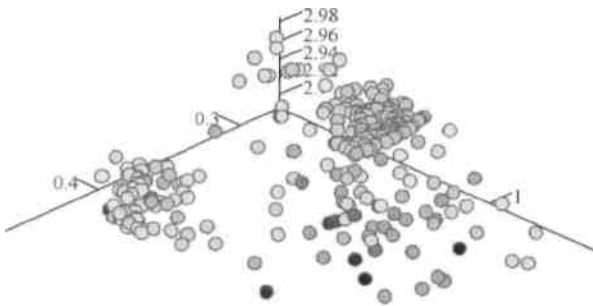
На основании данных таблицы упорядочим номера ЛСК по возрастанию вероятности ошибки. Для «а» и «у»: 2, 7, 8, 9, 6, 3, 5, 4, 10, 1.



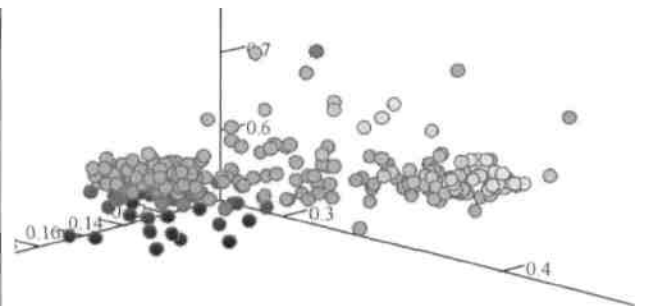
Фонемы «а» и «у». Файлы корней: а 10-120-ПО.rgp, u_10-120-120.rgp. Корни 2, 7, 8.



Фонемы «а» и «у». Файлы корней: а 10-120-120.rgp, u_10-120-120.rgp. Корни 2, 7, 9.



Фонемы «а» и «у». Файлы корней: а 10-120-120.rgp, u_10-120-120.rgp. Корни 2, 4, 10. Добавлены 2 корня -4 и 10 с большими ошибками. Ошибка разделения еще больше увеличилась.

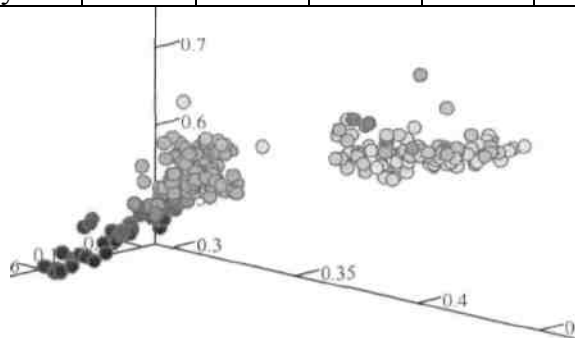


Фонемы «а» и «у». Файлы корней: а 10-120-120.rgp, u_10-120-120.rgp. Корни 1, 2, 3.

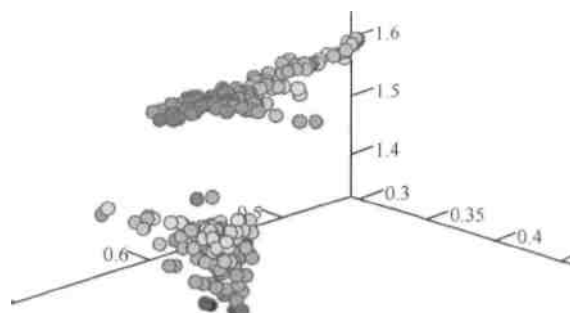
Если для разделения на 3-х ЛСК использовать другие параметры получения ЛСК, то:

Таблица 7

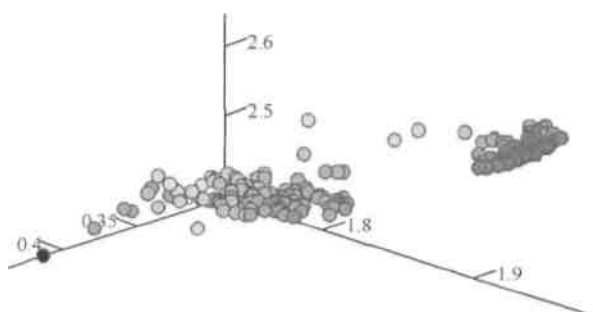
	1	2	3	4	5	6	7	8	9	10
a-y	0.576	0.007	0.015	0.766	0.128	0.152	0.162	0.229	0.22	0.893



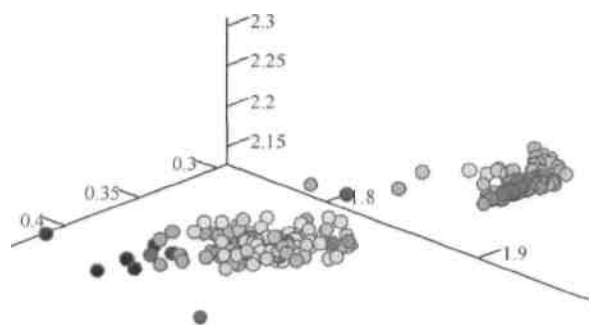
Фонемы «а» и «у». Файлы корней: a_10-120-ЮО.prn, u_10-120-100.prn. Корни 1, 2, 3.



Фонемы «а» и «у». Файлы корней: a_10-120-ЮО.prn, u_10-120-100.prn. Корни 1, 2, 5.



Фонемы «а» и «у». Файлы корней: a_10-120-ЮО.prn, u_10-120-100.prn. Корни 2, 7, 9.



Фонемы «а» и «у». Файлы корней: a_10-120-ЮО.prn, u_10-120-100.prn. Корни 2, 7, 8.

В ряде подпространств ЛСК фонемы можно уверенно разделить простыми линейными границами, используя при этом подпространства малых размерностей (пары или тройки ЛСК).

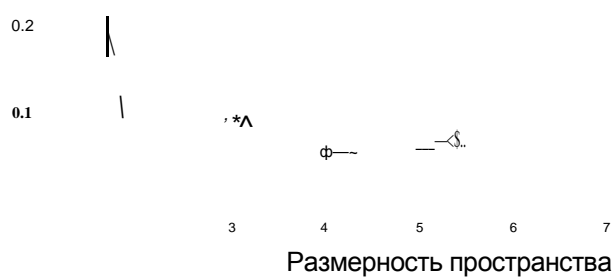
2.5.3 Оценка размерности вектора признаков

Для ответа на вопрос можно ли сократить исходное 10-мерное признаковое пространство, построим ряд характеристик, отражающих зависимость ошибки разделения от числа используемых для описания фонем ЛСК, т.е. зависимость величины ошибки распознавания от размерности вектора признаков. На рис. 2.5.1 (а-в) кривыми с кружками представлены эти

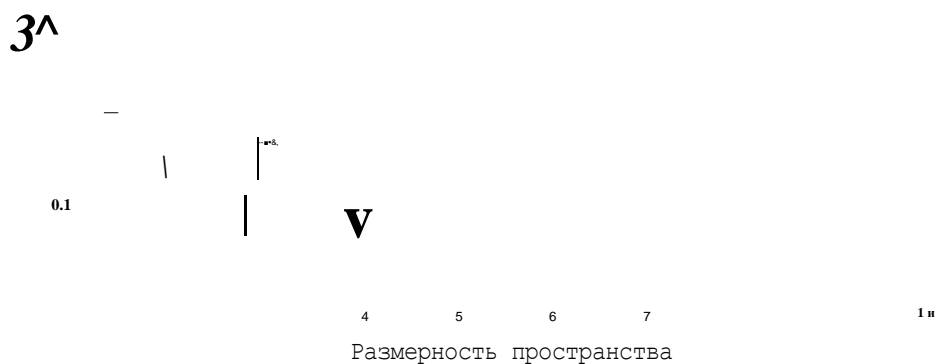
зависимости, построенные для разных пар фонем.



в) Фонемы а-о (10-120-120)



в) Фонемы у-о (10-120-120)



в) Фонемы и-о (10-120-120)

Рис. 2.5.1. Динамика уменьшения ошибки разделения.

По оси абсцисс на каждом графике отложены размерности признаков пространств (длины векторов признаков). По оси ординат - значение априорной ошибки разделения пары фонем. Таким образом, точка с координатами (2, 0.3) на рис. 100а означает, что априорная

ошибка разделения пары фонем «а-о» в подпространстве первых трех по порядку ЛСК равна

0.3. Ошибка рассчитывалась предложенным к использованию в настоящей работе в §2.4

-78-МДС. Следует обратить внимание на то, что в этом случае последовательно добавляются следующие по порядку ЛСК от первого до десятого. Изучая эти зависимости (здесь приведено их ограниченное количество (по понятным причинам)), можно сделать вывод, что, если начиная с 6-го-7-го ЛСК значение ошибки перестает уменьшаться (начинаются пологие участки кривых), то можно ограничиться таким количеством ЛСК для описания фонем. Это позволяет сократить размерность исходного признакового пространства на 30-40% (с 10 до 6-7).

2.6 Применение преобразования Карунена-Лоэва для сокращения размерности векторов признаков

При построении систем реального времени очень часто стоит задача минимизации расчетного времени, и, как следствие, уменьшения объема вычислений. При разработке сложной информационной системы это нужно учитывать на всех этапах ее проектирования.

В вокодерных системах применяется описание речевого сигнала линейными спектральными корнями. При этом используются следующие параметры расчета корней: 10-й порядок модели линейного предсказания, размер и шаг окна по времени равны 120 отсчетам при частоте дискретизации 8 кГц. Вокодерные системы предполагают восстановление речевого сигнала на «приемном конце». При этом добиваются приемлемого качества восстановленной синтетической речи при минимальных временных затратах на кодирование - декодирование. Минимума временных затрат, как правило, добиваются путем уменьшения объема вычислений за счет поиска новых алгоритмических решений.

В связи с тем, что распознавание речи должно производиться в реальном времени, необходимо исследовать возможность сокращения объема вычислений. Дело в том, что связка кодер - декодер не предполагает ресурсоемких надстроек на высоких языках программирования, т. е. у них вся программная часть реализована на цифровых процессорах обработки сигналов (ЦПОС). В системах распознавания речи обязательно должны присутствовать блоки синтаксического, семантического L. анализа, которые требуют достаточно больших словарей и реализуются с использованием более сложных и ресурсоемких методов анализа.

- Исходя из вышесказанного, представляет интерес исследование возможности сократить описание речевых единиц, т. е. исключить из рассмотрения ряд ЛСК.

Преобразование Карунена-Лоэва

В случае, когда исходное число p рассматриваемых, т. е. измеряемых на исследуемых объектах, признаков довольно велико, то для наглядности, простоты интерпретации и упрощения счета необходимо представить каждое из наблюдений в виде набора чисел, состояще-

-80-ГО из существенно меньшего (чем p)

количества признаков. При этом оставшиеся признаки

могут либо выбираться из числа исходных, либо определяться по какому-либо правилу по совокупности исходных признаков, например как линейные комбинации последних.

При формировании новой системы признаков к последним предъявляются разного рода требования, такие, как наибольшая информативность с точки зрения правильного разбиения наблюдений на классы, взаимная некоррелированность, наименьшее искажение внутренней и внешней геометрической структуры множества исходных наблюдений и т. п. В зависимости от варианта формальной конкретизации этих требований нужно выбирать тот или иной алгоритм снижения размерности.

Во многих задачах классификации и, в частности, в задаче распознавания речевых сигналов (РС) наиболее важными являются признаки, которые обнаруживают наибольшую изменчивость (разброс значений) при переходе от одного объекта (в данном случае объектами являются фонемы) к другому. При решении задач классификации, распознавания или сжатия информации такие признаки будут наиболее информативными. Одним из алгоритмов отбора признаков при решении таких задач является алгоритм, построенный на основе метода главных компонент.

Главные компоненты представляют собой новое множество используемых признаков $U^{(1)}, U, \dots, U^{(p)}$, каждый из которых получен в результате некоторой линейной комбинации непосредственно измеренных на объектах исходных признаков $l_1^{(1)}, x^{(2)}, \dots, l^{(p)}$. Полученные в результате такого преобразования новые признаки $U^{(1)}, U, \dots, U^{(p)}$ обладают удобным статистическим свойством - они упорядочены по степени рассеяния в используемой совокупности объектов. При этом первый признак обладает наибольшей дисперсией.

Для описания состояния объекта не обязательно использовать все или какие-то из исходных, непосредственно замеренных на нем признаков. Можно использовать признаки, являющиеся какими-то производными от измерений ряда параметров объекта. При этом мы, конечно, теряем какую-то долю информации, как бы огрубляя получающиеся при этом клас-

сы. Однако, как отмечено в [71] и [90], метод главных компонент дает достаточно качественное описание классов объектов, минимально искажая их структуру.

Для пояснения сущности того линейного преобразования исходной системы признаков, которое приводит к так называемым главным компонентам, рассмотрим его геометрическую интерпретацию на примере двумерной системы наблюдений $(x^{(1)}, x^{(2)}), j = 1, 2, \dots, n$, извлеченной из нормальной генеральной совокупности со средним значением $a = (a^{(1)}, a^{(2)})$ и ковариационной матрицей

$$\Sigma = \begin{pmatrix} \sigma_1^2 & r\sigma_1\sigma_2 \\ r\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}, \quad |r| < 1, \sigma_1^2 > 0, \sigma_2^2 > 0 \quad (1)$$

Здесь σ_1^2 и σ_2^2 - дисперсии компонент, соответственно $x^{(1)}$ и $x^{(2)}$, а r - коэффициент

корреляции между ними. Геометрически это означает, что точки $(x^{(1)}, x^{(2)}), j = 1, 2, \dots, n$ будут располагаться примерно в очертаниях эллипсоидов рассеивания вида (см. рисунки)

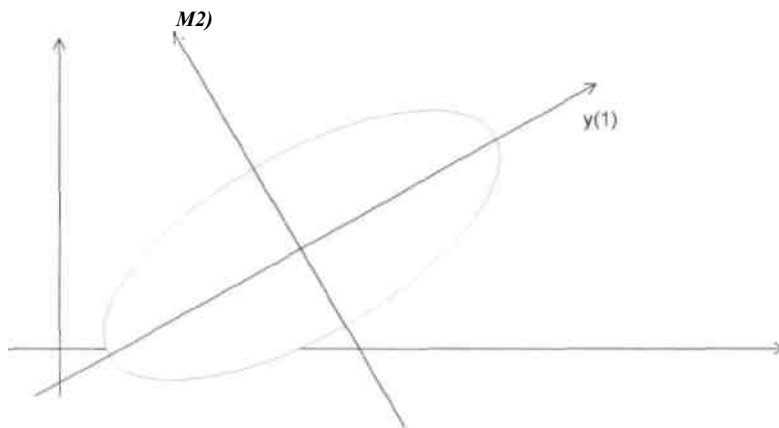


Рис. 2.6.1. Эллипсоид рассеивания исследуемых наблюдений и направление координатных осей главных компонент $y^{(1)}, y^{(2)}$

$$\sum_{j=1}^n (x_j^{(1)} - a^{(1)})^2 + \sum_{j=1}^n (x_j^{(2)} - a^{(2)})^2 = c \quad (2)$$

В этом случае для изучения $(x^{(1)}, x^{(2)})$ удобно перейти к новым координатам

$y^{(1)}, y^{(2)}$ с помощью преобразования:

$$\begin{aligned} y^{(1)} &= (x^{(1)} - a^{(1)}) \cos \alpha + (x^{(2)} - a^{(2)}) \sin \alpha \\ y^{(2)} &= -(x^{(1)} - a^{(1)}) \sin \alpha + (x^{(2)} - a^{(2)}) \cos \alpha \end{aligned}$$

$$\frac{2\alpha_1 c x_2}{\alpha_2 \cdot c, -\alpha_2 T_2} \quad \text{где} \quad \text{tg} \alpha = \frac{1}{\dots}$$

После этого преобразования точки (Y^1, Y^2) также будут распределены нормально, но компонента $y^{(1)}$ уже не будет зависеть от $y^{(2)}$. Кроме того, если выбрать направления так, что $Dy^{(1)} > Dy^m$, то геометрически это будет означать следующее: сначала производится перенос начала координат в точку $(a^m, a^{(2)})$, а затем оси поворачиваются на угол α так, чтобы ось $y^{(1)}$ шла вдоль главной оси эллипсоида рассеивания (см. рис.). Чем ближе $|\varepsilon|$ к единице, тем теснее группируются наблюдения около главной оси эллипсоида (т. е. около новой оси $y^{(1)}$) и тем менее значащим для исследователя является разброс точек в направлении оси y , а следовательно, и сама эта координата. В предельном случае $|\varepsilon| = 1$, исследуемые наблюдения в координатах $(y^{(1)}, y^{(2)})$ вообще не отличаются по координате $y^{(1)}$ (см. нижний рис.).

Определение главных компонент

Будем полагать, что исследуемые наблюдения X_1, X_2, \dots, X_n извлечены из некоторой p -мерной генеральной совокупности, определяемой соответствующей вероятностной мерой. Наиболее существенное значение в данных преобразованиях имеет ковариационная матрица $\Sigma = (\sigma_{ij})$, где

$$a_{ij} = M\{x^{(i)} - a^{(i)}\}(x^{(j)} - a^{(j)}), \quad i, j = 1, 2, \dots, p. \quad (4)$$

Здесь $a^{(i)}$ компоненты вектора a средних значений признаков X . Поскольку, как легко видеть, элементы σ_{ij} матрицы Σ не изменятся при замене признаков $x^{(i)}$ признаками $x^{(i)} = x^{(i)} - c^{(i)}$ (где $c^{(i)}$ — произвольные постоянные числа), то будем в дальнейшем считать, что вектор средних значений $a = 0$, чего всегда можно добиться, рассматривая в качестве признаков x_1^2, \dots, x_p^2 не сами измерения $x^{(1)}, x^{(2)}, \dots, x^{(p)}$ ($v = 1, 2, \dots, n$), а их отклонения от своих выборочных средних значений, т. е. полагая

$$x^{(i)} = x^{(i)} - \bar{x}^{(i)} \quad (5)$$

l -ой главной компонентой ($l = 1, 2, \dots, p$) называется такая нормированная линейная комби-

нация p исходных признаков $y^{(1)}, x^{(2)}, \dots, x^w$

$$XO = \sqrt{\lambda_1} x^{(1)} + h_2 x^{(2)} + \dots + h_p x^{(p)} = \sqrt{\lambda_1} x^{(1)} + \dots + h_p x^{(p)} \quad (6)$$

которая среди всех прочих линейных нормированных $\{h_1 x^{(1)} + h_2 x^{(2)} + \dots + h_p x^{(p)} = 1\}$ комбинаций, некоррелированных со всеми предшествующими главными компонентами $y^{(1)}, \dots, y^{(m)}$ (т. е. $\text{cov}(y^{(i)}, y^{(j)}) - M(y^{(i)} y^{(j)}) = 0$ для $i < j$), обладает наибольшей дисперсией. Здесь X - p -мерное наблюдение.

Вектор h , является i -м собственным вектором ковариационной матрицы наблюдений

$$B_{ij} = M(x^{(i)} - a^{(i)})(x^{(j)} - a^{(j)}) \quad i, j = 1, 2, \dots, p, \quad (7)$$

где $a^{(i)}$ компоненты вектора a средних значений признаков $x^{(i)}$. Его компоненты h_1, h_2, \dots, h_p

определяются как нормированное ($\sqrt{\lambda_1} = 1$) решение системы уравнений

$$(\lambda_1 E - B)h = 0 \quad (8)$$

где E - единичная матрица.

$$(E - \lambda_1^{-1} B)h = 0 \quad (9)$$

где E - единичная матрица.

После сопоставления предыдущих формул получаем

$$Bh = \lambda_1 h, \quad (10)$$

Таким образом, ковариационная матрица B главных компонент $y^{(1)}, \dots, y^{(p)}$ имеет

вид

$$B = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_p \end{pmatrix} \quad (11)$$

где $\lambda_i = D y^{(i)}$, $D y^{(i)}$ - дисперсия i -й главной компоненты.

Опираясь на то, что преобразование

$$\begin{vmatrix} (K_{Ki} & K) \end{vmatrix} \begin{vmatrix} [K] \end{vmatrix} \quad \text{я} \quad (12)$$

$$\begin{vmatrix} K_{hji} & K \\ L_{K2} & \dots \end{vmatrix} = \begin{vmatrix} K \\ L \end{vmatrix}$$

с помощью которого осуществляется переход от исходных компонент X к главным компонентам $Y(Y = LX)$, является ортогональным [], нетрудно выразить исходные переменные $x^{(1)}, x^{(2)}, \dots, x^{(p)}$ через главные компоненты

$$X'' = \dots \quad (2) \quad K Y^{(1)} + K Y^{(2)} + \dots + K Y^{(p)} \quad (13)$$

(в матричной записи $X = L' Y$), а также можно отметить, что обобщенная дисперсия $|W|$ и сумма дисперсий $(Dy^{(1)} + Z)y^{(2)} + \dots + Dy^{(p)})$ главных компонент равны обобщенной дисперсии $|2y|$ и сумме дисперсий $(Z)x^{(1)} + Z)x^{(2)} + \dots + Dx^{(p)}$ исходных признаков [3]. Последнее дает возможность вынести решение о том, сколько последних главных компонент можно без особого ущерба исключить из рассмотрения, сократив тем самым размерность исследуемого пространства.

Действительно, анализируя изменение относительной доли суммарной дисперсии

$$\begin{aligned} & \dots, \text{ч} \quad Dy^m + Dy^m + \dots + Dy^{(p)} \quad X_x + X_z + \dots + X_p \\ & Dx^m + Dx^m + \dots + Dx^{(p)} > \quad X_l + X_2 + \dots + L_p, (1 < p' \end{aligned} \quad (14)$$

$< p$), вносимой первыми p' главными компонентами, в зависимости от числа этих

компонент, можно разумно определить число компонент, которое целесообразно оставить в рассмотрении.

Так, при изменении $qfp'J$, изображенном на рис. 2.6.2, очевидно целесообразно было бы сократить размерность пространства $cp-10$ до:?' = 3, так как добавление всех остальных семи главных компонент может повысить суммарную характеристику рассеяния не более чем на 10%.

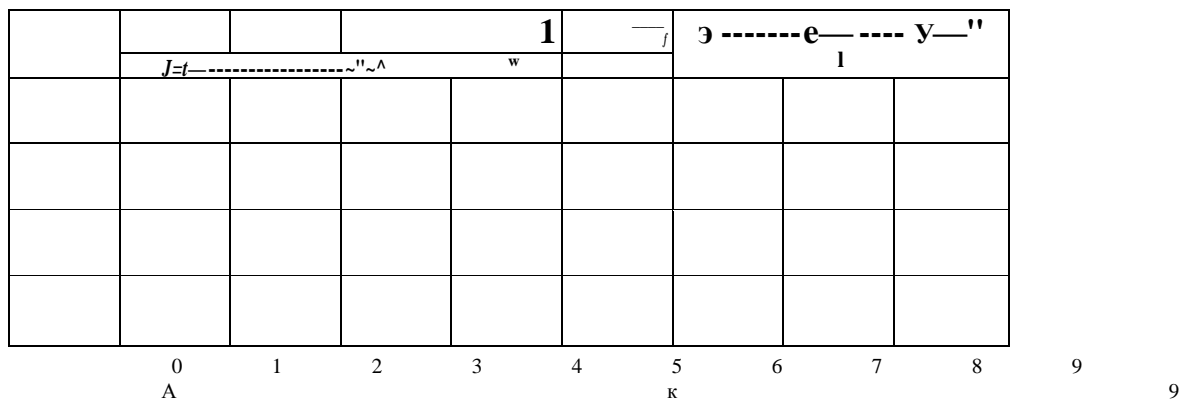


Рис. 2.6.2 Относительная доля суммарной дисперсии, вносимая первыми к главными компонентами

В реальных задачах точное знание ковариационной матрицы 2 является скорее исключением, чем правилом. Поэтому в тех случаях, когда 2 неизвестна, данные выше расчеты следует использовать применительно к выборочной ковариационной матрице, элементы которой подсчитываются на основании имеющихся наблюдений.

Главные компоненты, вычисленные на основании элементов выборочной ковариационной матрицы, называются выборочными главными компонентами, или главными компонентами выборки. Использование главных компонент оказывается наиболее естественным, когда все компоненты исследуемого вектора имеют общую физическую природу и соответственно измерены в одних и тех же единицах.

Свойства главных компонент **Свойство 1.** Свойство наименьшей ошибки «автопрогноза» или наилучшей самовоспроизводимости. С помощью первых p' главных компонент достигается наилучший прогноз этих признаков среди всех прогнозов [13, 21, 69], которые можно построить с помощью p' линейных комбинаций набора из p произвольных признаков.

Свойство 2. Свойства наименьшего искажения геометрической структуры исходных точек (наблюдений) при их проектировании в пространство меньшей размерности p' , «натянутое» на p' первых главных компонент, а) Сумма квадратов расстояний от исходных точек — наблюдений X_1, X_2, \dots, X_n до пространства, натянутого на первые p' ГК, наименьшая относи-

размерности p' , полученных с помощью произвольного

линейного преобразования исходных координат, б) Среди всех подпространств заданной размерности p' ($p' < p$), полученных из исследуемого факторного пространства X с помощью произвольного линейного преобразования исходных координат в подпространстве, натянутом на первые p' ГК, наименее искажается сумма квадратов расстояний между всевозможными парами рассматриваемых точек-наблюдений.

Свойство 3. Среди всех подпространств заданной размерности p' ($p' \leq p$), полученных из исследуемого факторного пространства X с помощью произвольного линейного преобразования исходных координат, в пространстве, натянутом на первые p' ГК, наименее искажаются расстояния от рассматриваемых точек-наблюдений до их общего «центра тяжести», а также углы между прямыми, соединяющими всевозможные пары точек-наблюдений с их общим «центром тяжести».

Критерием при выборе множества сохраняемых коэффициентов преобразования является сохранение p' коэффициентов с наибольшими дисперсиями, а остальные ($p - p'$) коэффициентов можно отбросить, задавшись значением увеличения суммарной дисперсии.

Приведенный выше критерий выбора коэффициентов определяется [4] как дисперсионный критерий. Графическим представлением дисперсионного критерия является график дисперсий коэффициентов преобразования, где дисперсии расположены в порядке убывания и нормированы к следу матрицы E_u . Нормировка производится потому, что отношение дисперсии к сумме дисперсий (т. е. к следу матрицы) дает меру (в процентах) среднеквадратической ошибки, возникающей при отбрасывании s коэффициентов.

Результаты, полученные после применения метода, можно интерпретировать двояко. В обоих случаях получаются совокупности точек, являющихся проекциями многомерных наблюдений, например, на плоскость двух первых главных компонент. Если в матрице наблюдений

$$(x_1, x_2, \dots, x_n) - \begin{vmatrix} *11 & *12 & *13 \\ -x_{1n} & x_{2n} & *22 & x_{23} \\ -3/4 \end{vmatrix} \quad (15)$$

$/?1 \quad /?2 \quad /?3 \quad pn$

рассматривать в качестве наблюдения столбцы X , то классифицируемыми объектами (в количестве n штук) будут объекты, на каждом из которых было замерено по p признаков $x^{(1)}, x^{(2)}, \dots, x^{(p)}$ характеризующих его состояние. Если же в качестве «наблюдения» рассматривать строки этой матрицы, то классифицируемыми объектами будут уже сами признаки (в количестве p штук), рассматриваемые, соответственно в n -мерном пространстве X^* . В этом случае наличие однородных групп признаков позволяет сделать вывод о близости (коррелированности, взаимном дублировании) признаков, входящих в одну группу и существенно снизить размерность исходного факторного пространства X , оставив для дальнейшего рассмотрения лишь по одному представителю от каждой такой группы.

Применение метода главных компонент (ГК) к ЛСК дало ряд интересных результатов. Возможны следующие подходы к применению метода ГК к ЛСК: 1) преобразовывать «образы» двух, трех фонем, классов фонем; 2) применять преобразование к «образу» каждой отдельной фонемы.

В первом случае мы отображаем пару, тройку или весь класс фонем в новом пространстве (или подпространствах) главных компонент. Предполагалось, что такое преобразование позволит в новом пространстве разделить фонемы независимо от диктора. Такой подход не дал сколько-нибудь хороших результатов. Однако если таким образом преобразовывать образы фонем одного диктора, то можно добиться разделения с достаточно хорошим качеством. Гласные фонемы одного диктора после преобразования методом ГК отображаются в пространстве трех первых ГК в виде отдельных четко разделимых (по крайней мере, визуально) сгущенностей точек.

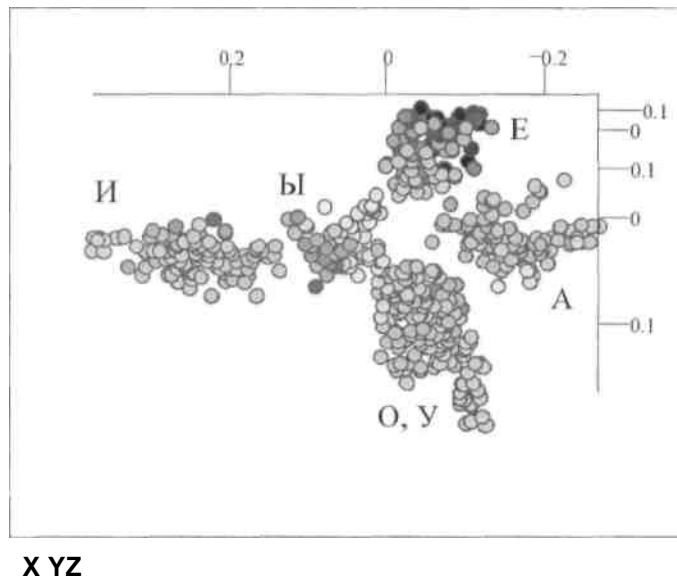


Рис. 2.6.3. Образы всех гласных фонем одного диктора в пространстве трех первых главных компонент.

Исходя из сказанного, можно сделать вывод о том, что такое преобразование применительно к ЛСК вполне позволяет строить систему распознавания реального времени с настройкой на конкретного диктора.

Во втором случае мы получаем «новые» образы фонем, развернутые вдоль главных осей собственных эллипсоидов рассеивания. Используя в качестве исходных данных ЛСК, полученные на 10-м порядке ЛП-модели, мы имеем дело с 10-мерным пространством. Естественно, визуальный анализ в этом случае невозможен. Применим описанный метод и покажем, что для описания фонем достаточно использовать значительно меньшую размерность пространства.

Покажем в качестве примера расчеты применения МГК к образу фонемы «а». Будем считать, что наборы ЛСК, соответствующие фонеме, уже получены (Прил. 2). Для расчета использовались следующие параметры: 10-й порядок ЛП-модели, кадр 120 отсчетов при частоте дискретизации 8 кГц. Для формирования образа использовалось 15 дикторов, общее число кадров 1000.

Для расчета ковариационной матрицы будем использовать центрированные величины ЛСК. Для этого воспользуемся формулой (5). Получим при этом матрицу размером $L \times 10$ центрированных значений ЛСК. Здесь N - число кадров.

Используя выражение (4) рассчитаем ковариационную матрицу для фонемы «а»

$$A = \begin{pmatrix} 0 & I & "2 & ; & 3 - & 4 & 5' & " ; 8 \dots ; & 9 \\ :468.245 & 53.459 & 137.144 & -20.58 & & & -84.942 & 15.504 & 403.51 \\ 211.455 & & & & & & & 130.564 & 126.17 \\ :53.459 & 187.308 & 113.417 & 54.676 & 108.917 & 175.912 & 49.52 & 10.548 & -7.426 \\ 13.364 & & & & & & & & \\ 137 & -11111111 & 616 & ; & ii & \wedge & 1 & . & 67 & 4 & 900 & - & 047 \\ & & & & & & 649.775 & 326.295 & 600.529 & 438.292 & & & 79.295 \\ | & -20.58 & 54.676 & 465.694 & 752.748 & 996.318 & 704.306 & 375.844 & 688.85 & 462.976 & 82.065 \\ 2JM5^{\wedge}[0^{\wedge} \cdot ^{\wedge} 7_900.047 & 996.328 & 2.002 & j0^3 & & & 1.34*10^* & 644.959 & 1.45S10^3 & 1.161.10^3 & 298.556 \\ -84.942 & 175.912 & 649.775 & 704.306 & 1.347.10^3 & 1.527.10^3 & 710.528 & 1.007.10^3 & 628.276 & 82.405 \\ 15.504 & 49.52 & 326.295 & 375.844 & 644.959 & 710.528 & 653.576 & 621.968 & 375.065 & 102.174 \\ | & 130.564 & 13.364 & 600.529 & 688.85 & 1.455.10^3 & 1.007.10^3 & 621.968 & 1.755.10^3 & 1.31\%10^3 & 372.59 \\ 1403.51 & J0.548 & 438.292 & 462.976 & 1.161.10^3 & 628.276 & 375.065 & 1.317.10^3 & 1.583.10^3 & 536.111 & 9 \\ 126.17 & -7.426 & 79.295 & 82.065 & 298.556 & 82.405 & 102.174 & 372.59 & 536.111 & 277.4 \end{pmatrix}$$

Далее с использованием выражений (8) и (9) определяем

собственные числа

$$I \begin{pmatrix} 0 & 12 & 3 & 4 & 5 & 6 & 7 & 8 & " & ' \\ SOB \ ZN \\ I \ 154.159 & 125.436 & 100.638 & 238.799 & 283.166 & 25.199 & 441.189 & 606.244 & 1.43.10^3 & 6.418.10^3 \end{pmatrix}$$

и собственные векторы ковариационной матрицы фонемы

	7 : 8									
B-0.16	-0.441	-0.09	0.135	0.274	0.346	-0.441	0.509	0.32	0.057	
B 0-681	-0.181	0.533	0.163	-0.153	-0.235	-0.281	0.161	-0.092	0.031	
■ 0.267	0.649	-0.345	0.337	0.181	0.036	-0.055	0.395	-0.154	0.241	
Щ0.407	-0.306	-0.183	-0.503	0.218	0.228	0.447	0.188	-0.221	0.267	
MI = 4	-0.474	8.632.10 ⁻³	0.415	-4.85%10 ⁻³	-0.095	-0.277	0.242	0.402	-0.127	0.529
Щ	-0.079	-0.13	-0.214	-6.862.10 ⁻³	-0.484	0.292	-0.415	-0.209	-0.488	0.396
B-0.075	0.102	0.124	-0.292	0.671	-0.242	-0.437	-0.314	-0.193	0.217	
НОЛ08	-0.177	0.07	0.555	0.211	0.183	0.27	-0.463	0.241	0.47	
НО.151	0.045	-0.317	-0.306	-0.275	-0.383	-0.156	-0.048	0.618	0.387	
К 0.041	0.445	0.466	-0.32	-0.087	0.609	-0.066	-0.054	0.29	0.105	

Матрица собственных векторов и является матрицей перехода к главным компонентам (формула 6). После преобразования получаем матрицу новых признаков размером 7Y*10, при этом новые признаки уже обладают всеми замечательными свойствами, описанными выше: они упорядочены по степени рассеяния; первый признак обладает наибольшей степенью рассеяния, т.е. наибольшей дисперсией. Расчеты приведены в приложении №

Переход к главным компонентам дает возможность оценить информативность каждого отдельно взятого признака (ЛСК), выраженную в относительной доле суммарной диспер-

и признаков.

Ниже представлен ряд иллюстраций, показывающих распределение дисперсий ГК (рис. 2.6.4 - 2.6.6). По графикам визуально можно оценить, какой вклад в суммарную дисперсию вносит та или иная ГК. Видно, что наибольшими дисперсиями обладают первые по счету ГК.

Используя выражение (14) построим графики относительной доли суммарной дисперсии, вносимой главными компонентами. По оси абсцисс на графиках (рис. 11-13) отложено число ГК, а по оси ординат - суммарная дисперсия, вносимая этим числом ГК. Исследования показали, что основная доля информации заложена не более чем в первых трех ГК.

Таким образом, применение МГК позволяет сократить размерность исходного признакового пространства с 10-мерного до 3-мерного. Последнее дает возможность визуализации результатов, что является достаточно важным при решении «многомерных» задач, а также позволяет сократить объем вычислений.

Рис. 2.6.6. График распределения дисперсий ГК фонемы «О».

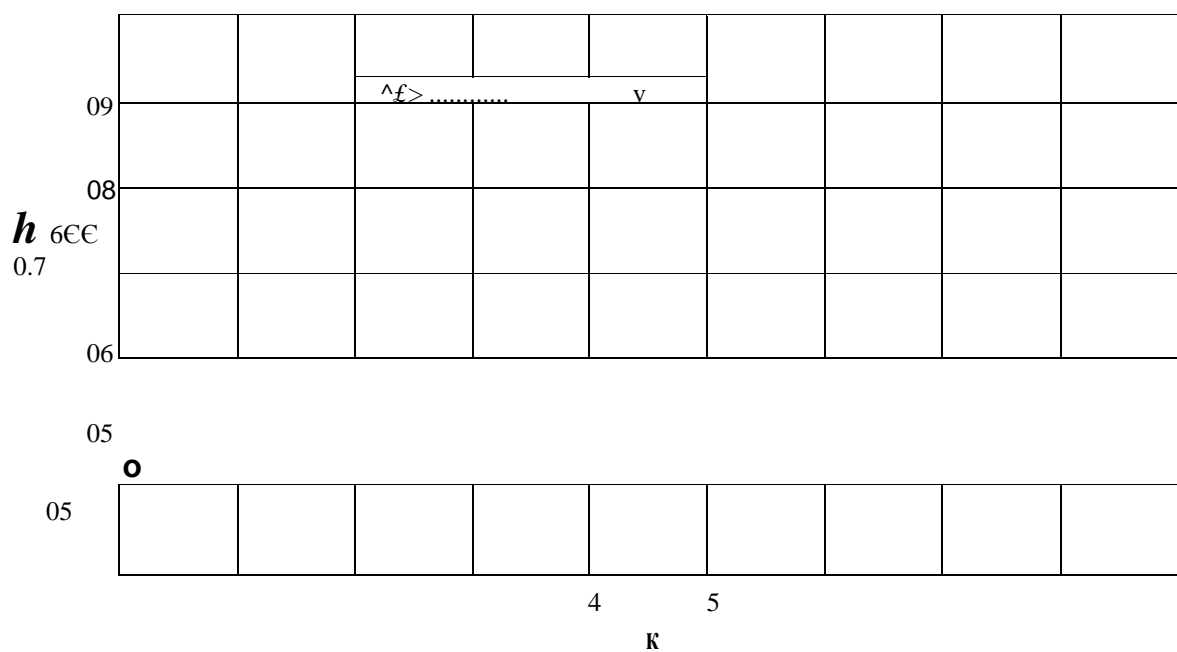


Рис. 2.6.7. Изменение относительной доли суммарной дисперсии главных компонент фонемы "А".

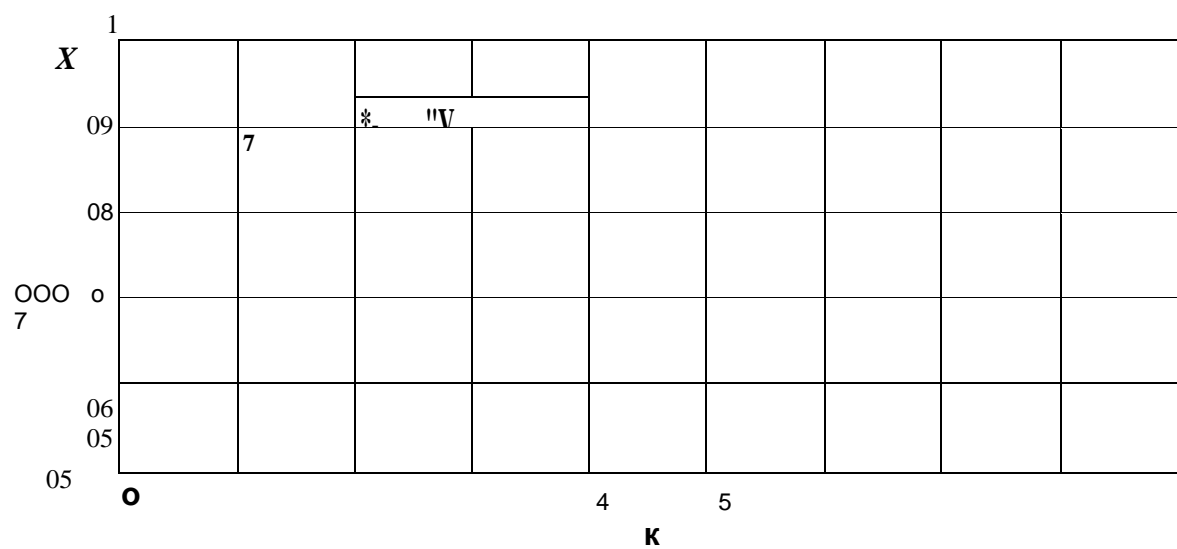


Рис. 2.6.8. Изменение относительной доли суммарной дисперсии главных компонент фонемы "И".

Л
09
08
07
06
05

				$\frac{\lambda}{\lambda_0} -$	$\frac{\lambda}{\lambda_0} <$		λ $\frac{\lambda}{\lambda_0}$	$\frac{\lambda}{\lambda_0}$
		$\frac{\lambda}{\lambda_0}$	$\frac{\lambda}{\lambda_0}$	$\frac{\lambda}{\lambda_0}$	$\frac{\lambda}{\lambda_0}$		$\frac{\lambda}{\lambda_0}$	$\frac{\lambda}{\lambda_0}$
$\frac{\lambda}{\lambda_0}$								

0

--	--	--	--	--	--	--	--	--

4

5

к

Рис. 2.6.9. Изменение относительной доли суммарной дисперсии главных компонент фонемы "О".

2.7 Влияние ранжирования признаков на размерность векторов признаков

Ранее (§§2.5.1, 2.5.2) было показано, что, если в качестве признаков фонем выбрать ЛСК, гистограммы которых имеют наименьшую площадь перекрытия, то фонемы можно разделить при помощи простых (линейных) разделяющих границ. При этом ошибка разделения оказывается сравнительно небольшой (близкой к нулю). Однако оценки эти проводились «вручную» применительно к различным парам фонем, что, с одной стороны, дало «быструю» оценку (результат) и позволило «нащупать» направление дальнейших исследований. Стало понятно, что размерность исходного признакового (10-мерного) пространства можно сократить и, тем самым, сократить требуемые ресурсы вычислителя и уменьшить время принятия решения о принадлежности входной реализации к одному из классов. С другой стороны оставался открытым вопрос формирования конкретных признаковых пространств (или подпространств) для всех фонем или, по крайней мере, для отдельных полных классов фонем (гласные, согласные огласованные и шипящие и т.д.). Поэтому оценки для пар фонем являлись в этом смысле достаточно удобными, но недостаточными. Выяснилось, какое наименьшее количество ЛСК можно оставить в рассмотрении без больших потерь информативности, т.е. насколько (или во сколько) можно сократить размерность вектора признаков фонем. Применение МГК позволило ответить на последний вопрос. Действительно, основная доля информации о фонеме (до 90%) сконцентрирована в 3-х-4-х ЛСК, что говорит о возможности исключения ряда ЛСК из рассмотрения (см. §2.6). Однако, МГК не позволяет определить, какие именно ЛСК нужно исключать.

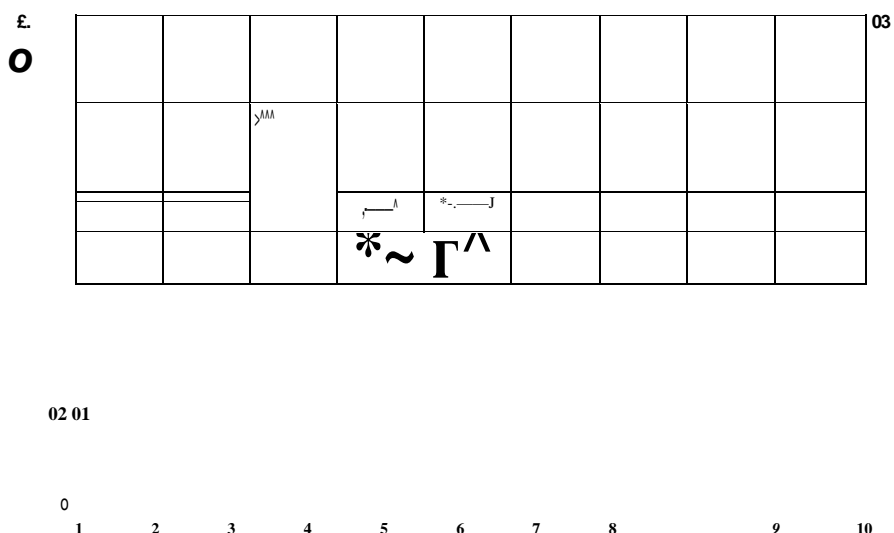
В §2.5.3 было определено, что нет смысла использовать всю десятку ЛСК, а достаточно 6-7-и ЛСК. Следует обратить внимание на то, что при построении зависимостей ошибки разделения от числа используемых признаков на каждой итерации последовательно добавлялись следующие по порядку ЛСК от первого до десятого. Учитывая результаты оценок, приведенных в §§ 2.5.1, 2.5.2, разумно предположить, что если расстановка ЛСК по убыванию

значительно влияет на величину ошибки разделения, то расстановка ЛСК по убыванию ошибки, рассчитанной МДС, также повлияет на величину ошибки разделения. Последнее наталкивает на достаточно простой эксперимент: нужно построить зависимости ошибки разделения от числа используемых признаков, но так, чтобы на каждой итерации добавлялись не следующие по порядку ЛСК от первого до десятого, а следующие по порядку возрастания ошибки разделения.

Результаты таких экспериментов приведены ниже и, для удобства сравнения, «наложены» на результаты, приведенные в §2.5.3. На графиках представлены по две кривых, одна из которых (помечена кружками) представляет зависимость величины ошибки разделения от числа используемых неранжированных ЛСК, а вторая (соответственно с квадратиками) - зависимость величины ошибки разделения от числа используемых ЛСК для ранжированных по возрастанию ошибки ЛСК. Также как и в первом случае, ошибка рассчитана с использованием МДС.



Рис. 2.7.1. а) Фонемы «а-о»



Размерность пространства

Рис. 2.7.1. б) Фонемы «у-о»

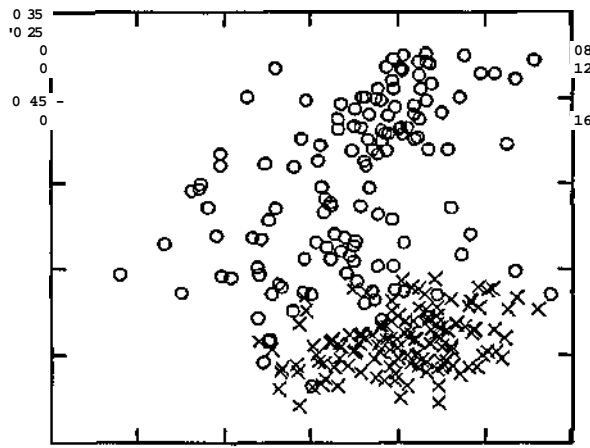


Рис. 2.7.1. в) Фонемы «и-о»

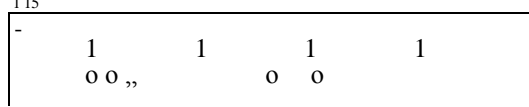
2.8 Оценка влияния параметров расчета ЛСК на качество разделения фонем

Модель расчета ЛСК, предложенная к исследованию в настоящей работе, включает в качестве параметров размер временного окна анализа и его шаг. Можно показать, что влияние этих параметров существенно сказывается на качестве разделения образов фонем пространства ЛСК. Для получения «быстрых» оценок будем считать, что фонемы описываются парой ЛСК вместо 10-ти. С одной стороны это потребует минимума расчетов, с другой - даст возможность визуальной оценки.

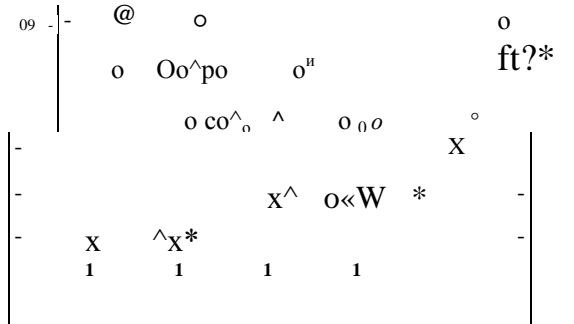
На рис.2.8.1 показаны 6 плоскостей ЛСК, в которых отображена одна и та же пара фонем. Если рисунки анализировать «по столбцам», то можно проследить влияние выбора плоскости ЛСК на качество разделения фонем. Этот вопрос подробно раскрыт в предыдущих параграфах диссертационной главы. Если же проанализировать «строки», то легко обнаружить, что в правом столбце фонемы разделяются лучше, чем в левом. Параметры расчета ЛСК для столбцов различны: для левого ЛСК получены при равных величинах размера и шага временного окна (размер окна = шаг = 120 отсчетов), т.е. окна «пристыковывались» друг к другу - речевой сигнал анализировался кадрами; для правого - размер временного окна оставался 120 отсчетов, а шаг уменьшен до 100 отсчетов, т. е. анализ РС проводился с окном с перекрытием в 20 отсчетов. Хорошо видно, что в последнем случае кластеры стали более компактными и хорошо разделимыми даже в плоскостях ЛСК.



Отображение в плоскости 2-х первых по ' по-
Отображение в плоскости 2-х первых по
рядку ЛСК (1-2)

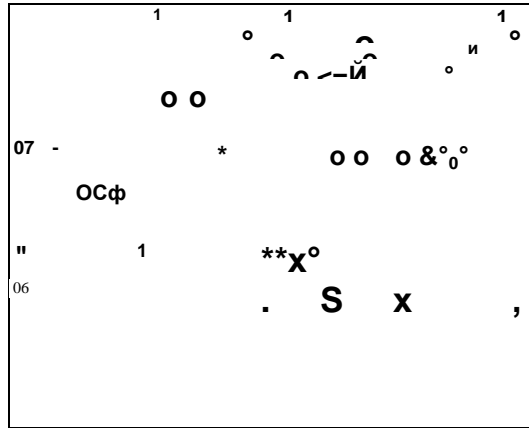


105 -
1 -



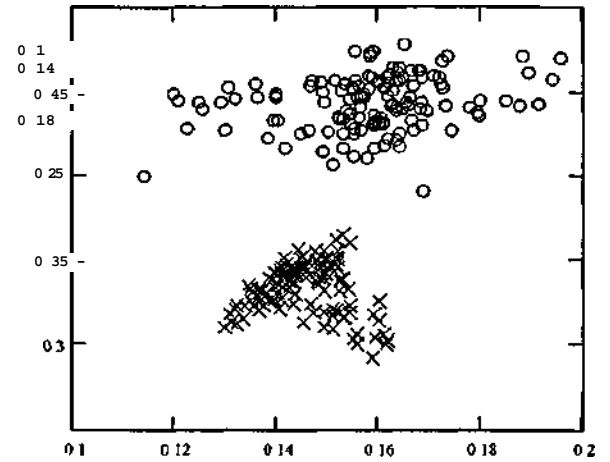
08 008 01 012 014 016 018 02

«Плохой» случай (1-4)



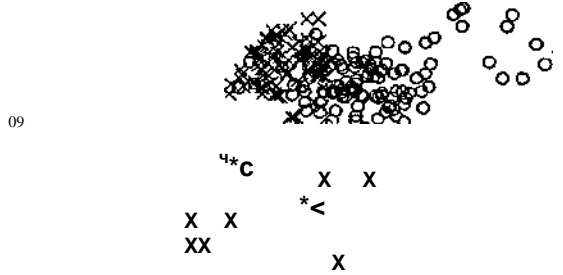
05 11 105 13 14

Отображение в
2-х «наилучших»
ЛСК (3-5)



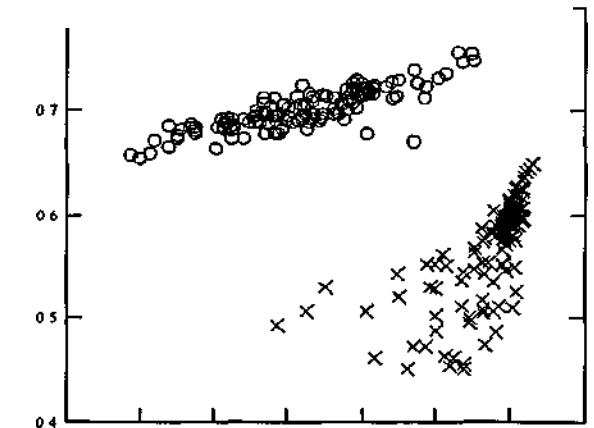
105

02



09
4* C
X X
XX X
X
ft?*

«Плохой» случай (1-4)



01 012 014 016 018 11 115 12 125 13

Отображение в плоскости 2-х «наилучших»
ЛСК (3-5)

Рис. 2.8.1. К оценке влияния размера и шага временного окна на качество разделения фонов

Для того, чтобы определить, на каких параметрах нужно проводить обработку РС, следует решить оптимизационную задачу поиска этих параметров.

Постановка задачи: для выбранного класса фонем нужно найти такую комбинацию

-98-размера временного окна и его шага, при которой разделение фоном окажется наилучшим. В качестве критерия предложено взять минимум функционала S , связывающего параметры расчета ЛСК с величиной ошибки разделения. Функционал S можно записать

$$S=f(N,L,A),$$

где

N - порядок ЛП-модели,

L - размер временного окна анализа в отсчетах (при частоте дискретизации 8 кГц),

L - шаг временного окна анализа в отсчетах,

S - суммарная априорная вероятность ошибки разделения пары фоном, рассчитанная методом динамических сгущений. Метод расчета ошибки подробно описан в § 2.4.2.

Будем считать, что на оптимальных параметрах обработки РС достигается минимум функционала S .

Для решения оптимизационной задачи разработана модель в пакете программ Math-Cad. Модель позволяет рассчитывать значения функционала S в зависимости от размера временного окна и от его шага.

Оценочные расчеты на тестовых РС небольшого числа значений указанных параметров показали, что существуют глобальные оптимумы функционала S , однако расчет значений функционала S для всего диапазона значений размера и шага окна в программе MathCad оказался затруднительным - вычисления занимали слишком много времени даже на мощных вычислительных машинах. Поиск оптимальных параметров расчета ЛСК реализован на языке программирования C++ (прил. 3). Оптимизационная задача решена, а результаты представлены в главе 3 диссертационной работы.

2.9 Выводы

1. Отсутствие перекрытия гистограмм распределений ЛСК в рамках одной фонемы позволяет использовать каждый ЛСК в качестве координаты признакового пространства. Это обстоятельство выгодно отличает предложенные в настоящей работе ЛСК от других известных акустических параметров, в том числе и от ЛСК, рассчитанных другими способами. Последние широко применялись в системах обработки речи, однако не позволили добиться удовлетворительного качества распознавания речи.
2. По площадям перекрытия гистограмм одноименных спектральных корней можно определять априорную ошибку разделения фонем в подпространствах ЛСК, возникающую на этапе обучения системы. Ошибку можно минимизировать, выбрав признаковые подпространства из ЛСК, имеющих минимальные площади перекрытия гистограмм.
3. В основу предложенного в работе метода расчета априорной ошибки классификации фонем положен метод динамических сгущений (МДС). Адаптация метода для задачи распознавания позволила рассчитать суммарную вероятность ошибки классификации фонем в подпространствах ЛСК. С целью решения задачи минимизации размерности анализируемого признакового пространства исследована (с использованием МДС) динамика изменения вероятности ошибки классификации с ростом размерности признакового пространства.
4. В отличие от задачи сжатия-восстановления, задача распознавания не требует последующего восстановления сигнала, что позволяет исключить из рассмотрения ряд параметров (ЛСК), получив при этом еще более компактное описание РС. Для анализа признаков, носящих случайный характер, целесообразно использовать методы, в основу которых положены статистические свойства исследуемых данных. В результате применения преобразования Карунена-Лоэва для оценки информативности ЛСК удалось сократить размерность исходного признакового пространства с 10-мерного до 2-3-мерного (три - четыре раза) без ощутимых потерь информативности вектора описания класса, что, естественно,

-100-упростило алгоритмы классификации и дало выигрыш в скорости принятия решения о принадлежности объекта к одному из классов. Неклассическое использование метода сделало возможной простую графическую интерпретацию результатов анализа признаков: удалось выделить близкие по своим свойствам (дублирующие друг друга) группы ЛСК и заменить эти группы единственными их представителями.

5. Проведенные оценки показали, что параметры вычисления ЛСК (размер временного окна и его шаг) оказывают существенное влияние на взаиморасположение образов фонем и на их компактность в параметрических подпространствах. Необходимо поставить решить оптимизационную задачу, заключающуюся в подборе оптимальных размера временного окна анализа речевого сигнала и его шага.
6. Линейное предсказание как метод параметрического описания речевых сигналов дает акустические признаки фонем, слабо чувствительные к смене диктора. Поэтому такое описание подходит для построения дикторонезависимой СРР.

3 ФОРМИРОВАНИЕ КЛАСТЕРОВ ФОНЕМ НА ОСНОВЕ РАБОЧЕГО СЛОВАРЯ ПРИЗНАКОВ

3.1 Поиск оптимальных параметров расчета ЛСК

3.1.1 Выбор размера временного окна

Учитывая результаты исследований, приведенных в предыдущих параграфах 2 главы (касающихся информативности ЛСК, различной ориентации образов фонем в ПП), исследуем, каким должен быть размер окна анализа (в отсчетах) РС, чтобы фонемы в ПП ЛСК разделялись наилучшим образом. Для этого рассчитаем зависимость вероятности ошибки классификации от размера временного окна для различных пар фонем.

Исследования проведем отдельно для гласных и огласованных согласных фонем. Выбор для анализа именно этих классов фонем обусловлен тем, что такие фонемы образованы при участии голосовых связок, т.е. имеют основной тон. Основной тон дает при расчете спектра этих классов фонем ярко выраженные отличные друг от друга спектральные картины (в отличие от фонем шипящих), на которых хорошо видны локальные максимумы (фор-мантные частоты). Расчет вероятности ошибки классификации именно для пар фонем обусловлен различной ориентацией образов фонем в многомерных пространствах ЛСК. Это показали результаты исследований, приведенных в §2.6.

Введем ограничения на диапазон исследуемых значений размера окна. Выбор размера окна анализа (интервала анализа) включает два фактора: местоположение этого интервала относительно периода основного тона (ПОТ) и его длительность. Хороший выбор длительности интервала зависит от того, анализируется переходный процесс или квазипериодический. Целесообразно проводить анализ внутри интервала, где изменения речевого тракта незначительны. Для большинства звуков этот интервал составляет примерно 15-20 мс. Такой интервал обычно достаточен для получения надежных оценок спектра, хотя в его пределах могут происходить значительные изменения речевого тракта для «скользящих» звуков («в» в

-102-слове вилы). Таким образом приемлемое число отсчетов, используемое на интервале анализа, равно частоте дискретизации в килогерцах, умноженной на 15-20 мс. Абсолютное местоположение во времени интервала в 15-20 мс не будет в большинстве случаев существенно влиять на результаты. Локализация интервала анализа в пределах одного периода порядка ОТ называется анализом, синхронным с ПОТ. Произвольное положение временного интервала (как реализовано в настоящей ДР) называется асинхронным анализом. В работе [11] приведено общее правило выбора временного окна для методов линейного предсказания. Оно состоит в следующем: временное окно целесообразно использовать для интервалов, превышающих 15 мс или включающих несколько периодов ОТ. Для точного анализа переходных звуков желательно брать меньший интервал - около 10 мс.

Известно, период ОТ колеблется в пределах примерно от 4 мс до 9 мс (35 - 70 отсчетов РС при частоте дискретизации 8 кГц) в зависимости от диктора. Поэтому нижняя граница значений этого параметра выбрана 10 отсчетов. Что касается выбора верхней границы, то обоснованием выбора вполне могут служить результаты исследований, приведенных в [И, 13]. В своих работах они уделили достаточно большое внимание вопросу максимального интервала анализа. Результаты исследований говорят о следующем: РС считается стационарным на временном интервале анализа не превышающем 25 мс, что соответствует 200 отсчетам при частоте дискретизации 8 кГц.

Семейство характеристик, показывающих зависимость значения функционала 5 (вероятности ошибки классификации) от размера временного окна (при фиксированном шаге) для различных пар фонем, представлены на рис. 3.1.1.

Большая ошибка разделения наблюдается на значениях размера окна меньших 50 отсчетов и больших 120 отсчетов. Чем больше размер окна, тем более стабильные оценки получаются. Поэтому целесообразно выбрать максимальный размер окна анализа, при котором ошибка еще остается небольшой. В нашем случае целесообразно выбрать окно размером в 120 отсчетов. Увеличение этого значения приведет к потере адекватности модели.

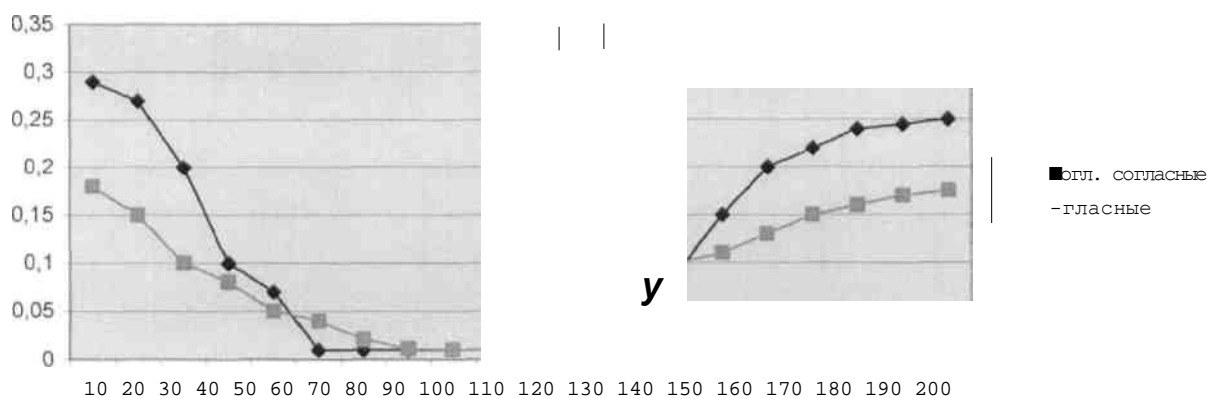


Рис. 3.1.1. Зависимость ошибки разделения фонем от размера временного окна при фиксированном шаге окна 20 отсчетов

3.1.2 Выбор шага временного окна

Рассчитаем зависимости значений функционала $S(L)$ от шага временного окна для различных пар фонем.

Исходя из соображений экономии времени обработки речевого потока (стремимся к обработке в реальном времени) лучше выбрать максимально возможный шаг окна анализа.

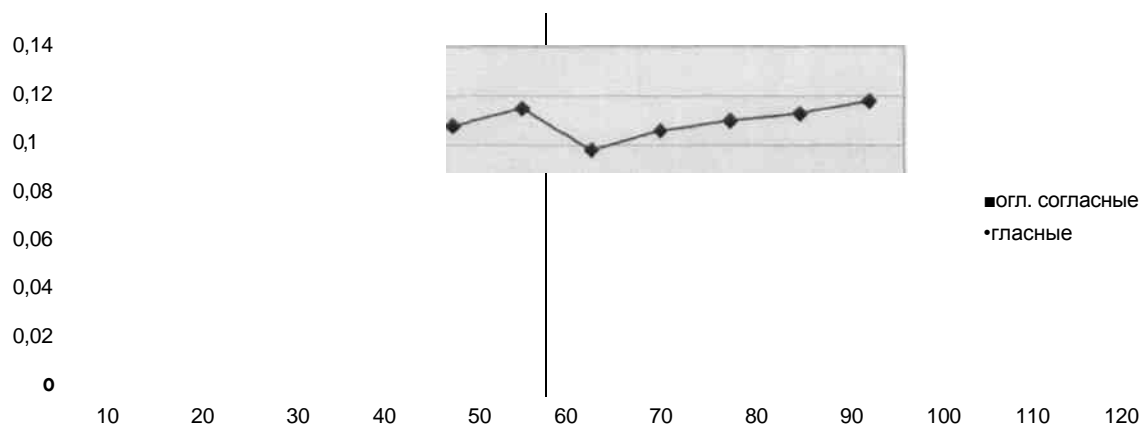


Рис. 3.1.2. Зависимость ошибки разделения фонем от шага временного окна при фиксированном размере окна 120 отсчетов

Эти зависимости могут дать лишь приблизительные значения оптимальных параметров расчета ЛСК. Их точные значения могут быть получены только в результате экспериментальных исследований.

3.2 Формирование признаков подпространств

В [19] описаны основные задачи распознавания объектов и явлений. Одна из основных задач заключается в разбиении пространства признаков на области, соответствующие классам объектов. Подобное разбиение должно быть выполнено так, чтобы при этом обеспечивалось минимальное значение ошибок, неизбежно сопровождающих распознавание поступающих на вход СР неизвестных объектов.

Объекты отображаются в виде точки в многомерном признаковом пространстве. Размерность признакового пространства определяется длиной векторов признаков - количеством замеренных на объекте признаков. На практике стараются использовать минимально возможное число признаков из доступных, выбирая из всего набора признаков наиболее информативные. Это связано со всевозможными ресурсными ограничениями (время, объем памяти и т. д.).

От того, насколько удачно выбрано ПП, зависит качество распознавания (выделения) фоном. В качестве основных требований, предъявляемых к ПП, обычно рассматривают следующие:

- минимальная размерность ПП;
- максимальная компактность образов в выбираемых ПП;
- минимальная степень перекрытия образов в ПП.

Решение задачи формирования ПП позволит перейти к решению следующих по смыслу задач - выбору решающих правил, построению решающих границ и т.д.

3.3 Значения ЛСК - признаки первого уровня

3.3.1 Формирование дерева принятия решений (ДПР) В [1] отмечается, что «речевые фонемы не поддаются простой многоступенчатой классификации, поскольку до сих пор не удалось найти (выявить) наборы признаков, кото-

рые бы позволили последовательно классифицировать неизвестный звук, относя его по признакам к той или иной группе, подгруппе, подподгруппе и т. д. (до исчерпания иерархии)». В этом утверждении существует противоречие: «не найдены» вовсе не означает, что таких признаков не существует.

Во второй главе диссертационной работы было показано, что кластеры фонем в ПП ЛСК представляются достаточно компактными сгущенностями точек. При этом в ряде ПП, размерность которых не превышает 2-3, фонемы могут быть легко разделены с очень небольшой вероятностью ошибки. Качество разделения фонем определяется как параметрами расчетной модели ЛСК (см. предыдущий параграф), так и номерами ЛСК, участвующими в формировании ПП. Последнее подразумевает, что, в зависимости от выбранных номеров ЛСК, меняется степень перекрытия образов фонем в ПП, и, стало быть качество разделения фонем. Выбор ЛСК, формирующих ПП, имеет исключительно важное значение.

Исследования показали, что фонемы могут быть разделены по значениям ЛСК с использованием простой многоступенчатой процедуры классификации - дерева принятия решений (ДПР).

В основу формирования ДПР положен МДС. Подход заключается в поиске такого стартового подпространства (а затем и остальных подпространств для узлов дерева) заданной размерности, в котором суммарная вероятность ошибки (в дальнейшем для сокращения записи будет использоваться термин *ошибка*) разделения двух групп (в любой комбинации) всех существующих в этом подпространстве объектов оказывается минимальной. Задача формирования ДПР разделена на две подзадачи: подзадачу поиска стартового подпространства и подзадачу поиска узловых подпространств.

3.3.2 Поиск стартового подпространства ДПР Выбор стартового подпространства (СтП) будем осуществлять исходя из следующих соображений: во-первых, нужно выбрать наилучшее ПП — в котором кластеры разделяемых объектов имеют минимальную степень перекрытия; во-вторых, нужно выбрать сочетания

классов фонем, которые лучше всего разделяются в выбранном ПП. СтП будем считать определенным, когда будут найдены, во-первых, номера ЛСК, составляющих ПП, во-вторых, сочетания наиболее хорошо разделяемых классов (или их сочетаний) фонем в этом ПП.

Процедура поиска СтП организована следующим образом. Пусть требуется найти СтП для разделения шести объектов, которые описаны десятью признаками. В одной из признакововых плоскостей, например, в плоскости ЛСК с номерами 1 и 3, кластеры фонем могут расположиться как показано на рис. 3.3.1. "hСК №3

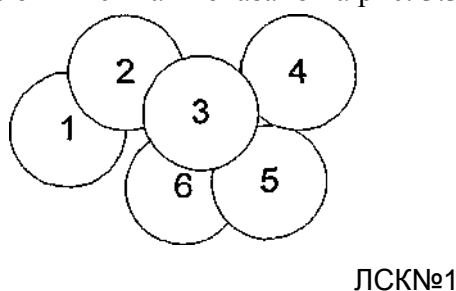


Рис. 3.3.1. Вариант расположения кластеров фонем в признаковом подпространстве ЛСК.

*

Часть кластеров перекрывается между собой, поэтому сначала будет произведена попытка отделить фонему 1 от остальных пяти (ошибка разделения, рассчитанная МДС, будет зафиксирована), затем фонемы 1 и 2 от остальных (также с фиксацией ошибки разделения) и т.д. до полного перебора всех сочетаний классов. В результате будет найдена пара классов фонем, которые наилучшим образом разделяются в текущем ПП. Поиск наилучшего разделения повторяется после смены текущего ПП. В результате полного перебора всех возможных ПП определяется такое ПП и пара сочетаний классов, которые дают наименьшую ошибку разделения.

На рис. 3.3.2 представлен пример определения пары сочетаний классов (следует из примера на рис. 3.3.1): объединенный класс объектов 1 и 2 может быть отделен линейной границей от объединенного класса фонем 4, 5, 6.

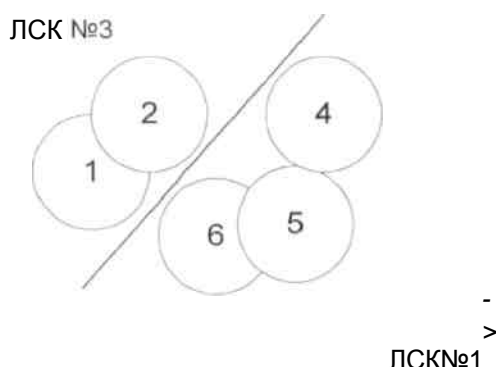


Рис. 3.3.2. Разделение сочетаний классов 1-2 и 4-5-6 линейной границей.

Для стартового пространства ДПР ошибка O рассчитывается по формуле

$$\mathcal{E}(k) = \frac{\sum_{j=1}^n \frac{u_j^2}{N_j} + n}{N; + N;} \quad (8)$$

где k - размерность подпространства; j - номера классов или сочетаний классов k -мерных объектов. N_j , N_j - число точек-представителей классов j и u_j , u_j - число точек-представителей класса j , оказавшихся ближе к центру тяжести класса u_j , u_j - число точек-представителей класса j , оказавшихся ближе к центру тяжести класса i .

Поиск осуществляется полным перебором всех возможных подпространств заданной размерности (их немного, поэтому нет смысла использовать специальные методы, например метод Монте-Карло). Размерность подпространств во всех узлах ДПР и параметры модели расчета ЛСК задаются перед началом поиска в окне программы **Параметры** (рис. 3.3.3).

Рис. 3.3.3. Диалоговое окно установки параметров модели расчета ЛСК и размерности ДПР

Параметры расчетной модели выбираются на основании результатов исследований, приведенных в § 3.2.

3.3.3 *Формирование узлов (переходов) ДПР* Предположим, что ПП, отображенное на рис.

3.3.2 - стартовое. Тогда следующим шагом требуется сформировать узловые ПП, в которых также наилучшим образом будут разделяться остальные классы фонем. В узлах ДПР первого уровня должны разделиться сочетания классов 1,2 - по одной ветке, и 4,5,6 - по другой (рис. 3.3.4).

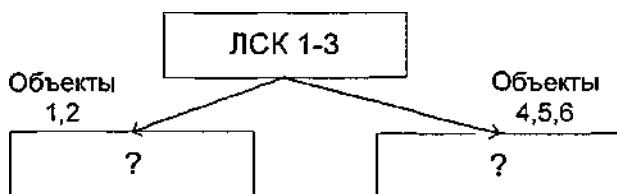


Рис. 3.3.4. Формирование следующего уровня ДПР.

Формирование второго и последующих уровней ДПР (узлов) производится практически по такому же алгоритму, что и формирование СтП. Отличие заключается лишь в том, что на остальных уровнях (в узлах) разделению подлежат оставшиеся классы фонем, которые не были разделены на предыдущих уровнях ДПР. Процедура поиска узловых ПП продолжается до тех пор, пока не будут сформированы все подуровни ДПР, включая последний. На последнем уровне принимается решение об отнесении входной реализации (фрагмента РС) к одному из классов фонем (какая фонема появилась на входе).

Один из возможных вариантов дерева принятия решений показан на рис. 3.3.5. Здесь в качестве стартового пространства выбрано пространство 1 и 2 спектральных корней. Обозначим его 1-2 и далее будем считать такую запись комбинацией 1 и 2 ЛСК2. пространство 1-2 позволяет отделить фонемы /а/, /э/, /о/ от фонем /и/ и /ы/. Следующее пространство 1-3 позволяет отделить фонемы /а/ и /о/ от фонемы /э/. При этом остальные фонемы (/ы/, /и/) в пространстве 1 -3 в рассмотрение не берутся.

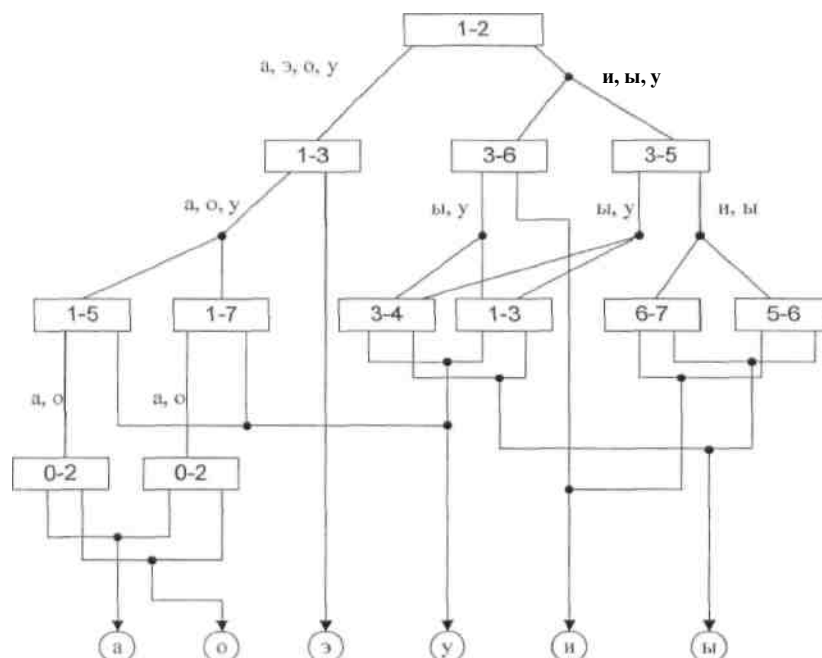


Рис. 3.3.5. Вариант ДПР для разделения гласных фонем.

3.3.4 *Связь размерности подпространств и величины ошибки в узлах ДПР* Во второй главе были приведены результаты исследований, касающиеся возможности сокращения исходного признакового пространства, где было показано, что для описания фонем спектральными корнями достаточно 2-3 наиболее информативных спектральных корней.

э	<p>Буквы: 1 2 3 4 5 6 7 8 9 10 Ошибка : 0,00000</p> <p>ВБуквы: 13 ЛСК: 2 7 Ошибка : 0,02101</p> <p>■ Буквы : 1 ЛСК : 1 2 Ошибка : 0,03230</p> <p>Буквы : 3 ЛСК : 1 2 Ошибка : 0,03230</p> <p>[3Буквы : 245 ЛСК : 2 7 Ошибка : 0,02101</p> <p>3 Буквы: 2 5 ЛСК: 14 Ошибка: 0,03536</p> <p>Буквы: 2 ЛСК: 6 9 Ошибка : 0,05681 Буквы : 5</p> <p>ЛСК : 6 9 Ошибка : 0,05681 Буквы : 4 ЛСК : 1 4</p> <p>Ошибка : 0,03536</p>	<p>Буквы: 1 2 3 4 5 6 7 8 9 10 Ошибка : 0,00000</p> <p>Буквы: 134 ЛСК: 1 2 3 4 5 7 8 9 10 Ошибка: 0,01676</p> <p>В- Буквы: 14 ЛСК: 1 2 3 4 5 6 8 9 10 Ошибка: 0,04640</p> <p>Буквы: 1 ЛСК: 1 2 3 5 6 7 8 9 10 Ошибка: 0,04728</p> <p>.....Буквы: 4 ЛСК: 1 2 3 5 6 7 8 9 10 Ошибка: 0,04728</p> <p>Буквы: 3 ЛСК: 1 2 3 4 5 6 8 9 10 Ошибка: 0,04640</p> <p>Буквы: 25 ЛСК: 1 2 3 4 5 7 8 9 10 Ошибка: 0,01676</p> <p>.....Буквы: 2 ЛСК: 1 2 3 4 6 7 8 9 10 Ошибка: 0,02869</p> <p>Буквы: 5 ЛСК: 12 3 4 6 7 8 9 10 Ошибка: 0,02869</p>
а)	б)	

Рис. 3.3.6. Варианты деревьев принятия решений с различными размерностями узловых подпространств: а) размерность подпространств 2; б) размерность подпространств 9

Если сравнить деревья принятия решений, представленные на рис. 3.3.6, то легко увидеть незначительное уменьшение значений ошибок в узлах ДПР с увеличением размерности ПП в узлах дерева с двух до девяти. Это еще раз подтверждает результаты, приведенные во 2 главе, которые говорят об отсутствии необходимости использования всех 10 ЛСК для описания фонем. Увеличение размерности ПП в узлах ДПР приводит лишь к увеличению требуемого расчетного времени.

ДПР для распознавания фонем позволяет ввести избыточность на каждом уровне в том случае, если вероятность ошибки в какой-то из ветвей дерева окажется достаточно большой. Последнее предполагает возможность создания параллельных «древовидных» структур принятия решений (рис 3.3.7).

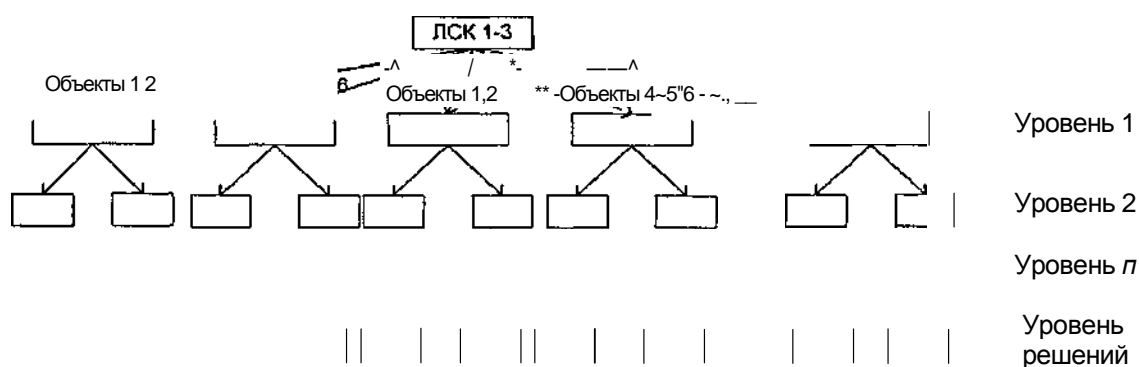


Рис. 3.3.7. Введение параллельных ДПР для **повышения** надежности **распознавания**.

Предложенное построение процедуры разделения позволяет ввести избыточность в алгоритм принятия решения и, таким образом, повысить вероятность правильного распознавания на акустическом уровне.

На основе ДПР могут быть разделены гласные фонемы с априорной вероятностью ошибки не превышающей 6%-го порога.

3.4 Признаки второго уровня

Временные диаграммы ЛСК (рис. 1) показывают явное различие в поведении ЛСК при смене фонем. На диаграммах четко прослеживаются границы фонем по большинству ЛСК. Исследования статистических характеристик ЛСК фонем позволили сделать выводы об их изменениях при переходе от фонемы к фонеме.

В качестве признаков фонем предлагается использовать значения параметров распределений ЛСК.

3.4.1 Формирование кластеров фонем на основе скользящих средних ЛСК

Исследования, проведенные во второй главе, показали четкие различия средних зна-

«одноименных» ЛСК у различных фонем (табл. 2.3.1).

Основываясь на логичном стремлении сделать кластеры объектов максимально компактными и, опираясь на то, что выборочное среднее случайной величины всегда имеет меньший разброс, нежели чем сама случайная величина, предлагается использовать в качестве признаков фонем не непосредственно значения ЛСК, а значения их скользящих средних (СС ЛСК). Предполагается, что используя СС ЛСК, можно добиться меньших радиусов кластеров фонем и, как следствие, меньшей вероятности ошибки распознавания.

Формула для расчета СС спектрального корня с номером k имеет вид

где L - величина усреднения (число ЛСК, по которым производится усреднение), A_{i-k} - значение корня с номером k ($k = p = 1, 2, \dots, 10$).

В процедуре расчета СС используется формула

$$M_{j,m} = \frac{1}{m+L} \sum_{i=m}^{m+L} Y_{ij} A_{ij}$$

где M_{jm} - m -е значение СС спектрального корня с номером/

Значение параметра L в данном случае выступает в качестве параметра модели и требует дополнительных исследований.

Качественное влияние L можно показать на примере, выбрав для сравнения ПП минимальной размерности (для простоты визуальной оценки). Возьмем любую плоскость ЛСК и отобразим в ней точки-представители фонем (рис. 3.4.1). Хорошо видно, что с увеличением параметра L кластеры фонем становятся более компактными.

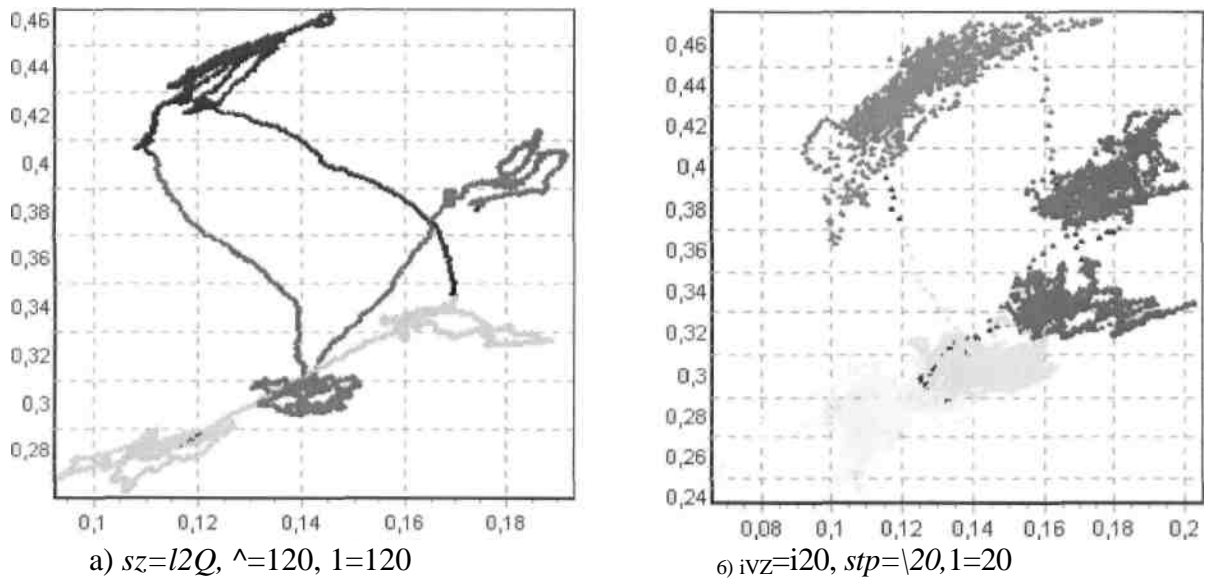


Рис. 3.4.1. а, б - кластеры фонем в пространствах скользящих средних ЛСК.

Выбор конкретного значения величины усреднения следует осуществлять с учетом максимальной и минимальной длительности фонем, определенных (найденных) ранее параметров расчета ЛСК - размера временного окна и его шага. И то и другое несомненно сильно связано между собой, например чем меньше выбирается шаг временного окна, тем больше может быть значение параметра усреднения, и наоборот.

Влияние размера временного окна анализа РС на свойства кластеров фонем

Известно, что в задаче формирования кластеров на основе предлагаемых признаков объектов (в рамках общей задачи распознавания) на первое место выдвигаются такие свойства признаков, использование которых приводит к максимальной компактности кластеров и к максимальному межкластерному расстоянию в пространствах этих признаков. При переходе от 10-мерных пространств к подпространствам меньшей размерности очень важным является правильный выбор подпространств и правильный выбор параметров расчета ЛСК (что было показано в предыдущих разделах работы), от которых зависят указанные выше свойства кластеров. Переход от значений ЛСК к их скользящим средним требует дополнительных исследований, направленных на поиск подходящих параметров расчета СС ЛСК - шага окна анализа, его размера и величины параметра усреднения.

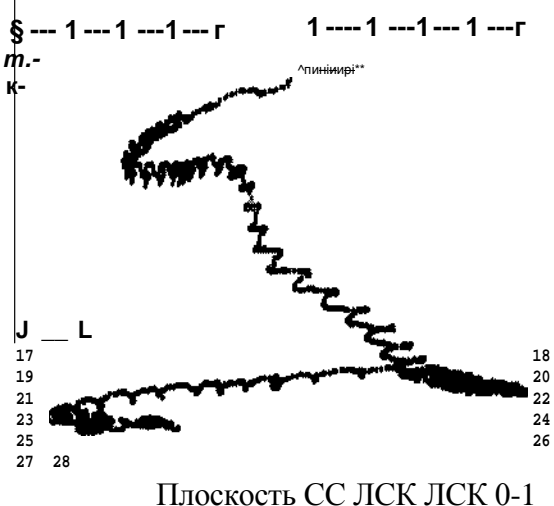
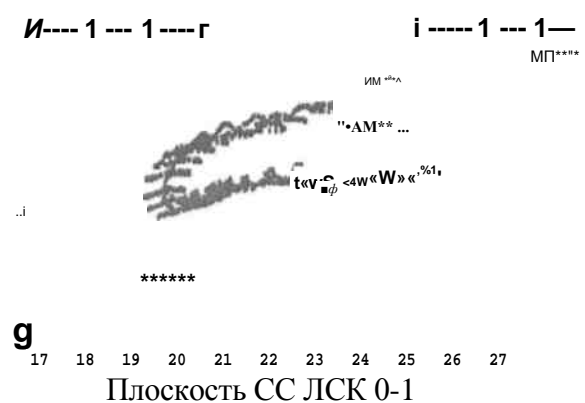
Проиллюстрировать влияние размера временного окна на указанные выше свойства кластеров можно на нескольких примерах, представленных на рис. 3.4.2. На рисунке изобра-

-113-жены плоскости, по осям которых в качестве координат отложены значения СС ЛСК. Каждая точка в этих плоскостях отображает участок речевого сигнала по длительности равный произведению размера окна анализа на величину параметра усреднения. СС ЛСК рассчитаны при фиксированных параметрах размера окна sz , шага окна stp и параметра усреднения L . В левом столбце СС ЛСК рассчитаны с параметрами $sz = 100$, $stp = 2$, $Z = 100$, в правом с параметрами $sz = 120$, $stp = 2$, $\xi = 100$. Даже при небольшой разнице значений параметра sz (100 и 120) хорошо видно, что увеличение размера окна анализа приводит большей стабилизации точки-представителя, траектория точки более плотно «концентрируется» вокруг некоторой предполагаемой средней линии. Такой же эффект стабилизации прослеживается и в случае когда кластеры фонем формируются в подпространствах ЛСК. Другими словами, при использовании в качестве признаков фонем СС ЛСК, размер окна анализа следует выбирать максимально возможным. Вопрос ограничения размера окна анализа сверху был рассмотрен в главе 2 (§ 2.2 и § 2.8) и в главе 3 (§ 3.2). Таким образом, размер окна анализа для 10-го порядка ЛП-модели не может превышать 120 отсчетов.

Оценка влияния шага окна на свойства кластеров фонем

Проводя оценки величины шага окна анализа, следует отталкиваться от реальной длительности фонем. Для фонем, в образовании которых принимают участие голосовые связки (присутствует ОТ), длительность периода ОТ колеблется в пределах примерно от 4 мс до 9 мс (35 - 70 отсчетов при частоте дискретизации 8 кГц) в зависимости от диктора. Минимальная длительность гласных и огласованных согласных фонем составляет примерно 6-8 периодов ОТ. Из этого следует, что длительность одной фонемы в отсчетах при частоте дискретизации 8 кГц не превышает 600. Если при этом выбрать шаг окна, равный 120 отсчетам, то фонема в подпространствах ЛСК будет представляться лишь пятью точками. Такой размер шага нельзя выбирать по двум причинам. Во-первых, результаты исследований, приведенные в § 3.1.2 говорят о том, что анализ речевого сигнала должен проводиться окном с перекрытием, т. е. размер окна анализа должен быть значительно больше его шага. Во-вторых,

очень малое количество точек-представителей фонемы ПП ЛСК не позволит получить достаточно стабильные центры тяжести классов. Исходя из вышесказанного, нужно стремиться к тому, чтобы число точек-представителей фонем в ПП ЛСК было достаточным. На рис. 3.4.3 проиллюстрировано влияние шага окна анализа на компактность кластеров фонем. На рисунках представлены кластеры фонем в плоскостях 0-1 СС ЛСК. СС ЛСК рассчитывались при фиксированном значении размера окна и различных значениях шага окна. Хорошо видно, что увеличение шага окна от 2 до 60 (четыре верхних плоскости СС ЛСК) оказывает очень сильное влияние на форму кластеров (и на их компактность). При шаге окна, равном 2, в плоскостях СС ЛСК хорошо различимы кластеры трех фонем. При шаге окна равном 10 и 20, кластеры фонем уже представляют собой не сгущенности точек, а траектории точек. При шаге окна равном 60 отсчетам картина сильно меняется, и по ней уже нельзя определить положения кластеров фонем.



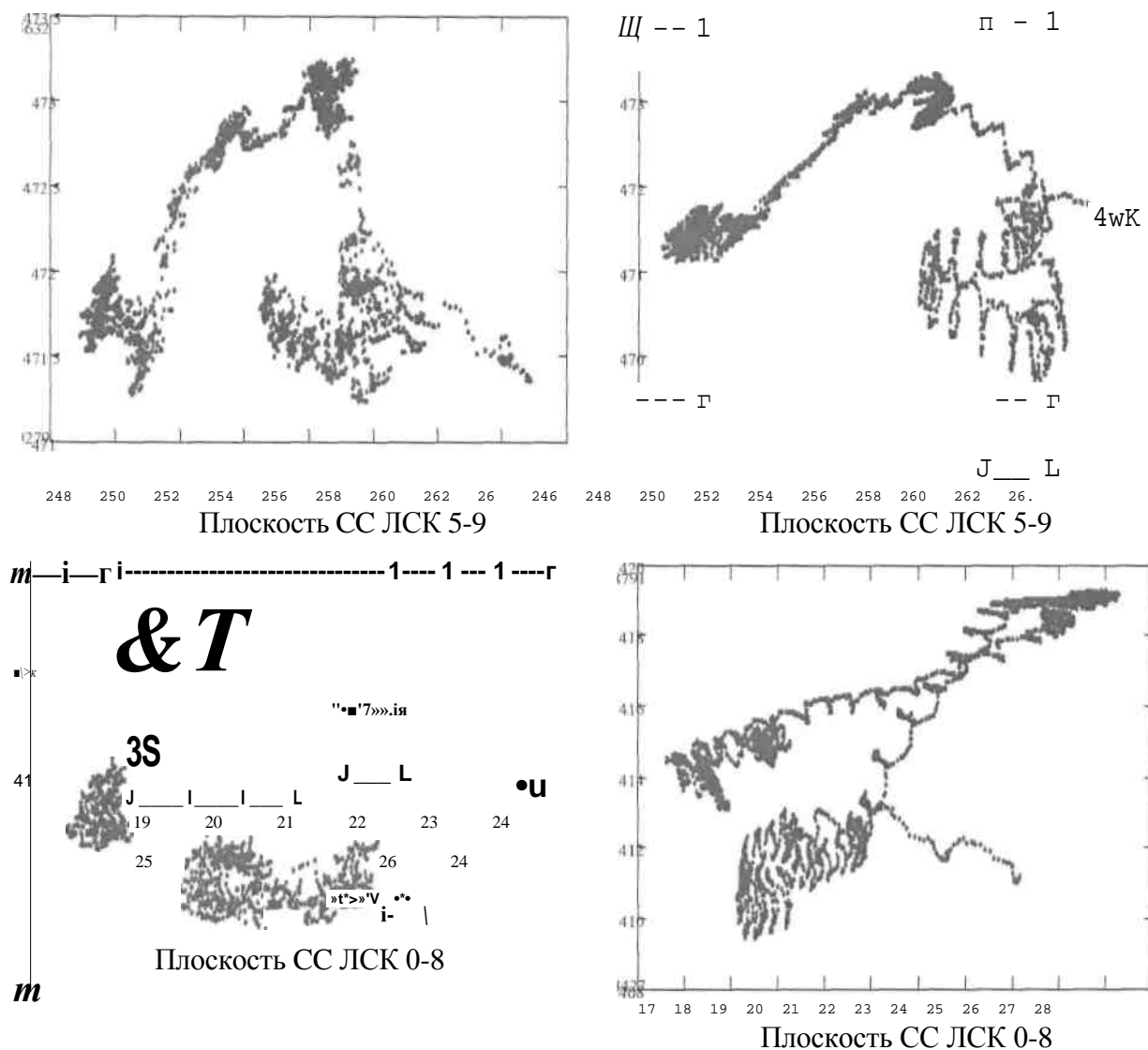
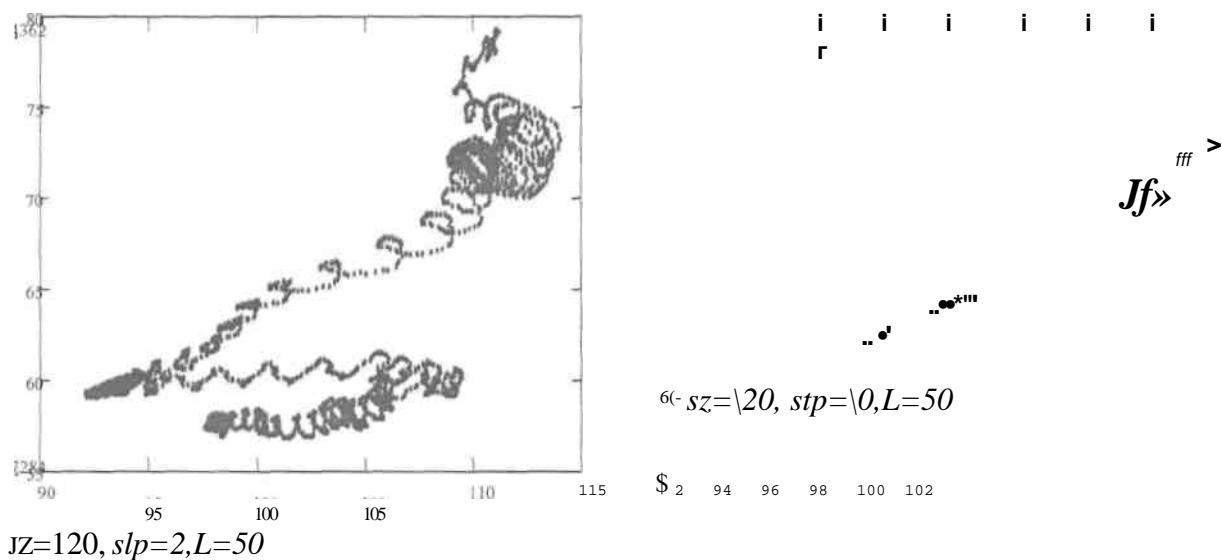


Рис. 3.4.2. Иллюстрация влияния размера временного окна на компактность кластеров фонем «а-о-ы».
 Параметры расчета: $sz=100, stp=2, \text{ } \text{ } = 100$ $sz=120, stp=2, \text{ } \text{ } = 100$



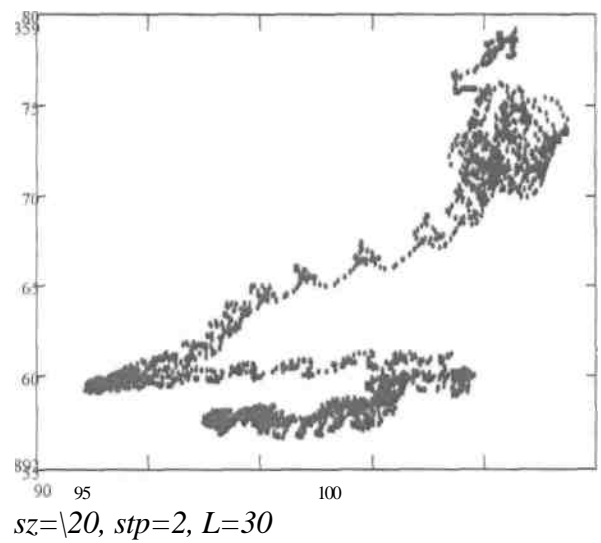
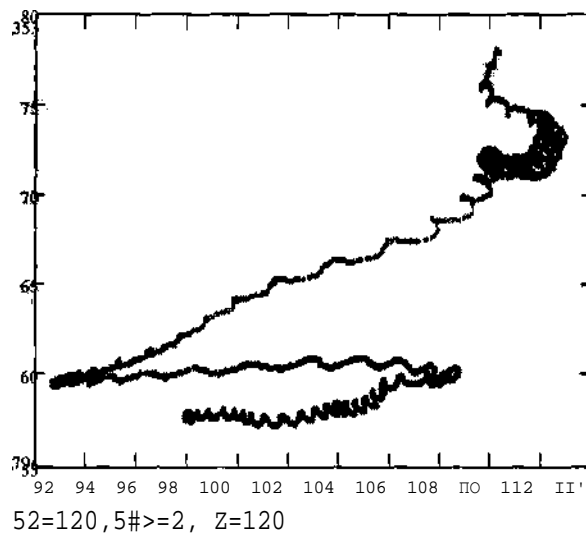
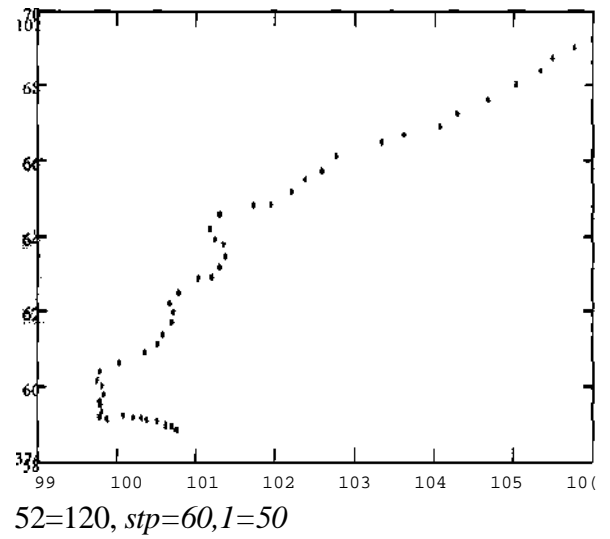
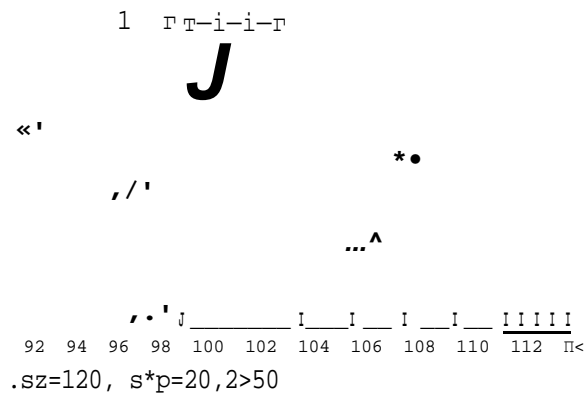


Рис. 3.4.3. Иллюстрация влияния шага временного окна на компактность кластеров фонем «а-о-ы». На всех иллюстрациях показана плоскость СС ЛСК 0-1.

Оценка влияния параметра усреднения

Сильное влияние на свойства кластеров (такие, как компактность, взаиморасположение в признаковых подпространствах) оказывает величина параметра усреднения L , что иллюстрируется примерами на рис. 3.4.4. В плоскости 0-1 СС ЛСК отображены кластеры всех гласных фонем: «а-и-о-у-ы-э». СС ЛСК рассчитаны при различных значениях размера окна анализа, шага окна и усреднения. Основное влияние величины параметра усреднения оказывается на степень компактности кластеров. Не трудно предположить, что с ростом значения параметра усреднения кластеры становятся более компактными.

Выбор величины L следует осуществлять исходя из полученных экспериментальным путем минимальной и максимальной длительностей фонем. Слишком большое значение параметра усреднения L может привести к тому, что в расчет скользящего среднего окажутся включенными отсчеты РС, относящиеся к другой фонеме или переходному участку (дифтонгу). Увеличение значения L на единицу равнозначно тому, что анализируемый участок РС станет больше на величину, равную сумме значений размера окна анализа и шага окна ($sz + step$). Увеличение значения L приведет к необходимости уменьшить шаг окна анализа (для сохранения такого же числа точек-представителей фонем в подпространствах СС ЛСК), что вытекает из ограниченной длительности звучания фонем сверху.

Значения параметров расчета СС ЛСК, показанные на рисунках, значительно больше тех, которые будут реально использоваться при обработке речи. Здесь кластеры фонем получены в результате обработки больших отрезков речевого сигнала с целью исследования влияния вышеуказанных параметров на свойства кластеров фонем.

На приведенных примерах (рис. 3.4.4) можно также проиллюстрировать одно замечательное свойство СС ЛСК. Оно заключается в том, что кластеры всех гласных фонем компактно размещаются в плоскости (!) СС ЛСК и при этом очень хорошо разделимы. Это дает повод исследовать возможность разделения фонем, представляемых СС ЛСК, в едином для всех фонем подпространстве СС ЛСК. Такие исследования проведены в рамках диссертаци-

онной работы. Результаты исследований будут приведены в главе 4.

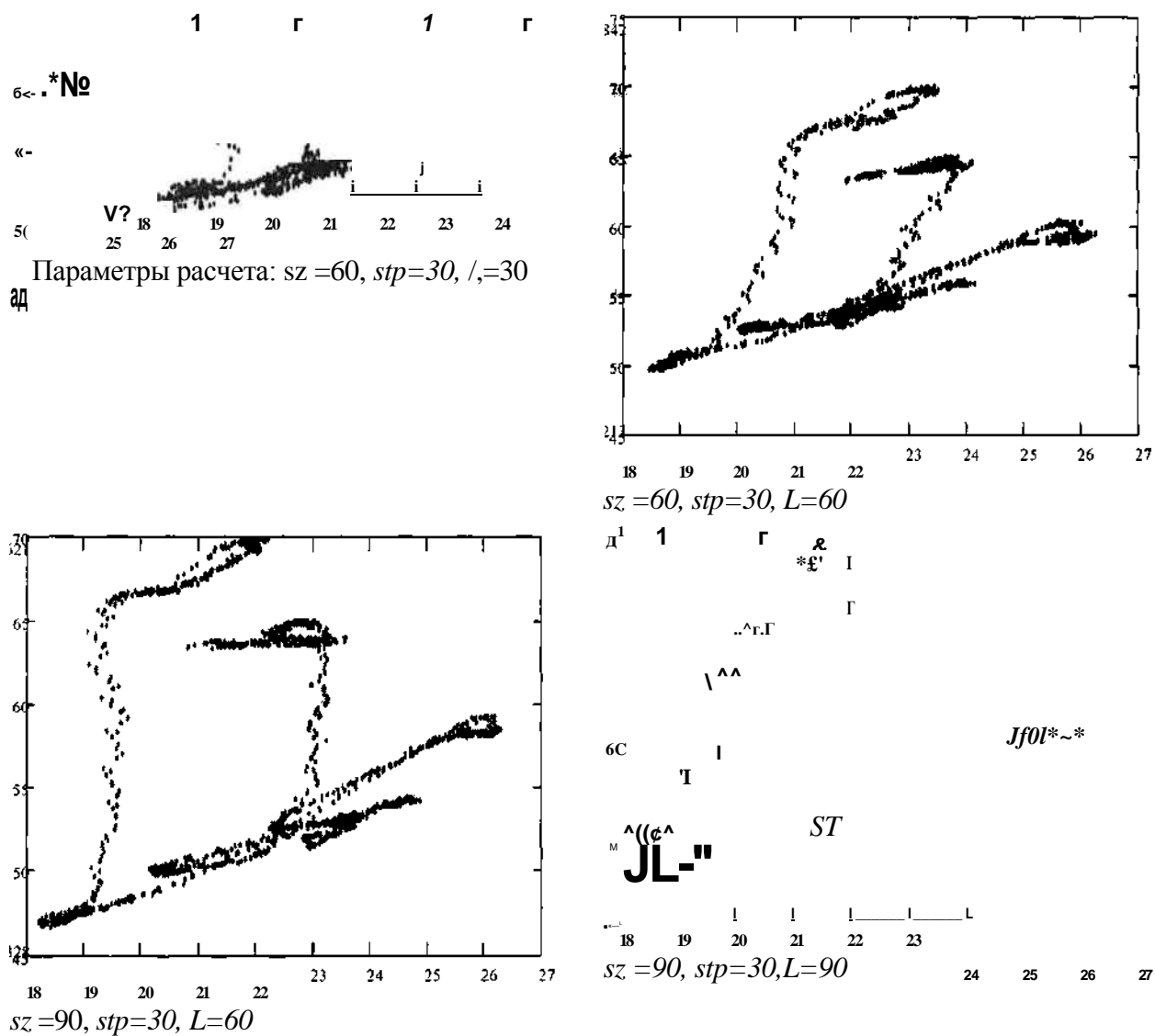


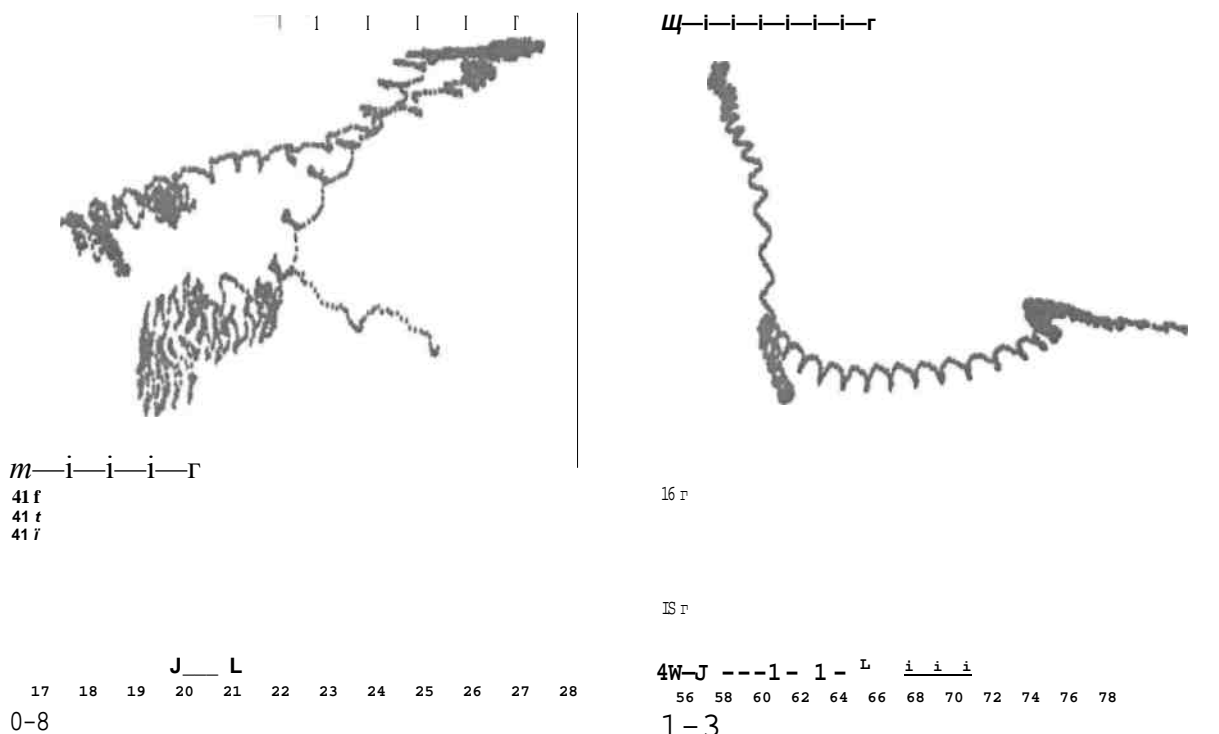
Рис. 3.4.4. Иллюстрация влияния параметра усреднения. Плоскость 0-1 «а-и-о-у-ы-э».

Влияние выбора подпространств

Влияние выбора подпространств достаточно подробно описано в работе в разделах, посвященных исследованию свойств ЛСК. Здесь следует обратить внимание на то, что практически в любой плоскости СС ЛСК (рис. 3.4.5) кластеры фонем легко разделяются.

Естественно, что и в этом случае - когда кластеры фонем формируются в подпространствах СС ЛСК - выбор подпространств имеет определяющее значение с точки зрения вероятности ошибки классификации. На этапе разработки алгоритма распознавания фонем обязательно потребуется программная реализация процедуры автоматического поиска (выбора) наилучших (в смысле минимума вероятности ошибки классификации) подпространств СС ЛСК. Такой поиск программно реализован и является частью программы обучения для одного из предложенных в работе методов распознавания фонем (метод распознавания на основе СС ЛСК описан и исследован в главе 4).

Определение точных значений параметров расчета СС ЛСК (размера окна, шага окна, величины параметра усреднения) возможно только посредством проведения экспериментальных исследований. Результаты исследований приводятся в главе 4.



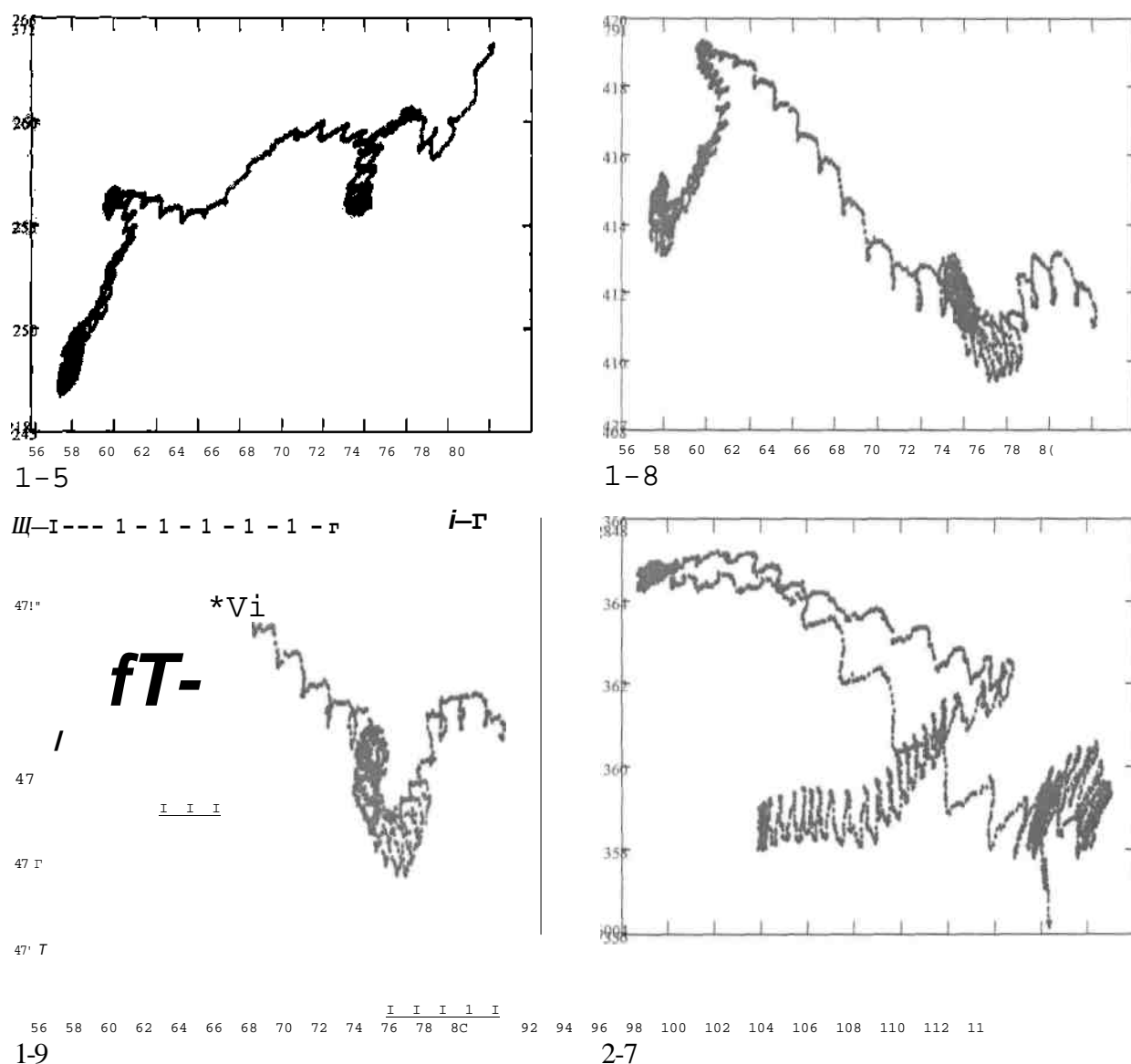


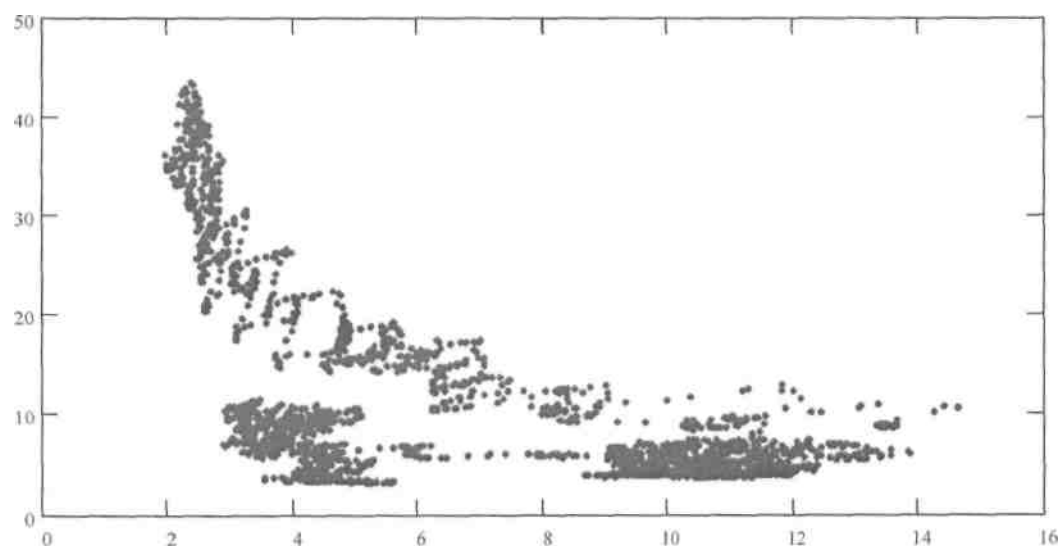
Рис. 3.4.5. Иллюстрация влияния выбора подпространств СС ЛСК. Параметры расчета ЛСК: размер окна 120 отсчетов, шаг окна 2 отсчета, величина параметра усреднения 120.

3.4.2 Скользящие дисперсии как дополнительные признаки фонем Исследования статистических свойств ЛСК (глава 2) показали, что при переходе от фонеме к фонеме меняются не только средние значения ЛСК, но также и их дисперсии. На рис. 3.4.6 отображены точки-представители трех гласных фонем в плоскостях скользящих дисперсий ЛСК. Как и в предыдущих параграфах, для простоты визуальной оценки, используются плоскости.

Для расчета скользящих дисперсий использованы следующие параметры: размер временного окна анализа 120 отсчетов, шаг временного окна 2 отсчета, величина параметра ус-

- 121-реднения 120 (частота дискретизации 8 кГц). В плоскостях скользящих дисперсий можно выделить сгущенности точек-представителей, соответствующие фонемам (кластеры). Кластеры достаточно сильно «размыты», но при этом между собой не перекрываются. Это дает основание полагать, что значения скользящих дисперсий также можно использовать в качестве признаков фонем.

В рамках настоящей не ставилась задача разработки метода распознавания фонем с использованием скользящих дисперсий ЛСК в качестве дополнительных признаков фонем. Однако проведенные оценки вполне могут служить основой для дальнейших исследований.



а)

30

25 -

20 -

15

10



i^*

u, i

10

$i:$

14

10

б)

Рис. 3.4.6. Кластеры фонем в плоскостях скользящих дисперсий ЛСК: а) плоскость скользящих дисперсий ЛСК 0-1; б) плоскость скользящих дисперсий ЛСК 0-2.

3.4.3 Собственные векторы ковариационных матриц фонем — альтернативные признаковые подпространства

Приведенные в предыдущей главе результаты показали, что главные компоненты фонем различаются и этот факт является следствием различия статистических свойств ЛСК разных фонем. Представленные ниже рисунки иллюстрируют это.

Пересчет исходных признаков (ЛСК) фонем в главные компоненты дает совершенно разные значения главных компонент для всех фонем (рис. 3.4.7), что, теоретически, позволяет использовать главные компоненты в качестве признаков фонем и формировать признаковые пространства на их основе.

Метод главных компонент применялся для обработки образа каждой фонемы в отдельности, что в результате давало представление о форме многомерного облака - образа объекта (фонемы). Конфигурации N -мерных облаков (N - порядок ЛП - модели) различны, т. к. они тесно связаны с формой речевого тракта человека. Использование «собственных» главных компонент в процедуре обучения и распознавания интересно, хотя и не позволяет создать единого (общего) многомерного пространства (или подпространств) для отображения всех фонем, в котором можно было бы строить решающие границы для разделения фонем. Сведение (отображение или анализ) кластеров, преобразованных к главным компонентам, в одно координатное пространство невозможно - теряется физический смысл.

Результаты исследований показывают, что дифференцировать речевые единицы позволяют компоненты собственных векторов ковариационных матриц РС. Собственные векторы ковариационных матриц в данном случае являются векторами перехода от исходных признаков фонем (ЛСК) к главным компонентам. При этом кластеры фонем формируются в новых пространствах - пространствах значений собственных векторов ковариационных матриц фонем. Отличие значений главных компонент фонем (рис. 3.4.7.) подтверждает тот факт, что главные оси рассеивания кластеров фонем имеют отличные друг от друга направления в многомерных признаковых подпространствах ЛСК.

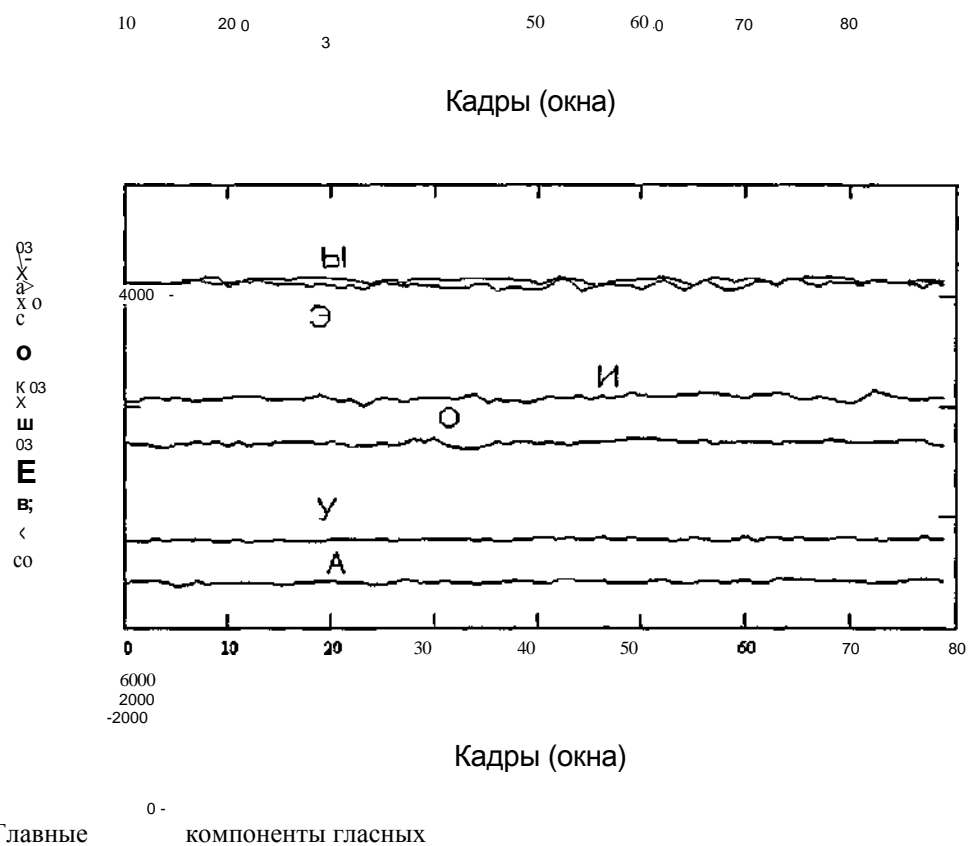


Рис. 3.4.7. Главные фонем.

-124-На рис. 3.4.8 и 3.4.9 приведены примеры

образов фонем, представленных в плоскостях компонент собственных векторов ковариационных матриц фонем. Такой подход к формированию признаков подпространств может рассматриваться как альтернатива уже описанным в работе подходам.

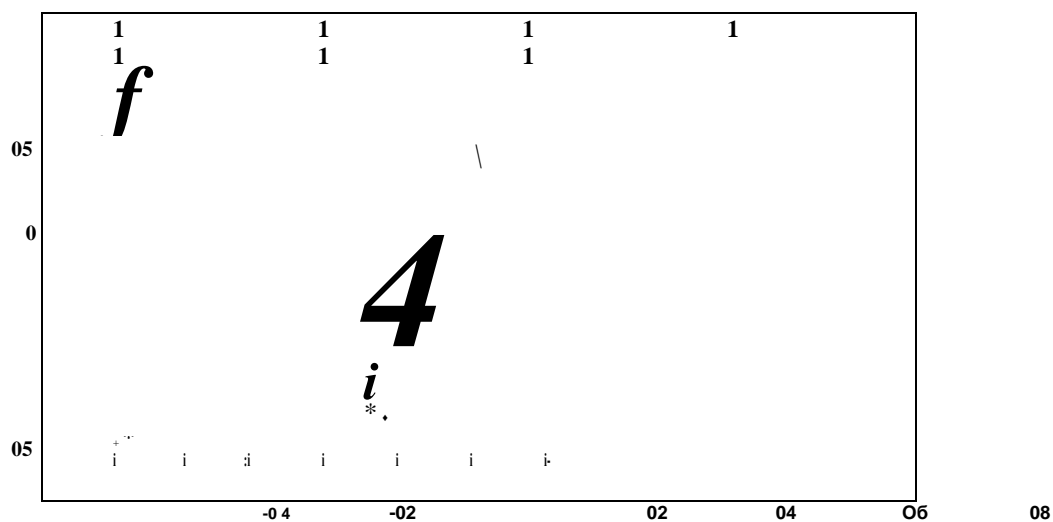


Рис. 3.4.8. Фонемы «А», «И». Плоскость первых двух значений первого собственного вектора.

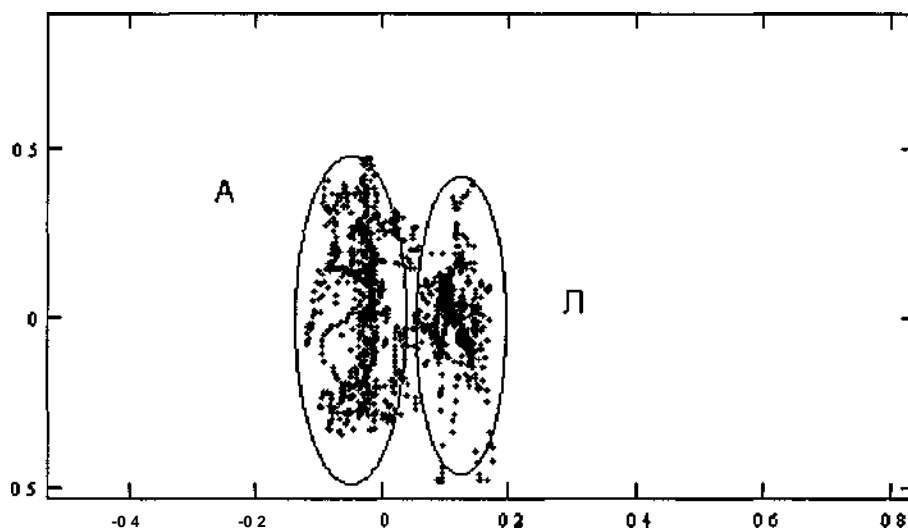


Рис. 3.4.9. Слово «АЛЛА». Плоскость первых значений 1-го и 2-го собственных векторов.

3.5 Выводы

1. Предложены методы формирования кластеров с использованием акустических признаков фонем первого уровня (ЛСК) и признаков второго уровня (скользящие средние ЛСК, скользящие дисперсии ЛСК, значения собственных векторов ковариационных матриц фонем).
2. Предложены методы распознавания фонем:
 - на основе многоступенчатой классификации (дерева принятия решений) по значениям ЛСК;
 - на основе максимального сходства с эталоном по значениям скользящих средних ЛСК;
 - на основе многоступенчатой классификации (дерева принятия решений) по значениям скользящих средних ЛСК.
3. Разработан программный комплекс - инструмент для анализа ЛСК. Реализованы в виде модели для ЭВМ основные (предложенные в работе) методы анализа речевого сигнала. С использованием этих методов программно реализованы:
 - автоматический подбор параметров ЛП-модели и подпространств ЛСК, оптимальных в смысле ошибки распознавания;
 - расчет функций зависимости вероятности ошибки от параметров расчета ЛСК;
 - процедуры формирования кластеров с использованием признаков фонем первого уровня (ЛСК) и второго уровня (скользящие средние ЛСК, значения собственных векторов ковариационных матриц фонем);
 - процедура формирования дерева принятия решений с параметром размерности узловых подпространств. При этом определяется вероятность суммарной априорной ошибки принятия правильного решения для каждого узла дерева.
 - процедуры обучения и распознавания.

-126-4. Осуществлен выбор оптимальных (в смысле минимума ошибки распознавания) параметров вычисления ЛСК для всех предложенных методов распознавания при помощи разработанной программной параметрической модели для ПЭВМ.

РАСПОЗНАВАНИЕ ФОНЕМ НА ОСНОВЕ ЛСК

Устойчивая коммутация, при которой слушатель без напряжения понимает сказанное, возможна при фонемной разборчивости 80-90%, а в ряде ситуаций (например, при ограниченном лексиконе или хорошей предсказуемости высказывания) достаточно 70-80% [1]. В целях экономии энергии говорящий и не стремится к более качественной артикуляции. Следовательно, ошибки идентификации звуков на фонемном уровне есть не какой-то досадный, редко встречающийся фактор, а неотъемлемая закономерность речевой коммутации. Значит, полезные (действующие) модели восприятия устной речи должны учитывать этот фактор ошибок и включать в себя процедуры исключения (нейтрализации) этих ошибок на более высоких уровнях распознавания.

Задача оценки эффективности технической системы состоит в определении степени соответствия системы ее назначению. Количественные характеристики этого соответствия называют критериями эффективности системы. Одним из наиболее существенных критериев эффективности системы распознавания является вероятность получения правильных решений при распознавании объектов, относящихся к различным классам. Этот критерий зависит от следующих факторов: от ошибок априорного описания классов, определения признаков распознаваемых объектов, количества и свойств используемых признаков и т. п. Чтобы оценить эффективность системы распознавания, можно использовать методы математического моделирования функционирования системы, лабораторных и натурных экспериментов. Для математического моделирования системы распознавания необходимо построить специальную статистическую модель, на основе которой реализуется многократное повторение процесса распознавания объектов для каждого из рассматриваемых классов. Полученное в результате испытаний множество случайных исходов решения задачи распознавания дает возможность, опираясь на методы математической статистики, определить искомые оценки для вероятностей правильных и ошибочных решений, а также установить характер зависимости этих вероятностей от условий функционирования системы.

- 128-Натурные испытания дают наиболее

полную объективную оценку эффективности СР,

однако проведение таких испытаний представляет весьма дорогостоящую операцию. Поэтому в диссертационной работе лабораторные испытания заменены машинным экспериментом с использованием разработанных программных моделей, реализованных на языке C++ для ПЭВМ.

Экспериментальные исследования качества всех предложенных в диссертационной работе методов распознавания проведены как с настройкой на конкретного диктора, так и без настройки на диктора (дикторонезависимое распознавание).

Настройка на диктора подразумевает, что обучение проводится с использованием обучающих речевых фрагментов только одного конкретного диктора. Качество распознавания проверяется на тестовых фразах, произнесенных этим же диктором. При проверке качества дикторонезависимого распознавания обучение проводится по реализациям фонем, произнесенных различными дикторами - женщинами и мужчинами. В экспериментах было задействовано 15 дикторов, при этом количество фонем, подлежащих распознаванию, в тестовых фразах было не меньше 200. В процессе экспериментов исследована вероятность распознавания всех гласных фонем (а, о, у, э, и, ы) и огласованных согласных (л, р, з, в).

Вероятность правильного распознавания фонемы рассчитывается как отношение числа «правильно» выделенных из входного речевого потока распознаваемой фонемы к общему числу присутствующих в анализируемой фразе распознаваемых фонем.

4.1 Метод распознавания на основе ЛСК с использованием ДПР

4.1.1 Процедура обучения Разработка любой системы распознавания речи — это попытка создания прототипа человеческого «анализатора и распознавателя» речи. При этом каждый человек, прежде чем он начинает понимать речь, «проходит процедуру обучения».

Поэтому исключение блока обучения из структуры СРР никогда не приводило к сколько-нибудь хорошим результатам рас-

существующих на сегодняшний день на рынке СРР, процедура обучения имеет место в каждой такой системе. СРР подразделяют на дикторозависимые и дикторонезависимые. И те и другие делятся на системы с обучением, системы без обучения и на системы с частичным обучением (подстройкой). Дикторозависимость СРР подразумевает, что прежде чем работать с системой, пользователь обязан предоставить ей свои голосовые образцы всех слов из словаря. Дикторонезависимость, в свою очередь, дает возможность работать с системой без подстройки под голос диктора, т.е. любой диктор сразу может вступить в диалог с системой, не проходя процесс обучения. Системы с частичной подстройкой требуют обучения, но при этом диктор произносит не весь словарь, а какие-то отдельные его части (что существенно при большом объеме словаря).

В работе исследуется предложенная модель фонетико-акустического блока для СРР, которая нацелена на работу независимо от диктора с большими словарями. Такая система может быть отнесена к системам с частичным обучением (подстройкой). Основная процедура обучения проводится на этапе разработки системы.

Обучение исследуемых в диссертационной работе моделей проводилось с использованием отдельно записанных фонем, а также фонем, «вырезанных» из непрерывной речи. Процедура обучения реализована в виде отдельной подпрограммы, входящей в созданный в рамках диссертационной работы пакет программ (приложение 3).

Блок-схема обучения представлена на рис. 4.1.1.

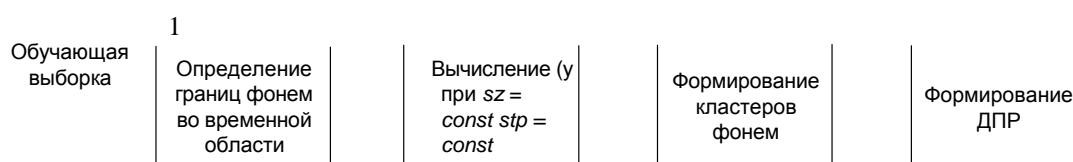


Рис. 4.1.1. Блок - схема обучения

В качестве входных данных программы обучения использовались отрезки речевых сигналов, записанные с использованием звуковой платы Sound Blaster Live 1024 фирмы Creative. Перед началом процедуры обучения требуется определить границы фонем во временной области (блок 1 блок-схемы обучения).

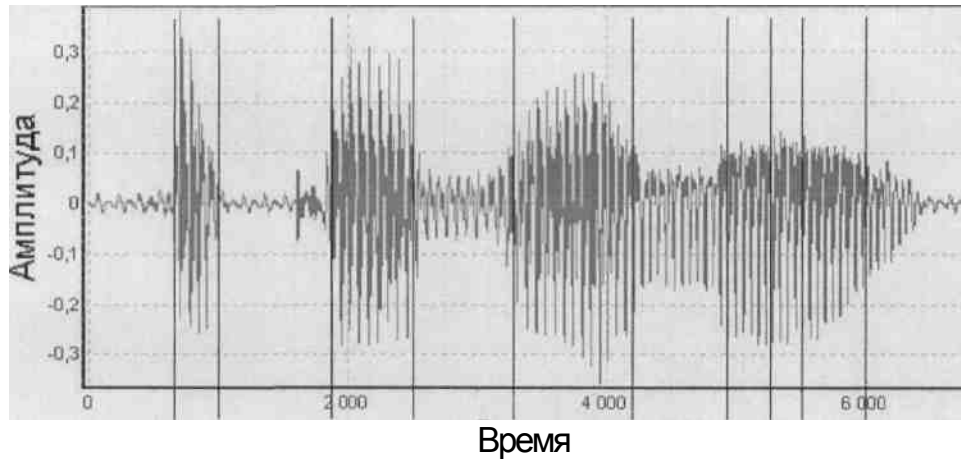


Рис. 4.1.2. Временная развертка РС используется для определения границ фонем

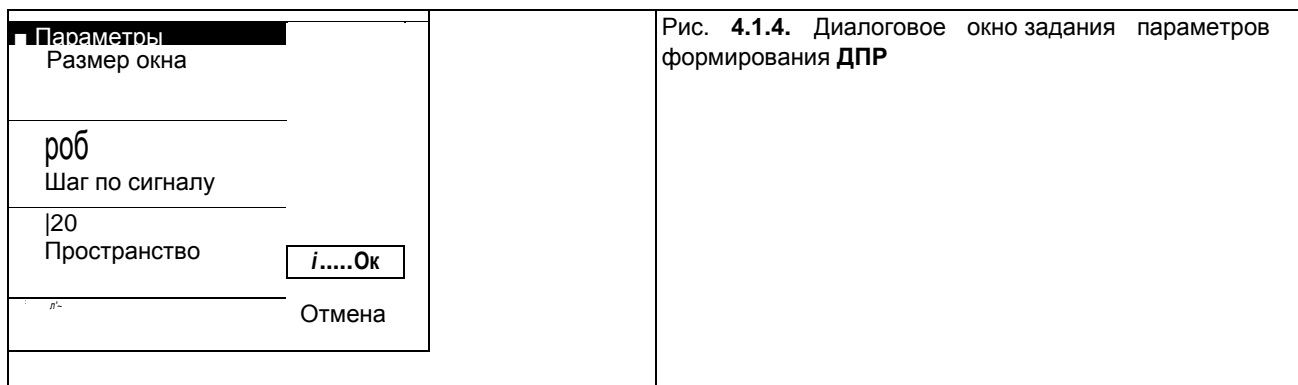
Это позволяет сделать окно временной развертки управляющей оболочки пакета программ (рис. 4.1.2). Границы фонем задаются по номерам отсчетов речевого сигнала, оцифрованного с частотой дискретизации 8 кГц. Границы фонем устанавливаются, что переходные участки (дифтонги) исключаются, а в пределах границ оставляются только стационарные участки фонем. «Координаты» границ фонем и параметры расчета ЛСК вводятся в отдельном окне программы (рис. 4.1.3) и используются в качестве входных данных блока 2 блок-схемы обучения.

660	960	л1	Скорость
1600	1825	отображения	
1900	2500	0	
2530	3140	Размер окна	
3300	4130	[100	
4300	4800	Шаг	
4850	5200	20	
5250	6050	Усреднение	
		120	
		Отмена	
		ОК	

Рис. 4.1.3. Диалоговое окно ввода временных границ фонем и параметров расчета ЛСК

Границы фонем используются для формирования кластеров фонем в подпространствах ЛСК (блок 3 блок-схемы обучения). Параметры расчета ЛСК (размер окна анализа и его шаг) выбираются на основании проведенных оценок (глава 2) и расчетов (глава 3). В блоке 3 реализован поиск центров тяжести и радиусов кластеров фонем. Формирование ДПР про-

- 131-граммно реализовано в виде отдельной поисковой процедуры (блок 4). В качестве исходных данных для формирования ДПР используются границы фонем, центры тяжести кластеров, их радиусы и значение размерности подпространств в узлах ДПР. Значение размерности задается пользователем как отдельный параметр перед началом процедуры формирования дерева (рис. 4.1.4).



Метод формирования ДПР (стартового ПП и ПП других уровней) подробно описан в главе 3 настоящей работы. По окончании обучения фиксируется ДПР, предназначенное для разделения заданного на этапе обучения набора фонем. Полученное ДПР используется в процедуре распознавания.

4.1.2 Исследование алгоритма распознавания Речевой сигнал, поступающий на вход модели распознавания, обрабатывается в соответствии с блок-схемой распознавания, представленной нарис. 4.1.5.

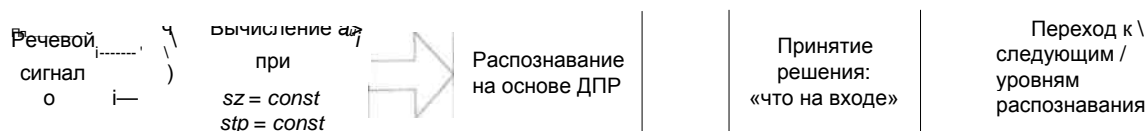


Рис. 4.1.5. Блок-схема распознавания

Блок 1 блок-схемы предполагает расчет ЛСК входного РС, при этом параметры расчета ЛСК (размер и шаг окна анализа) остаются неизменными для процедур обучения и распознавания. ЛСК вычисляются на отрезках речевого сигнала, равных размеру окна анализа с заданным шагом. Полученные значения ЛСК поступают на вход ДПР, сформированного на этапе обучения (блок 2).

- 132-Принятие решения осуществляется на основании проверок на каждом уровне ДПР в его узлах. Принятие решения об отнесении точки-представителя окна РС к тому или иному классу производится на основании результата сравнения евклидовых расстояний до центров тяжести классов в каждом узле ДПР.

В §3.1 (выбор размера и шага окна анализа) было указано на необходимость уточнения теоретически полученных параметров расчетной модели ЛСК путем экспериментальных исследований. В процессе выполнения работы было проведено большое число экспериментов с целью определения оптимальных параметров расчета ЛСК.

Покажем экспериментально влияние размера и шага окна анализа на качество распознавания. Для этого проведем эксперимент по распознаванию фонем и посчитаем ошибку распознавания для параметров расчета ЛСК, наиболее часто используемых в системах обработки речи: размер окна анализа и его шаг 120 отсчетов.

После окончания процедуры обучения получим ДПР, предназначенные для распознавания конкретно заданных на этапе обучения фонем. Для иллюстрации связи вероятности ошибки распознавания с размерностью подпространств в узлах ДПР, сформированы ДПР с размерностями подпространств в узлах 3 и 9 (рис. 4.1.5).

Фонемы: 12345 6 ЛСК: 1 23 4 5 678910 Оценка: 0.00000									
Фонемы : 1 ЛСК: 3 510 Оценка : 0,05924									
В Фонемы : 2 3 4 5 6 ЛСК: 3 510 Оценка: 0,05924									
В Фонемы :23 45 ЛСК: 2 4 6 Оценка : 0,11005									
Ё!..... Фонемы: 2 5 ЛСК: 2 4 7 Оценка : 0,15843									
Фонемы: 2 ЛСК: 2 5 7 Оценка: 0,28770 ■ Фонемы: 5									
ЛСК: 2 5 7 Оценка : 0,28770 В Фонемы: 3 4 ЛСК : 2									
4 7 Оценка : 0,15843									
Фонемы: 3 ЛСК: 2 4 8 Оценка : 0,26879									
.....Фонемы: 4 ЛСК: 2 4 8 Оценка : 0,26879									
Фонемы: Ё ЛСК: 2 4 6 Оценка : 0,11005									

Рис. 4.1.6. ДПР с размерностью подпространств в узлах равной 3

Фонемы: 1 2 3 4 5 6	ЛСК: 1 2 3 4 5 6 7 8 9 10	Оценка: 0,00000	
Фонемы : 1	ЛСК : 1 2 3 4 5 7 8 9 10	Оценка : 0,08933	
Фонемы: 2 3 4 5 6	ЛСК: 1 2 3 4 5 7 8 9 10	Оценка: 0,08933	
В Фонемы : 2 3 4 5	ЛСК : 1 2 3 4 5 6 8 9 10	Оценка: 0,11633	
В.....Фонемы : 2 5	ЛСК : 1 2 3 4 6 7 8 9 10	Оценка : 0,13991	
	Фонемы: 2	ЛСК: 1 2 3 4 5 7 8 9 10	Оценка: 0,28746
	.. Фонемы: 5	ЛСК : 1 2 3 4 5 7 8 9 10	Оценка: 0,28746
;-	Фонемы: 3 4	ЛСК : 1 2 3 4 6 7 8 9 10	Оценка : 0,13992
	Фонемы : 3	ЛСК : 1 2 3 5 6 7 8 9 10	Оценка : 0,25512
 Фонемы: 4	ЛСК : 1 2 3 5 6 7 8 9 10	Оценка : 0,25512
Фонемы: 6	ЛСК: 1 2 3 4 5 6 8 3 1 0	Оценка : 0,11633	

Рис. 4.1.7. ДПР с размерностью подпространств в узлах равной 9

Под словом «оценка» (рис. 4.1.6, 4.1.7) подразумевается априорная ошибка классификации. Если сравнить ошибки классификации в узлах обоих деревьев, то можно сделать вывод, что увеличение размерности подпространства в узлах ДПР с 3 до 9 приводит к незначительному снижению вероятности априорной ошибки классификации. Это подтверждает результаты оценок и расчетов, приведенных в предыдущих главах, которые говорят о том, что для разделения фонем по значениям ЛСК достаточно (для описания фонем ЛСК) использовать 2-3 ЛСК. Дальнейшее увеличение размерности векторов-признаков практически не уменьшает вероятность априорной ошибки разделения. Для параметров расчета ЛСК 120-120 (размер окна - шаг окна) значения ошибок в узлах ДПР (рис 4.1.6, 4.1.7) оказываются слишком большими для качественного распознавания фонем.

Наилучшие полученные экспериментально параметры расчета ЛСК: размер окна анализа 100 отсчетов, шаг окна 20 отсчетов. ДПР, полученные в результате обучения, показаны на рис. 4.1.8.

Фонемы: 1 2 3 4 5 6	ЛСК: 1 2 3 4 5 6 7 8 9 10	Оценка: 0,00000	
Фонемы : 1	ЛСК : 1 3 4	Оценка : 0,00156	
Фонемы : 2 3 4 5 6	ЛСК: 1 3 4	Оценка: 0,00156	
В.....Фонемы: 2 3 4 5	ЛСК: 2 4 8	Оценка : 0,02228	
В	Фонемы: 2 5	ЛСК : 1 2 4	Оценка: 0,02582
	Фонемы: 2	ЛСК: 1 2 6	Оценка : 0,09313
	Фонемы: 5	ЛСК: 1 2 6	Оценка : 0,09313
В.....Фонемы: 3 4	ЛСК : 1 2 4	Оценка: 0,02582	
	Фонемы : 3	ЛСК: 2 4 9	Оценка : 0,10475
Фонемы: 4	ЛСК : 2 4 9	Оценка : 0,10475
.....Фонемы : 6	ЛСК : 2 4 8	Оценка: 0,02228	

Рис. 4.1.8. Вариант ДПР с размерностями подпространств в узлах 3

Средний процент правильного распознавания для гласных и огласованных согласных с настройкой на конкретного диктора составил соответственно 71% и 60%. Вероятность пра-

- 134-вильного распознавания независимо от диктора для гласных составила 65%, для огласованных согласных 54%.

Таблица 4.1.1. Результаты распознавания

Средний % правильного распознавания С настройкой на диктора	Средний % правильного распознавания Независимо от диктора
Гласные- 71%	Гласные - 65%
Огласованные согласные - 60 %	Огласованные согласные - 54%

4.2 Метод распознавания на основе СС ЛСК

Аналитическое задание границ образов в едином для всех классов многомерном пространстве при большом числе образов практически невозможно. Поэтому в распознавании речи абсолютное признание получил эвристический принцип классификации по максимальному сходству с эталоном. В диссертационной работе такой подход применен для распознавания фонем по значениям скользящих средних ЛСК.

4.2.1 Процедура обучения В отличие от процедуры обучения, описанной в § 4.1.1, здесь формирование кластеров фонем происходит после расчета СС ЛСК.



Рис. 4.2.1. Блок - схема обучения

В соответствии с блок-схемой (рис. 4.2.1) сначала определяются границы фонем во временной области (как и в случае обучения по значениям ЛСК). ЛСК рассчитываются при заданных значениях размера и шага окна (блок 2), после чего рассчитываются скользящие средние ЛСК при заданном параметре усреднения L (рис. 4.1.3).

Оценки влияния параметра усреднения L приведены в 3 главе. Однако конкретное значение, которое нужно использовать в процедурах обучения и распознавания, можно оп-

ределить лишь экспериментальным путем. Результаты эксперимента будут приведены ниже.

4.2.2 Исследование алгоритма распознавания

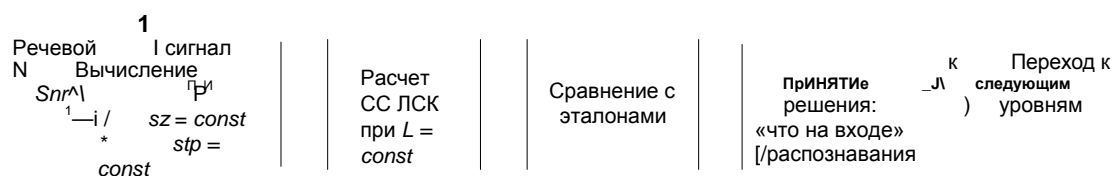


Рис. 4.2.2. Блок - схема распознавания

В отличие от алгоритма разделения, использующего ДПР, когда признаковые подпространства меняются в каждом узле ДПР, в данном случае используется одно общее признаковое подпространство, определяемое предложенными в работе методами (глава 3). Принятие решения об отнесении входной реализации к тому или иному классу осуществляется на основании сравнения p евклидовых расстояний до центров тяжести всех классов в едином (общем)/7-мерном признаковом подпространстве ($p = 1, 2, \dots, 10$).

Средний процент правильного распознавания для гласных и огласованных согласных с настройкой на конкретного диктора составил соответственно 75% и 68%. Вероятность правильного распознавания независимо от диктора для гласных составила 69%, для огласованных согласных 61%).

Таблица 4.2.1. Результаты распознавания

Средний % правильного распознавания С настройкой на диктора	Средний % правильного распознавания Независимо от диктора
Гласные - 75 %	Гласные - 69%)
Огласованные согласные - 68 %	Огласованные согласные - 61%>

представителей РС должно находиться в пределах одного кластера, чтобы было принято решение о том, что на входе распознавателя фонема а не переходной участок (дифтонг).

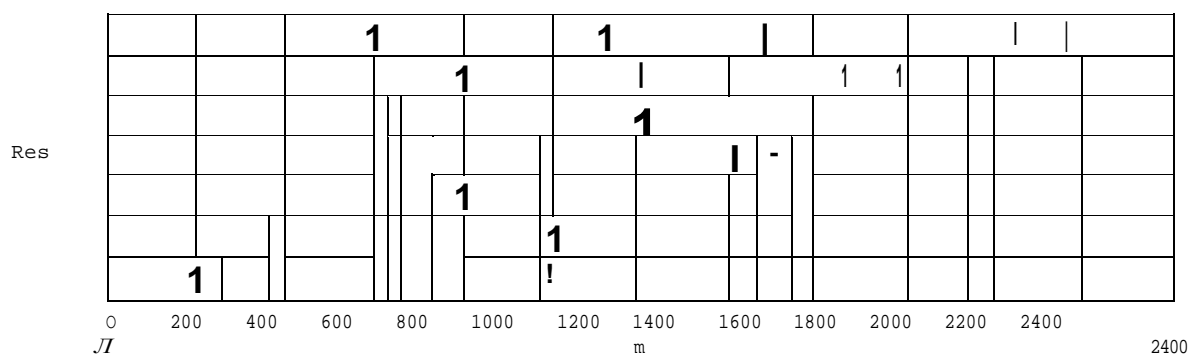


Рис. 4.2.4. Результат выделения фонем из речевого потока распознавания по 2-м признакам (СС ЛСК) (без нейтрализации дифтонгов)

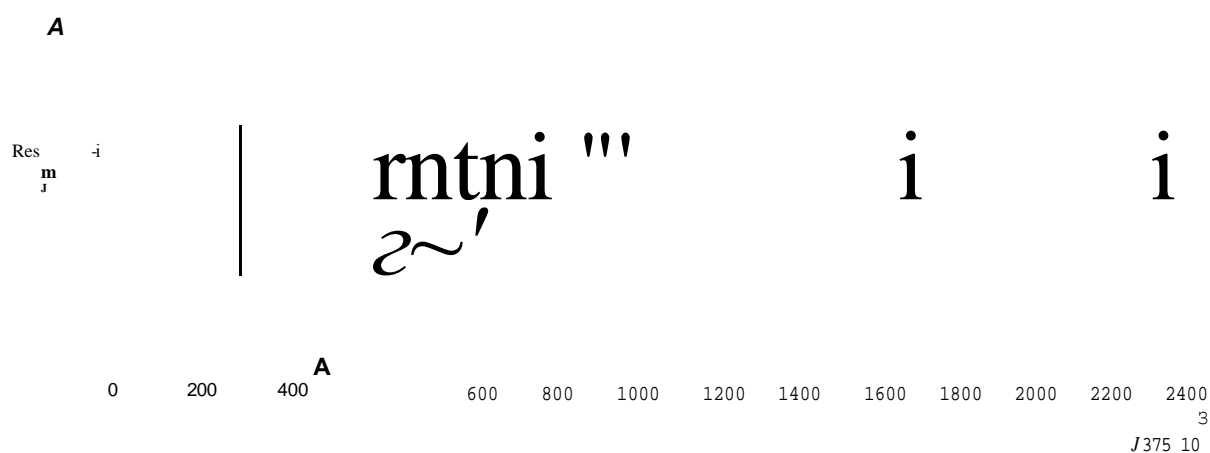


Рис. 4.2.5. Результат выделения фонем из речевого потока распознавания по 3-м признакам (без нейтрализации дифтонгов)

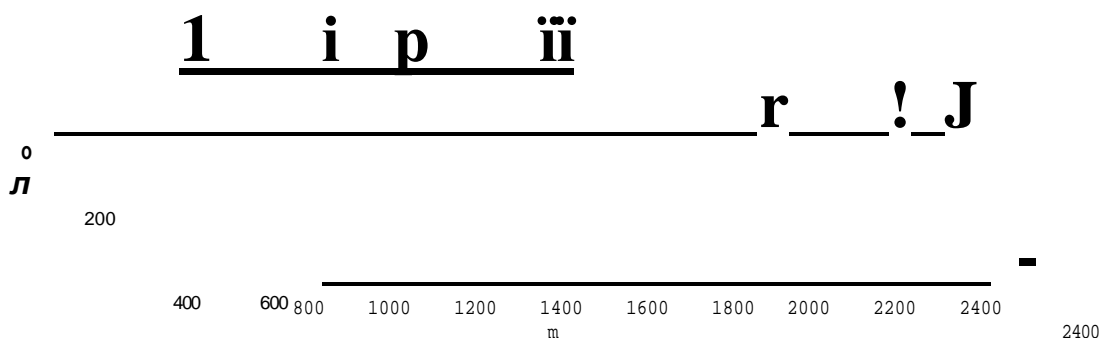


Рис. 4.2.6. Результат выделения фонем из речевого потока распознавания по 3-м признакам (с нейтрализацией дифтонгов)

На рис. 4.2.6 показан результат распознавания фонем с применением нового алгоритма, исключающего влияние дифтонгов.

Введение в алгоритм распознавания дополнительного параметра Q - времени нахождения точки вблизи одного и того же центра тяжести - позволило снизить ошибку до 19 %.

- 138-Наилучший результат был достигнут при использовании следующих параметров расчета

ЛСК: размер окна анализа 100 отсчетов, шаг окна 5 отсчетов. Параметр усреднения $L = 10$. Значение параметра $Q = 9$. Средний процент правильного распознавания для гласных и огласованных согласных с настройкой на конкретного диктора составил соответственно 84% и 73%. Вероятность правильного распознавания независимо от диктора для гласных составила 81%, для огласованных согласных 60%.

Таблица 4.1.3. Результаты распознавания фонем

Средний % правильного распознавания С настройкой на диктора	Средний % правильного распознавания Независимо от диктора
Гласные - 84%	Гласные - 81%о
Согласованные согласные - 73%	Огласованные согласные 60%)

4.3 Распознавание на основе комбинированного алгоритма с нейтрализацией влияния дифтонгов

Процедура обучения для исследуемого метода распознавания отличается от процедуры обучения для СС ЛСК лишь наличием блока формирования ДПР. Однако это введение дает существенный выигрыш в качестве распознавания.

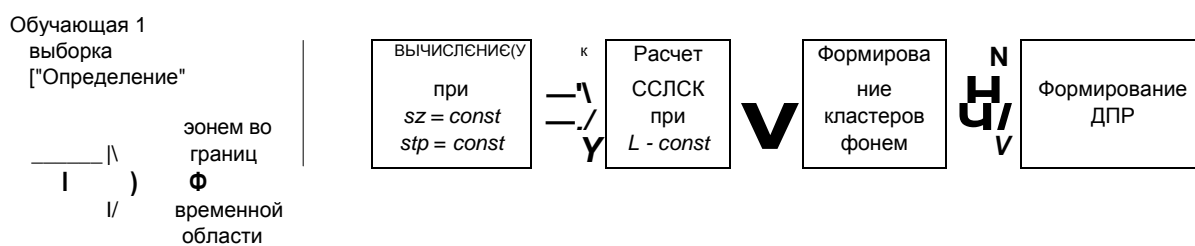


Рис. 4.3.1. Блок-схема обучения

Алгоритм распознавания использует все преимущества уже исследованных методов распознавания, включая в том числе возможность нейтрализации дифтонгов.

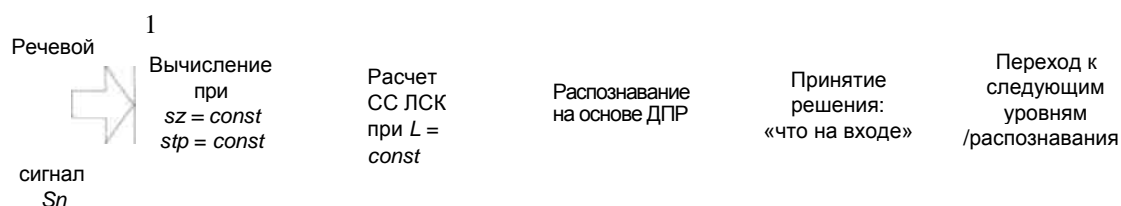


Рис. 4.1. Блок-схема распознавания

Наилучший результат был получен при использовании следующих параметров: раз-

мер окна 100 отсчетов, шаг окна 5 отсчетов, $L - 10 Q = 9$.

- 139-Средний процент правильного

распознавания для гласных и огласованных согласных

с настройкой на конкретного диктора составил соответственно 95% и 87%. Вероятность правильного распознавания независимо от диктора для гласных составила 85%, для огласованных согласных 81%.

Таблица 4.1.4. Результаты распознавания

Средний % правильного распознавания С настройкой на диктора	Средний % правильного распознавания Независимо от диктора
Гласные - 94%	Гласные - 85%
Огласованные согласные - 87%	Огласованные согласные - 81 %

4.4 Выводы

1. Произведена оценка качества распознавания фонем на тестовых речевых сигналах. Машинные эксперименты по распознаванию фонем показали хорошее соответствие результатов оценок, расчетов и эксперимента.
2. Применение признаков первого уровня (ЛСК) для описания фонем в совокупности с методом классификации на основе ДПР дало возможность выделять фонемы из речевого потока с вероятностью правильного принятия решения 65%. Относительно других предложенных в работе алгоритмов, такой алгоритм выделения фонем требует минимальных вычислительных затрат.
3. Причина невысокого качества распознавания фонем на основе СС ЛСК связано с наличием переходных участков (дифтонгов), которые в подпространствах СС ЛСК преобразовываются в межкластерные траектории. Вследствие случайного расположения кластеров в подпространствах СС ЛСК, траектории могут пересекать границы кластеров. При этом возникает дополнительная ошибка распознавания. Предложен метод минимизации ошибки за счет введения в алгоритм распознавания дополнительного параметра - времени нахождения точки-представителя распознаваемого объекта внутри границ кластера. В результате использования предложенного метода удалось повысить качество распознавания до 81%.
4. Наилучший результат (15% ошибок) достигнут с применением комбинированного алгоритма распознавания: рассчитывались СС ЛСК и далее принятие решения о принадлежности к одному из классов проводилось на основании отнесения точки-представителя окна РС к той или иной ветке оптимизированного ДПР.

ЗАКЛЮЧЕНИЕ

Речевой сигнал существенно отличается от всех искусственных, технических сигналов своей сложностью, неустойчивостью параметров, избыточностью. Если сравнить осциллограммы или спектрограммы одного и того же слова, произнесенного дважды одним и тем же диктором, а тем более разными дикторами, то бросаются в глаза значительные отличия уровней, длительностей участков, формы колебаний, спектральных картин. Поскольку органы речеобразования практически находятся в состоянии непрерывной перестройки, в речевом сигнале трудно указать границы отдельных фонем. Эти границы можно указать лишь с большой долей условности, при этом переходные участки обладают большим разнообразием и вносят основные трудности в процесс классификации звуков. При акустическом анализе речи формируют различные системы признаков (спектральные, автокорреляционные и др.). Области, отображающие фонемы (кластеры), в любой известной системе признаков частично перекрываются, что является первопричиной исходной неопределенности речи. Вряд ли возможна такая система признаков, в которой фонемы живой разговорной речи разделялись бы без ошибок. Даже человек не может безошибочно членить речевой поток на фонемы на основе только акустической информации, т. е. на основе знания фонетического состава речи. Основные результаты работы состоят в следующем:

1. Предложены новые акустические признаки фонем русской речи - ЛСК, которые до сих не применялись в СРР.
2. Разработана параметрическая модель расчета ЛСК для ЭВМ. На ее основе исследованы свойства ЛСК, произведена оценка влияния параметров модели (размер временного окна и его шага) на качество разделения фонем в пространствах ЛСК.
3. Предложен метод расчета вероятности ошибки классификации фонем, основанный на методе динамических сгущений.
4. Предложен статистический метод оценки информативности ЛСК, позволивший сократить в 3-4 раза размерность признакового пространства.

5. Разработаны методы формирования кластеров фонем, в основе которых лежат статистические свойства ЛСК.

6. Предложен метод минимизации ошибки распознавания, возникающей вследствие наличия дифтонгов.

7. Разработана программная модель фонемного уровня распознавания СРР для ПЭВМ.

СПИСОК ЛИТЕРАТУРЫ

1. Сапожков М.А. Речевой сигнал в кибернетике и связи. - М.: Связь, 1963. 452 с.
2. Fralik S.C. Learning of recognize pattern without a teacher. // IEEE Trans.- 1997.- IT-13.-№1.
3. Айвазян С.А., Бажаева З.И., Староверов О.В. Классификация многомерных наблюдений.- М.: Статистика, 1974.
4. Акинфиев Н.Н. Об одном психоакустическом законе восприятия человеком речевого сигнала и об объективных параметрах сигнала, содержащих речевую информацию. // Тез.докл. 15 Всесоюз. шк.-семинара "Автоматическое распознавание слуховых образов". - Таллинн: ИК АН ЭССР, 1989. - С. 179-181.
5. Арчер Т., Уайтчепел Э. «Visual C++ .net». Издательство «Диалектика», 2003.
6. Ахмед Н., Рао К.Р. Ортогональные преобразования при обработке цифровых сигналов.-М : Связь, 1980.
7. Вентцель Е.С. Теория вероятностей.- М.: Наука, 1964.
8. Виноградов СВ. Методы предварительной обработки речевого сигнала в системах распознавания речи. - Л., ЛИИАН, 1987 г.
9. Вокодерная телефония. Методы и проблемы. Под. ред. А.А. Пирогова.
10. Выгодский Я.Н. Справочник по элементарной математике. - М.: Наука, 1967.
11. Г. Корн, Т. Корн «Справочник по высшей математике». Издательство «Наука», Москва, 1984.
12. Г.Н. Воробьева, А.Н. Данилова «Практикум по вычислительной математике».
13. Г.Стренг «Линейная алгебра и ее применение». Издательство «Мир», 1980.
14. Галунов В.И. Бионическая модель системы распознавания речи. // Исследование моделей речеобразования и речевосприятия.- Л.: 1981, С. 36-51.
15. Галунов В.И., Жаков М.Л. и др. Первичный анализ в системах автоматического распознавания. // Тез. докл. 15 Всесоюз. шк.-семинара" Автоматическое распо-

образов". - Таллинн: ИК АН ЭССР, 1989.- С. 49-58.

16. Горелик А.Л., Скрипкин В.А. Методы распознавания. Учебное пособие для вузов
- М., Высшая школа, 1984 г.

0 17. Загоруйко Н.Г. Комбинированный метод принятия решений. Сб. тр. ИМСО
АН СССР «Вычислительные системы» вып. 19, Новосибирск, 1965.

18. Загоруйко Н.Г. Методы распознавания и их применение. - М., Советское радио,
1972 г.

19. Загоруйко Н.Г., Елкина В.Н., Емельянов СВ., Лбов Г.С. Пакет прикладных про-
грамм ОТЭКС. - М.: Финансы и статистика, 1986.

20. Зигангиров К.Ш., Сорокин В.Н. Об использовании последовательного декодиро-
вания для распознавания слитной речи. // ППИ. 1977 №4. с. 81-88.

21. Зяблов В.В. Речевая информатика. -М.: Наука, 1989.

22. Кельманов А.В. О некоторых проблемах построения систем распознавания инва-
риантных к диктору. // Тез.докл 15 Всесоюз.шк.-семинара " Автоматическое
рас-познавание слуховых образов".- Таллинн: ИК АН ЭССР, 1989.- С. 103-104.

23. Косарев Ю.А. Естественная форма диалога с ЭВМ. - Л.: Машиностроение.
Ле-нингр. отд- ние, 1989. - 143с.

24. Косарев Ю.А., Естественная форма диалога с ЭВМ - Л., Машиностроение, 1989г.

25. Кулагина О.С. Исследования по машинному переводу. -М.: Наука, 1979.

26. Ланнэ А.А. Оптимальный синтез линейных электрических цепей. - М: Связь,
1969.

27. Ланнэ А.А., Матюшкин Б.Д., Улахович Д.А. Основы цифровой обработки сигнала-
лов.-Л: ВАС, 1995.

28. Ланнэ А.А., Улахович Д.А. Передача информации о состоянии
фильтра-предсказателя с помощью спектральных пар // Радиоэлектроника и
связь. - 1991г. -№1.

29. Лбов Г.С. Методы обработки разнотипных экспериментальных данных. - Новосибирск, Наука, 1981 г.
30. Лбов Г.С. Методы обработки разнотипных экспериментальных данных. - Новосибирск: Наука (Сибирское отделение), 1981.
31. Любимов А.Ю., Евсиков М.М. Линейное предсказание речи - это просто // Монитор №4, 1995.
32. Люблинская В.В. Восприятие речи. Общие представления и подходы к исследованию. // Тез. докл. 15 Всесоюз. шк.-семинара "Автоматическое распознавание слуховых образов". - Таллинн, ИК АН ЭССР, 1989. - С.32-36.
33. Макхоул Д. Линейное предсказание. Обзор.
34. Маркел Д.Д., Грей А.Х. Линейное предсказание речи: Пер. с англ. - М.: Связь, 1980.
35. Методы анализа данных: Подход, основанный на методе динамических спусков: Пер. с фр. / Кол. авт. под рук. Э. Дидэ; Под ред. и с предисл. С.А. Айвазяна и В.М. Бухштабера. - М.: Финансы и статистика, 1985. - 357 с.
36. Михайлов В.Г., Златоустова Л.В. Измерение параметров речи. - М.: Радио и связь, 1987.- 168с.
37. Мышкис А.Д. Лекции по высшей математике. - М: Наука, 1969.
38. Покровский Н.Б. Расчет и измерение разборчивости речи. - М.: Связь, 1962. -391с.
39. Попов Э.В. Общение с ЭВМ на естественном языке. - М., Наука, 1982 г.
40. Рабинер Л.Р., Шафер Р.В. Цифровая обработка речевых сигналов.- М.: Радио и связь, 1981.
41. Сорокин В.Н. Истинные и ложные цели в распознавании и синтезе речи. // Речевая информатика.- Киев: Ин-т кибернетики АН УССР, 1989.- С. 40-45.
42. Сорокин В.Н. Теория речеобразования. - М.: Наука, 1985.

43. Турбович И.Т., Файн В.С. Распознавание образов. - М.: Наука, 1977.
44. Файн В.С. Распознавание образов и машинное понимание естественного языка. -М: Наука, 1987.
45. Чичагов А.В., Математическое моделирование сигналов. /М.:ВЦ РАН, 1995.
46. Чучупал В.Я., Маковкин К.А. Распознавание последовательностей слов для компьютерной телефонии. /Современные речевые технологии. Сборник трудов 9 сессии Российского акустического общества. -М.: ГЕОС, 1999, с. 81-84.
47. Чучупал В.Я., Маковкин К.А. Система распознавания слитно-произносимых названий цифр для телекоммуникационных приложений. /М.:ВЦ РАН, 1997.
48. Улахович Д.А., Сергеев М.В. Статистические свойства спектральных корней русской речи. // Труды учебных заведений связи / СПбГУТ.-СПб, 1988, №64.
49. Бьерн Страуструп «Язык программирования С++. Специальное издание». Издательство «Бином», 2001.

ПУБЛИКОВАННЫЕ СТАТЬИ

1. Кисляков СВ. Ортогональные полиномы в распознавании речевых сигналов //51-я НТК: тез. докл. / СПбГУТ. СПб, 1998.
2. Кисляков СВ., Шульгин Е.А. Применение преобразования Карунена-Лоэва для классификации фонем при дикторонезависимом распознавании речи // Труды учебных заведений связи / СПбГУТ. СПб, 2000. №166.
3. Кисляков СВ., Симонина О.А. Модель дикторонезависимого распознавания речи. // 2-я Межд. НТК «Техника и технология связи» / СПбГУТ. СПб, 2000.
4. Кисляков СВ. Шульгин Е.А. Анализ признаков речевых фонем, полученных на основе полинома Гурвица // 53-я НТК: тез. докл. / СПбГУТ. СПб, 2001.
5. Кисляков СВ. Некластерный алгоритм разделения согласных звуков речи независимо от диктора // 53-я НТК: тез. докл. / СПбГУТ. СПб, 2001.
6. Кисляков СВ., Шульгин Е.А. Метод расчета максимальной внутриклассовой ошибки распознавания фонем для систем реального времени. // 3-я Межд. научно-практическая конференция «Компьютерные технологии в науке, производстве, социальных и экономических процессах». / Новочеркасск, 2002.
7. Кисляков СВ. Выбор порядка ЛП-модели при построении системы распознавания речи // 54-я НТК: тез. докл. / СПбГУТ. СПб, 2002.
8. Кисляков СВ. Современное состояние проблемы создания систем речевого диалога человека и ЭВМ // 54-я НТК: тез. докл. / СПбГУТ. СПб, 2002.
9. Кисляков СВ., Шульгин Е.А. Оценка качества признаков речевых единиц в задаче распознавания // 55-я НТК: тез. докл. / СПбГУТ. СПб, 2003.
10. Фундаментальные аспекты новых информационных и ресурсосберегающих технологий: Отчет/рук. Е.А. Шульгин. НИР №190-93-054. СПб, 2002.

ПРИЛОЖЕНИЯ

Приложение 1. Вычисление ЛСК

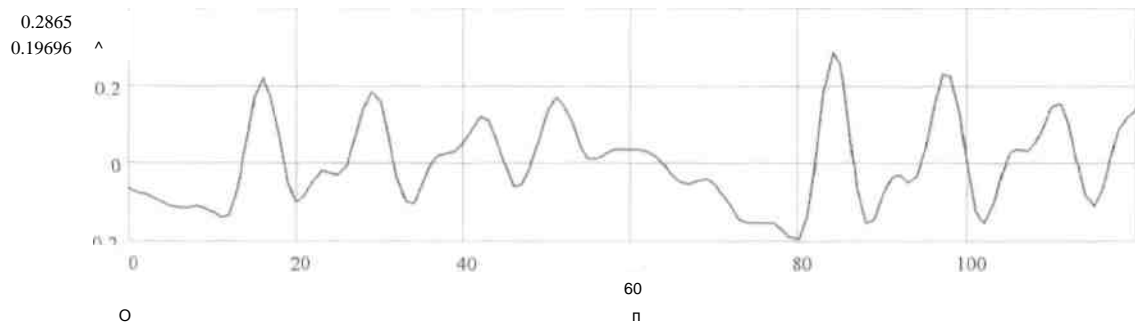
Чтение звукового файла

`s` `READPRNJ "c:\users\sergey\study\sounds\sound_txt\5.txt"` `)` `last(s) = 315`
`T` `φ)-1 :2 3 "R" 5 i 6 : 7 N`
`s` `0-0.066-0.074-0.08-0.088-0.1 -0.112-0.117-0.115-0.11 -0.113-0.126-0.142-0.136-0.069 0.056`

п о.. 120 количество отсчетов сигнала

120

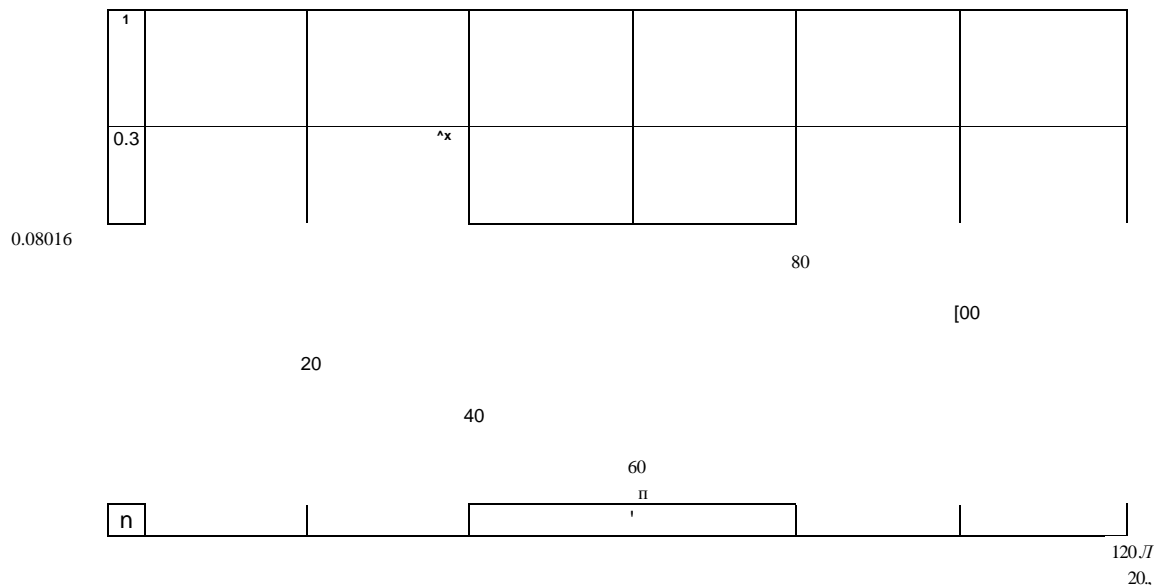
..120..



Взвешивание сигнала весовой функцией - окном Хэмминга
 окно N=120 отсчетов функция окна:

$$w = 0.54 - 0.46 \cos \frac{2\pi l}{N-1}$$

Вид весовой функции



Взвешивание сигнала окном Хемминга

`sw` `iifn` `0)(n N l),s w,0`
`n` `[` `'` `"` `a` `J`

Взвешенный сигнал

sw	0.2	
n		
0.19696	02	
	20	40
		6
		0
		1
ft)		
	100	120
		.120
frame	120	Длина кадра (количество отсчетов)^
order	10	Порядок модели (экстраполятора)

Расчет коэффициентов $\{a_k\}$ экстраполятора для одного кадра сигнала

```

for k ∈ 0.. order
     $V_k = \sum_{n=0}^{frame-1} \frac{SW_n}{SW_k} x_{n+k}$ 
    for n ∈ 0.. frame-1
         $V_k = \sum_{n=0}^{frame-1} \frac{SW_n}{SW_k} x_{n+k}$ 
    for k ∈ 0.. order
         $a_k = \frac{V_k}{V_0}$ 
    if  $V_0 \neq 0$ 
        re
            0
        a
            1
        V
            re
        alf
            alfa
             $\frac{2}{c} \cdot \frac{1}{c}$ 
        a
            0
    for me 1.. order-1
        for k ∈ 1.. m+1
             $b_k = a_{m+1-k}$ 
        S
            0
        for ip ∈ 0.. m
             $s = s + \frac{c}{ip} \cdot a_{m+1-ip} \cdot b_{ip}$ 
        rc
            $
            if alfa 0
                alfa
            rc
            0 otherwise
        for ip ∈ 1.. m
             $a_{ip} = a_{ip} + rc \cdot b_{ip}$ 
        a
            m+1
            , -- rc
        alfa- alfa
            rc
            alfa

```

Результат - параметры экстраполятор;
заданного порядка (order =)

```

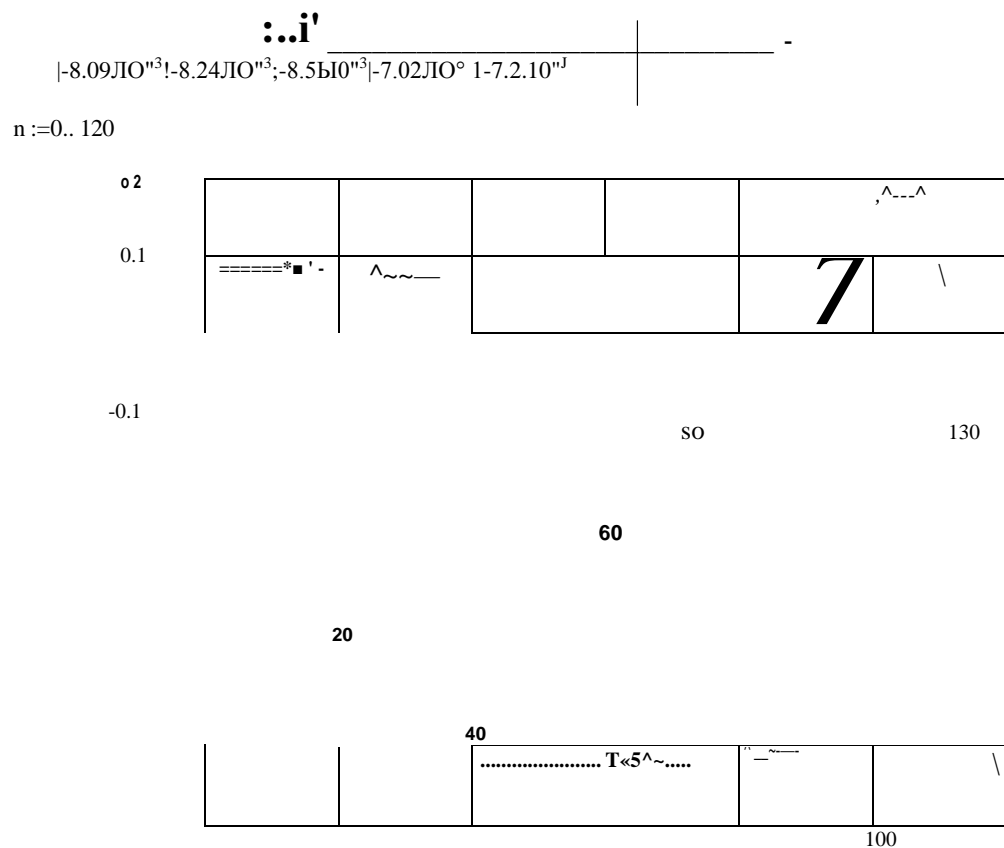
1
.56750256
10.72197083
10.25183397
£J1046787
0.16288009
-----
6
-0.05072512 $|
0.0925222
8 0.11569486
9 -0.00704348
110 -0.1432011

```


Приложение 2. Вычисление ЛСК на сигнале любой длительности

s := READPRN(" d:\sergey\study\sounds\tat for mcad\i big.tjrt-")

kst(s) = 1.335*10"



окно

frame := 120

окно Хемминга

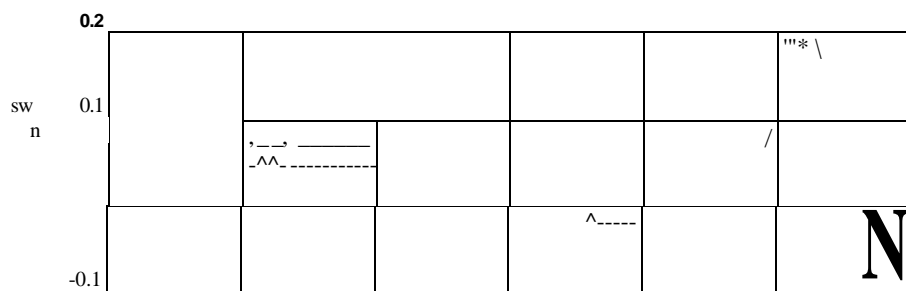
$w_n := 0.54 + QM \cdot \cosi^{-\wedge-\wedge-} \frac{\backslash \text{frame} - 1}{\text{frame} - 1}$

«

Взвешивание сигнала окном Хемминга

sw := iff(n>0)<n<frame- 1),si-w ,0]

Получившийся сигнал



20

40

6
0

SO 100

120

•

order:=10 Порядок модели (экстраполятора) frame :=120 Длина кадра (количество отсчетов)

```

q_0
for steps 0,frame., lengths) - frame
  for i € 0..frame
    snrt*p si i
    sw.— ifT(i>0) (i<frame), sn. w., 01
  for ke 0.. order
    Vo
    for n€0.. frame - k- 1

```

Результат- параметры экстраполятора
заданного порядка (order =)

```

for ke 1.. order a,
  -0

```

```

alfa— cQ
- c
re. ----- : if cQ?0

```

```

re— 0 otherwise

```

```

Vo
a1,9 —re

```

```

alfa— alfa- re c

```

```

for me 1.. order- 1

```

```

  for ke 1 ..m+ 1
    κ m+ 1 -k,q

```

```

s-0

```

```

for ipe 0.. m
  s— s + c „ . a.

```

```

re— — if alfa*0

```

```

  alfa
re—0 otherwise

```

```

for ipe 1.. m

```

```

  a. —a. +rcb.
  Φ.1 ΦPI Φ>

```

```

a „ —re

```

```

alfa— alfa - re alfa

```

```

q- q+i

```

I ¹	Ji ⁱ	[i ⁴	Л	[i
Λ ¹				
И-1.1297-■.0.9166^-0.9238 ¹ -0.8434 ¹ -0.9398 ¹ -0.7184 ¹ -0.7024				0.0669 1-0.0321
2 110314 10.2849 :0.4258 0.1829 0.0934				-0.6961 -0.6661
3 0.0576 1-1.0766 -1.352 0.7589 -0.4095				
	0.8245 0.4186 0.4603 0.3586 0.3412			
		-0.0476		
0.0269 ■.0.4715		-0.0797		
		-0.1044		
1-0.0181j 0.6333				
-0.1979			-0.0107 0.0277	
0.5609				
	0.925 0.254		0.2631 0.332	
-0.2218				
И ⁰ 0654 -0.3994				
	■.0.068 0.0602 -0.1394 -0.1835			
0 0.1591 0.1969 -0.0423 -0.1703 i-0.0552 -0.0143				
	-0.0493 0.2726 0.3485 0.1561 0.1			
J '0 0113				
■.0.0199				
0.0167				
	-0.1496 -0.101 -0.1789 -0.1573 -0.0518 -0.0494			
		■-0.0049		

¹ ^,4^.-1 °C₀°:=1 C»»'---I °C_{0,10}°:=1

-7 C_{ij}»5 C₁₇°:=-3 C₁₉°:=1

■32 C, ,:=18 °2,8°:=2
C..
2,4 2,0

56 C_{3ii}°:=-20 C₃₇°:=4

160 C₄₄°:=-48 C

112 C, ,:=16
5,5

■256 C_{0.4}°:=32

C.
5,5

128

	0	°■-°+°-° °-°+°5 ■-°-°	Γ 4	°Γ°
0	-1	0 i i i i i	:-1 0	j1 0 I-1 I0 j1 1
1	0	9 I0 1-7	[o 5	lo !- lo [l lo ;
2	50	0 -32 0	[IS 0	j 0 I2 io lo 1
3	0	-120 0 50	j0 -2	- I lo io lo I
4	-400	0 :160	0 ;:-48	0 !* lo lo lo io °
■	0	432 10	-11 o	16 io io 0 io I0 •
6	1.12.10 ³	0 -256	0 I 32	0 io iO lo lo lo \
7	0	-576 0 64	j0 0	lo lo lo io io
	-1.28.10 ³	0 I128 0	o 0	o io o o Q
	0	256 0 0	lo 0	lo lo 0 lo lo
	512	0 0 0	lo 0	lo io lo I0 lo \

i	I	i°I			
09099	°-09371	TO-7716	[-1.1183	-1.4336	
-10.4537	1-0.8782	1-0.0593		-5.6622	
148.1873	47.7955		49.5649	55.5932	
			60.5255	91.042	
1-2.3236					
	44.2096				
!138.3033	52.7762	42.9816			
-393.5908	-380.3269	■364.0516	■388.7954	-407.7849	
O - T.	■278.5736	■255.2177	-281.0261	-360.9084	
I 494.073	1067.3305	1037.3853	1086.5721	1110.814	
6					
1111.3937					
I 654.4214	459.0828	445.5992	437.2372	515.1368	
-1275.9867	1312	-1243.5325	512	■1225.5006	
-289.2153		-234.6608		-236.4999	

512 ■ 1256.5891 ■
-215.9152 1268.0423!
512 -240.5964
512

$$f := 0 \dots \text{cols}(G) - 1$$
$$\text{length}\{G^{(0)}\}=11$$
$$\text{коїш} \quad ' := \text{polyroots}(G \quad)$$
$$m := 0 \quad \text{.length}\{G^{<0>}\}$$
$$\text{omegi} \quad , := \text{acos/komi} \quad \backslash$$

	-0.986:-0.979	-0.973	■0.984	-0.988	-0.982	-0.981:-0.984
-0.88		0.786	-0.827	■0.841	-0.838	-0.842:-0.835
i-0.808						
<u>-0.679</u> i-0.659	-0.659	-0.667		-0.669		
-0.394-0.368	-0.361	-0.381	-0.662			-0.681:-0.662
ai=H-0.07	-0.159:-0.16			-0.153		
-0.153			-0.412	-0.4		
			-0.134	-0.401:-0.387		
					-0.1561-0.168	
10.256 :0.151 :0.134	0.18		0.222	0.155	0.153 :0.151	
10.555 10.509 :0.49	0.511		0.558	0.516	0.516 :0.517	
10.807 :0.812 :0.816	0.792		0.789		0.799 i0.796	
			0.788			
»0.962 :0.965 :0.967	0.957		0.947	0.949	0.952 :0.957	
10.993 :0.994 :0.994	0.993		0.992		0.992 :0.991	
			0.992			

	j j	2976	2.93712.91 :2.96112.989 2.954	2 947 2 965 2 912:2.967:2.96	2 0S2 2 975	2.978	2971 ■■
omegi			2.512 2.474 2.544 2.571 2.565	2 572 2 559 2 479 2 542 2 539	2.617 j 2.617	2.567	2.5931
	■	2317	2.29 :2.29 12.3 j 2.294 j2.303	2.32 2.294 2.263 2.274 2.282	2.316 j 2.333	2.273	2.2991
	■	T ₃ Te	1.948:1.94 11.96211.99511.982	1984 1.96811.947 [1.96 :1.966	2.014 11.973	1.949	1.941 j

```
prom := csort(omegi, 0)
```

$$W := \text{prom}$$

W-	0 [11] z	.		5	6	7	8	9
	D 0 11460.2767 JO.6325	U.9827	1.3114	1.6407 j	1.9758	2.3166	2.6458	2.9759
	1 0 1073! 0.2646!0.6233	1.0372	1.4188	1.7305 j	1.9475 j	2.2897	2.5123	2.9369
	2 = 0.1081 0.2595 0.6172	1.0583	1.436	1.731911.9404	2.2898	2.4744		2.9099
	3 и 1224JO.2944ПН.6566	1.0343	1.3898	1 7239	1 9o21	2 3004	2 5441	2.9611
	4 0.1281 0.3285 0t>612	0.9785	1.3469	1.7057	1.9955	2.2939	2.5705	2.9894
	5 0 125 0 321 0 6627	1.0281	1.4149	1.7242	1.9818	2.3035	2.5651	2.9535

```
WRITEPRN(" d:\us ers\s ergeyVstudyVs ounds\txt_for_mc adWmS .pm" ) := W
```

```
rows(W) = 1.112-10-
```

Приложение 3. Программный комплекс для анализа речевых сигналов

Программа позволяет:

- Открывать и добавлять файлы речевых сигналов.

"По заданным пользователем параметрам рассчитывать ЛСК, их скользящие средние и собственные векторы ковариационных матриц фонем.

- Автоматически подбирать параметры модели, при которых отделение фонем в плоскости друг от друга будет наилучшим.
- Рассчитывать среднюю ошибку разделения фонем при фиксированном шаге с переменным размером окна, и наоборот.
- Формировать дерево принятия решений.
- Распознавать тестовые речевые сигналы по скользящему среднему или по дереву принятия решений.
- Сохранять в файл линейные спектральные корни и графики.
- Оценить затраты времени на расчет одного и всех линейных спектральных корней, их скользящего среднего и собственных векторов ковариационных матриц.

Для анализа полученных результатов программа отображает следующие графики:

- Речевого сигнала во временной области.
- Линейных спектральных корней во временной области.
- Точек-представителей фонем в плоскости выбранных ЛСК.
- Точек-представителей фонем в плоскости выбранных СС ЛСК.
- Точек-представителей фонем в плоскости выбранных СВ ковариационных матриц.
- Ошибок разделения фонем.
- Результата распознавания тестового речевого сигнала.

Вид главного окна программы

Внешний вид программы продемонстрирован на рисунке:

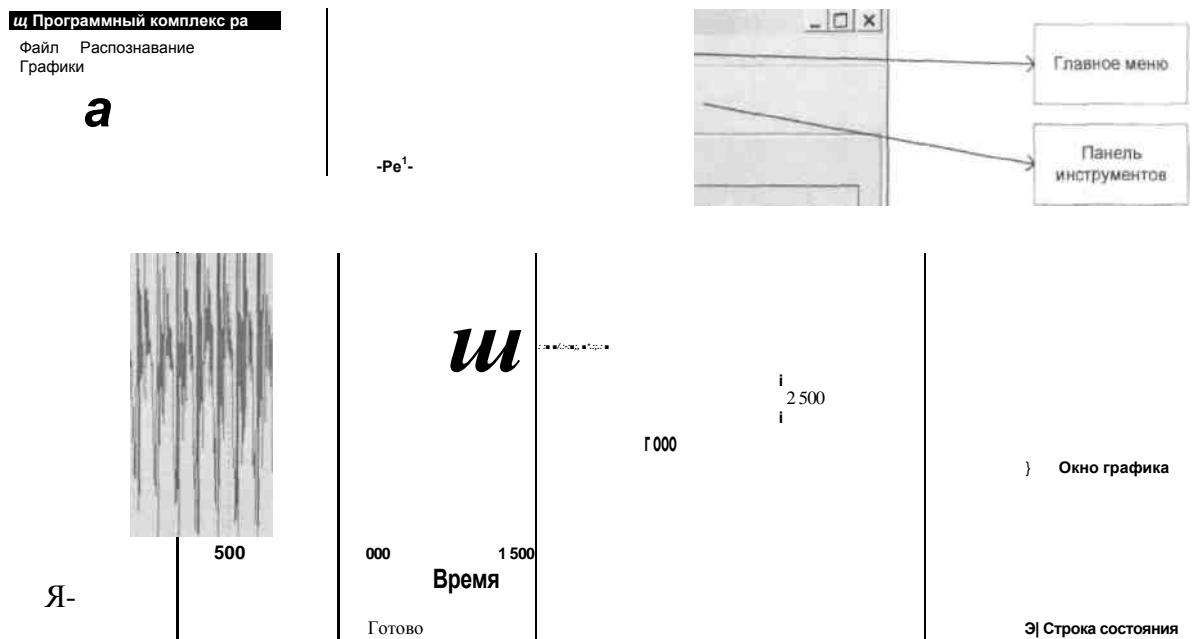


Рис.1. Главное окно программы

Команды главного меню

4.4.1.1 Меню «Файл»

Открыть - открыть речевой сигнал из файла, в нем должны храниться значения амплитуд речевого сигнала.

Добавить - добавить к открытому речевому сигналу еще один.

Сохранить - сохранить в файл линейные спектральные корни или текущий график.

Выход - выход из программы.

4.4.1.2 Меню «Распознавание»

Опции - вызывает окно, где пользователь может задать параметры модели.

Вычислить ЛСК

Вычислить СС ЛСК

Вычислить СВ ЛСК

Вычислить все

Обучить программу по СС ЛСК - фиксирование границ заданных фонем.

скользящее среднее тестового речевого сигнала по зафиксированным границам.

Распознать ЛСК по ДПР - распознать линейные спектральные корни тестового речевого сигнала по дереву принятия решений.

Распознать СС ЛСК по ДПР - распознать скользящее среднее тестового речевого сигнала по дереву принятия решений.

4.4.1.3 Меню «Графики»

Речевой сигнал - отобразить речевой сигнал.

ЛСК - отобразить все ЛСК по времени

ЛСК в плоскости - по заданной паре ЛСК, отобразить их в плоскости.

СС ЛСК - по заданной паре СС ЛСК, отобразить их в плоскости.

СВ ЛСК - по заданному собственному вектору и паре его значений отобразить их в плоскости.

4.4.1.4 Меню «Поиск»

Поиск лучших параметров - по заданным пользователем параметрам осуществляется поиск по СС ЛСК таких параметров модели, при которых разделение фоном будет наилучшим. Результатом поиска будет отчет в виде таблицы.

Расчет средней ошибки - по заданным пользователем параметрам осуществляется расчет по СС ЛСК с целью оценки размера окна и шага

Поиск дерева принятия решений - по заданным пользователем параметрам осуществляется поиск дерева принятия решений.

4.4.1.5 Меню «Результаты»

Оптимальные параметры — отображает таблицу лучших параметров.

График средней ошибки от размера окна

График средней ошибки от шага

Дерево принятия решений - отображает дерево принятия решений.

4.4.1.6 Меню «Помощь»

- Время расчета - отображает время расчета ЛСК, СС и СВ
- О программе

Команды панели инструментов

Команды панели инструментов дублируют действия пунктов главного меню следующим образом:

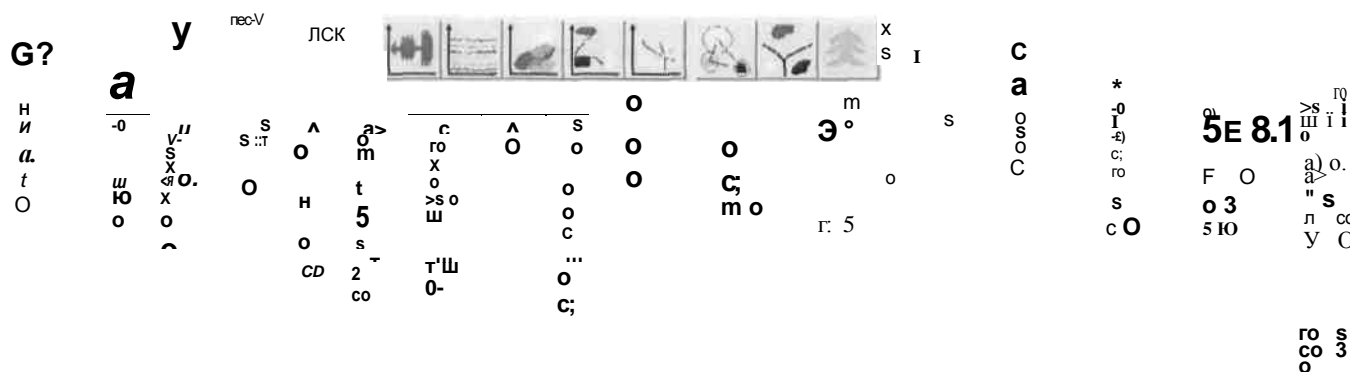


Рис.2. Команды панели инструментов

Принципы работы с программой

В следующих разделах, предлагаются описания типичных действий и приемов работы с программой.

Открытие файла речевого фрагмента

Чтобы открыть или добавить файл речевого сигнала необходимо:

1. Откройте меню **Файл** и выберите команду **Открыть**.
2. В списке **Папка** выберите диск, папку или ресурс Интернета, содержащие файл, который требуется открыть.
3. В списке папок найдите и откройте папку, содержащую нужный файл.
4. Выделите файл, а затем нажмите кнопку **Открыть**.

Ввод границ речевых единиц

Чтобы определить опции необходимо:

1. Откройте меню **Файл** и выберите команду **Опции**.

2. В появившемся диалоговом окне введите количество фонем в речевом сигнале и нажмите кнопку **Ок**.
3. Появится следующее диалоговое окно:

***1**

	Скорость отображения		
Начале	Конец	Цвет	Буква
300	6500	255	A
8500	14000	65535	И
15000	21000	42273270	
22500	30000	65280	У
33000	38000	1671168Ы	
40000	49000	1677696Э	

Размер окна
[Тоо

Шаг _____
(20

Усреднение ||C)

Рис.3. Диалоговое окно

«Опции»

4. Задайте параметры модели **Размер окна**, **Шаг**, **Усреднение**. **Размер окна** не может быть меньше 10, **шаг** не может быть больше размера окна, **усреднение** не может быть меньше 2.
5. Задайте **Скорость отображения** скользящего среднего. Этот параметр характеризует динамику отображения.
6. Заполните элементы таблицы. Количество строк в таблице зависит от заданного количества фонем (пункт 2). Каждая строка должна характеризовать **Начало**, **Конец**, **Цвет** и **Название** (столбец **Буква**) фонемы. **Начало** и **конец** должны задаваться по речевому сигналу. Параметр **Цвет** определяет цвет отображения фонемы на графиках.
7. После ввода всех параметров нажмите кнопку **Ок**.

Если количество заданных фонем больше одной, то перед вами откроется возможность автоматически осуществлять **Поиск** (см. ниже).

Вычисление ЛСК, скользящих средних ЛСК и собственных векторов

ковариационных матриц

Откройте меню **Распознавание** и выберите команду **Вычислить ЛСК**, **Вычислить СС ЛСК** или **Вычислить СВ ЛСК**.

3. В списке **ЛСК по x** и **ЛСК по y** выберите номера ЛСК, которые будут формировать плоскость. Нажмите кнопку **Ок**.

Точки-представители фонем в плоскости выбранных СС ЛСК.

1. Откройте меню **Графики** и выберите команду **СС ЛСК**.
2. Появится диалоговое окно. См. рис.6.
3. В списке **ЛСК по x** и **ЛСК по y** выберите номера ЛСК, которые будут формировать плоскость. Нажмите кнопку **Ок**.

Пример отображения СС ЛСК сигнала «АИОУЫЭ» в плоскости изображен на рисунке:

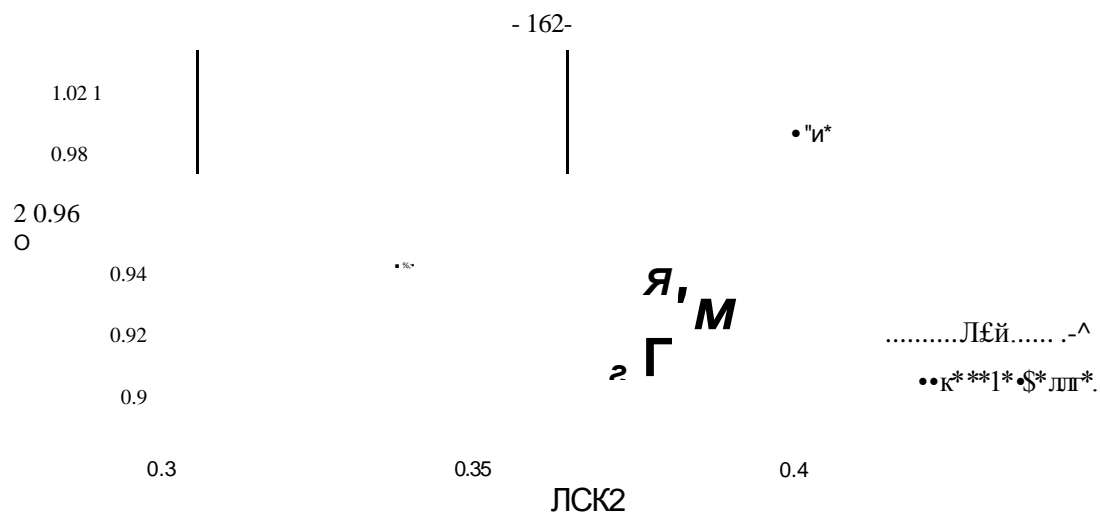


Рис.8. Точки-представители фонем в плоскости выбранных СС ЛСК.

4. Точки-представители фонем в плоскости выбранных СВ ковариационных матриц.
5. Откройте меню **Графики** и выберите команду **СВ ЛСК**.
6. Появится диалоговое окно:

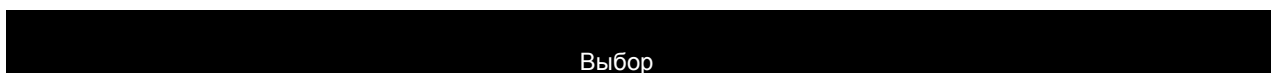


Рис.9. Диалоговое окно «Выбор»

7. Выберите из списка номер собственного вектора и два его значения, которые будут формировать плоскость. Нажмите кнопку **Ок**.

Пример отображения СВ ковариационных матриц ЛСК сигнала «АИОУЫЭ» в плоскости изображен на рисунке:

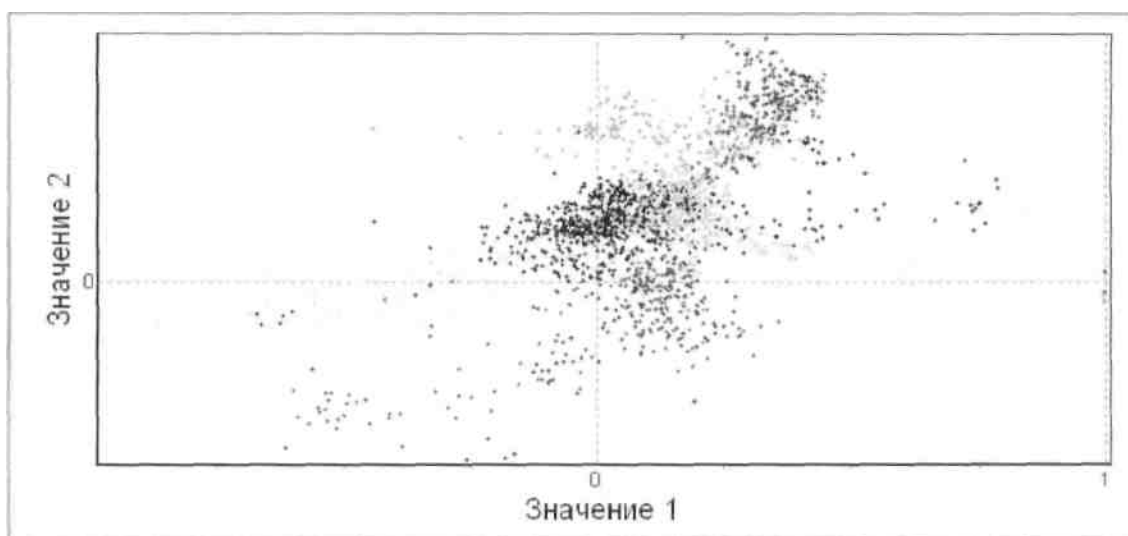


Рис.10. Точки-представители фонем в плоскости выбранных СВ ковариационных матриц.

Все графики можно масштабировать. Для этого установите курсор мыши на область графика, нажмите и держите левую кнопку мыши, затем переместите курсор вниз и вправо. Получится прямоугольник, отпустите левую кнопку мыши и область графика, заключенная в прямоугольник будет отображена более подробно. Для возврата нормального режима, необходимо повторить все вышеперечисленные действия, за исключением направления движения мыши - курсор следует передвигать в противоположном направлении.

Реализация методов поиска оптимальных параметров модели расчета ЛСК

Программный комплекс обеспечивает четыре поисковых операции: ■ «Поиск лучших параметров». Производится поиск параметров модели, при которых разделение фонем будет наилучшим. Поиск осуществляется методом динамических сгущений. Результатом поиска является таблица со списком параметров и плоскостей, упорядоченная от «лучшего» к «худшему» результату по критерию средней ошибки. Для выполнения поиска:

1. Откройте меню **Поиск** и выберите команду **Поиск Лучших параметров**
2. Появится диалоговое окно:

Параметры поиска	
Минимальный размер окна	Шаг по окнам
10	10
Максимальный размер окна	Шаг по шагу
200	
Усреднение	
120	
Отмена	

Рис.11. Диалоговое окно «Параметры поиска»

3. Задайте критерии поиска: диапазон варьирования **Размера окна** и **Усреднение**.
4. Нажмите **Ок** и подождите завершения поиска.

Пример результата поиска:

Результаты поиска					
Кол-во окон	Размер окна	Размер шага	ЛСКпох	ЛСКпо ч	Средняя ошибка Δy
10005	10	5	1	7	0.10361352524118
5002	10	10	3	10	0.10382478960597
10005	10	5	3	10	0.10426910782860
10005	10	5	3	10	0.10457425795676
10005	10	5	5	10	0.11704430679478
5002	10	10	7	10	0.11966079096361
10005	10	5	2	8	0.12107995351687
10005	10	5	1	2	0.12538949301021
10005	10	5	1	2	0.12617277668503jg

Ок

Рис.12. Диалоговое окно «Результаты поиска»

- «Расчет средней ошибки» позволяет рассчитать зависимость средней ошибки деления

фона от размера окна при фиксированном шаге и, наоборот. Для выполнения поиска: 1.

Откройте меню **Поиск** и выберите команду **Расчет средней ошибки**. Появится диалоговое окно:

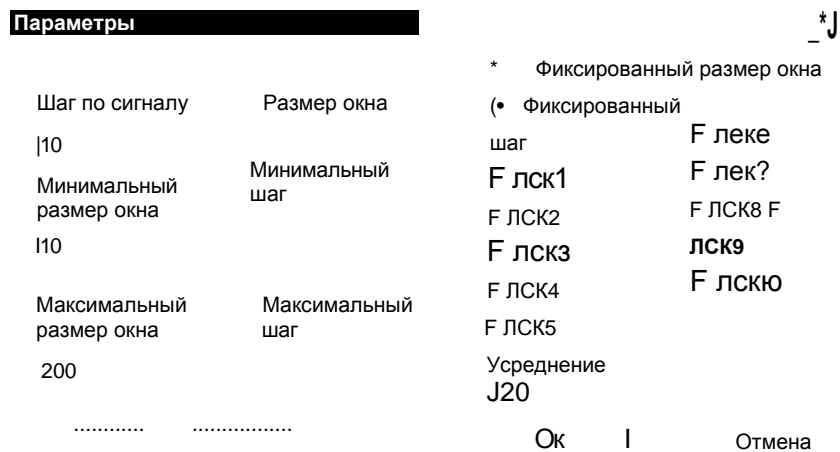


Рис.13. Диалоговое окно «Параметры»

2. Диалоговое окно рассчитано на два случая, на поиск с **фиксированным размером окна** и на поиск с **фиксированным шагом**. Задайте ЛСК, которые будут участвовать в поиске, **Усреднение**, а также: для первого случая - диапазон варьирования **Шага** и **Размер окна**; для второго - наоборот.

3. Нажмите Ок и подождите завершения поиска.

- «Поиск дерева принятия решений» позволяет построить бинарное дерево распознавания

фонем. 1. Откройте меню **Поиск** и выберите команду **Поиск дерева принятия решений**. Появится диалоговое окно:

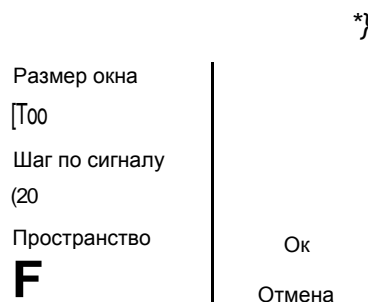


Рис.14. Диалоговое окно параметров поиска ДПР

2. Задайте **Размер окна**, **Шаг** и размерность **Пространства** дерева.
3. Нажмите кнопку Ок.

Пример ДПР приведен на рисунке:

Дерево принятия решений

J<J

```

Фонемы: 1 2 3 4 5 6   ЛСК: 1 234 567 8 910   Оценка: 0.00000
i .... Фонемы :1     ЛСК: 3 4   Оценка: 0.01 23Э
В ... Фонемы: 2 3 4 5 6   ЛСК: 3 4   Оценка : 0.01289
  ф .. Фонемы:2345   ЛСК:26   Оценка: 0.02682
    В   Фонемы :25   ЛСК: 2 4   Оценка: 0.05070
      Фонемы : 2   ЛСК: 4 8   Оценка :
      0.18471 Фонемы: 5   ЛСК: 4 8   Оценка
      : 0.1 8471
        В.....Фонемы :34   ЛСК: 2 4   Оценка: 0.05070
          Фонемы :3   ЛСК: 2 4   Оценка : 0.06833 Фонемы: 4
          ЛСК: 2 4   Оценка : 0.06833 Фонемы: 6   ЛСК: 2 6
          Оценка : 0.02682
  
```

Ok

Рис.15. Диалоговое окно «Дерево принятия решений»

Реализация методов распознавания

Программа позволяет распознавать речевые сигналы двумя методами: по значениям СС ЛСК и по значениям ЛСК на основе ДПР. ■ Распознавание по значениям СС ЛСК.

о Обучение

1. **Откройте** речевой сигнал.
2. Задайте количество фонем не равное 0.
3. Задайте параметры модели - **размер окна, шаг, усреднение и границы фонем** (обязательное условие обучения).
4. Вычислите скользящее среднее.
5. Откройте меню Распознавание и выберите команду Обучить программу по СС ЛСК.
6. В появившемся диалоговом окне

Выбор про.		л!
FIQCKIj	Ф леке	
Ф ЛСК2	Ф ЛСК7	
Ф лск3	Ф ЛСК8	
Ф ЛСК4	Ф ЛСК9	
Ф ЛСК5	Ф лею о	
Ok	Отмена	

Рис.16. Диалоговое окно «Выбор пространства»

укажите номера ЛСК, по которым будет производится распознавание.

спознавание

Откройте меню Распознавание и выберите команду Распознавание по СС ЛСК

В списке Папка выберите диск, папку или ресурс Интернета, содержащие файл, который требуется открыть.

В списке папок найдите и откройте папку, содержащую нужный файл.

Выделите файл, а затем нажмите кнопку Открыть.

Распознавание значений ЛСК на основе ДПР

Обучение

Откройте речевой сигнал.

Задайте количество фонов не равное 0 или 1.

Введите границы фонов.

Найдите дерево принятия решений.

Распознавание

Откройте меню Распознавание и выберите команду Распознавание ЛСК по ДПР

В списке Папка выберите диск, папку или ресурс Интернета, содержащие файл, который требуется открыть.

В списке папок найдите и откройте папку, содержащую нужный файл.

Выделите файл, а затем нажмите кнопку Открыть.

Распознавание значений СС ЛСК на основе ДПР.

Обучение

Откройте речевой сигнал.

Задайте количество фонов не равное 0 или 1.

Введите границы фонов.

Найдите дерево принятия решений.

Распознавание

Откройте меню Распознавание и выберите команду Распознавание СС ЛСК по ДПР

2. В списке Папка выберите диск, папку или ресурс Интернета, содержащие файл, который требуется открыть.
3. В списке папок найдите и откройте папку, содержащую нужный файл.
4. Выделите файл, а затем нажмите кнопку Открыть.

При обучении программы фонемы записываются в порядке следования, т.е. каждой фонеме присваивается порядковый номер.

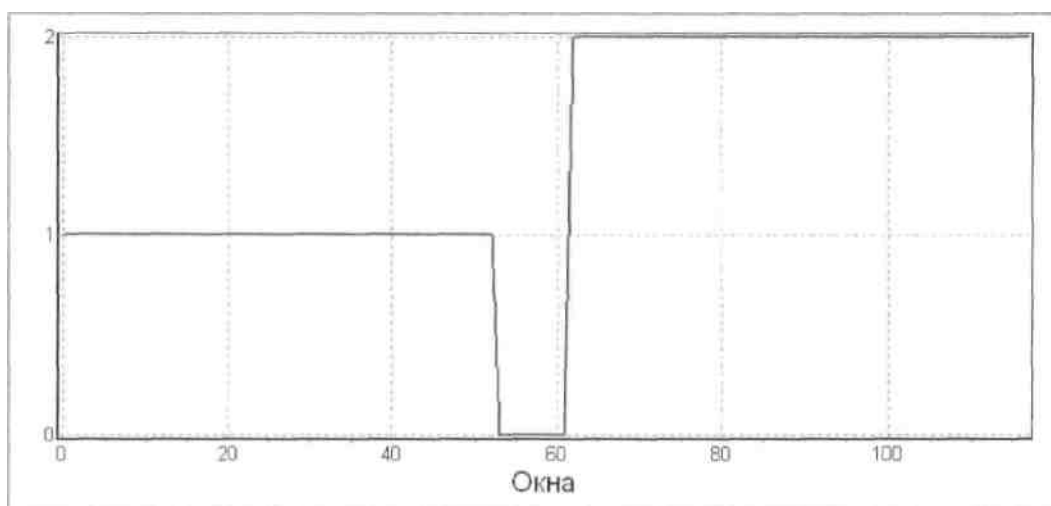


Рис.9. Пример распознавания фонем «^Hj^ojDC»

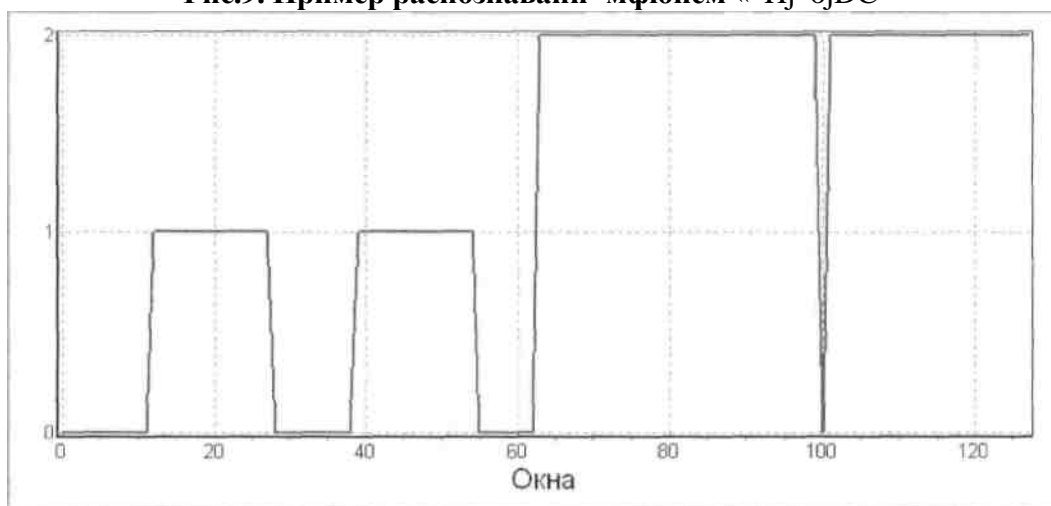


Рис.10. Пример распознавания фонем «АИ» по ЛСК ДПР

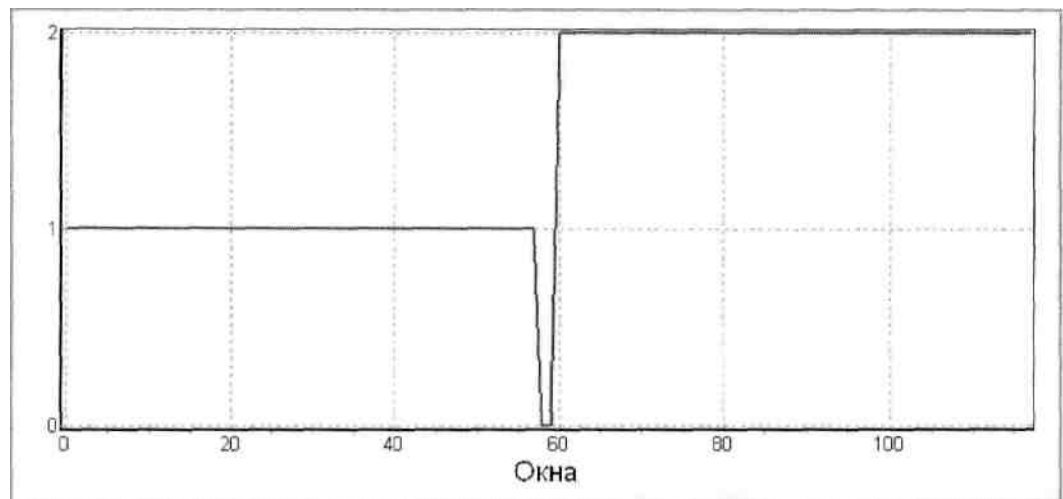


Рис. 11. Пример распознавания фонем «АИ» по СС ДПР

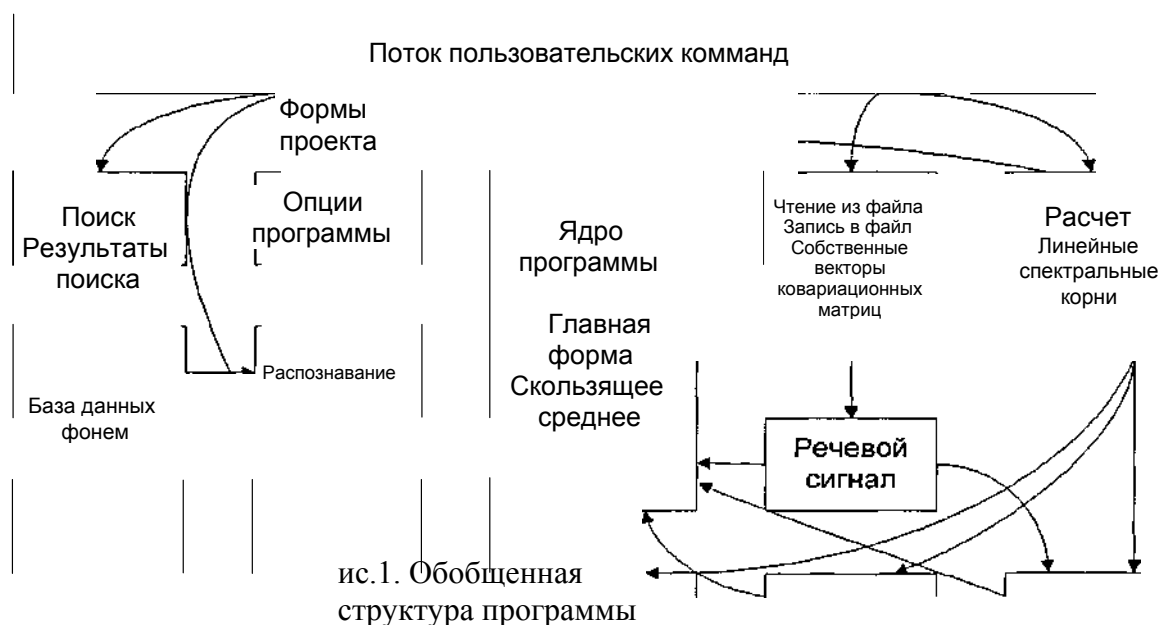
На графиках: значение 0 - фонема не
 распознана значение 1 - распознана
 фонема «А» значение 2 - распознана
 фонема «И»

Приложение 4. Описание реализации программы

Приложение. Описание реализации программы

Структура программы

Рассмотрим рисунок, на котором представлена обобщенная структура программы:



В ней можно выделить следующие составные части:

- *Речевой сигнал* - модель речевого сигнала, представленная в виде вектора, который содержит значения амплитуд речевого сигнала.
- *Линейные спектральные корни, скользящее среднее и собственные векторы* — основные элементы программы. Они являются основой ко всем последующим вычислениям, позволяют производить анализ полученных результатов.
- *Ядро программы* - основное окно программы, ее интерфейс. Все функции программы вызываются с главной формы.
- *Поиск* - интерфейс к автоматическому поиску лучших параметров, средней ошибки перекрытия областей фонем для заданного пространства и дерева принятия решений.
- *Распознавание* — инструмент для распознавания тестовых речевых сигналов.

Работа программы

Работа программы представлена на рисунке:

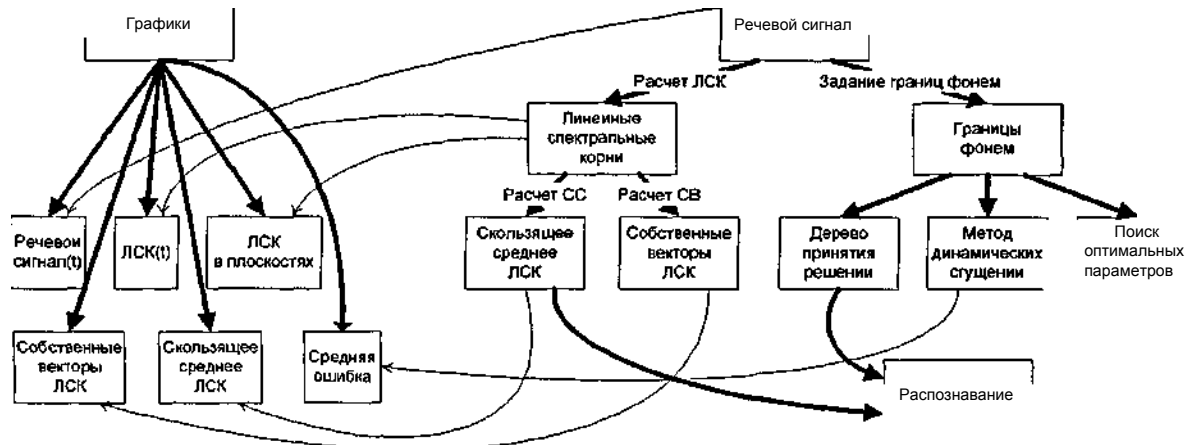


Рис.2. Работа программы

Работа программы развивается в двух направлениях, первый из них - всевозможные расчеты и поиски, второй - отображение полученных результатов. Таким образом, первая ветвь - это реализация программы с точки зрения ее реализации, вторая - интерфейс программы с точки зрения пользователя. Рассмотрим более подробно ее реализацию.

Взаимодействие классов

Рассмотрим рисунок, на котором приведено обобщенное взаимодействие классов:

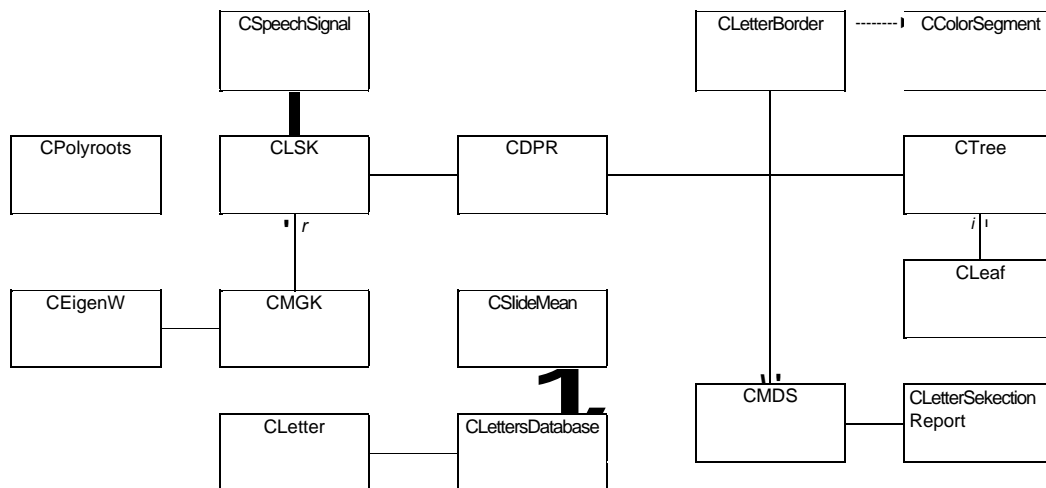


Рис.3. Взаимодействие классов

Краткое описание классов:

CSpeechSignal - класс, обеспечивающий интерфейс доступа к речевому сигналу.

CPolyroots - класс поиска всех корней полинома. Класс используется при расчете линейных спектральных корней. CLSK - класс расчета линейных спектральных корней. CEigenVV - класс поиска собственных чисел и векторов. Собственные числа рассчитываются по QR-алгоритму, собственные векторы - по методу Леверье-Фаддеева.

CMGK - класс поиска собственных векторов ковариационных матриц линейных спектральных корней.

CSlideMean - класс поиска скользящих средних линейных спектральных корней.

CLetterBorder - класс, содержащий границы фонем, является основой для автоматического поиска различных параметров.

CColorSegment — класс необходимый для отображения графиков, содержит текущие границы фонем и цвет каждой из фонем. CLeaf - класс, содержащий информацию о вершине дерева принятия решений.

CTree - класс, содержащий структуру бинарного дерева. Используется при построении дерева принятия решений.

CMDS - класс поиска средней ошибки и наилучших параметров разделения методом динамических сгущений.

CLetterSelectionReport - класс отчета, содержащий результаты поиска оптимальных параметров.

CDPR - класс поиска дерева принятия решений.

CLetter - класс содержит информацию о фонемах. Используется для распознавания по СС ЛСК.

CLettersDatabase — база данных речевых фонем, содержит инструменты для распознавания фонем.

Подробное описание классов приведено в следующей главе.

Разработанные классы

CSpeechSignal

Класс, обеспечивающий интерфейс доступа к речевому сигналу.

Атрибуты:

std::vector<double> m_SpeechSignal - вектор значений речевого сигнала

Методы:

- ReadFromFile - чтение сигнала из файла.
- GetAt - функция возвращает элемент в определенной позиции.
- GetSize - функция возвращает количество элементов в речевом сигнале.
- GetSpeechSignal - функция возвращает весь сигнал.
- Clear - очистка речевого сигнала. Листинг

класса приведен в приложении.

CLSK

Класс расчета линейных спектральных корней и доступа к ним. Инициализация объекта требует вектор значений амплитуд речевого сигнала. Атрибуты:

- std::vector<double> m_SpeechSignal - вектор значений речевого сигнала
- std::vector<std::vector<double>> mJLSK - матрица значений ЛСК.

Методы:

- GetLSK - функция возвращает матрицу линейных спектральных корней.
- GetAt - функция возвращает элемент матрицы в определенной позиции, также присутствует перегруженный вариант, он возвращает все десять корней
- GetSize - функция возвращает количество строк в матрице.
- SetSpeechSignal - функция устанавливает речевой сигнал.
- Min и Max - поиск наименьшего и наибольшего значений для определенного столбца матрицы.
- Clear - очистка матрицы.

FindLSK - поиск линейных спектральных корней. Алгоритм поиска представлен на рис.4:

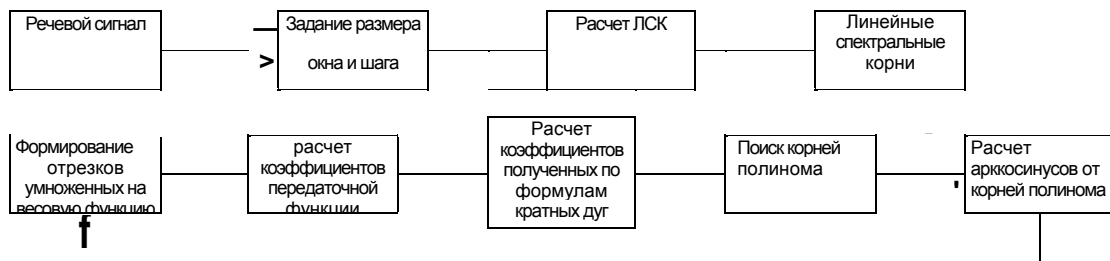


Рис.4. Расчет линейных спектральных корней.

Для расчета линейных спектральных корней необходимо задать параметры модели - размер окна, шаг и порядок модели. Для заданных параметров производится расчет ЛСК: формируются отрезки, взвешенные по Хэммингу, рассчитываются коэффициенты полинома передаточной функции и коэффициенты, полученные по формулам кратных дуг. Далее вычисляются корни полинома, с этими коэффициентами. И в завершении, для вычисления ЛСК от полученных корней вычисляются арккосинусы.

Вспомогательные функции:

- Segment - формирует отрезок, умноженный на весовую функцию, по заданному размеру окна, шагу и порядковому номеру отрезка.
- TransmissionFunctionFactors - расчет коэффициентов передаточной функции, их количество определяется порядком модели.
- PolynomFactors - формирование вектора коэффициентов, полученных по формулам кратных дуг.
- FindOneLSK - функция рассчитывает ЛСК для одного отрезка.

Листинг класса приведен в приложении.

CPolyroots

Для поиска корней полинома/[^] используется метод Ньютона. После нахождения корня, вычисляется следующий корень от полинома $g(x) = f(x)/(x-r)$ и т.д. до тех пор, пока не будут найдены все корни. При инициализации объекта задается точность поиска, начальное приближение и коэффициенты полинома.

После можно вызывать функцию поиска корней, она их найдет и возвратит вектор корней, упорядоченных по возрастанию. Атрибуты:

- double m_X0 - начальное приближение.
- double m_Accuracy - точность поиска
- std::vector<double> m_Factors - коэффициенты полинома.

Методы:

- FindRoots - поиск всех корней полинома.

Вспомогательные функции:

- NthDerivative - вычисление значения производной заданного порядка.
- NumberOfRoots - вычисление количества действительных корней.
- Root - поиск одного корня полинома.
- PolynomDivision - деление полинома на полином первой степени.

Листинг класса приведен в приложении.

CSlideMean

Класс поиска скользящих средних линейных спектральных корней и доступа к ним. По заданному усреднению для уже полученных линейных спектральных корней рассчитывается скользящее среднее.

Атрибуты:

- UINT m_Average - усреднение.
- std::vector<std::vector<double>> m_SS - матрица СС ЛСК.

Методы:

- GetSlideMean - возвращает матрицу СС ЛСК.
- GetAt - функция возвращает элемент матрицы в определенной позиции, также присутствует перегруженный вариант, он возвращает все десять значений скользящего среднего.
- GetSize - функция возвращает количество строк в матрице.
- SlideMean — вычисление СС ЛСК. Алгоритм вычисления продемонстрирован на рис.5:

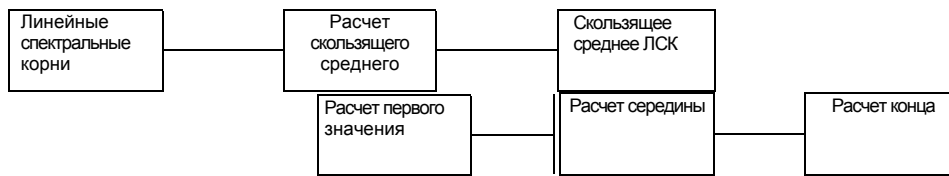


Рис.5. Расчет скользящего среднего линейных спектральных корней

Сначала рассчитывается первое значение скользящего среднего, это среднее по первым элементам линейных спектральных корней. Далее производится расчет середины. Этот метод реализован таким образом, что из предыдущей полученной суммы вычитается «значение слева» и прибавляется «значение справа». Расчет конца выделен в отдельный метод для имитации тишины после сигнала. Результатом вычислений будет матрица таких же размеров, что и матрица линейных спектральных корней.

- Min и Max - поиск наименьшего и наибольшего значений для определенного столбца матрицы.

Вспомогательные функции:

- SummaO - расчет первого значения.
- Summa - расчет середины
- Summa 1 - расчет конца.

Листинг класса приведен в приложении.

CMGK

класс поиска собственных векторов ковариационных матриц линейных спектральных корней и доступа к ним.

Атрибуты:

- UINT m_Average - усреднение
- std::vector<double> msumma - вектор значений промежуточной суммы
- std::vector<std::vector<double> > m_vals - матрица собственных чисел.
- std::vector<std::vector<std::vector<double> > > m_vecs - вектор матриц собственных векторов.

Основные функции:

EigenVectors - расчет всех собственных векторов. Алгоритм расчета представлен на рис.6:

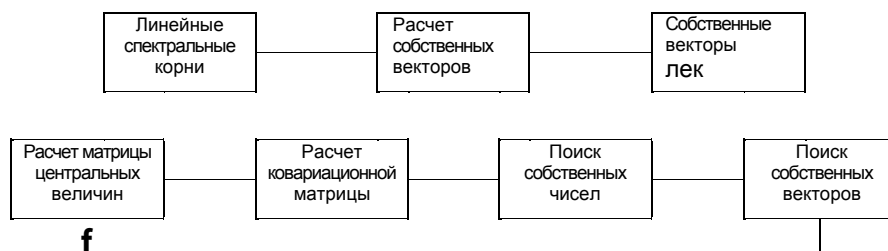


Рис.6. Расчет собственных векторов линейных спектральных корней

Матрица центрированных случайных величин формируется по массивам линейных спектральных корней, на ее базе рассчитывается ковариационная матрица. Далее производится расчет собственных чисел и векторов.

- GetValAt - функция возвращает собственное число в определенной позиции.
- GetVecAt - функция возвращает определенное значение собственного вектора.
- Min и Max - поиск наименьшего и наибольшего значений для определенного столбца матрицы.
- Clear - очистка собственных векторов.
- MatrixMultiplection - перемножение двух матриц.
- TransMatrix - транспонирование матрицы.

Вспомогательные функции:

- Mean - расчет математического ожидания.
- CenterValueMatrix - расчет матрицы центрированных величин.
- KovMatrix - расчет ковариационной матрицы.
- MatrixMultiplection - перемножение двух матриц.
- TransMatrix - транспонирование матрицы.

Листинг класса приведен в приложении.

CEigenVV

Класс поиска собственных чисел и векторов. Расчет собственных чисел производится по QR-алгоритму, собственных векторов - по методу Леверье-

Фаддеева. Объект класса инициализируется квадратной матрицей. Затем пользователь класса должен вызвать метод поиска собственных чисел и векторов.

Атрибуты:

- `std::vector<double> m_p` - вектор значений p . Необходим для метода Ле-верье-Фаддеева.
- `std::vector<std::vector<double> > m_A` - матрица A . Необходима для метода Леверье-Фаддеева.
- `std::vector<std::vector<std::vector<double> > > m_B` - вектор матриц B . Необходим для метода Леверье-Фаддеева.
- `std::vector<double> m_vals` - вектор найденных собственных чисел.
- `std::vector<std::vector<double> > m_vecs` - матрица найденных собственных векторов. Собственные векторы расположены в столбцах матрицы. Методы:
- `FindValsVecs` - поиск собственных чисел и векторов.
- `GetVals` — функция возвращает вектор собственных чисел.
- `GetVecs` - функция возвращает матрицу собственных векторов.

Вспомогательные функции:

- `Sp` - сумма значений главной диагонали.
- `MatrixMultipection` - перемножение двух матриц.
- `MatrixAddition` - сложение двух матриц.
- `MatrixOnDigitMultipection` - умножение матрицы на число.
- `VectorMultipection` - умножение вектора на число.
- `VectorAddition` - сложение двух векторов.
- `TransMatrix` - транспонирование матрицы.
- `Norm` - возвращает норму вектора.
- `Normirovka` — нормировка собственных векторов.
- `FormB` — формирование матриц B .
- `FindP` - поиск вектора p .
- `EigenVals` - поиск вектора собственных значений.

- EigenVecs - поиск матрицы собственных векторов.

Листинг класса приведен в приложении.

CLetterBorder

Класс, содержащий границы фонем. Является основой для автоматического поиска различных параметров. Атрибуты:

- Begin - начало фонемы.
- End — конец фонемы.

CMDS

класс поиска средней ошибки и наилучших параметров разделения методом динамических сгущений. Инициализация объекта класса осуществляется по речевому сигналу.

Атрибуты:

- CLSK m_LSK -линейные спектральные корни.
- CSlideMean m_SlideMean - скользящее среднее линейных спектральных корней.
- UINT m_min - минимальное количество точек, по которому производится сравнение сгущений.
- UINT m_cSize - значение размера окна.
- UINT m_cStep - значение шага.
- UINT m_SignalLength - длина речевого сигнала.

Методы:

- FindErrorsStep - расчет средней ошибки для фиксированного шага и переменного размера окна.
- FindErrorsSize - расчет средней ошибки для фиксированного размера окна и переменного шага.

Схема этих двух методов приблизительно одинакова, поэтому обобщенный алгоритм расчета средней ошибки представлен на рис.7:

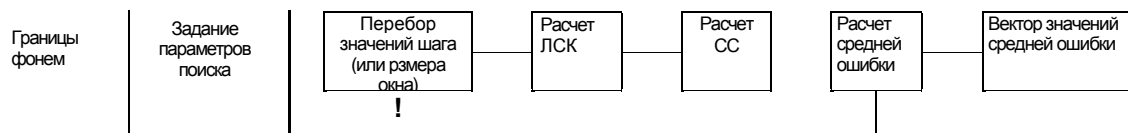


Рис.7. Алгоритм расчета средней ошибки

FindOptimal - перебор всех значений размера окна и шага, заданных в диапазоне. Для каждого значения ищется ЛСК и СС ЛСК. Для каждой возможной плоскости находится ошибка перекрытия фонем. Все эти значения записываются в упорядоченный по возрастанию ошибки, вектор.

Вспомогательные функции:

ErrorsSumma - возвращает суммарную ошибку перекрытия двух фонем Mean - вычисляет математическое ожидание в указанном диапазоне. Error - ошибка перекрытия двух сгущений.

CLetterSelectionReport

класс отчета, содержащий результаты поиска оптимальных параметров.

Атрибуты:

- UINT m_N - количество значений.
- UINT mJSize - размер окна.
- UINT m_Step - размер шага.
- UINT mxLSK - ЛСК по оси абсцисс.
- UINT m__yLSK- ЛСК по оси ординат.
- double m_S - суммарная ошибка.

CLeaf

класс, содержащий информацию о вершине дерева принятия решений.

Атрибуты:

- std::vector<int> letters - номера фонем, которые будут содержаться в вершине дерева.
- std::vector<int> dimensions - пространство ЛСК, которое лучшим образом отделяет фонемы равного уровня иерархии, double Error- оценка разделения фонем.

- CPointIO Center - центр тяжести фонем, содержащихся в letters.
- double Radius - радиус от центра тяжести фонемы до ее максимально отдаленной точки.

Методы: fk

GetLeafInfo - возвращает строку, содержащую информацию о вершине.

CTree

класс содержащий структуру бинарного дерева. Используется при построении дерева принятия решений.

Атрибуты:

CLink* Head - указатель на вершину дерева. CLink - это класс, характеризующий связь вершин.

Методы:

- Clear - Очистка дерева
- Add - добавление вершины в дерево.
- GetHead — вернуть «корень» дерева.

*

CDPR

класс поиска дерева принятия решений.

Атрибуты:

- UINT nDimensions - размерность пространства ЛСК, по которому будет строиться дерево.
- std::vector<std::vector<std::vector<double> > > mJLSKLetterBorders - вектор матриц ЛСК.
- std::vector<std::vector<int> > cloud1 - матрица всех возможных комбинаций фонем.
- std::vector<std::vector<int> > cloud2 - матрица всех возможных комбинаций фонем в соответствии с cloud1.

ф

Методы:

FindTree — построение дерева.

Вспомогательные функции:

- FindLetters - формирование вектора матриц ЛСК.
- AddNode - добавление вершины в дерево.
- FormAUCombinations - формирование всех возможных комбинаций фонем.
- Combination - формирование одной комбинации фонем.
- EvklidDistanceSqr - вычисление квадрата эвклидова расстояния.
- Mean — вычисление математического ожидания.
- Clear - очистка промежуточных данных.
- MinError - поиск вершины с наименьшей ошибкой.
- Err - расчет ошибки перекрытия двух сгущений.

CLetter

класс содержит информацию о фонеме. Используется для распознавания по значениям СС ЛСК.

Атрибуты:

- std::vector<double> Summa - сумма значений СС ЛСК для этой фонемы
- UINT N — количество значений в сумме
- CPoint 10 Center — десятимерный центр тяжести фонемы
- std::vector<CPoint10> ExtrimePoints - крайние точки фонемы
- std::vector<int> LSK - используемое пространство для определения данной фонемы.
- double Radius — радиус от центра тяжести до дальней крайней точки.
- AnsiString LetterName - название фонемы.

Методы:

- CalculateCenter - вычисление центра тяжести фонемы.
- CalculateRadius — вычисление радиуса фонемы.

CLettersDatabase

база данных речевых фонем, содержит инструменты для распознавания фонем.

Атрибуты:

`std::vector<CLetter> m_letters` — вектор содержащий информацию о каждой фонеме в базе данных. Методы:

- `AddLetter` - добавить фонему в базу данных, если такая фонема уже есть, то происходит модификация информации о фонеме.
- `Clear` - очистка базы данных о фонеме.
- `RecognizeBySlideMean` - распознать значения СС ЛСК на принадлежность к какой- либо фонеме. Распознавание производится по СС ЛСК.

`RecognizeByDPR` - распознать значения ЛСК или СС ЛСК на принадлежность к какой- либо фонеме. Распознавание производится по ДПР.

Приложение 5. Листинг программы

Класс CSpeechSignal

```

//speechsignal.h
#pragma once W          #ifndef
SpeechSignalH #define
SpeechSignalH

#include <vector>
#include <fstream>
#include <string>
#include <stdlib.h>

class CSpeechSignal
{
public:
    CSpeechSignal(){} ;
    ~CSpeechSignal(){} ;

public:
    //чтение сигнала из файла
    void ReadFromFile(char* FileName);
    //функции доступа
    double GetAt(unsigned int l)const{return m_SpeechSignal[l]/}
    void Clear () {m_SpeechSignal. clear ();};
    unsigned int GetSize()const{return m_SpeechSignal. size ();}
    std::vector<double> GetSpeechSignal() const{return m_SpeechSignal;}
private:
    std::vector<double> m_SpeechSignal;
}; #endif

//speechsignal.cpp
#include "SpeechSignal.h"
//чтение сигнала из файла
void CSpeechSignal::ReadFromFile(char* FileName)
{
    std::ifstream FileStream(FileName);
    std::string str;
    char* a;
    while(std::getline(FileStream, str))
    {
        const char* s = str.c_str();
        m_SpeechSignal.push_back(strtod(s, &a));
    }
}

```

Класс CLSK

J/sk.h

```

#pragma once Imclude
<vector> finclude <math.h>
finclude "polyroots.h"
finclude "LetterBorders.h"

*
#ifndef LSK_H
#define LSK_H

class CLSK
{
    //конструктор и деструктор
}

```

```
public:

CLSKO ; //инициализирует речевой сигнал вектором уже полученного речевого сигнала
CLSK(const std::vector<double>& SpeechSignal); ~CLSK(void); private: //Функция
формирует i-ый отрезок, size - размер "окна", step - шаг по сигналу
    std::vector<double> Segment(UINT size, UINT step, UINT l); //функция ищет
коэффициенты передаточной функции. Order - порядок полинома, //segment - отрезок
    std::vector<double> TransmissionFunctionFactors(UINT order,
std::vector<double> segment); //функция формирует вектор коэффициентов,
полученных по формулам кратных дуг
    std::vector<double> PolynomFactors(std::vector<double> a);
//функция ищет ИСК для одного отрезка
    std: : vector<double> FmdOneLSK(std: : vector<double> factors);
public:
//функция ищет все ЛСК. Кол-во строк = N+1. Кол-во столбцов = order //N - кол-во
"окон", size - размер "окна", step - шаг по сигналу, order -//порядок полинома
    void FindLSK(UINT N, UINT size, UINT step, UINT order);
//функция возвращает матрицу ЛСК
    std: :vector<std::vector<double> > GetLSKO const{return m_LSK; }
//функция возвращает iый jый элемент матрицы ЛСК
    double GetAt(UINT i, UINT j) const{return m_LSK[l] [j]; }
//функция возвращает количество строк в матрице ЛСК
    UINT GetSize()const{return m_LSK.size (); }
//функция устанавливает новый речевой сигнал
    void SetSpeechSignal(const std::vector<double>& SpeechSignal){m_SpeechSignal =
SpeechSignal; }
    CPomt10 GetAt(UINT l) ;
    double mm (UINT j ) ;
    double max(UINT j);
    void Clear (){m_LSK.clear (); }

private:
    std: :vector<double> m__SpeechSignal; //Речевой сигнал
    std: :vector<std::vector<double> > m_LSK; }; //Линейные спектральные корни

#endif

//lsk.cpp
#include "lsk.h"
CLSK::CLSK() { }

CLSK::CLSK(const std::vector<double>& SpeechSignal) :
m_SpeechSignal(SpeechSignal)
{
}

CLSK::~~CLSK(void)
{
}

// Функция формирует отрезки взвешенного по Хеммингу сигнала в вектор
std::vector<double> CLSK::Segment(UINT size, UINT step, UINT l)
{
    std::vector<double> segment(size);
    for(UINT k=0, k = i*step; j < size; ++j, ++k)
```

```

        segment[i;j] = m_SpeechSignal [к] * (0.54f + 0.46f *
cos(2*3.14159265358979*]/(size-1)));
        return segment; }

//Функция позволяет получить коэффициенты {a} полинома передаточной функции.
//Порядок полинома определяется переменной order. В качестве аргумента нужно
//подставлять вектор
std::vector<double> CLSK::TransmissionFunctionFactors(UINT order,
std::vector<double> segment)
{
    std::vector<double> factors(order+1, 0);
    std::vector<double> c(order+1, 0);
    std::vector<double> b(order+1, 0);
    double re = 0;
    double alpha = 0;

    for(UINT i=0; i <= order; ++i)
        for(UINT j = 0; : < segment.size () - i; ++j) c[i]
            += segment[j] * segment[j+1];

    if(c[0] != 0)
        re = -c[1] / c[0];
    else
        re = 0;
    factors[0] = 1;
    factors[1] = re;

    alpha = c[0] - re*re * c[0];
    for(UINT i=1; i < order; ++i) {
        for(UINT : = 1; : <= 1 + i; ++: ) b[i]
            = factors[i+:-3];

        double s = 0;
        for(UINT : = 0; : <= 1; + + :)
            s += c[i + :-j] * factors [:];

        if(alpha != 0)
            re = -s / alpha;
        else
            re = 0; for(UINT : = 1; : <= 1; ++j)
            factors[i] += re*b[i]; factors[i+1] =
            re; alpha -= re*re * alpha; }
    return factors; }

//функция формирует вектор коэффициентов, полученных по формулам кратных дуг
std::vector<double> CLSK::PolynomFactors(std::vector<double> a)
{
    std::vector<double> factors(a.size ());
    factors[0] = -1+ a[2] - a[4] + a[6] - a[8] + a[10];
    factors[1] = 9*a[1] - 7*a[3] + 5*a[5] - 3*a[7] + a[9];
    factors[2] = 50 - 32*a[2] + 18*a[4] - 8*a[6] + 2*a[8];
    factors[3] = -120*a[1] + 56*a[3] - 20*a[5] + 4*a[7];
    factors[4] = -400 + 160*a[2] - 48*a[4] + 8*a[6];
    factors[5] = 432*a[1] - 112*a[3] + 16*a[5];
    factors[6] = 1120 - 256*a[2] + 32*a[4];
    factors[7] = -576*a[1] + 64*a[3];
    factors[8] = -1280 + 128*a[2];
    factors[9] = 256*a[1];

```

```
factorstl0] = 512; return
factors; }
```

// функция ищет ЛСК для одного "окна"

```
std::vector<double> CLSK::FindOneLSK(std::vector<double> factors)
{
    CPolyroots p(0.1, 0.0000000000001, factors);
    std::vector<double> roots = p.FmdRoots ();
    for(UINT i = 0; i < roots.size (); ++i)
        roots [i] = acos(roots[i]); return
    std::vector<double>(roots.rbegin(), roots.rend());
}
```

//формирует матрицу из ЛСК

```
void CLSK::FindLSK(UINT N, UINT size, UINT step, UINT order)
{
    m_LSK.resize(N+1) ;
    for(UINT i=0; i <= N; ++i) m_LSK[i] = FindOneLSK (PolynomFactors
    (TransmissionFunctionFactors(order, Segment(size, step, i)))); }

double CLSK::mm (UINT j) {
    double m = 10;
    for(UINT i = 0; i < m_LSK.size();
    ++i) if(m_LSK[i][j] < m) m = m_LSK[i] [j] ;
    return re-
```

```
double CLSK::max(UINT j) {
    double m = 0;
    for(UINT i = 0; i < m_LSK.size(); ++i)
    if(m_LSK[i][j] > m) m = m_LSK[i] [j] ; return m;
```

```
CPomtIO CLSK::GetAt (UINT i) {
    std::vector<double> v;
    for(UINT j = 0; j < 10;
    ++j) v.push_back(m_LSK[i][j]);
    return CPomtIO (v); }
```

Класс CSlideMean

//slidemean.h

```
#pragma once
#include <vector>
#include <math.h>
#include "LSK.h"
#include "LetterBorders.h"
```

```
*
#ifdef SLIDE_MEAN_H
#define SLIDE_MEAN_H
```

```
class CSlideMean
{
    // конструкторы и деструктор
```

```
public:
    CSlideMean();
    ~CSlideMean();
private:
    // расчет первого значения
    //nLSK - номер ЛСК
    double SummaO(UINT nLSK, const CLSK& LSK);
    //расчет середины //index - индекс, prev -предыдущая
    сумма
    double Summa(UINT index, UINT nLSK, double prev, const CLSK& LSK); //расчет
    конца //index - индекс, prev -предыдущая сумма
    double Summa1(UINT index, UINT nLSK, double prev, const CLSK& LSK); public:
    //функции доступа
    double GetAt(UINT i, UINT j) const{return m_SS[i] [j];}
    CPomt10 GetAtf(UINT i);
    std::vector<std::vector<double> > GetSlideMean()const{return m_SS;}
    UINT GetSize()const{return m_SS.size() - m_Average;}

    //функция вычисляет скользящее среднее для ЛСК //LSK
    - ЛСК, Average - усреднение
    void SlideMean(const CLSK& LSK, UINT Average);

    UINT mm (UINT i, UINT begin, UINT end);
    UINT max(UINT j,  UINT begin, UINT end);
    void Clear (){m_SS.clear () ; }
private:
    UINT m_Average; // усреднение
    std::vector<std::vector<double> > m_SS; // матрица CC };
#endif

//slidemean.cpp
#include "SlideMean. h"

CSlideMean::CSlideMean()
{
}

CSlideMean::~CSlideMean()
{
}

// расчет первого значения
//nLSK - номер ЛСК
double CSlideMean::SummaO(UINT nLSK, const CLSK& LSK)
{
    double summa = 0;
    for(UINT i=0; i < m_Average; ++i) summa
        += LSK.GetAt(i, nLSK);
    return (double)summa; }

//расчет середины
//index - индекс, prev -предыдущая сумма
double CSlideMean::Summa(UINT index, UINT nLSK, double prev, const CLSK& LSK)
{
    return (prev - LSK.GetAt(index - 1, nLSK) + LSK.GetAt(index + m_Average -nLSK));
}
```

```
//расчет конца
//index - индекс, prev -предыдущая сумма
double CSLideMean::Summa(UINT index, UINT nLSK, double prev, const CLSK& LSK)
{
    return (prev - LSK.GetAt(index - 1, nLSK)); }

//функция вычисляет скользящее среднее для ЛСК •
//LSK - ЛСК, Average - усреднение
void CSLideMean::SlideMean(const CLSK& LSK, UINT Average) {

    m_Average = Average;

    m_SS.resize(LSK.GetSize0, std::vector<double>(10, 0));

    std::vector<double> v(10);

    for(UINT j = 0; j < 10; ++j) {
        v[j] = Summa(0, LSK);
        m_SS[0][j] = v[j] / m_Average; }

    for(UINT i=1; i < (LSK.GetSize() - m_Average); ++i)
        for(UINT j = 0; j < 10; ++j) {
            v[j] = Summa(i, LSK);
            m_SS[i][j] = v[j] / m_Average; }

    for(UINT i = LSK.GetSize() - m_Average; i < LSK.GetSize(); ++i)
        for(UINT j = 0; j < 10; ++j)
        {
            v[j] = Summa(i, LSK);
            m_SS[i][j] = v[j] / m_Average; }
    }

UINT CSLideMean::min(UINT j, UINT begin, UINT end) {
    double m = 1000000; UINT result =
    begin; for(UINT i = begin; i < end;
    ++i) if(m_SS[i][j] < m) {
result = i; m = m_SS[i][j];
    } return result; }

UINT CSLideMean::max(UINT j, UINT begin, UINT end) {
    double m = -1000000; UINT result =
    begin; for(UINT i = begin; i < end;
    ++i) if(m_SS[i][j] > m) {
result = i; m = m_SS[i][j];
    } return result; }
```



```
CPomtlO CSlideMean::GetAt(UINT x) {
    std::vector<double> v;
    for(UINT i = 0; i < 10; ++i)
        v.push_back(m_SS[i][0]);
    return CPointlO(v); }
```

Класс CMGK

```
#pragma once #include
<vector> #include
<math.h> #include
"LSK.h" #include
"eigenvv.h"

#ifdef MGK_H
#define MGK_H

//разделение речевых фонов
//метод главных компонент
//собственные векторы
class CMGK
{
public:
    CMGK();
    ~CMGK(); private : //перемножение
    матриц. M1 и M2 - матрицы
        std::vector<std::vector<double> >
    MatnxMultiplec-tion(std::vector<std::vector<double> > M1,
    std::vector<std::vector<double> M2); //транспанирование матрицы M - матрица
        std::vector<std::vector<double> >
    TransMa-trix(std::vector<std::vector<double> > M);

    //математическое ожидание
    //index - откуда начинать
        std::vector<double> Mean(UINT index, const CLSK& LSK);
    //расчет первой суммы
        std::vector<double> Summa0 (const CLSK& LSK);
    //расчет середины
        std::vector<double> Summa1(UINT index, const CLSK& LSK);

    //расчит матрицы центрированных случайных величин
        std::vector<std::vector<double> > CenterValueMatrix(UINT index, const
    CLSK& LSK);
    //расчет ковариационной матрицы cvm - матрица центрированных величин
        std::vector<std::vector<double> > KovMa-
    trix(std::vector<std::vector<double> > cvm);
public:
    //расчет всех собственных векторов Average - усреднение
        void Eigenvectors(const CLSK& LSK, UINT Average);
    //функции доступа
        double GetValAtfUINT i, UINT j) const{return m_vals[i] [j];}
        double GetVecAt(UINT k, UINT i, UINT j)const{return m_vecs[k] [i] [3];}
        UINT GetSize ()const{return (m_vecs. size () - m_Average);}

        double mm (UINT i, UINT j);
        double max(UINT i, UINT j);
        void Clear () {
            m_vals.clear();
```

```

    m_vecs. clear (); }
private:
    UINT m_Average;
    std::vector<double> m_summa;
    std::vector<std::vector<double> > m_vals;
    std::vector<std::vector<std::vector<double> > > m_vecs; };
#endif

//mgk.cpp
#include "MGK.h"

CMGK::CMGK()
: m_summa(10, 0){
}

CMGK::~~CMGK()
{
}

//перемножение матриц. M1 и M2 - матрицы
std::vector<std::vector<double> >
CMGK::MatnxMultiplection (std::vector<double> > M1,
std::vector<std::vector<double> > M2)
{
    unsigned mt rows = M1.size (),■
    unsigned int n = M1 [0]. size ();
    unsigned mt cols = M2 [0], size ();
    std::vector<std::vector<double> > M(rows);

    for(unsigned int i = 0; i < rows; ++i)
        for(unsigned int j = 0; j < cols; ++j)
        {
            double summa = 0;
            for(unsigned int k = 0; k < n; ++k) summa += M1[i][k] +
M2[k][j]; M[i].push_back(summa); } return M; }

//транспанирование матрицы M - матрица
std::vector<std::vector<double> > TransMatrix(std::vector<std::vector<double> >
M)
{
    std::vector<std::vector<double> > TM(M);
    for(unsigned int i = 0; i < TM.size(); ++i)
        for (unsigned int j = i + 1; j < TM.size(); ++j)
        {
            double temp = TM[i][j]; TM[i][j] =
TM[j][i]; TM[j][i] = temp; } return
TM;

//расчет всех собственных векторов Average - усреднение void
CMGK::eigenvectors(const CLSK& LSK, UINT Average) {
    m_Average = Average;
    m_vals.resize(LSK.GetSize());

```

```

m_vecs.resize(LSK.GetSize()); m_summa =
std::vector<double>(10, 0);

for(UINT i = 0; i < (LSK.GetSize() - m_Average); ++i) {
    CEigenVV vv(KovMatrix(CenterValueMatrix(i, LSK)));
    vv.FmdValsVecs ();
    m_vals[i] = vv.GetVals();
    m_vecs[i] = vv.GetVecs();
}
/*
std::ofstream FileStream("e:\\eigen_vals.txt");
for(UINT i = 0; i < LSK.GetSizeO - m_Size; ++i) {
    AnsiString str;
    for(UINT j = 0; j < 10; ++j)
        str += AnsiStringO.FormatFloat("0.000000000000\t", m_vals[i][j]);
    FileStream « str.c_str() « std::endl;

UINT s = m_Size;
for(UINT i = (LSK.GetSizeO - s); i < LSK.GetSizeO; ++i)
{
    --m_Size;
    CEigenVV vv(KovMatrix(CenterValueMatrix(i, LSK)));
    vv.FmdValsVecs ();
    m_vals[i] = vv.GetValsf();
    m_vecs[i] = vv.GetVecs(); }
    m_Size = s;
*/ }

//математическое ожидание
//index - откуда начинать
std::vector<double> CMGK::Mean(UINT index, const CLSK& LSK)
{
    if(index == 0) {
        std::vector<double> mean = SummaO(LSK);
        for(UINT j = 0; j < 10; ++j)
            meant[j] /= m_Average;
        return mean; } else
    {
        std::vector<double> mean = Summal(index, LSK);
        for(UINT J = 0; j < 10; ++j)
            mean[j] /= m_Average; return
mean; } }

//расчет первого значения суммы
std::vector<double> CMGK::SummaO(const CLSK& LSK)
{
    for(UINT i=0; i < m_Average; ++i)
        for(UINT D = 0; D < 10; ++j)
            m_summa[i] += LSK.GetAt(i, j);
    return m_summa;

//расчет середины

```

```

std::vector<double> CMGK::Summal(UINT index, const CLSK& LSK) {
    for(UINT j = 0; j < 10; ++j) {
        m_summa[j] -= LSK.GetAt(index - 1, j);
        m_summa[j] += LSK.GetAt(index + m_Average - 1, j);
    }
    return m_summa;
}

//расчит матрицы центрированных случайных величин
std::vector<std::vector<double> > CMGK::CenterValueMatrix(UINT index, const
CLSK& LSK)
{
    std::vector<std::vector<double> > cvm(m_Average, std::vector<double>(10, 0));
    std::vector<double> mean = Mean(index, LSK); for (UINT i = 0, ii = index;
    i < m_Average; ++i, ++n) for(UINT j = 0; j < 10; ++j )
        cvm[i][j] = LSK.GetAt(ii, j) - mean[j];
return cvm; }

//расчет ковариационной матрицы cvm - матрица центрированных величин
std::vector<std::vector<double> >
CMGK::KovMatrix(std::vector<std::vector<double> > cvm)
{
    std::vector<std::vector<double> > KM(10, std::vector<double>(10, 0));
    for(UINT i=0; i < 10; ++i)
        for(UINT j = i; j < 10; ++j) {
            for(UINT k = 0; k < cvm.size(); ++k)
                KM[i][j] += cvm[k][i] * cvm[k][j];
            KM[j][i] = KM[i][j];
        }
    return KM;
}

double CMGK::mm(UINT i, UINT j) {
    double m = 1000000;
    for(UINT k = 0; k < GetSize(); ++k)
        if(m_vecs[k][i][j] < m) m = m_vecs[k][i][j]; return m; }

double CMGK::max(UINT i, UINT j) {
    double m = -1000000; for (UINT k = 0; k <
    GetSize(); ++k) if(m_vecs[k][i][j] > m) m =
    m_vecs[k][i][j]; return m; }

```

Класс CPolroots

//polyroots.h

```

#pragma once
#include <vector>
#include <math.h>
#include <algorithm>

typedef POLYROOTS H

```

```
ttdefine POLYROOTS_H

class CPolyroots
{
private:
    double m_X0; // начальное приближение
    double m_Accuracy; // точность
    std::vector<double> m_Factors; // коэффициенты полинома
public:
    CPolyroots(double x0, double Accuracy, std::vector<double> factors);
    ~CPolyroots(void) ; //поиск всех корней полинома с коэффициентами factors,
    с точностью epsilon и //начальным приближением x0
    std::vector<double> FindRoots () ;
private:
    //вычисляет значение производной N порядка в точке x для полинома с
    //коэффициентами factors
    double NthDerivative(std::vector<double> factors, double x, unsigned int N);
    //возвращает количество корней полинома с коэффициентами factors
    unsigned int NumberOfRoots(std::vector<double> factors); //поиск корня с
    точностью epsilon, и начальным приближением x0 для полинома с //коэффициентами
    factors
    double root(std::vector<double> factors, double x0, double epsilon);
    //деление полинома с коэффициентами dividend на полином (x - root)
    std::vector<double> PolynomDivision(std::vector<double> dividend, double
    root); }; #endif
    //polyroots.cpp
#include "polyroots .h"

CPolyroots::CPolyroots(double x0, double Accuracy, std::vector<double> factors)
    : m_X0(x0)
    , m_Accuracy(Accuracy)
    , m_Factors(factors) {
}

CPolyroots::~CPolyroots(void)
{
}

//вычисляет значение производной N порядка в точке x для полинома с
//коэффициентами factors
double CPolyroots::NthDerivative(std::vector<double> factors, double x, unsigned
int N)
{
    double y = 0;
    for(unsigned int i = N; i < factors.size(); ++i)
    {
        double p = 1;
        for(unsigned int j = 0; j < N; ++j)
            p *= i - j;
        y += pow(x, i-N) * factors[i] * p;
    }
    return y;

//возвращает количество корней полинома с коэффициентами factors
unsigned int CPolyroots::NumberOfRoots(std::vector<double> factors) {
/*
    unsigned int w = 0;
```

```

    if(factorsf0] == 0)
        ++w; for(unsigned int i = 1; i < factors.size
        ()); ++i)
            if ((factors[i] * factors [l-1]) < 0)
                ++w;

    for(unsigned int i = 1; i < factors.size (); ++i)
        factors[i] *= pow(-1, i);

    for(unsigned int i = 1; i < factors.size (); ++i) if
    ((factors[i] * factors[l-1]) < 0) ++w; return w; */
    return 10; }

//поиск корня с точностью epsilon, и начальным приближением x0 для полинома с
коэффициентами factors
double CPolyroots::root(std::vector<double> factors, double x0, double epsilon)
{
    double x = x0;
    do
    {
        x0 = x;
        x -= NthDerivative(factors, x, 0) / NthDerivative(factors, x, 1);
    }while(fabs((x - x0) / x) > epsilon); return x; //      return (m_X0 = x) ; }
//деление полинома с коэффициентами dividend на полином (x - root)
std::vector<double> CPolyroots::PolynomDivision(std::vector<double> dividend,
double root) {
    std::vector<double> quotient; unsigned
    int size = dividend.size (); for(unsigned
    int i = 0; i < size-1; ++i) {
        std::vector<double> divisor(size - i, 0);
        divisor[size - i - 2] = -root * dividend[size - i - 1];
        divisor[size - i - 1] = dividend[size - i - 1];

        for(unsigned int j = 0; j < (size-i); ++j)
            dividendfj] -= divisorfj];

        dividend.pop_back();
        quotient.push_back(divisor[size - i - 1]); }
    return std::vector<double>(quotient.rbegin(), quotient.rend()); }

//поиск всех корней полинома с коэффициентами factors, с точностью epsilon и на-
чальным приближением x0
std::vector<double> CPolyroots::FindRoots() {
    std::vector<double> factors(m_Factors);
    unsigned int nRoots = NumberOfRoots(factors);
    std::vector<double> roots(nRoots);
    for(unsigned int i = 0; i < nRoots; ++i)
        factors = PolynomDivision(factors, (roots[i] = root(factors, m_X0,
m_Accuracy)));
    std::sort(roots.begin () , roots.end());
    return roots; }

```

Класс CEigenVV

```
//EigenVV.h
#pragma once finclude <vector>
#include <math.h> tminclude
<algorithm> //tminclude "polyroots
.h"

#ifdef EIGEN_VV_H
#define EIGEN_VV_H

class CEigenVV
{
public:
//конструктор, инициализируется квадратной матрицей

    CEigenVV(std::vector<std::vector<double> > A);
    ~CEigenVV(void); private: //вычисление суммы
диагональных элементов матрицы M
    double Sp(std::vector<std::vector<double> > M) ;
//перемножение матриц M1 и M2 - матрицы
    std::vector<std::vector<double> >
MatnxMultiplication(std::vector<std::vector<double> > M1,
std::vector<std::vector<double> > M2); //сложение двух матриц
    std::vector<std::vector<double> >
MatrixAddition(std::vector<std::vector<double> > M1,
std::vector<std::vector<double> > M2); //умножение всех элементов матрицы на число
digit
    std::vector<std::vector<double> >
MatrixOnDigitMultiplication(std::vector<std::vector<double> > M,
double digit); //умножение всех элементов вектора на число digit
    std::vector<double> VectorMultiplication(std::vector<double> V, double digit);
//сложение двух векторов
    std::vector<double> VectorAddition(std::vector<double> V1,
std::vector<double> V2); //транспонирование квадратной матрицы
    std::vector<std::vector<double> > TransMatrix(const
std::vector<std::vector<double> > &M); //вычисляет норму вектора
v
    double Norm(std::vector<double> v);

//Формирование вектора матриц B
    std::vector<std::vector<double> > ForraB(std::vector<std::vector<double> A,
double p); //нормировка вектора
    std::vector<double> Normirovka(std::vector<double> v);

//поиск P
    void FindP(); //расчет всех собственных чисел матрицы epsilon -
точность расчета
    void EigenVals(double epsilon);
//расчет всех собственных векторов
    void EigenVecs(); public: //вычисление собственных
чисел и векторов матрицы
    void FindValsVecs();
//функции доступа
    std::vector<double> GetVals() const{return m_vals;}
    std::vector<std::vector<double> > GetVecs()const{return m_vecs;}
private:
```

```

std::vector<double> m_p;
std::vector<std::vector<double> > m_A;
std::vector<std::vector<std::vector<double> > > m_B;

std::vector<double> m_vals;
std::vector<std::vector<double> > m_vecs; };
#endif

//EihenVV.cpp
#include "eigenvv.h"

//конструктор, инициализируется квадратной матрицей
CEigenVV::CEigenVV(std::vector<std::vector<double> > A) : m_A(A)
{
    m_p.resize(A.size () ) ;
    m_B.resize(A.size () - 1) ; }

CEigenVV::~CEigenVV(void) {}

//перемножение матриц M1 и M2 - матрицы
std::vector<std::vector<double> > CEigenVV:
:MatrixMultiplection(std::vector<std::vector<double> > M1,
std::vector<std::vector<double> > M2) {
    unsigned mt rows = M1. size (),•
    unsigned int n = M1[0] . size () ;
    unsigned int cols = M2 [0]. size ();
    std::vector<std::vector<double> > M(rows);

    for(unsigned int i = 0; i < rows; ++i)
        for(unsigned int j = 0; j < cols; ++j) {
            double summa = 0;
            for(unsigned int k = 0; k < n; ++k) summa += M1 [i]
[k] *M2[k][j]; M[i].push_back(summa); } return M;
}

*

//сложение двух матриц
std::vector<std::vector<double> > CEigenVV: :MatrixAddition
(std::vector<std::vector<double> > M1,
std::vector<std::vector<double> > M2) {
    for(unsigned int l = 0; l < M1.size (),• ++l)
        for(unsigned int j = 0; j < M1. size (); ++j)
            M1[i][j] += M2[i][j];
    return M1; }

//умножение всех элементов матрицы на число digit
std::vector<std::vector<double> > CEigenVV: :MatrixOnDigitMultiplection (std:
:vector<std::vector<double> > M, double digit) {
    for(unsigned int l = 0; l < M.size(); ++l)
        for(unsigned int j = 0; j < M[l].size (); ++j)
            M[l][j] *= digit;
    return M; }

```



```

//умножение всех элементов вектора на число digit
std::vector<double> CEigenVV::VectorMultiplection(std::vector<double> V, double
digit)
{
    for (unsigned int i = 0; i < V.size(); ++i) V[i]
        *= digit;
    return V; }

//сложение двух векторов
std::vector<double> CEigenVV::VectorAddition(std::vector<double> VI,
std::vector<double> V2)
{
    for(unsigned int i = 0; i < VI.size (); ++i)
        VI[i] += V2[i];
    return VI;

std::vector<std::vector<double> > CEigenVV::TransMatrix(const
std::vector<std::vector<double> >& M)
{
    std::vector<std::vector<double> > MM(M);
    for(unsigned int i = 0; i < M.size(); ++i)
        for (unsigned int j = i+1; j < M.size(); ++j) {
MM[i][j] = M[j][i]; MM[j][i] = M[i][j];
} return MM; }

//вычисление суммы диагональных элементов матрицы M double
CEigenVV::Sp(std::vector<std::vector<double> > M) {
    double summa = 0;
    for (unsigned int i = 0; i < M.size(); ++i)
        summa += M[i][i];
    return (-summa);

std::vector<std::vector<double> > CEigenVV::FormB (std:
:vector<std::vector<double> > A, double p) {
    for(unsigned int i = 0; i < A.size(); ++i)
        A[i][i] += p;
    return A; }

void CEigenVV::FmdP () {
    std::vector<std::vector<double> > A(m_A);

    for(unsigned int i = 0; i < m_B.size(); ++i) {
        m_p[i] = Sp(A) / d + 1;
        m_B[i] = FormB(A, m_p[i]);
        A = MatnxMultiplection(m_A, m_B[i]); }
    m_p[m_B.size()] = Sp(A) / (m_B.size() + 1);
}

double CEigenVV::Norm(std::vector<double> v) {

```

```

double n = 0;
for(unsigned int i = 0; i < v.size (),• ++i)
    n += v[i] * v[i]; return
sqrt(n); }

//расчет всех собственных чисел матрицы epsilon - точность расчета
void CEigenVV::EigenVals(double epsilon)
{
/*
    std::vector<double> factors = std::vector<double>(m_p.rbegin(),
m_p. rend ());
    factors.push_back(1);
    CPolyroots polynom(10, 0.000000001, factors);
    m_vals = polynom.FmdRoots ();

*/
    std::vector<std::vector<double> > A = m_A;
    unsigned int size = A.size (),•
    for(unsigned int nEigenValue = 1; nEigenValue < size; ++nEigenValue)
    {
        double approximate = 0;
        bool condition = 0; do {
            std::vector<std::vector<std::vector<double> > > H;
            std::vector<std::vector<double> > Q;

            std::vector<std::vector<double> > E(A.size(),
std::vector<double>(A.size (), 0));
            for(unsigned int i = 0; i < E.sizeJ); ++i)
                E[i][i] = 1;

            if(condition)
                A = MatrixAddition(A, MatrixOnDigitMultiplection(E,
-approximate));
            for(unsigned int i = 0; i < A.size () - 1; ++i) {
                std::vector<double> x(A.size (), 0);
                std::vector<double> z(A.size(), 0);
                for(unsigned int j = i; j < A.size(); ++j)
                    x[D] = Af3] fib'
                z[i] = 1;
                std::vector<std::vector<double> > v(A.size(),
std::vector<double>(A.size (), 0));
                v[0] = VectorAddition(x, VectorMultiplection(z,
Norm(x)));

                H.push_back(MatrixAddition(E, MatrixOnDigitMultiplection
(MatrixMultiplection (TransMatrix (v) , v), -2.0 / (Horm(v[0]) * Norm(v[0]))));
                A = MatrixMultiplection(H.back() , A); }
            Q = H[0];
            for(unsigned int i = 1; i < H.size(); ++i) Q
                = MatrixMultiplection(Q, H[i]);

            double temp = A.back ().back();
            if(condition)
                A = MatrixAddition(MatrixMultiplection(A, Q), MatrixOnDigitMultiplection
(E, approximate)); else {
                A = MatrixMultiplection(A, Q);

```

```

        condxtion = 1; }
        approximate = temp; }while(A[A.size() -
1][A.size() - 2] > epsilon);

        m_vals.push_back(A.back().back());
        A.resize(A.size () - 1);
        for(unsigned int i = 0; i < A.size (),• ++i) A[i] .
        resize(A[i]. size () - 1); }
        m_vals.push_back(A[0][0] ) ; }

std::vector<double> CEigenVV::Normirovka(std::vector<double> v) {
    double max = 0;
    for(unsigned int i = 0; i < v.size(); ++i)
        if(fabs(max) < fabs(v[i])) max = v[i];
    for(unsigned int i = 0; i < v.size (),• ++i)
        v[i] /= max;
    return v;

//расчет всех собственных векторов
void CEigenVV::EigenVecs()
{
    for(unsigned int i = 0; i < m_vals. size (); ++i) {
        std::vector<double> u(m_A.size (), 0);
        u[0] = 1;
        for(unsigned int j = 0; j < m_B.size(); ++j)
            u = VectorAddition(VectorMultiplection(u, m_vals[i]),
m_BtD][0]);
        m_vecs.push_back(Normirovka(u));
    }
    m_vecs = TransMatrix(m_vecs);
}

void CEigenVV::FindValsVecs()
{
    FindP();
    EigenVals(0.000001); std:::
    sort(m_vals.begin (), m_vals.end());
    EigenVecs(); }

```