МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
ИМЕНИ МУХАММАДА АЛ-ХОРАЗМИЙ

М.М.МУСАЕВ, Ф.А.РАХМАТОВ, А.К.ЭРГАШЕВ

АЛГОРИТМЫ РАСПОЗНАВАНИЯ РЕЧИ

Учебное пособие

Ташкент-2018

УДК 621.391

М.М.Мусаев, Ф.А.Рахматов, А.К.Эргашев. Алгоритмы распознавания речи: Учебное пособие. ТУИТ. – Ташкент: Изд-во _____, 2018.– 232 с.

В учебном пособии рассматриваются вопросы анализа и распознавания речевых сигналов, а также синтеза речи. Цифровая обработка речевых сигналов включает алгоритмы фильтрации звукового сигнала, вычисления параметров речи, спектральный анализ и параметрическое представление речевого сигнала. Задачи распознавания включают интеллектуальные алгоритмы нейронных сетей, вычисления коэффициентов линейного предсказания, скрытых Марковских моделей, методы динамического программирования. Рассмотрены также системы анализа и синтеза речи, голосового интерфейса при обработке речи. Большое внимание уделено качеству распознавания и инструментарию для разработки систем обработки речи.

Предназначено для студентов направления подготовки бакалавров 5330500-Компьютерный инжиниринг ("Компьютерный инжиниринг", "ИТ-сервис", "Мультимедийные технологии") и магистров 5А330501-«Компьютерный инжиниринг («Проектирование компьютерных систем», «Проектирование прикладных программных средств», «Информационные и мультимедийные технологии», «Информационная безопасность, криптография и криптоанализ»).

Рекомендовано Учебно-методическим советом ТУИТ в качестве учебного пособия для бакалавров и магистров соответствующих специальностей.

Рецензенты: Заведующий кафедрой «Системы обработки информации и управления» ТГТУ имени Ислама Каримова, д.т.н., проф. И.Х. Сиддиков;

Заведующий кафедрой «Мультимедийные технологии» ТУИТ имени Мухамада ал-Хоразмий, к.т.н., доц. Э.Ш.Назирова.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Одной из естественных форм взаимодействия для человека является речь. Голосовой интерфейс может улучшить существующий пользовательский интерфейс - он обеспечивает более удобный и менее ограниченный способ взаимодействия человека с компьютером. Качественный голосовой интерфейс изменяет способ, а следовательно и эффективность взаимодействия пользователя с системой. Распознавание речи играет ключевую роль в организации удобного интерфейса при взаимодействии человека и компьютерного оборудования. В таких компьютеризированных системах, как речевое управление агрегатами, идентификация личности, IP-телефония, военные команды управления вооружением, прием заявок в справочных службах, автоматизированная стенография, распознавание отдельных слов и фраз играет основную роль в повышении эффективности технических систем.

Созданные к настоящему времени компьютерные голосовые системы анализа и синтеза речи человека являются этому яркими примерами, подтверждающими необходимость внедрения речевых технологий, в частности голосовых интерфейсов.

Голосовой интерфейс является необходимой компонентой, когда речь идет о создании комфортных условий жизни для людей с нарушениями зрения или опорно-двигательного аппарата, а также специалистам утратившим возможность использовать естественные средства общения в результате профессионального заболевания, травмы или увечья. Такие системы со временем войдут в повседневный быт в процессе реализации концепции так называемых «умных домов».

Наиболее перспективной областью применения речевых технологий является в настоящее время телекоммуникация. Некоторые из этих технологий сыграют огромную роль в этой коммуникационной революции, но одним из ключевых моментов станет развитие речи. Благодаря

использованию синтеза речи и технологии распознавания, телефонные станции используются как индивидуальные терминалы для связи с компьютерными системами. Ожидается, что в будущем техника распознавания говорящего будет широко использоваться как метод проверки идентичности в банковском деле, сферах обслуживания, службах информации.

Не все задачи разработки системы распознавания речи в настоящее время можно считать решенными. Проблема разработки таких систем является достаточно сложной и комплексной, что требует от разработчика знаний в различных предметных областях, таких как компьютерные науки, численные методы, методы цифровой обработки сигналов, лингвистика и психология поведения человека. Даже при наличии совершенных средств проектирования, разработка эффективного голосового пользовательского интерфейса требует от его создателей детального понимания как задач, выполняемых системой, так и психологии пользователей системы.Таким образом, задача распознавания речи очень сложна и решена лишь отчасти, задача синтеза речи намного проще (хотя и там есть немало проблем, ждущих своего решения). Уже сейчас владельцы мобильных телефонов могут общаться с автоматической сервисной службой для определения остатка средств на счету, переключения тарифных планов, подключения или отключения услуг. Сервисная служба общается голосом с применением технологий синтеза речи. Выпущено немало детских игрушек, «говорящих» человеческим голосом. В этих игрушках также применяются простейшие синтезаторы речи или цифровые магнитофоны.

Синтезаторы речи применяются в различных голосовых системах предупреждения, устанавливаемых в автомобилях и самолетах. Такие системы позволяют привлечь внимание человека к возникновению той или иной критической ситуации, не отвлекая его от процесса управления автомобилем, самолетом или другим аналогичным средством.

Данное учебное пособие направлено на изучение содержания предмета, поможет войти в круг проблем распознавания речи, уяснить, какую роль она играет в жизни современного человека.

Дисциплина основывается на знаниях, полученных при изучении: высшей математики, языка программирования C++, информатики. Знания и навыки, полученные в процессе изучения алгоритмов распознавания речи, могут использоваться в других дисциплинах, где необходимо решение задач идентификации и распознавания.

В результате обучения по дисциплине «Алгоритмы распознавания речи» студенты должны достигнуть следующих уровней подготовленности:

1. Иметь представление:

- о направлениях развития речевых технологий;
- об основных алгоритмах обработки речи;
- о системах распознавания речи.

2. Знать:

- программное и аппаратное обеспечение речевых систем;
- инструментальные средства разработки программ;
- основные этапы создания и организации речевых систем;
- программные реализации существующих алгоритмов распознавания речи.

3. Уметь:

- использовать компьютерные технологии для распознавания и синтеза речи;
- использовать вычислительные методы для обработки речевых сигналов;
- использовать диалоговые интерфейсы и инструментальные средства;
- использовать сетевые технологии для систем распознавания речи.

ГЛАВА 1. РЕЧЬ И ФОНЕТИЧЕСКИЕ ПРОЦЕССЫ

1.1. Речевой тракт человека

Речевые органы человека отличаются совершенством, с их помощью человек может не только говорить и петь, но и подражать звукам, издаваемым различными животными.

В этом разделе мы рассмотрим роль в формировании звуков отдельных речевых органов человека, таких как голосовые связки, гортань, язык.

Схема речевого тракта. На рис. 1.1 показана упрощенная схема речевого тракта человека. «Двигателем» этой системы, необходимым для ее функционирования, являются легкие. При выдохе воздух из легких поступает через трахею в гортань, а затем в ротовую и носовую полость.

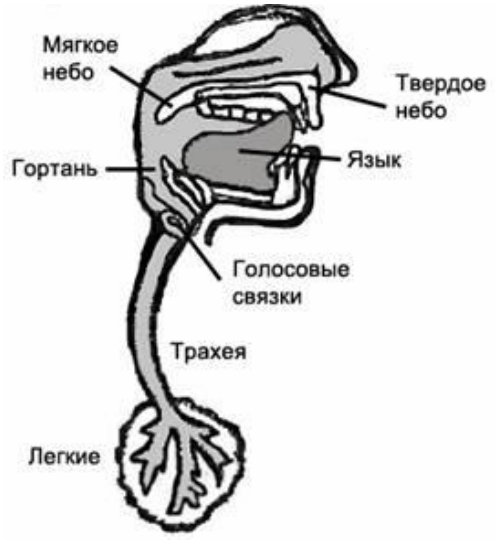


Рис. 1.1. Речевой тракт человека.

Схематически движение воздуха показано на рис. 1.2. На выходе из гортани поток воздуха может раздваиваться, поступая одновременно в носовую и ротовую полость.

Органы, расположенные в ротовой полости, наряду с голосовыми связками, играют решающую роль в формировании звуков. Что же касается

носовой полости, то она служит резонатором, усиливая колебания определенных частот.

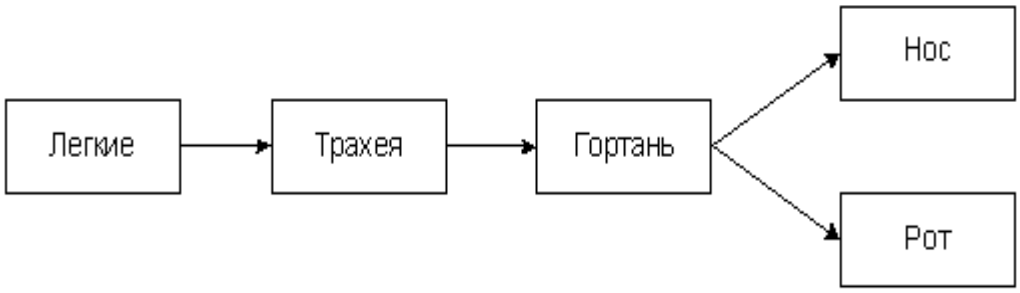


Рис. 1.2. Схема прохождения воздуха при образовании голоса.

Голосовые связки. Колебания голосовых связок, расположенных на входе в гортань, наполняет речь человека голосом, позволяет издавать звук. Голосовые связки не работают, когда человек говорит шепотом. При повреждении голосовых связок в результате болезни или травмы человек способен только шептать.

В зависимости от того, какие звуки и как произносит человек, может работать либо одна из полостей, либо обе полости. Носовые звуки произносятся при закрытом рте. Что же касается остальных звуков, то в их формировании принимают участие как носовая, так и ротовая полость. Размеры, эластичность голосовых связок и сила выходящего из легких воздуха определяют личность говорящего – мужчина, женщина, ребенок.

Активные и пассивные органы речи. Все органы, участвующие в формировании речи, можно разделить на активные и пассивные. При этом в процессе речи активные органы совершают различные движения, формируя звуки. В состав активных органов речи входят: голосовые связки, язык, губы, мягкое небо, язычок, задняя спинка зева, нижняя челюсть.

Пассивные органы речи играют лишь вспомогательную роль. Они, в частности, определяют форму полостей, от которой, в свою очередь, зависят резонансные свойства этих полостей. Следующие органы речи являются пассивными: зубы, альвеолы, твердое небо, верхняя челюсть.

Несмотря на то, что пассивным органам речи отведена вспомогательная роль, их значение нельзя преуменьшать (например, отсутствие нескольких зубов может привести к заметным дефектам речи.

Работа речевого тракта.

Поступая из легких и проходя через гортань, воздух проходит мимо голосовых связок. Колебания этих связок и создают звук, который мы слышим, когда человек говорит или поет. Многочисленные резонаторы, форму которых человек может изменять при помощи активных органов речи, формируют звуковую окраску голоса.

Рассмотрим весь этот процесс подробнее, остановив внимание на некоторых деталях, существенных для систем синтеза и распознавания речи.

Артикуляция. Движения, выполняемые органами речи в процессе произнесения звуков, называются артикуляцией. Артикуляция является сложным процессом, описание которой охватывает много различных факторов. Процесс артикуляции состоит из трех фаз:

- приступ (экскурсия);
- выдержка;
- отступ (рекурсия).

Во время приступа артикуляции органы речи переходят из спокойного состояния в положение, необходимое для произнесения данного звука.

Во время фазы выдержки органы речи сохраняют свое положение, необходимое для произнесения текущего звука.

На фазе отступа органы речи переводятся в спокойное состояние.

Голосовые и шумовые звуки. Колебания связок придают голосу звучание. В этом звучании выделяется так называемый основной тон, или частота основного тона. Значение частоты основного тона зависит от размеров и степени натяжения голосовых связок.

У разных людей могут быть разные размеры связок, поэтому тональность голоса разных людей обычно различается. Регулируя натяжение

связок в процессе артикуляции, человек может менять частоту основного тона.

Помимо голосовых, человек может издавать и шумовые звуки.

Все шумовые звуки можно разделить на два типа: турбулентные и импульсные.

Турбулентные звуки образуются при прохождении звука через сужения речевого тракта. Например, согласные ***с, ф, х, ц, ч, ш, щ*** произносятся без использования голоса, с использованием только турбулентных шумовых звуков.

Импульсные шумовые звуки образуются при резком изменении давления при прерывании струи воздуха. Это происходит, когда произносится такие согласные, как ***п, к, т, д.***

Гармоники. Звук идеально чистого тона содержит колебания только одной частоты. График изменения амплитуды звукового сигнала чистого тона может быть представлен в виде идеальной синусоиды. На практике, однако, звуки с идеально чистым тоном встречаются редко. Если, например, скрипач, пианист и певец возьмут ноту **«ля»**, то отличия в звучании будет заметно на слух, хотя тон звука во всех трех случаях будет одинаковый.

Помимо тона основной частоты, в голосе всегда присутствуют так называемые гармоники. Гармоники представляют собой звуки других частот, отличных от основной частоты. В общем случае любой звук можно представить в виде некоторого бесконечного набора абсолютно чистых звуков различных частот. Совокупность частот таких чистых звуков называется спектром звука.

Таким образом, практически в любом звуке помимо основной частоты присутствуют и другие частоты спектра, называемые гармоническими составляющими, или просто гармониками. Они обогащают звучание произносимой речи. От процентного соотношения гармонических составляющих зависит окраска звука.

Формантные частоты. Голосовые органы человека добавляют к основному тону, формируемому голосовыми связками, дополнительные гармонические составляющие. Эти составляющие придают окраску голоса, по которой сможете узнавать речь знакомых людей.

В результате исследований было установлено, что в образовании речи активно участвуют четыре частоты, образующиеся в резонансных полостях речевого тракта. Эти частоты называются формантами**.** В процессе артикуляции происходит постоянное изменение амплитуды формантных частот, которое можно обнаружить при помощи программ спектрального анализа. Такие программы позволяют развернуть спектр сигнала во времени, отображая его в трехмерном виде.

На рис.1.3 четко виден формантный состав гласных ***и*** и ***у*** при произнесении последовательности этих звуков. При переходе от гласной **и** происходит смещение частоты форманты F2 с 2400 Гц на 784 Гц, а также одновременное ослабление формант F3 и F4.

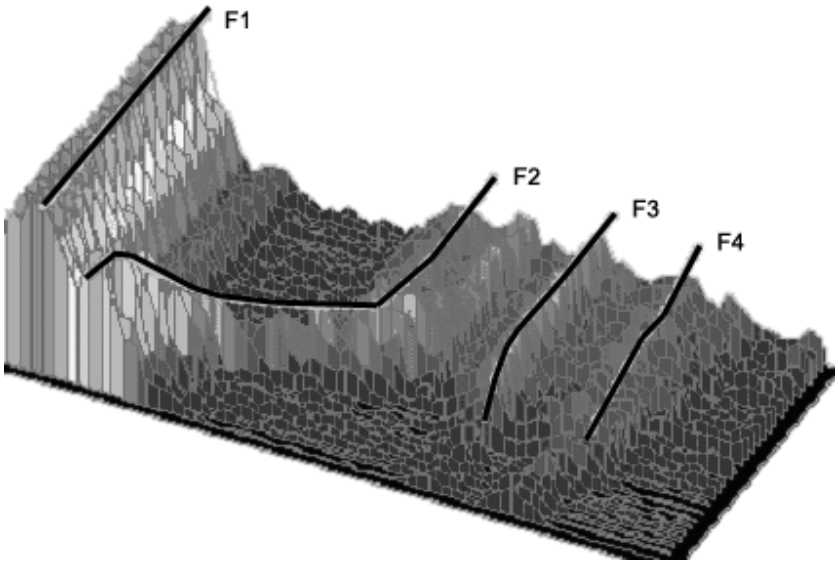


Рис. 1.3. Формантный состав гласных ***и*** и ***у.***

Как видим, в процессе артикуляции может изменяться как амплитуда, так и частота форматных составляющих звука. При этом, однако, количество самих формант в голосовых звуках остается постоянным и всегда равно 4.

Что же касается шумовых звуков, то в них затруднительно выделить формантные составляющие. Это видно на рис.4, где приведен спектр звука, представляющего собой турбулентный шум.

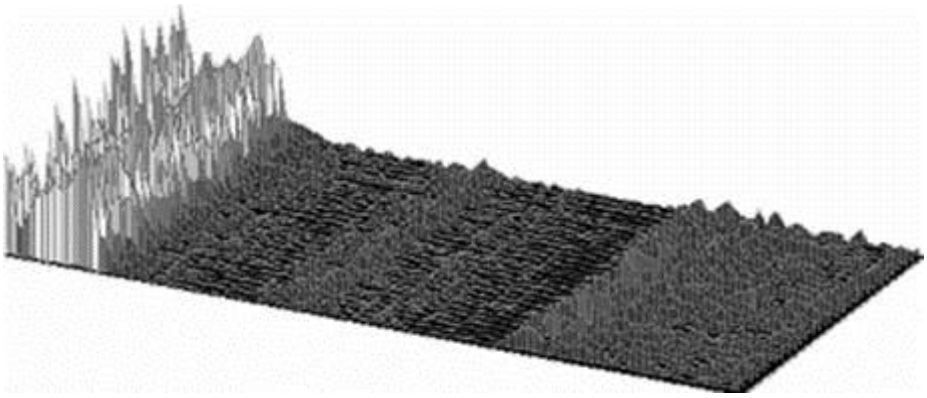


Рис. 1.4. Спектр звука **x**

Современные системы распознавания речи выполняют спектральный анализ, который позволяет выделить из звуковых сигналов речи наиболее информативные составляющие. Это формантные частоты, а также шум. Помимо спектрального анализа используются и более совершенные методы, такие, например, как вейвлет-преобразования.

1.2. Иерархия лингвистических уровней

Все лингвистические понятия, рассмотренные далее в этой главе, специалисты в области языкознания относят к нескольким уровням. Иерархическая структура этих уровней показана на рис.1.5.

Фонетический уровень.

На фонетическом уровне определяются такие понятия, как фонемы и аллофоны. Это кирпичики, из которых строятся все лингвистические элементы более высокого уровня.

Фонемы - минимальная смыслоразличительная единица речи. С точки зрения человека, наименьшей смысловой единицей является слово. Слово делится на слоги, слоги состоят из составляющих единиц звукового

строя — фонем. Они не отождествляются напрямую со словами или слогами, они играют роль неделимых частиц, атомов языка и представляют собой последовательности звуков. Из фонем составляются все другие конструкции языка, такие как слоги и слова.

Рис.1.5. Иерархия лингвистических уровней

Фонемы обладают многочисленными признаками, которые можно использовать для их классификации и распознавания. В качестве примера можно привести следующие признаки:

- звонкость и глухость;
- твердость и мягкость;
- взрывность и фрикативность;
- отсутствие или присутствие назальности;
- переднеязычность и заднеязычности.

Такой признак, как звонкость, проявляется в звуке ***д,*** когда он входит в состав слова **дом**. В противовес этому, звук ***т*** в слове **том** проявляет

глухость. Аналогично, наблюдается твердость звука ***д*** в слове **дома** и мягкость того же самого звука ***д*** в слове **Дёма**. Признак взрывности имеет звук ***д*** в слове **дал**, а признак фрикативности — звук **з** в слове **зал.** В слове **дам** наблюдается отсутствие назальности ***д,*** но в слове **нам** присутствует назальность звука ***н***. В слове **дам** звук ***д*** является переднеязычным, а в слове **гам** звук ***г*** — заднеязычным.

Те признаки, изменение которых приводит к изменению фонемы, называются фонологически существенными признаками фонем.

Аллофоны. Ситуация с многообразием признаков фонем усложняется еще одним обстоятельством — одни и те же фонемы могут изменяться.

Например, гласная буква ***о*** произносится по-разному в словах **вода** и **водяной**. Вместе с тем, эти гласные являются представителями одной и той же фонемы ***о,*** так как они занимают одно и то же положение в звуковой структуре корневой морфемы **вод** и чередуются друг с другом в силу действующих в современном русском языке фонетических закономерностей.

Такие различные реализации фонем называются ***аллофонами***. При этом один из аллофонов, в котором свойства фонемы проявляются в наибольшей степени, играет роль главного варианта фонемы. Всего в русском языке насчитывается 43 фонемы (37 согласных и 6 гласных). В английском языке 43 фонемы - 20 гласных и 24 согласных. В узбекском языке 28 фонемы - 6 гласных (а, о, и, у, э,ў.) и 22 согласных (б, в, г, д, з, й, к, л, м, н, п, р, с, т, ф, х, ц, ч, ш, қ, ғ, ҳ.). К этому добавляются многочисленные аллофоны.

Фонологический уровень.

На фонологическом уровне определяются комбинации фонем и аллофонов, реально встречающихся в человеческой речи. При этом учитывается, что различные комбинации фонем и аллофонов могут встречаться в речи с различной вероятностью.

Морфологический уровень.

На морфологическом уровне накладываются ограничения на структуру таких лингвистических элементов, как слоги и морфемы. Элементы состоят из фонем и аллофонов. При формировании речи добавляются различные фонетические процессы.

Однако сами по себе фонемы не несут никакой смысловой нагрузки. Это просто отдельные **звуки речи**, и ничего больше. Поэтому если система распознавания выделила из речи отдельные фонемы, она должна еще суметь составить из нее слова и предложения. А это непросто, особенно в случае слитной речи, наиболее удобной для человека.

Слоги. Слог - это минимальная фонетическая единица речевого потока, включающая в свой состав, как правило, один гласный звук с примыкающими к нему согласными звуками. Слоги бывают открытыми, закрытыми, условно закрытыми, прикрытыми и неприкрытыми.

Открытые слоги заканчиваются на гласный звук, а закрытые — на согласный. При этом закрытые слоги нельзя открыть, например, **рубль, морс**. Условно-закрытые слоги можно открыть, изменяя слово, например, **кот-коты, клоп-клопы**. Прикрытые слоги начинаются с согласного звука, например, **до-ма, мо-ло-ко.** Неприкрытые слоги начинаются с гласного звука: **о-ло-во, а-ре-на.**

Сами по себе слоги могут нести, а могут и не нести смысловую нагрузку. Например, такой закрытый слов как **рубль**, несет вполне определенную смысловую нагрузку. Что же касается прикрытых слогов **мо, ло** и **ко**, то сами по себе они никакой смысловой нагрузки не несут. Из этого следует, что системе распознавания недостаточно выделить из речи отдельные слоги.

Морфемы. Определению морфемы учили еще в школе. Согласно школьному определению, морфемой называется наименьшая значимая часть слова. Например, в слове **кусочный** можно выделить морфему **кусоч.**

В русском языке морфемы могут совпадать со слогами, а может быть и так, что морфема состоит из нескольких слогов (как, например, в морфеме **ку-соч**, состоящей из двух слогов). Однако есть языки, в которых слоги всегда совпадают с морфемами.

Сами по себе морфемы не могут образовывать предложения, но с их помощью создаются более крупные лингвистические единицы — лексемы.

Лексический уровень.

Лексемой называется множество словоформ с общим лексическим значением. Лексема способна выступать членом предложения и образовывать предложения. Она может быть простой и составной. В первом случае лексема состоит из одного слова, а во втором — из нескольких, например, железная дорога, дом отдыха.

На лексическом уровне определяются слова и словоформы, которые возможны для данного языка. Комбинируя между собой морфемы и слоги, можно образовать бесконечное количество словоподобных конструкций, но в каждом конкретном языке далеко не все они будут наполнены каким либо значением. Слова могут играть роль лексем, но не все слова являются лексемами. Служебные слова, такие, например, как **из** и **бы** не являются лексемами.

Системы распознавания речи могут пользоваться словарями лексем. С помощью этих словарей можно сделать процесс распознавания лексем надежнее, исключая заведомо ложные комбинации, не несущие смысловой нагрузки и появившиеся в результате ошибки механизма распознавания.

Семантический уровень.

Высшим уровнем языка является ***семантика***. Именно на этом уровне человеческий мозг отображает речевые конструкции на понятия и образы, устанавливая отношения между объектами и обозначающими их словами. Наличие стройной системы семантических связей необходимо для создания систем распознавания речи. Только с ее помощью можно получить на выходе этой системы не простой набор слов, извлеченных из потока речи, а

осмысленный набор понятий и отношений между ними, встречающихся в реальной жизни.

Предложения. На семантическом уровне определяется такое понятие, как «предложение»**.** Согласно определению, предложение - это грамматически оформленная по законам данного языка целостная единица речи, являющаяся главным средством формирования, выражения и сообщения мысли.

Но предложение - это не просто осмысленный набор слов и лексем. Предложение может передавать отношение говорящего человека к озвучиваемой мысли. Предложение может иметь особую интонацию, а также заключать в себе предикативность, то есть отношение сообщения к действительности, независимо от того, имеется в этом сообщении глагол или нет. Только такие системы распознавания, которые способны выделять из речи предложения, можно считать достаточно совершенными. Ибо главным образом, именно в виде предложений человек формулирует свои мысли.

Эмоции. К семантическому уровню можно отнести не только словесное представление речи, но и эмоции, выражаемые человеком во время ее произнесения при помощи различных звуков и жестов.

Жесты, сопровождающие речь, могут дополнять и менять смысл произнесенных слов, а также придавать им совершенно другой смысл. Поэтому даже если компьютер будет идеально распознавать слова и составлять из них предложения, в некоторых случаях этого окажется недостаточно для полного понимания сказанного.

1.3. Фонетические процессы

Когда человек говорит, то произносимые слова составляется из фонем и аллофонов. Однако не все так просто. В процессе образования речи происходят различные фонетические процессы, усложняющие общую картину.

Выделены несколько таких процессов:

- аккомодация;
- ассимиляция;
- диссимиляция;
- эпентезы;
- протезы;
- диерезы;
- фонетические чередования;
- традиционные чередования.

Учет этих процессов необходим для качественного синтеза речи. В противном случае мы получим «машинный» голос, напоминающий голос роботов из старых фильмов. Для качественного компьютерного синтеза речи необходимо учитывать и другие процессы, например, изменение тона речи, выделение слов паузами и другие процессы.

Аккомодация.

Аккомодация (приспособление) возникает между согласными и гласными звуками, стоящими рядом. Аккомодация может приводить к появлению дополнительных звуков (так называемых ***глайдов*)**. Например, в произношении слова **воля,** можно расслышать очень короткий звук ***у*** между звуками ***в*** и **о.**

Ассимиляция.

В процессе ***ассимиляции*** происходит артикуляционное и акустическое сближение звуков — согласных с согласными, гласных с гласными. Например, слово **отдать** произносится как [**аддать**], в результате чего последующий звук ***д*** уподобляет предшествующий звук ***т***, создавая ассимиляцию.

Диссимиляция.

Диссимиляция представляет собой процесс, обратный ассимиляции. При взаимодействии согласных звуков с согласными, а также гласных с гласными эти звуки могут расподобляться. Например, в разговорной речи

слово **трамвай** произносится как [**транвай**]. Здесь происходит диссимиляция — два губно-губных звука ***м*** и ***в*** расподобляются, образуя переднеязычный звук ***н*** и губно-губной звук ***в*.**

Эпентезы.

Процессы с названием ***эпентезы*** (вставки) имеют диссимилятивную основу. В результате этого процесса происходит вставка звуков ***в*** или ***й*** между гласными. Например, слово **радио** произносится как [**радиво**], слово **скорпион** — как [**скорпиён**], а слово **какао** — как [**какаво**]. Иногда происходит вставка очень короткого звука между двумя согласными, например, слово **нрав** может произносится как [**ндрав**].

Протезы.

Протезы (надставки) — это разновидность эпентез, но они приставляются спереди к началу слова. Например, в южнорусских диалектах слово **шла** произносится как [**ишла].** Здесь приставляемый звук ***и*** позволяет разгрузить группу начальных согласных. Другой пример — произнесение слова **это** как [**ето]**.

Диерезы.

Диерезы (выкидки) могут иметь ассимилятивную или диссимилятивную основу. В первом случае устраняются звуки между гласными, а во втором — выкидывается один из двух одинаковых или подобных слогов. Например, слово **честный** произносится как **[чесный**], а **минералология** — как [**минералогия**].

Фонетические чередования.

Фонетическими чередованиями называются изменения звуков в потоке речи, вызванные фонетическими процессами современного языка. Например, в словах **воды-вода-водовоз** (читается как [**вады-вада-вадавоз**]) чередуются ударные и безударные гласные, образуя различные варианты фонемы о. В словах **друг-друга** происходит чередование звонких и глухих согласных звуков. Эти слова читаются как [**друк-друга],** при этом фонема **к** является вариантом фонемы **г.**

Традиционные чередования.

Традиционные чередования не обусловлены фонетической позицией, а складываются исторически. Они не имеют ни смысловой, ни фонетической причины появления, а сохраняются лишь в силу традиции. Например, чередования **сон-сна, пень-пня, простой-упрощение, брюзга-брюзжать, запоздать-позже.**

1.4. Интонация, ударные слоги, дифонная модель и частотный диапазон звуковых колебаний

Интонация. Человек может менять высоту основного тона голоса, растягивая голосовые связки. Кроме этого, в широких пределах может изменяться громкость речи и ее темп. Набор этих характеристик называется интонацией речи.

Изменение интонации используется для выделения отдельных слов в предложении, для создания вопросительных предложений. В зависимости от того, каким именно образом меняется интонация, смысл одного и того же предложения может полностью измениться.

Вспомните задачку из детского мультфильма, где ученику предлагалось правильно поставить запятую в предложении «Казнить нельзя, помиловать». Здесь возможны два исключающих друг друга варианта: «Казнить, нельзя помиловать» и «Казнить нельзя, помиловать».

Проговаривая это предложение, паузу в том месте, где находится запятая. В первом варианте при помощи интонации будет выделено слово **казнить**, а во втором - **помиловать**, что и придает предложению противоположный смысл.

Многие компьютерные системы синтеза речи, преобразующие текстовые файлы в речь, не в состоянии корректно изменять интонацию речи (просто потому, что они не понимают смысла произносимого). Это может

привести к тому, что синтезированная речь будет звучать монотонно, а смысл произносимых предложений окажется искажен.

Проблема правильной расстановки интонационных ударений не так проста, как может показаться на первый взгляд. Для выделения нужных слов компьютерная программа должна понимать смысл текста, так как даже знаков препинания может оказаться недостаточно, чтобы изменять интонацию слов надлежащим образом.

Ударные слоги.

Словесное ударение — это выделение одного или двух слогов в составе многосложного слова с помощью интонации. При ударении меняется сила, высота и длительность звуков. Ударение связывает звучание слова в единое целое, отделяя при этом одно слово от другого. При этом в различают динамическое, музыкальное и количественное ударение.

В результате динамического ударения происходит усиление звучания. Музыкальное ударение связано с изменением тона, а количественное — с изменением продолжительности звучания.

Динамическое ударение может привести к редукции, т.е. к ослаблению и изменению звучания безударных слогов.

Количественная редукция приводит к потере долготы и силы звучания, а качественная редукция приводит дополнительно к изменению тембра голоса (т.е. звуковой окраски голоса).

Дифонная модель. Одна из проблем, с которой сталкиваются разработчики систем распознавания речи, — выделение из слитного потока элементарных лингвистических единиц, таких как фонемы и аллофоны. Исследователи пытаются использовать различные модели, с помощью которых можно было бы выполнить такое выделение.

Фонемная модель — только одна из них. Другая модель называется дифонной моделью. В рамках этой модели вводится понятие элементарной речевой единицы — дифона. Дифоном называется звуковая единица, протяженная от середины одного звука до середины последующего.

Дифонная модель предполагает, что из речи можно выделить некие стационарные участки, на звучание которых не влияют соседние звуки. В середине этих стационарных участков проводится граница между дифонами. При этом, однако, общее количество дифонов в том или ином языке будет не меньше, чем общее количество аллофонов в это же языке. Сравнение дифонной модели речи с фонемной моделью получается не в пользу дифонной модели. В дифонной модели отмечается ряд недостатков.

Один из этих недостатков связан с созданием дифонной базы данных. В процессе ее наполнения диктор должен монотонно начитывать речевой материал, намеренно растягивая слова. Это делается для облегчения поиска границ дифонов.

Трудности возникают и при попытках использовать дифонную базу данных для синтеза речи. Если речь формируется посредством соединения дифонов, то в местах соединений образуются заметные перепады формантных частот. Образующиеся в результате спектральные разрывы заметны на слух — речь, «склеенная» из отдельных дифонов, звучит неестественно.

При попытке избавиться от этого недостатка за счет увеличения размеров дифонной базы данных и учета контекста расположения дифонов происходит усложнение алгоритмов формирования речевого сигнала.

Фонемная модель не обладает этим недостатком. Правильный выбор аллофонов позволяет синтезировать речь без заметных на слух разрывов. А для создания базы данных фонем и аллофонов диктор должен читать текст естественным голосом.

Частотный диапазон звуковых колебаний. Человеческое ухо воспринимает звуковые волны длиной примерно от 1,6 см до 20 м, что соответствует частотному диапазону 16 - 20 000 Гц. Животные могут слышать звуки более низкой или более высокой частоты. Так, например, дельфинам и летучим мышам доступно общение при помощи ультразвука, а

китам — инфразвука. Поэтому человек не слышит весь частотный диапазон звуков, издаваемых этими и некоторыми другими животными.

Что же касается человеческой речи, то ее частотный диапазон 300-4000 Гц. Надо заметить, что разборчивость речи останется вполне удовлетворительной при ограничении этого диапазона до 300-2400 Гц. Частотный диапазон обычных телефонных каналов тоже не слишком широкий, однако это не сказывается заметным образом на разборчивость речи.

Сказанное означает, что для улучшения качества распознавания речи компьютерные системы могут исключить из анализа частоты, лежащие вне диапазона 300-4000 Гц или даже вне диапазона 300-2400 Гц.

Контрольные вопросы

1. Каковы схемы речевого тракта?
2. Как и для чего используются формантные частоты?
3. Что означают формантные частоты?
4. Объясните иерархии лингвистических уровней.
5. Что такое фонема?
6. Каковы особенности фонетических процессов?
7. Объясните суть дифонной модели.
8. Каковы особенности фонемной модели?
9. Каковы отличия дифонной модели от фонемной?
10. Каков частотный диапазон звуковых колебаний?

ГЛАВА 2. ЦИФРОВАЯ ОБРАБОТКА РЕЧЕВЫХ СИГНАЛОВ

2.1. Введение в цифровую обработку сигналов

Цифровая обработка сигналов (ЦОС или DSP – digital signal processing) является одной из новейших и самых мощных технологий, которая активно внедрилась в широкий круг областей науки и техники: коммуникации, метеорология, радиолокация и гидролокация, медицинская визуализация изображений, цифровое аудио- и телевизионное вещание, разведка нефтяных и газовых месторождений, и многих других. Цифровая обработка сигналов – это информатика реального времени. Сегодня технология ЦОС относится к числу базовых знаний, которые необходимы ученым и инженерам всех отраслей без исключения.

Физические переменные природы, как основного объекта наших измерений и источника информационных сигналов, как правило, имеют непрерывную природу и отображаются непрерывными (аналоговыми) сигналами. Цифровая обработка сигналов оперирует с дискретными величинами, причем с квантованием как по координатам динамики своих изменений (во времени, в пространстве, и по любым другим изменяемым аргументам), так и по значениям физических величин. Математика дискретных преобразований зародилась в недрах аналоговой математики еще в 18 веке в рамках теории рядов и их применения для интерполяции и аппроксимации функций, однако ускоренное развитие она получила в 20 веке после появления первых вычислительных машин. В своих основных положениях математический аппарат дискретных преобразований подобен преобразованиям аналоговых сигналов и систем.

Стимулом развития дискретной математики является и то, что стоимость цифровой обработки данных меньше аналоговой и продолжает снижаться, а производительность вычислительных операций непрерывно возрастает. Кроме того, системы ЦОС отличаются высокой гибкостью. Их

можно дополнять новыми программами и перепрограммировать на выполнение различных операций без изменения оборудования. В последние годы ЦОС оказывает возрастающее влияние на все отрасли современной промышленности: телекоммуникации, средства информации, цифровое телевидение. Интерес к научным и к прикладным вопросам цифровой обработки сигналов возрастает во всех отраслях науки и техники.

Цифровые сигналы формируются из аналоговых операцией дискретизации – последовательным квантованием (измерением) амплитудных значений сигнала через определенные интервалы времени Δt или любой другой независимой переменной Δx. В результате равномерной дискретизации непрерывный по аргументу сигнал переводится в упорядоченную по независимой переменной последовательность чисел. В принципе разработаны методы ЦОС для неравномерной дискретизации данных, однако области их применения достаточно специфичны и ограничены. Условия, при которых возможно полное восстановление аналогового сигнала по его цифровому эквиваленту с сохранением всей исходно содержавшейся в сигнале информации, выражаются теоремами Найквиста, Котельникова, Шеннона, сущность которых практически одинакова. Для дискретизации аналогового сигнала с полным сохранением информации в его цифровом эквиваленте максимальные частоты в аналоговом сигнале должны быть не менее чем вдвое меньше, чем частота дискретизации, то есть $f_{max} \leq (1/2)f_d$, т.е. на одном периоде максимальной частоты должно быть минимум два отсчета. Если это условие нарушается, в цифровом сигнале возникает эффект маскирования (подмены) действительных частот более низкими частотами. При этом в цифровом сигнале вместо фактической регистрируется "кажущаяся" частота, а, следовательно, восстановление фактической частоты в аналоговом сигнале становится невозможным. Восстановленный сигнал будет выглядеть так, как если бы частоты, лежащие выше половины частоты дискретизации, отразились от частоты $(1/2)f_d$ в нижнюю часть спектра и наложились на

частоты, уже присутствующие в этой части спектра. Этот эффект называется наложением спектров или алиасингом (aliasing). Наглядным примером алиасинга может служить иллюзия, довольно частая в кино – колесо автомобиля начинает вращаться против его движения, если между последовательными кадрами (аналог частоты дискретизации) колесо совершает более чем пол-оборота.

Преобразование сигнала в цифровую форму выполняется аналого-цифровыми преобразователями (АЦП). Как правило, они используют двоичную систему счисления с определенным числом разрядов в равномерной шкале. Увеличение числа разрядов повышает точность измерений и расширяет динамический диапазон измеряемых сигналов. Потерянная из-за недостатка разрядов АЦП информация невосстановима, и существуют лишь оценки возникающей погрешности «округления» отсчетов, например, через мощность шума, порождаемого ошибкой в последнем разряде АЦП. Для этого используется понятие отношения «сигнал/шум» - отношение мощности сигнала к мощности шума (в децибелах). Наиболее часто применяются 8-, 10-, 12-, 16-, 20- и 24-х разрядные АЦП. Каждый дополнительный разряд улучшает отношение сигнал/шум на 6 децибел. Однако увеличение количества разрядов снижает скорость дискретизации и увеличивает стоимость аппаратуры. Важным аспектом является также динамический диапазон, определяемый максимальным и минимальным значением сигнала.

Обработка цифровых сигналов выполняется либо специальными процессорами, либо на универсальных компьютерах по специальным программам. Наиболее просты для рассмотрения линейные системы. Линейными называются системы, для которых имеет место суперпозиция (отклик на сумму входных сигналов равен сумме откликов на каждый сигнал в отдельности) и однородность или гомогенность (изменение амплитуды входного сигнала вызывает пропорциональное изменение выходного сигнала). Для реальных объектов свойства линейности могут выполняться

приближенно и в определенном интервале входных сигналов.

Если входной сигнал $x(t-t_0)$ порождает однозначный выходной сигнал $y(t-t_0)$ при любом сдвиге t_0, то систему называют инвариантной во времени. Ее свойства можно исследовать в любые произвольные моменты времени. Для описания линейной системы вводится специальный входной сигнал - единичный импульс (импульсная функция). В силу свойства суперпозиции и однородности любой входной сигнал можно представить в виде суммы таких импульсов, подаваемых в разные моменты времени и умноженных на соответствующие коэффициенты. Выходной сигнал системы в этом случае представляет собой сумму откликов на эти импульсы. Отклик на единичный импульс (импульс с единичной амплитудой) называют импульсной характеристикой системы h(n). Соответственно, отклик системы на произвольный входной сигнал s(k) можно выразить сверткой

$$g(k) = h(n) \circledast s(k-n).$$

Если $h(n)=0$ при $n<0$, то систему называют каузальной (причинной). В такой системе реакция на входной сигнал появляется только после поступления сигнала на ее вход. Некаузальные системы физически невозможно реализовать в реальном масштабе времени. Если требуется реализовать свертку сигналов с двусторонними операторами (при дифференцировании), то это выполняется с задержкой (сдвигом) входного сигнала минимум на длину левосторонней части оператора свертки.

Природа сигналов. По своей природе сигналы могут быть случайными или детерминированными.

К детерминированным относят сигналы, значения которых в любой момент времени или в произвольной точке пространства являются априорно известными или могут быть определены (вычислены) по известной или предполагаемой функции, даже если мы не знаем ее явного вида.

Случайные сигналы непредсказуемы по своим значениям во времени или в пространстве. Для каждого конкретного отсчета случайного сигнала можно знать только вероятность того, что он примет какое-либо значение в

определенной области возможных значений. Закон распределения случайных значений далеко не всегда известен. Одним из самых распространенных является нормальное распределение, плотность которого имеет вид симметричного колокола. Для его описания достаточно двух первых моментов распределения случайных величин.

Наиболее простые характеристики законов распределения – среднее значение случайных величин (математическое ожидание) и дисперсия (математическое ожидание квадрата отклонения от среднего), характеризующая разброс значений случайных величин относительно среднего значения. Параметры динамики случайных сигналов во времени характеризуются функциями автокорреляции (количественная оценка взаимосвязи значений случайного сигнала на различных интервалах) или автоковариации (то же, при центрировании случайных сигналов). Аналогичной мерой взаимосвязи двух случайных процессов и степени их сходства по динамике развития является кросскорреляция или кроссковариация (взаимная корреляция или ковариация). Максимальное значение взаимной корреляции достигается при совпадении двух сигналов. При задержке одного из сигналов по отношению к другому положение максимума корреляционной функции дает возможность оценить величину этой задержки.

Функциональные преобразования сигналов. Одним из основных методов частотного анализа и обработки сигналов является преобразование Фурье. Различают понятия “преобразование Фурье” и “ряд Фурье”. Преобразование Фурье предполагает непрерывное распределение частот, ряд Фурье задается на дискретном наборе частот. Сигналы также могут быть заданы в наборе временных отсчетов или как непрерывная функция времени. Наиболее практична с точки зрения цифровой обработки сигналов дискретизация и во временной, и в частотной области, но не следует забывать, что она является аппроксимацией непрерывного преобразования. Непрерывное преобразование Фурье позволяет точно представлять любые

явления. Сигнал, представленный рядом Фурье, может быть только периодичен. Сигналы произвольной формы могут быть представлены рядом Фурье только приближенно, т.к. при этом предполагается периодическое повторение рассматриваемого интервала сигнала за пределами его задания. На стыках периодов при этом могут возникать разрывы и изломы сигнала, и возникать ошибки обработки, вызванные явлением Гиббса, для минимизации которых применяют определенные методы (весовые окна, продление интервалов задания сигналов).

При дискретизации и во временной, и в частотной области, обычно говорят о дискретном преобразовании Фурье (ДПФ):

$$S(n) = \sum_k s(k) \exp(-j2\pi\, kn/N),$$

где s(k) – отсчеты входного сигнала,

N- количество отсчетов сигнала,

exp(-j2π kn/N) - это система (матрица) базисных функций в виде синусов и косинусов.

ДПФ применяется для вычисления спектров мощности, оценивания передаточных функций и импульсных откликов, быстрого вычисления сверток при фильтрации, расчете корреляции, расчете преобразований Гильберта. Расчет ДПФ по приведенной формуле требует вычисления **n** коэффициентов, каждый из которых зависит от **k** элементов исходного отрезка, так что число операций не может быть меньше **nk**. Существуют алгоритмы быстрого преобразования Фурье - БПФ, сокращающее число операций для вычисления коэффициентов до **nlog(k).** При точной арифметике результаты расчетов ДПФ и по алгоритмам БПФ совпадают.

Находят применение и другие варианты преобразования Фурье, в которых идея умножения вектора отсчетов сигнала на матрицу базисных функций сохраняется, однако сами базисные функции имеют или переключательный или линейно-ступенчатый характер. В последнее время в

задачах спектрально-временнного анализа нестационарных сигналов, изучении нестационарностей и локальных особенностей сигналов широко применяются так называемые "короткие волны" (вейвлеты), локализованные как во временной, так и в частотной области.

Традиционные методы анализа данных предназначены, как правило, для линейных и стационарных сигналов и систем, и только в последние десятилетия начали активно развиваться методы анализа нелинейных, но стационарных и детерминированных систем, и линейных, но нестационарных данных. Между тем, большинство естественных материальных процессов, реальных физических систем и соответствующих этим процессам и системам данных в той или иной мере являются нелинейными и нестационарными, и при анализе данных используются определенные упрощения, особенно в отношении априорно устанавливаемого базиса разложения данных.

Ключевые операции цифровой обработки сигналов.

Алгоритмы ЦОС реализуются как в классической временной области (телекоммуникации, связь, телевидение, радиолокация), так и в спектральной области в самых различных отраслях науки и техники (геоинформатике, геологии и геофизике, медицине, биологии, военном деле). Все эти алгоритмы, как правило – блочного типа, построенные на сколь угодно сложных комбинациях достаточно небольшого набора типовых цифровых операций, к основным из которых относятся свертка (конволюция), корреляция, фильтрация, модуляция, спектральный анализ. Далее приводятся только ключевые позиции по этим операциям.

Линейная свертка – основная операция ЦОС, особенно в режиме реального времени. Для двух конечных причинных последовательностей h(n) и y(k) длиной соответственно N и K свертка определяется выражением:

$$s(k) = h(n) \circledast y(k) \equiv h(n) * y(k) = \sum_{n=0}^{N} h(n)\, y(k-n), \qquad (2.1)$$

где: $\circledast$ или $*$ - символьные обозначения операции свертки.

Как правило, в системах обработки одна из последовательностей y(k)

представляет собой обрабатываемые данные (сигнал на входе системы), вторая h(n) – оператор (импульсный отклик) системы, а функция s(k) – выходной сигнал системы. В компьютерных системах с памятью для входных данных оператор h(n) может быть двусторонним от $-N_1$ до $+N_2$, например – симметричным h(-n) = h(n), с соответствующим изменением пределов суммирования в (2.1), что позволяет получать выходные данные без сдвига относительно входных.

При строго корректной свертке с обработкой всех отсчетов входных данных, размер выходного массива равен $K+N_1+N_2-1$, должны задаваться начальные условия по отсчетам y(k) для значений y(0-n) до $n=N_2$, и конечные для y(K+n) до $n=N_1$. Пример (Mathcad) выполнения свертки приведен на рис. 2.1.

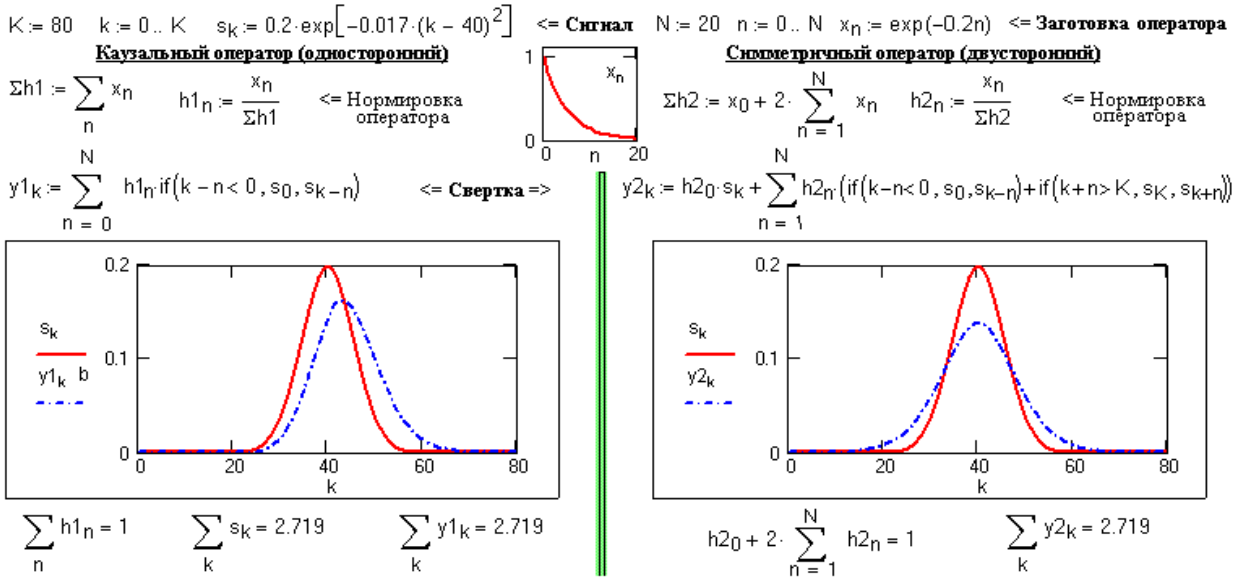


Рис. 2.1. Примеры дискретной свертки

Преобразование свертки однозначно определяет выходной сигнал для установленных значений входного сигнала при известном импульсном отклике системы. Обратная задача деконволюции - определение функции y(k) по функциям s(k) и h(n), имеет решение только при определенных условиях. Это объясняется тем, что свертка может существенно изменить

частотный спектр сигнала s(k) и восстановление функции y(k) становится невозможным, если определенные частоты ее спектра в сигнале s(k) полностью утрачены.

Корреляция существует в двух формах: автокорреляции и взаимной корреляции.

Взаимно-корреляционная функция (ВКФ, cross-correlation function - CCF), и ее частный случай для центрированных сигналов функция взаимной ковариации (ФВК) – это показатель степени сходства формы и свойств двух сигналов. Для двух последовательностей x(k) и y(k) длиной K с нулевыми средними значениями оценка взаимной ковариации выполняется по формулам:

$$K_{xy}(n) = (1/(K-n+1)) \sum_{k=0}^{K-n} x(k)\, y(k+n), \quad n = 0, 1, 2, \ldots$$
$$K_{xy}(n) = (1/(K-n+1)) \sum_{k=0}^{K-n} x(k-n)\, y(k), \quad n = 0, -1, -2, \ldots \qquad (2.2)$$

Пример определения сдвига между двумя детерминированными сигналами, представленными радиоимпульсами, по максимуму ФВК приведен на рис. 2.2. По максимуму ФВК может определяться и сдвиг между сигналами, достаточно различными по форме.

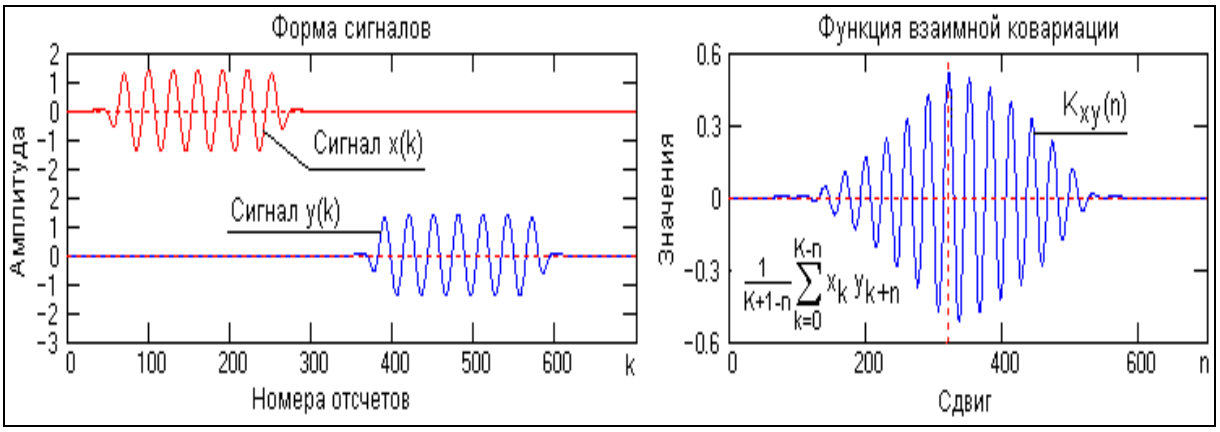


Рис. 2.2. Функция взаимной ковариации двух детерминированных сигналов.

На рис. 2.3 приведен аналогичный пример ФВК двух одинаковых по форме сигналов, на один из которых наложен шумовой сигнал. Мощность шума превышает мощность сигнала. Вычисление ФВК на рисунке выполнено в двух вариантах. Вариант 1 полностью соответствует формуле (2.2). Но в условиях присутствия в сигналах достаточно мощных шумов вычисление ФВК обычно выполняется по варианту 2 – с постоянным нормировочным множителем. Это определяется тем, что по мере увеличения сдвига **n** и уменьшения количества суммируемых членов в формуле (2.2) за счет шумовых сигналов существенно нарастает ошибка оценки ФВК, которая к тому же увеличивается за счет нелинейного увеличения значения нормировочного множителя, особенно при малом количестве отсчетов. Сохранение множителя постоянным в какой-то мере компенсирует этот эффект.

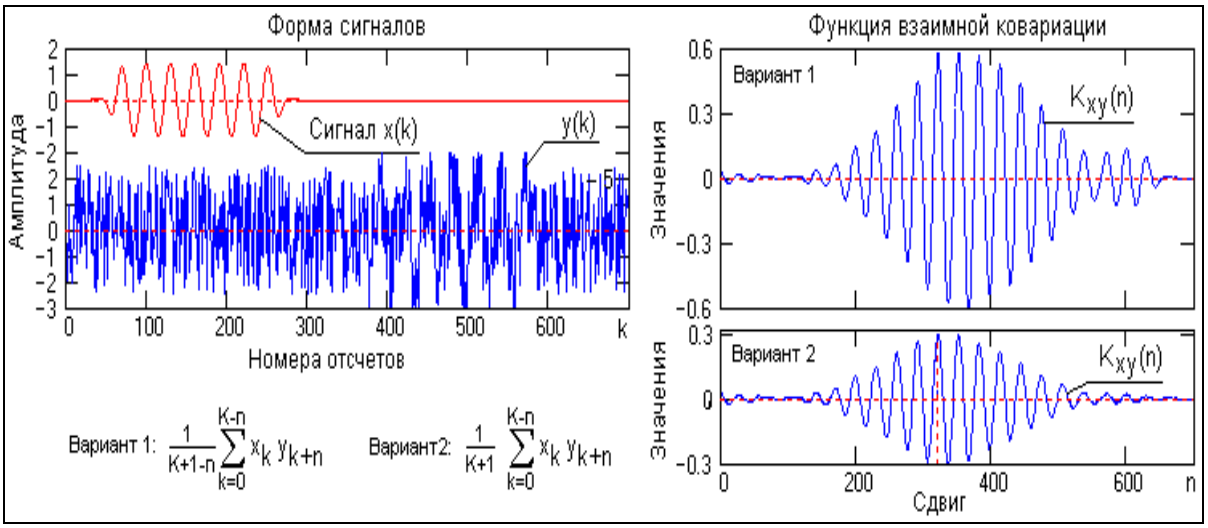


Рис. 2.3. ФВК двух сигналов, один из которых сильно зашумлен.

На рис. 2.4 приведен пример вычисления функции взаимной ковариации двух одинаковых сигналов, скрытых в шумах. ФВК позволяет не только определить величину сдвига между сигналами, но и уверенно оценить период колебаний в исследуемых радиоимпульсах.

Относительный количественный показатель степени сходства двух сигналов x(k) и y(k) - функция взаимных корреляционных коэффициентов

$\rho_{xy}(n)$. Она вычисляется через центрированные значения сигналов (для вычисления взаимной ковариации нецентрированных сигналов достаточно центрировать один из них), и нормируется на произведение значений стандартов (средних квадратических вариаций) функций x(k) и y(k):

$$\rho_{xy}(n) = K_{xy}(n)/(\sigma_x \sigma_y). \qquad (2.3)$$

$$\sigma_x^2 = K_{xx}(0) = (1/(K+1)) \sum_{k=0}^{K} (x(k))^2,$$

$$\sigma_y^2 = K_{yy}(0) = (1/(K+1)) \sum_{k=0}^{K} (y(k))^2. \qquad (2.4)$$

Интервал изменения значений корреляционных коэффициентов при сдвигах n может изменяться от –1 (полная обратная корреляция) до +1 (полное сходство или стопроцентная корреляция). При сдвигах **n**, на которых наблюдаются нулевые значения $r_{xy}(n)$, сигналы некоррелированны. Коэффициент взаимной корреляции позволяет устанавливать наличие определенной связи между сигналами вне зависимости от физических свойств сигналов и их величины.

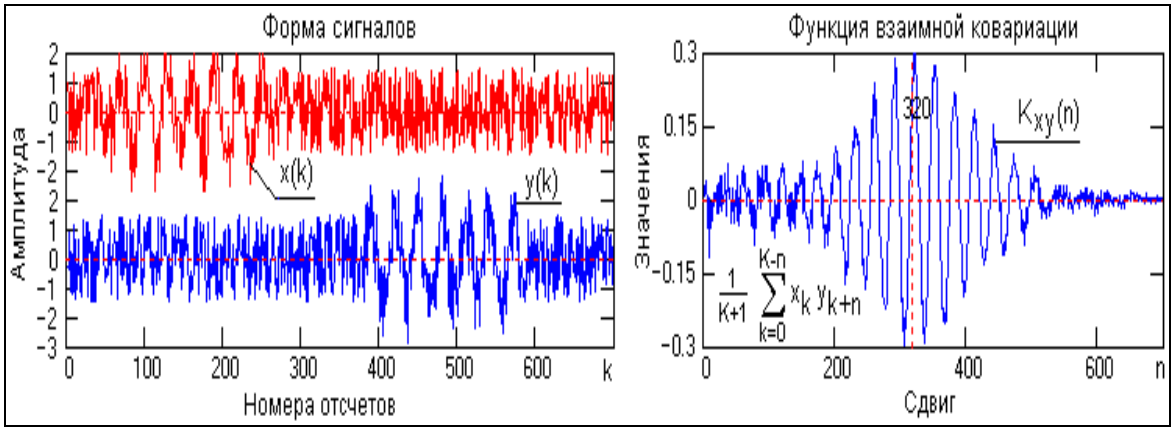


Рис. 2.4. ФВК двух зашумленных радиоимпульсов.

В технической литературе в терминах "корреляция" и "ковариация" в настоящее время существуют накладки. Корреляционными функциями называют как функции по нецентрированным, так и по центрированным сигналам, а также и функцию взаимных корреляционных коэффициентов.

Автокорреляционная функция (АКФ, correlation function, CF) является

количественной интегральной характеристикой формы сигнала, дает ρинформацию о структуре сигнала и его динамике во времени. Она, по существу, является частным случаем ВКФ для одного сигнала и представляет собой скалярное произведение сигнала и его копии в функциональной зависимости от переменной величины значения сдвига:

$$B_x(n) = (1/(K-n+1)) \sum_{k=0}^{K-n} x(k)\, x(k+n), \quad n = 0, 1, 2, \ldots \qquad (2.5)$$

АКФ имеет максимальное значение при n=0 (умножение сигнала на самого себя), является четной функцией $B_{xy}(-n)=B_{xy}(n)$, и значения АКФ для отрицательных координат обычно не вычисляются. АКФ центрированного сигнала $K_x(n)$ представляет собой функцию автоковариации (ФАК). ФАК, нормированная на свое значение $K_x(0)=\sigma_x^2$ в n=0:

$$\rho_x(n) = K_x(n)/K_x(0) \qquad (2.6)$$

называется функцией автокорреляционных коэффициентов.

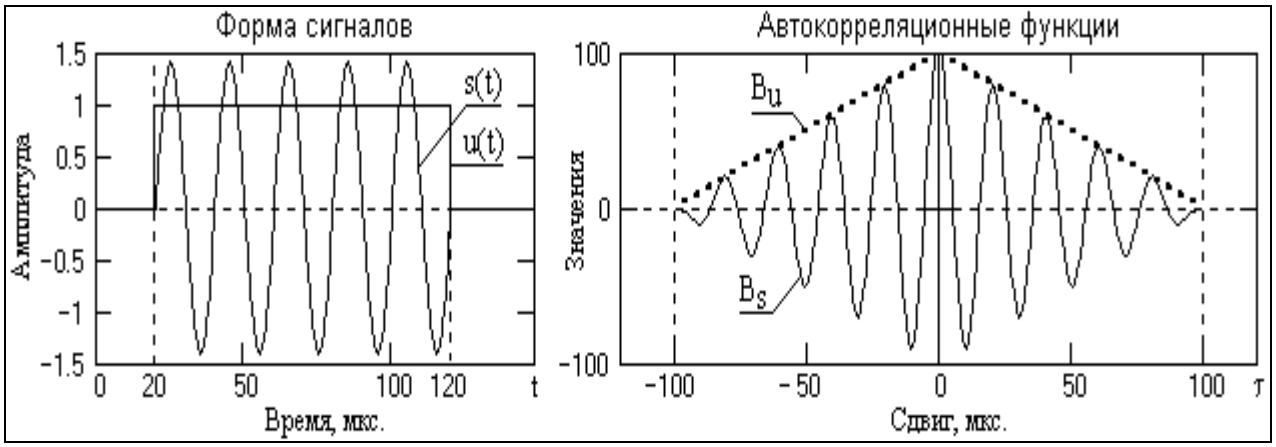


Рис. 2.5. Автокорреляционные функции.

В качестве примера на рис. 2.5 приведены два сигнала – прямоугольный импульс и радиоимпульс одинаковой длительности T, и соответствующие данным сигналам формы их АКФ. Амплитуда колебаний радиоимпульса установлена равной $\sqrt{T}$ амплитуды прямоугольного импульса, при этом энергии сигналов будут одинаковыми, что подтверждается равными значениями максимумов АКФ. При конечной

длительности импульсов длительности АКФ также конечны, и равны удвоенным значениям длительности импульсов (при сдвиге копии конечного импульса на интервал его длительности как влево, так и вправо, произведение импульса со своей копией становится равным нулю). Частота колебаний АКФ радиоимпульса равна частоте колебаний заполнения радиоимпульса (боковые минимумы и максимумы АКФ возникают каждый раз при последовательных сдвигах копии радиоимпульса на половину периода колебаний его заполнения).

Линейная цифровая фильтрация является одной из операций ЦОС, имеющих первостепенное значение, и определяется как

$$s(k) = \sum_{n=0}^{N} h(n)\, y(k-n), \qquad (2.7)$$

где: h(n), n=0, 1, 2, … , N – коэффициенты фильтра, y(k) и s(k) – вход и выход фильтра. Это по сути свертка сигнала с импульсной характеристикой фильтра.

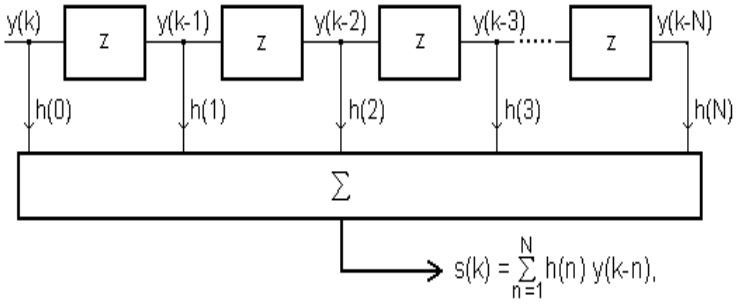


Рис.2.6. Трансверсальный цифровой фильтр.

На рис. 2.6 показана блок-схема фильтра, который в таком виде широко известен, как трансверсальный (z – задержка на один интервал дискретизации).

К основным операциям фильтрации информации относят операции сглаживания, прогнозирования, дифференцирования, интегрирования и разделения сигналов, а также выделение информационных (полезных) сигналов и подавление шумов (помех). Основными методами цифровой

фильтрации данных являются частотная селекция сигналов и оптимальная (адаптивная) фильтрация.

Дискретные преобразования позволяют описывать сигналы с дискретным временем в частотных координатах или переходить от описания во временной области к описанию в частотной. Переход от временных (пространственных) координат к частотным необходим во многих приложениях обработки данных.

Самым распространенным является дискретное преобразование Фурье, описанное ранее. Как отмечалось ранее прямое преобразование равноценно обратному, это условия информационной равноценности динамической и частотной форм представления дискретных сигналов. Другими словами для преобразований без потерь информации число отсчетов функции и ее спектра должны быть одинаковыми.

В принципе, согласно общей теории информации, последнее заключение действительно и для любых других видов линейных дискретных преобразований.

Модуляция сигналов. Системы регистрации, обработки, интерпретации, хранения и использования информационных данных становятся все более распределенными, что требует коммуникации данных по высокочастотным каналам связи. Как правило, информационные сигналы являются низкочастотными и ограниченными по ширине спектра, в отличие от широкополосных высокочастотных каналов связи, рассчитанных на передачу сигналов от множества источников одновременно с частотным разделением каналов. Перенос спектра сигналов из низкочастотной области в выделенную для их передачи область высоких частот выполняется операцией модуляции. При модуляции значения информационного (модулирующего) сигнала переносятся на определенный параметр высокочастотного (несущего) сигнала.

Самые распространенные схемы модуляции для передачи цифровой информации по широкополосным каналам – это амплитудная (amplitude shift

keying – ASK), фазовая (phase shift keying – PSK) и частотная (frequensy shift keying – FSK) манипуляции. При передаче данных по цифровым сетям используется также импульсно-кодовая модуляция (pulsecodemodulation – PCM).

Области применения цифровой обработки.

Цифровая обработка сигналов применяется в самых различных областях науки и техники. Поэтому коснемся только тех областей, где применение ЦОС развивается наиболее быстрыми темпами.

Процессоры ЦОС. Обработка данных в реальном времени обычно выполняется на специальных процессорах (чипах) ЦОС. Они, как правило, имеют следующие средства.

- Встроенные умножители или умножители-накопители, работающие параллельно.
- Отдельные шины и области памяти для программ и данных.
- Команды организации циклов.
- Большие скорости обработки данных и тактовые частоты.
- Использование конвейерных методов обработки данных.
- Применение параллельных методов обработки.

Запись, воспроизведение, использование звука. Цифровое микширование – регулирование и смешивание многоканальных аудиосигналов от различных источников. Это выполняется аудиоэквалайзерами (наборами цифровых полосовых фильтров с регулируемыми характеристиками), смесителями и устройствами создания специальных эффектов (реверберация, динамическое выравнивание и пр.).

Синтезаторы речи - представляют собой достаточно сложные устройства генерации голосовых звуков. Микросхемы синтезаторов вместе с процессорами обычно содержат в ПЗУ словари слов и фраз в форме кадров (25 мс речи) с внешним управлением интонацией, акцентом и диалектом, что позволяет на высоком уровне имитировать человеческую речь.

Распознавание речи – эта область активно изучается и развивается, особенно для целей речевого ввода информации в компьютеры. Как правило, в режиме обучения выполняется их настройка на речь пользователя, в процессе которой система оцифровывает и создает в памяти эталоны слов. В режиме распознавания речь также оцифровывается и сравнивается с эталонами в памяти. Системы распознавания речи внедряются и в товары бытового назначения (набор телефонных номеров, включение/выключение телевизора), в управление промышленным оборудованием, в робототехнику.

Аудиосистемы воспроизведения компакт-дисков. При плотности записи выше 10^6 бит на мм^2 обеспечивают очень высокую плотность хранения информации. Аналоговый звуковой сигнал в стереоканалах дискретизируется с частотой 44.1 кГц и оцифровывается 16-битным кодом. При записи на диск сигналы модулируются (EFM – преобразование 8-ми разрядного кода в 14-ти разрядный для надежности), при считывании сигналы демодулируются, исправляются ошибки (по возможности) и выполняется цифро-аналоговое преобразование.

Применение ЦОС в телекоммуникациях. Цифровая сотовая телефонная сеть – двусторонняя телефонная система с мобильными телефонами через радиоканалы и связью через базовые радиостанции. Мировым стандартом цифровой мобильной связи является система GSM. Частотный диапазон связи 890-960 МГц, частотный интервал канала 200 кГц, скорость передачи информации 270 кбит/с. В мобильной связи ЦОС используется для кодирования речи, выравнивания сигналов после многолучевого распространения, измерения силы и качества сигналов, кодирования с исправлением ошибок, модуляции и демодуляции.

Цифровое телевидение дает потребителям интерактивность, большой выбор, лучшее качество изображения и звука, доступ в Интернет. ЦОС в цифровом телевидении играет ключевую роль в обработке сигналов, кодировании, модуляции/демодуляции видео- и аудиосигналов от точки захвата до момента появления на экране. ЦОС лежит в основе алгоритмов

кодирования MPEG, которые используются для сжатия сигналов перед их передачей и при декодировании в приемниках.

ЦОС в биомедицине. Основное назначение – усиление сигналов, которые измеряют параметры тела человека и не отличаются хорошим качеством, и/или извлечение из них информации, представляющей определенный интерес, на фоне существенного уровня шумов и многочисленных артефактов (ложных изображений как от внешних, так и от внутренних источников). Так, например, при снятии электрокардиограммы плода регистрируется электрическая активность сердца ребенка на поверхности тела матери, где также существует определенная электрическая активность. Применение ЦОС во многих областях медицины позволяет переходить от чисто качественных показателей к объективным количественным оценкам, как например, в анестезии к оценке глубины анестетического состояния пациента при операции по электрической активности мозга.

2.2. Параметры речевого сигнала

Процесс оцифровки характеризуется двумя важными параметрами, определяющими качество полученного сигнала. Первый параметр называется частотой дискретизации, или частотой сэмплирования (samplerate), и определяет, с какой частотой будет происходить мгновенная регистрация аналогового напряжения. К примеру, если частота дискретизации равна 44,1 кГц, то это означает, что аналоговый сигнал регистрируется (измеряется) 44 100 раз в секунду.

По известным теоремам Котельникова, для точного восстановления сигнала по его отсчетам частота дискретизации должна быть как минимум в два раза выше максимальной звуковой частоты. Так как максимальная частота звукового сигнала, различаемая человеческим ухом, составляет 20 кГц, то частота дискретизации 44,1 кГц полностью соответствует

требованиям преобразования. Тем не менее, производители конвертеров в последнее время неумолимо повышают частоту дискретизации, считая, что это оказывает влияние на общее качество звучания. Например, новый формат DVD предусматривает запись звуковой информации с частотой дискретизации 96 кГц, но и это не предел. Существуют устройства, работающие на 192 кГц и даже выше.

Второй важный параметр, характеризующий процесс оцифровки, - это разрядность, измеряемая в битах. Она связана с точностью измерения мгновенных амплитуд сигнала.

Представление речевых сигналов в цифровой форме.

В основе аналого-цифрового преобразования речи лежит получение мгновенных значений амплитуды речевого сигнала через определенные временные интервалы. Эта операция называется дискретизацией сигнала и показана на рисунке 2.7.

Известно, что любой сигнал можно охарактеризовать его спектром или занимаемой полосой частот. Как говорилось выше, частота дискретизации (взятия отсчетов) должна быть равна или превосходить удвоенное значение полосы, которую занимает сигнал. Человеческое ухо может улавливать сигналы с частотами до 20 кгц. Высокочастотные составляющие придают звуковому сигналу индивидуальность и выразительность и важны, например, при передаче музыкальных композиций. Так в цифровой записи на компакт-дисках используется частота дискретизации 44,1 кГц.

В аналоговой телефонии значимой для передачи речи является полоса 0,3-3,4 кГц., обеспечивающая так называемое телефонное качество. При этом достаточна частота дискретизации 8 кГц. Такая низкая частота дискретизации может служить причиной искажений при восстановлении оцифрованного естественного речевого сигнала, поэтому перед аналого-цифровым преобразованием он ограничивается по полосе с помощью фильтра низких частот (ФНЧ). Процесс получения дискретных значений называется амплитудно-импульсной модуляцией (АИМ).

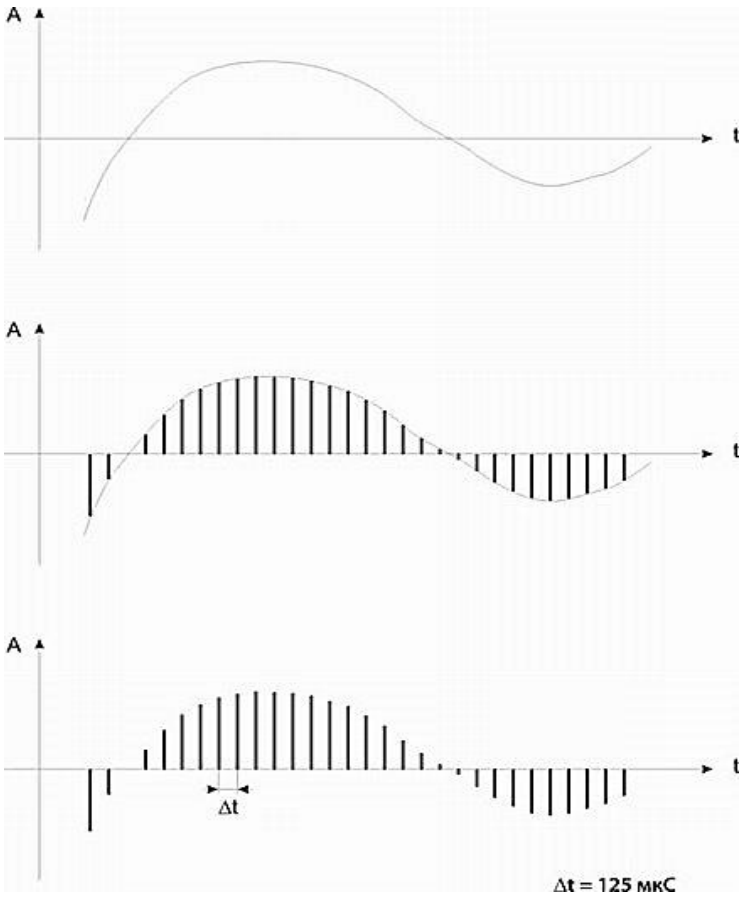


Рис.2.7. Дискретизация сигнала

Для передачи по цифровым каналам значений амплитуды каждого отсчета речевого сигнала производится их квантование по уровню. В зависимости от требований к качеству восстановленного аналогового сигнала, для представления его в цифровом виде могут использоваться кодовые последовательности с различной разрядностью. Для записи на компакт-дисках используется 16 разрядов, что соответствует 65536 шагам квантования. В телекоммуникациях используются только 8 разрядов, обеспечивающих 256 шагов квантования. Операция квантования по уровню изображена на рисунке 2.8.

Кодовая последовательность должна позволять описывать весь возможный диапазон амплитуд речевого сигнала, т.н. динамический диапазон. Линейное квантование, когда шаг квантования имеет линейную зависимость от входного сигнала, малоэффективно, так как вероятность

появления отсчетов с большой амплитудой достаточно мала и кодовое пространство используется неэффективно. Вдобавок при линейной зависимости соотношение сигнал/шум для отсчетов с большой амплитудой выше, чем для отсчетов с малыми амплитудами. Для устранения этого явления и более точного кодирования малых значений применяется операция компадирования, когда шаг квантования имеет разное значение, увеличивающиеся с ростом амплитуды отсчетов.

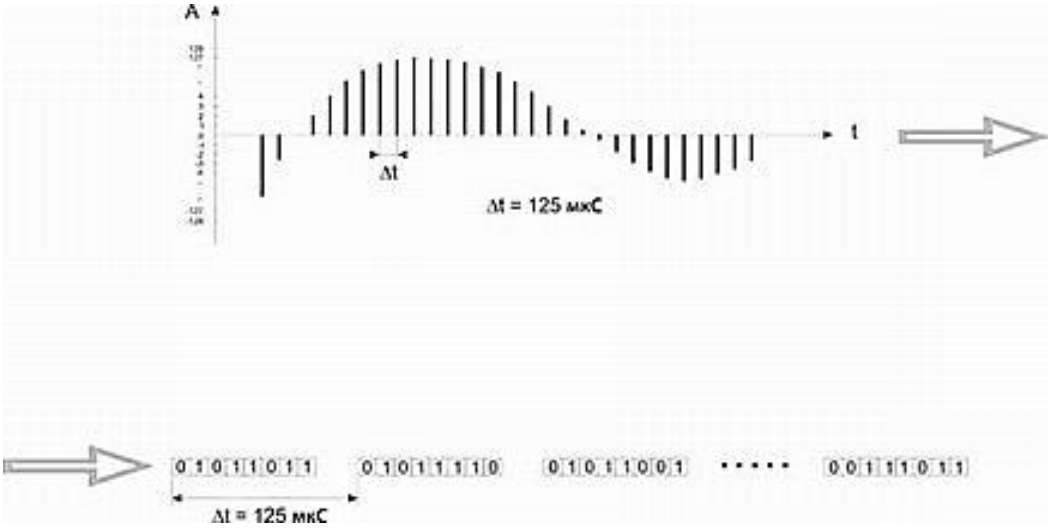


Рис.2.8. Операция квантования по уровню

Используются две характеристики компадирования, одобренные ITU-T (рекомендация G.711). В США, Канаде, Японии и некоторых других странах нашло распространение компадирование по закону "μ", в Европе применяется компадирование по закону «А».

Оба этих закона построены на одинаковых принципах. Весь динамический диапазон разделяется на 16 сегментов, 8 для положительной и 8 для отрицательной полярностей входного сигнала. Каждый сегмент в свою очередь разделен на 16 шагов квантования. Таким образом, кодовая последовательность состоит из одного разряда полярности, трех разрядов с номером сегмента и четырех разрядов с номером шага квантования. Характеристика компадирования по закону "μ" обеспечивает лучшую передачу слабых сигналов, но уступает закону А по динамическому

диапазону. При использовании закона А реально получается 13 сегментов, четыре сегмента малых амплитуд аппроксимированы в один сегмент. На рисунке 2.9 показана кривая компадирования по закону А.

На приемной стороне для восстановления исходного сигнала до цифро-аналогового преобразования применяется операция экспандирования. В настоящее время операции компадирования и экспандирования реализуются с помощью таблиц в ПЗУ компьютера.

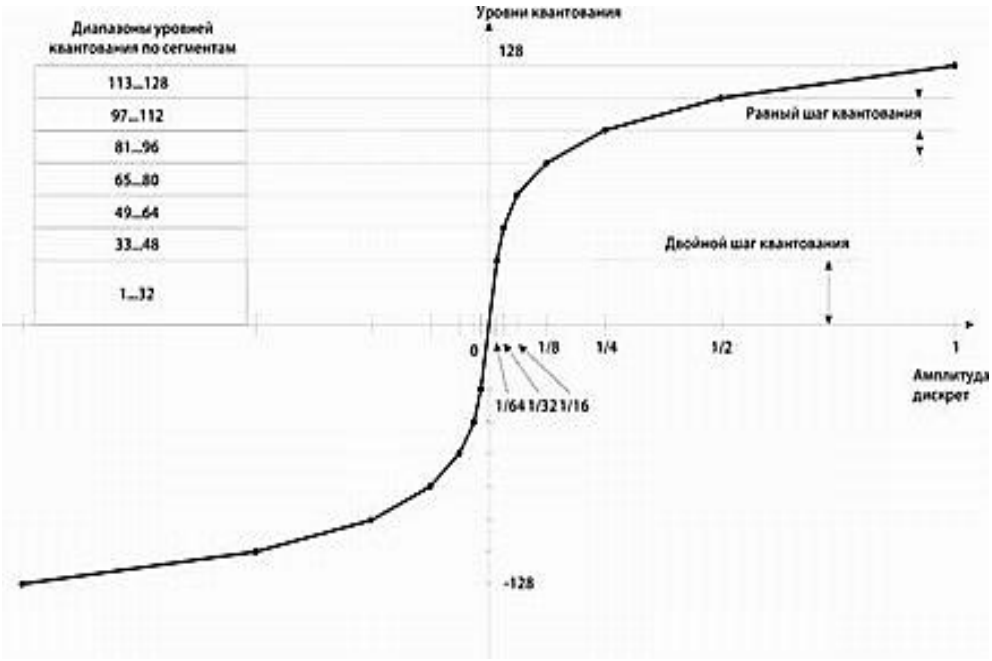


Рис.2.9. Компадированияпо закону А

Частотный диапазон звука находится в пределах 70-7000 Гц. При оценке уровня громкости звука в качестве эталона звукового давления P_0 выбирается его минимальное значение на частоте 1 кГц, при котором звук становится уже слышимым, т.е. $P_0=2\times10^{-5}$Н/м^2. Уровень звукового давления определяется соотношением

$$L = 20\lg\frac{P}{P_0} \quad (Дб),$$

где P - значение звукового давления. Под динамическим диапазоном понимают разность между максимальным и минимальным уровнями сигналов. Динамический диапазон речи составляет 35-45 Дб.

Цифровая система звукозаписи требует представления аналогового речевого сигнала в цифровом виде. В результате аналого-цифрового преобразования непрерывный сигнал переводится в ряд дискретных отсчетов, каждый из которых представляет собой целое число, характеризующее аналоговый сигнал в этой точке с определенной точностью. Точность представления зависит от ширины диапазона получаемых чисел, а следовательно от разрядности АЦП.

Процесс измерения сигнала с округлением до разряда АЦП носит название квантования. Задаваясь требуемым динамическим диапазоном цифровой системы звукозаписи, необходимое число разрядов квантования можно определить из выражения $D=6n+1.8$, где D - динамический диапазон (в Дб), n - число двоичных разрядов. Отсюда получаем, что для записи речи необходимо отводить не менее восьми бит на каждый отсчет.

Приведем некоторые из используемых частот дискретизации звука:

- 8 000 Гц — телефон, достаточно для речи;
- 11 025 Гц;
- 16 000 Гц;
- 22 050 Гц — радио;
- 32 000 Гц;
- 44 100 Гц — используется в Audio CD;
- 48 000 Гц — DVD, DAT;
- 96 000 Гц — DVD-Audio (MLP 5.1);
- 192 000 Гц — DVD-Audio (MLP 2.0);

Чем выше частота дискретизации, тем более широкий спектр сигнала может быть представлен в дискретном сигнале. Как следует из теоремы Котельникова, для того, чтобы однозначно восстановить исходный сигнал, частота дискретизации должна в два раза превышать наибольшую частоту в

спектре сигнала.

Параметры речевого сигнала.

При распознавании речевых сигналов, как правило, оперируют не с исходным речевым сигналом, а с его параметрами, вычисленными на кадре. Длина кадра обычно выбирается такой, чтобы длительность кадра по времени $T=N/\nu$ (сек.) составляла 10-20 мс. Пусть на текущем кадре длины N наблюдается последовательность отсчетов $s_1,...,s_k,...,s_N$.

Рассмотрим основные параметры речевого сигнала, используемые ниже.

1. Кратковременная энергия речевого сигнала

$$E = \frac{1}{N}\sum_{k=1}^{N} s_k^2 \quad .$$

2. Число нулей интенсивности Z

$$Z = \frac{1}{2}\sum_{k=2}^{N} \left|\mathrm{sign}(s_k) - \mathrm{sign}(s_{k-1})\right| \quad ,,$$

$$\mathrm{sign}(s) = \begin{cases} 1, & s \geq 0; \\ -1, & s < 0. \end{cases} \quad .$$

3. Коэффициенты разложения в ряд Фурье.

Кадр определяет периодическую функцию с периодом 1, заданную на сетке из точек вида $x_l=l/N$:

$f_l=f(x_l)=s_{k+1}$, если $l=Nt+k$, где $0\leq k\leq N-1$, t - целое.

Такую функцию можно разложить в ряд Фурье, т.е. представить в виде

$$f_l = \sum_{q=0}^{N-1} A_q \exp\{2\pi i q x_l\} \quad .$$

Скалярное произведение для функций на сетке определяется сдедующим образом:

$$(f,g) = \frac{1}{N}\sum_{l=0}^{N-1} f_l \overline{g}_l \quad .$$

Функции $g_q(x_l)=\exp\{2\pi iqx_l\}$ при $0\le q<N$ образуют ортонормированную систему относительно так введенного скалярного произведения. Коэффициенты Фурье можно найти по формуле

$$A_q = (f, g_q) = \frac{1}{N}\sum_{l=0}^{N-1} f_l \exp\{2\pi iqx_l\} \qquad (2.8)$$

Непосредственное осуществление этих преобразований требует N^2 арифметических операций. Для сокращения этого числа применяется алгоритм быстрого преобразования Фурье. Алгоритм основан на том, что при $N=2^m$ в слагаемых правой части выражений (2.8) можно выделить группы, входящие в выражения различных коэффициентов A_q. Вычисляя каждую группу только один раз можно сократить число операций до $N\times\log_2 N$. Если $N\neq 2^m$, то в нашем случае можно добавить нулевые отсчеты.

4. Распределение энергии сигнала по частотным группам $p_1,...,p_{20}$.

Одним из важнейших свойств слуха является разделение спектра звука на частотные группы. Слух может образовывать частотные группы на любом участке шкалы частот. В области частот ниже 500 Гц ширина частотных групп почти не зависит от средней частоты групп и составляет примерно 100 Гц. В области выше 500 Гц она увеличивается пропорционально средней частоте. Если частотные группы совместить в один ряд, то в диапазоне от 70 Гц до 7 кГц разместятся 20 частотных групп. Распределение энергии по частотным группам можно найти либо непосредственно с помощью гребенки соответствующих фильтров, либо с помощью коэффициентов разложения в ряд Фурье. Значение p_i для частотной группы от частоты ν_{i-1} до ν_i с шириной $H_i=\nu_i-\nu_{i-1}$ определяется по формуле:

$$p_i = \left(\frac{1}{n}\sum_{j=0}^{n-1} C_{q_i+j}^2\right) H_i .$$

2.3. Спектральный анализ речевого сигнала

Предварительная обработка речевых сигналов.

Создание естественных для человека средств общения с компьютером является в настоящее время важнейшей задачей современной науки, при этом речевой ввод информации осуществляется наиболее удобным для пользователя способом. Распознавание речи является задачей классификации образов акустических характеристик речевых сигналов.

Предварительная обработка речевого сигнала включает в себя следующие этапы:

- процесс ввода речевого сигнала;
- выделение границы речевого сигнала;
- цифровая фильтрация;
- сегментация речевого сигнала перекрывающимися кадрами;
- обработка сигнала в окне;
- спектральное преобразование;
- нормирование частотного спектра.

Процесс ввода речевого сигнала. Ввод звука осуществляется в реальном времени через звуковую карту или через файлы формата WAV в кодировке PCM. Частота дискретизации 8 КГц и квантование 16 бит являются типовыми параметрами в системах передачи, хранения и обработки речевой информации. Работа с файлами была предусмотрена, чтобы облегчить многократное повторение обработки нейронной сети, что особенно важно при обучении.

Выделение границы речевого сигнала. Для вычленения из входного сигнала участков, содержащих только речь, используются следующие характеристики речевого сигнала:

- кратковременная энергия речевого сигнала;
- число нулей интенсивности (мгновенная частота);
- плотность распределения значения отчетов паузы.

Кратковременная энергия звукового сигнала и число нулей интенсивности одновременно используются для выделения речи из входного

сигнала. Кроме того, можно удалить паузу из выходного сигнала методом на основе нормального (гауссова) распределения.

Цифровая фильтрация. Вместе с полезным сигналом обычно попадают различные шумы. Шум оказывает отрицательное воздействие на качество работы систем распознавания речи, поэтому с ним приходится бороться. Для снижения уровня шума в подсистеме применяются два типа цифрового фильтра:

- пропускающий полосовой фильтр;
- предварительный фильтр.

Пропускающий полосовой фильтр можно представить себе в виде комбинации фильтра нижних и верхних частот. Такой фильтр задерживает все частоты, ниже так называемой нижней частоты пропускания, а также выше верхней частоты пропускания.

Предварительная фильтрация представляется для снижения влияния локальных искажений на характерные признаки, которые в дальнейшем будут использоваться для распознавания. Для спектрального выравнивания речевого сигнала его следует пропустить через взвешивающий низкочастотный фильтр.

Сегментация речевого сигнала перекрывающимися кадрами. Для того чтобы получить векторы признаков одинаковой длины, нужно разделть речевой сигнал на равные части, а затем выполнять преобразования внутри каждого кадра. Перекрытие используется для предотвращения потери информации о сигнале на границе.

Чем меньше перекрытие, тем меньшей размерностью в итоге будет обладать вектор свойств, характерный для рассматриваемого участка. Перекрытие иногда пропускается по причине экономии вычислительных ресурсов, поскольку он существенно замедляет скорость обработки данных. Обычно выбирается длина сегментов, соответствующая временному интервалу в 20-30мс.

Обработка сигнала в окне. Обработка сигнала в окне представляется для снижения граничных эффектов, возникающих в результате сегментации. Для подавления нежелательных граничных эффектов принято умножать сигнал на оконную функцию. Существует 4 типа оконных функций:

- прямоугольное окно;
- окно Ханна;
- окно Хемминга;
- окно Блэкмана.

Чаще всего в качестве функции используется окно Хэмминга.

Спектральное преобразование. Информации об амплитуде и форме огибающей речевого сигнала недостаточно для выделения из речи лексических элементов. В зависимости от различных обстоятельств форма огибающей речевого сигнала может меняться в широких пределах. Для решения задачи распознавания необходимо выделить первичные признаки речи, которые будут использованы на последующих этапах процесса распознавания. Первичные признаки выделяются посредством анализа спектральных характеристик речевого сигнала. Для получения частотного спектра речевого сигнала используется быстрое преобразование Фурье (БПФ). БПФ представляется для получения амплитудного спектра и информации о фазе сигнала (в реальных и мнимых коэффициентах). Информация о фазе сигнала отбрасывается и вычисляются амплитудные спектры. При этом чаще используется логарифм этого значения.

$$S_i = \sqrt{ReC_i * ReC_i + ImC_i * ImC_i}$$

$$L_i = 20Log_{10}S_i,\ i = 1 \dots NS,$$

$$NS = \frac{N}{2}$$

где S_i–амплитудный спектр i-ой частоты, ReC_i–реальный коэффициент, ImC_i–мнимый коэффициент, N – размер БПФ, NS– размер информативной части спектра.

Так как звуковые данные не содержат мнимой части, то по свойству БПФ результат получается симметричным, т.е. Таким образом, размер информативной части спектра NS равен N/2.

Нормирование частотного спектра. Все вычисления в нейронных сетях производятся нормирование над числами с плавающей точкой. Поэтому значения параметров объектов, классифицируемых с помощью нейронных сетей, ограничены диапазоном [0.0, 1.0]. Для выполнения обработки спектра нейронной сетью полученный спектр нормируется на 1.0. Для этого каждый компонент вектора делится на его максимальный компонент.

Обработка спектра речевого сигнала. В системах обработки аналоговый речевой сигнал поступает на вход микрофона, с выхода которого снимается электрический сигнал. Далее сигнал подвергается дискретизации по времени и квантованию по амплитуде (рис.2.10).

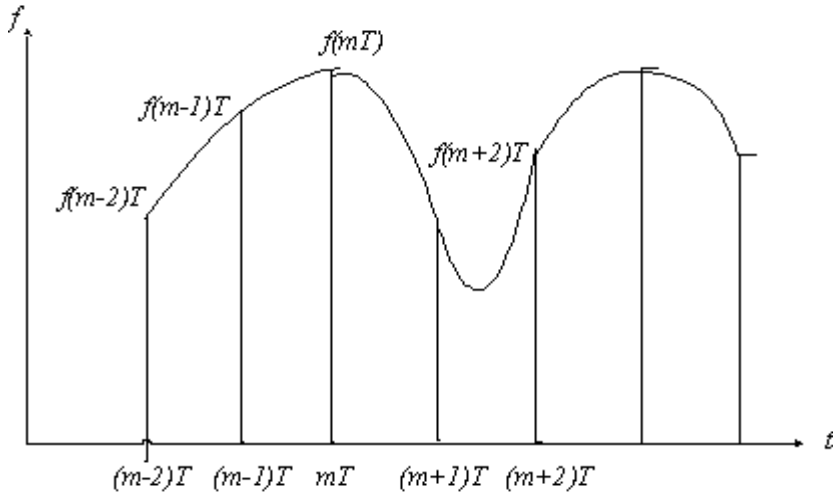


Рис. 2.10. Квантование сигнала

В процессе квантования возникают искажения (ошибки квантования), которые, в сущности, означают потерю информации.

Исходная информация представлена в виде зависимости амплитуды от времени (обычно это .wav файлы).

Полученная последовательность цифровых данных в дальнейшем подвергается обработке, с целью определения частотного диапазона и других характеристик сигнала, на основе которых его можно воспроизвести. Поскольку сигнал обычно зашумлен, простейшим способом удаления шума является обнуление тех значений сигнала, которые меньше некоторого порогового значения.

Временная форма представления сигнала, т. е. изменения сигнала в зависимости от времени, позволяет определить амплитуду, энергию, мощность и длительность.

Кроме временных характеристик сигнала важны и его частотные свойства. Для их исследования используются частотные представления функции в виде спектра. Спектральное представление сигнала – разложение его на конечную или бесконечную сумму гармонических сигналов. Знание частотных свойств сигнала позволяет решать задачи идентификации сигнала (определение его наиболее информативных параметров), фильтрации (выделение полезного сигнала на фоне помех), выбора частоты дискретизации непрерывного сигнала, так как этот параметр является определяющим для аппаратуры обработки.

Один из вариантов предварительной обработки речевого сигнала приведен на рис. 2.11.

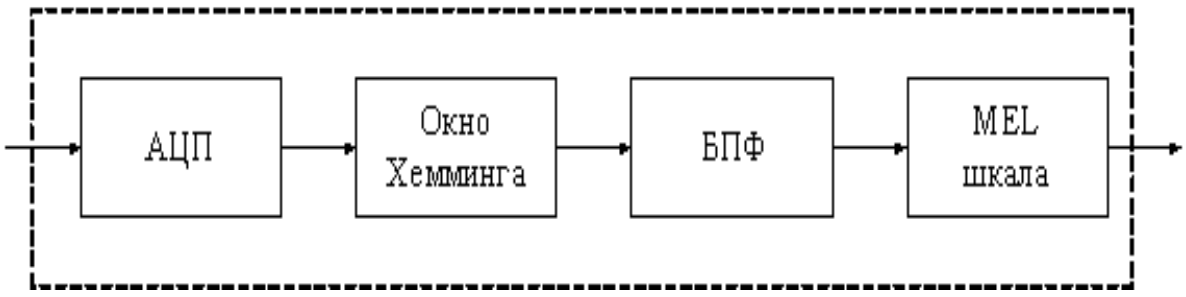


Рис. 2.11. Предварительная обработка речевого сигнала

Взвешивание сигнала весовой функцией окна Хэмминга (рис. 2.12) уменьшает спектральные искажения сигнала из-за граничных условий, т.е.

возникновения при сегментации сигнала разрывов на границе сегментов (на математическом языке это называется разрывом функции первого рода). Применение временного окна целесообразно для интервалов, превышающих 15 мс или включающих несколько периодов основного тона. Значение взвешивающей функции задается формулой:

$$W_n = \begin{cases} 0{,}54 - 0{,}46 \cdot \cos\left(\frac{2\pi n}{N-1}\right), 0 < n < N \\ 0, \textit{иначе} \end{cases} \qquad (2.13)$$

Информативность различных частей спектра неодинакова: в низкочастотной области содержится больше информации, чем в высокочастотной.

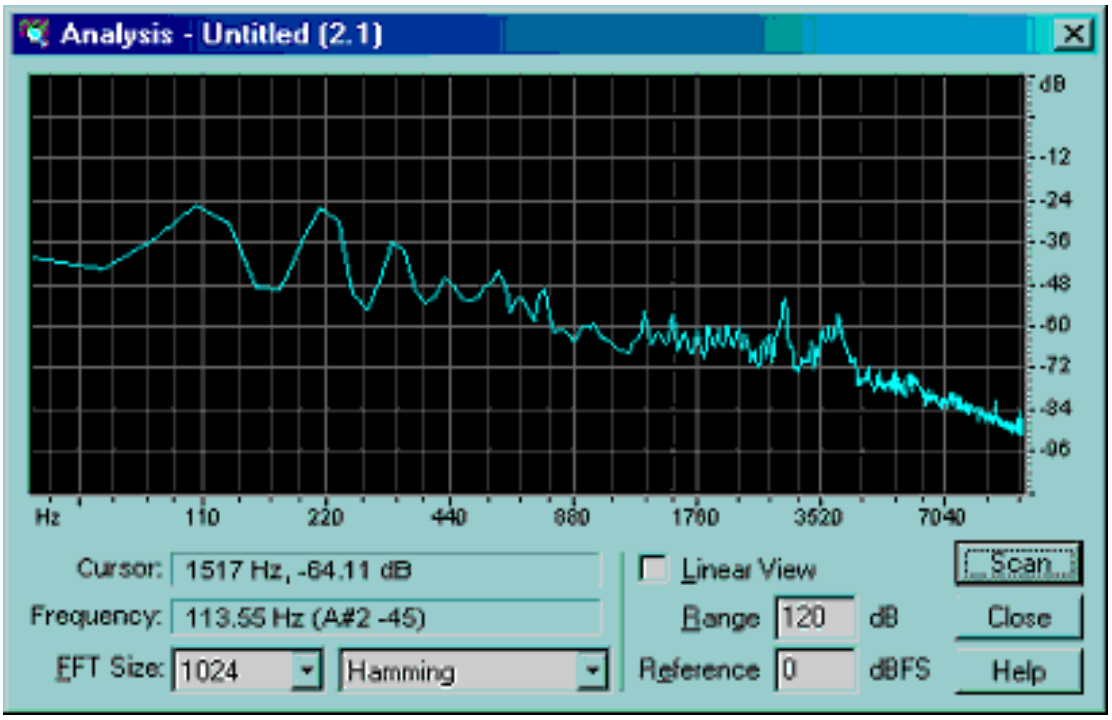


Рис. 2.12. Фурье – преобразование речевого сигнала с использованием окна Хэмминга (Hamming)

Это объясняется тем, что сам речевой сигнал является низкочастотным, поэтому сжимают высокочастотную область спектра в пространстве частот.

Наиболее распространенным методом (благодаря его относительной простоте) это – логарифмическое сжатие, или mel-сжатие:

$$m = 1125 \cdot \log(0.0016 f + 1) \qquad (2.14)$$

где f – частота в спектре, Гц; m - частота в новом сжатом частотном пространстве.

Образцы сегментов речевых сигналов приведены на рис. 2.13.

На рис. 2.14 показан результат частотного анализа 16-битного речевого сигнала с частотой дискретизации 11025 Гц, выполненный в окне анализа Analyze – Frequency Analysis стандартного звукового редактора Cool Edit. Подобный спектр колебаний воздуха формируется голосовыми связками и источником звука в ротовой полости путем избирательного резонанса, возникающего при передаче звука по речевому тракту. Речевой тракт образуют гортань, ротовая полость, язык, носовая полость.

Редактор позволяет записывать и проигрывать файлы в разных аудио-форматах, редактировать, конвертировать и смешивать звуковые файлы, генерировать шум и различные тона, выполнять частотный анализ.

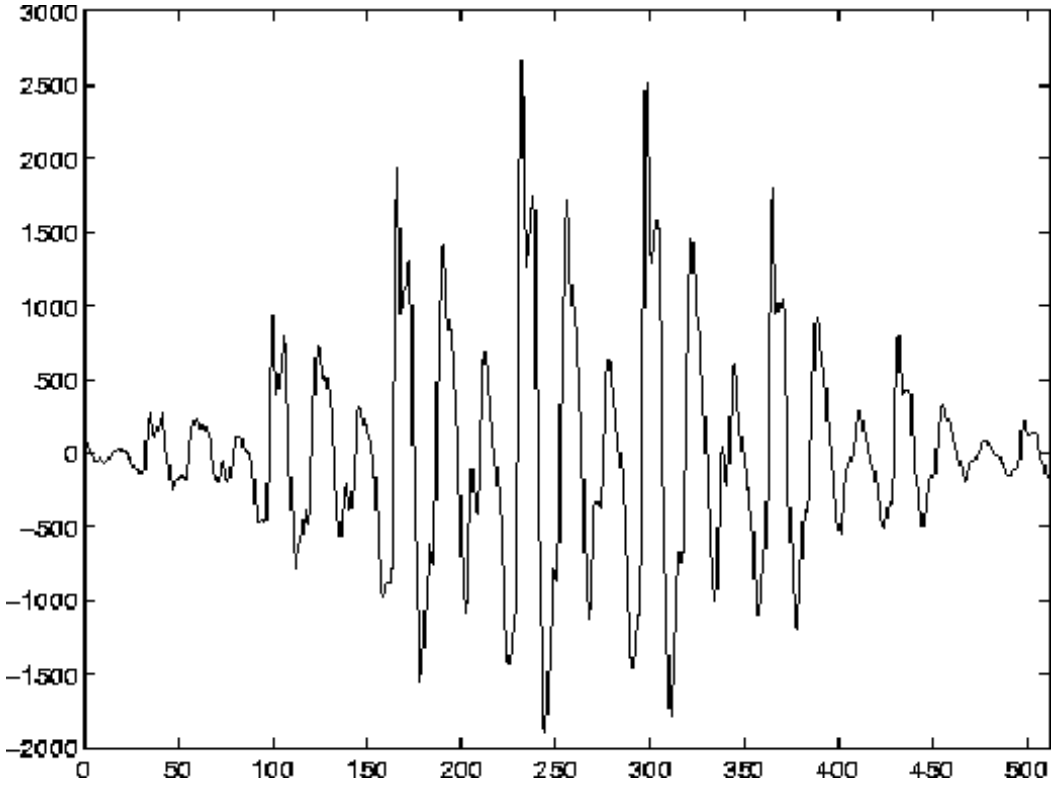
Речевой сигнал имеет ряд особенностей, которые необходимо учитывать:

- свойства сигнала не постоянны на выбранном для анализа отрезке длиной в слово, это нестационарный случайный процесс;

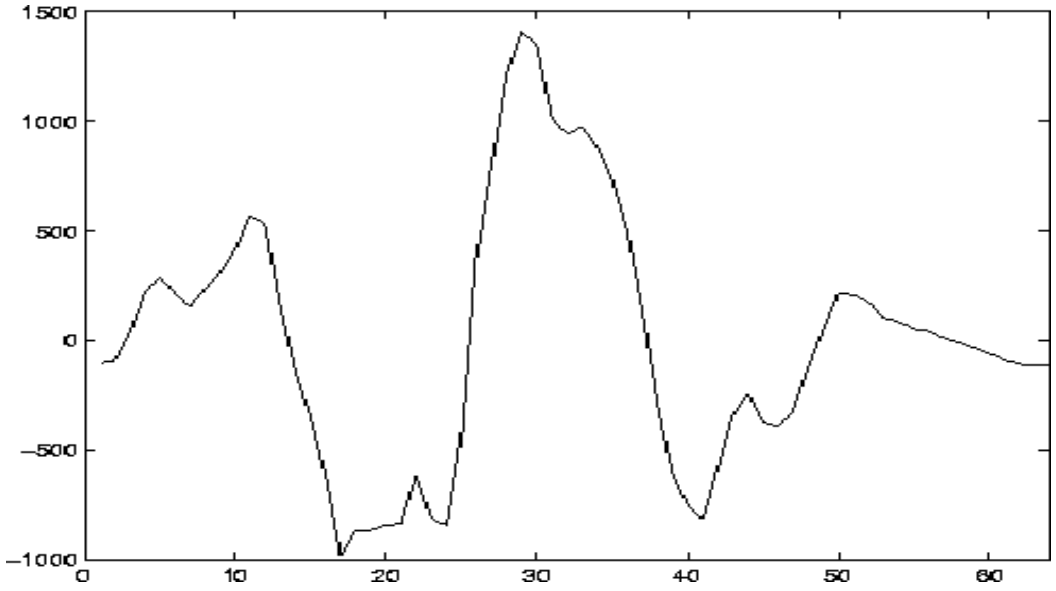
- сложность формы сигнала (речь напоминает скорее шум, чем регулярный сигнал).

Для преодоления этих трудностей, как указывалось выше, дискретный случайный процесс оцифрованного речевого сигнала считается стационарным на интервале порядка 10 мс. Считается, что параметры голосового тракта на этом интервале значительно не изменяются. Это обоснованный экспериментально временной интервал, который используется

при сегментации и последующей обработке текущих фрагментов входного сигнала.



а)



б)

Рис. 2.13. Сегменты речевых сигналов: а) сегмент выделен с использованием окна Hamming, б) сегмент гласной

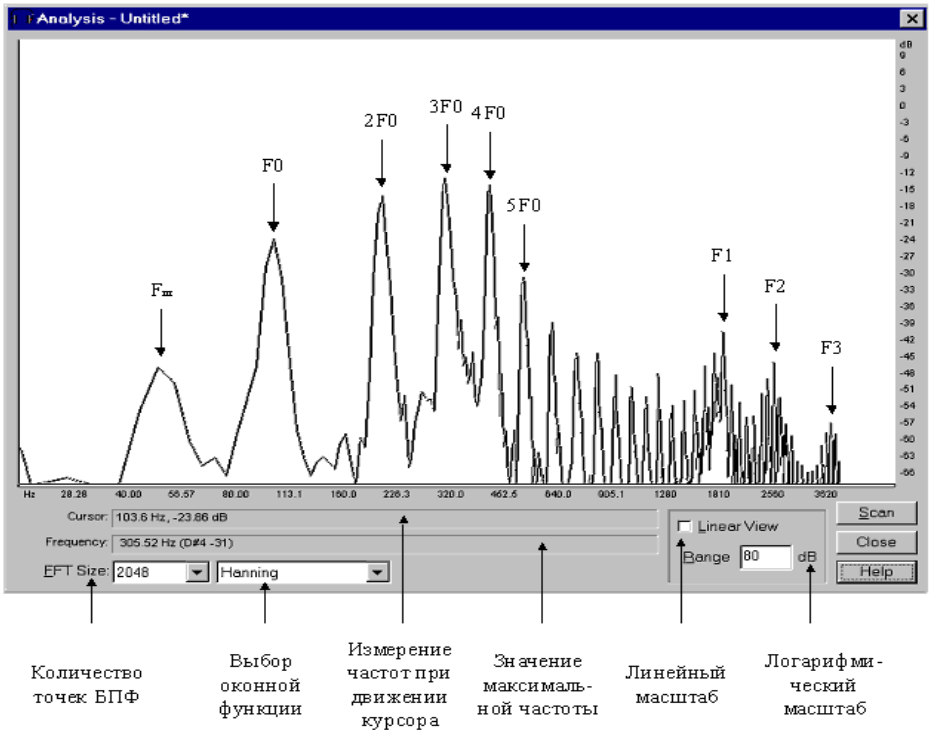


Рис. 2.14. Окно спектрального анализа:

Fш – частота шума, Fo – частота основного тона,

2F0-5F0 – обертоны, 3F0,F1–F3 – формантные частоты

Основная задача обработки сигнала состоит в вычислении по входному сигналу совокупности параметров (признаков), которые содержат информацию о сигнале, используемую при синтезе и распознавании (рис.2.15).

Обычно определяют следующие параметры сигнала:

- частоту основного тона для формирования траектории периода основного тона;
- кратковременную энергию для синтеза траектории кратковременной энергии;
- коэффициенты линейного предсказания (КЛП) для построения траектории передаточной функции речеобразующего тракта;

- формантные частоты для воспроизведения траектории формантных частот.

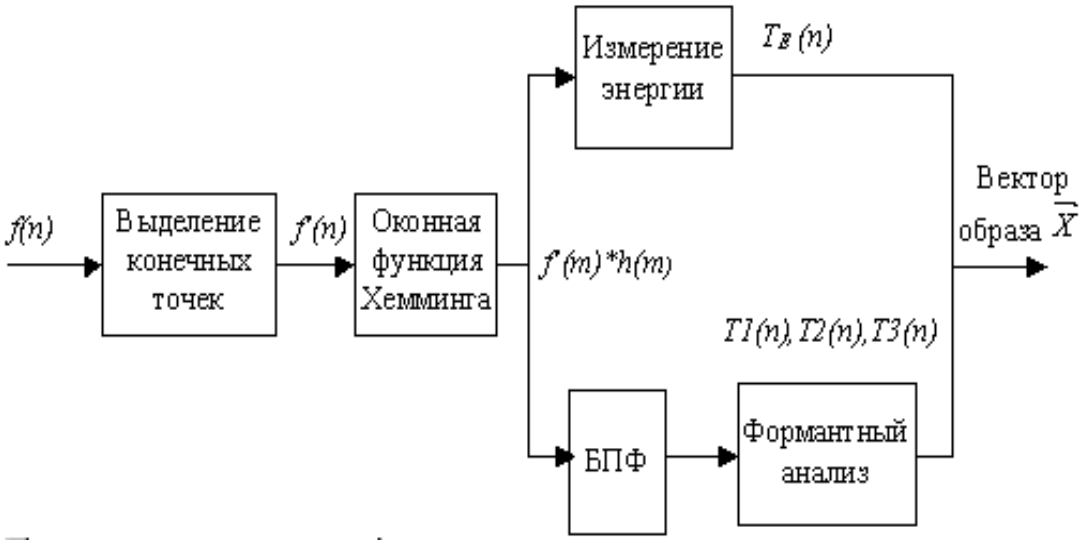


Рис. 2.15. Блок обработки сигнала

Форманты – максимумы распределения энергии звукового сигнала в координатах амплитуда, частота, время. Для получения хорошего качества сигнала достаточно задать параметры нескольких старших формант основного тона. Когда нужно достичь высокого качества, применяются некоторые из перечисленных параметров или их комбинации. Проблема отделения речи от шума довольно сложна, т.к. при произнесении некоторых согласных ("**ф**", "**п**", "**т**") энергия речевого сигнала практически равна энергии шума.

Один из алгоритмов выделения фразы основан на измерении двух простых характеристик – энергии и числа переходов через нуль. При подсчёте среднего значения энергии используется окно в 10 мс (примерно 110 отсчётов), в котором суммируются квадраты отсчётов (рис. 2.16).

В пределах выбранного временного сегмента, вычисляется среднее значение энергии шума $E_{шума}$ и порог P, который берется равным удвоенной энергии шума. При дальнейшей обработке, если среднее значение энергии превысило порог, то фиксируется момент записи речевого сигнала (начало

фразы), который запоминается. Если среднее значение энергии станет меньше порога, то запоминается конец фразы.

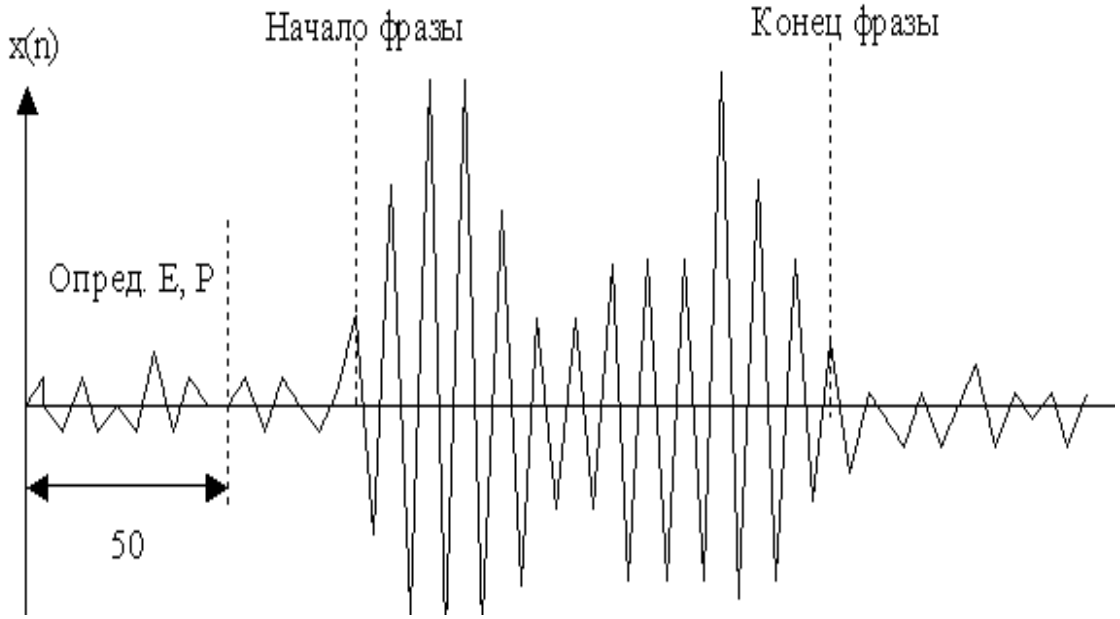


Рис. 2.16. Выделение фразы

На рис. 2.17–2.20 приведены графики изменения энергии сигнала для фонем "**Р**", "**Л**" в зависимости от гласного звука, следующего за ними. Предполагается, что первые 50 мс сигнал не содержат речевого сигнала.

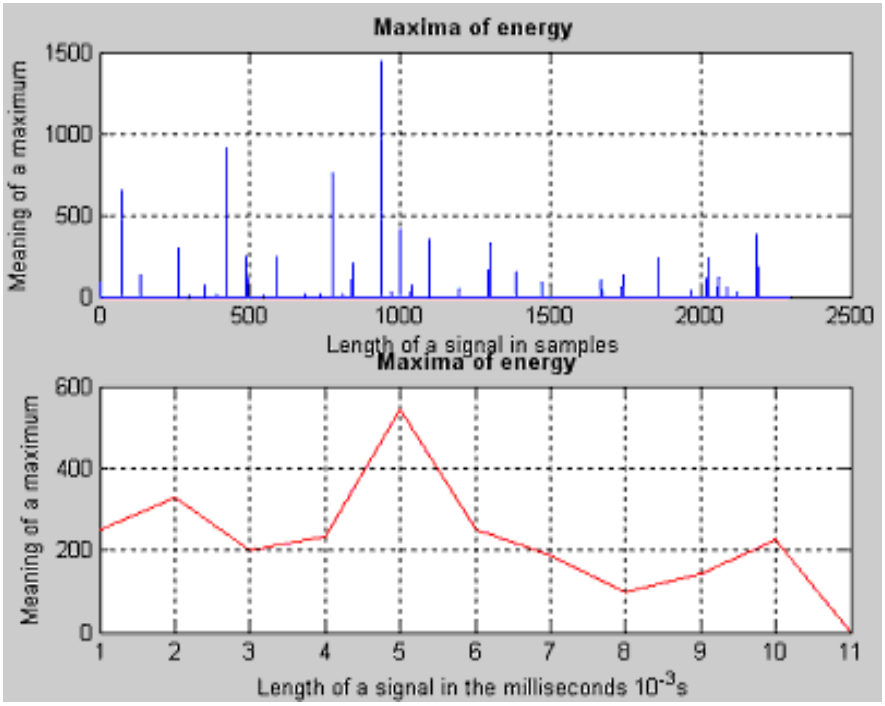


Рис. 2.17. Максимумы энергии в спектре фонемы "Р" в слове «РОК»

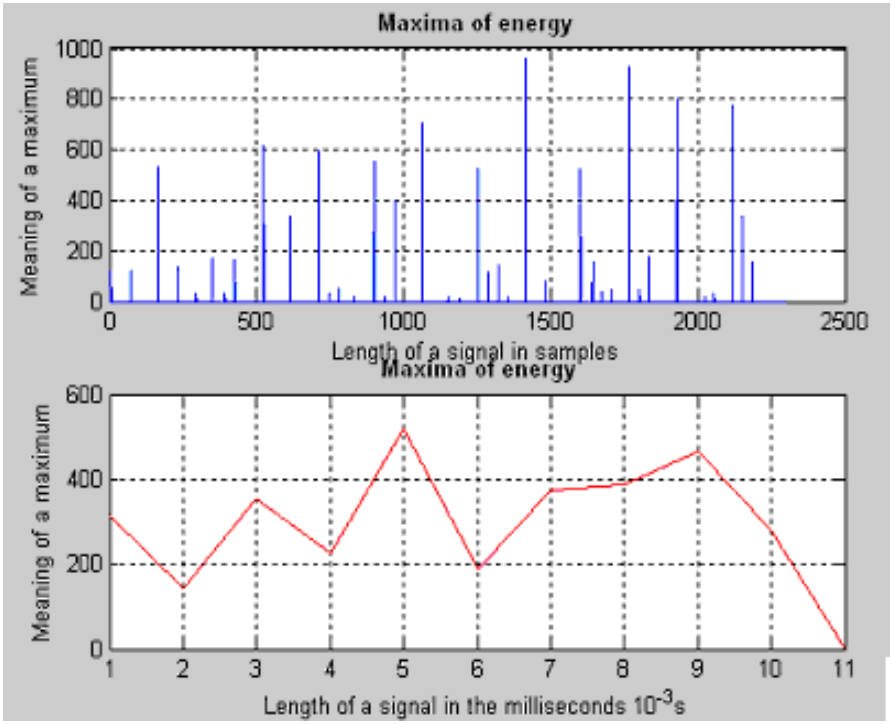


Рис. 2.18. Максимумы энергии в спектре фонемы "Л" в слове «ЛОК»

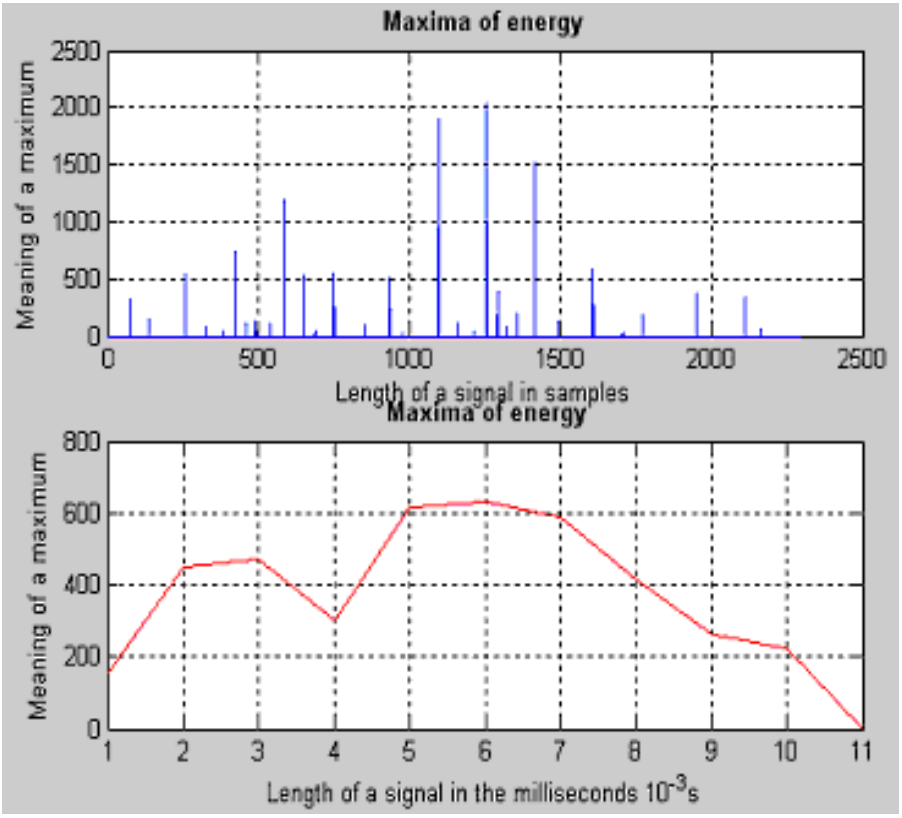


Рис. 2.19. Максимумы энергии в спектре фонемы "Р" в слове «РЁВ»

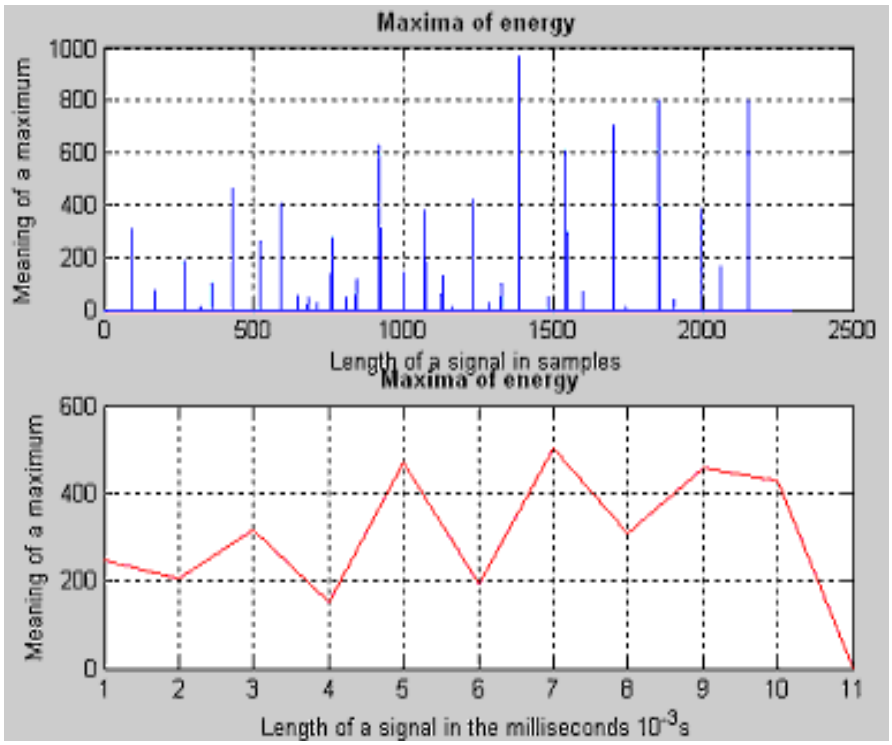


Рис. 2.20. Максимумы энергии в спектре фонемы "Л" в слоге «ЛЁВ»

Образцы сигналов и их спектрограмм приведены на рис. 2.21.

Частота основного тона, энергия и длительность обеспечивают формирование просодических характеристик речи.

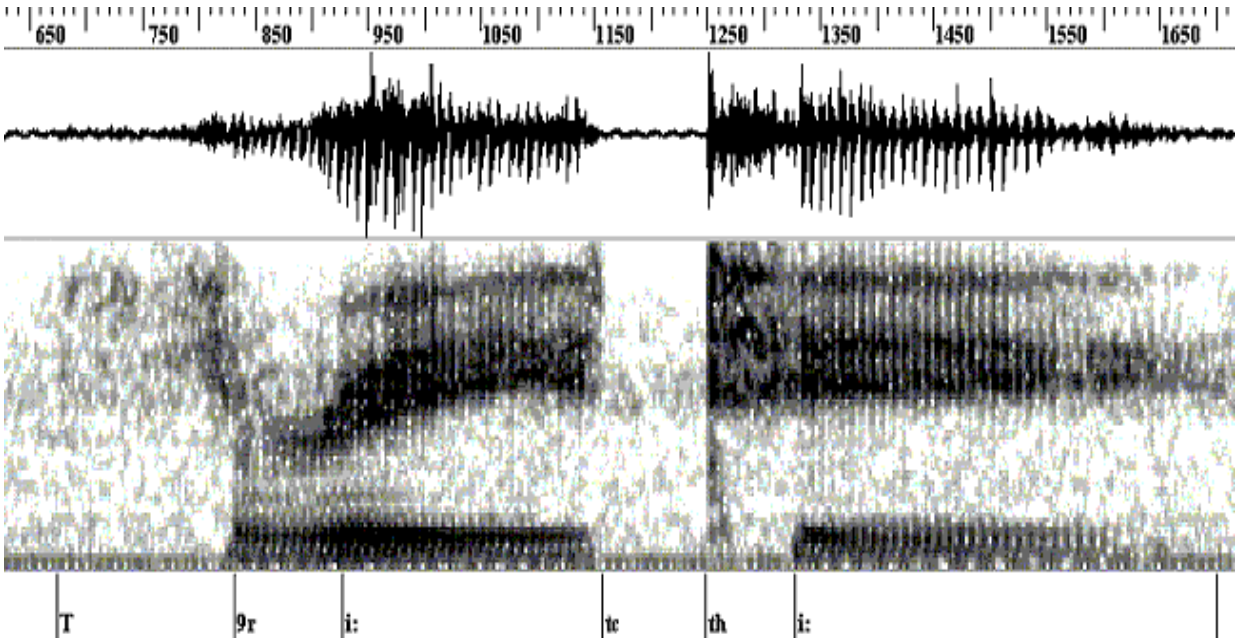


Рис. 2.21. Сигналы и их спектрограммы

Визуализация параметров сигнала в координатах амплитуда, частота, время приведена на рис. 2.22.

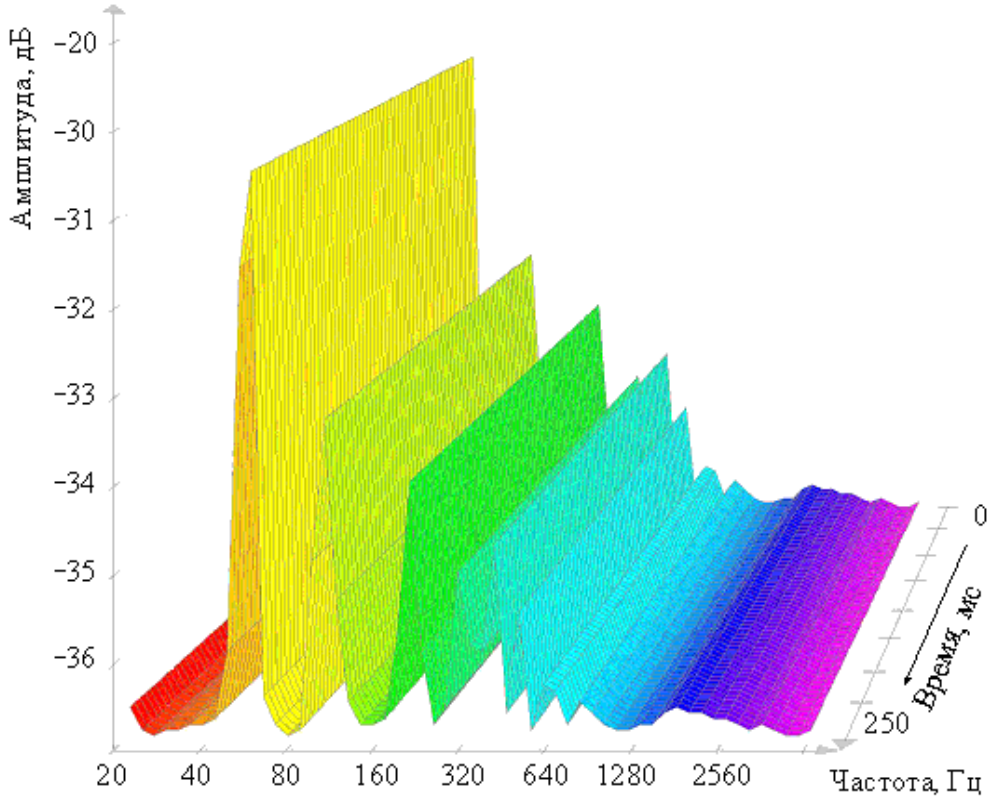


Рис. 2.22. Сигнал в координатах частота-амплитуда-время

Измерение частоты основного тона. Частота основного тона является одной из важнейших характеристик речевого сигнала. Существуют различные способы оценки этого параметра, в частности, можно воспользоваться спектральным анализом. Если найдено ДПФ, то можно восстановить исходный сигнал (обратное преобразование Фурье) по дискретным значениям сигнала. Структура системы вычисления частоты основного тона приведена на рис. 2.23.

Поскольку обратное ДПФ линейно, сигнал в точке D (называемый «кепстром» сигнала в точке A) равен сумме кепстров функции возбуждения и импульсной характеристики голосового тракта. Можно показать, что кепстр в точке D позволяет разделить эффекты возбуждения и характеристики голосового тракта. Действительно, сигнал возбуждения

можно рассматривать как квазипериодическую импульсную последовательность с преобразованием Фурье, близким к линейчатому, причем спектральные линии соответствуют гармоникам частоты основного тона. Вычисление логарифма модуля не меняет линейчатого характера спектра функции возбуждения. Обратное ДПФ дает новую квазипериодическую последовательность импульсов с интервалами между импульсами, равными периоду основной частоты. Таким образом, кепстр сигнала возбуждения должен состоять из импульсов, расположенных вблизи $n = 0, T, 2T,...,$ где T – период основного тона. Импульсная характеристика голосового тракта обычно представляет собой последовательность, отличную от нуля на интервале 20–30 мс. После вычисления логарифма модуля и обратного ДПФ получается последовательность из небольшого числа ненулевых отсчетов, которое обычно меньше, чем число отсчетов на периоде основного тона.

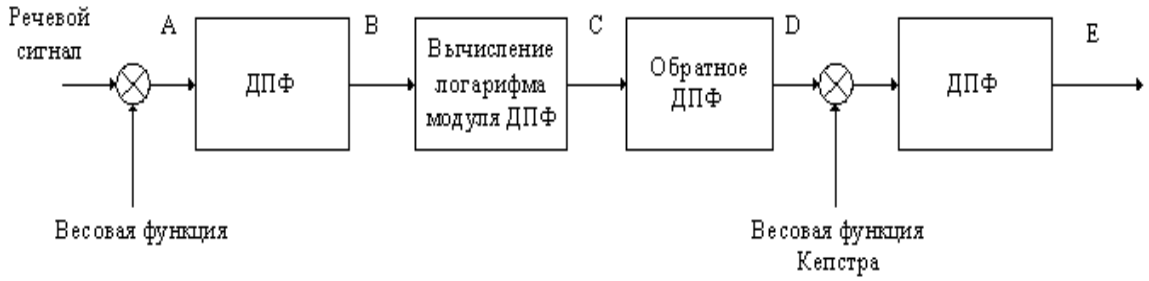


Рис. 2.23. Вычисление частоты основного тона

Результат вычисления кепстра вокализованного сигнала показан на рис. 2.24.

Исследования показали, что для вокализованного сегмента речи в кепстре возникает пик, соответствующий периоду основного тона. Для невокализованного сегмента такие пики в кепстре не возникают. Это свойство кепстра может быть использовано для классификации звуков на вокализованный, невокализованный и для вычисления периода основного тона вокализованной речи.

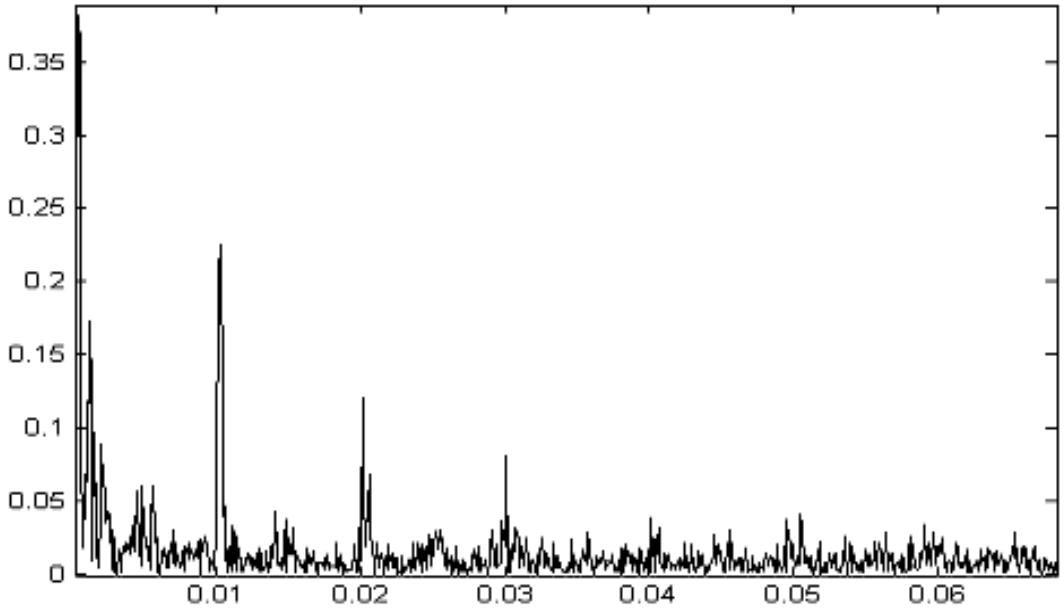


Рис. 2.24. Кепстр вокализованного сигнала

Кепстр, полученный описанным выше способом, исследуется с целью отыскания пика в области возможных значений основного тона (4–40 мс).

Если максимум кепстра не превышает порога, то сегмент классифицируется как невокализованный. Если пик в кепстре превышает установленный порог, то сегмент классифицируется как вокализованный, а координата пика дает оценку периоду основного тона, и соответственно вычисляется частота основного тона. Таким образом можно построить эффективный алгоритм выделения частоты основного тона. Листинг одного из вариантов алгоритма приведен ниже.

Листинг программы вычисления кепстра и определения частоты основного тона речевого сигнала в среде Matlab

```
[x,fs]=wavread('c:\MATLAB6p5\work\wav\a.wav');
                                    % Открываем wav-файл, содержащий
речевой сигнал
                                    % x - отсчеты речевого сигнала
                                    % fs - частота дискретизации
```

```
nfft=2048;                          % количество точек ДПФ
window='hamming';                   % вид окна сглаживания ДПФ
nlap=0.75;                          % количество точек перекрытия (75%)
nlap = round(nlap*nfft);
nx=length(x);
nwin=nfft;
w=feval(window,nwin,'periodic');
x = x(:);
window = window(:);
                                    % Расчет размеров выходной матрицы
ncol = fix((nx-nlap)/(nwin-nlap));  % ncol - количество необходимых ДПФ
colindex = 1 + (0:(ncol-1))*(nwin-nlap);
rowindex = (1:nwin)';

                                    % Формирование выходной матрицы
y = zeros(nwin,ncol);
y(:)=x( rowindex(:,ones(1,ncol)) + colindex(ones(nwin,1),:)  -1 );
y = w(:,ones(1,ncol)).*y;

y = fft(y,nfft);                    % ДПФ входного речевого сигнала
y2 = y;
                                    % Убираем мнимую часть спектра
select = [1:nfft/2+1];
y = y(select,:);

[lll,ll]=size(y2);                  % Размеры матрицы ДПФ
ll=round(ll/2)
%sm=ll;
```

```
                                        % Вычисляемкепстр
r= ifft ( log(abs(y2)) );           % ОбратноеДПФотлогарифмаДПФ
r=r(:,ll);                          % Выделяем кепстр на отрезке сигнала
r1=r;                                   % Сохраняем отсчеты кепстра для
построения графика
r1(1)=0;
r1(2)=0;

r(1:0.002*fs)=0;                        % Устранение из кепстра информации
о речевом тракте

ll=size(r);
lll=round(ll(1)/2);
ss = [1 : lll];
r = r(ss,:);                            % Убираем мнимую часть кепстра
r1=r1(ss,:);
[f0m,T0]=max(r);                        % Определяем временную координату
пика кепстра
f0=1/(T0/fs)                            % Значение частоты основного тона в
Герцах

if f0m<0.05                             % если амплитуда пика кепстра <0.05,
речевой
                                        сигнал - не вокализован
f0='сегмент невокализован';
end

                                        % Графическое отображение кепстра
сигнала
time=[1:length(r1)]'/fs;            % Отсчеты времени
```

```
plot(time,abs(r1));
xlabel('Time');
```

2.4. Фильтрация звукового сигнала

Вместе с полезным сигналом в микрофон обычно попадают различные шумы - шум с улицы, шум ветра, посторонние разговоры. Шум оказывает отрицательное воздействие на качество работы систем распознавания речи, поэтому с ним приходится бороться. Один из способов мы уже упоминали - сегодняшними системами распознавания речи лучше всего пользоваться в тихой комнате, оставаясь с компьютером один на один.

Однако идеальные условия удается создать далеко не всегда, поэтому приходится использовать специальные методы, позволяющие избавиться от помех. Для снижения уровня шума применяются специальные ухищрения при конструировании микрофонов и специальные фильтры, удаляющие из спектра аналогового сигнала частоты, не несущие полезную информацию. Кроме того, используется такой прием, как сжатие динамического диапазона уровней входного сигнала.

Расскажем обо всем этом по порядку.

Применение частотных фильтров. Частотным фильтром называется устройство, преобразующее частотный спектр аналогового сигнала. При этом в процессе преобразования происходит выделение (или поглощение) колебаний тех или иных частот.

Можно представить себе это устройство в виде некоего черного ящика с одним входом и одним выходом. Применительно к нашей ситуации, к входу частотного фильтра будет подключен микрофон, а к выходу — аналого-цифровой преобразователь.

Частотные фильтры бывают разные:

- фильтры нижних частот;
- фильтры верхних частот;

– пропускающие полосовые фильтры;
– заграждающие полосовые фильтры.

Фильтры нижних частот (low-passfilter) удаляют из спектра входного сигнала все частоты, значения которых находятся ниже некоторой пороговой частоты, зависящей от настройки фильтра.

Так как звуковые сигналы лежат в диапазоне 16-20 000 Гц, то все частоты меньше 16 Гц можно отрезать без ухудшения качества звука. Для распознавания речи важен частотный диапазон 300-4000 Гц, поэтому можно вырезать частоты ниже 300 Гц. При этом из входного сигнала будут вырезаны все помехи, частотный спектр которых лежит ниже 300 Гц, и они не будут мешать процессу распознавания речи.

Аналогично, фильтры верхних частот (high-passfilter) вырезают из спектра входного сигнала все частоты выше некоторой пороговой частоты.

Человек не слышит звуки с частотой 20 000 Гц и выше, поэтому их можно вырезать из спектра без заметного ухудшения качества звука. Что же касается распознавания речи, то здесь можно вырезать все частоты выше 4000 Гц, что приведет к существенному снижению уровня высокочастотных помех.

Пропускающий полосовой фильтр (band-passfilter) можно представить себе в виде комбинации фильтра нижних и верхних частот. Такой фильтр задерживает все частоты, ниже так называемой нижней частоты пропускания, а также выше верхней частоты пропускания.

Таким образом, для системы распознавания речи удобен пропускающий полосовой фильтр, который задерживает все частоты, кроме частот диапазона 300-4000 Гц.

Что же касается заграждающих полосовых фильтров (band-stopfilter), то они позволяют вырезать из спектра входного сигнала все частоты, лежащие в заданном диапазоне. Такой фильтр удобен, например, для подавления помех, занимающих некоторую сплошную часть спектра сигнала.

На рис.2.25 показано подключение пропускающего полосового фильтра.

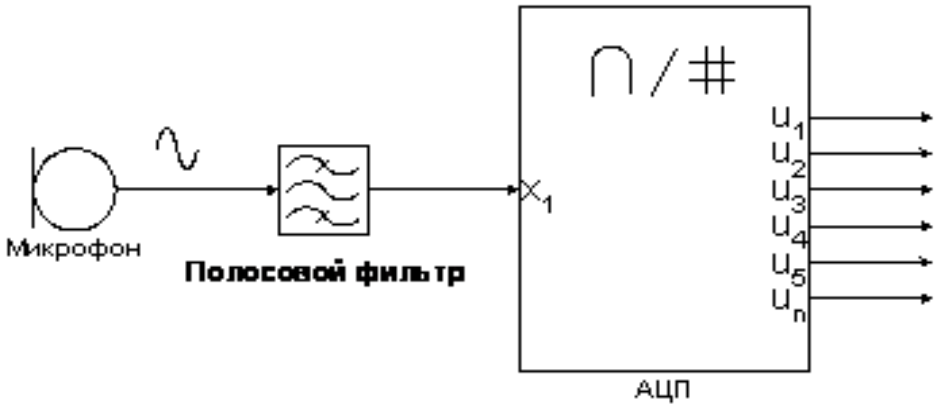


Рис. 2.25. Фильтрация звукового сигнала перед оцифровкой

Надо сказать, что обычные звуковые адаптеры, установленные в компьютере, имеют в своем составе полосовой фильтр, через который проходит аналоговый сигнал перед оцифровкой. Полоса пропускания такого фильтра обычно соответствует диапазону звуковых сигналов, а именно 16-20 000 Гц (в разных звуковых адаптерах значения верхней и нижней частоты могут изменяться в небольших пределах).

А как добиться более узкой полосы пропускания 300-4000 Гц, соответствующей наиболее информативной части спектра человеческой речи?

Можно сделать свой фильтр из микросхемы операционного усилителя, резисторов и конденсаторов. Примерно так и поступали первые создатели систем распознавания речи.

Однако промышленные системы распознавания речи должны быть работоспособны на стандартном компьютерном оборудовании, поэтому путь изготовления специального полосового фильтра тут не подходит.

Вместо этого в современных системах обработки речи используются так называемые цифровые частотные фильтры, реализованные программно.

Это стало возможным, после того как центральный процессор компьютера стал достаточно мощным.

Цифровой частотный фильтр, реализованный программно, преобразует входной цифровой сигнал в выходной цифровой сигнал. В процессе преобразования программа обрабатывает специальным образом поток числовых значений амплитуды сигнала, поступающий от аналого-цифрового преобразователя. Результатом преобразования при этом также будет поток чисел, однако этот поток будет соответствовать уже отфильтрованному сигналу.

Сжатие динамического диапазона звукового сигнала. Рассказывая об аналогово-цифровом преобразователе, мы отметили такую его важную характеристику, как количество уровней квантования. Если в звуковом адаптере установлен 16-разрядный аналого-цифровой преобразователь, то после оцифровки уровни звукового сигнала могут быть представлены в виде 216=65536 различных значений.

Если уровней квантования мало, то возникает так называемый шум квантования. Чтобы уменьшить этот шум, в высококачественных системах оцифровки звука следует применять аналого-цифровые преобразователи с максимально доступным количеством уровней квантования.

Однако есть еще один прием, позволяющий снизить влияние шума квантования на качество звукового сигнала, который используется в цифровых системах записи звука. При использовании этого приема перед оцифровкой сигнал пропускается через нелинейный усилитель, подчеркивающий сигналы с малой амплитудой сигнала. Такое устройство усиливает слабые сигналы сильнее, чем сильные.

Это иллюстрируется графиком зависимости амплитуда выходного сигнала от амплитуды входного сигнала, показанным на рис. 2.26.

На этапе обратного преобразования оцифрованного звука в аналоговый перед выводом на звуковые колонки аналоговый сигнал снова пропускается через нелинейный усилитель. На этот раз используется другой усилитель,

который подчеркивает сигналы с большой амплитудой и имеет передаточную характеристику (зависимость амплитуда выходного сигнала от амплитуды входного сигнала), обратную той, что применялась при оцифровке.

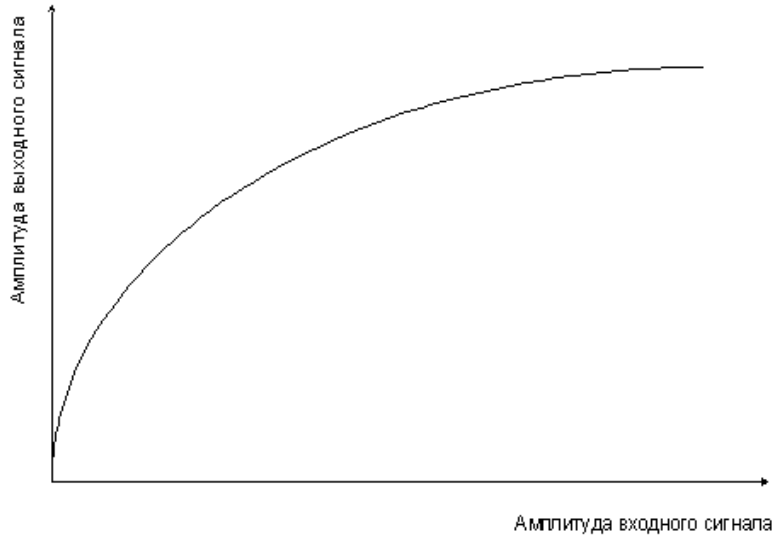


Рис. 2.26. Нелинейное усиление перед оцифровкой

Чем все это может помочь создателям систем распознавания речи?

Человек, как известно, достаточно хорошо распознает речь, произнесенную тихим шепотом или достаточно громким голосом. Можно сказать, что динамический диапазон уровней громкости успешно распознаваемой речи для человека достаточно широк.

Сегодняшние компьютерные системы распознавания речи, к сожалению, пока не могут похвастаться этим. Однако с целью некоторого расширения указанного динамического диапазона перед оцифровкой можно пропустить сигнал от микрофона через нелинейный усилитель, передаточная характеристика которого показана на рис. 2.26. Это позволит снизить уровень шума квантования при оцифровке слабых сигналов.

Разработчики систем распознавания речи, опять же, вынуждены ориентироваться в первую очередь на серийно выпускаемые звуковые

адаптеры. В них не предусмотрено описанные выше нелинейное преобразование сигнала.

Тем не менее, можно создать программный эквивалент нелинейного усилителя, преобразующего оцифрованный сигнал перед передачей его модулю распознавания речи. И хотя такой программный усилитель не сможет снизить шум квантования, с его помощью можно подчеркнуть те уровни сигнала, которые несут в себе наибольшую речевую информацию. Например, можно уменьшить амплитуду слабых сигналов, избавив таким способом сигнал от шумов.

Фильтрация верхних частот цифрового сигнала. Для сглаживания импульсов, получающихся после цифро-аналогового преобразования, на платах звуковых адаптеров имеются специальные фильтры верхних частот (рис. 2.27). Эти фильтры отрезают все частоты, находящиеся выше диапазона звуковых частот, т.е. выше 20 000 Гц.

Благодаря инерционности излучающих систем, головные телефоны и звуковые колонки тоже действуют как фильтры верхних частот. Если звуковая колонка активная и содержит внутри себя усилитель, то этот усилитель может также снабжаться фильтром верхних частот.

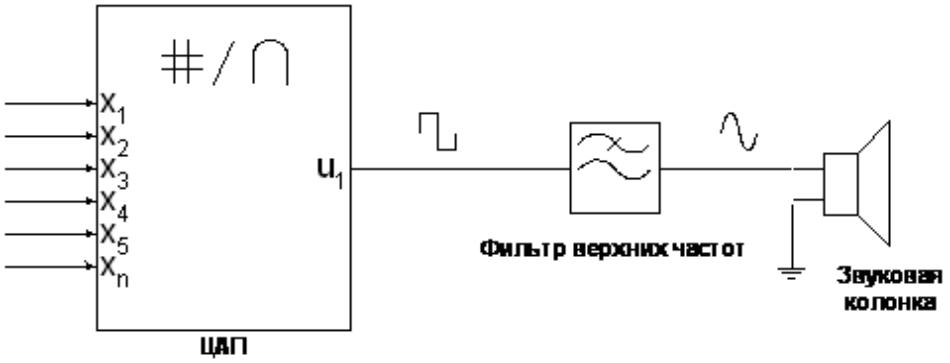


Рис. 2.27. Подключение фильтра высоких частот

На рис.2.28 показан результат работы фильтра верхних частот. Теперь прямоугольные импульсы превратились в кривую линию, форма которой приблизительно соответствует форме исходного сигнала до оцифровки.

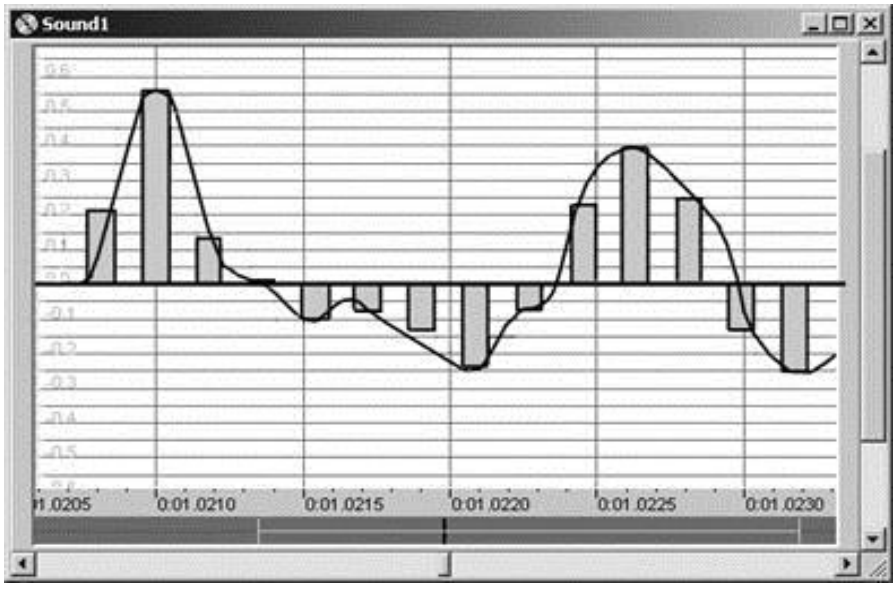


Рис.2.28. Сглаженный сигнал.

При необходимости в программе синтеза речи можно создать программный фильтр верхних частот. Однако на практике в этом нет необходимости, так как для работы вполне достаточно фильтра, предусмотренного в стандартном звуковом адаптере.

2.5. Обработка речевых сигналов

Основным подходом к проблеме распознавания речи в настоящее время является иерархический подход. Он базируется на иерархическом принципе обработки информации и на использовании многозначных решений на всех уровнях этой обработки. Опыт исследований показывает, что для достижения приемлемой для практики надежности распознавания речи требуется решение проблемных задач на всех уровнях. А это требует больших затрат и времени. Поэтому выдвигается ряд промежуточных, но важных для практики задач:

1. Распознавание отдельно произносимых слов.
2. Выделение ключевых слов в потоке речи.
3. Распознавание слитной речи, составленной из слов заданного словаря.

Оказалось, что и решение перечисленных задач для произвольного диктора или неограниченного словаря требует серьезных усилий и остается еще целый ряд принципиальных вопросов, требующих глубокой проработки.

В данном разделе ставится другая задача. Отличительной особенностью этой задачи является то, что заранее известно, какую фразу должен произнести человек. Требуется распознать лишь то, что он ее действительно правильно произнес (идентификация). То есть задача состоит в сравнении поступающего нового произнесения фразы с эталонным произнесением для проверки того, что это произнесения одной и той же фразы. Качество распознавания можно оценить по контрольным произнесениям фраз, про которые уже ясно, правильные ли они или ошибочные.

Параметрическое представление речевого сигнала.

Рассмотрим основные параметры речевого сигнала. При распознавании речевых сигналов, как правило, оперируют не с исходным речевым сигналом, а с его параметрами, вычисленными на кадре (об этом уже упоминалось):

- кратковременной энергией;
- числом нулей интенсивности;
- коэффициентами разложения в ряд Фурье;

Непосредственное осуществление преобразования Фурье требует N^2 арифметических операций. Для сокращения этого числа применяется алгоритм быстрого преобразования Фурье. Алгоритм основан на том, что при $N=2^m$ в слагаемых правой части ранее приведенных выражений преобразования Фурье (раздел 2.2) можно выделить группы, входящие в выражения различных коэффициентов A_q. Вычисляя каждую группу только один раз можно сократить число операций до $N\times\log_2 N$. Если $N\neq 2^m$, то в нашем случае можно добавить нулевые отсчеты. Разложение в ряд Фурье дает представление речевого сигнала в виде суммы гармонических колебаний с частотами $\nu(q)$. Запишем соотношение между частотой $\nu(q)$ и индексом q:

$$\nu(q) = q\nu_D/N = 2^{-m}q\nu_D \text{ при } q=0,1,...,N/2.$$

Здесь ν_D - частота дискретизации.

Значения спектра от $q = N/2 + 1$ до N-1 не содержат новой информации, т.к. значения f_l действительны. Более точно

$$A_q = \overline{A}_{N-q} \text{ при } q=0,1,...,N/2.$$

Удвоенное значение A_q - это комплексная амплитуда. Вещественные амплитуды получаются из них по формулам:

$$c_0=A_0,\ c_q=2|A_q| \text{ при } q=1,...,N/2.$$

1. Распределение энергии сигнала по частотным группам $p_1,...,p_{20}$

Одним из важнейших свойств слуха является разделение спектра звука на частотные группы. Слух может образовывать частотные группы на любом участке шкалы частот. В области частот ниже 500 Гц ширина частотных групп почти не зависит от средней частоты групп и составляет примерно 100 Гц. В области выше 500 Гц она увеличивается пропорционально средней частоте. Если частотные группы совместить в один ряд, то в диапазоне от 70 Гц до 7 кГц разместятся 20 частотных групп.

Распределение энергии по частотным группам можно найти либо непосредственно с помощью гребенки соответствующих фильтров, либо с помощью коэффициентов разложения в ряд Фурье. Значение p_i для частотной группы от частоты ν_{i-1} до ν_i с шириной $H_i=\nu_i-\nu_{i-1}$ определяется по формуле:

$$p_i = \left(\frac{1}{n}\sum_{j=0}^{n-1} c_{q_i+j}^2\right)H_i \qquad (2.15)$$

Алгоритм разделения реализации фразы на речь и паузы.

Эталонная реализация фразы не содержит кадров с паузами и поделена на слова. Поэтому алгоритм применяется только к поступающей новой реализации фразы $X^{(1)},...,X^{(t)},...,X^{(L)}$, где L - длина новой реализации фразы. Требуется отделить кадры, содержащие речь, от кадров, содержащих паузу. Звонкие звуки речи, особенно гласные, имеют высокий уровень кратковременной энергии. По этому параметру они легко отделяются от пауз.

Глухие звуки имеют низкий уровень кратковременной энергии. Однако большая часть их энергии лежит в области высоких частот, что приводит к большому числу переходов интенсивности сигнала через нуль. Это используется для отделения от пауз глухих звуков речи. Таким образом, совместное использование контуров кратковременной энергии E_t и числа нулей интенсивности Z_t позволяет точнее отделить речь от пауз. Под контуром параметра понимается последовательность значений параметра, вычисленных на каждом кадре $X^{(t)}$.

Предполагается, что первые 10 кадров не содержат речевого сигнала. По этому участку вычисляются среднее значение и дисперсия каждой из величин E_t, Z_t для определения статистических характеристик шума. Затем с учетом этих характеристик и максимальных на реализации фразы значений E_t, Z_t вычисляются пороги T_E для кратковременной энергии сигнала (P) и T_Z для числа нулей интенсивности. Экспериментально были выбраны следующие формулы:

$$T_E = M(E,10) + 3\sqrt{D(E,10)} + \frac{1}{400}\max_{1\le t\le I} E_t \le \frac{1}{25}\max_{1\le t\le I} E_t \qquad (2.16)$$

$$T_Z = M(Z,10) + 3\sqrt{D(Z,10)} + \frac{1}{20}\max_{1\le t\le I} Z_t \le \frac{1}{5}\max_{1\le t\le I} Z_t \qquad (2.17)$$

где

$$M(P,n) = \frac{1}{n}\sum_{t=1}^{n} P_t, \quad D(P,n) = \frac{1}{n-1}\sum_{t=1}^{n}\left(P_t - M(P,n)\right)^2.$$

Каждому кадру $X^{(t)}$ мы должны поставить в соответствие двоичный признак b_t, равный 1, если кадр содержит речь, и 0 - в противном случае. Сначала отмечают единицами кадры, на которых кратковременная энергия $E_t \ge T_E$, и нулями - остальные кадры. Полученные отметки сглаживают медианной фильтрацией с окном шириной в $2h+1$ кадров. (Например, h=3.) Признаки b_t могут принимать всего два значения. Поэтому

фильтрация сводится к тому, что последовательно для $t=h+1,...,L-h$ значение b_t заменяется на единицу, если $\sum_{i=t-h}^{t+h} b_i > h$. В противном случае значение b_t заменяется на ноль.

В результате выделяются непрерывные участки, содержащие речь. Далее каждый такой участок пытаются расширить. Пусть, например, участок начинается с кадра $X^{(N1)}$ и заканчивается на кадре $X^{(N2)}$. Перемещаются влево от $X^{(N1)}$ (вправо от $X^{(N2)}$) и сравнивают число нулей интенсивности Z_t с порогом T_Z. Это перемещение не должно превышать 20 кадров слева от $X^{(N1)}$ (справа от $X^{(N2)}$). Если Z_t превысило порог в три и более раз, то начало речевого участка переносится туда, где Z_t впервые превышает порог. В противном случае началом участка считается кадр $X^{(N1)}$. Аналогично поступают и с $X^{(N2)}$. Если два участка перекрываются, то их объединяют в один. Таким образом, окончательно выделяются непрерывные участки, содержащие речь. Такие участки будем называть реализациями слов. Приведенный алгоритм позволяет перейти от сравнения реализаций фраз к сравнению реализаций слов.

Нелинейный метод временной нормализации.

Реализация слова, в отличие от реализации фразы, не содержит кадров с паузами. Пусть даны две реализации слова:

$$X^{(0)},...,X^{(i)},...,X^{(m)} \text{ и } Y^{(0)},...,Y^{(j)},...,Y^{(n)} .$$

Первая реализация слова считается эталонной, хранимой в памяти, вторая - входной.

Прежде чем сравнивать их между собой необходимо провести временную нормализацию, т.е. привести реализации слов к одинаковой длине. Линейное сжатие или растяжение одной реализации слова до величины другой не решает вопрос вследствие одного важного свойства речевого сигнала - неравномерности его протекания во времени. Это свойство речи выражается в трудно контролируемой зависимости времени образования и звучания ее элементов от контекста, темпа, диалектных и

индивидуальных особенностей диктора. Поэтому сравнение должно опираться на нелинейную временную нормализацию.

Для этого находится деформирующая функция, применение которой минимизирует расхождение между эталонной и новой реализациями слов. Точнее находятся две функции:

$$\omega_X : \{1,...,l\} \rightarrow \{1,...,m\}$$

$$\omega_Y : \{1,...,l\} \rightarrow \{1,...,n\}$$

$$(\max\{m,n\} \le l < m+n)$$

такие, что

$$\omega_X(1)=1,\ \omega_Y(1)=1,\ \omega_X(l)=m,\ \omega_Y(l)=n,$$

$$\omega_X(i+1)=\omega_X(i) \text{ или} \omega_X(i)+1 \ \forall\, i=1,...,m-1$$

$$\omega_Y(j+1)=\omega_Y(j) \text{ или} \omega_Y(j)+1 \ \forall\, j=1,...,n-1$$

и, кроме того, $\sum_{k=1}^{l} \rho_{\omega_X(k),\omega_Y(k)}$ минимальна.

Здесь $\rho_{i,j}=(S_X(i)-S_Y(j))^2$, где $S_X(i)$, $S_Y(j)$ - значения сегментирующей функции из соответствующих контуров.

Сегментирующая функция должна характеризовать суммарное изменение используемых ею параметров речевого сигнала и зависит от двух кадров: текущего и предыдущего. В качестве параметров речевого сигнала мы будем использовать распределение энергии сигнала по частотным группам.

Опишем процедуру нахождения контура сегментирующей функции $S_X(1),...,S_X(i),...,S_X(m)$ для эталонной реализации слова.

1. На каждом кадре $X^{(i)}$ находится распределение энергии сигнала по частотным группам: $p_1^{(i)},...,p_{20}^{(i)}$; $i = 0,1,...,m$;
2. Вычисляются модули конечных разностей:

$$\Delta_k^{(i)}=| p_k^{i}-p_k^{i-1} | ;\ i =1,...,m;\ k =1,...,20;$$

3. Вычисляются средние разности:

$$\Delta_k^{(i)} = \left| p_k^{i} - p_k^{i-1} \right| ;\ k =1,...,20;$$

4. Вычисляются средневзвешенные разности:

$$\overline{\Delta}_k = \frac{1}{m}\sum_{i=1}^{m}\Delta_k^{(i)} \quad ; i = 1,...,m;\ k = 1,...,20;$$

5. Контур сегментирующей функции S_X :

$$\delta_k^{(i)} = \frac{\Delta_k^{(i)}}{\overline{\Delta}_k} \quad ; i = 1,...,m.$$

Аналогично находится контур сегментирующей функции $S_Y(1),...,S_Y(j),...,S_Y(n)$ для новой реализации слова.

Процедура нахождения деформирующих функций ω_X, ω_Y реализуется методом динамического программирования и дает возможность произвести внутреннее нелинейное выравнивание реализаций слов по времени.

Сначала строится матрица расстояний $R = \{ \rho_{i,j} \}$ размера ($m \times n$). По ней затем вычисляется матрица $D = \{ d_{i,j} \}$ такого же размера ($m \times n$):

1. $d_{m,n} = \rho_{m,n}$;
2. $d_{i,n} = \rho_{i,n} + d_{i+1,n}$, $i = m\text{-}1,...,1$;
3. $d_{m,j} = \rho_{m,j} + d_{m,j+1}$, $j = n\text{-}1,...,1$;
4. $d_{i,j} = \rho_{i,j} + \min\{ d_{i+1,j+1} , d_{i+1,j} , d_{i,j+1} \}$, $i = m\text{-}1,...,1$; $j = n\text{-}1,...,1$.

Матрица D в свою очередь используется для нахождения функций ω_X, ω_Y. Сначала присваивают: $\omega_X(1)=1$, $\omega_Y(1)=1$. Далее на k-ом шаге находят $\omega_X(k+1)$ и $\omega_Y(k+1)$. Возможны четыре случая:

1. Если $\omega_X(k)=m$ и $\omega_Y(k)=n$, то деформирующие функции найдены;
2. Если $\omega_X(k)=m$, а $\omega_Y(k)<n$, то присваивают: $\omega_X(k+1)=m$, $\omega_Y(k+1)=\omega_Y(k)+1$;
3. Если $\omega_X(k)<m$, но $\omega_Y(k)=n$, то присваивают: $\omega_X(k+1)=\omega_X(k)+1$, $\omega_Y(k+1)=n$;
4. Если $\omega_X(k)<m$ и $\omega_Y(k)<n$, то сравниваются $d_{i1,\,j1}$, $d_{i2,\,j2}$, $d_{i3,\,j3}$ для нахождения среди нихминимальногоисоответствующих$i_{\min}$, $j_{\min}$.

Здесь$i_1=i_2=\omega_X(k)+1$,$i_1=\omega_X(k)$,$j_1=j_3=\omega_Y(k)+1$,$j_2=\omega_Y(k)$.

Затем присваивают: $\omega_X(k+1)=i_{\min}$, $\omega_Y(k+1)=j_{\min}$.

Зная деформирующие функции ω_X , ω_Y мы можем для любого участка эталонной реализации слова найти соответствующий ему участок новой реализации. Применим это для разделения новой реализации слова на звуковые диады. Звуковая диада - переходный процесс от фонемы к фонеме,

отображающий перестройку артикуляционного аппарата. В отличие от реализаций фонемы, реализации звуковой диады значительно меньше подвержены влиянию контекста и отражают взаимосвязь соседних фонем речевого потока. Границами диад являются центры квазистационарных участков фонем. Таким образом, диада состоит из второй половины первой фонемы и первой половины второй фонемы.

Эталонная реализация слова делится на звуковые диады вручную: отмечаются номера $a_0,...,a_L$ кадров, являющихся центрами квазистационарных участков фонем. Затем выбираются точки n_l , $l = 0,...,L$ такие, что $\omega_X(n_l)=a_l$. Теперь с помощью функции ω_Y можно определить номера $b_0,...,b_L$ кадров, являющихся центрами квазистационарных участков фонем в новой реализации слова: $b_l=\omega_Y(n_l)$, $l = 0,...,L$. Приведенный алгоритм позволяет перейти от сравнения реализаций слов к сравнению реализаций звуковых диад.

Сравнение двух реализаций слов.

Пусть задана эталонная реализация слова: $X^{(0)},...,X^{(i)},...,X^{(m)}$ и получена некая новая реализация слова: $Y^{(0)},...,Y^{(j)},...,Y^{(n)}$. Требуется сравнить их и определить, являются ли они реализациями одного и того же слова. Будем считать, что мы уже провели временную нормализацию и нашли деформирующие функции ω_X , ω_Y. Пусть также известны границы звуковых диад $n_0,n_1,...,n_L$. Сравнение новой реализации слова с эталоном заключается в сравнении звуковых диад из новой реализации с соответствующими диадами из эталонной реализации. При сравнении соответствующих диад с номером l вычисляется расстояние R_l между ними и сравнивается с порогом $T_{\varphi l}$. Здесь φ_l - код типа звуковых диад, известный заранее. Если $R_l \le T_{\varphi l}$, то считается что, диада номер l в новой реализации произнесена правильно. Если все диады правильно произнесены, то новая реализация слова считается правильной.

Расстояние R_l вычисляется, например, так:

$$R_l = \sum_{u=n_{l-1}}^{n_l} \rho_{\omega x(u),\omega \tau(u)},$$

где $\rho_{i,j}$ было определено в описании алгоритма временной нормализации.

2.6. Алгоритм обработки звуковых сигналов на основе спектральных функций

В современных средствах обработки аудио и речевых сигналов широко применяются специализированные алгоритмы обработки, такие как анализ и синтез звука, сжатие и компрессия звуковых сигналов.

Разнообразие аппаратуры кодирования, хранения, обработки и воспроизведения аудиоинформации усложняет процесс согласования форматов приема/передачи. Особенно остро стоит проблема изменения частоты дискретизации при интеграции средств аудио обработки. Одним из способов согласования частот дискретизации является применение численных методов, обеспечивающих быстрое и, по возможности, произвольное изменение частоты дискретизации обрабатываемого сигнала.

С точки зрения теории цифровой обработки сигналов наиболее правильно разложить сигнал в ряд Фурье по количеству имеющихся отсчетов, после чего реализовать алгоритм обратного преобразования с помощью функции sin(x)/x с нужной частотой. Однако такой подход требует больших вычислительных ресурсов, очень громоздок и трудно реализуется в режиме реального времени. Более целесообразно применение линейной, квадратичной или кубической аппроксимации 2-го и 3-го порядка для получения недостающих или формирования дополнительных значений сигнала.

Время обработки одной порции информации и визуализации результатов должно быть соизмеримо с чтением следующей порции в реальном масштабе времени, т.е. пока одна порция считывается, другая

должна быть готова для отображения, тогда весь процесс обработки идет в реальном масштабе времени.

Звуковые сигналы имеют большую длительность и их обработка (сжатие, изменение частоты дискретизации, фильтрация) требует скоростных алгоритмов как при чтении с накопителей, так и при работе с входными каналами. Звуковые карты на базе сигнальных процессоров работают, как правило, при частоте дискретизации 48 кГц, поэтому информация с накопителей и другие аудиопотоки должны быть приведены к этой частоте (например, дисковые накопители имеют частоту записи звука в 44 кГц).

Для обработки с целью изменения частоты дискретизации можно использовать аналитические выражения, связывающие коэффициенты алгебраических полиномов A_k со спектральными коэффициентами входного сигнала.

Для решения такой задачи выбраны три рассмотренных ранее базисные системы: вейвлет-функции V, система пилообразных функций (P-базис) и базис Адамара *W*. Эти системы обладают как локальными (V,W), так и интегральными свойствами (P).

Общий алгоритм обработки сигнала содержит следующие процедуры:

- спектральное преобразования в базисах V_i, P_i или W_i;
- получение значений промежуточных переменных;
- вычисление значений аппроксимирующих полиномов A_k;
- воспроизведение сглаженных значений *f(t)*.

Для упрощения вычислительного алгоритма предлагается метод, основанный на перемножении спектров входного сигнала и классического полинома в двоично-ортогональных базисах.

Для представления сигналов или их фрагментов в полиномиальной форме достаточно выбрать k=1,2,3. Это существенно упрощает обработку и не нарушает традиционных подходов к решению подобных задач.

Обозначим спектральные коэффициенты выбранных для реализации базисных систем: Р-базис - b_i, преобразование Адамара-Пэли - a_i, вейвлет-функции - υ_i.

Аппроксимирующие структуры в базисе *W* и *P*. Искомый классический многочлен можно представить в виде:

$$f(u) = c_0 T_0(u) + c_1 T_1(u) + ... + c_k T_k(u) \qquad (2.18)$$

$$u = \frac{x - \bar{x}}{h}, \quad \bar{x} = \frac{x_1 + x_N}{2}, \quad h = x_{i+1} - x_i, \quad x \in [0,1],$$

(i=0, 1, 2,…,N-1)

где $T_k(u)$ - многочлены Чебышева для равностоящих значений аргумента; c_k - коэффициенты многочлена, которые вычисляются по формулам:

$$c_k = \frac{1}{H_k} \sum_{i=0}^{N-1} f(u_i) T_k(u_i) \quad (k = 0,1,2,...,n;\ n < N) \qquad (2.19)$$

Многочлены Чебышева выбраны в связи тем, что они имеют вид степенного полинома.

Обеспечивая свойства ортогональности можно вычислить коэффициенты многочлена Чебышева c_k. Обычно в литературе свойство ортогональности обеспечивается в интервале $x \in [-1,1]$. Решение прикладных задач обработки сигналов на таком интервале усложняет проблему, поэтому удобнее использовать интервал $x \in [0,1]$. Эта проблема решается традиционным способом.

Рассмотрим сам метод получения аппроксимирующего выражения для базисной системы *W*. Равенство (2.19) определяет взаимную мощность входного сигнала $f(u_i)$ и соответствующего многочлена Чебышева $T_k(u_i)$. Если к (2.19) применить общий случай равенства Парсеваля и приравнять к ним мощности их спектров, то можно получить:

$$\frac{1}{H_k}\sum_{i=0}^{N-1} f(u_i)T_k(u_i) = \frac{N}{H_k}\sum_{i=0}^{N-1} a_i \tau_i^k$$

где τ_i^k – спектральные коэффициенты многочлены Чебышева в выбранном базисе *W*, a_i – спектральные коэффициенты входного сигнала тоже в базисе *W*. С учетом (2.19) получается формула для вычисления параметров c_k через произведение спектральных коэффициентов сигнала и классического полинома сигнала в базисе *W*:

$$c_k = \frac{N}{H_k}\sum_{i=0}^{N-1} a_i \tau_i^k \qquad (2.20)$$

В этих формулах величины H_k вычисляются по следующим выражениям:

$$H_k = \sum_{i=1}^{N} T_k^2(u_i),\ j = 1,2,3,\ldots \qquad (2.21)$$

или

$$H_k = \frac{(k)^2 (N+k)(N+k-1)\ldots(N-k)}{4^k \left[(2k-1)!!\right]^2 (2k+1)},\ k = 1,2,3,\ldots \qquad (2.22)$$

Например,

$$H_0 = N,\ H_1 = \frac{N(N^2-1)}{12},\ H_2 = \frac{N(N^2-1)(N^2-4)}{180},\ H_3 = H_2 \cdot \frac{(N^2-9)\cdot 9}{140}\ldots$$

Примеры многочленов Чебышева $T_k(u)$ для $k \leq 3$ приведены в:

$$T_0(u) = 1,\ T_1(u) = u,\ T_2(u) = u^2 - 5\frac{1}{4},\ T_3(u) = u^3 - 9\frac{1}{4}u$$

где $u = \frac{x - \bar{x}}{h}$, $\bar{x} = \frac{x_1 + x_N}{2}$, $h = x_{k+1} - x_k$, x=[0, 1] с шагом $\frac{1}{N-1}$.

При k=2 и N=8 подставляя полученные члены в формулу (2.18) можно получить:

$$f(u) = c_0 - \frac{21}{4} c_2 + c_1 u + c_2 u^2 \qquad (2.23)$$

Вводим следующие обозначения: $A_2 = c_2$, $A_1 = c_1$, $A_0 = c_0 - \frac{21}{4} A_2$ (2.24)

При k=3 и N=8 подставляя полученные члены в формулу (2.18) можно получить:

$$f(u) = \left(c_0 - \frac{21}{4} c_2 \right) + \left(c_1 - \frac{37}{4} c_3 \right) u + c_2 u^2 + c_3 u^3 \qquad (2.25)$$

Для упрощения вводим обозначения:

$$A_3 = c_3,\ A_2 = c_2,\ A_1 = c_1 - \frac{37}{4} A_3,\ A_0 = c_0 - \frac{21}{4} A_2 \qquad (2.26)$$

С использованием (2.32) выражение (2.31) напишем в следующем виде:

$$f(u) = A_0 + A_1 u + A_2 u^2 + A_3 u^3 \qquad (2.27)$$

Далее можно перейти к представлению сигналов в виде аппроксимирующих полиномов используя разложение $T_k(u)$ в базисе W.

Исходная матрица значений многочленов в интервале $x \in [0,1]$ имеет вид:

$$\begin{pmatrix} p_0(u_i) \\ p_1(u_i) \\ p_2(u_i) \\ p_3(u_i) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -7/2 & -5/2 & -3/2 & -1/2 & 1/2 & 3/2 & 5/2 & 7/2 \\ 7 & 1 & -3 & -5 & -5 & -3 & 1 & 7 \\ -21/2 & 15/2 & 21/2 & 9/2 & -9/2 & -21/2 & -15/2 & 21/2 \end{pmatrix}$$

После разложение многочленов $T_k(u)$ по базису W определяются спектральные коэффициенты τ_i^k и в результате получается матрица:

$$\begin{pmatrix} \tau_i^0 \\ \tau_i^1 \\ \tau_i^2 \\ \tau_i^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & -1 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 2 & 1 & 0 \\ 0 & 3 & -9/2 & 0 & -3 & 0 & 0 & -6 \end{pmatrix}$$

Чтобы вычислить c_k в выражении (2.19) нужно определить H_k с использованием формулы (2.20). При N=8: $H_0 = 8$, $H_1 = 42$, $H_2 = 168$, $H_3 = 594$.

Поставив найденные значения τ_i^k и H_k в формулу (2.19) можно получить следующие зависимости:

$$\begin{aligned} c_0 &= a_0 \\ c_1 &= \frac{4}{21}(-2a_1 - a_2 - \frac{1}{2}a_4) \\ c_2 &= \frac{1}{21}(4a_3 + 2a_5 + a_6) \\ c_3 &= \frac{4}{99}(a_1 - \frac{3}{2}a_2 - a_4 - 2a_7) \end{aligned} \qquad (2.28)$$

Поставив c_k в формулу (2.22) и (2.23) и можно получить аналитические выражения, связывающие коэффициенты A_k со спектральными коэффициентами a_i входного сигнала:

При k=2:

$$A_2 = \frac{1}{21}(4a_3 + 2a_5 + a_6), \quad A_1 = \frac{4}{21}(-2a_1 - a_2 - \frac{1}{2}a_4), \quad A_0 = a_0 - \frac{21}{4}A_2 \qquad (2.29)$$

При k=3:

$$A_3 = \frac{2}{99}(2a_1 - 3a_2 - 2a_4 - 4a_7), \quad A_2 = \frac{1}{21}(4a_3 + 2a_5 + a_6)$$
$$A_1 = -\frac{2}{21}(4a_1 + 2a_2 + a_4) - \frac{37}{4}A_3, \quad A_0 = a_0 - \frac{21}{4}A_2 \qquad (2.30)$$

Аппроксимирующие структуры вида (2.29) и (2.30) дают непосредственную связь коэффициентов полинома со спектральными коэффициентами разложения сигнала в базисе *W*.

Аналогичным способом можно получить аппроксимирующие структуры в виде полиномов для базисной системы *P*.

Выражение (2.19) имеет вид:

$$c_k = \frac{N}{H_k}\sum_{i=0}^{N-1} b_i \tau_i^k, \qquad (2.31)$$

где b_i - спектральные коэффициенты разложения сигнала *f(t)* в базисе P_i.

Определение коэффициентов разложения $T_k(x)$ по базису P_i выполняется в виде:

$$\begin{pmatrix} \tau_i^0 \\ \tau_i^1 \\ \tau_i^2 \\ \tau_i^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -293/128 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 143/32 & 0 & 0 & 0 \\ 0 & 0 & 0 & -43/64 & 0 & -393/64 & -6 & 0 \end{pmatrix}$$

После преобразований получаются аппроксимирующие структуры:
Для случая k=2, N=8:

$$A_2 = \frac{1}{21}(b_2 + \frac{9}{2}b_4), \; A_1 = -\frac{37}{84}b_1, \; A_0 = b_0 - \frac{21}{4}A_2 \qquad (2.32)$$

Для случая k=3, N=8:

$$A_3 = -\frac{1}{12}b_5 - \frac{2}{25}b_6,\ A_2 = \frac{1}{21}(b_2 + \frac{9}{2}b_4),\ A_1 = -\frac{37}{4}(\frac{1}{21}b_1 + A_3),\ A_0 = b_0 - \frac{21}{4}A_2 \qquad (2.33)$$

Таким образом, в обеих интегральных базисных системах величины A_k для текущих N значений сигнала *f(t)* получаются в виде алгебраической модели, где значения A_k вычисляются в виде суммы произведений (свертки) спектральных коэффициентов разложения сигнала и классических полиномов в выбранной базисной системе.

Аппроксимирующие структуры в вейвлет-базисе. Метод перехода от спектра сигнала к его полиномиальной аппроксимирующей структуре аналогичен предыдущему методу.

Выражение (2.19) имеет вид:

$$c_k = \frac{N}{H_k}\sum_{i=0}^{N-1}\upsilon_i\tau_i^k\ , \qquad (2.34)$$

где υ_i - спектральные коэффициенты разложения сигнала *f(t)* в базисе V_i.

Далее выполняется разложение $T_k(x)$ в базисе V_i.

Исходная матрица значений многочленов $T_k(x)$ в интервале $x \in [0,1]$ и N=8 дана в предыдущем разделе.

В результате разложения многочленов $T_k(x)$ по базису V_i получается набор коэффициентов разложения:

$$\begin{pmatrix}\tau_i^0\\ \tau_i^1\\ \tau_i^2\\ \tau_i^3\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & -2 & -1 & -1 & -1/2 & -1/2 & -1/2 & -1/2\\ 0 & 0 & 4 & -4 & 3 & 1 & -1 & -3\\ 0 & 3 & -9/2 & -9/2 & -9 & 3 & 3 & -9\end{pmatrix}$$

Подставив найденные значение H_k и τ_i^k в формулу (2.34) получаются значения c_0, c_1, c_2 и c_3 как функции от произведений υ_i и τ_i^k. Произведя подстановку вместо c_k коэффициентов A_k и упростив выражения можно получить алгебраическую модель аппроксимирующей структуры:

Для случая k=2, N=8:

$$\begin{aligned} A_2 &= \frac{1}{84}\left[\upsilon_5 - \upsilon_6 + 8(\upsilon_2 - \upsilon_3) + 3(\upsilon_4 - \upsilon_7)\right] \\ A_1 &= -\frac{1}{42}\left[\upsilon_4 + \upsilon_5 + \upsilon_6 + \upsilon_7 + 4(4\upsilon_1 + \upsilon_2 + \upsilon_3)\right] \\ A_0 &= \upsilon_0 - \frac{21}{4}A_2 \end{aligned} \qquad (2.35)$$

Для случая k=3, N=8:

$$\begin{aligned} A_3 &= \frac{1}{99}\left[\upsilon_5 + \upsilon_6 + 4\upsilon_1 - 3(\upsilon_2 + \upsilon_3 + \upsilon_4 + \upsilon_7)\right] \\ A_2 &= \frac{1}{84}\left[\upsilon_5 - \upsilon_6 + 8(\upsilon_2 - \upsilon_3) + 3(\upsilon_4 - \upsilon_7)\right] \\ A_1 &= -\frac{1}{42}\left[\upsilon_4 + \upsilon_5 + \upsilon_6 + \upsilon_7 + 4(4\upsilon_1 + \upsilon_2 + \upsilon_3)\right] - \frac{37}{4}A_3 \\ A_0 &= \upsilon_0 - \frac{21}{4}A_2 \end{aligned} \qquad (2.36)$$

Данный метод дает возможность с помощью программ сигнальных процессоров одновременно решать несколько задач обработки:

- изменять в широких пределах частоту дискретизации (при отсчетах на фрагмент, кратных степени 2);
- осуществлять сглаживание пульсации в аудиосигналах;
- обеспечивать сжатие за счет хранения вместо текущих отсчетов сигнала значения коэффициентов полинома A_k на участках обработки.

На рис.2.29 представлена цифровая запись звукового сигнала в объеме 256 отсчетов. Длительность звучания данного отрезка звукового сигнала составляет (при частоте дискретизации 48 кГц) 5,1 мс.

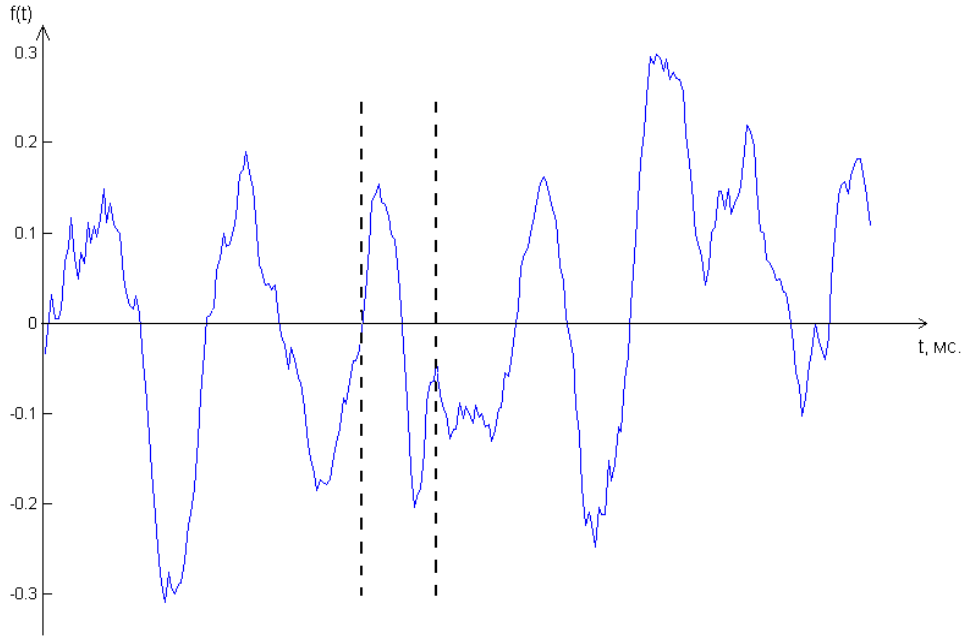


Рис.2.29. Фрагмент звукового сигнала, длительностью 5,1 мс.

В процессе исследований были разработаны прикладные программы для сжатия и сглаживания аудиосигналов, результаты которых представлены на рис.2.30.

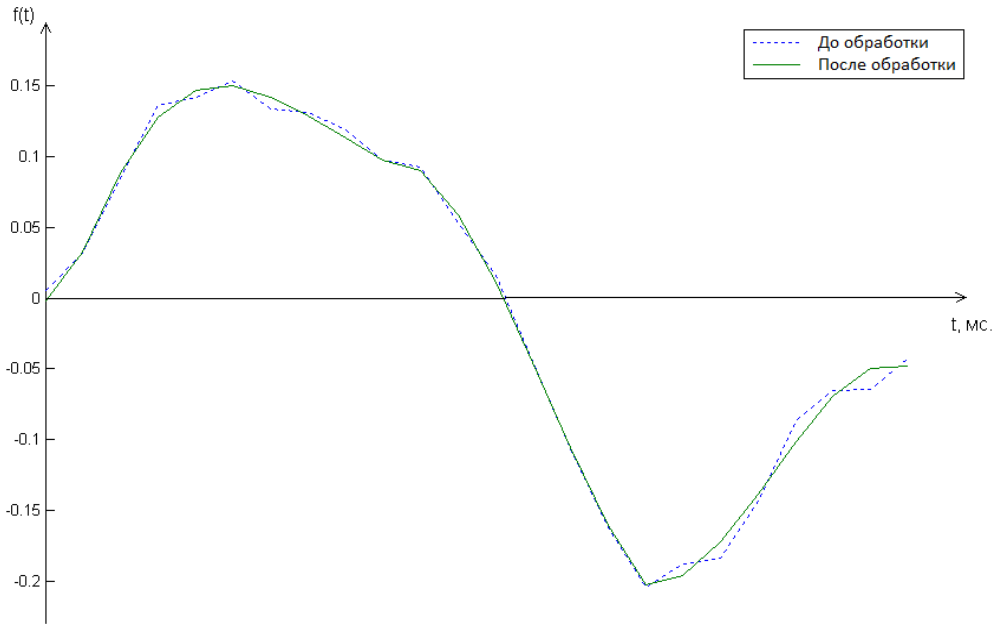


Рис.2.30. Результаты обработки выделенного фрагмента звукового сигнала.

На данном отрезке звучания для реализации предлагаемого метода выбран фрагмент (выделен пунктирными линиями), состоящий из трех участков по 8 отсчетов сигнала в каждом. Для каждого из участков применен предлагаемый алгоритм обработки. В таблице 2.1 приведены вычисленные коэффициенты аппроксимирующих полиномов на каждом из участков, используя метод свертки спектров на основе выражений (2.33), а также соответствующие погрешности замены звукового сигнала его алгебраической моделью. Аппроксимирующие структуры вида (2.33) позволяют изменять число отсчетов восстановления аудиосигнала независимо от частоты дискретизации на входе процессора обработки, при этом аргументу придаются произвольные приращения в пределах [0,1].

Таблица 2.1. Коэффициенты аппроксимирующих полиномов

Коэффициенты	1-участок	2-участок	3-участок
A_0	-0.0013	-0.0004	0.0000
A_1	-0.0044	-0.0049	0.0109
A_2	0.0389	-0.0291	-0.0062
A_3	0.1053	0.0703	-0.1972
$\Delta_{ср.кв.}$ (%)	0,55	0,29	0,96

Сигнальные процессоры современных звуковых карт по своим возможностям (длина слова, тактовая частота, объем встроенной памяти) приближаются к параметру автономных сигнальных процессоров для скоростных приложений. И те и другие обеспечивают высококачественное воспроизведение звуковых сигналов от различных источников: CD и DVD-проигрывателей, музыкальных центров, компьютеров. Многие сетевые карты имеют возможность интерполяции сигналов, но вопрос произвольного

изменения частоты дискретизации в сторону увеличения или уменьшения остается трудной проблемой.

Алгоритм обработки имеет сглаживающие свойства. Используя эти свойства можно сглаживать аудио сигналы от фоновых помех. Для примера берем фрагмент звукового сигнала, длительностью 5,1 мс (рис.2.29). На рис.2.30 приведены результаты сглаживания выделенного фрагмента звукового сигнала.

Эксперименты показали, что разработанные алгоритмы дают эффективные результаты при изменении частоты дискретизации аудиосигналов, исполненных на нескольких музыкальных инструментах. Очень хорошие результаты показала обработка аудиосигналов, исполненных национальными музыкальными инструментами, например, гижжак, танбур, най.

На рис.2.31 показана цифровая запись звуков, полученных от одного музыкального инструмента (на примере пианино) в объеме 128 отсчетов, на рис.2.32 представлены результаты обработки при k=3 (среднеквадратическое отклонение составило от $8{,}5*10^{-5}$ до $1{,}7*10^{-4}$).

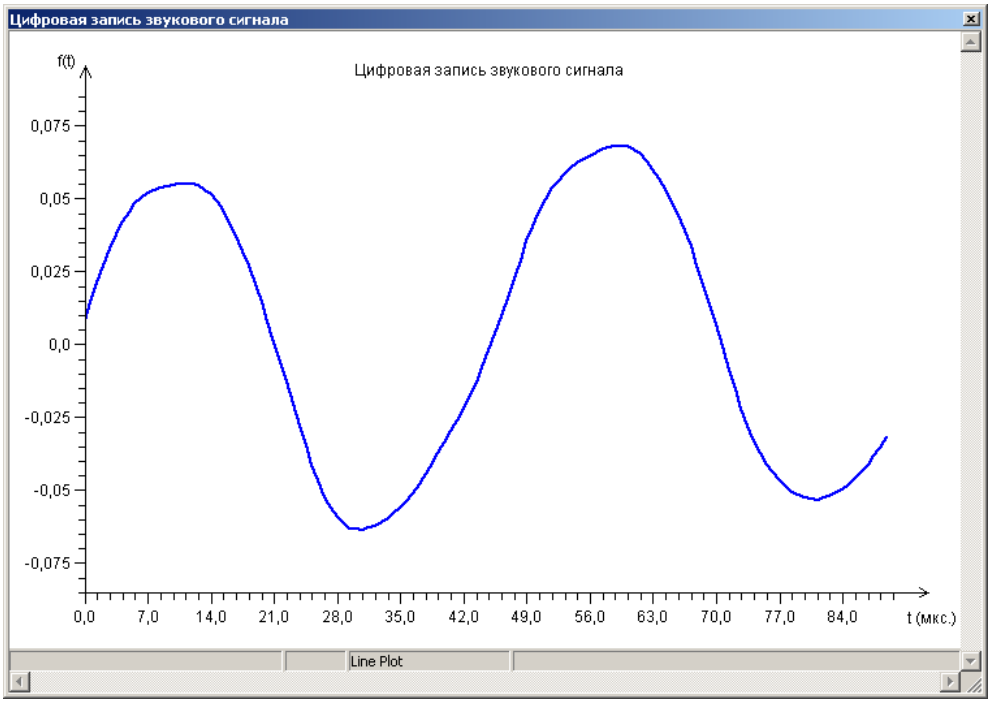


Рис.2.31. Цифровая запись звукового сигнала

В данном примере применение аппроксимирующей структуры полученной с помощью метода свертки спектров (формула (2.33)) эффективно решает задачу «передискретизации» с целью улучшения качества аудиосигнала. Использование предлагаемого алгоритма дает возможность перехода между стандартными частотами 22 кГц, 44.1 кГц, 48 кГц, 96 кГц, 128 кГц, применяемыми в устройствах хранения и воспроизведения звука.

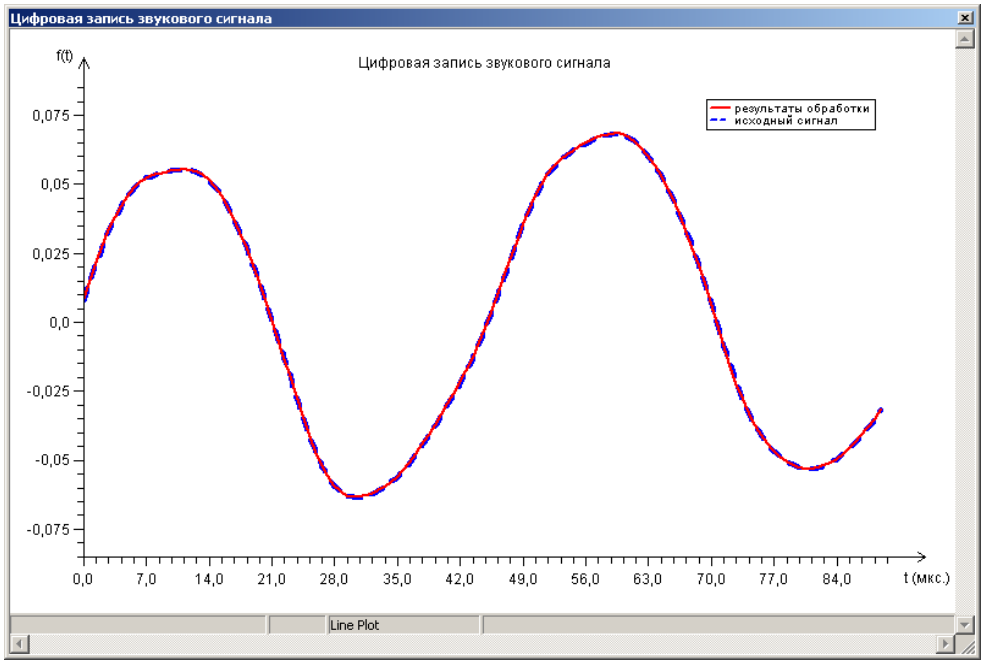


Рис.2.32. Результаты обработки при k=3

Рассмотрим возможность применения аппроксимирующих структур для решения задачи компактного представление (сжатие) аудиосигналов за счет хранения коэффициентов полинома.

Для примера выбран звуковой (wav) файл с общей длительностью 10 сек. Результаты применения метода свертки спектров, основанный W-базиса приведено в таблице 2.2.

Файлы для равнения представлены в различных форматах используемых в аппаратуре: wav, mp3, em1, smp. Формат mp3 взят в версии Constant Bitrate. Исходный файл представлен в формате wav.

Затем он переведен для сравнительного анализа в другие указанные форматы.

Таблица 2.2. Результаты компрессии аудио сигнала

Форматы	До обработки				Результаты обработки
	wav (исходный файл)	mp3 (Constant Bitrate)	em1	smp	
Размер файла	216 КБ	160 КБ	128 КБ	676 КБ	108 КБ

Сравнительный анализ используемых форматов и предлагаемого алгоритма при одинаковом размере исходного файла показал, что предлагаемый алгоритм обеспечивает более высокий коэффициент компрессии, чем используемые в реальной аппаратуре методы сжатия.

Контрольные вопросы

1. Как формируются цифровые речевые сигналы?
2. Какова разрядность АЦП в речевом предпроцессоре?
3. Объясните основную суть теоремы Котельникова.
4. Как происходить преобразование сигнала в цифровую форму?
5. Перечислите ключевые операции цифровой обработки сигналов?
6. Какова роль процессора ЦОС?
7. Приведите примеры используемых частот дискретизации звука.
8. Перечислите основные параметры речевого сигнала.
9. Как вычисляется кратковременная энергия речевого сигнала?
10. Как определяется число нулей интенсивности?
11. Как определяется коэффициенты разложения в ряд Фурье?

12. Для чего используется распределение энергии сигнала по частотным группам?
13. Объясните основные этапы предварительной обработки речевых сигналов.
14. Как происходить сегментация речевых сигналов?
15. Объясните Фурье – преобразование речевого сигнала с использованием окна Хэмминга.
16. Что дает спектральный анализ речевого сигнала?
17. Назовите шесть спектральных параметров, которые оцениваются на этапе анализа речевого сообщения.
18. Как измеряется частота основного тона?
19. Объясните основные возможности частотных фильтров.
20. Каковы преимущества алгоритма обработки звуковых сигналов на основе спектральных функций?

ГЛАВА 3. МЕТОДЫ И АЛГОРИТМЫ РАСПОЗНАВАНИЯ РЕЧИ

3.1. Основные этапы распознавания речи

Архитектуру системы распознавания речи выбирают исходя из самой сложности решаемой задачи. Упростить задачу выбора структуры системы позволяет ее предварительная классификация по ряду признаков, наборам которых могут быть поставлены в соответствие стандартные решения.

В качестве признаков используются: тип речи, распознаваемый системой; зависимость системы от распознаваемых голосов дикторов; степень детализации эталонов; количество распознаваемых слов. В некоторых работах вводится понятие полноты словаря, а задачи поиска ключевых слов интерпретируются, как распознавание с неполным словарем. В других предлагаются иные классификационные признаки: назначение системы, ее потребительские свойства и механизмы функционирования.

Рассмотрим вариант классификации речевых систем, которые могут найти применение в повседневной деятельности человека. Варианты систем учитывают потребительские качества, объем словаря, тип речи основные сферы назначения (рис.3.1).

Речевой сигнал – это модель сложного динамического процесса, при его анализе необходимо оперировать несколькими показателями (параметрами), характеризующими сигнал или его фрагмент. Такими основными характеристиками речевых сигналов являются: формантные частоты, частота основного тона, спектральный состав. В конкретной области применения в качестве эталона для правильного распознавания целесообразно составлять словарь основных слов, объем которого может составлять 1000 и более.

Одной из форм речи, наиболее широко применяемой и относительно простой для анализа, являются изолированные слова. Часто используются допустимые последовательности слов со строгими ограничениями на словарь. В настоящее время даже в наиболее сложных системах анализа

слитной речи элементом обработки является выделение ключевых слов в общем потоке.

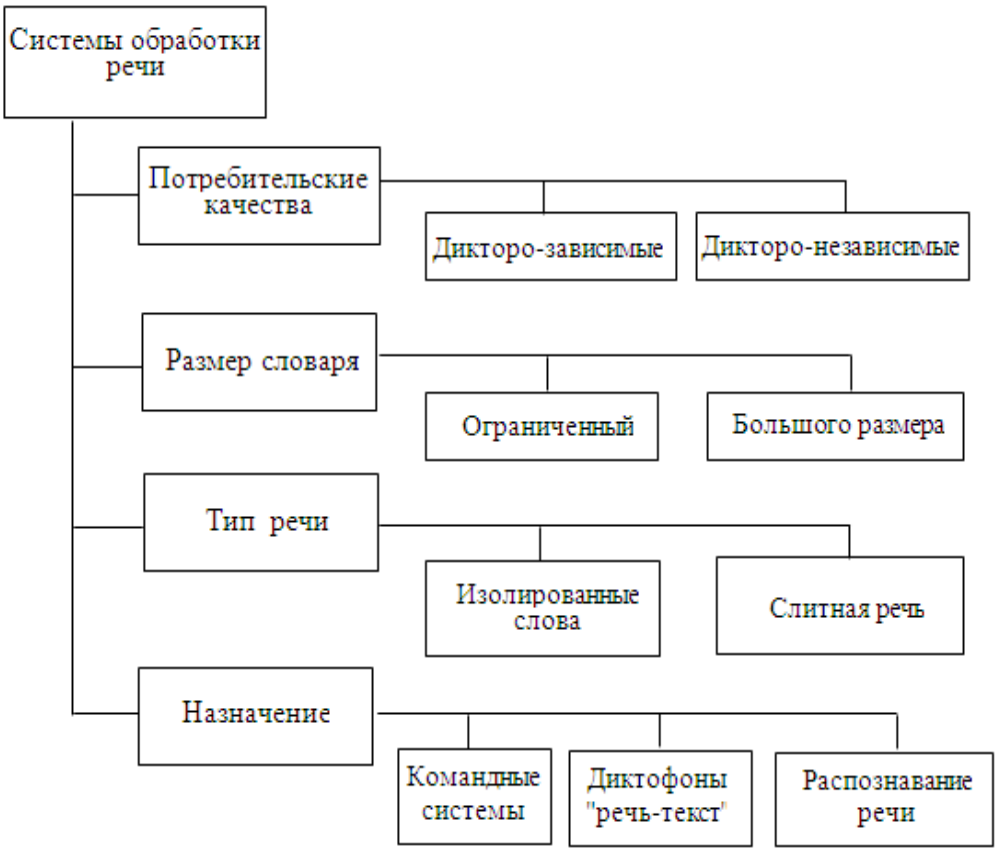


Рис.3.1. Классификация систем обработки речи

В системах автоматического распознавания речи как правило выделяются три основных этапа: выделение признаков, обучение и распознавание. На первом этапе из исходного сигнала получают вектор признаков – сжатое представление речевого сигнала, в котором присутствуют только необходимые для распознавания параметры. Опыт показывает, что основными этапами анализа речи и выделения признаков могут быть:

- акустический;
- фонетический;
- лексический;
- синтаксический;

- семантический.

Первый, наиболее простой уровень обработки речевых сигналов (акустический и фонетический анализ) посвящен распознаванию отдельных слов, изолированных или внутри фразы. Второй, более высокий уровень обработки (лексический, синтаксический и семантический анализ) посвящен пониманию слитной речи в виде предложений.

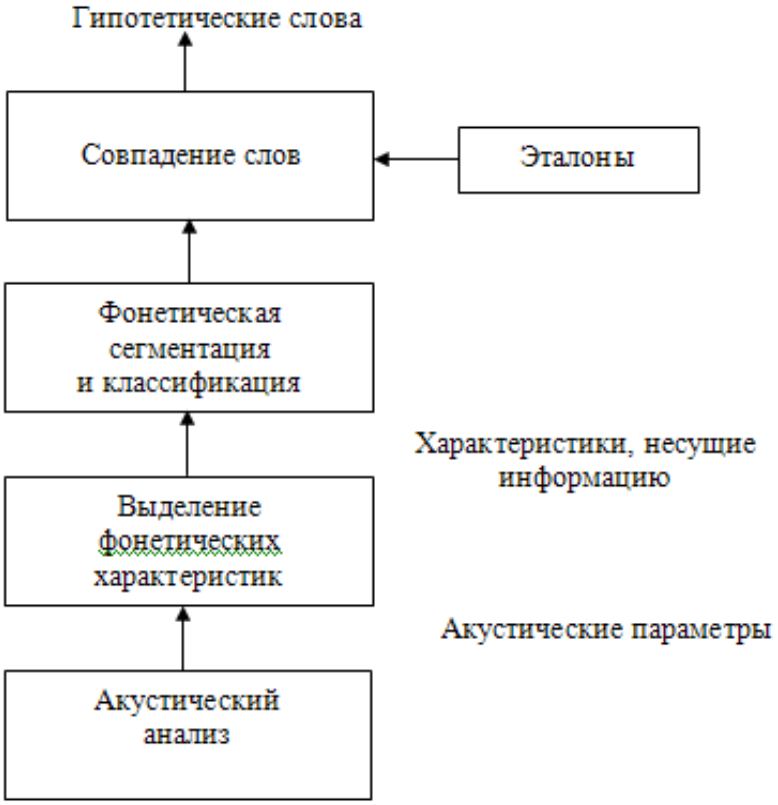


Рис.3.2. Основные этапы обработки изолированных слов

На рис.3.2. представлены два этапа обработки изолированных слов. Основной задачей данного этапа является локализация изолированного слова, выделение его локальных акустических параметров, информативных признаков и лингвистически сопоставимых элементов, сегментация на элементы – фонемы, слоги. По результатам полученных временных и спектральных параметров реализуются алгоритмы сравнения с эталоном.

С точки зрения научных разработок и экспериментальных исследований, среди методов и алгоритмов анализа к распознаванию речи

можно выделить следующие процедуры обработки изолированных слов (рис.3.3).

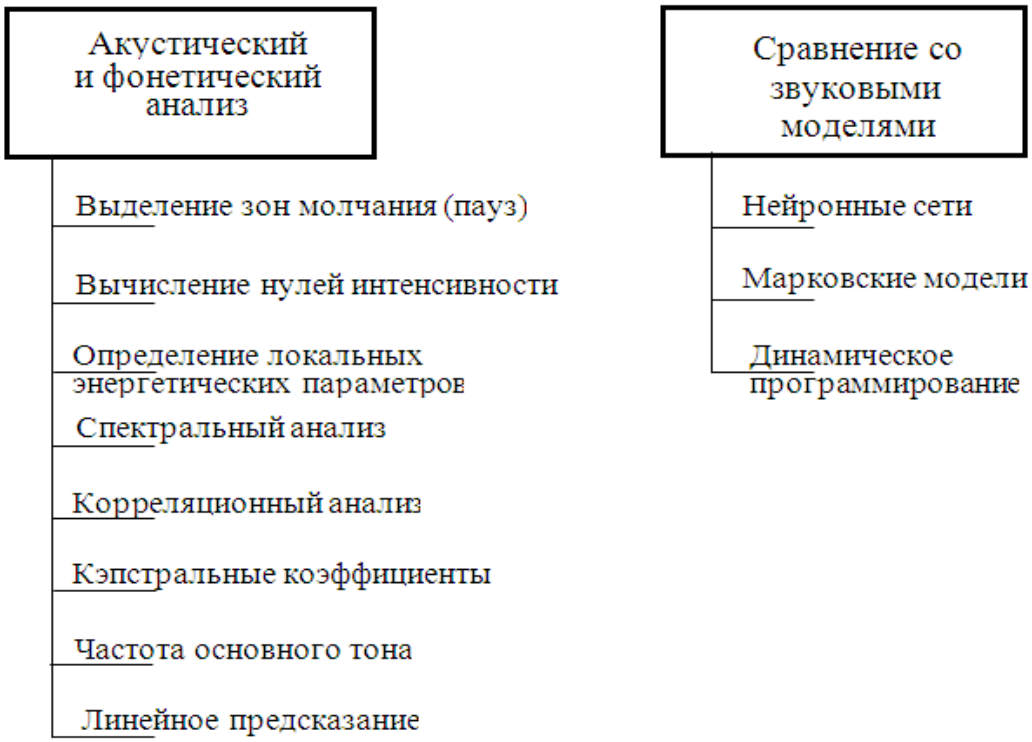


Рис.3.3. Методы и алгоритмы обработки изолированных слов

Выше было показано, что на первых двух этапах обработки речевых сигналов применяются традиционные методы и алгоритмы цифровой обработки сигналов.

Задачи цифровой обработки сигналов, в том числе речевых сигналов, традиционно делятся на две категории по характерам численных методов, способам представления сигналов и алгоритмам обработки.

Первая категория – это методы и алгоритмы обработки сигналов во временной области, когда сам сигнал представляется последовательностью цифровых отсчетов на оси времени.

Вторая категория – это методы обработки сигналов в частотной (спектральной) области, когда сигнал переводится в форму коэффициентов рядов Фурье или вейвлет - анализа.

Численные алгоритмы в обеих категориях различны (хотя многие параметры сигналов можно вычислять как тем, так и другим способом), но в целом они дополняют друг друга в процессе анализа и распознавания речевых сигналов.

На этапе **акустической и фонетической обработки** применяются как стандартные процедуры предварительной фильтрации и подавления шумов (не показаны на рис.3.3), так и алгоритмы извлечения полезной для дальнейшего применения информации. Алгоритмы фонетического анализа тесно переплетаются с алгоритмами акустической обработки, поэтому многие показатели фонетического анализа могут быть определены на этапах обработки во временной или частотной областях. В связи с этим в общей технологии распознавания речи акустический и фонетический анализ расположены на одном этапе. Этот этап – процесс описания сигналов с последующим преобразованием в требуемую форму для возможности выделения информативных признаков.

Процедуры акустической обработки:

- регистрация;
- фильтрация и подавление шума;
- сегментация «сигнал/пауза»;
- сегментация на фреймы;
- сегментация «тон/не тон».

Регистрация представляет собой аудиозахват речевой команды в режиме реального времени и преобразование ее в цифровой вид с использованием стандартных средств: микрофон, предварительный и основной усилитель, аналоговый фильтр низкой частоты, аналого-цифровой преобразователь.

Фильтрация – это этап обработки речевых команд, который позволяет повысить разборчивость, уменьшить долю шумов, вызванных как акустическими, так и технологическими причинами. Шум применительно к речевым сигналам – это совокупность апериодических звуков различной

интенсивности и частоты, которые изменяют информативные параметры сигнала.

Сегментация «сигнал/пауза» представляет собой задачу определения моментов начала и окончания фразы. При наличии шума данная задача является одной из важных в области обработки речевых команд. В частности, при голосовом управлении важно точно определить моменты начала и окончания слова (команды).

Сегментация на фреймы – линейное деление речевого потока на составляющие отрезки, называемые фреймами. Речевые сигналы являются нестационарными сигналами сложной формы, параметры и характеристики которых, как правило, меняются в течение времени. Это предположение приводит к методам кратковременного анализа, в которых сегменты речевого сигнала выделяются и обрабатываются так, как если бы они были короткими участками отдельных звуков с отличающимися свойствами. Для того чтобы получить наборы информативных признаков одинаковой длины, нужно сегментировать речевой сигнал на равные отрезки, называемые фреймами, считая, что сигнал на таком отрезке примерно стационарен. Перекрытие фреймов используется для предотвращения потери информации о сигнале на границе.

Сегментация «тональных/нетональных» участков в речевых сигналах является одной из важных задач в обработке. Под тональными участками понимают интервалы времени, в течение которых генерация звуков речи происходит с участием голосового источника. К нетональным участкам относятся интервалы времени, на которых образование звуков речи происходит без участия голосового источника. Наибольшую ценность при анализе речевых команд играют тональные участки. Анализируя их, можно получить достаточную информативную картину как об акустических характеристиках, так и о смысловом значении речевых сигналов.

Начальными процедурами фонетической обработки является спектральный и корреляционный анализ Определение информативных

параметров – задача выявления информативных признаков и характеристик речевых сигналов. Основные понятия, характеризующие информативные параметры речи человека, связаны с формой, размерами, динамикой изменения речевого аппарата. Выделение тональных участков в некоторых случаях может являться главной целью в обработке речевых команд. К таким случаям относят определение важного параметра речи – **частоты основного тона говорящего** в задаче распознавания и идентификации диктора. По результатам спектрального анализа составляется обобщенная спектрограмма, характеризующая изменение текущих спектральных характеристик отдельных фреймов на всем протяжении выделенного слова.

Использование метода линейного предсказания в фонетическом анализе позволяет отделить периодические составляющие от сложных зависимостей в речи, определить местоположение пиков формантных частот. Этот параметр в свою очередь используется в дальнейших алгоритмах распознавания.

На основании анализа достижений в области выделения информативных признаков параметры речевых сигналов можно разделить на три группы, позволяющие различать речевые образцы.

Амплитудно-частотные признаки:

- количество максимумов амплитуд и нулей интенсивности;

– локальные энергетические параметры;

– частота основного тона;

– формантные частоты.

Речевой сигнал акустически представляет собой распространяемые в воздушной среде сложные по своей структуре звуковые колебания, которые характеризуются в отношении их частоты (числа колебаний в секунду), интенсивности (амплитуды колебаний) и длительности. Амплитудно-частотные признаки несут необходимую и достаточную информацию для человека по речевому сигналу при минимальном времени восприятия.

Спектрально-временные признаки:

– длительность сегмента минимальной структурной единицы речи (фонемы, вокализованного и невокализованного участков);

– относительное время пребывания сигнала в полосах спектра;

– относительная мощность спектра речи в полосах;

– периодических (тональных) участков звуковой волны;

– непериодических участков звуковой волны (шумовых, взрывных);

– вариация огибающей спектра речи.

Спектрально-временные признаки позволяют отражать своеобразие формы временного ряда и спектра голосовых импульсов у разных лиц, характеризуют особенности речевого потока, связанные с динамикой перестройки артикуляционных органов речи говорящего и являются интегральными характеристиками речевого потока.

Кепстральные признаки:

– мел-частотные коэффициенты;

– кепстральные коэффициенты мощности;

– коэффициенты спектра линейного предсказания;

Большинство современных систем голосового управления сосредотачивают усилия на извлечении частотной характеристики речевого тракта человека, отбрасывая при этом характеристики сигнала возбуждения. Это объяснено тем, что коэффициенты первой модели обеспечивают лучшее разделение звуков. Для отделения сигнала возбуждения от сигнала речевого тракта прибегают к кепстральному анализу.

Для реализации всех вышеперечисленных процедур вычисления информативных признаков речевых сигналов необходимы численные методы и соответствующие алгоритмы обработки: спектральный анализ в различных базисах Фурье, вейвлет-преобразование, КИХ-фильтрация, свертка, кепстральный анализ.

В результате данного этапа обработки появляется информационная среда для обнаружения и распознавания слова как части (временного кадра) речевого сигнала (рис.3.4). Эти фонетические элементы (метки) позволяют

рассматривать слово в виде сочетания параметрических признаков при детальном анализе его составляющих.

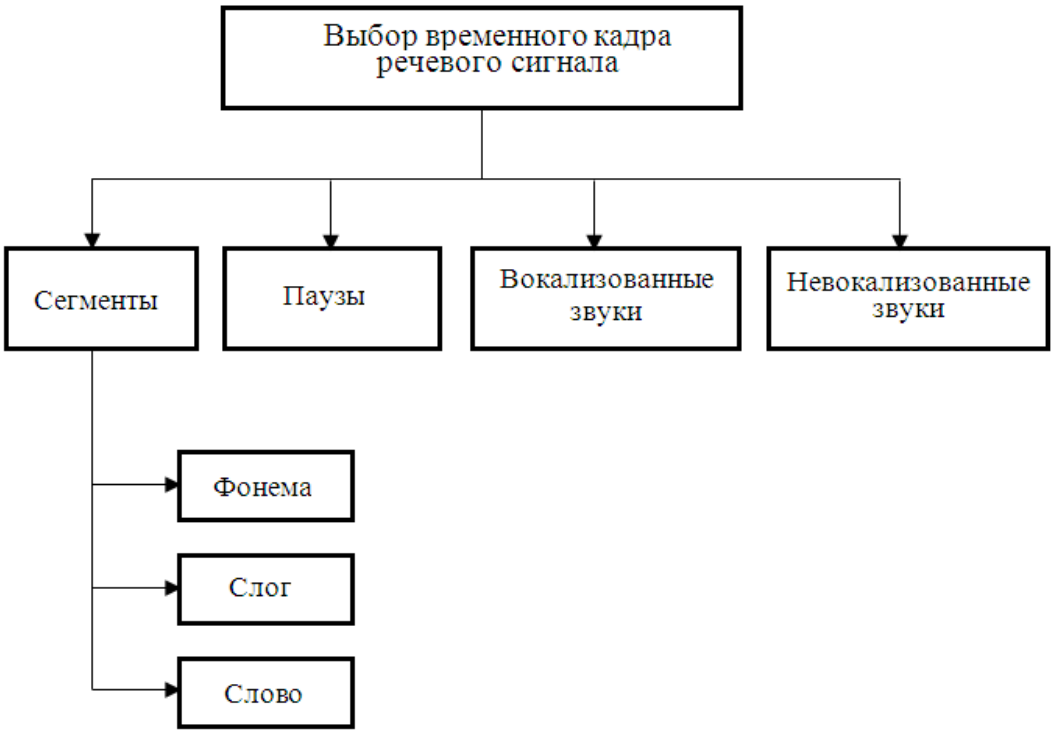


Рис.3.4. Самостоятельные единицы анализа речевого сигнала

Процедуры второго типа, представленные на рис.3 при сравнении со звуковыми моделями, реализуют процессы установления идентичности неизвестного входного сигнала одному из элементов заранее записанного в памяти словаря (эталона). При этом используются процедуры обучения и распознавания.

Применение методов статистической теории распознавания образов стало важным этапом в развитии автоматического распознавания речи. Это позволило исследователям использовать мощный аппарат математической статистики и теории вероятностей, что, в свою очередь, привело к существенному повышению качества распознавания. В настоящее время практически все известные системы распознавания речи основаны на статистических методах.

В рамках такого подхода речевой сигнал представляется как случайный образ, который необходимо распознать и преобразовать в некоторую последовательность слов. Тогда задача распознавания речевого сигнала может быть сформулирована как классическая задача классификации образов по критерию максимума апостериорной вероятности.

3.2. Применение нейронных сетей для распознавания речи

Имея значения существенных параметров речевого сигнала, мы можем приступить к распознаванию звуков. Под существенными параметрами здесь предполагаются такие характеристики звука, которые образуют множество, по которому можно с высокой вероятностью отличить один вид (класс) звука от другого или прийти к заключению о том, что два звука принадлежат одному виду (классу).

Человеческий слуховой аппарат при распознавании звука ориентируется на частотный домен звукового сигнала, при этом, как уже упоминалось ранее, для него практически не имеет значения фаза сигнала. Существенным является лишь абсолютные значения амплитуд частот сигналов, точнее некие соотношения и сочетания абсолютных значений частот. Для программной реализации выберем ранее упоминавшийся распределение энергии сигналов по группам смежных частот (суммарная энергия сигнала в диапазоне частот вычисляем как сумму квадратов амплитуд частот, входящих в диапазон).

Для того, чтобы распознать звук, необходимо иметь образцы значений всех существенных параметров каждого из звуков речи и оценить, относится ли к какому-нибудь из них наш звук, сравнивая значения его параметров со значениями параметров образцов.

Наиболее часто употребляются два подхода к классификации и распознаванию.

В первом некая функция служит мерой близости параметров. Такая функция называется метрикой.

Второй подход не использует вспомогательных функций, но моделирует процесс распознавания в биологических системах. Такой подход использует технологии так называемых нейронных сетей. Для программной реализации мы выберем именно этот, представляющийся более перспективным в настоящее время, подход.

Итак, проведем краткий экскурс в предмет компьютерных нейронных сетей. Нейронные сети – это аппаратные или программные средства, моделирующие работу человеческого мозга. Как и всякая модель, они являются приближением. Но даже несмотря на то, что в подобных средствах имитируются лишь отдельные стороны биологического прототипа, они уже сейчас позволяют добиться определенных успехов во многих областях, в частности связанных с классификацией и распознаванием образов.

Как известно, нервная система человека состоит из огромного числа элементов, называемых нейронами, соединяемыми между собой нитеобразными отростками-дендритами. Возбуждение или торможение (возбуждение со знаком минус) передается от нейрона к нейрону по дендритам, где те принимают сигналы в точках соединения, называемых синапсами. Принятые синапсом входные сигналы передаются к телу нейрона, где суммируются. Если уровень возбуждения превышает некоторую пороговую величину, возбуждение передается из тела нейрона в выходную точку, называемую аксоном, откуда по дендритам поступает в другие нейроны.

Именно указанные выше характеристики и стали существенными при создании искусственных нейронных сетей.

Основу нейронной сети составляют, как правило, однотипные элементы, имитирующие работу биологического нейрона, и называемые обычно так же. Каждый из нейронов в каждый момент времени находится, как и биологический нейрон, в некотором текущем состоянии. Он имеет

группу однонаправленных входных связей-синапсов, идущих от входа в сеть или от других нейронов. Кроме того, он имеет одну однонаправленную выходную связь-аксон.

Схематически нейрон можно представить так, как показано на рис.3.5.

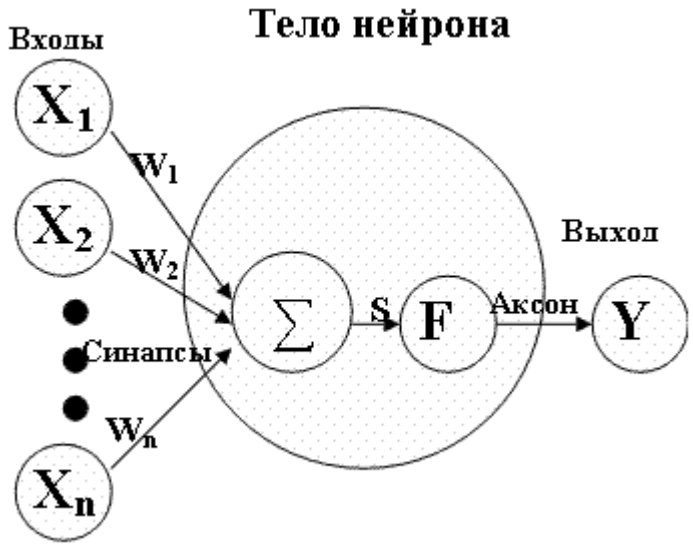


Рис.3.5. Схематически нейрон

Синаптические связи характеризуются весами w_i. Текущее состояние S нейрона равно взвешенной сумме входов:

$$S = \sum_{i=0}^{n} X_i w_i \qquad (3.1)$$

В векторном виде это можно записать как S=XW, то есть вектор S есть произведение вектора входных значений X на матрицу весов W, где строки соответствуют слоям, а столбцы – нейронам внутри каждого слоя.

Функция S далее преобразуется **активационной функцией** F и дает выходной сигнал Y нейрона.

$$Y=F(S) \qquad (3.2)$$

Активационная функция должна обладать свойством резко возрастать на коротком интервале аргумента в окрестностях порогового значения T, принимать приблизительно одно значение до этого интервала и приблизительно одно (большее) значение – после этого интервала. Этим

требованиям соответствует, например функция Y, равная 1 при S>T, и 0 при S≤T.

Эта функция называется также **функцией единичного скачка** (рис.3.6).

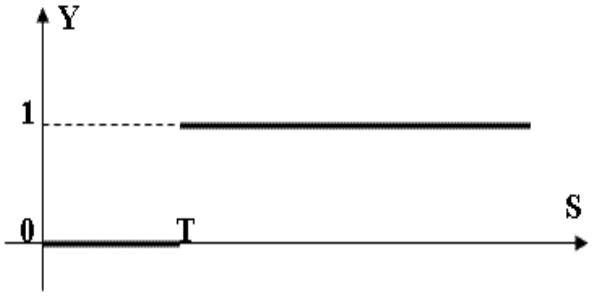


Рис.3.6. График функции единичного скачка

Но, к сожалению, опыт показал, что многие алгоритмы нейронных сетей плохо работают или не работают с линейными функциями. К тому же эта функция содержит разрыв в точке T.

Самой распространенной в нейронных сетях активационной функцией является сигмоид или логистическая функция (рис.3.7), вычисляемая как

$$F(S)=1/(1+e^{-S}) \qquad (3.3)$$

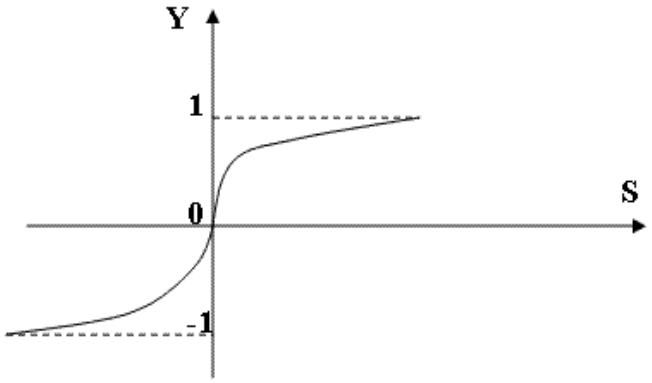


Рис.3.7. График сигмоид или логистической функции

На самом деле, в общем виде формула логистической функции выглядит так:

$$F(S)=1/(1+e^{-\alpha s}) \qquad (3.4)$$

где α –некоторая константа. При уменьшении логистическая функции становится более пологой, при α=0 принимая вид горизонтальной линии Y=0.5. При увеличении сигмоид приближается по внешнему виду к функции единичного скачка.

Еще одним весьма полезным свойством логистической функции является ее дифференцируемость при любых S и простота вычисления ее производной:

$$F'(S)= \alpha F(S)(1-F(S)) \qquad (3.5)$$

Простейшей моделью нейронной сети является однослойный персептрон. Однослойность означает, что входной сигнал входов (x_1,x_2,…x_n) подается на одну группу нейронов, именуемых слоем нейронной сети, а выходные сигналы этих нейронов поступают сразу на выход сети. Для двуслойной сети выходные сигналы подавались бы не на выход сети, а на вторую группу-слой нейронов, а оттуда на вход. Понятно, что трехслойный нейрон имеет уже три группы-слоя, N-слойный – N групп-слоев и т.д. (см.рис.3.8).

Можно считать, что на вход первого слоя подаются сигналы со всех входов сети, просто веса некоторых синапсов равны нулю. Аналогично можно считать, что на входы всех слоев кроме первого подаются сигналы с выходов всех нейронов предыдущего слоя.

Но вернемся к однослойному персептрону. Он обнаружил ряд положительных свойств, которые и заставили многих ученых обратить свой взор на исследование нейронных сетей. Главными из обнаруженных свойств персептрона была способность к обучению и распознаванию.

Оказалось возможным в ряде случаев установить то, как можно настроить веса синапсов персептрона, чтобы при различных комбинациях значений входов получать заранее установленные, “правильные” значения

выходов. То есть, однослойный персептрон оказался способным воспроизводить некоторые математические функции.

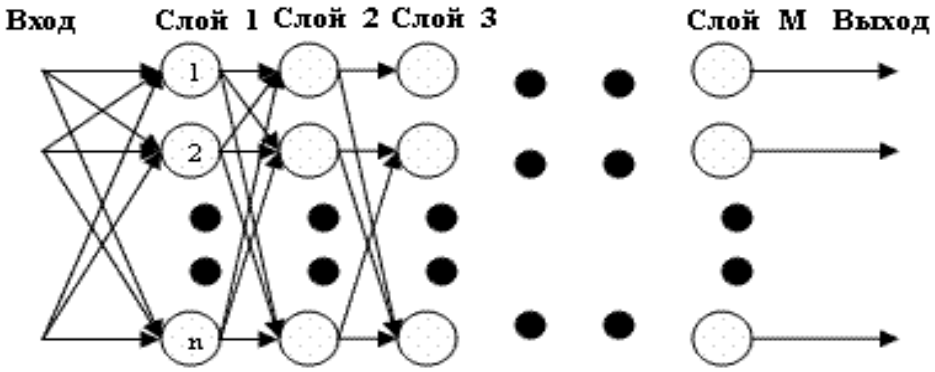


Рис.3.8. Многослойная нейронная сеть

Однако радость по поводу этих открытий оказалась недолгой: были обнаружены существенные ограничения в способностях однослойного персептрона к обучению и распознаванию.

Во-первых, было доказана его неспособность воспроизводить некоторые простые функции, например, ИСКЛЮЧАЮЩЕЕ ИЛИ.

Если T_j – пороговое значение j-го выхода, то однослойный персептрон описывается уравнениями:

$$T_j = \sum_{i=1}^{n} x_i w_{1,j,i} \qquad (3.6)$$

где n – число нейронов в слое. В n-мерном пространстве эти функции оказываются прямыми. Точки n-мерного пространства входов для значений дающих разные значения (0 или 1) выхода (то есть F>Tj или F<=Tj) должны лежать по разные стороны от таких прямых. Но для многих функций (например ИСКЛЮЧАЮЩЕЕ ИЛИ) это невыполнимо.

Эта задача решается с помощью **многослойных нейронных сетей.** Уже при количестве слоев равном двум сеть описывается уравнением:

$$T_j = \sum_{i=1}^{n} \left(\sum_{i=1}^{n} x_i w_{1,j,i} \right) w_{2,j,i} \qquad (3.7)$$

Теперь включая достаточное число нейронов во входной слой можно получить выпуклый многоугольник любой формы. Трехслойная же сеть есть еще более общий случай. В ней ограничения на выпуклость отсутствуют: нейрон третьего слоя принимает на вход выпуклые многоугольники, комбинация которых может быть уже невыпуклой.

Таким образом, способности к распознаванию у многослойных сетей значительно превосходят те же способности у однослойного персептрона. Зато несколько усложняется процесс обучения этой сети. Под **обучением сети** мы понимаем процесс настройки весов синапсов, так чтобы выход сети был ожидаемым.

Если в сети только один слой, процесс ее обучения довольно очевиден. Рассмотрим, например, алгоритм Кохонена.

1. Весовые коэффициенты w_{ij}, определяющие i-й вход j-го нейрона, устанавливаются в некоторые малые значения. Устанавливаются входы Xi .
2. Вычисляются значения вспомогательного вектора Rj для каждого входа по формуле:

$$R_j = \sum_{i=1}^{n} (X_i - w_{1,j,i})^2 \qquad (3.8)$$

где n – число нейронов в сети.

3. Определяется m, такое что

$$R_m \min_j R_j \qquad (3.9)$$

4. Пересчитать все wij для нейрона с номером m по формуле

$w_{im} = w_{im} + M(X_i - w_{ij})$, где $0.5 <= M <= 1$.

5. Если решение не было достигнуто, переходим к шагу 2.

Существуют и другие алгоритмы для однослойной сети.

Алгоритмы обратного распространения.

Сложнее обстоит дело с многослойными сетями, так как изначально неизвестны желаемые выходы слоев сети (за исключением последнего) и их

невозможно обучить, руководствуясь только величиной ошибок на выходе сети, как это было с однослойной сетью.

Наиболее приемлемым вариантом решения проблемы стала идея распространения сигнала ошибки от выхода сети к ее входу, слой за слоем. Алгоритмы, реализующие обучение сети по этой схеме, получили название алгоритмов обратного распространения.

Алгоритм требует дифференцируемости активационной (или как ее по-другому называют, сжимающей) функции на всей оси абсцисс. По этой причине, функция единичного скачка не может использоваться и в качестве сжимающей функции обычно применяют упомянутый выше сигмоид (логистическую функцию), хотя существуют и другие варианты.

Рассматриваем “классический вариант” многослойной сети, где синаптические связи могут определяться любыми действительными числами, а выход нейрона – действительными числами из интервала от 0 до 1. В качестве активационной функции используем сигмоид. Число слоев произвольное.

Описание алгоритма.

1. Определяем M матриц весовых коэффициентов W размером NxN, где M – число слоев, N – число нейронов в одном слое. Wi, j, k будет обозначать вес j-го входа k-го нейрона в i-м слое. Инициализируем матрицы некоторыми малыми случайными (не одинаковыми) значениями.
2. Подаем на входы сети определенные значения X, для которых известны правильные значения выходов сети Y*.
3. Вычисляем значения выходов сети для текущего состояния матриц W. То есть для входного вектора X вычисляется выходной вектор Y. Для этого необходимо последовательно вычислить выход для каждого слоя сети с первого по последний. Для i-го слоя в векторном виде это можно записать так:

$O_i=F(XW_i)$, если i – не первый слой.

$O_i=F(O_{i-1}W_i)$, если i – не первый слой.

где O_i – вектор выхода i-го слоя, F – активационная функция, X – вектор входов, O_{i-1} – вектор выхода (i-1)-го слоя, W_i – матрица весовых коэффициентов i-го слоя.

4. Вычисляем вектор ΔY=Y-Y*

5. Если ΔY меньше заданной погрешности, переходим к шагу 9.

6. Для слоя с номером M (т.е. в последнем слое) производим следующие операции:

6.1. Для всех нейронов в слое с номера 1 по N производим следующие операции:

6.1.1. Для всех весов нейрона с номера 1 по N производим следующие операции:

6.1.1.1. Рассчитываем вектор δM=X(1-X)ΔY

6.1.1.2. Рассчитываем величину $\Delta W_{M,j,k}=\eta\delta_{M,k}O_{i-1,j}$,

где η – коэффициент скорости обучения (от 0.01 до 1.0)

6.1.1.3. Корректируем величину весового коэффициента, добавляя к $W_{M,\ j,\ k}$ величину $\Delta W_{M,j,k}$

7. Для слоев с номером M-1 по первый последовательно производим следующие операции:

7.1. Для всех нейронов в слое с номера 1 по N производим следующие операции:

7.1.1. Для всех весов нейрона с номера 1 по N производим следующие операции:

7.1.1.1. Рассчитываем вектор

$$\delta_i = \mathbf{O}_{i+1}\left(\mathbf{1}-\mathbf{O}_{i+1}\right)\left[\sum_{k=1}^{N}\delta_{i+1,j}W_{i+1,j,k}\right] \qquad (3.10)$$

7.1.1.2. Рассчитываем величину $\Delta W_{i,j,k}=\eta\delta_{i,k}O_{i-1,j}$,

где η – коэффициент скорости обучения (от 0.01 до 1.0)

7.1.1.3. Корректируем величину весового коэффициента, добавляя к $W_{M,\ j,\ k}$ величину $\Delta W_{M,j,k}$

8. Переход к шагу 3.

9. Конец (обучение окончено).

Описанный алгоритм применяется достаточное количество раз, чтобы все варианты выходных значений могли правильно выходить при задании произвольных значений входа с заданной вероятностью ошибки.

Это алгоритм может быть усовершенствован. Например, выяснилось, что обычный диапазон для входов и выходов от 0 до 1 не является оптимальным. Из-за того, что $\Delta W_{i,j,k}$ прямо пропорционален выходному уровню нейрона, нулевой выходной уровень приводит к нулевому значению $\Delta W_{i,j,k}$, то есть величина веса не изменяется и обучение не происходит. Выход состоит в приведении входов к значениям от -0.5 до 0.5. Активационная функция должна приобрести вид:

$$F(S) = (-0.5)+1/(1+e^{-s}) \qquad (3.11)$$

После того, как сеть будет надлежащим образом обучена, она может быть использована для распознавания, в том числе, для распознавания звуков. Подаем на вход сети параметры звукового сигнала и получаем на выходе последовательность значений от -0.5 до 0.5 (или – после обратной корректировки – от 0 до 1), по которым и определяем звук (каждому звуку сопоставляется уникальная комбинация выходов до начала процесса обучения, и, собственно по ней мы на этапе обучения и определяем, правильно ли определен звук).

3.3. Вычисление коэффициентов линейного предсказания

Линейное предсказание (англ. linear prediction) — вычислительная процедура, позволяющая по некоторому набору предшествующих отсчётов цифрового сигнала предсказать текущий отсчёт.

Суть линейного предсказания в нахождении коэффициентов a_k (k=1..p) для формулы:

$$x[n] = \sum_{k=1}^{p} (a_k\, x[n-k]) \qquad (3.12)$$

Другими словами, строится линейный многочлен, позволяющий с хорошей точностью вычислять значение любого отсчета в сигнале по значениям предыдущих p отсчетов. Коэффициенты a_k и называются **коэффициентами линейного предсказания**.

Фактически, имея некоторый сигнал, можно иметь статистическую выборку, которую можно представить в виде таблицы:

x[n-p]	**x[n-p+1]**	**x[n-p+2]**	**. . .**	**x[n-1]**	**x[n]**
x[0]	x[1]	x[2]	. . .	x[p-1]	x[p]
x[1]	x[2]	x[3]	. . .	x[p]	x[p+1]
x[2]	x[3]	x[4]	. . .	x[p+1]	x[p+2]
. . .	. . .	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .	. . .
x[N-p-1]	x[N-p]	x[N-p+1]	. . .	x[N-2]	x[N-1]

То есть нахождение коэффициентов линейного предсказания сводится к вычислению коэффициентов линейной регрессии для данной статистической выборки и можно пользоваться методами математической статистики.

Минимизируем сумму квадратов ошибок для каждого из вычисляемых отсчетов.

Ошибка для отсчета x[n] равна:

$$\delta[n] = x[n] - \sum_{k=1}^{p} (a_k\, x[n-k]) \qquad (3.13)$$

Минимизируемая функция равна

$$\sum_{n=p}^{N-1} \left(x[n] - \sum_{k=1}^{p} (a_k\, x[n-k]) \right)^2 = \sum_{n=p}^{N-1} \left(x[n] - \sum_{k=1}^{p} (a_k\, x[n-k]) \right) \left(x[n] - \sum_{k=1}^{p} (a_k\, x[n-k]) \right)$$

$$E = \sum_{n=0} \delta^2[n] = \sum_{n=0} x[n] - \sum_{k=1} (a_k x[n-k]))^2 = \sum_{n=0} x^2[n] - 2 \sum_{n=0} x[n] \sum_{k=1} (a_k x[n-k]))+$$

$$+ \sum_{n=0}^{N-1} (\sum_{k=1}^{p} (a_k x[n-k]))^2 = \sum_{n=0}^{N-1} x^2[n] - 2 \sum_{k=1}^{p} (a_k \sum_{n=0}^{N-1} (x[n] x[n-k])) +$$

$$+ \sum_{j-1}^{p} \sum_{k=1}^{p} a_k a_j \sum_{n=0}^{N-1} (x[n-k]x[n-j])) \qquad (3.14)$$

Продифференцируем E по a_k и приравняем частные производные нулю для нахождения экстремума:

$$dE/da_k = \sum_{n=0}^{N-1} (x[n] x[n-k])) + \sum_{j=1}^{p} a_j \sum_{n=0}^{N-1} (x[n-k]x[n-j]))=0 \qquad (3.15)$$

Заменив для удобства восприятия j на i, a k на j получим систему p линейных уравнений с p неизвестными:

$$\sum_{i=1}^{p} a_i c_{ij} = c_{0j} \qquad (3.16)$$

где

$$c_{ij} = c_{ji} = \sum_{n=0}^{N-1} x[n-i]x[n-j]) \qquad (3.17)$$

Эта система называется системой уравнений Юла-Уокера. Погрешность найденных коэффициентов оценивается как:

$$E = c_{00} - 2 \sum_{i=1}^{p} a_i c_{0i} + \sum_{i=1}^{p} a_i \sum_{j=1}^{p} a_j c_{ij} = c_{00} - \sum_{i=1}^{p} a_i c_{0i} \qquad (3.18)$$

Есть два основных подхода для решения системы уравнений Юла-Уокера.

Решение системы уравнений Юла-Уокера.

Подход 1. Ковариационный метод.

Если мы разделим выражение для c_{ij} на N, то получим функцию взаимной корреляции между двумя сегментами сигнала:

$$R(i,j) = c_{ij}/N = 1/N \sum_{n=0}^{N-1} (x[n-i]x[n-j]) \qquad (3.19)$$

В матричном виде система уравнений Юла-Уокера примет вид:

$$A \cdot P = C \qquad (3.20)$$

где A – вектор коэффициентов a_i, P – массив значений R(i,j) , C – вектор значений R(0,i), i=1..p, j=1..p

Далее система решается традиционными методами решения систем линейных уравнений, которые в общем виде требуют порядка p^3 операций.

Подход 2. Автокорелляционный метод.

Изменим пределы суммирования в исходном выражении для c_{ij}:

$$c_{ij}=c_{ji}= \sum_{n=0}^{N-1-|i-j|} (x[n]x[n+|i-j|]) \qquad (3.21)$$

Если далее мы разделим выражение для c_{ij} на N, то получим функцию автокорреляции $R(\tau)$, вычисленную для $\tau=|i-j|$

$$R(|i-j|) = cij /N = 1/N \sum_{n=0}^{N-1} (x[n]x[n+|i-j|]) \qquad (3.22)$$

$$\sum_{i=1}^{p} a_i R(|i-j|)=R(j), \qquad (3.23)$$

где j=1..p.

В матричном виде система уравнений Юла-Уокера примет вид:

$$A \cdot R = B \qquad (3.24)$$

где A – вектор коэффициентов a_i, C – вектор значений R(i), i=1..p, а

$$R := \begin{pmatrix} R(0) & R(1) & .. & R(p-1) \\ R(1) & R(0) & .. & R(p-2) \\ . & . & . & . \\ R(p-1) & R(p-2) & .. & R(0) \end{pmatrix}$$

Матрица R симметрична относительно главной диагонали и является матрицей Теплица, то есть каждая строка получается из предыдущего сдвига вправо на одну позицию. Для подобных матриц система линейных уравнений может быть решена по более простому по сравнению с классическими методом Левинсона-Дарбина, алгоритм которого требует порядка p2 операций (то есть значительно быстрее классических, применяемых при ковариационном методе).

Алгоритм Левинсона-Дарбина следующий.

1. Начальные условия:

l=0, E=R[0], a[i]=0 для i=1..p

2. Для l=1 по p цикл l-1

2.1. k[l] = (a[i]R[l-1]-R[l])/E i=1

2.2. a[l] = -k[l]

2.3. Для j=1 по l-1 цикл

2.3.1. a[j]=a[j]+k[l]a[l-j]

2.4. E=E(1-k[l]2)

На выходе алгоритма получаем оценку ошибки E и вектор коэффициентов линейного предсказания a[i].

Приводим реализацию алгоритма вычисления линейных коэффициентов автокорреляционным методом с применением алгоритма Левинсона-Дарбина на языке C.

```
//Функция вычисляет коэффициенты линейного предсказания для сигнала x длиной //N, возвращает оценку ошибки. Параметр p задает число отсчетов, на основании //которых делается предсказание, в параметре a возвращается ссылка на массив //полученных коэффициентов
double LineFactors(double* x, int N, int p, double* a)
{
double E;
double* R=(double*)malloc(sizeof(double)*p);
double* k=(double*)malloc(sizeof(double)*p);
int i,j,l,n;
a=(double*)malloc(sizeof(double)*p);
for(i=0;i‹p;i++)
{
a[i]=0;
R[i]=0;
for(n=0;n‹N;j++)
{
R[n]+=x[n]*x[n+i];
}
R[n]/=N;
```

```
}
E=R[0];
l=0;
for(l=1;l‹=p;l++)
{
k[l]=0;
for(i=1;i‹l;i++)
{
k[l]+=a[i]*R[l-i];
}
k[l]=(k[l]-R[l])/E;
a[l]=-k[l];
for(j=1;j‹l;j++)
{
a[j]+=k[l]*a[i-j];
}
E=E*(1-k[l]*k[l]);
}
return E;
}
```

3.4. Применение скрытых марковских моделей для распознавания речи

Скрытая Марковская модель (СММ) - стохастический метод, в котором некоторая временная информация может быть объединена. В этом разделе даны основные принципы алгоритмов распознавания речи, которые используют СММ. Рисунок 3.9 показывает блок-схему типичной системы распознавания речи. Для начала, векторы свойств извлекаются из речевой звуковой волны.

Затем, наиболее вероятная последовательность слова для данных речевых характеристических векторов находят, используя два вида источников знаний, т.е., акустические и лингвистические знания. СММ используется, чтобы захватить акустические особенности разговорного звука, а модель стохастического языка используется, чтобы представить лингвистические знание.

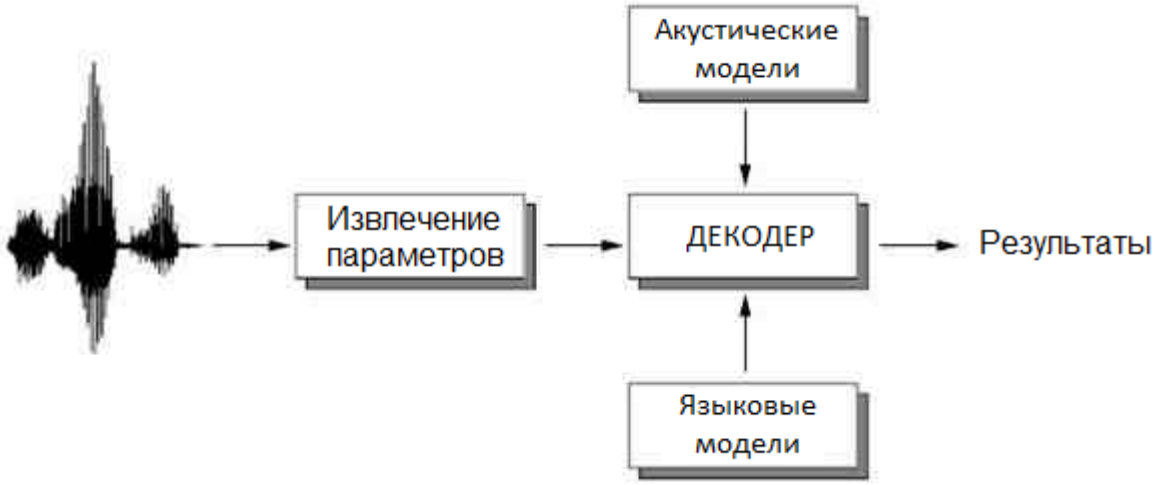


Рис.3.9. Система расспознования речи

Выделение признаков. Так как воздух выходит из легких, напряжение голосовых связок заставляет вибрировать воздушный поток. Эти квазипериодические импульсы затем фильтруются, проходя через голосовой тракт и носовой трактат, создавая озвученные звуки. Различные позиции артикуляционных органов, как например челюсть, язык, губы, и мягкое небо, производят различные звуки. Когда голосовые связки расслаблены, воздушный поток проходит через сокращение в голосовом тракте, или создает давление сзади пункта прекращения и давление внезапно ослабевает, порождая глухие звуки. Позиции сокращения или прекращения создают различные звуки. Речь это просто последовательность озвученных и не озвученных звуков, которые изменяют медленно (5..100 ms) поскольку конфигурация органов артикуляции изменяется медленно. Рисунок 3.10,а показывает пример звуковой формы волны предложения, “У нее есть ваш темный костюм”, который произносит мужчина диктор. Для автоматического

распознавания речи компьютерами, характеристические векторы извлекаются из звуковой формы волны. Характеристический вектор обычно считается от окна разговорных сигналов (20..30 ms) в каждом коротком интервале времени (около 10 ms). Произнесение представлено как последовательность этих характеристических векторов особенностей. Cepstrum - широко используемая особенность вектор для распознавания речи. Cepstrum определен, как обратное преобразование логарифмического спектра короткого времени. Низшие порядковые cepstral-коэффициенты представляют голосовой ответ импульса тракта.

С целью взять слуховые характеристики во внимание, взвешенные средние величины спектральных значений на логарифмическом частотном масштабе используются вместо спектра величины, производя mel-частотные cepstral-коэффициенты (MFCC). Производные MFCC обычно присоединены для захватывания динамики речи. Рисунок 3.10 (b) и (c) - спектрограмма и MFCC, извлеченный от примера произнесения, рассмотренного выше.

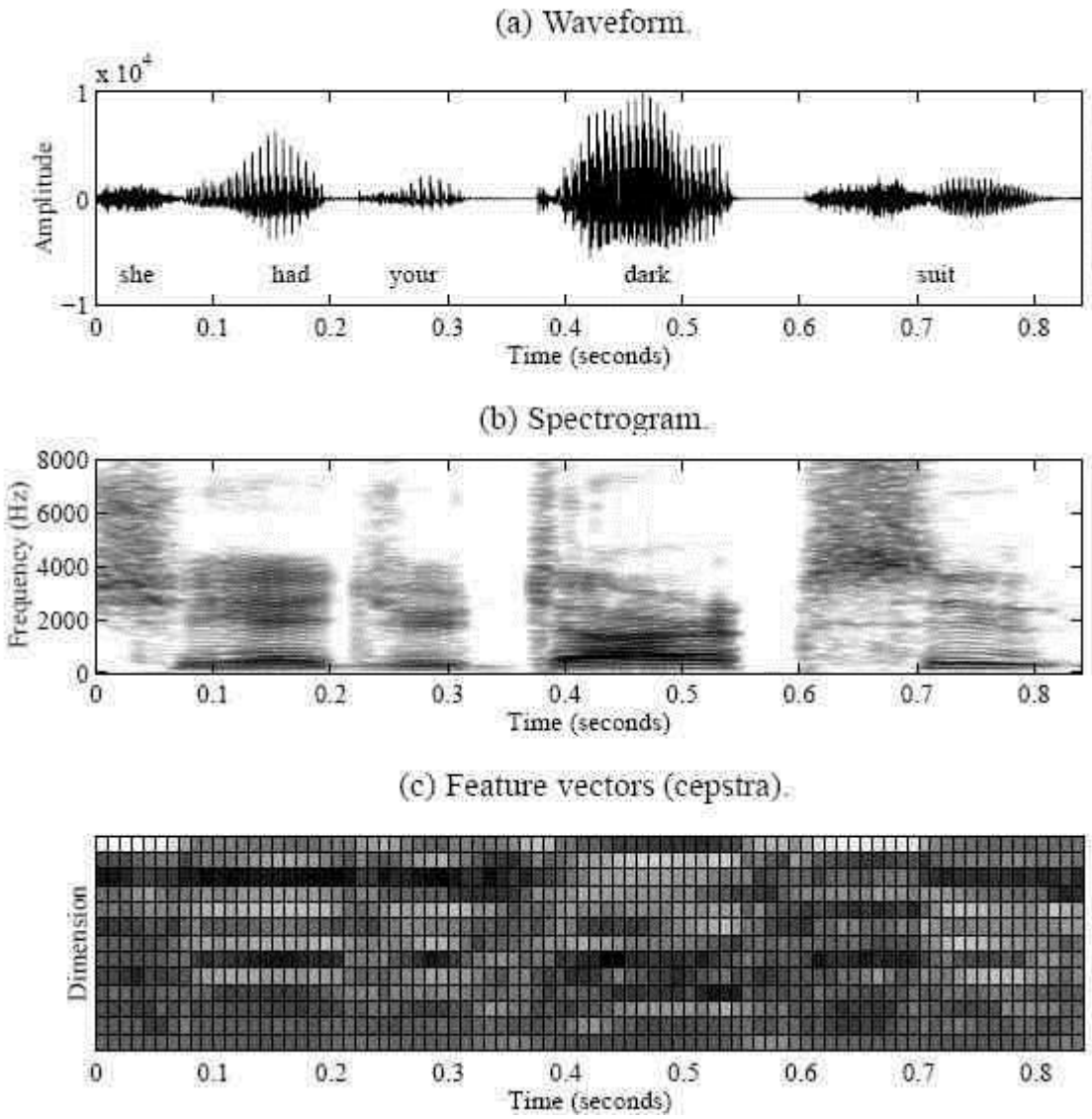


Рисунок 3.10. Пример звуковой формы волны, спектрограммы, и характеристических векторов.

Техника для устойчивого распознавания речи, которая применяется к cepstral-коэффициентам - это нормализация cepstral средины (CMN). С тех пор, как искажения такие, как например отражение и различные микрофоны становятся аддитивными ответвлениями после логарифмирования, вычитание шумового компонента из искаженной речи будет обеспечивать чистый речевой компонент. Однако, оценивая шум искаженной речи это нелегкая задача. CMN приближает шумовой компонент со средним cepstra, предполагая, что средняя величина линейной речи спектра равен 1, который

очевидно не верен. Средний вектор каждого произнесения вычисляется и вычитается от речевых векторов. Опыт показал, что CMN представляет устойчивые характеристики для шума.

Несмотря на то, что CMN прост и быстр, его эффективность ограничена шумом, потому что это перемещает спектральное наклонное положение, вызванное шумом. Также, оценивание среднего вектора не надежно, когда произнесение слишком коротко.

Скрытые Марковские Модели.

Распознавание речи может рассматриваться, как проблема распознавания образов. Если распределение разговорных данных известно, Байесовский классификатор,

$$\overline{U} = \arg\max_{U \in U^*} P(U|X) \qquad (3.25)$$

находит самое вероятное высказывание U (последовательность слова), для предоставленных характеристических векторов X (последовательность наблюдения). Классификаторы Байеса оптимальны в смысле, что вероятность ошибки минимальна. CMM может рассматриваться, как особый случай Байесовского классификатора. В этом разделе обсуждается, как речи представляется CMM.

Акустическое моделирование. Одна из отличительных характеристик речи является ее динамичность. Даже в пределах маленького сегмента, как например фонема, звуки изменяются постепенно. Начало фонемы зависит от предыдущих фонем, средняя часть фонемы в общем стабильна, и на конец воздействуют следующие фонемы. Временная информация о характеристических векторах играет важную роль в процессе распознавания. Для того, чтобы захватить динамичные характеристики речи в рамках классификатора Байеса, нужно наложить определенные временные ограничения. Обычно используется ориентированная слева направо CMM, состоящая из 3 состояний, чтобы представить фонему. Рисунок 3.11 показывает пример такой CMM, где A_{ij} представляет вероятность изменения

состояний от состояния i к состоянию j, и $b_i(x)$ - вероятность наблюдения характеристического вектора X, полученного в состоянии i. Каждое состояние в СММ моделирует распределение звука в фонеме.

Фонема в СММ на рисунке 3.11 состоит из 3 последовательных распределений.

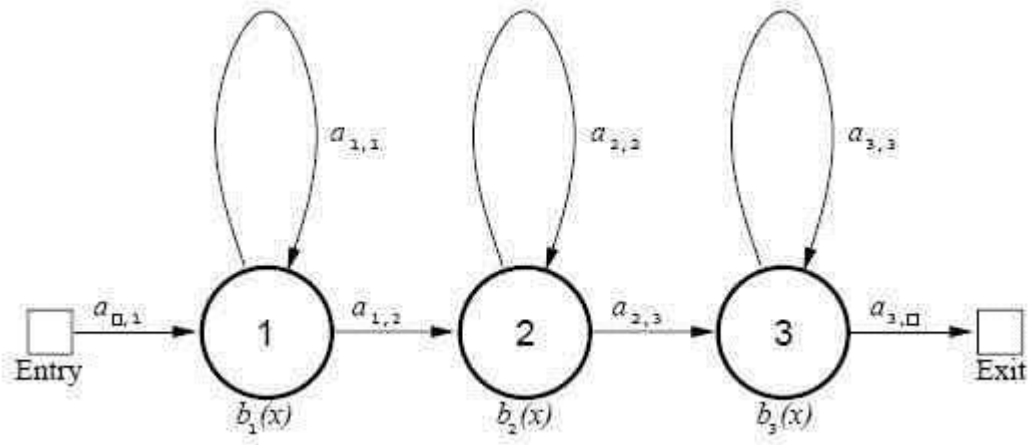


Рис.3.11. Трехступенчатая Скрытая марковская модель

Слово в СММ может быть сконструировано, как конкатенация фонем СММ. Предложение СММ может быть сконструировано соединением слов СММ. Вероятность характеристических векторов, производимых СММ, считается используя вероятности переходов между состояниями и вероятности наблюдений характеристических векторов в данных состояниях. Например, рассмотрим последовательность наблюдения, состоящую из семи векторов:

$$X = x^{(1)}x^{(2)}\cdots x^{(7)}, \text{ where } x^{(t)}$$

Это означает характеристический вектор во время T в последовательности. Предполагается, что первые два вектора принадлежат к первому состоянию, следующие три вектора принадлежат ко второму состоянию, и остальные принадлежат к последнему состоянию. Вероятность последовательности

наблюдения X и это состоянию связывается с S, полученное произнесением CMM U, может быть вычислено, как указано ниже:

$$P(X,S|U) = a_{1,1}b_1(x^{(1)})a_{1,1}b_1(x^{(2)})a_{1,2}b_2(x^{(3)})a_{2,2}b_2(x^{(4)})a_{2,2}b_2(x^{(5)}) \times \times a_{2,3}b_3(x^{(6)})a_{3,3}b_3(x^{(7)})a_{3,3} \quad , \tag{3.26}$$

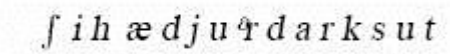
где A_{ij} - вероятность изменения состояний, и $B_i(x(t))$ - вероятность наблюдения характеристического вектора x(t), полученного в состоянии i. Чтобы вычислить вероятность последовательности наблюдения X полученное CMM U, все условные вероятности X и S предоставленные U приходится суммироваться по всем возможным состояниям/векторными назначениям (также называется состояние/рамка выравнивания):

$$P(X|U) = \sum_{S \in S^*} P(X,S|U) \tag{3.27}$$

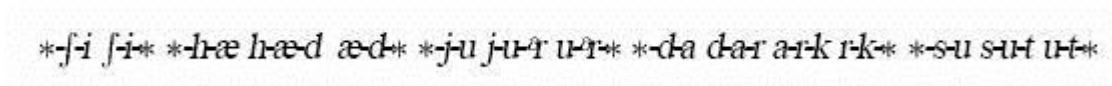
где S* это все возможные состояния последовательности. Это суммирование занимает |S*|·|X| времени, где |S*| - число состояний в CMM и |X| - число характеристических векторов. Существует более эффективный алгоритм.

Моделирование под-слова. В большом словаре распознавания (LVCSR) речи, трудно надежно оценить параметры всех слов CMM в словаре, потому что большинство из слов не были проработаны достаточно часто в учебных данных. К тому же, некоторые из слов словарей могут быть совсем не рассмотрены в учебных данных, которые ухудшают точность распознавания. С другой стороны, число единиц под-слов, как например фонемы обычно намного меньше, чем число слов. Большинство языков имеют около 50 фонем. Есть больше данных на модель фонемы, чем на модель слова, и все фонемы происходят справедливо часто в разумном размере в учебных данныех. Монофонемные CMM моделирует одну фонему. Это – контекстно-независимая единица в смысле, что это не отличает его от соседнего фонетического контекста. В спокойно произнесенной речи,

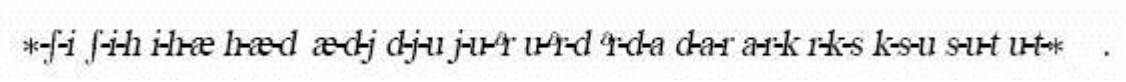
однако, на фонему сильно воздействуют его граничащие фонемы, производя различные звуки в зависимости от фонетического контекста. Это названо коартикуляционным эффектом. Он есть благодаря факту, что артикуляционные органы не могут двигаться мгновенно от одной позиции к другой. Для того, чтобы управлять коартикуляционным эффектом более эффективно, могут использоваться контекстно-зависимые единицы, как например бифоны или трифоны. Бифонемная СММ моделирует фонему со своим левым или правым контекстом. Трифонемная СММ представляет фонему со своим левым и правым контекстом. Например, предложение “У нее есть ваш темный костюм” может быть представлено, как



использование монофонем. Такое же предложение может быть представлено, как



использование трифонемных моделей. Непрерывно в разговорной речи, произношении текущего слова связано с соседними словами. Трифонемная СММ с пересекающимися словами управляет этим коартикуляционным эффектом между словами. Когда кроссворд трифонов используется, пример предложение представлено, как



Более детальные контекстно-зависимые единицы используются, большее число единиц увеличивается. Число трифонов, возможно, становится большим, чем число слов словаря. Это дает начало проблемы способности к обучению снова; т.е., мало данные на модель. Эта проблема решается слиянием подобных моделей контекстов вместе. Слияние может быть сделано на фонемном уровне или на уровне состояний. Так или иначе, СММ требует большого количества данных для обучения, чтобы надежно оценить параметры. Хотя процедура оценки параметра вычислительно

эффективна, сбор данных для обучения - очень дорогая задача. Для новой или неизвестной окружающей среды, перетренировка или многостильное обучение дорого в терминах сбора данных. В этих случаях, применяются такие подходы, как адаптация параметра.

Реализация СММ на компьютере.

Приведем пример реализации решения основных задач для скрытых Марковских моделей на языке С.

```
#include "stdio.h"
#include "stdlib.h"
//Структура, описывающая скрытую Марковскую модель
typedef struct{
int N; //Число состояний
int M; //Число вариантов наблюдений
double**A; //Матрица вероятностей переходов между состояниями
double**B; //Матрица вероятностей каждого наблюдения в каждом
состоянии
double*pi; //Вектор вероятностей начальных состояний
}HMM;
//Функция инициализации скрытой Марковской модели из N состояний и M
вариантов наблюдений
HMM* InitHMM(int N, int M)
{
HMM* ret=(HMM*)malloc(sizeof(HMM));
ret->pi=(double*)malloc(N*sizeof(double));
ret->A=(double**)malloc(N*sizeof(double*));
ret->B=(double**)malloc(N*sizeof(double*));
for(int i=0;i
{
ret->A[i]=(double*)malloc(N*sizeof(double));
ret->B[i]=(double*)malloc(M*sizeof(double));
```

```
}
return ret;
}
//Функция, возвращающая для указанной модели Lambda
//вероятность появления заданной последовательности Seq длиной tau,
//содержащей номера вариантов наблюдений, методом прямого хода
double Seq_Probability_Forward(HMM* Lambda, int* Seq, int tau)
{
double ret=0;
double *alpha=(double*)malloc((Lambda->N)*sizeof(double));
double *alpha_next=(double*)malloc((Lambda->N)*sizeof(double));
for(int i=0;i<(Lambda->N);i++)
alpha[i]=Lambda->pi[i]*Lambda->B[i][0];
for(int t=0;t
{
for(int j=0;jN;j++)
{
alpha_next[j]=0;
for(i=0;iN;i++)
alpha_next[j]+=alpha[i]*Lambda->A[i][j];
alpha_next[j]*=Lambda->B[j][Seq[t]];
}
free(alpha);
alpha=alpha_next;
alpha_next=(double*)malloc((Lambda->N)*sizeof(double));
}
return(ret);
}
//Функция, возвращающая для указанной модели Lambda
//вероятность появления заданной последовательности Seq длиной tau,
```

```
//содержащей номера вариантов наблюдений, методом обратного хода
double Seq_Probability_Backward(HMM* Lambda, int* Seq, int tau)
{
double ret=0;
double *beta=(double*)malloc((Lambda->N)*sizeof(double));
double *beta_prev=(double*)malloc((Lambda->N)*sizeof(double));
for(int i=0;i<(Lambda->N);i++)
beta[i]=1;
for(int t=tau-1;t>0;t--)
{
for(i=0;iN;i++)
{
beta_prev[i]=0;
for(int j=0;iN;i++)
beta_prev[i]+=Lambda->A[i][j]*Lambda->B[j][t]*beta[i];
}
free(beta);
beta=beta_prev;
beta_prev=(double*)malloc((Lambda->N)*sizeof(double));
}
return(ret);
}
//Функция, возвращающая для указанной модели Lambda наиболее
// вероятную цепочку номеров событий
//по заданной последовательности Seq длиной tau,
//содержащей номера вариантов наблюдений, по алгоритму Витерби
int* Viterbi(HMM* Lambda, int* Seq, int tau)
{
int *ret=(int*)malloc(tau*sizeof(int));
double **gamma=(double**)malloc(tau*sizeof(double*));
```

```
int **fi=(int**)malloc(tau*sizeof(int*));
for(int t=0;t
{
gamma[t]=(double*)malloc((Lambda->N)*sizeof(double));
fi[t]=(int*)malloc((Lambda->N)*sizeof(int));
}
for(int i=0;i<(Lambda->N);i++)
{
gamma[0][i]=(Lambda->pi[i])*(Lambda->B[i][0]);
fi[0][i]=0;
}
for(t=1;t
{
for(int j=0;j<(Lambda->N);j++)
{
double Max=gamma[t-1][0]*(Lambda->A[0][j]);
int Max_ind=0;
for(i=0;i<(Lambda->N);i++)
{
if(gamma[t-1][i]*(Lambda->A[i][j])>Max)
{
Max=gamma[t-1][i]*(Lambda->A[i][j]);
Max_ind=i;
}
}
gamma[t][j]=Max;
fi[t][j]=Max_ind;
}
}
double P=gamma[tau][0];
```

```
int P_arg=0;
for(i=0;i<(Lambda->N);i++)
{
if(gamma[tau-1][i]>P)
{
P=gamma[tau-1][i];
P_arg=i;
}
}
ret[tau-1]=P_arg;
for(t=t-2;t>=0;t--)
{
ret[t]=fi[t+1][ret[t+1]];
}
return(ret);
}
//Функция, настраивающая заданную модель Lambda
//по заданной последовательности Seq длиной tau,
//содержащей номера вариантов наблюдений, по алгоритму Баума-Уэлша
void Baum_Welsh(HMM* Lambda, int* Seq, int tau)
{
// Инициализация вспомогательных переменных
double ***ksi=(double***)malloc((Lambda->N)*sizeof(double**));
double **alpha=(double**)malloc(tau*sizeof(double*));
double **beta=(double**)malloc(tau*sizeof(double*));
double **psi=(double**)malloc((Lambda->N)*sizeof(double*));
for(int t=0;t
{
ksi[t]=(double**)malloc((Lambda->N)*sizeof(double*));
psi[t]=(double*)malloc((Lambda->N)*sizeof(double));
```

```
for(int i=0;i<(Lambda->N);i++)
{
ksi[t][i]=(double*)malloc((Lambda->N)*sizeof(double));
}
}
for(t=0;t
{
alpha[t]=(double*)malloc((Lambda->N)*sizeof(double));
beta[t]=(double*)malloc((Lambda->N)*sizeof(double));
}
//Рассчитываем вспомогательные коэффициенты Альфа
for(int i=0;i<(Lambda->N);i++)
alpha[0][i]=Lambda->pi[i]*Lambda->B[i][0];
for(t=0;t
{
for(int j=0;jN;j++)
{
alpha[t+1][j]=0;
for(i=0;iN;i++)
alpha[t+1][j]+=alpha[t][i]*Lambda->A[i][j];
alpha[t+1][j]*=Lambda->B[j][Seq[t]];
}
}
//Рассчитываем вспомогательные коэффициенты Бета
for(i=0;i<(Lambda->N);i++)
beta[tau-1][i]=1;
for(t=tau-1;t>0;t--)
{
for(i=0;iN;i++)
{
```

```
beta[t-1][i]=0;
for(int j=0;iN;i++)
beta[t-1][i]+=Lambda->A[i][j]*Lambda->B[j][t]*beta[t][i];
}
}
//Рассчитываем вспомогательные переменные Кси
for(t=0;t
{
for(i=0;i<(Lambda->N);i++)
{
for(int j=0;j<(Lambda->N);j++)
{
double Sum=0;
for(int i2=0;i2<(Lambda->N);i2++)
{
for(int j2=0;j2<(Lambda->N);j2++)
{
Sum+=alpha[t][i2]*
Lambda- >A[i2][j2]*Lambda->B[j2][t+1]*
beta[t+1][j2];
}
}
ksi[t][i][j]=alpha[t][i]*Lambda->A[i][j]*Lambda->B[j][t+1]*beta[t+1][j]/Sum;
}
}
}
//Рассчитываем вспомогательные переменные Пси
for(t=0;t
{
for(i=0;i<(Lambda->N);i++)
```

```
{
double Sum=0;
for(int j=0;j<(Lambda->N);j++)
{
Sum+=ksi[t][i][j];
}
psi[t][i]=Sum;
}
}
//Вычисляем уточненные параметры марковской модели
for(i=0;i<(Lambda->N);i++)
{
Lambda->pi[i]=psi[0][i];
}
for(i=0;i<(Lambda->N);i++)
for(int j=0;j<(Lambda->N);j++)
{
double Sum1=0;
double Sum2=0;
for(t=0;t
{
Sum1+=ksi[t][i][j];
Sum2+=psi[t][j];
}
Lambda->A[i][j]=Sum1/Sum2;
}
for(int j=0;j<(Lambda->N);j++)
for(int k=0;k<(Lambda->M);k++)
{
double Sum1=0;
```

```
double Sum2=0;
for(t=0;t
{
Sum2+=psi[t][j];
if(Seq[t]==k)
{
Sum1+=psi[t][j];
}
}
Lambda->B[i][j]=Sum1/Sum2;
}
}
```

3.5. Алгоритм вычисления кепстральных коэффициентов

Задача распознавания отдельных слов речи. Цифровая система обработки звукового сигнала предполагает представление аналогового речевого сигнала в цифровом виде. В результате аналого-цифрового преобразования (АЦП) непрерывный сигнал переводится в ряд дискретных временных отсчетов, каждый из которых представляет собой число. Это число храктеризует сигнал в точке с определенной точностью. Точность представления зависит от ширины диапазона получаемых чисел, а, следовательно, от разрядности АЦП. Процесс извлечения из сигнала численных значений называется квантованием. Процесс разбиения сигнала на отсчеты – дискретизацией. Число отсчетов в секунду называется частотой дискретизации. Процесс обработки звуковой волны схематически показан на рис.3.12.

Аналоговый акустический сигнал, поступающий с микрофона, подвергается с помощью АЦП дискретизации и квантованию. Происходит

так называемая **реализация слова,** т. е. цифровая запись произнесения слова (звука) в виде последовательности отсчётов звукового сигнала $\{s_k\}$. Реализация слова (звука) в процессе цифровой обработки разбивается на последовательность кадров $\{X_i\}$. **Кадром** X (длины N) назовем последовательность отсчетов звукового сигнала s_1, s_2, ... , s_N. Длина кадра фиксирована во времени. Например, при$N=100$ и частоте дискретизации *8000 Гц* она соответствует длительности в *12.5 мс*. Кадры часто смещают друг относительно друга для того, чтобы не происходило потери информации на границе кадров. **Шаг смещения кадра** – количество звуковых отсчётов между началами следующих друг за другом кадров. Шаг смещения меньший, чем N (длина кадра) означает, что кадры идут «внахлёст».

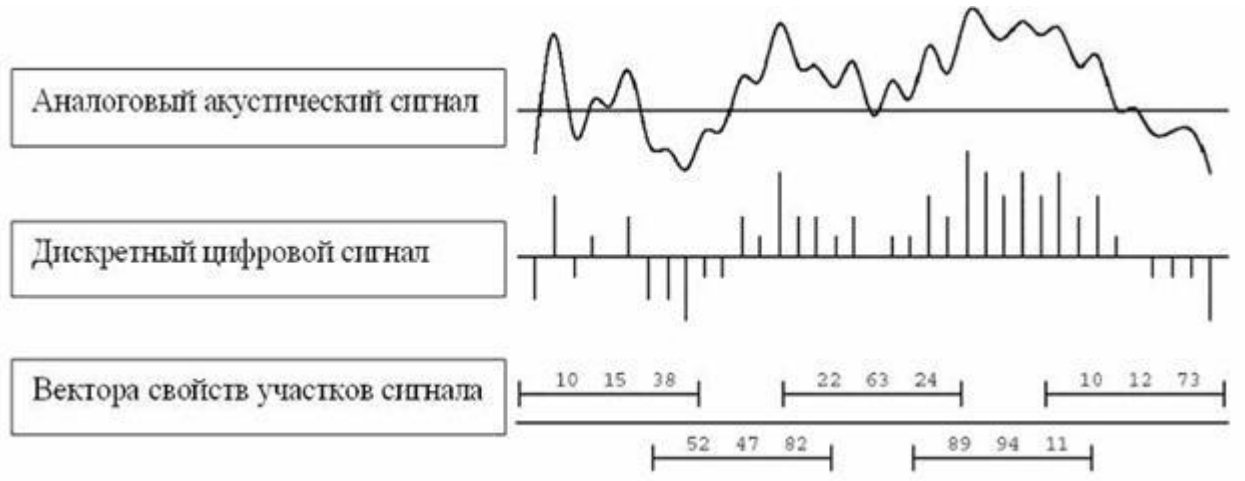


Рис.3.12. Этапы обработки звуковой волны.

Далее в целом ряде задач, таких как распознавание слов речи или идентификации личности, каждому кадру сопоставляются некоторые данные, характеризующие звук наилучшим образом. Такие данные формируют **вектор свойств** (или вектор признаков). С математической точки зрения это может быть как вектор из пространства R^M, так и набор функций или одна функция.

Задачей системы является отождествление каждого слова, поступающего на вход, с заранее определенным классом. К сожалению,

существует целое множество различных факторов **распознавания отдельных слов** речи, которые могут оказывать негативное влияние на точность распознающей системы – настроение и состояние говорящего, шум окружающей среды, скорость произнесения фраз.

Распознающая система является независимой от диктора, если она распознает слово независимо от того, кто его произносит. На практике реализовать такую систему сложно по той причине, что звуковые сигналы сильно зависят от громкости, тембра голоса, состояния и настроения диктора. Для извлечения информации из таких сигналов нередко используют фильтры тоновых частот (мел-скейл фильтры), которые усредняют спектральные составляющие в определенных диапазонах частот, тем самым делая сигнал менее зависимым от диктора. Такие фильтры являются основой технологии MFCC (Mel-Frequency Cepstral Coefficients), которая используется в распознающей системе, рассматриваемой в этой лекции.

Вопросы предварительной фильтрации, сегментации сигнала на отрезки равной длины (окна преобразования), обработки сигнала в окне были ранее рассмотрены в предыдущих разделах. Рассмотрим алгоритм извлечения векторов свойств (или вектора признаков), определенным образом характеризующего сигнал. В рассматриваемой далее модели используется классический подход кепстральных коэффициентов.

Существует две основных технологии извлечения из сигнала вектора свойств, состоящего из кепстральных коэффициентов: на основе кепстральных коэффициентов тональной частоты (MFCC) и на основе кепстральных коеффициентов линейного предсказания (LPCC). MFCC является наиболее распространенным способом формирования вектора свойств. Рассмотрим основные его этапы.

1. Входной сигнал разбивается на сегменты, к которым применяется функция окна Хемминга и фразового выделения.

2. *Pre-emphasis* - предварительное выделение фразы (или акцентирование) за счет фильтрации звукового сигнала. Этот шаг вызван необходимостью

спектрального сглаживания сигнала, который становится менее восприимчивым к различным шумам, возникающим в процессе обработки.

3. Далее изучают спектрограмму сигнала. Все множество присутствующих в спектрограмме частот разделяется на пронумерованные интервалы, каждому из которых определяется свой диапазон. Для каждого такого интервала подсчитывается среднее значение интенсивности сигнала в выделенном диапазоне и строится диаграмма, где ось абсцисс состоит из номеров интервалов, а ординат из "усиленных" амплитуд (значения амплитуд возводятся в квадрат, чтобы не было отрицательных величин при дальнейшей операции логарифмирования). Этот процесс называется мел-скейл фильтрацией.

4. Далее амплитуды сигнала сжимаются с помощью применения логарифма, поскольку человеческое ухо воспринимает громкость сигналов по логарифмической шкале, а вектора свойств получают на основе человеческого восприятия звука.

5. Заключительным шагом является применение к спектру обратного преобразования Фурье. Результатом этого шага является выделение кепстральных коэффициентов, которые формируют вектор свойств данного сегмента.

Кепстральные коэффициенты математически могут быть описаны следующим образом:

$$c_n = \sum_{k=1}^{K} (\log S(k)) e^{ikn},$$

где *S(k)* - усредненный спектр сигнала усиленной интенсивности, характерный для *k*–ого частотного интервала (бенда) в мел-скейл фильтре; *K* - общее количество интервалов, на которые разбивается спектр.

3.6. Алгоритм динамической трансформации

Алгоритм динамической трансформации временной шкалы (DTW-алгоритм, от англ. dynamic time warping) — алгоритм, позволяющий найти оптимальное соответствие между временными последовательностями. Впервые применен в распознавании речи, где использован для определения того, как два речевых сигнала представляют одну и ту же исходную произнесённую фразу. Впоследствии были найдены применения и в других областях.

Временные ряды — широко распространенный тип данных, встречающийся, фактически, в любой научной области, и сравнение двух последовательностей является стандартной задачей. Для вычисления отклонения бывает достаточно простого измерения расстояния между компонентами двух последовательностей (евклидово расстояние). Однако часто две последовательности имеют приблизительно одинаковые общие формы, но эти формы не выровнены по оси X. Чтобы определить подобие между такими последовательностями, мы должны «деформировать» ось времени одной (или обеих) последовательности, чтобы достигнуть лучшего выравнивания.

Алгоритм динамической трансформации. Измерение расстояния между двумя временными рядами нужно для того, чтобы определить их подобие и классификацию. Таким эффективным измерением является евклидова метрика. Для двух временных последовательностей это просто сумма квадратов расстояний от каждой n-ой точки одной последовательности до n-ой точки другой. Однако использование евклидова расстояния имеет существенный недостаток: если два временных ряда одинаковы, но один из них незначительно смещен во времени (вдоль оси времени), то евклидова метрика может посчитать, что ряды отличаются друг от друга. DTW-алгоритм был введён для того, чтобы преодолеть этот недостаток и предоставить наглядное измерение расстояния между рядами, не обращая внимание как на глобальные, так и на локальные сдвиги на временной шкале.

Классический алгоритм.

Рассмотрим два временных ряда — Q длины n и C длины m [57]:

$$Q = q_1, q_2, \ldots, q_i, \ldots, q_n; \quad (3.28)$$

$$C = c_1, c_2, \ldots, c_j, \ldots, c_m. \quad (3.29)$$

Первый этап алгоритма состоит в следующем. Строится матрица d порядка $n \times m$ (матрицу расстояний), в которой элемент $n \times m$ есть расстояние $d(q_i, c_j)$ между двумя точками q_i и c_j. Обычно используется евклидово расстояние: $d(q_i, c_j) = (q_i - c_j)^2$, или $d(q_i, c_j) = |q_i - c_j|$. Каждый элемент (i, j) матрицы соответствует выравниванию между точками q_i и c_j.

На втором этапе строится матрица трансформаций (деформаций) D, каждый элемент которой вычисляется исходя из следующего соотношения:

$$D_{i\ j} = d_{i\ j} + min(D_{i-1\ j}, D_{i-1\ j-1}, D_{i\ j-1}). \quad (3.30)$$

После заполнения матрицы трансформации, переходится к заключительному этапу, который заключается в том, чтобы построить некоторый оптимальный путь трансформации (деформации) и DTW расстояние (*стоимость пути*).

Путь трансформации W — это набор смежных элементов матрицы, который устанавливает соответствие между Q и C. Он представляет собой путь, который минимизирует общее расстояние между Q и C. k -ый элемент пути W определяется как $w_k = (i, j)_k, \quad d(w_k) = d(q_i, c_j) = (q_i - c_j)^2$. Таким образом:

$$W = w_1, w_2, \ldots, w_k, \ldots, w_K; \qquad max(m, n) \leqslant K < m + n,$$

где K — длина пути.

Путь трансформации должен удовлетворять следующим ограничивающим условиям:

- **Граничные условия**: начало пути $w_1 = (1, 1)$, его конец — $w_K = (n, m)$. Это ограничение гарантирует, что путь трансформации содержит все точки обоих временных рядов.
- **Непрерывность** (условие на длину шага): любые два смежных элемента пути W, $w_k = (w_i, w_j)$ и $w_{k+1} = (w_{i+1}, w_{j+1})$, удовлетворяют

следующим неравенствам: $w_i - w_{i-1} \geqslant 0$, $w_j - w_{j-1} \geqslant 0$. Это ограничение гарантирует, что путь трансформации передвигается на один шаг за один раз. То есть оба индекса *i* и *j* могут увеличиться только на 1 на каждом шаге пути.

- **Монотонность**: любые два смежных элемента пути W, $w_k = (w_i, w_j)$ и $w_{k+1} = (w_{i+1}, w_{j+1})$, удовлетворяют следующим неравенствам: $w_j - w_{j+1} \leqslant 1$, $w_j - w_{j-1} \geqslant 0$. Это ограничение гарантирует, что путь трансформации не будет возвращаться назад к пройденной точке. То есть оба индекса *i* и *j* либо остаются неизменными, либо увеличиваются (но никогда не уменьшаются).

Хотя существует большое количество путей трансформации, удовлетворяющих всем вышеуказанным условиям, однако нас интересуют только тот путь, который минимизирует DTW расстояние.

DTW расстояние между двумя последовательностями рассчитывается на основе оптимального пути трансформации с помощью формулы:

$$DTW(Q,C) = min\left\{\frac{\sum_{k=1}^{K} d(w_k)}{K}\right\}. \qquad (3.31)$$

K в знаменателе используется для учёта того, что пути трансформации могут быть различной длины.

Пространственная и временная сложность алгоритма — квадратичная, K, так как DTW алгоритм должен изучить каждую клетку матрицы трансформации.

Недостатки алгоритма.

Хотя алгоритм успешно используется во многих областях, он может выдавать неправильные результаты. Алгоритм может попытаться объяснить непостоянство оси *y* с помощью трансформации оси *x*. Это может привести к выравниванию, при котором одной точке первой последовательности

ставится в соответствие большая подгруппа точек второй последовательности.

Другая проблема заключается в том, что алгоритм может не найти очевидное выравнивание двух рядов вследствие того, что особая точка (пик, впадина, плато, точка перегиба) одного ряда расположена немного выше или ниже соответствующей ей особой точки другого ряда.

Разновидности DTW алгоритма.

Различные доработки DTW алгоритма предназначены для ускорения его вычислений, а также для того, чтобы лучше контролировать возможные маршруты путей трансформации.

Общие (глобальные) ограничения.

Один из распространенных вариантов DTW алгоритма — наложение общих (глобальных) ограничивающих условий на допустимые пути деформации. Пусть $R \subseteq [1:n] \times [1:m]$ — подмножество, задающее область глобального ограничения. Теперь путём трансформации является путь, который содержится в области R. Оптимальный путь трансформации — путь, принадлежащий R, и минимизирующий стоимость пути среди всех путей трансформации из R.

Быстрый DTW-алгоритм. Этот алгоритм обладает линейной пространственной и временной сложностью. Быстрый DTW алгоритм использует многоуровневый подход с тремя ключевыми операциями:

1. Уменьшение детализации — уменьшаем размер временного ряда с помощью усреднения смежных пар точек. Полученный временной ряд — это ряд, имеющий в два раза меньше точек, чем исходный. Мы проводим эту операцию несколько раз, чтобы получить много различных разрешений временного ряда.
2. Планирование — берем путь трансформации при низкой детализации и определяем через какие клетки будет проходить путь трансформации

при следующей детализации (на порядок выше предыдущей). Так как разрешение увеличивается в два раза, одной точке, принадлежащей пути трансформации в низком разрешении, будут соответствовать, по крайней мере, четыре точки в большем разрешении. Затем этот планируемый путь используется в качестве эвристического правила в процессе обработки, чтобы найти путь в высоком разрешении.

3. Обработка — поиск оптимального пути деформации в окрестности спланированного пути.

Разреженный DTW-алгоритм.

Основная идея данного метода состоит в том, чтобы динамически использовать наличие подобия и/или сопоставления данных для двух временных последовательностей. Данный алгоритм имеет три особых преимущества:

1. Матрица трансформации представляется с помощью разреженных матриц, что приводит к уменьшению средней пространственной сложности по сравнению с другими методами.
2. Разреженный DTW алгоритм всегда выдает оптимальный путь трансформации.
3. Так как алгоритм выдает оптимальное выравнивание, то он может быть легко использован в сочетании с другими методами.

Области применения: распознавание речи, интеллектуальный анализ данных, распознавание жестов, робототехника, медицина, биоинформатика, верификация подписи.

3.8. Динамическое программирование в алгоритмах распознавания речи

В системах распознавания речи, содержащих слова, распознавание требует сравнения между входным словом и различными словами в словаре. Эффективное решение проблемы лежит в динамических алгоритмах

сравнения, целью которого является введение временных масштабов двух слов в оптимальное соответствие. Алгоритмы такого типа являются динамическими алгоритмами трансформации временной шкалы. В данной параграфе представлено два варианта реализации алгоритма предназначенные для распознавания отдельных слов.

Исследования в области распознавания речи, так же как и в других областях, следуют по двум направлениям: фундаментальные исследования, целью которых является разработка и тестирование новых методов, алгоритмов и концепций на некоммерческой основе; и прикладные исследования, целью которых является улучшение существующих методов, следуя определенным критериям.

Фундаментальные исследования направлены на получение среднесрочной или долгосрочной выгоды, в то время как прикладные исследования направлены на быстрое улучшение существующих методов или расширения их использования в областях, где такие методы еще практически не используются.

Улучшить скорость распознавания речи можно при учете следующих критериев:

- размер узнаваемой лексики;
- степень спонтанности речи, которую необходимо распознать;
- зависимость/независимость от диктора;
- время, необходимое для приведения системы в движение;
- время приспособления системы для новых пользователей;
- время выбора и распознавания;
- степень распознавания (выраженная словом или предложением).

Сегодня системы распознавания звука строятся на основе принципов признания форм распознавания. Методы и алгоритмы, которые использовались до сих пор, могут быть разделены на четыре больших класса:

- методы дискриминатного анализа, основанные на Байесовской дискриминации;

– скрытые модели Маркова;
– динамическое программирование – временные динамические алгоритмы (DTW);
– нейронные сети.

Алгоритм динамического трансформирования времени (DTW) вычисляет оптимальную последовательность трансформации (деформации) времени между двумя временными рядами. Алгоритм вычисляет оба значения деформации между двумя рядами и расстоянием между ними.

Предположим, что у нас есть две числовые последовательности (a_1, a_2, ..., a_n) и (b_1, b_2, ..., b_m). Как видим, длина двух последовательностей может быть различной. Алгоритм начинается с расчета локальных отклонений между элементами двух последовательностей, использующих различные типы отклонений. Самый распространенный способ для вычисления отклонений является метод, рассчитывающий абсолютное отклонение между значениями двух элементов (Евклидово расстояние). В результате получаем матрицу отклонений, имеющую n строк и m столбцов общих членов:

$$d_{ij} = \left|a_i - b_j\right|,\ i = \overline{1,n},\ j = \overline{1,m}. \qquad (3.32)$$

Минимальное расстояние в матрице между последовательностями определяется при помощи алгоритма динамического программирования и следующего критерия оптимизации:

$$a_{ij} = d_{ij} + \min(a_{i-1,j-1}, a_{i-1,j}, a_{i,j-1}), \qquad (3.33)$$

где: a_{ij} — минимальное расстояние между последовательностями (a_1, a_2, ..., a_n) и (b_1, b_2, ..., b_m). Путь деформации – это минимальное расстояние в

матрице между элементами a_{11} и a_{nm}, состоящими из тех a_{ij} элементами, которые выражают расстояние до a_{nm}.

Глобальные деформации состоят из двух последовательностей и определяются по следующей формуле:

$$\mathrm{GC} = \frac{1}{p}\sum_{i=1}^{p} w_i \,, \qquad (3.34)$$

где: w_i – элементы, которые принадлежать пути деформации; p – их количество. Расчеты производились для двух коротких последовательностей и указаны в таблице, в которой выделена последовательность деформации.

	-2	10	-10	15	-13	20	-5	14	2
3	5	12	25	37	53	70	78	89	90
-13	16	28	15	43	37	70	78	105	104
14	32	20	39	16	43	43	62	62	74
-7	37	37	23	38	22	49	45	66	71
9	48	38	42	29	44	33	47	50	57
-2	48	50	46	46	40	55	36	52	**54**

Существует три условия, налагаемых на DTW алгоритм для обеспечения быстрой конвергенции:

1. Монотонность – путь никогда не возвращается, то есть: оба индекса, i и j, которые используются в последовательности, никогда не уменьшаются.
2. Непрерывность – последовательность продвигается постепенно: за один шаг индексы, i и j, увеличиваются не более чем на 1.
3. Предельность – последовательность начинается в левом нижнем углу и заканчивается в правом верхнем.

Пример деформации последовательности с использованием языка программирования Java приведен ниже:

```
public static void dtw(double a[],double b[],double dw[][], Stack<Double> w){
        // a,b - the sequences, dw - the minimal distances matrix
        // w - the warping path
        int n=a.length,m=b.length;
        double d[][]=new double[n][m]; // the euclidian distances matrix
        for(int i=0;i<n;i++)
                for(int j=0;j<m;j++)d[i][j]=Math.abs(a[i]-b[j]);
        // determinate of minimal distance
        dw[0][0]=d[0][0];
        for(int i=1;i<n;i++)dw[i][0]=d[i][0]+dw[i-1][0];
        for(int j=1;j<m;j++)dw[0][j]=d[0][j]+dw[0][j-1];
        for(int i=1;i<n;i++)
                for(int j=1;j<m;j++)
                        if(dw[i-1][j-1]<=dw[i-1][j])
                                if(dw[i-1][j-1]<=dw[i][j-1])dw[i][j]=d[i][j]+dw[i-
1][j-1];
                                else dw[i][j]=d[i][j]+dw[i][j-1];
                        else
                                if(dw[i-1][j]<=dw[i][j-1])dw[i][j]=d[i][j]+dw[i-
1][j];
                                else dw[i][j]=d[i][j]+dw[i][j-1];
        int i=n-1,j=m-1;
        double element=dw[i][j];
        // determinate of warping path
        w.push(new Double(dw[i][j]));
        do{
                if(i>0&&j>0)
                        if(dw[i-1][j-1]<=dw[i-1][j])
                                if(dw[i-1][j-1]<=dw[i][j-1]){i--;j--;} else j--;
                        else
```

```
                    if(dw[i-1][j]<=dw[i][j-1])i--; else j--;
            else if(i==0)j--; else i--;
            w.push(new Double(dw[i][j]));
        }
        while(i!=0||j!=0);
}
```

Так как для определения основы последовательности в динамическом программировании оптимальным является использование метод обратного программирования, необходимо использовать определенный динамический тип структуры, который называется «стек». Подобно любому динамическому алгоритму программирования, DWT имеет полиноминальную сложность. Когда мы имеем дело с большими последовательностями, возникают два неудобства:

- запоминание больших числовых матриц;
- выполнение большого количества расчетов отклонений.

Существует улучшенная версия алгоритма, FastDWT, которая решает две вышеуказанные проблемы. Решение заключается в разбиении матрицы состояний на 2, 4, 8, 16 и т.д. меньших по размеру матриц, посредством повторяющегося процесса разбиения последовательности ввода на две части. Таким образом, расчеты отклонения производятся только на этих небольших матрицах, и путях деформации, рассчитанных для небольших матриц. С алгоритмической точки зрения, предлагаемое решение основано на методе “Divide et Impera” (прим. пер. от лат. «Разделяй и влавствуй»).

Использование DWT алгоритма в распознавании речи.

Анализ голосового сигнала. Звук проходит через среду, как продольная волна со скоростью, зависящей от плотности среды. Самый простой способ представления звуков – синусоидный график, графическое

представление вибраций воздуха под давлением на протяжении некоторого времени.

Форма звуковой волны зависит от трех факторов: аплитуды, частоты и фазы. Напомним их основные свойства.

Амплитуда — перемещение синусоидальных графов выше и ниже временной оси (y = 0), что соответствует энергии загруженной звуковой волны. Измерение амплитуды может быть произведено в единицах давления (децибелах DB), которые измеряют амплитуду обычного звука при помощи логарифмических функций. Измерение амплитуды используя децибелы очень важно на практике, так как это прямое представление о том, как громкость звука воспринимается людьми.

Частота – число циклов синусоиды за одну секунду. Цикл колебаний начинается со средней линией, потом достигает максимума и минимума, а после возвращается к средней линии.

Последний фактор – фаза. Она измеряет положение относительно начала синусоидальной кривой. Фаза не может быть услышана человеком, однако ее можно определить относительно положения между двумя сигналами. Тем не менее, слуховой аппарат воспринимает положение звука на разных фазах.

Для того чтобы разобрать звуковые волны на синусоидальной кривой пользуются разложением Фурье, его результатом является набором амплитуд, фаз и частот для каждого синусоидального компонента волны. Складывая эти синусоидальные кривые вместе, получаем оригинальную звуковую волну. Точка частоты или фазы, взятая вместе с амплитудой, называется спектром. Спектр показывает частоту короткой последовательности звуков, и если мы хотим проанализировать ее развитие в течение времени, необходимо найти способ, позволяющий продемонстрировать это. Это можно показать на спектрограмме. Спектрограмма — это диаграмма в двух измерениях: частота и время, — в

которой цвет точки (темный – сильный, светлый – слабый) определяет амплитуду интенсивности.

Выявление слов. Современные методы выявления могут точно определить начальную и конечную точку произнесенного слова в звуковом потоке, на основе обработки сигналов, меняющихся в течение времени. Данные методы оценивают энергию и среднюю величину в коротком отрезке времени, а также вычисляют средний уровень пересечения нуля.

Создание начальной и конечной точки – простая задач, если аудиозапись сделана в идеальных условиях. В этом случае отношение сигнал-шум велик, так как определить действительный сигнал в потоке путем анализа образов не представляет труда. В реальных условиях все не так просто: фоновый шум имеет огромную интенсивность и может нарушить процесс отделения слов в потоке речи.

Простейший алгоритм отделения слов — алгоритм Рабинера-Ламеля. Если рассматривать стоб-импульсов $\{s_1, s_2, \ldots, s_n\}$, где n – число образов строб-импульсов, а si, i=1, n – численное выражение образцов, общая энергия строб-импульсов вычисляется:

$$E(n) = \frac{1}{n}\sum_{i=1}^{n} s_i^{\,2}. \qquad (3.35)$$

Средний уровень пересечения нулевого уровня:

$$ZCR(n) = \sum_{i=1}^{n-1} sign(s_i) \cdot sign(s_{i+1}), \qquad (3.36)$$

где:

$$sign(s_i) = \begin{cases} 1 & \text{if } s_i > 0 \\ 0 & \text{if } s_i < 0 \end{cases}.$$

Метод использует три числовых уровня: два для энергии (верхний, нижний) и один для среднего пересечения нулевого уровня. Точка, начиная с

которой энергия перекрывает верхний уровень и уровень положительных и отрицательных значений, не отменяет установленный уровень, который считается отправной точкой голосового звучания (не тишины). Поиск первой такой точки производится путем скрещивания импульсов от начала и до конца, и это определит первую область с речью. Обратный переход, из конца в начало, позволяет определить конечную точку последней области с речью. Определение внутри области может быть сделано путем скрещивания импульсов между двумя этими точками. Начало глухой области начинается в точке, в которой энергия становится меньше значения нижнего уровня.

Это можно рассмотреть на рис.3.13, на котором показана запись звукового сигнала до и после удаления глухой области.

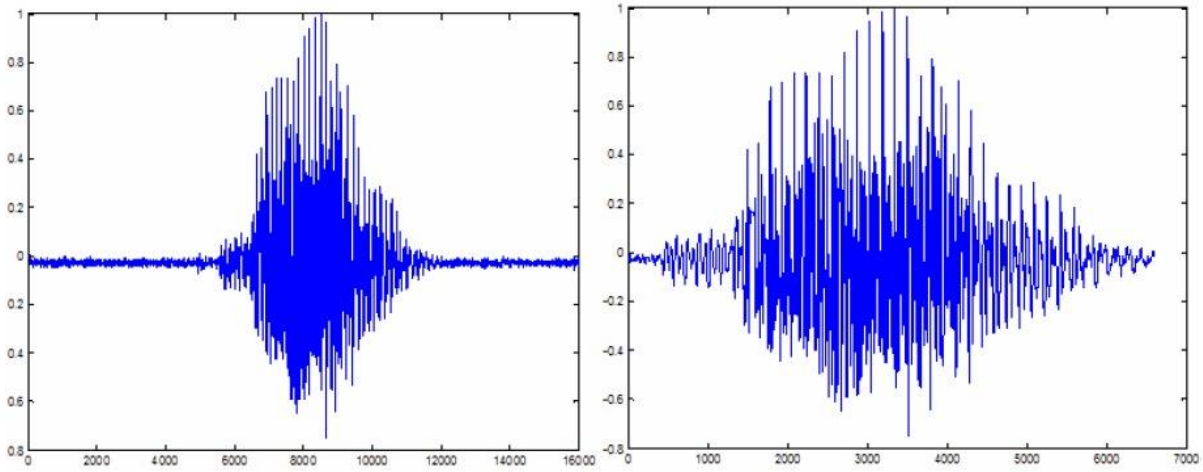


Рис.3.13. Звуковой сигнал слова «нова»

Определение слов с использованием алгоритма DWT. Определение слова может осуществляться путем сравнения числовых форм сигналов или путем сравнения спектрограммы сигналов. Процесс сравнения в обоих случаях должен компенсировать различные длины последовательности и нелинейный характер звука. DWT алгоритму удается разобрать эти проблемы путем нахождения деформации, соответствующей оптимальному расстоянию между двумя рядами различной длины.

Существуют 2 особенности применения алгоритма:

1. Прямое сравнение числовых форм сигналов. В этом случае, для каждой числовой последовательности создается новая последовательность, размеры которой значительно меньше. Алгоритм имеет дело с этими последовательностями. Числовая последовательность может иметь несколько тысяч числовых значений, в то время как подпоследовательность может иметь несколько сотен значений. Уменьшение количества числовых значений может быть выполнено путем их удаления между угловыми точками. Этот процесс сокращения длины числовой последовательности не должен изменять своего представления. Несомненно, процесс приводит к уменьшению точности распознавания. Однако, принимая во внимание увеличение скорости, точность, по сути, повышается за счет увеличения слов в словаре.
2. Представление сигналов спектрограмм и применение алгоритма DTW для сравнения двух спектрограмм. Метод заключается в разделении цифрового сигнала на некоторое количество интервалов, которые будут перекрываться. Для каждого импульса, интервалы действительных чисел (звуковых частот), будет рассчитывать Быстрым преобразование Фурье, и будет храниться в матрице звуковой спектрограммы. Параметры будут одинаковыми для всех вычислительных операций: длин импульса, длины преобразования Фурье, длины перекрытия для двух последовательных импульсов. Преобразование Фурье является симметрично связанным с центром, а комплексные число с одной стороны связаны с числами с другой стороны. В связи с этим, только значения из первой части симметрии можно сохранить, таким образом, спектрограмма будет представлять матрицу комплексных чисел, количество линий в такой матрице является равной половине длины преобразования Фурье, а количество столбцов будет определяться в зависимости от длины звука. DTW будет применяться на матрице

вещественных чисел в результате сопряжения спектрограммы значений, такая матрица называется матрицей энергии.

DTW алгоритмы являются очень полезными для распознавания отдельных слов в ограниченном словаре. Для распознавания беглой речи используются скрытые модели Маркова. Использование динамического программирования обеспечивает полиминальную сложность алгоритма: n^2v, где n – длина последовательности, а v количество слов в словаре. DWT имеют несколько слабых сторон. Во-первых, n^2v сложность не удовлетворяет большим словарям, которые увеличивают успешность процесса распознавания. Во-вторых, трудно вычислить два элемента в двух разных последовательностях, если принять во внимание, что существует множество каналов с различными характеристиками. Тем не менее, DTW остается простым в реализации алгоритмом, открытым для улучшений и подходящим для приложений, которым требуется простое распознавание слов: телефоны, автомобильные компьютеры, системы безопасности и т.д.

3.9. Качество распознавания и синтеза речи

В настоящее время общество вкладывает гигантское количество денег в развитие know-how и научно-исследовательские разработки для решения проблем автоматического распознавания и понимания речи. Это стимулируется практическими требованиями, связанными с созданием системы военного и коммерческого назначения. При этом следует обратить внимание на то, что в практическом использовании отсутствуют системы, считающиеся по непонятным причинам вершиной развития систем автоматического распознавания речи. Это системы, которые можно назвать демонстрационными и которые 50 лет назад назывались «фонетическими печатающими машинками». Их целью является перевод речи в соответствующий письменный текст.

Выделяют несколько основных способов распознавания речи:

Распознавание отдельных команд. Суть технологии*:* раздельное произнесение и последующее распознавание слова или словосочетания из небольшого заранее заданного словаря.

Техническая реализация*:* точность распознавания ограничена объемом заданного словаря. При соблюдении этого условия данная технология позволяет достичь самой высокой достоверности распознавания.

Применение*:* в настоящее время наиболее ярким примером использования технологии распознавания отдельных команд в коммерческих приложениях является голосовая навигация по сайтам.

Распознавание по грамматике. Суть технологии*:* распознавание фраз, соответствующих определенным заданным правилам (грамматике).

Техническая реализация*:* для задания грамматик используются стандартные XML-языки (VoiceXML), обмен данными между системой распознавания и приложением, как правило, осуществляется по протоколу MRCP.

Применение: технология распознавания по грамматике широко применяется в системах голосового самообслуживания.

Поиск ключевых слов в потоке слитной речи. Суть технологии*:* распознавание отдельных участков речи.

Техническая реализация*:* в этом случае речь может быть, как спонтанной, так и соответствующей определённым правилам. Произнесенная речь не полностью преобразуется в текст - в ней автоматически находятся лишь те участки, которые содержат заданные слова или словосочетания.

Применение*:* данная технология распознавания часто применяется в поисковых системах, в системах мониторинга речи.

Распознавание слитной речи на большом словаре (LVCSR — large vocabulary continuous speech recognition). Суть технологии*:* эта технология наиболее близка к мечте человека о взаимодействии человека и машины – все, что сказано, дословно преобразуется в текст. Поэтому иногда эта технология так и называется STT – speech-to-text.

Техническая реализация: задача полноценного распознавания слитной речи не решена нигде в мире, однако, достоверность распознавания уже достаточно высока для использования технологии на практике (рис.3.14).

Так как последний метод наиболее сложный, то рассмотрим его поподробнее.

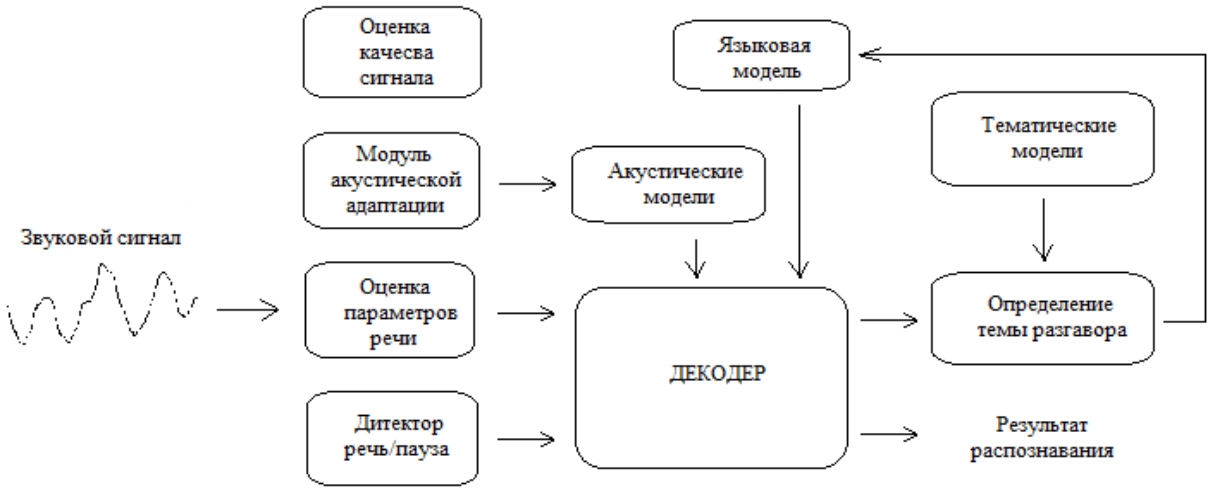


Рис. 3.14. Процесс распознавания слитной речи на большом словаре

Этапы распознавания.

1. Обработка речи начинается с оценки качества речевого сигнала. На этом этапе определяется уровень помех и искажений.

2. Результат оценки поступает в модуль акустической адаптации, который управляет модулем расчета параметров речи, необходимых для распознавания.

3. В сигнале выделяются участки, содержащие речь, и происходит оценка параметров речи.

4. Далее параметры речи поступают в основной блок системы распознавания – декодер. Это компонент, который сопоставляет входной речевой поток с информацией, хранящейся в акустических и языковых моделях, и определяет наиболее вероятную последовательность слов, которая и является **конечным результатом распознавания.**

а) **Акустические модели.** При сравнительно небольшом рабочем словаре высокой достоверности распознавания можно достигнуть, лишь сопоставляя входной поток речи с шаблонами отдельных звуков – акустическими моделями. Современная тенденция технологии описания звуковых образов подразумевает комбинирование различных подходов. Так, в Центре речевых технологий (Россия) для описания акустических моделей используют комбинацию классической теории цифровой обработки сигналов и технологии искусственных нейронных сетей. Такие модели наиболее устойчивы к междикторской вариативности, а также к помехам и искажениям, вносимым окружением или каналом передачи.

б) **Языковые модели.** С ростом словаря увеличивается количество слов, схожих или даже одинаковых по звучанию. При слитном произнесении акустическая схожесть отдельных фрагментов речи проявляется настолько, что часто и человек, прослушивая запись вне контекста, не может в точности распознать то, что было произнесено. Поэтому значительную роль в распознавании речи играют так называемые языковые модели. Они позволяют определить наиболее вероятные словные последовательности. Сложность построения языковой модели во многом зависит от конкретного языка. Так, для английского языка достаточно использовать статистические модели (так называемые N-граммы). Для высокофлективных языков (языков, в которых существует много форм одного и того же слова), к которым относится и русский, языковые модели, построенные только с использованием статистики, уже не дают такого эффекта – слишком много нужно данных, чтобы достоверно оценить статистические связи между словами. Задача осложняется тем, что в русском языке допустим произвольный порядок слов («мама мыла раму» - «раму мыла мама»). Поэтому в Центре речевых технологий используются гибридные языковые модели, использующие правила русского языка, информацию о части речи и форме слова и классическую статистическую модель.

в) При распознавании на большом словаре также используется модуль определения темы разговора. Это позволяет в зависимости от тематики речи автоматически менять словарь и языковые модели. Модуль определения темы разговора разработан с использованием теории data mining. По сути этот компонент – зачатки системы искусственного интеллекта, которая в будущем все чаще будет использоваться совместно с модулем распознавания, делая процесс преобразования речи в текст более осмысленным.

Причины снижения качества автоматического распознавания речи. В настоящее время активно ведутся разработки в области распознавания речи. Системы и программы работают на высоком, качественном уровне, но достигнутые результаты не позволяют говорить о том, что машина может распознавать речь так же, как человек. Существует ряд причин, снижающих качество распознавания. В данной исследовательской работе рассмотрены такие причины, как физические (неречевые) помехи, речевые сбои и акцентная речь (речевые помехи). Одной из целей современных разработчиков систем распознавания речи является создание программы, которая бы давала высокие показатели в условиях физических помех, а также при распознавании акцентной речи и спонтанной речи с речевыми сбоями.

В процессе проведения исследования было выдвинуто следующее предположение: если современные технологии не имеют трудностей с распознаванием отдельных команд и даже слитных заранее подготовленных текстов, а физические помехи при этом устраняются с помощью технологий шумоочистки, то присутствие в записи речевых сбоев (оговорок, хезитаций, самокоррекций), свойственных спонтанной речи, или акцентной речи, может являться причиной большого количества сбоев. Анализ теоретического материала в сочетании с собственными наблюдениями позволили сформулировать задачи практического исследования в данной работе. Для выработки рекомендаций по совершенствованию программ распознавания

речи был проведён анализ работы приложений S Voice от компании Samsung и Dragon Dictation от компании Nuance. Новизна исследования состоит в использовании акцентной (не эталонной) речи для тестирования указанных речевых систем, при этом некоторые записи содержат также признаки речевых сбоев и физические помехи.

Источником материала для проведения эксперимента послужил интернет-архив The Speech Accent Archive, в котором представлены образцы акцентной английской речи. Была произведена выборка образцов чтения текста на английском языке носителями китайского языка. В группе информантов присутствуют представители женского и мужского полов, разных возрастных категорий. Для проведения данного исследования были отобраны образцы речи 8-ми представителей Китая обоих полов и разных возрастных категорий: 2 девушки и 2 парней в возрасте 22-25 лет; 2 женщины и 2 мужчин в возрасте 42-49. Информанты подобраны с учетом диалектной вариативности в китайском языке: 1 группа (молодые информанты) – носители кантонского диалекта, 2 группа (взрослые информанты) – носители мандаринского диалекта. Оба приложения по распознаванию речи тестировались с использованием образцов речи каждого информанта по 3 раза. Материалом для озвучивания информантами послужил специально составленный разработчиками интернет-архива «The Speech Accent Archive» текст, в котором учтены звукокомплексы современного английского языка, сложные для произнесения носителями азиатских языков.

В ходе исследования было выявлено, что из 204 сбоев, допущенных приложением S Voice, 12 были вызваны речевыми сбоями, 154 – присутствием акцентной речи, что составляет 6% и 75,5 % соответственно от общего количества. При каждом прослушивании образца звучащего текста в реализации одного и того же информанта результаты тестирования системы различны. В случае если система не успевает обрабатывать получаемый сигнал, этот вид сбоя отражается в категории «Другие причины».

Установлено, что система не справляется с акустическими особенностями английской речи в реализации дикторов-китайцев. Особую трудность при распознавании представляет отсутствие в китайском английском межзубных звуков, замена глухих звуков звонкими и наоборот, добавление эпентетического гласного, редукция консонантных окончаний и опущение конечной согласной. Легкая хезитация не влияет на качество распознавания, но хезитация в совокупности с самокоррекцией вызывает серьезные проблемы. В ходе исследования было выявлено, что из 213 сбоев, допущенных приложением Dragon Dictation, 47 были вызваны речевыми сбоями, 115 – присутствием акцентной речи, что составляет 22% и 54% соответственно от общего количества.

В случае если система не успевает обрабатывать получаемый сигнал, этот вид сбоя отражается в категории «Другие причины». Определено также, что система не справляется с особенностями китайского акцента ванглийской речи. Особую трудность представляет отсутствие в китайском английском межзубных звуков, редукция консонантных окончаний и опущение конечной согласной, трудность произнесения последовательности согласных звуков (консонантных кластеров). Выводы относительно распознавания различных видов хезитации совпадают с описанными для системы SVoice. Следует отметить, что различия между результатами распознавания образов речи разных дикторов незначительны. При этом, в отдельных случаях система может демонстрировать кардинально отличные результаты распознавания образца звучащего текста в реализации одного и того же информанта: от полного распознавания, до распознавания с большим количеством сбоев. Выявлены единичные варианты 100% распознавания образцов акцентной речи. При этом при тестировании системы отдельными образцами речи фиксируется постоянный сбой системы: можно предположить, что система не успевает обрабатывать поступающий сигнал. Данное наблюдение позволяет выдвинуть предположение (которое может иметь значимость для дальнейшего исследования процессов распознавания речи) о том, что

качество автоматического распознавания нестабильно, и это может иметь своим следствием получение лишь спорадически успешных результатов. При суммировании количественных данных, полученных в ходе проведения практического тестирования работы приложений SVoice и DragonDictation, были получены следующие результаты: из общего количества сбоев, допущенных обоими приложениями (417 (204+213)), причиной некачественного распознавания являются речевые сбои 59 (12+47), причем акцентная речь является причиной 269 (154+115) сбоев.

Таким образом, в результате проведения практического исследования качества работы приложений SVoice и Dragon Dictation, наше предположение о том, что речевые сбои и акцентная речь представляют большую сложность для современных систем и программ автоматического распознавания речи, было подтверждено, доказательством чему служат полученные эмпирические данные и их количественный анализ. Можно сделать вывод, что в настоящее время не существует системы, эффективно справляющейся с распознаванием акцентной речи или речи с присутствующими в ней речевыми сбоями. Одним из направлений, где полученные результаты могут быть полезными, является создание встраиваемого модуля по идентификации акцента в системы распознавания речи, а также разработка на базе подобного модуля автоматизированных тренажеров по устранению акцента в речи на неродном языке.

Обеспечение высокого качества распознавания. Качество распознавания зависит от двух факторов – структуры каркаса системы распознавания речи (набора программных модулей и алгоритмов, использующихся при распознавании) и качества моделей – акустических, языковых, тематических.

Все модели обучаются с использованием большого объема материала. Так, для акустических моделей используются сотни часов записей речи тысяч дикторов. Для повышения устойчивости распознавания к помехам и искажениям, при обучении используются записи в различных каналах и

различных условиях. Для обучения языковых моделей и моделей тематик используются текстовые корпуса объемом от сотен миллионов словоформ до нескольких миллиардов. Подготовка такого объема обучающего материала – это сложная и кропотливая работа. Центр речевых технологий [39] в течение нескольких десятилетий накапливал обучающий материал и на данный момент обладает уникальным по своим объемам, разнообразию и качеству набором записей и текстов, способных обеспечить высочайшее качество распознавания речи.

Контрольные вопросы

1. Дайте определение нейронной сети.
2. Опишите формальную модель искусственного нейрона.
3. Расскажите об общем построении нейронных сетей.
4. Расскажите о применениях нейронных сетей.
5. Каковы преимущества применение нейронных сетей для распознавания речи?
6. Для чего применяется коэффициенты линейного предсказания?
7. Как вычисляется коэффициенты линейного предсказания?
8. Расскажите о применении скрытых Марковских моделей для распознавания речи?
9. Каковы особенности скрытых Марковских моделей для решения задачи распознавания речи?
10. Опишите алгоритма вычисления кепстральных коэффициентов.
11. Как и для чего производится сегментация речи на этапе анализа?
12. Объясните алгоритм динамической трансформации.
13. Каковы особенности алгоритма динамической трансформации по сравнению с другимми алгоритмами распознавания речи?
14. Опишите алгоритм определения слов с использованием алгоритма DWT.

15. Каким образом определяются границы фонем при анализе слитной речи.
16. Перечислите основные причины снижения качества автоматического распознавания речи.
17. Какова ошибка распознавания дискретной речи? Слитной речи?

ГЛАВА 4. ТЕХНОЛОГИИ АНАЛИЗА И СИНТЕЗА РЕЧИ

4.1. История систем анализа речи

При исследованиях в области речевых технологий необходимо понимать, что есть задача анализа речи и есть задача синтеза речи. Анализ речи предполагает создание компьютерных программ перевода речевого сигнала в параметрическое представление и распознавание звуков, слогов, слов и предложений с использованием алгоритмов искусственного интеллекта (спектральный анализ, формирование кепстров, мел-преобразование, нейронные сети, скрытые цепи Маркова, алгоритмы динамического программирования и линейного предсказания). При этом основным технологическим приемом распознавания является создание эталонных единиц записи речи или ее фрагментов и последующее распознавание соответствия входной записи сигнала одному из эталонов. Такие системы часто называются «Системы речь-текст», так как результат распознавания отображается в текстовом варианте.

В задачах синтеза речи (система «текст-речь») алгоритмы интеллектуального анализа служат для распознавания текстовых сообщений и поиску их соответствия хранимым в памяти образцам речевых фрагментов в виде звуков, слогов или слов. В результате обработки формируются речевых сообщения в виде естественной речи с некоторым искусственным звучанием.

Большинство систем распознавания речи (Automatic Speech Recognition - ASR) состоит из процесса анализа и обработки аналогового сигнала и процесса распознавания в цифровой форме. При анализе аналогового сигнала из речи выделяются свойства, которые используются далее в процессе распознавания для того, чтобы определить, что было сказано. Рассмотрим краткую историю развития систем ASR в контексте этих двух процессов.

Самые первые попытки создания ASR систем осуществлялись в 1950-х годах. Была построена зависимая от диктора система, распознававшая

цифры. В качестве свойств сигнала использовались спектральные резонансы гласных в словах. В 1959 году был создан модуль, способный распознавать десять гласных вне зависимости от диктора.

В 60-х годах в Японии было построено несколько машин, которые распознавали гласные звуки, используя специальный спектральный анализатор. Также было создано устройство, распознающее фонемы.

В 70-х годах в области распознавания речи было совершено два значительных открытия: использование метода динамического программирования DTW, основанное на временном выравнивании речевых диалектов, и метод кодирования линейного предсказания LPC, который успешно использовался в распознавании сигналов с низким битрейтом (количество битов информации, передаваемых в секунду). В компаниях AT&T и Bell Laboratories были построены распознающие системы, обработка акустического сигнала в которых была основана на LPC анализе, а процесс распознавания на DTW.

В 80-х годах от подходов, основанных на применении шаблонов, исследования в области распознавания речи перешли к методам статистического моделирования. Использовались скрытые модели Маркова (Hidden Markov Models - HMM). В конце 80-х годов к проблеме распознавания был применен метод, основанный на искусственных нейронных сетях (Artificial Neural Network - ANN). В наши дни большинство ASR систем в процессе распознавания используют HMM.

С 90-х годов распознавание речи несколько усовершенствовалось. Словарь распознаваемых слов вырос до нескольких десятков тысяч. Использование быстрых алгоритмов декодирования позволило производить распознавание в реальном времени. В современных дикторозависимых системах, распознающих отдельные слова, количество которых достигает двадцати тысяч слов, ошибки составляют менее 1%. И около 5% ошибок в независимых от диктора системах, которые распознают слитную речь из тысячи слов.

Распознавание речи в реальном времени с помощью современных методов требует больших вычислительных ресурсов, объем которых часто бывает ограничен. Невозможность широкого применения многих алгоритмов сегодня, например, в мобильных устройствах, заставляет исследователей искать более эффективные и оптимизированные методы.

Основные подходы к решению задачи анализа речи. Первый подход, который используется для улучшения показателей анализа речи, основывается на выделении векторов свойств из сигнала с учетом особенностей восприятия звука человеческим ухом. Он включает в себя анализ несущих частот и выравнивание сигнала по громкости. Наиболее распространенными технологиями, использующими такой подход, являются метод кепстральных коэффициентов тоновой частоты MFCC и метод коэффициентов линейного предсказания. Одновременное и опережающее сопоставление с шаблоном (маскирование), характерное для человеческого восприятия, может быть смоделировано и использовано для выделения свойств, обеспечивающих большую устойчивость от шумов. С этой целью был создан метод варьирования размерностей кадров (Variable Frame Rate analysis, VFR). Учитывая специфику работы нервных клеток, отвечающих за слуховые рецепторы, был предложен метод диапазонной автокорреляции (Subband-Autocorrelation, SBCOR).

Другой подход основан на анализе звуковых сигналов. Различие поступающих в систему зашумленных сигналов от шаблонов, полученных в ходе обучения «чистыми» сигналами, является основной причиной неустойчивости работы систем распознавания. Целью подхода является уменьшение этого различия. Предполагается, что шум в звуковых сигналах аддитивный и стационарный. Оценки среднего значения усредненного шума вычитаются из кепстра или спектра, вычисленного по зашумленным данным. Некоторые модификации таких методов включают в себя нелинейное спектральное вычитание, которые используют спектральные огибающие.

Такие техники требуют хорошей оценки шума, которую на практике бывает сложно получить, особенно в случае нестационарного фонового шума.

Еще одним способом борьбы с разницей между полученными свойствами из зашумленных и чистых сигналов является использование высокочастотного фильтра. Предполагается, что шум в сигнале не стационарный, а медленно изменяющийся во времени. Метод RASTA (Relative Spectral Analysis, Hermansky&Morgan, 1994) представлен таким образом, что относительные спектральные изменения фиксируются. И те медленные изменения, которые были вызваны шумом, удаляются. В этом случае отпадает необходимость в явном оценивании шума.

Третий подход основан на использовании многомерных пространств. Основной идеей этого подхода является нахождение линейного отображения, которое минимизирует функцию стоимости. Часто в качестве такого отображения берется умножение вектора свойств на матрицу преобразования. Примерами данного подхода могут служить основной компонентный анализ (Principal Component Analysis, PCA) и независимый компонентный анализ (Independent Component Analysis, ICA), а также проектирование на многомерные подпространства.

Современные системы распознавания речи. В настоящее время речевое распознавание находит все новые и новые области применения, начиная от приложений, осуществляющих преобразование речевой информации в текст и заканчивая бортовыми устройствами управления автомобилем. Все многообразие существующих систем распознавания речи можно условно разделить на следующие группы.

1. Программные ядра для аппаратных реализаций систем распознавания речи.

2. Наборы библиотек, утилит для разработки приложений, использующих речевое распознавание.

3. Независимые пользовательские приложения, осуществляющие речевое управление и/или преобразование речи в текст.

4. Специализированные приложения, использующие распознавание речи.

5. Устройства, выполняющие распознавание на аппаратном уровне.

6. Теоретические исследования и разработки.

Рассмотрим каждую из этих групп подробнее.

Технологии для аппаратных реализаций. В основе любой речевой технологии лежит так называемый «engine» или ядро программы – набор данных и правил, по которым осуществляется обработка данных. В зависимости от назначения этого ядра различают TTS (Text-to-Speech) и ASR. TTS предоставляет возможность синтеза речи по тексту, а ASRengine – для анализа речи.

Существует несколько крупных производителей, занимающихся созданием ASR ядер. Среди них такие компании, как SPIRIT, Advanced Recognition Technologies, IBM.

Корпорация IBM уже более 30 лет занимается вопросами автоматического распознавания речи и достигла в этой области больших успехов. Так компания ProVoxTechnologies на основе программного ядра ViaVoice® от IBM создала систему VoxReports для диктовки отчетов врачей-радиологов. По результатам тестирований данная система с точностью 95-98% распознает слитную речь нормального темпа (до 180 слов в минуту) в независимости от диктора. Однако словарь системы ограничен набором специфических медицинских терминов.

Наборы библиотек для разработки приложений. С развитием речевых технологий и все большим внедрением мобильных устройств, возникла идея применения речевого управления при построении сетевых приложений. Для этого было необходимо разработать унифицированный стандарт для интеграции речевых технологий.

Один из открытых стандартов на основе XML-языка – Voice XML (Voicee Xtensible Markup Language), первая версия опубликована в мае 2000 г. международным консорциумом World Wide Web (W3 Consortium) – предназначен для разработки интерактивных голосовых приложений

(Interactive Voice Response, IVR) для управления медиа ресурсами. Цель создания стандарта - привнесение всех преимуществ web-программирования в разработку IVR-приложений.

Однако интерес к много модальным приложениям, сочетающим распознавание речи с другими формами ввода информации (при помощи клавиатуры, пера или набора цифровых кнопок) побудил ряд компаний, в том числе Microsoft, поддержать проект SALT Forum (Speech Application Language Tags - теги языка речевых приложений). И теперь вокруг SALT и VoiceXML консорциума W3C формируются два разных лагеря. До сих пор компании не могут прийти к единому мнению о выборе главного стандарта, и сейчас оба направления развиваются в равной степени.

Некоторые компании занимаются разработкой пакетов для создания речевых приложений, так называемых Software Development Kit (SDK), поддерживающих тот или иной стандарт. Так компания Philips создала пакет SpeechSDK, который поддерживает спецификацию VoiceXML и выполнен для связи с C/C++ API.

Независимые пользовательские приложения. В настоящее время рынок программных распознавателей речи представлен множеством приложений. Необходимо отметить пакет Dragon Naturally Speaking Preferred фирмы Dragon Systems – это единственная программа, приблизившаяся к тому, чтобы соответствовать заявленным характеристикам. В целом она очень близко подходит к достижению заявленной безошибочности распознавания - 95%. Хотя пакет Dragon и уступает некоторым из конкурентов в том, что касается перемещения по экрану, правки и форматирования, он превосходит всех в главном - способности с первого раза правильно распознавать произнесенные слова.

Специализированные приложения. Распознавание речи может применяться не только для ввода текста или подачи команд, но и для более специфичных целей. Так российская компания «Центр Речевых Технологий» разрабатывает и производит программные продукты, технологии и образцы

техники для подразделений МВД, ФСБ, МЧС, служб экстренной помощи, центров обработки вызовов и для других пользователей, в деятельности которых особое значение придается регистрации и обработке речевой информации.

Компанией созданы следующие приложения: «ИКАР Лаб» – инструментальный комплекс криминалистического исследования фонограмм речи, «Трал» – автоматизированный комплекс распознавания дикторов в фонограммах телефонных переговоров, «Территория» – автоматизированная система диагностики диалектов и акцентов русской устной речи.

Устройства, выполняющие распознавание на аппаратном уровне. Для использования функций речевого распознавания в различных устройствах, роботах, игрушках, разрабатываются аппаратные методы. Так американская компания SensoryInc. разработала интегральную схему Voice Direct™ 364, осуществляющую дикторозависимое распознавание небольшого числа команд (около 60) после предварительного обучения. Перед началом эксплуатации модуль необходимо обучить всем командам, используемым в работе. Команды сохраняются во внешнюю память в виде образов размером 128 байт. Во время работы, образ очередной команды сравнивается с эталонными из памяти в нейросетевом модуле и принимается решение о совпадении

Теоретические исследования и разработки. Разработкой теоретической базы в области речевых технологий занимается множество исследовательских групп по всему миру. В первую очередь это такие крупные корпорации как IBM, Intel, Microsoft, AT&T. Эти компании занимаются теорией распознавания уже не один десяток лет и являются законодателями в этой области.

В Узбекистане также ведутся исследования в области речевого распознавания. Например, в лаборатории речевые технологии ТУИТ занимаются этой проблемой уже более 5 лет. Главным научным и практическим направлением деятельности лаборатории в настоящее время

является улучшение качества распознавания речевых сигналов, разработка новых эффективных алгоритмов и их аппаратная реализация с возможностью использования узбекского языка.

4.2. Методы и программы синтеза речи

В то время как задача распознавания речи очень сложна и решена лишь отчасти, задача синтеза речи намного проще (хотя и там есть немало проблем, ждущих своего решения).

Технологии синтеза речи применяются в метро при объявлении остановок. Владельцы мобильных телефонов могут общаться с автоматической сервисной службой для определения остатка средств на счету, переключения тарифных планов, подключения или отключения услуг и пр. Сервисная служба общается голосом с применением технологий синтеза речи. Выпущено немало детских игрушек, «говорящих» человеческим голосом. В этих игрушках также применяются простейшие синтезаторы речи или цифровые магнитофоны.

Синтезаторы речи применяются в различных голосовых системах предупреждения, устанавливаемых в автомобилях и самолетах. Такие системы позволяют привлечь внимание человека к возникновению той или иной критической ситуации, не отвлекая его от процесса управления автомобилем, самолетом или другим аналогичным средством.

Также разработано немало компьютерных программ, способных читать голосом содержимое текстовых файлов или текст, расположенный в окнах приложений. Эти системы могут оказаться полезными тем, у кого ослаблено или полностью отсутствует зрение.

Модели синтеза речи.

Все существующие в настоящее время методы синтеза человеческой речи основаны на использовании двух моделей -модели компилятивного синтеза и формантно-голосовой модели.

Рассмотрим вкратце особенности этих моделей.

Модель компилятивного синтеза. Модель компилятивного синтеза предполагает синтез речи путем конкатенации (составления) записанных образцов отдельных звуков, произнесенных диктором.

При использовании этой модели составляется база данных звуковых фрагментов, из которых в дальнейшем будет синтезироваться речь.

На первый взгляд этот подход не должен вызывать особых затруднений.

Действительно, пользуясь микрофоном и звуковым редактором, например, редактором GoldWave, можно создать набор файлов различных звуковых фрагментов, а затем сохранить их содержимое в базе данных.

Создавая звуковые WAV-файлы с текстовыми сообщениями, можно озвучить операционную систему Microsoft Windows и многие ее приложения, такие как почтовые программы, инструментальные средства разработки и пр.

Модель компилятивного синтеза подходит, главным образом, только в простейших случаях, когда синтезатор должен произносить относительно небольшой и заранее известный набор фраз. При этом обеспечивается довольно высокое качество речи. Впрочем, этот факт не слишком удивителен, если вспомнить, что для синтеза используется естественная человеческая речь.

Тем не менее, на стыке составляемых звуковых фрагментов возможны интонационные искажения и разрывы, заметные на слух. Кроме того, создание крупной базы данных звуковых фрагментов, учитывающей все особенности произношения фонем и аллофонов с разными интонациями, представляет собой сложную и кропотливую работу.

Формантно-голосовая модель. Формантно-голосовая модель основана на моделировании речевого тракта человека.

Эта модель может быть реализована с применением нейронных сетей и допускает самообучение. К сожалению, ввиду сложности точного моделирования особенностей речевого тракта, а также учета интонационной модуляции речи формантно-голосовая модель обладает относительно низкой точностью синтезируемых звуков речи. Тем не менее, современные программы синтеза речи, построенные с использованием этой модели, синтезируют вполне разборчивую речь и могут применяться в ряде случаев.

Заметим, что системы голосового предупреждения о возникновении аварийных ситуаций лучше строить с использованием модели компилятивного синтеза, так как разборчивость речи в таких системах выходит на передний план.

Что же касается «бытовых» синтезаторов речи, то в них можно с успехом применять и форманто-голосовую модель. Схематически эта модель показана на рис.4.1.

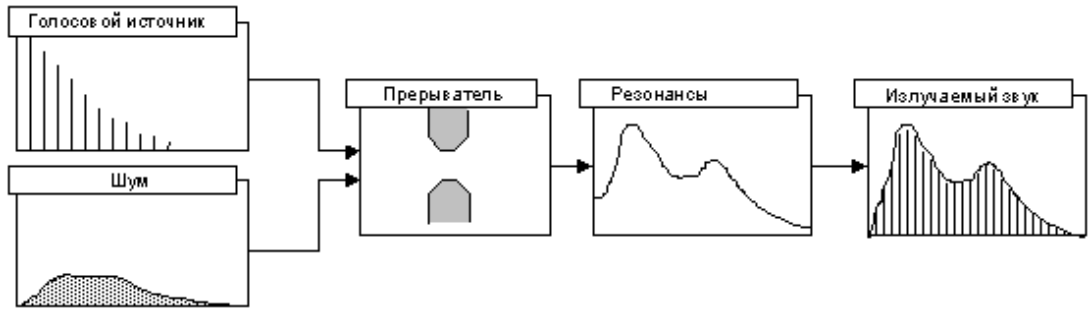


Рис.4.1. Формантно-голосовая модель синтеза речи

При построении модели использовались данные об артикуляционном аппарате человека, а также данные фонетики и лингвистики. Как видите, в качестве исходного сигнала применяется комбинация голосового источника и генератора шума. Прерыватель и резонансное устройство моделирует работу речевого тракта. В результате этого моделирования образуется излучаемый звук речи.

При этом для достижения компромисса между качеством модели и ее сложностью были выбраны следующие основные параметры исследуемой системы:

- частота основного тона;
- частота шума;
- количество формант;
- центральная частота каждой форманты;
- вклад каждой форманты.

Частота основного тона определяет высоту голоса. Этот параметр не должен вызывать у Вас никаких вопросов. Что же касается частоты шума, то здесь нужно сделать пояснение.

Образование шума представляет собой достаточно сложный процесс, зависящий от многих факторов, таких как давление и скорость воздушной струи, геометрической формы воздушного тракта, акустических свойств материала и пр. Чтобы полностью смоделировать шум речи на физическом уровне, необходимо создать точную модель речевого аппарата человека, что представляет собой очень сложную задачу.

В качестве альтернативы можно использовать белый шум, спектр которого распределен по некоторому закону (например, по Гауссу) относительно некоторой центральной частоты. При этом закон распределения подбирается экспериментально, а частотой шума в этом случае является упомянутая выше центральная частота.

Количество активных формант, участвующих в образовании речи, выбирается экспериментально, причем в качестве ориентировочного значения используется 4.

Так как форманта представляет собой резонанс в речевом тракте, у неё есть частота резонанса и огибающая. Вид огибающей также определяется экспериментально, в первом приближении это Гауссово распределение.

Вклад каждой форманты определяет, насколько сильно форманта воздействует на основной сигнал.

Все приведенные выше параметры, кроме количества формант, изменяются в процессе образования речи для получения различных звуков. Хотя для более качественного синтеза речи необходимо строить более детальную модель, приведенные параметры достаточны для того, чтобы синтезированные звуки были разборчивы.

Синтез речи с помощью нейронной сети. Для исследования формантно-голосовой модели синтеза речи был создан инструмент «Модель синтеза», в котором ручным заданием параметров можно синтезировать практически любой гласный или шипящий звук. Также приводятся уже готовые образцы некоторых звуков (в форме параметров модели).

Алгоритм синтеза речи.

Процесс синтеза речи выглядит следующим образом.

Уровни выходов нейронов эффекторного слоя нейросети при помощи карты эффекторов (рис.4.2) преобразуются в значения выбранных параметров модели синтеза. Карта эффекторов определяет соответствие между каждым нейроном эффекторного слоя и конкретным параметром модели синтеза, а также предельные значения каждого параметра. Число эффекторов и число параметров модели может не совпадать. Если параметру не соответствует ни один эффектор, используется некоторое фиксированное значение (значение по умолчанию).

Далее по текущему состоянию модели синтезируется сигнал в пространстве частот: генерируется линейка частот, представляющих голосовой источник. На эту линейку частот накладывается формантная структура (резонансы). Для синтеза шума используется генератор случайной амплитуды и фазы.

На последнем этапе выполняется обратное преобразование Фурье для получения звуков речи.

При обучении системы формировались нейронные ансамбли для каждого звука из обучающей последовательности **а, б, в, г, д**. Затем

проводилось обучение синтезу. В результате в эффекторном слое установились правильные связи с символьным слоем.

Система успешно обучилась синтезу — синтезируемые звуки в точности соответствуют тонам из обучающей выборки. На рис. 4.3 показана обучающая выборка, а на рис. 4.4 — результат синтеза.

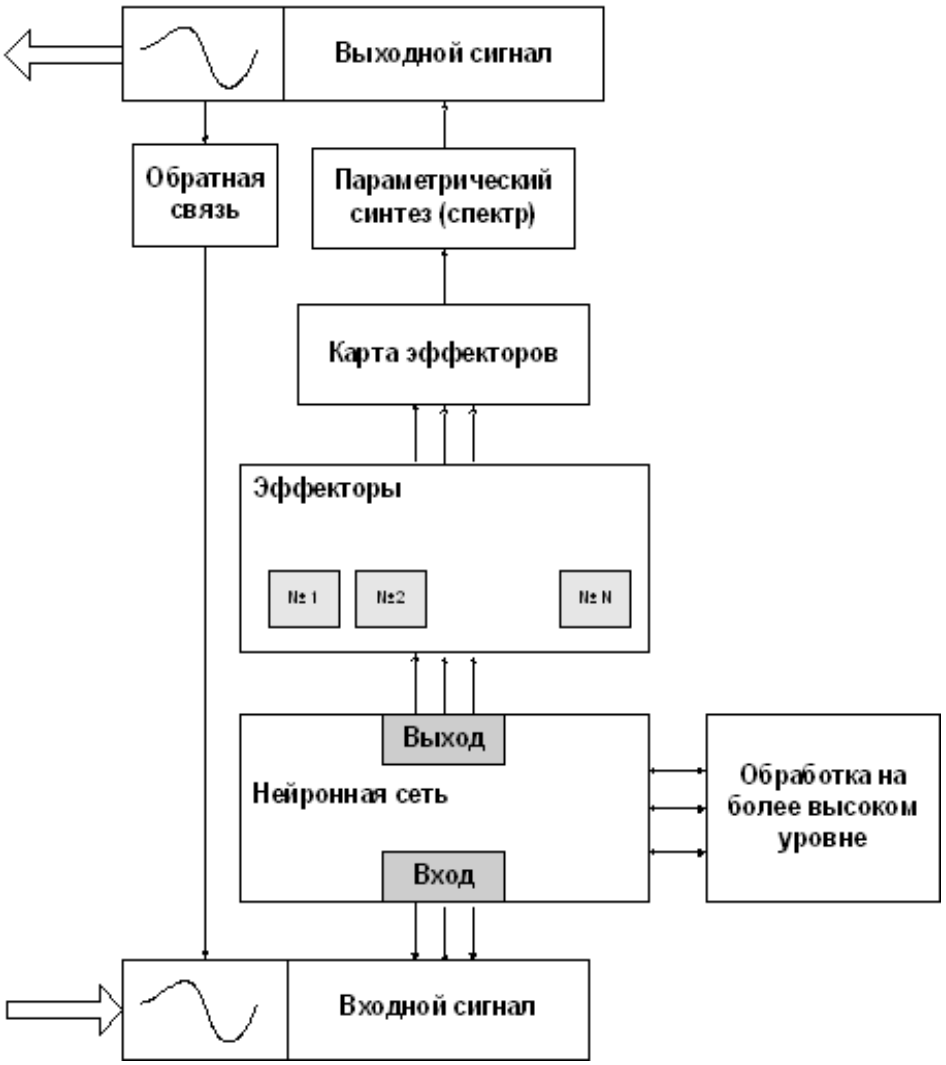


Рис. 4.2. Блок-схема уровня ввода/вывода

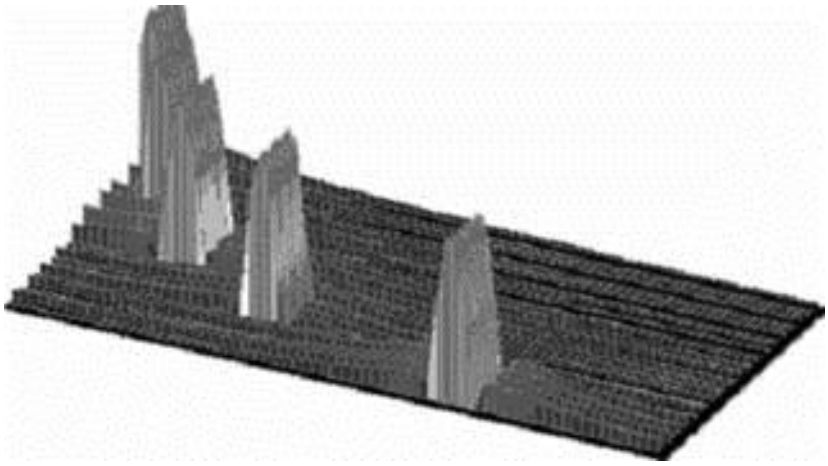


Рис. 4.3. Обучающая выборка

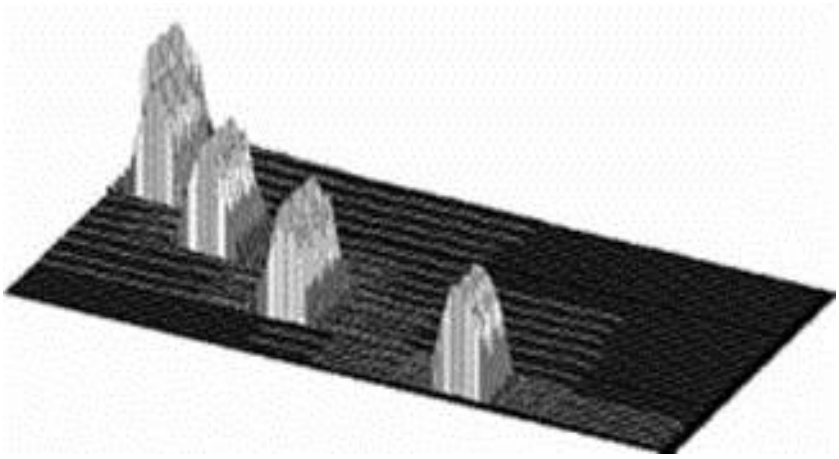


Рис. 4.4. Результат синтеза

На рис. 4.5 показан результат обучения синтезу звуков а,и,о,у. Спектрограммы синтезированных звуков близки к спектрограммам оригинальных звуков, хотя видны и отличия.

Ограничения использованного алгоритма. В этом алгоритме узким местом является размер окна дискретного преобразования Фурье ДПФ. В данной модели синтезируются статичные звуки, при этом не происходит изменение параметров в процессе синтеза.

В реальной же речи параметры звука меняются при переходе от одного звука к другому, причем меняются непрерывно. Очевидно, при использовании окон ДПФ такой результат получить невозможно в пределах окна параметры звука меняться не будут. Теоретически благодаря полной

обратимости дискретного преобразования Фурье возможно получить спектр для любого сигнала, в том числе и с динамически меняющимися параметрами.

Поэтому для генерации звука с изменяющимися параметрами нужно сокращать размер окна ДПФ или брать не весь сгенерированный кадр, а только его часть (не забывая при этом синхронизировать фазу сигнала). В идеале размер кадра можно свести к одному отсчету дискретизации по времени. Этот способ генерации речи дает лучшие результаты по сравнению с ДПФ, но работает гораздо медленнее ДПФ. В системе SAS можно выбрать используемый способ генерации.

Программные реализации синтезаторов речи.

Большинство таких синтезаторов разработано для платформы Microsoft Windows и пользуется речевым программным интерфейсом Speech API, разработанным компанией Microsoft.

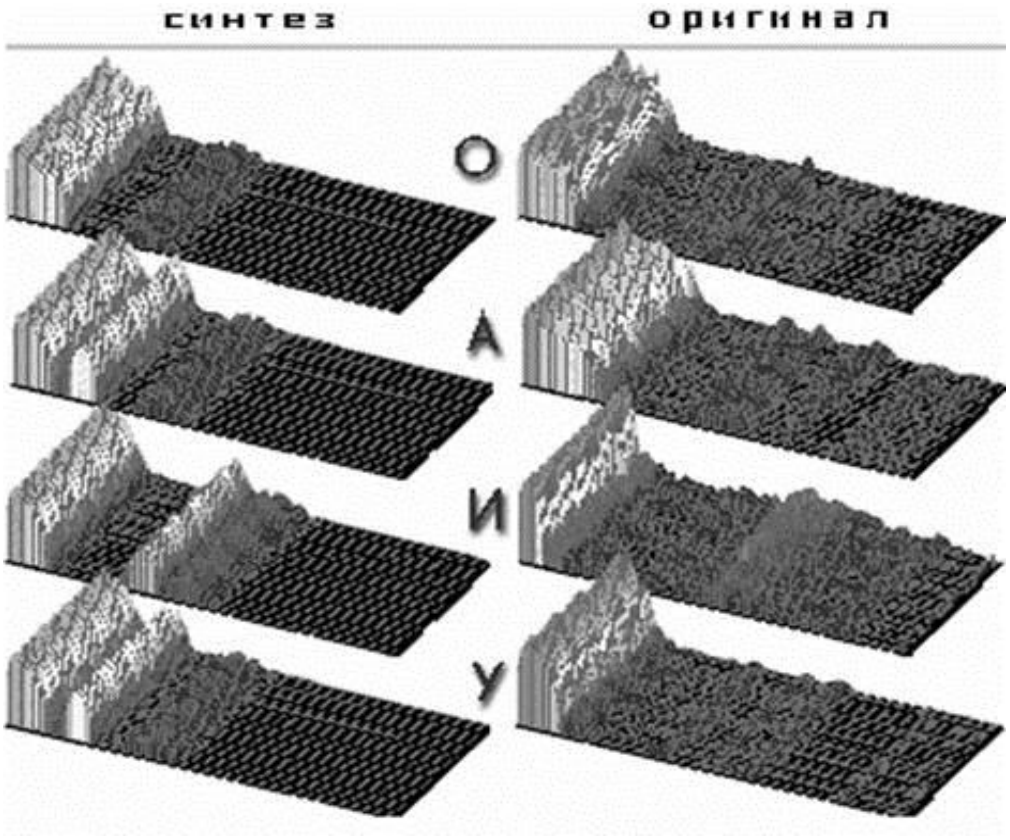


Рис. 4.5. Результат обучения синтезу

В комплекте с операционной системой Microsoft Windows не поставляются средства распознавания или синтеза речи. Однако разработчики могут создавать такие средства, используя при этом упомянутый выше программный интерфейс Speech API.

Что же касается пользователей, то для того чтобы снабдить компьютер речевым интерфейсом, необходимо установить на него речевые программные модули (speech engine). Как Вы знаете, в составе пакета офисных программ Microsoft Office XP поставляются такие модули, но не для русского языка.

Синтезатор речи Govorilka. В зависимости от установленных речевых модулей, программа Govorilka (рис.4.6) может читать текст разными голосами и на разных языках, в том числе и на русском языке.

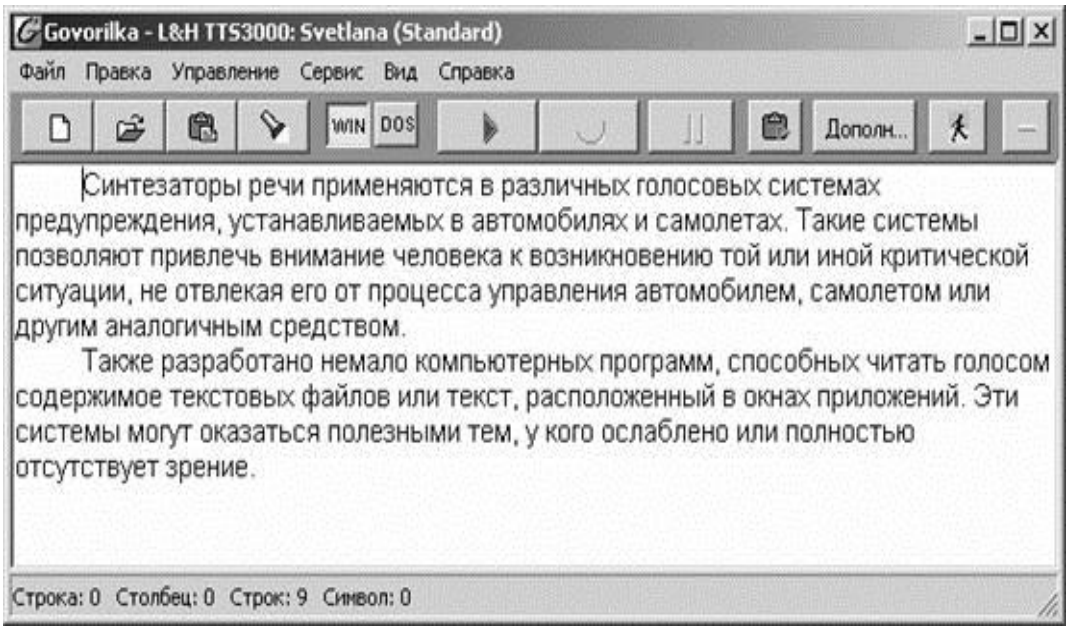


Рис. 4.6. Синтезатор речи Govorilka

Исходный текст для чтения может быть загружен из текстового файла, файла RTF и универсальный буфер обмена операционной системы Clipboard. Для загрузки текста можно также перетащить значок текстового файла на значок программы Govorilka или в окно этой программы.

Программа способна читать файлы с символами кириллицы в кодировке операционной системы Microsoft Windows 7/8/10.

Размер читаемого текста практически не ограничен. Загруженный однажды текст, а также текущая позиция при чтении запоминается программой. Таким образом, если текст большой, Вы можете слушать его по частям, даже выключая на время компьютер.

Для запуска текста, загруженного в окно программы, достаточно щелкнуть кнопку «**Читать текст»** (с изображением треугольника зеленого цвета) или нажать клавишу F5. прочитанный текст выделяется синим цветом.

С помощью кнопок «**Стоп»** (клавиша F6) и «**Пауза/Продолжить»** (клавиша F8) можно остановить, временно приостановить или продолжит чтение текста.

Можно читать как весь загруженный текст, так и любой его фрагмент. Для чтения фрагмента текста этот фрагмент нужно выделить мышью или при помощи клавиатуры, а затем щелкнуть кнопку **Читать текст.**

Программа позволяет сохранить результаты синтеза речи, записав синтезированную речь в файл формата WAV или MP3. Заметим, что запись речи в файл происходит не в реальном времени, а ускоренно. В самом деле, для выполнения операции записи речи в файл не требуется проговаривать текст, что необходимо делать со скоростью, привычной для человека.

Для коррекции произношения отдельных слов в программе Govorilka предусмотрен легко пополняемый словарь произношений.

Программа Better Text to MP3. Программа Better Text to MP3, разработанная компанией United Research (www.research-lab.com) интересна тем, что она может автоматически преобразовывать текстовые файлы в звуковые файлы популярных форматов WAV и MP3. Впоследствии такие файлы можно прослушивать при помощи любого проигрывателя звуковых файлов, такого, например, как Winamp.

Следует заметить, что программа Better Text to MP3 и сама может служить проигрывателем звуковых файлов.

Работа программы основана на использовании речевого интерфейса Microsoft SAPI 4.0, при этом программа может говорить (а точнее говоря, записывать в файл текст, произнесенный синтезированным голосом) на 11 языках. Предусмотрена также регулировка темпа речи.

Если на компьютере установлен программный компонент преобразования текста в речь для русского языка, программа Better Text to MP3 сможет читать и русские тексты.

Так как программа способна преобразовывать файлы текстовых документов в звуковые файлы, с ее помощью Вы можете преобразовать библиотеку текстовых документов в набор звуковых файлов для прослушивания. Эти файлы можно прослушать не только на компьютере, но и на обычном портативном плеере MP3-файлов (например, по дороге на работу).

Программа Better Text to MP3 может также переслать файл MP3, полученный в результате преобразования, по электронной почте.

Программа снабжена клавиатурным интерфейсом, предназначенным для людей с ограниченными возможностями. Такой интерфейс доступен в зарегистрированной версии программы по дополнительному запросу.

4.3. Голосовой интерфейс в технологии распознавания речи

Создание интерфейсов, поддерживающих и предлагающих более эргономичные и естественные формы диалога между пользователями и компьютерной техникой, движется и ускоряется внедрением информационных технологий в след растущим потребностям профессиональной и повседневной деятельности человека. В области информационных технологий средства взаимодействия пользователя с технической системой принято называть интерфейсом. Интерфейсы бывают

разные и реализуются разными средствами и методами. Одной из важнейших задач разработки современных технических систем является обеспечение наиболее интуитивного и естественного интерфейса с пользователем, то есть современные компьютерные приложения ориентированны на пользователя.

Одной из естественных форм взаимодействия для человека является речь. Голосовой интерфейс может улучшить существующий пользовательский интерфейс - он обеспечивает более удобный и менее ограниченный способ взаимодействия человека с компьютером. Качественный голосовой интерфейс помогает преодолевать неприятие технологии пользователями, так как для его использования не нужно овладевать новыми навыками. Голосовой интерфейс качественным образом изменяет способ, а следовательно и эффективность взаимодействия пользователя с системой. Голосовой поиск от компании Google и голосовой ассистент Siri от компании Apple являются этому яркими примерами, подтверждая насущную необходимость внедрения речевых технологий, в частности распознавания речи и голосовых интерфейсов.

Важный и практический аспект данных проблем связаны с тем, что голосовой интерфейс является необходимой компонентой, когда речь идет о создании комфортных условий жизни для людей с нарушениями опорно-двигательного аппарата, а также специалистам утратившим возможность использовать стандартные средства в результате профессионального заболевания, травмы или увечья. Такие системы со временем войдут в повседневный быт в процессе реализации концепции так называемых «умных домов».

В связи с вышесказанным становится актуальной проблема возможности создания голосовых интерфейсов для специалистов и систем предоставляющих такие средства, учитывая невозможность мгновенного перехода и необходимость адаптации к новым инструментам и средствам, очевидной становится потребность в интеграции с уже существующими системами. На практике решение подобной задачи и создание необходимой

интегрируемой системы оказывается нетривиальным.

Далеко не все задачи разработки голосового интерфейса в настоящее время можно считать решенными. Проблема разработки голосового интерфейса является достаточно сложной и комплексной, что требует от разработчика знаний в различных предметных областях, таких как компьютерные науки, лингвистика и психология поведения человека. Даже при наличии продвинутых средств проектирования, разработка эффективного голосового пользовательского интерфейса требует от его создателей детального понимания как задач, выполняемых системой, так и психологии пользователей системы.

Проведенный библиографический поиск и анализ информации в Internet подтвердил актуальность данной темы тем, что в настоящее время многие ведущие компании усиливают работу в направлении развития голосовых интерфейсов и технологии распознавания речи.

Распознавание речи - технология, позволяющая использовать естественный для человека речевой интерфейс для взаимодействия с электронной техникой. Сложность распознавание речи состоит в том, что совокупность таких характеристик голоса и речи как тембр, громкость, высота, темп, интонация, качество дикции делают речь каждого человека по-своему неповторимой и уникальной как отпечатки пальцев. Задачей компьютерной техники и программного обеспечения в состоит в том, чтобы распознать сказанные человеком слова в любых, не беря экстремальные, условиях без какой-либо предварительной адаптации под конкретный голос.

Интерфейсы - основа реализации взаимодействия всех современных информационных систем. Попытки научить компьютеры общаться с людьми при помощи естественного голосового интерфейса предпринимались с первых лет истории компьютерной техники.

Интерфейс – способ взаимодействия компьютерной системы (программы) с пользователями и устройствами.

На основе интерфейса реализуется взаимодействие всех современных

информационных систем. Под интерфейсом понимается набор средств, правил и методов, за счет которых осуществляется коммуникация между элементами системы, различными программами и устройствами. Под совокупностью средств, методов и правил подразумевают: средства вывода информации из устройства (системы) пользователю — весь доступный спектр воздействий на организм человека (зрительных, слуховых, тактильных, обонятельных и других.), средства ввода информации/команд пользователем реализуются сейчас множеством всевозможных устройств. Методы как набор правил, заложенных разработчиком устройства, согласно которым совокупность действий пользователя должна привести к необходимой реакции устройства и выполнению требуемой задачи, и правила эти должны быть достаточно ясны для понимания и легки для запоминания.

По наличию тех или иных средств ввода, интерфейсы разделяются на типы:

- голосовой,
- жестовый,
- возможны смешанные варианты.

Пользовательский интерфейс (англ. user interface,) - разновидность интерфейсов взаимодействия управляемых человеком систем. Термин применяется по отношению к компьютерным программам (приложениям).

Как любая система общения с устройствами, которые способны к интерактивному взаимодействию с пользователем, существуют: графический интерфейс пользователя (программные функции реализуются графическими элементами экрана), диалоговый интерфейс (поисковая строка), интерфейс программирования приложений, сетевой интерфейс, интерфейс операционной системы (ОС).

Одним из самых важных показателей, характеризующих интерфейс пользователя, является usability – логичность и простота элементов управления, удобство программы или системы в пользовании с целью быть необходимыми и достаточными, удобными и практичными, расположенными

и скомпонованными разумно и понятно, и соответствовать психофизиологии человека.

Увеличение в устройстве (при равной функциональности) средств ввода-вывода дает упрощение построения методов управления и упрощение правил пользования, но зато приводит к сложности восприятия информации пользователем — интерфейс становится перегруженным.

И наоборот — уменьшение средств отображения и контроля приводит к усложнению правил управления, так как каждый элемент несет на себе слишком много функций.

В связи с увеличением интенсивности обмена информацией в системе «человек-машина» особое значение имеет снижение нагрузки на тактильно-зрительные каналы человека. Например, в системах управления востребованной является идея голосового контроля и управления состоянием системы (речевое общение для контроля состояния работы самолета, бес кнопочный телефон, речевое управление производственными процессами).

Внедрение голосового интерфейса оставит глаза и руки оператора (пилота, водителя, рабочего за станком) свободными от перегрузки, что повысит надежность и качество управления.

Использование речевого диалога в системах массового обслуживания населения также актуально. Помимо исключительного удобства для населения, такие системы повышают коммерческую выгоду как за счет привлечения дополнительной клиентуры, так и путем замены человека-оператора компьютерными системами с голосовым интерфейсом.

Голосовые интерфейсы, компоненты, виды и задачи системы распознавания речи.

Преимущества голосового интерфейса:

- оперативность и естественность;
- минимум специальной подготовки пользователя;
- возможность управления объектом в темноте, за пределами его визуальной видимости (в частности, с использованием существующей

телефонной сети);

- возможность использования одновременно ручного (с помощью клавиатуры) и голосового ввода информации;
- обеспечение мобильности оператора при управлении.

К основным классам задач голосового интерфейса следует отнести:

- синтез речи – эта задача включает в себя комплекс подзадач и заключается в обеспечении возможности произнесение речи компьютером на основе произвольного орфографического текста;
- анализ и распознавание речи – комплекс задач, включающих запись, оцифровку и анализ речи для распознавания полученного речевого сообщения компьютерной системой;
- понимание (интерпретация) речи – это комплекс задач, связанных с анализом смысла речевых сообщений и формированием реакции (ответа) компьютерной системы;
- распознавание голоса – комплекс задач, включающих анализ особенностей голоса говорящего с целью выявления каких-либо его индивидуальных (личностных) особенностей и качеств;
- компьютерное клонирование голоса и дикции – это создание близкой копии, но не биологической, а компьютерной, и не всего существа в целом (в данном случае человека), а только одной из его интеллектуальных функций: чтение произвольного орфографического текста.

Общая структура голосового интерфейса включает два основных компонента:

- синтез речи;
- распознавание речи.

Каждая из задач голосового интерфейса является достаточно сложной, то в соответствие указанным компонентам ставятся два отдельных класса систем:

- системы синтеза речи;
- системы распознавания речи.

Реализация речевого диалога происходит посредством диалога, при

котором запрос и ответ со стороны пользователя ведется на языке, близком к естественному. Пользователь свободно формулирует задачу, но с набором установленных программной средой слов, фраз и синтаксиса языка Разновидностью интерактивного естественного диалога является речевое общение с компьютерной системой. В этом случае человеческий голос может преобразовываться, например, в текст, или использоваться для интерактивного управления системой, или для идентификации личности. В основе данных процессов лежит технология и решение задачи распознавания речи.

Речь в физическом смысле - это акустический сигнал, генерируемый артикуляционными органами человека, передающийся через физическую среду, воспринимаемый ухом человека. При естественной или искусственной генерации речи в акустическом сигнале изменяются физические параметры. Эти изменения воздействуют на мембрану уха, создают траектории звуковых образов, понимаемых человеком как соответствующие звуки данного языка, или иначе говоря, при произнесении слов человек генерирует звуки (фонемы), которые несут информацию о тех символах, с помощью которых эти слова могут быть записаны в виде текста.

Математическую модель генерации звука можно представить в виде возбуждающих генераторов тонового и белого шума, группы резонаторов, модуляторов и ключей (рот, нос, язык, губы), обеспечивающих формирование ощущения определенного звука.

Системы распознавания речи - это системы, анализирующие акустический сигнал алгоритмами, основанными на разнообразных теориях, предполагающих, какие характеристики речевого сигнала создают ощущения звуков данного языка, и математических методах, с той или иной точностью выделяющих значащие параметры акустического сигнала и преобразующие его в различной полноте в необходимую форму.

Заблаговременно формируется база фонем языка, содержащая шаблоны базового набора слов при «усредненной» речи, то есть независящей от

диктора. Речь переводится в фонемное описание и поступает в файл описания фонем, откуда это описание поступает в блок распознавания, проводящий сравнение поступившей информации с той, которая хранится в базе. Формируются распознанные слова, которые преобразуются в текстовые данные или команду.

Системы распознавания речи состоят из двух частей - акустической и лингвистической. В общем случае могут включат в себя фонетическую, фонологическую, морфологическую, лексическую, синтаксическую и семантическую модели языка.

Акустическая - отвечает за представление речевого сигнала, за его преобразование в некоторую форму, в которой в более явном виде присутствует информация в содержании речевого сообщения.

Лингвистическая - интерпретирует информацию, получаемую от акустической модели, и отвечает за представление результата распознавания потребителю.

Задачи распознавания речи - автоматическое восстановление текста произносимых человеком слов, фраз или предложений на естественном языке и проблемы идентификации, шумоочистки, распознавания языков, оценки психофизического состояния человека. При решении задачи распознавания слитной речи человек применяет свои знания о естественном языке, а также смысл произносимого для устранения неоднозначности при восстановлении текста предложения.

Поэтому задачу распознавания речи дополнительно разделяют на две независимые задачи:

- задачу локального распознавания речи;
- задачу восстановления текста слитной речи по множеству возможных гипотез распознавания.

Рассмотрим разработанные системы распознавания речи и голосовых интерфейсов.

Программа обеспечение Dragon Dictate – первая коммерческая

программа для обычных пользователей (1990 год).

VAL(voice-activated link) от BellSouth – первый голосовой портал, система с целью обработки справочных и поисковых запросов для покупателей в крупных торговых центрах и абонентов телефоных компаний по заданным запросам, услугам, торговым маркам (1996 год).

Улучшенная версия программы от компании Dragon Systems. Dragon Naturally Speaking была способна распознавать нормальную речь, около 100 слов в минуту (1997 год).

Microsoft выпускает свою систему распознавания речи - Windows Speesh Recognition. При многих недостатках данная программа стала массовой (2001 год).

Google запускает, в тестовом режиме, Voice Search - сервис голосового поиска в сети интернет, но из-за необходимости звонить на специальный номер данная разработка была сразу свернута. Но компания Google продолжила разработки в этом направлении (2002 год).

Первая операционная система с функцией распознания речи Mac OS X Tiger, но это был не полноценный продукт, а тестовая версия. Voice Over была способна не только на распознание речи, но и являлась её синтезатором, программа могла читать текстовые документы, почтовые и веб-страницы, являсь при этом дикторонезависимой, и даже обслуживала нескольких пользователей одновременно (2005 год).

Microsoft выпускает операционную систему с полноценной поддержкой функции распознания речи Windows Vista (2006 год).

Приложение Voice Search от Google для iPhone (2009 год). Работа данного приложения опирается на облачные вычисления, позволившие провести крупномасштабный анализ данных поиска совпадений между огромным числом голосовых запросов пользователей и их словами, такая процедура способствовала быстрому росту и совершенствованию системы. Позднее появилась версия для операционной системы Android.

Google внедрена функция распознавания голоса в браузер Chrome. В

базах на серверах компании насчитывается около 230 миллиардов слов на многих языках мира.

Конец 2011 год начало продаж Apple iPhone 4S с программой Siri, которая не просто распознает речь,а выступает в качестве персонального виртуального ассистента, способного обрабатывать естественную речь, отвечать на заданные вопросы и предоставлять рекомендации, с поддержкой английского, французского и немецкого языков.

Ford of Europe и компания Nuance Communications, представляют SYNC (2012 год), которая на начальном этапе будет поддерживать британский вариант английского языка, французский, испанский, португальский, немецкий, итальянский, турецкий, голландский и русский языки. Пользователи системы смогут давать такие инструкции как «Позвонить (имя контакта)» или «Проигрывать исполнителя (имя исполнителя)». Языковые возможности системы обеспечивают работу функции помощи в экстренных ситуациях (Emergency Assistance), завоевавшей премию «Global Mobile Award 2012». Функция помогает находящимся в автомобиле людям в случае аварии оповестить операторов местных экстренных служб на соответствующем языке. Система SYNC установлена уже на более чем 4 миллионах автомобилей в США.

Используемые в распознавании речи методы и классификация систем распознавания речи. Практически все известные методы распознавания речи обладают рядом основных общих свойств:

- для распознавания используется метод сравнения с эталонами;
- сигнал может быть представлен либо в виде непрерывной функции времени, либо в виде слова в некотором конечном алфавите;
- для сокращения объема вычислений используются методы динамического программирования. Динамическое программирование (ДП) - метод решения задач путем составления последовательности из подзадач таким образом, что:
- первый элемент последовательности (возможно несколько элементов)

имеет тривиальное решение

- последний элемент этой последовательности - это исходная задача
- каждая задача этой последовательности может быть решена с использованием решения подзадач с меньшими номерами.

Методы распознавания речи можно разделить на две большие группы: непараметрические — с использованием непараметрических мер близости к эталонам (к ним можно отнести методы на основе формальных грамматик и методы на основе метрик на множестве речевых сигналов) — и параметрические (вероятностные — на основе метода скрытых моделей Маркова, нейросетевые).

Непараметрические методы, **о**снованы на мерах близости на множестве речевых сигналов. Метод Винцюка, основанный на методе динамического программирования (Беллман), развитый Итакурой и другими, позволил сократить время вычисления значений функции близости к эталонным сигналам с экспоненциального (от длины сигнала) до квадратичного. В силу того, что основной спецификой метода являлось нелинейное искажение временной оси одной из сравниваемых функций, метод получил название «динамической деформации времени». К достоинствам относятся простота его реализации и обучения. К недостаткам можно отнести сложность вычисления меры близости, которая пропорциональна квадрату длины сигнала, и большой объем памяти, необходимый для хранения эталонов команд - пропорциональный длине сигнала и количеству команд в словаре.

Параметрические - методы, применяемые к задаче распознавания речи в настоящее время, были впервые предложены рядом американских исследователей (Бейкер и Джелинек) в 1970-е годы прошлого века. В них применяется теория скрытых моделей Маркова - дважды стохастические процессы и цепи Маркова по переходам между состояниями и множества стационарных процессов в каждом состоянии цепи.

Достоинствами метода скрытых моделей Маркова являются:

– быстрый способ вычисления значений функции расстояния (вероятности);
– существенно меньший объем памяти, по сравнению с методом «динамической деформации времени», необходимый для хранения эталонов команд.

Основными недостатками:

– большая сложность его реализации;
– необходимость использования больших фонетически сбалансированных речевых корпусов для обучения параметров.

Основные характеристики и признаки, по которым можно классифицировать современные системы распознавания речи.

– словари размером в десятки и сотни тысяч слов;
– распознавание раздельной или слитной речи;
– работа в реальном времени;
– дикторозависимость или дикторонезависимость системы;
– надежность работы 95–98% для грамматически правильных текстов;
– назначение.

Классификация систем распознавания речи по сложности:

– системы автоматического распознавания изолированных слов для распознавания произносимых человеком команд по словам;
– системы автоматического распознавания слитной речи — с возможностью выделять слова в естественном частично слитном потоке человеческой речи;
– системы понимания речи - с элементами интеллекта, что позволяет, во-первых, на основе смыслового анализа более правильно выделять слова в потоке речи, а, во-вторых, сохранять информацию в некой базе знаний, откуда она может быть легко извлечена для решения определенных интеллектуальных задач.

Основные компоненты систем распознавания речи:

- графическая среда для разработки, компиляции и оптимизации грамматических и лексических блоков распознавания, проверки и редактирования лексиконов;
- система для протоколирования диалогов из работающего приложения с целью оценки качества распознавания и подстройки системы;
- инструмент оценки качества работы системы (проверка соответствия слова, сказанного абонентом, используемой грамматике);
- система для создания «тренируемых» языковых моделей, повышающих производительность и ускоряющих процесс распознавания;
- система для распределения множества параллельных запросов различных типов и прозрачной интеграции различных речевых модулей в сети.

Приведем более полную (чем на рис.3.1) классификацию систем распознавания речи (рис. 4.7).

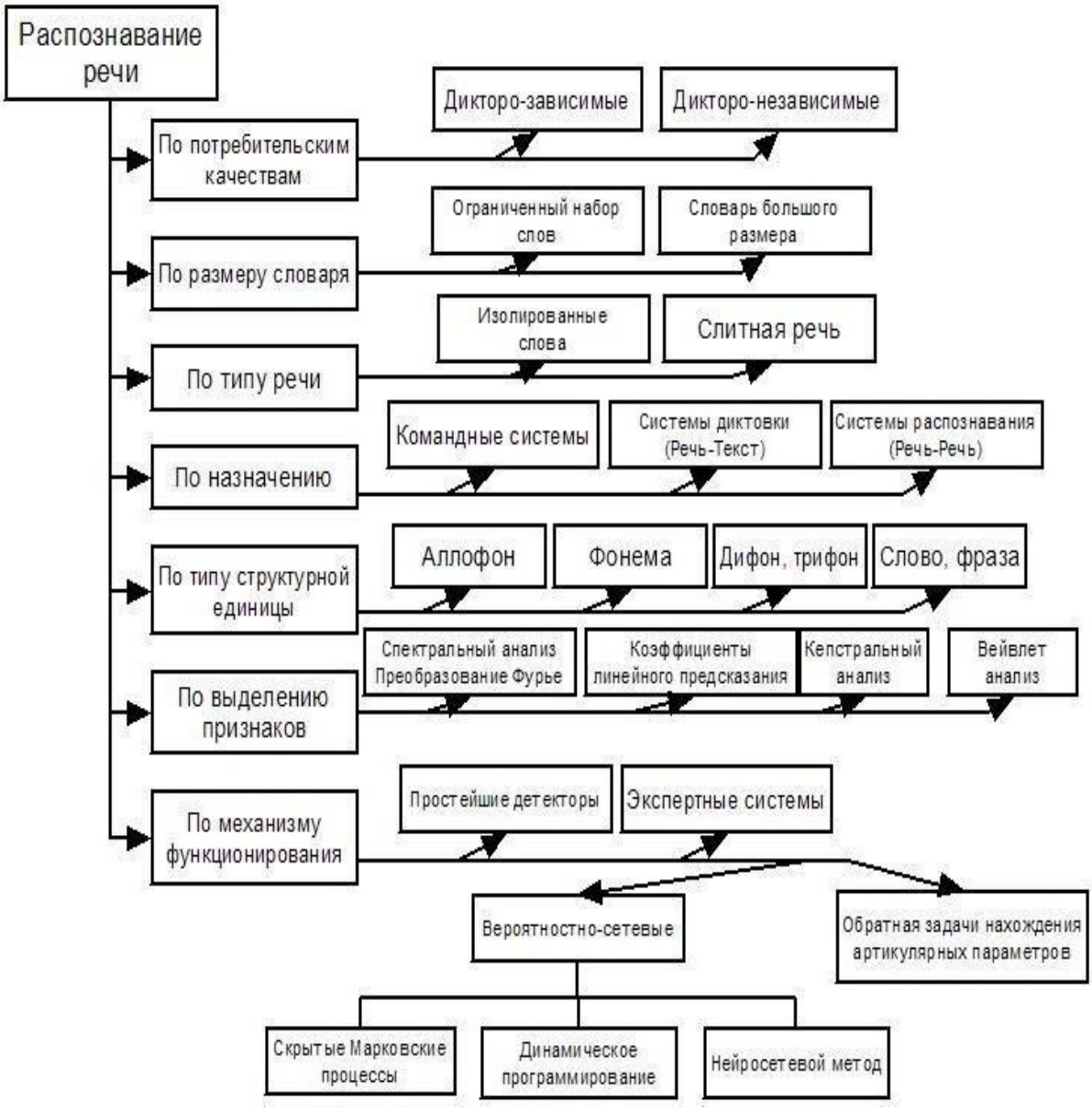


Рис.4.7. Классификация систем распознавания речи

Условия возникновения проблем систем распознавания речи:

- произвольный, наивный пользователь;
- спонтанная речь;
- наличие акустических помех и искажений, в том числе меняющихся;
- наличие речевых помех;
- недостаточная капитальная база, не дающая возможность интенсивно проводить исследования и разрабатывать новые инновационные алгоритмы в речевых технологиях.

Условия, на основе которых выявляются следующие требования и

ограничения:

- требуется предварительная настройка системы на голос от нескольких десятков минут до нескольких часов предварительного наговаривания текстов;
- некоторые проверки не дают результатов лучших, чем 5% ошибок;
- вероятность правильного распознавания слов не превышает одной трети даже для хорошо организованных спонтанно произнесенных текстов.

Далее рассмотрим современные отечественные и зарубежные продукты с использование технологий распознавания речи.

Обзор продуктов использующих технологии распознавания речи и голосового интерфейса, потребителей и перспективы.

Горыныч ПРОФ 3.0 — первая русскоязычная система автоматического распознавания речи для диктовки и голосового управления компьютером с поддержкой русского языка.

Характеристики:

- дикторозависимость;
- языкозависимость (русский язык и английский язык);
- точность распознавания зависит от ядра системы американской программы "Dragon Dictate";
- предоставляет средства голосового управления отдельными функциями операционной системы, текстовых редакторов и прикладных программ;
- требует обучения.

VoiceNavigator (компания «Центр речевых технологий») - высокотехнологичное решение для контакт-центров, предназначенное для построения Систем Голосового Самообслуживания. VoiceNavigator позволяет автоматически обрабатывать вызовы с помощью технологий синтеза и распознавания речи. VoiceNavigator Web - навигация по веб-ресурсам при помощи голоса. Разработка позволяет управлять навигацией сайта при

помощи речевых команд.

Характеристики:

- дикторонезависимость;
- устойчивость к окружающим шумам и помехам в телефонном канале;
- распознавание русской речи работает с надежностью 97% (словарь 100 слов).

Speereo Speech Recognition (Российская компания «Speereo Software»). Программный продукт для разработки приложений в индустрии мобильных устройств и персональных компьютеров. Распознавание речи происходит непосредственно на устройстве, а не на сервере, что является ключевым преимуществом, по мнению разработчиков.

Характеристики:

- распознавание русской речи работает с надежностью около 95%;
- дикторонезависимость;
- словарный запас около 150 тыс. слов;
- одновременная поддержка нескольких языков;
- компактный размер движка.

Sakrament ASR Engine (разработка компании «Сакрамент») - технология распознавания речи используется при создании средств речевого управления – программ, управляющих действиями компьютера или другого электронного устройства с помощью голосовых команд, а также при организации телефонных справочных и информационных служб.

Характеристики:

- дикторонезависимость;
- языконезависимость;
- точность распознавания достигает 95-98%;
- распознавание речи в виде выражений и небольших предложений;
- нет возможности обучения.

Google Voice Search (компания «Google»). Ранее поиск применялся исключительно в мобильных устройствах. С недавнего времени голосовой поиск от Google встроен в браузер Google Chrome, что позволяет использовать этот сервис на различных платформах.

Характеристики:

- поддержка русского языка;
- возможность встраивать распознавание речи на веб-ресурсы;
- голосовые команды, словосочетания;
- для работы необходимо постоянное подключение к сети internet.

Dragon NaturallySpeaking (компания «Nuance») - мировой лидер в программном обеспечении по распознаванию человеческой речи. Возможность создавать новые документы, отправлять электронную почту, управлять популярными браузерами и разнообразными приложениями посредством голосовых команд.

Характеристики:

- отсутствует поддержка русского языка;
- точность распознавания до 99%.

ViaVoice (компания «IBM») представляет собой программный продукт для аппаратных реализаций. Компания ProVox Technologies на основе этого ядра создала систему для диктовки отчетов врачей-радиологов VoxReports.

Характеристики:

- точность распознавания достигает 95-98%;
- дикторонезависимость;
- словарь системы ограничен набором специфических терминов.

Sphinx – самый известное и наиболее работоспособное из открытых программных продуктов для распознавания речи на сегодняшний день. Разработка ведется в университете Карнеги-Меллона, распространяется на условиях лицензии Berkley Software Distribution (BSD) и доступен как для коммерческого, так и для некоммерческого использования.

Характеристики:

– дикторонезависимость;
– распознавание слитной речи;
– обучаемость;
– наличие версии для встраиваемых систем — Pocket Sphinx.

Наиболее значительные потребители голосовых технологий - электронная коммерция, производители всевозможных устройств домашнего применения, таких как телевизоры, видеомагнитофоны, микроволновые печи, стиральные машины и т.д. Рынок голосовой навигации в Web сайтах, осуществляющих электронную торговлю услугами по продаже авиа- и железнодорожных билетов, продуктов, другими услугами и сервисами, как по телефону, так и через интернет.

Речевые технологии, позволяющие распознавать команды в условиях шумов, позволяют дополнить управление в автомобилях таким функциями как управление светом, радио, замками. Голосовое управление функциями автомобильных аудио и навигационных систем уже реализовано в некоторых моделях BMW, Mercedes-Benz, Ford, Toyota. Такие системы помогают водителю не отвлекаться от дороги, однако для того, чтобы их эффективно использовать, водитель должен знать специальные голосовые команды, которых, к примеру, в системе Ford SYNC около десяти тысяч. Система SYNC установлена уже на более чем 4 миллионах автомобилей в США. До конца 2012 года система появится в Европе на автомобилях Focus, C-MAX, Transit и Fiesta.

Также существует потребность в речевых технологиях в военно-промышленном комплексе: тренажеры-имитаторы боевой техники; военная техника, системы оповещения (голосового оповещения оператора о неисправностях или повреждениях систем, а также о выполненных операциях/задачах), системы безопасности (например, возможность остановки боевой техники при ее повреждении либо ранении оператора при помощи голоса), комплексы ПВО, радиолокационные станции и др.

В образовательной сфере востребованы, в частности, системы обучения

языкам технология выделения и измерения фонем речи открывает новые возможности для обучения языкам. Она вводит в процесс обучения языку, кроме звуковой, визуальную обратную связь, позволяет увидеть свою и эталонную речь, сравнить их визуально, увидеть ошибки произношения и получить оценку произнесения фонемы, слова и фразы. Визуализация процесса произношения с выделением фонем и показом положения артикуляционных органов по анализу произношения, позволяет создать уникальные системы для обучения произношению для людей с ограниченными возможностями.

Российский «Центр речевых технологии» является ведущим разработчиком инновационных систем в сфере высококачественной записи, обработки и анализа аудио-видео информации, синтеза и распознавания речи. Создаваемые в «Центре речевых технологий» биометрические решения обеспечивают высокую точность распознавания личности по голосу и изображению лица в реальном времени. Эти решения находят успешное применение в государственном и коммерческом секторе, от небольших экспертных лабораторий до сложных систем безопасности национального масштаба.

Потребительское программное обеспечение распознавания речи все же в основной своей массе сконцентрировано на диктовке, позволяя печатать документы, электронные письма, помогая специальностям, где работа связанна с длительными периодами печати, с повышенным риском получения заболевания и травм опорно-двигательной и нервной системы. Ряд профессий таких как журналисты, программисты, писатели и ученые, в условиях сжатых сроков продолжают работу несмотря на рекомендации и останавливаются только когда утомлены и в особенности при боли или дискомфорте.

Что в последствии проявляется в тендините, сильных шейных и спинных болях и приводят к потере самооценки, снижению качества жизни, ухудшению семейных отношений, такие осложнения здоровья, вызываемые профессиональными заболеваниям, можно было бы снизить при

возможности замены длительных периодов печати, на альтернативный способ ввода, например с помощью средств голосового интерфейса.

Специализированные голосовые интерфейсы, голосовой интерфейс в разработке программного обеспечения.

Использование обычных программ распознавания речи для разработки программного обеспечения затруднено тем, что хотя и ключевые слова могут быть распознаны, но большая часть кодовых текстов состоит из имен переменных, названий процедур, которые представлены словосочетаниями и/или аббревиату рами. Также в языках программирования использование специальных/технических символов (отношения, сравнения, пунктуации) синтаксически отличается от естественных языков.

Основные затруднения при использовании не специализированных средств голосового интерфейса для разработки программного обеспечения:

- определение имен переменных, классов и функций;
- написание конструкций;
- написание конструкций;
- навигация;
- использование меню.

Определение имен переменных, классов и функций - если бы все имена переменных и классов были одиночным английским словом, тогда это не было бы проблематичным. В реальности это крайне редко, имена должны отражать/описывать функцию или содержимое переменной. Компиляторы требуют чтобы имена переменных не содержали пробелов, так что программистам приходится находить различные методы концентрации/сопряжения слов, сохраняя простоту чтения. Самая тривиальная из проблем заключается в том, что существующее программное обеспечение распознавания речи спроектировано для задач диктовки текстов и автоматически добавляет пробел между словами. Например, продиктовать «CamelCase» или «under_score», используя Dragon Naturally Speaking потребуется произнести:

«came, no space, capitalise next, case» или «under, no space, underscore, no space, score».

Имена переменных обычно длиннее, чем два слова, и чем длиннее имя, тем более значительно замедляется рабочий процесс, за счет увеличения времени диктовки.

Написание конструкций - все конструкции циклов и «if» утверждение обычно определяются наличием логического условия в скобках, при котором секция кода исполняется. Именно скобками затрудняется диктовка и использование этих конструкций становится неудобным.

Пример кода:

```
for(counter=0;counter<10;counter++)
{
printf(counter);
|}
```

Потребует произнести: «For, no space, open brackets counter equals−sign zero. semi−colon. counter less−than ten semi−colon. counter plus−sign plus−sign .close brackets. new line. open braces. print. no space. open brackets counter. close brackets. semi−colon. new line. cose braces».

Навигация - проблема распространяющееся на все, что связано с использованием средств голосовых интерфейсов. Обычно «мышь» позволяет перемещать курсор, «прокручивать» документ вверх и вниз в поисках необходимой секции, осуществляя быструю навигацию. Типовыми решениями проблемы навигации программного обеспечения распознавания речи являются переходы в начало или конец предложения, или абзаца, или по номеру строки, или на определенное количество строк, но в условиях разработки программного обеспечения необходимы более специализированные решения, также это недоступно тем кто не может пользоваться своими руками или ограничен условиями окружения.

Использование меню - множество задач требует использования различных меню (открытие, закрытие, сохранение файлов), что легко

реализуется «мышью», но трудно при помощи только голоса. Существует традиционная реализация индикации своих намерений воспользоваться опциями меню при помощи голосовой команды, на что в ответ программное обеспечение переходит в режим «управления». Задача разработки программного обеспечения потребует реализовать дополнительные режимы, диктуемые из ее потребностей, но основным ограничением является то, что обычно интерфейсы графически ориентированы и используют флаг элементы, списки и другие диалоговые элементы - затрудненное использование списков и диалоговых элементов возможно, но использование флаг элементов на первый взгляд кажется невозможным.

В результате требуется как минимум адаптированная или специализированная, а может и в целом иная по типу система предоставления средств голосового интерфейса в программировании.

Система предоставления средств голосового интерфейса позволит людям с ограниченными возможностями открыть себе двери в огромную индустрию, в противном случае недоступную для них, а также программистам, которые в связи с травмами или заболеваниями ограниченны в использовании стандартных интерфейсов, чтобы продолжать свою работу.

В мире также ведутся разработки программного обеспечения и инструментов предоставления средств голосового интерфейса для разработки программного обеспечения с различной степенью успеха. Приведем наиболее заметные из них.

VoiceCode — программное обеспечение инициатива Института Информационной Технологии Национального Исследовательского Консульства Канады. Задачей является разработка инструментария совместимых компонентов поддерживающего текущие лучшие практики голосовых интерфейсов для разработки программного обеспечения.

ShortTalk and EmacsListen — разработка специализированного разговорного языка для человека-компьютерного взаимодействия.

Voice Grip – дополнительный макрос для редактора Emacs созданный с

целью упрощения использования коммерческого программного обеспечения распознавания речи программистами программистам.

Java by voice — серия макросов для редактора Emacs спроектированные для упрощенного ввода кода на языке Java.

Cache Pad – макрос для редактора Emacs для кэширования недавно продиктованных имен функций и переменных пере использования.

Emacs VR Mode - макрос для редактора Emacs добавляющий функционал «Select and Say» в редактор из Dragon Naturally Speaking.

На основании обзора предметной области, наибольшие результаты достигнуты в распознавании отдельных звуков, слов и фраз, а также в создании программного обеспечения для голосового управления операционными системами, мультимедийным программным обеспечением и текстовыми редакторами.

Выявлено, что в настоящее время мало инструментов предоставляющих специализированные средства голосового интерфейса, в частности для разработки программного обеспечения, или это продукты текущих исследований, а не полноценное/коммерческое программное обеспечение.

Данная работа ставит перед собой две цели:

- первая, разработка модульной интегрируемой системы предоставления средств голосового интерфейса для разработки программного обеспечения и ее программной реализации, с совместимостью с любым пакетом программного обеспечения распознавания речи, для основных и мобильных платформ;
- второй целью является улучшение задачи программирования, а также функционала программного обеспечения, и разработка средств специализированного голосового интерфейса.

4.4. Инструментарий для разработки систем распознавания речи

Программирование как задачу можно разбить на четыре области:

- написание;
- отладка;
- компиляция;
- исполнение.

Программист обычно пишет небольшую секцию для решения конкретной проблемы. Затем эта часть кода компилируется, если компиляция успешна, запускается и тестируется с применением инструментов отладки. Когда код исполняется правильно, то код расширяется для выполнения большей задачи. Процесс повторяется до того момента когда код решает всю поставленную задачу. Таким образом за написанием следует исправление ошибок тестирование и отладка. Программист обычно работает в среде, которая предоставляет инструменты для выполнения всех четырех функций.
Написание кода в свою очередь обычно делится на следующие задачи (зависит от языка программирования):

- определение имени класса;
- определение используемых функций;
- определение переменных используемых в функциях;
- манипуляция переменными с использованием вызова других функций или специфических конструкций for и while циклов, if и switch конструкций.

Если одно и тоже задание дать группе программистов результат будет разный потому, что у программистов есть личный стиль.

Несколько вещей определяющих стиль программиста:

- способ записи и именования переменных, классов и функций;
- структура и группирование кода;
- положение скобок;
- отступы в коде;
- использование комментариев.

Далее произведем иерархический анализ показывающий различные шаги, которые пользователь выполняет на каждой стадии написания кода.

Основные стадии написания кода:

- открытые файл;
- редактирование файла;
- сохранение файла;
- компилирование файла;
- выход.

Каждая стадия задачи будет раскрыта далее. Анализ проводится для объектно-ориентированных языков, таких как Java или C++, но основные задачи и их субзадачи будут одинаковы для любого современного языка программирования. Все задачи ниже второго уровня содержат все себе процесс печати или диктовки в редакторе, а для задачах ниже уровнем редакторы предоставляют соответствующие инструменты.

Задачи выше второго уровня обобщенно больше представляют собой мыслительный процесс, чем реально исполняемые действия, где возможно, задачи ниже четвертого уровня необходимо автоматизировать.

Открытие файла:

- выбор Открыть файл опции;
- ввод имени открываемого файлы или щелчок курсором по имени файла в отображаемом списке.

Редактирование Файла:

Уровень 1:

- подключение необходимых библиотек;
- определение нового класса;
- определение новой функции/метода.

Уровень 2:

- ввод ‘#include’;
- ввод имени используемой библиотеки;

- ввод ключевого слова ‘class’;
- ввод имени класса;
- определение области класса;
- определение классовых переменных;
- определение конструктора класса;
- определение области функции;
- определение типа результата функции;
- определение входных параметров функции;
- определение тела функции;
- определение новой переменной;
- присвоение значение переменной;
- использование функции.

Уровень 3:

- определение наследуемого/расширяемого класса;
- ввод директивы наследования/расширения;
- ввод имени родительского/супер класса;
- ввод пар (тип/класс, имя) параметров;
- определение области переменной;
- определение типа переменной;
- определение имени переменной;
- определение параметров конструктора;
- определение тела конструктора;
- определение используемой функции/метода;
- определение параметров используемой функции/метода;
- определение используемых функцией переменных;
- определение переменной для присвоения;
- присвоение значений используемым функцией переменным;
- присвоение значения;
- использование ‘if ‘ оператора;

- использование ‘else’ оператора;
- использование ‘for’ оператора;
- использование ‘while’ оператора;
- использование ‘do.. while’ оператора.

Уровень 4:

- ввод ключевых слов ‘if ‘, ‘else’, ‘for’, ‘while’, ‘do.. while’;
- инициализация начальных значений переменных/итераторов;
- задание условия сравнения с переменной/константой или остановки цикла;
- постановка открывающих/закрывающих скобок;
- постановка символьных операторов языка;
- постановка точки с запятой.

Сохранение файла:

- выбор сохранить файл опции;
- ввод имени, с которым сохраняется файл или;
- выбор Сохранить опции (если файл был предварительно сохранен).

Закрытие файла:

- выбор Закрыть файл опции;
- если запрошено сохранение изменений выбор между опциями Да или Нет.

Закрытие приложения:

- выбор выйти из приложения опции;
- сохранение файлов;
- подтверждение выхода.

Ниже третьего уровня действия синтаксически ориентированы (зависят от языка программирования) на этом уровне диктовка синтаксиса наиболее затруднена, а при печати происходит наибольшее количество опечаток. Если бы синтаксис языков содержал бы меньше технических и вспомогательных символов и позволялись бы пробелы после символов пунктуации, тогда бы

задача предоставления средств голосового интерфейса была бы проще.

Условия разработки системы и ее программной реализации. Развитие вычислительной техники сопровождается созданием новых и совершенствованием существующих языков программирования. Появляются новые тренды в ИТ: облачные вычисления, мобильные технологии, мультипроцессорная разработка.

В настоящее время существует достаточно большое количество языков программирования. Далее в таблице 4.1 представлены наиболее востребованные языки программирования за последние 5, 15, и 25 лет по данным индекса TIOBEA.

Табл.4.1. Наиболее востребованные языки программирования

Язык программирования	Май 2015	Май 2012	Май 2008	Май 2000
C	1	2	1	1
Java	2	1	3	-
Objective-C	3	42	-	-
C++	4	3	2	5
C#	5	8	-	-
PHP	6	4	-	-
(Visual)Basic	7	5	4	8
Python	8	7	28	-
Perl	9	6	6	-
Ruby	10	10	-	-
Lisp	13	16	19	2
Ada	23	17	10	3

Индекс TIOBEA — индекс, оценивающий популярность языков программирования, на основе подсчета результатов поисковых запросов, содержащих название языка. Для формирования индекса используется поиск в нескольких наиболее посещаемых порталах: Google, Blogger, Wikipedia,

YouTube, Baidu, Yahoo!, Bing, Amazon. Расчет индекса происходит ежемесячно и он может быть полезен при принятии стратегических решений.

В настоящее время, существует множество сред разработки программного обеспечения, рассмотрим наиболее востребованные.

Qt Creator - кроссплатформенная свободная интегрированная среда разработки - Integrated Development Enviroment (IDE) - для разработки программного обеспечения на языках C, C++ и QML. Разработанная Trolltech (Digia) для работы с фреймворком Qt, которая является частью Qt SDK. Включает в себя графический интерфейс отладчика, с возможностью отладки приложений на QML и отображения данных из контейнеров Qt, и визуальные средства разработки интерфейса как с использованием QtWidgets и/или QML. Поддерживает компиляторы: Gcc, Clang, MinGW,MSVC, Linux ICC, GCCE, RVCT, WINSCW.

Eclipse — свободная кроссплатформенная интегрированная среда разработки кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.

Eclipse служит в первую очередь платформой для разработки расширений, чем он и завоевал популярность, любой разработчик может расширить Eclipse своими модулями. Уже существуют Java Development Tools (JDT), C/C++ Development Tools (CDT),а также средства для языков Ada, COBOL, FORTRAN, PHP и пр. от различных разработчиков. Множество расширений дополняет среду Eclipse менеджерами для работы с базами данных, серверами приложений и др.

Eclipse разработана на языке Java, потому является платформо-независимым продуктом, за исключением библиотеки Standard Widget Toolkit (SWT), которая разрабатывается для всех распространенных платформ. Библиотека SWT используется вместо стандартной для Java, библиотеки Swing. Она полностью опирается на нижележащую платформу.

Microsoft Visual Studio — включает в себя интегрированную среду разработки программного обеспечения, а также ряд других

инструментальных средств. Данные продукт позволяет разрабатывать программное обеспечение, а также веб-сайты, веб-приложения, веб-службы для всех поддерживаемых платформ Microsoft Windows, Windows Mobile, Microsoft Silverrlight и др.

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как например, Subversion и Visual Source Safe), добавление новых наборов инструментов.

Xcode — программа для разработки приложений под OS X и iOS, разработанная компанией Apple и распространяется бесплатно через Apple App Store.

Основным приложением пакета является встроенная среда разработки, которая называется Xcode. Помимо этого, пакет Xcode включает в себя большую часть документации разработчика от Apple и Interface Builder — приложение, использующееся для создания графических интерфейсов.

Пакет Xcode поддерживает языки программирования C, C++, Objective-C, Objective-C++, Java, AppleScript, Python и Ruby с различными моделями программирования, включая (но не ограничиваясь) Cocoa, Carbon и Java, и включает в себя необходимые для этого инструменты,а также использует GNU Debugger (GDB) в качестве back-end’а для своего отладчика.

NetBeans — свободная интегрированная среда разработки приложений на языках программирования Java, JavaFX, Python, PHP, JavaScript, C, C++, Ада и ряда других.

NetBeans поддерживает плагины, позволяя разработчикам расширять возможности среды. NetBeans доступна в виде готовых дистрибутивов для платформ Microsoft Windows, Linux, FreeBSD, Mac OS X, OpenSolaris и Solaris, для всех остальных платформ доступна возможность скомпилировать NetBeans самостоятельно.

Такое разнообразие языков программирования, сред разработки,

технологий распознавания речи, их целевых платформ и бурное развитие отрасли - говорят о необходимости уже на этапе разработки системы предоставления средств голосового интерфейса, учитывать ограничения, возможность применения и разработки на различных платформах (кроссплатформенность) и закладывать возможности к модификации, расширению и адаптации, для поддержания своей востребованности и конкурентоспособности.

Для решения задач программирования с учетом вышеописанных тенденций и условий развития отрасли информационных технологий, вытекающих из них ограничений, разработана следующая концептуальная архитектура модульной, интегрируемой системы предоставления средств голосового интерфейса.

Описание схемы представленной на рис. 4.8:

Центральная система — управляет модулями и их взаимодействием, предоставляет пользовательский интерфейс, а также средства управления и конфигурации модулей;

- модули плагинов/расширений:
  - модули интеграции с системами распознавания речи,
  - плагины/расширения обеспечивающие управление, конфигурацию;
  - модули поддержки голосового интерфейса языков программирования;
  - плагины/расширения предоставляющие преобразование результатов распознавания речи в соответствующие команды, директивы, конструкции языка;
  - модули интеграции сред разработки программного обеспечения
  - плагины/расширения трансляции, преобразования, исполнения команд системы средствами среды разработки.

Рассмотрим далее задачу разработки и выбора платформы плагинов, с целью определения возможных условий критериев и ограничений.

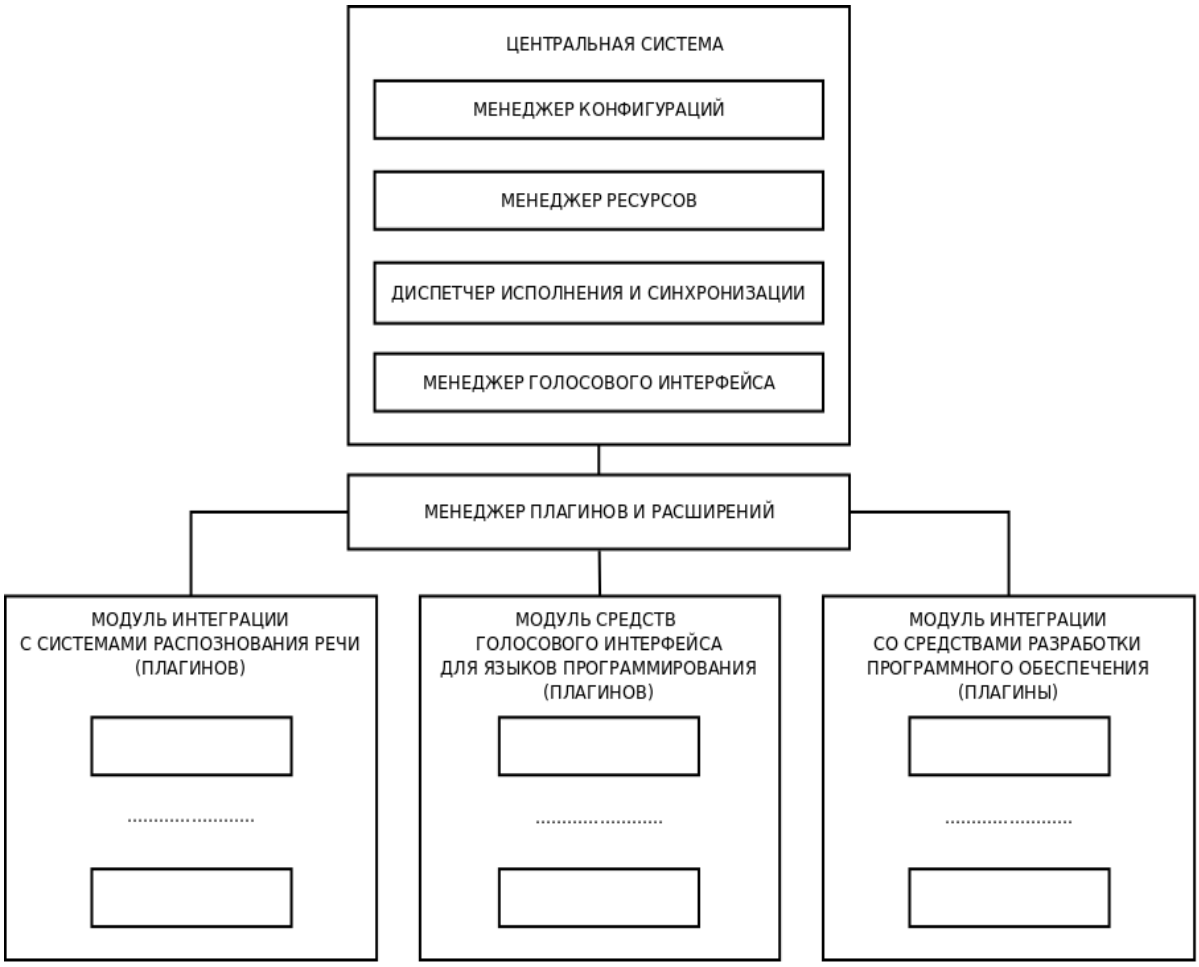


Рис. 4.8. Концептуальная схема архитектуры системы

Инструменты разработки. Из доступных на рынке и в отрасли инструментов разработки программного обеспечения, по критериям доступности, наличия и полноты документации, условиям лицензирования и стоимости, были выбраны следующие инструменты и приведены их описание и основные особенности.

Simplified Wrapper and Interface Generator (SWIG) — свободный инструмент для связывания (**англ.)** программ и библиотек написанных на C/C++ со скриптовыми языками, такими как Tcl, Perl, Python, Ruby, PHP или другими языками (Java, C#, Scheme или OcamlP). SWIG можно использовать, модифицировать и распространять практически без ограничений, для коммерческих и некоммерческих целей. Основная цель — достигнуть связи с минимальными усилиями. В файлы заголовка программы добавляется

небольшое количество указаний, по которым SWIG генерирует исходный код для связывания C/C++ и нужного языка.

В зависимости от языка, результат связывания может быть представлен в трех формах:

- исполняемый файл исходной программы со встроенным интерпретатором скриптового языка;
- разделяемая библиотека, к которой существующий интерпретатор может подключаться в виде расширения;
- разделяемая библиотека, которая может подключаться к другим программам, написанным на нужном языке (например, с помощью JNI для Java).

Qt Software Development Kit (SDK) включает в себя инструменты, необходимые для сборки десктопа, встроенных и мобильных приложений с Qt.

Можно выбрать для установки следующие цели и инструменты:

- средства разработки:
- инструмент цепи для разработки приложений для рабочего стола (доступно на Windows, Linux и Mac OS);
- Symbian инструмент цепи для разработки приложений для Symbian устройств (только для Windows);
- Qt Creator интегрированной среды разработки (IDE);
- Qt Simulator для тестирования мобильных функций API;
- Qt Assistant;
- Qt Designer;
- Qt Лингвист.
- документация;
- служба удаленного компиляции, которая обеспечивает простой, стандартизированной среду для создания приложений на Qt и создания инсталляционных пакетов для Symbian, Maemo, MeeGo

Harmattan и устройств.

Поддерживаемые платформы:

- рабочий стол;
- Qt Simulator;
- Maemo 5;
- MeeGo Harmattan;
- Symbian.

Компилятор GCC. GNU Compiler Collection (GCC, ранее GNU Compiler C Collection) - это свободно доступный из самых распространенных компиляторов на многих Unix-подобных системах, таких как GNU/Linux, члены семейства систем BSD, Mac OS X и др, и позволяют генерировать код для множества процессоров. GCC компилирует код программ в объектные модули для компоновки полученных модулей в единую исполняемую программу. Поддерживаются следующие языки программирования:

- C++,
- Objective-C,
- Objective-C++,
- Fortran,
- Java,
- Ada.

CMake — это кроссплатформенная система автоматизации сборки программного обеспечения из исходного кода. CMake не осуществляет сборку проекта, он лишь генерирует из своих CMakeLists.txt файлов Makefile для конкретной платформы, например может быть сгенерирован проект Microsoft Visual Studio, NMake makefile, проект XCode для MacOS, Unix makefile, Watcom makefile, проект Eclipse или CodeBlocks.

Cygwin представляет собой инструмент для портирования ПО UNIX в Windows, включает в себя инструменты разработки GNU для выполнения основных задач программирования, а также и некоторые прикладные программы, эквивалентные базовым программам UNIX, обеспечивает тесную

интеграцию Windows приложений, данных и ресурсов с приложениями, данными и ресурсами UNIX-подобной среды. Из среды Cygwin можно запускать обычные Windows приложения, также можно использовать инструменты Cygwin из Windows.

Cygwin состоит из двух частей: динамически или статически подключаемая библиотека, которая обеспечивает совместимость и реализует значительную часть стандарта POSIX и огромная коллекция приложений, которые обеспечивают привычную среду UNIX, включая Unix shell, и распространяется под GNU General Public License v2.

Android Native Development Kit (NDK) это набор инструментов, который позволяет встраивать компоненты, которые используют машинный код в Android приложении. Android приложения, запущенные в виртуальной машины Dalvik, позволяет реализовать части вашего приложения, используя язык программирования C/C++. Это может обеспечить преимущества для определенных классов приложений, в виде повторного использования существующего кода, а в некоторых случаях увеличение скорости.

Android NDK обеспечивает:

- набор инструментов и создавать файлы, используемые для создания собственных библиотек кода C и C++ источников;
- способ внедрения соответствующих родной библиотеки в файл приложения пакета, которые могут быть развернуты на устройствах Android;
- набор собственных заголовков системы и библиотеки, которые будут поддерживаться во все будущие версии платформы Android;
- документация, примеры и учебники.

Последний выпуск NDK поддерживает эти наборы инструкций микропроцессоров архитектуры Advanced RISC (Restricted (reduced instruction set computer) Machine (ARM):

- ARMv5TE (включая Thumb-1 инструкцию);
- ARMv7(в том числе Thumb-2 и VFPv3-D16 инструкции, с

дополнительной поддержкой для NEON/VFPv3-D32 инструкции);

- ARMv5TE машинный код будет работать на всех ARM-устройств на базе Android;
- NDK обеспечивает стабильные заголовки Libc (библиотеки C), в libm'ax (математическая библиотека), OpenGL (3D графическая библиотека), интерфейс JNI, и другие библиотеки.

Apache Another Neat Tool(Ant) — платформонезависимая утилита для автоматизации процесса сборки программного продукта, в отличие от другого сборщика проектов Apache Maven, обеспечивает императивную, а не декларативную сборку проекта.

Утилита Ant полностью независима от платформы, требуется лишь наличие установленной рабочей среды Java — Java Runtime Enviroment (JRE). Отказ от использования команд операционной системы и применение Extensive Markup Language (XML) обеспечивают переносимость сценариев.

Также для осуществления распознавания речи будет осуществлена интеграция с системой распознавания речи Sphinx, в частности с ее облегченной версией PocketSphinx.

Прикладная программа распознавания речи SpeechPearl.

Speech Pearl - это интегрированная среда разработки телефонных приложений с распознаванием речи. В состав этой среды входит набор инструментов, оптимизированных для создания, тестирования и настройки приложений распознавания речи.

Встроенный графический интерфейс предоставляет дружественный интерфейс для создания, настройки и тестирования грамматик и языковых ресурсов.

После того, как разработчик приложения создал диалоги и определил задачи распознавания, SpeechPearl предоставляет соответствующий инструмент для создания и оптимизации грамматик и языковых ресурсов.

Разработка крупных многорежимных систем с распознаванием речи требует интеллектуального управления распределенной архитектурой

речевых серверов. Это необходимо для надежности, масштабируемости и эффективности использования ресурсов.

SpeechPath - это контроллер ресурсов. Это программный модуль, выполненный в клиент-серверной TCP/IP архитектуре. Модуль может распределять множество параллельных запросов различных типов, что позволяет осуществлять прозрачную интеграцию различных речевых модулей в сети. Балансировка нагрузки оптимизирует использование всех имеющихся в сети ресурсов распознавания речи. Интеллектуальные механизмы восстановления дают возможность строить отказоустойчивые конфигурации.

Распознавание речи. SpeechKit JavaScript API предоставляет доступ к технологии распознавания речи. С помощью этой технологии можно реализовать заполнение форм голосом, диктовку писем или, например, голосовой поиск.

Источником аудиоданных в JavaScript API является микрофон. Для распознавания речи из другого источника (например, из файла) можно воспользоваться SpeechKit Cloud API.

Для запуска процесса распознавания API предоставляет три инструмента. В зависимости от сложности задачи можно выбрать тот или иной инструмент:

Инструмент класс Textline.

Добавляет на страницу элемент управления, состоящий из текстового поля и значка микрофона. При нажатии мышью на значок микрофона запускается процесс распознавания.

Класс Textline не предоставляет возможность запускать или останавливать процесс распознавания через клиентский код. Для этого предназначены два других инструмента:

1. Статическая функция recognize()

Процесс распознавания запускается сразу после вызова этой функции. С помощью recognize() распознавание можно начать в любой момент, например, сразу после загрузки страницы.

2. Класс SpeechRecognition

Класс SpeechRecognition позволяет управлять процессом распознавания на низком уровне. С его помощью можно не только начать распознавание в любой момент, но также и остановить его. Кроме того, класс SpeechRecognition предоставляет детальную информацию о ходе распознавания.

Класс Textline.

Класс Textline предназначен для распознавания коротких голосовых запросов. Он добавляет на страницу элемент управления «Поле для голосового ввода». Данный элемент управления состоит из текстового поля и привязанного к нему значка микрофона.

При нажатии на значок микрофона запускается процесс распознавания — браузер запрашивает у пользователя доступ к микрофону, и как только пользователь даст разрешение, начинается запись звука и его отправка на сервер. Распознанный текст отображается в поле ввода в режиме реального времени. При наступлении тишины длительностью в несколько секунд запись звука прекращается и распознавание завершается.

Для добавления элемента управления на страницу необходимо создать экземпляр класса Textline. Конструктору могут быть переданы следующие параметры:

- идентификатор контейнера, в котором будет размещен элемент управления (обязательный параметр);
- объект, содержащий настройки API.

Ниже приведен пример кода, добавляющего элемент управления на страницу:

Ниже приведен пример кода, добавляющего элемент управления на страницу:

```
<script type="text/javascript">
    window.onload = function () {
        // Задаем API-ключ (подробнее см. Глобальные настройки API).
        window.ya.speechkit.settings.apikey = 'acdf...';
        // Добавление элемента управления "Поле для голосового ввода".
        var textline = new ya.speechkit.Textline('my_id', {
                onInputFinished: function(text) {
                    // Финальный текст.
                    alert(text);
                }
            });
    }
</script>
...
<body>
    <div id="my_id" style="width: 200px"></div>
</body>
```

Функция recognize

Как и класс Textline (см. выше), функция recognize() предназначена для распознавания коротких голосовых запросов.

Отличие данной функции от Textline заключается в том, что recognize() позволяет запустить процесс распознавания из клиентского кода. Для запуска достаточно вызвать recognize() с нужными параметрами. Список доступных параметров, которые могут быть переданы данной функции, см в справочнике.

При наступлении тишины длительностью в несколько секунд запись звука прекращается и распознавание завершается. Финальный распознанный текст передается в callback-функцию, заданную в параметре doneCallback.

Класс SpeechRecognition

SpeechRecognition — это базовый класс для распознавания речи. Он позволяет управлять процессом распознавания на низком уровне, а также предоставляет более широкую функциональность по сравнению с Textline и recognize(). С помощью этого класса можно не только запустить процесс распознавания в нужный момент, но также приостановить или завершить его. Кроме того, класс SpeechRecognition предоставляет детальную информацию о ходе процесса распознавания.

Класс SpeechRecognition рекомендуется использовать для распознавания большого потока аудиоданных, например, для диктовки писем или записей. Также этот класс может использоваться для реализации голосового заполнения форм или голосового управления веб-приложением.

Перед запуском процесса распознавания необходимо создать экземпляр класса SpeechRecognition. Конструктор данного класса не требует никаких параметров.

var streamer = new ya.speechkit.SpeechRecognition();

Список доступных настроек приведен в справочнике.После создания объекта SpeechRecognition необходимо запрограммировать, в какой момент будет произведен запуск процесса распознавания. Таким моментом может быть окончание загрузки страницы или, например, нажатие какой-нибудь кнопки. Для запуска процесса распознавания предназначена функция start(). В качестве ее параметров передаются настройки распознавания.

```
streamer.start({
    // Вызывается после успешной инициализации сессии.
    initCallback: function () {
        console.log("Началась запись звука.");
    },
    // Данная функция вызывается многократно.
```

```
        // Ей передаются промежуточные результаты распознавания.
        // После остановки распознавания этой функции
        // будет передан финальный результат.
        dataCallback: function (text, done, merge, time) {
            console.log("Распознанный текст: " + text);
            console.log("Является ли результат финальным:" + done);
            console.log("Число обработанных запросов, по которым выдан ответ от сервера: " + merge);
            console.log("Время начала и конца распознанного фрагмента речи: " + time);
        },
        // Вызывается при возникновении ошибки (например, если передан неверный API-ключ).
        errorCallback: function (err) {
            console.log("Возникла ошибка: " + err);
        },
        // Содержит сведения о ходе процесса распознавания.
        infoCallback: function (sent_bytes, sent_packages, processed, format) {
            console.log("Отправлено данных на сервер: " + sent_bytes);
            console.log("Отправлено пакетов на сервер: " + sent_packages);
            console.log("Количество пакетов, которые обработал сервер: " + processed);
            console.log("До какой частоты понижена частота дискретизации звука: " + format);
        },
        // Будет вызвана после остановки распознавания.
        stopCallback: function () {
            console.log("Запись звука прекращена.");

        },
```

```
    // Возвращать ли промежуточные результаты.
    particialResults: true,
    // Длительность промежутка тишины (в сантисекундах),
    // при наступлении которой API начнет преобразование
    // промежуточных результатов в финальный текст.
    utteranceSilence: 60
});
```

Остановка процесса распознавания.

В отличие от recognize() и Textline, в которых распознавание завершается автоматически, при использовании класса SpeechRecognition разработчику необходимо самостоятельно остановить процесс распознавания. Для этого предназначена функция stop(). После ее вызова запись звука остановится, соединение с сервером закроется и сессия завершится.

```
// Остановка процесса распознавания.
// Функция вызывается без аргументов.
streamer.stop();
```

Примечание. После вызова функции stop() все настройки, которые были заданы при запуске процесса распознавания, сбрасываются.

Приостановка процесса распознавания.

При необходимости процесс распознавания можно приостановить. Для этого предназначена функция pause(). В результате ее вызова запись звука прекратится, но соединение с сервером не завершится.

```
// Приостановка процесса распознавания.
// Функция вызывается без аргументов.
streamer.pause();
```

Примечание. После вызова функции pause() все настройки, которые были заданы при запуске процесса распознавания, сохранятся.

Для возобновления процесса распознавания необходимо повторно вызвать функцию start(). В результате ее вызова API снова начнет записывать звук и отправлять его на сервер. Следует иметь в виду, что при повторном запуске распознавания в параметрах функции start() можно переопределить только callback-функции (кроме initCallback). Все остальные настройки (API-ключ, формат звука, языковую модель и др.) при возобновлении распознавания изменить нельзя.

```
// Приостановка процесса распознавания. Функция вызывается без аргументов.
streamer.pause();
...
// Для возобновления процесса распознавания необходимо
// повторно вызвать функцию start().
streamer.start({
  // Переопределяем dataCallback.
  dataCallback: function (text) {
```

Примечание. При возобновлении процесса распознавания браузер не будет повторно запрашивать доступ к микрофону.

Процесс распознавания в JavaScript API.

Процесс распознавания в JavaScript API состоит из следующих этапов:

1. Запрос доступа к микрофону.
2. Захват аудиопотока с микрофона, его обработка и отправка на сервер.
3. Обработка финального результата.

Ниже подробно описаны каждый из трех этапов.

1. Запрос доступа к микрофону

Для получения доступа к микрофону в API используется функция MediaDevices.getUserMedia(). Эта функция вызывается автоматически сразу после запуска процесса распознавания. В результате ее вызова в браузере в верхней части страницы появится всплывающая панель с запросом доступа к

микрофону. До тех пор пока пользователь не даст разрешение, следующие действия в API выполняться не будут.

2. Захват аудиопотока с микрофона, его обработка и отправка на сервер

После того как пользователь даст доступ к микрофону, API инициализирует сессию — выставит настройки распознавания и установит соединение с сервером по технологии WebSockets. Для установки соединения на сервер отправляется специальный запрос, содержащий настройки распознавания и сведения о клиенте. В случае успеха сервер вернет UUID сессии, в противном случае — сообщение об ошибке (например, если указан неверный API-ключ).

После успешной инициализации сессии API начинает захват аудиопотока с микрофона. Захваченный аудиопоток преобразуется в нужный формат и отправляется на сервер частями. В ответ на каждое сообщение сервер возвращает *промежуточный результат* — распознанный фрагмент аудиопотока. Промежуточные результаты API записывает в буфер и передает в callback-функцию.

Примечание. Для захвата аудиопотока с микрофона используется Web Audio API.

3. Обработка финального результата

Как только в момент записи звука наступает тишина длительностью в несколько секунд, API прекращает захват аудиопотока и завершает соединение с сервером. Полученные промежуточные результаты склеиваются в один текст, с которым затем выполняется специальное преобразование — расставляются знаки препинания, даты преобразуются в числа и т. д. Этот преобразованный текст является *финальным результатом распознавания*.

Контрольные вопросы

1. Что такое программное обеспечение систем распознавания речи?

2. Что такое системы генерации речи?
3. Как выглядит схема речевого взаимодействия человека и компьютера?
4. Подумайте, как могут выглядеть основные направления использования голосовых интерфейсов.
5. Какие сложности возникают в процессе автоматического распознавания речи?
6. По каким признакам, характеризующим основные возможности, можно классифицировать системы автоматического распознавания речи?
7. Каковы основные компоненты любой системы автоматического распознавания речи?
8. Что необходимо иметь в виду разработчику использующему модуль распознавания речи?
9. Каковы особенности программирования задач распознавания речи?
10. Каковы особенности программирования задач голосового управления?
11. Каковы особенности программирования задач синтеза речи?
12. В чем заключаются основные трудности при вводе речевых сообщений?
13. Назовите единицы речи при распознавании речевых сообщений.
14. В чем заключается настройка анализатора речи на голос оператора?
15. Опишите основные возможности программы Sphinx?

ЗАКЛЮЧЕНИЕ

Создание программных продуктов, реализующих новые численные методы и интеллектуальные алгоритмы распознавания речи, на сегодняшний день являются весьма актуальным и направлено на реализацию государственной стратегии в области развития информационных технологий.

Исследования по анализу и синтезу речи начались в середине прошлого века, однако трудности формального описания речевых сигналов, необходимость не только акустической, но и лексической и семантической обработки ставили сложные задачи перед наукой.

Характерные особенности речи для каждого диктора, зависимость качества распознавания от скорости произнесения слов и объема словаря, большое количество самих языков и наречий и сейчас серьезно затрудняют создание универсальных методов распознавания речи с помощью компьютеров. Наконец, в зависимости от задачи меняется допустимая точность распознавания – при стенографировании требуется распознавание всех значимых слов, а в диалоговых системах для формирования запроса к информационным ресурсам иногда достаточно распознать несколько ключевых слов во фразе. Задача обработки речи в ее полном смысле еще далеко не решена, несмотря на усилия многочисленных групп исследователей во многих странах мира.

На современном этапе развития информатизации страны в связи с отсутствием теоретических основ распознавания узбекской речи с учетом особенностей строения и восприятия речевого сигнала возникает сложная научная проблема разработки и обоснования обобщенного описания методов и моделей распознавания узбекской речи. Таким образом, задача распознования и синтеза речи остается актуальной.

СПИСОК СОКРАЩЕНИЙ (GLOSSARIY)

ANN - Artificial Neural Network.

ASR - Automatic Speech Recognition.

Cepstral коэффициенты - голосовой ответ импульса тракта.

Cepstrum - обратное преобразование логарифмического спектра.

CF - correlation function.

DSP – digital signal processing.

DTW - dynamic time warping.

FIR - finite impulse response.

HMM - Hidden Markov Models.

ICA - Independent Component Analysis.

IVR - Interactive Voice Response.

LPC - Linear Predictive Coding.

LVCSR — large vocabulary continuous speech recognition.

MFCC - mel-частотные cepstral коэффициенты.

MPEG – Moving Picture Experts Group.

PCA - Principal Component Analysis.

SALT - Speech Application Language Tags.

SDK - Software Development Kit.

Speech Pearl - интегрированная среда разработки телефонных приложений с распознаванием речи.

Sphinx – самый известный и наиболее работоспособный из открытых программных продуктов распознавания речи.

STT – Speech-to-Text.

SWIG - Simplified Wrapper and Interface Generator.

TTS - Text-to-Speech.

VFR - Variable Frame Rate analysis.

VoiceXML - Voicee Xtensible Markup Language.

Xcode - программа для разработки приложений под OS X и iOS, разработанная компанией Apple.

АДТ - алгоритм динамической трансформации.

АИМ - амплитудно-импульсной модуляцией.

АКФ - автокорреляционная функция.

Артикуляция - движения, выполняемые органами речи в процессе произнесения звуков.

АЦП - аналого-цифровой преобразователь.

БПФ – быстрые преобразование Фурье.

ДП - Динамическое программирование.

ДПФ - дискретное преобразование Фурье.

Линейное предсказание (англ. linear prediction) - вычислительная процедура, позволяющая по некоторому набору предшествующих отсчётов цифрового сигнала предсказать текущий отсчёт.

Нейронные сети – программные средства, моделирующие работу человеческого мозга.

СГС - системах голосового самообслуживания.

СММ - скрытая Марковская модель.

Фильтры верхних частот (high-passfilter) – процесс, который вырезает из спектра входного сигнала все частоты выше некоторой пороговой частоты.

Фильтры нижних частот (low-passfilter) – процесс, который удаляет из спектра входного сигнала все частоты, значения которых находятся ниже некоторой пороговой частоты, зависящей от настройки фильтра.

Фонема - минимальная смыслоразличительная единица речи.

Формантные частоты – частоты, активно участвующие в образовании речи.

ЦАП – цифро-аналогоый преобразователь.

ЦОС - цифровая обработка сигналов.

ЛИТЕРАТУРЫ

Основные литературы

1. Lawrence R. Rabiner and Biing-Hwang Juang, "Fundamental of Speech Recognition", Prentice Hall (1993).
2. Manolakis, Dimitris G. Applied digital signal processing: theory and practice / MIT Press, Cambridge, 2012, ISBN 978-0-521-11002-0 (Hardback).
3. Ivica Rogina, “Automatic speech recognition”, Carnegie Mellon University, 1998.
4. Christopher J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, Kluwer Academic Publishers, Boston, 1998.
5. Gold B., Morgan N. Speech and Audio Signal Processing. John Wiley and Sons, Inc, 2000.
6. Айфичер Э., Джервис Б. Цифровая обработка сигналов. Практический подход. / М., "Вильямс", 2004, 992 с.
7. Сергиегко А.В. Цифровая обработка сигналов – СПб.: Питер, 2002. – 608с.
8. Рабинер Л.Р., Шафер Р.В. Цифровая обработка речевых сигналов. Пер. с англ.-М.:Радио и связь,1981.

Дополнительные литературы

1. Мирзиёев Ш.М. Эркин ва фаровон, демократик ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимига киришиш тантанали маросимига бағишланган Олий Мажлис палаталарининг қўшма мажлисидаги нутқ / Ш.М. Мирзиёев. – Тошкент: Ўзбекистон, 2016. - 56 б.
2. Мирзиёев Ш.М. Танқидий таҳлил, қатъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қоидаси бўлиши

керак. Мамлакатимизни 2016 йилда ижтимоий-иқтисодий ривожлантиришнинг асосий якунлари ва 2017 йилга мўлжалланган иқтисодий дастурнинг энг муҳим устувор йўналишларига бағишланган Вазирлар Маҳкамасининг кенгайтирилган мажлисидаги маъруза, 2017 йил 14 январь / Ш.М. Мирзиёев. – Тошкент: Ўзбекистон, 2017. – 104 б.

3. Мирзиёев Ш.М. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Ўзбекистон Республикаси Конституцияси қабул қилинганининг 24 йиллигига бағишланган тантанали маросимдаги маъруза. 2016 йил 7 декабрь /Ш.М.Мирзиёев. – Тошкент: “Ўзбекистон”, 2017. – 48 б.
4. Мирзиёев Ш.М. Буюк келажагимизни мард ва олижаноб халқимиз билан бирга қурамиз. Мазкур китобдан Ўзбекистон Республикаси Президенти Шавкат Мирзиёевнинг 2016 йил 1 ноябрдан 24 ноябрга қадар Қорақалпоғистон Республикаси, вилоятлар ва Тошкент шаҳри сайловчилари вакиллари билан ўтказилган сайловолди учрашувларида сўзлаган нутқлари ўрин олган. /Ш.М.Мирзиёев. – Тошкент: “Ўзбекистон”, 2017. – 488 б.
5. Мирзиёев Ш.М. Миллий тараққиёт йўлимизни қатъият билан давом эттириб, янги босқичга кўтарамиз. / Ш. М. Мирзиёев. – Тошкент: Ўзбекистон, 2017. -592 б.
6. Ахмед Н., Рао К.Р. Ортогональные преобразования при обработке цифровых сигналов. Пер. с. англ. - М.: Связь, 1980 г. - 248 с.
7. Румшиский Л.З. Математическая обработка результатов эксперимента. - М.: «Наука». 1971 г. - 192 с.
8. Aubert X., Haeb-Umbach R. and Ney H. “Continuous mixture densities and linear discriminant analysis for improved context-dependent acoustic models”, Proc. of ICASSP, Vol. II, pp. 648-651 (1993).
9. Dan Tran, Michael Wagner and Tongtao Zheng, “A Fuzzy approach to Statistical Models in Speech and Speaker Recognition”. 1999 IEEE International Fuzzy Systems Conference Proceedings, Korea, 1275-1280.

10. Huang, N. E., Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, 1998: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc. R. Soc. London, Ser. A, 454, 903-995.
11. Das S., Bakis R., Nadas A., Hahamoo D. and Picheny M. “Influence of background noise and microphone on the performance of the IBM tangora speech recognition system”, Proc. of ICASSP, Vol. II, pp. 71-74 (1993).
12. Граничин О. Н., Измакова О. А. Рандомизированный алгоритм стохастической аппроксимации в задаче самообучения. Автоматика и телемеханика. 2005. № 8. С. 52 - 63.
13. Граничин О. Н., Поляк Б. Т. Рандомизированные алгоритмы оптимизации и оценивания при почти произвольных помехах. М., Наука, 2003.
14. «Создание параллельных методов, алгоритмов и программ анализа речевых сигналов на базе спектральных преобразований» Отчет по НИР, ТУИТ, 2017г. – 186 с.
15. Мусаев М.М., Ходжаев Л.К. Спектральный метод полиномиальной аппроксимации для цифровой обработки сигналов // Электронное моделирование. - 1987. т.9, № 6. - с. 30-33.
16. Мусаев М.М., Рахматов Ф.А. Методы обработки аудио сигналов на базе сигнальных процессоров //Вестник ТУИТ. - Ташкент, 2010. - №1.-с.10-14.
17. Мусаев М.М., Рахматов Ф.А., Шеров Ж.Э. Программа «Audio COMPACT» // Государственное патентное ведомство РУз. Свидетельство № DGU 02006. 28.07.2010 г. // Расмий ахборотнома. - 2010. - №8.
18. Мусаев М.М. Современные методы цифровой обработки речевых сигналов. // Вестник ТУИТ - № 2, 2017, с.2-13.
19. Ле Н. В., Панченко Д. П. Предварительная обработка речевых сигналов для системы распознавания речи // Молодой ученый. - 2011. - №5. Т.1. - С. 74-76.

Интернет ресурсы

1. Audio-Visual Speech Recognition (AVSR): http://www.intel.com
2. Cnews - С. Мельников: Точность распознавания речи. http://www.cnews.ru/reviews/index.shtml?2007/12/24/280965
3. Dragon NaturallySpeaking Solutions: http://www.dragonsys.com
4. Давыдов А.В. Теория сигналов и систем. http://prodav.narod.ru/signals/index.html.
5. IBM embedded ViaVoice Enterprise Edition: http://www.ibm.com/software/speech/
6. Speech Recognition Home: http://www.philips.com/speechrecognition/
7. Speeding Medical Documentation: http://www.provox.com
8. Voice Recognition Module: http://www.sensoryinc.com
9. Алгоритмы распознавания: http://speech-text.narod.ru/
10. Машеров Е. Цифровая обработка сигналов – некоторые основные понятия. http://www.nsi.ru/~EMasherow/DSP.htm
11. Распознавание речи в автомобильной индустрии, http://www.drive.ru/blogs/4fa5694b09b602b73900001c.html
12. Речевые Технологии для бизнеса - распознавание речи, http://speech2b.ru/rus/applications/app-war/
13. Синтез и распознавание речи. Современные решения - http://www.frolov-lib.ru/books/hi/index.html
14. Центр Речевых Технологий: http://www.speechpro.ru
15. Обработка речевых сигналов: http://www.iki.rssi.ru/magbase/RESULT/APPENDIX/fractan.boom.ru/sound.htm
16. Синтез и распознавание речи. Современные решения: http://www.frolov-lib.ru/books/hi/index.html
17. Speech Technology: http://www.speechtechmag.com/
18. Introducing Android: http://developer.android.com/

19. Speechtek: http://www.speechtek.com/
20. Development Resources: http://www.qt-project.org
21. The Platform for Open Innovation and Collaboration: http://eclipse.org/
22. Eamonn J. Keogh, Michael J. Pazzani Derivative Dynamic Time Warping, Section 1, page 2 30 июля 2016 года. (англ.): https://www.cs.rutgers.edu/~pazzani/Publications/sdm01.pdf
23. DTW Algorithm Review. Section 3.3 (англ.): http://www2.hawaii.edu/~senin/assets/papers/DTW-review2008draft.pdf
24. Алгоритм динамической трансформации временной шкалы: https://ru.wikipedia.org/w/index.php?title=Алгоритм_динамической_трансформации_временной_шкалы&oldid=92796276
25. Dynamic Programming Algorithms in Speech Recognition by Titus Felix FURTUNĂ: http://revistaie.ase.ro/content/46/s%20-%20furtuna.pdf
26. Автоматическое распознавание речи, используя скрытые Марковские модели: http://masters.donntu.org/2008/fvti/verenich/library/th_rus.htm

Алгоритмы распознавания речи

Учебное пособие для студентов направления подготовки бакалавров 5330500-Компьютерный инжиниринг ("Компьютерный инжиниринг", "ИТ-сервис", "Мультимедийные технологии") и магистров 5А330501-«Компьютерный инжиниринг («Проектирование компьютерных систем», «Проектирование прикладных программных средств», «Информационные и мультимедийные технологии», «Информационная безопасность, криптография и криптоанализ»).

Рассмотрено и рекомендовано к изданию на заседание кафедры «Компьютерные системы» от 25 сентября 2018 г., протокол №4

Рассмотрено и рекомендовано к изданию на научно-методическом Совете факультета «Компьютерный инжиниринг» от 15 октября 2018 г., протокол №9

Рассмотрено и рекомендовано к изданию на научно-методическом Совете ТУИТ от «___»_____2018 г., протокол №_____

Разработчик(и): М.М.Мусаев, Ф.А.Рахматов, А.К.Эргашев

Рицензенты: И.Х.Сиддиков, Э.Ш.Назирова

Ответсвенный редактор: М.М.Мусаев

Корректор: С.Х.Абдуллаева