

Вступительный слайд (Слайд 1)

Приветствие. Озвучивание: кто я такой, тема моей работы.

В наши дни всё большую популярность приобретают различные голосовые помощники (Siri, Cortana, Дюся) и голосовые меню (IVR - Interactive Voice Response). В их основе зачастую лежит не только распознавание речи, но и её передача для последующей обработки. **[В частности – на стороне сервера ИИ синтезирует речь для ответа. Другой пример – сохранение для самообучения; к примеру, Дюся сохраняет неизвестные ей запросы пользователей, чтобы использовать их в последующем.]**

Поэтому одна из главных проблем наряду с распознаванием речи – это то, что не существует настолько надёжных каналов связи, которые могут обеспечить полное отсутствие помех, воздействующих на передаваемую речь. Три наиболее популярных метода защиты от помех – это аппаратное улучшение канала связи, повышение отношения сигнал/шум, а также помехоустойчивое кодирование.

Первые 2 метода не гарантируют 100% защиту от помех, и в случае появления оных не могут определить и исправить возникшую ошибку. Поэтому в данной работе в качестве методов защиты от помех мы рассматриваем помехоустойчивое кодирование.

Слайд 2

Рисунок на данном слайде демонстрирует наиболее распространённую последовательность обработки речевого сигнала при кодировании (*при декодировании всё происходит в обратном порядке*). Следует заметить, что в отдельных случаях можно отказаться от некоторых этапов в помехоустойчивом кодировании. **[Например, при низкой вероятности возникновения ошибки можно отказаться от блочного кодирования и интерливинга]**

Речевое кодирование не является помехоустойчивым, поэтому данные с речевого кодера поступают на каналный. Его задача – сделать передаваемую информацию помехоустойчивой, т.е. дать возможность приёмнику обнаружить (и, в некоторых случаях, исправить) ошибки, которые возникают при передаче информации. Помимо этого, каналное кодирование может выполнять такие функции, как добавление управляющей информации и шифрование.

Слайд 3

Рассмотрим более подробно каждый из упомянутых этапов и начнём с речевых кодеков. Вокодер – самый простой вариант речевого кодека, он определяет номер фонемы (как следствие – любая речь превращается в однотипный вариант; фонема – минимальная смысловозначительная единица языка, не имеющая лексического или грамматического значения). Липридер, в свою очередь, определяет параметры фильтра для синтеза речевого сигнала (что позволяет сделать передаваемый сигнал более похожим на речь человека-источника: тон, интонация, тембр). ***Следует заметить, что речевой кодек обычно оценивают по таким критериям, как скорость работы, усреднённая субъективная оценка (MOS – Mean Opinion Score) и требуемая скорость передачи данных.***

Говоря про помехозащищённое кодирование, выделяют 2 вида каналных кодеков: блочные (кодируют информацию по частям - блокам) и свёрточные (кодируют сразу весь информационный пакет). Однако ни один из помехоустойчивых кодов не в состоянии восстановить правильную информационную последовательность, если ошибкам подверглось слишком много подряд идущих бит. Решением данной проблемы является интерливинг: если при передаче информация подверглась пачечным ошибкам, то на приёмной стороне после сборки последовательности для декодирования повреждённые биты окажутся на значительном расстоянии друг от друга.

Исходя из имеющейся информации, были проведены исследования, в ходе которых были рассмотрены существующие методы обработки речи и предложена своя реализация библиотек, предназначенных для обработки речи, а также методология выбора более подходящего варианта обработки речи с учётом имеющегося канала связи.

Слайд 4

В качестве языка программирования был выбран Си. *Причины выбора – не буду называть, но если спросят, то это было сделано по следующим причинам:*

*-язык Си наравне с ассемблерами используется для программирования микроконтроллеров – таким образом область применения написанных библиотек значительно расширяется
-написанное на классическом Си приложение будет кроссплатформенным (на уровне исходных кодов): данные библиотеки получат возможность встраиваться в приложения, написанные на разных ОС*

Сам код писался в свободно распространяемых текстовых редакторах.

Компиляторы: на Windows CL (поставляется вместе с Visual Studio) и MinGW (адаптированный под Windows GCC), на Linux – GCC.

Про тестирование – всё на слайде (*можно просто зачитать*).

Слайд 5

Перед разработкой библиотек было проведено их проектирование и поставлены следующие вопросы (*указать на слайд; прочитать*).

- Область применения – всюду, где требуется обрабатывать и передавать речевой сигнал (помимо упомянутых ранее голосовых помощников и меню - голосовой чат в играх, беспроводные усилители звука).
- Целевая аудитория – разработчики, которые не владеют предметной областью.

Следует заметить: библиотеки разрабатывались исходя из того, что их можно использовать независимо друг от друга (*в случае несовпадения формата входных данных предлагается применять паттерн адаптер*).

Слайд 6

Одной из вставших перед нами проблем было то, что мы заранее не знаем все требования (*показать на слайд*).

Исходя из этого, разработка библиотек шла по такой технологии проектирования ПО, как прототипирование. В пользу данной технологии были следующие факторы:

-нам известны не все требования (формат, в котором мы будем передавать данные в канал связи; скорость передачи данных; вероятность ошибки);
-данная технология позволила нам быстро увидеть некоторые свойства продукта и сравнить с желаемыми (универсальность, соответствие реально существующим методам).

Слайд 7

Для начала было решено создать объект для хранения речевого сигнала: помимо уже упоминавшейся ранее фонемы, речевой сигнал можно охарактеризовать следующими основными параметрами (*показать на слайд*).

Это параметры фильтра линейного предсказания и параметры сигнала возбуждения. Для справки: в стандарте GSM около 20 параметров речевого сигнала.

Слайд 8

Также существует множество стандартов речевых кодеков, наиболее популярные приведены на слайде (показать на слайд).

G.711 и G.726 – это стандарты для телефонии; 726 также применяется в некоторых камерах видеонаблюдения.

GSM (Global System for Mobile Communications) – глобальный стандарт цифровой мобильной сотовой связи.

iLPC (internet Low Bitrate Codec) – кодек для голосовой связи через интернет. Используется в Skype, Google Talk, Yahoo! Messenger.

Существующих стандартов по речевым кодекам крайне много, и на перечисление всех уйдёт много времени. Однако при детальном рассмотрении выясняется, что так или иначе все указанные кодеки имеют почти одинаковую структуру.

Слайд 9

Изучив структуры существующих речевых кодеров, было замечено, что существующие речевые кодеки отличаются только форматом представления данных и алгоритмами расчёта параметров речевого сигнала, имея при этом общую структуру.

Исходя из этого была предложена структура, удовлетворяющий большинству из рассмотренных кодеров (данная структура позволяет создать как вокодер, так и липридер).

Слайд 10

Библиотека для речевого кодирования выполнена на интерфейсах. Это позволяет разработчику определить, что ему нужно, но при этом оставляет за ним выбор алгоритмов для речевого кодека.

Слайд 11

Слайд для перехода от речевого кодирования к помехоустойчивому.

Речевое кодирование не является помехозащищённым, поэтому требуется ввести защиту от помех. Следует заметить, что в простейшем случае защита от ошибок заключается только в обнаружении ошибки.

Слайд 12

В ходе исследования из всех рассмотренных методов помехоустойчивого кодирования были выбраны и реализованы следующие методы, т.к. они отвечали следующим критериям: могут находить и исправлять ошибки; простота реализации; применяются в существующих каналах связи.

Следует заметить, что говоря о свёрточных кодах обычно подразумевают кодирование полиномом с задержкой (т.к. данный метод наиболее распространён). Однако для того, чтобы отличать между собой 2 представленных метода, автором (мною) введено такое название. Это же касается дополнения до чётности: термины «выколотое прямое – /итерированное произведение кодов» встречаются крайне редко, поэтому вместо них использовался термин «дополнение до чётности».

Слайд 13

Основной вопрос, который встаёт при выборе метода помехоустойчивого кодирования – это (показать на слайд и зачитать).

Для ответа на данный вопрос в первую очередь следует рассчитать вероятность возникновения ошибки при передаче информации. Ранее уже проводились исследования, связанные с расчётом вероятности возникновения ошибки в канале связи (источники (с 13 по 16) указаны в бакалаврской работе; в частности, данные исследования проводились академией Федеральной Службы Охраны РФ в 2013 году). После изучения данных исследований, стало известно, что

вероятность возникновения ошибки в канале связи составляет около 1% (с округлением в большую сторону).

Пример с iLBC: 400 бит / 30 мс / 13.33 кбит/с / 33 пакета/с

Слайд 14

Как говорилось вначале, эффективность речевого кодека оценивается по трём критериям: требуемая полоса пропускания, скорость работы и усреднённая субъективная оценка. Т.к. последние 2 критерия не всегда можно рассчитать однозначно, в основу данной методологии легло требование к полосе пропускания.

- 1) Сначала предлагается получить вероятность возникновения ошибки в имеющемся канале связи (рассчитать или узнать её, если данная информация имеется в спецификации к каналу связи/была рассчитана ранее).
- 2) Выбор метода защиты от помех – исходя из того, чтобы мы могли передать max информации.
- 3) В зависимости от доступного размера для полезной информации, можно определиться с кодеком: просто передача фоновой, передача основных параметров речевого сигнала или же полный набор параметров.

Слайд 15

В ходе бакалаврской работы было сделано следующее (показать на слайд).

Следует заметить: на фоне иных существующих методов выбора речевого кодека (экспертное оценивание проектных решений, или метод, в основе которого лежит согласование алгоритма речевого кодирования с цифровым каналом связи) предложенный метод является более гибким.