

Вступительный слайд (Слайд 1)

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

В наши дни всё большую популярность приобретают различные голосовые помощники (Siri, Cortana) и голосовые меню (IVR - Interactive Voice Response). В их основе лежит передача речи и приём речевой информации.

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 последовательность обработки речевого сигнала при кодировании *(при декодировании всё происходит в обратном порядке)*.

Речевое кодирование не является помехоустойчивым, поэтому данные с речевого кодера поступают на канальный. 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, канальное 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).

Говоря про помехозащищённое кодирование, выделяют 2 вида канальных 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

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.

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

- Область применения – всюду, где требуется передавать речевой сигнал (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 5

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 6

Данный слайд – напоминание того, как выглядит прототипирование (*рисунок позаимствован с первой лекции по курсу «Основы программной инженерии», лектор Ицыксона Владимир Михайлович*).

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.

Для удобства хранения данных параметров была создана сущность Voice_type (и её же предлагается использовать далее при работе с речевым сигналом).

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.

Слайд 9

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.

Далее приведены структуры двух упомянутых кодеков (в упрощённом варианте и чуть более подробном).

Слайды 10-12: речевые кодеры

В 10-ом слайде: $1/A(z)$ – синтезирующий фильтр с линейным предсказанием

По 12-ому слайду:

Учитывая всё вышесказанное, здесь приведена структура типичного речевого кодера.

Подводя итог проведённому блиц-обзору существующих кодеков можно сказать следующее: существующие речевые кодеки отличаются только форматом представления данных и алгоритмами расчёта параметров речевого сигнала, имея при этом общую структуру.

Как следствие, мы можем сказать, ЧТО должен уметь делать речевой кодек, но при этом ответ на вопрос КАК остаётся на усмотрение разработчика.

Слайд 13

Исходя из сказанного, для разработки речевого кодека были использованы интерфейсы и, как следствие, набор модулей, реализующих речевой кодек, состоит из них. Следует заметить: реализация речевого кодека – это нетривиальная задача. Применяя данную библиотеку, сторонний разработчик будет знать, ЧТО ему требуется реализовать для создания речевого кодека. КАК – пусть решает он сам.

Слайд 14

После завершения речевого кодирования мы должны сделать полученный код помехозащищённым. Для этого применяются следующие методы (**показать на слайд**). Они были выбраны по ходу исследования из всех рассмотренных методов помехоустойчивого кодирования, т.к. отвечали следующим критериям: могут находить и исправлять ошибки; простота реализации; применяются в существующих каналах связи.

Слайд 15

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 проводилось много исследований, связанных 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).

Слайд 16

При расчёте вероятности возникновения ошибки следует учитывать следующие факторы (показать на слайд и зачитать). Учесть все источники ошибки невозможно, поэтому в расчётах присутствует определённая погрешность.

С её учётом вероятность возникновения ошибки в канале связи без помехоустойчивого кодирования составляет...

Слайд 17

...примерно 1% (для большинства исследуемых каналов; с округлением в большую сторону).

На самом деле это довольно много. Если взять стандарт iLBC, то у него один из режимов передачи данных составляет 13.33 кбит/с: длина пакета в таком режиме составляет 400 бит. Всего за секунду разговора будет передано ~34 пакета, и в каждом из них будут ошибки.

По предлагаемой методологии выбора метода помехоустойчивого кодирования в данном случае можно предложить использовать только свёрточный код в сочетании с интерливингом: мы будем иметь не более 4 ошибок на 400 бит, что легко поправится любым из предложенных свёрточных кодов (применение блочных кодов будет излишним).

Слайд 18 – Спасибо за внимание!