

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное бюджетное образовательное учреждение высшего**  
**образования**  
**ЮГО-ЗАПАДНЫЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

---

На правах рукописи



Шиленков Егор Андреевич

**СТРУКТУРНО-ПАРАМЕТРИЧЕСКИЙ СИНТЕЗ ОРТОГОНАЛЬНОГО РЕЧЕВОГО**  
**КОДЕРА, АДАПТИВНОГО КРИТЕРИЯМ СТЕПЕНИ СЖАТИЯ И**  
**РАЗБОРЧИВОСТИ РЕЧИ**

Специальность 05.13.01 – Системный анализ, управление и  
обработка информации (технические системы)

**Диссертация на соискание учёной степени**  
**кандидата технических наук**

Научный руководитель  
кандидат технических наук,  
профессор А.М. Потапенко

Курск 2016

## Оглавление

<b>Введение .....</b>	<b>4</b>
<b>1 Анализ методик и выбор параметров для модели речевого кодирования .....</b>	<b>10</b>
<b>1.1 Определение актуальности обеспечения вариативностью средства потокового сжатия речи.....</b>	<b>11</b>
<b>1.2 Параметрический анализ методов кодирования и алгоритмов сжатия звука для определения критичных для качества свойств речевого сигнала .....</b>	<b>13</b>
1.2.1 Аналоговое сжатие звука.....	14
1.2.2 Компрессия за счёт адаптивного предсказания .....	17
1.2.3 Перцептивное кодирование.....	21
<b>1.3 Постановка задачи структурно-параметрического синтеза для реализации ортогонального речевого кодера адаптивного определённым критериям степени сжатия и разборчивости речи.....</b>	<b>24</b>
<b>1.4 Выводы по первой главе .....</b>	<b>26</b>
<b>2 Математическое и методическое приспособление средств компрессии .....</b>	<b>27</b>
<b>2.1 Адаптация ортогонального преобразования время-частота .....</b>	<b>27</b>
2.1.1 Применение дискретного синусного преобразования.....	27
2.1.2 Быстрое преобразование Фурье с прореживанием по времени.....	31
<b>2.2 Адаптация методик потокового сжатия без потерь информации.....</b>	<b>35</b>
2.2.1 Словарное сжатие без потерь в кодеке.....	35
2.2.2 Методика сжатия кадров по статическому словарю.....	37
2.2.3 Методика сжатия кадров по динамическому словарю .....	39
2.2.4 Методика сжатия кадра по принципу скользящего окна .....	42
<b>2.3 Выводы по второй главе .....</b>	<b>44</b>
<b>3 Синтез математической модели для функциональной и алгоритмической реализации устройства динамического кодера.....</b>	<b>45</b>
<b>3.1 Синтез псевдо-квадратурного зеркального банка фильтров в перцептивном кодировании.....</b>	<b>45</b>
<b>3.2 Обоснование и разработка алгоритма предиктора в ортогональном кодере</b>	<b>63</b>
<b>3.3 Определение параметров ортогональных преобразований .....</b>	<b>70</b>
3.3.1 Гибридная реализация адаптивного преобразования Фурье .....	70
3.3.2 Реализация быстрого дискретного косинусного преобразования.....	76
<b>3.4 Построение частотно-временной маски для ортогонального речевого кодирования на основе психоакустической модели слуха .....</b>	<b>78</b>

3.5 Выводы по третьей главе .....	86
<b>4 Структурный синтез адаптивного кодера и средства оценки его качественных свойств.....</b>	<b>87</b>
4.1 Функциональное взаимодействие моделей масок .....	87
4.2 Внедрение техники сжатия без потерь для ортогонального кодирования ....	91
4.3 Формирование выходного кадра и декодирование .....	93
4.4 Результирующая структурная схема кодера и предварительная оценка сжатия .....	97
4.5 Программное обеспечение имплементации речевого кодека .....	99
4.6 Оценка временных задержек сигнала в линии связи.....	101
4.7 Выводы по четвёртой главе.....	103
Заключение.....	104
Список сокращений и условных обозначений .....	106
Список терминов.....	107
Список литературы .....	109
<b>ПРИЛОЖЕНИЕ А Исходный код приложения «Имплементация речевого кодера»</b>	<b>119</b>

## Введение

**Актуальность темы исследования.** В процессах сбора, обмена и обработки информации одно из важных мест занимают вопросы её компактного описания. Это подтверждается ростом вида и числа оборудования преобразования и хранения данных, используемого в локальных и глобальных системах селекции, передачи и представления информации. При этом в ряде важных практических приложений объективно имеет место противоречие между потребностью повышения степени компактности (сжатия) описания исходных данных и необходимостью сохранения требуемого уровня их качества. В этой связи проблема повышения степени сжатия описания данных в системах сбора, обработки, хранения и представления является актуальной. Во всей совокупности задач, направленных на разрешение данных проблем, значительное место занимают задачи сжатия аудиоданных. И это связано с широким разнообразием целей и условий аудиообмена и, соответственно, разнообразием требований к степени сжатия и допустимому уровню качества воспроизведения речевых данных.

**Степень разработанности темы.** Вопросам сжатия данных в общем, и речевых в частности, начиная со второй половины XX века, постоянно уделялось должное внимание российской и зарубежной наукой. В трудах классиков Котельникова В.А., Давыдова А.В., Шпигеля М.Ф., Бернштейна Л.Р. обоснованы основополагающие вопросы цифрового преобразования речевых сигналов. В работах Сергиенко А.Б., Борискевича А.А., Ронжина А.Л., Алдошиной И.А., Скляра Б., Кестера У., Терхарда Э., Янга К. детально рассмотрены методы и способы реализации процедур сжатия звукового потока данных. Полученные в этих и других работах результаты создают необходимую научно-методическую и практическую базу для дальнейшего повышения эффективности процессов хранения и передачи аудиоданных в системах их сбора, обработки и представления.

В качестве одного из важных перспективных направлений исследований представляются такие как: более полный учёт нестационарности свойств исходных речевых сигналов и условий их формирования, разработка методов речевых кодеков с динамической структурой, алгоритмической и параметрической адаптацией. Достижение приемлемых результатов возможно, в основном, за счёт сжатия потока данных канального уровня в устройствах хранения буферных данных. Таким образом, преимущество достигается теми устройствами связи, в которых показатель качества предоставляемой услуги связи балансирует с пониженными скоростями передачи данных.

Начиная со второй половины XX века, с развитием цифровых технологий передачи данных параллельно предпринимались подходы к техникам оцифровки речевых аудио-

сигналов. Первые ИКМ кодеры применяли технологии неравномерного квантования для снижения разрядности кода не в ущерб качеству речи. Далее были замечены особенности произношения слов и фраз человеком, связанные с тем, что уровень громкости голоса в динамике скачкообразно измениться не может и описывается известными математическими функциями. Данная особенность была реализована в кодере с адаптивным предсказанием, который успешно применяется и в настоящее время в сотовой, интернет связи и т.д. В случае с большим количеством передаваемой информации получили развитие ортогональные кодеры, в которых реализована особенность неравномерного частотного восприятия информации человеком.

Однако главными требованиями к речевому общению являются не только критерии разборчивости, но и задержка сигнала. Так, в области критерия степени сжатия лидируют ортогональные кодеры. В тоже время минимальная задержка является главным преимуществом адаптивных кодеров с линейным предсказанием. В настоящее время ведутся активные исследования по поиску гибридных решений.

Объектом исследования выбраны устройства цифровой обработки звуковых сигналов в системах их хранения и передачи.

Предметом работы являются алгоритмы и процедуры сжатия потока речевых данных в реальном времени: психоакустическое маскирование, дискретное преобразование, кодирование, цифровая фильтрация, словарная компрессия и предсказание.

**Цель и задачи исследования.** Целью является повышение эффективности хранения и передачи речевой аудиоинформации в условиях высокой степени нестационарности свойств сигнала. Под повышением эффективности понимается уменьшение объёма аудиоданных при поддержании заданного качества воспроизведения аудиоинформации.

Научно–техническая задача состоит в разработке, исследовании и адаптации методов и алгоритмов кодирования звука, позволяющих оперативно согласовать параметры процедур сжатия нестационарного речевого сигнала для сохранения уровня разборчивости голоса при понижении скорости потока данных.

Использование данных методов в системах передачи, хранения и обработки информации позволит повысить степень их использования при заданных требованиях к её качеству. Таким образом, решается актуальная проблема ухудшения воспроизведения речевого сигнала при уменьшении его объёма.

Для достижения поставленной цели в работе сформулированы задачи.

1. Анализ текущей ситуации в области речевых кодеров для исследования перспективных структур и техник сжатия потока данных.

2. Анализ и исследование методов и процедур преобразования широкополосного звукового сигнала с целью определения наиболее значимых операций для повышения степени его сжатия.

3. Разработка математической модели процессов цифровой обработки и преобразования, обеспечивающей оперативную алгоритмическую и параметрическую адаптивность устройств кодирования и сжатия при высокой степени нестационарности свойств речевого сигнала.

4. Разработка и модификация методик и алгоритмов процедур сжатия в устройствах хранения и передачи сигнала, позволяющих сократить объем данных.

5. Синтез структурно–функциональной схемы речевого кодера, приспособленного к поставленным требованиям передачи речевой информации в реальном времени, и экспериментальная оценка результата сжатия данных.

**Научная новизна** состоит в следующем:

- разработана математическая модель, основанная на учёте нестационарности свойств речевого сигнала и особенностей его слухового восприятия, позволяющая провести многомерный анализ информации для выбора процедур сжатия и контроля потерь качества;

- реализованы алгоритмы прореживания речевого сигнала, основанные на математическом анализе его свойств, осуществляющие частотно-временные преобразования для уменьшения скорости передачи;

- предложена структурно–функциональная организация и программная реализация речевого кодера, отличающегося выбором перечня алгоритмов и методик сжатия и позволяющего оценить уровень качества воспроизведения речи при изменении объёма информации.

**Теоретическая и практическая значимость работы.** Теоретическая значимость определена как совокупность разработанных моделей и методик линейной и ортогональной цифровой обработки речевых сигналов:

- модель расчёта коэффициентов импульсной характеристики КИХ–фильтров банка полифазных фильтров, позволяющая управлять критерием выразительности речи диктора;

- математическая модель дискретных частотно-временных преобразований, осуществляющая анализ нестационарного сигнала, для управления критерием разборчивости речи;

- алгоритм предиктора позволяет выявлять невыразительные временные диапазоны речевого сигнала и выполнять сжатие путём прореживания, тем самым используя особенность восприятия нередуцированных звуков русской речи.

Практическая значимость:

– методика и алгоритм построения частотно–временной маски, реализующие прореживание при маскировании слуха, применение которых позволяет сжимать размер дискретных значений до 8 раз при условии разборчивости голоса;

– адаптированные методики сжатия без потерь: словарное сжатие по динамическому и статическому словарю кодовых деревьев; сжатие при помощи скользящего окна, использование которых в совокупности с прореживанием позволяет достигнуть степени сжатия в 14 раз;

- структурный синтез речевого кодера позволяет реализовать средство сжатия голосового сигнала, контролирующее качество воспроизведения речи и скорость потока данных, зависящих от выбранных процедур частотно-временных преобразований;

- методики сжатия могут быть использованы в учебном процессе по дисциплине «Цифровая обработка сигналов».

Работа выполнена по плану инициативных НИР 2011–2016 гг. кафедры Космического приборостроения и систем связи Юго-западного государственного университета:

– разработка материалов в пояснительную записку технического проекта инновационно–образовательного профиля в части разработки технологической и методической поддержки организации инновационного и образовательного процесса на базе использования данных ДЗЗ и ГИС–технологий. Разработка математических моделей в интересах модернизации и создания базовых элементов РНИС, заказчик - «НПК» РЕКОД», г. Москва;

– создание системы дистанционного обучения и учебно–тренировочного модуля в интересах организации подготовки и повышения квалификации специалистов в области использования РКД, заказчик - Администрация Курской области;

– исследование и разработка научно–технических путей создания инфологической системы информационно–аналитического обеспечения научных исследований ВУЗа, заказчик – РФФИ.

Результаты работы подтверждены актами о внедрении.

**Методология и методы исследований.** Для решения задач в работе использованы методы математического моделирования, теории вероятности и математической статистики, ортогональных преобразований, оконного преобразования, скользящего окна, цифровой обработки сигналов.

Для анализа входного и выходного сигнала проведена разработка программного обеспечения имплементации речевого кодера с использованием распределённых сред и параллельного объектно-ориентированного программирования.

Экспериментальная оценка производилась с использованием специально синтезированного модуля, разработанного по принципу модельно-ориентированного

проектирования и реализующего работу модели процедур сжатия, их анализ и регулирование путём построения спектров, временных диаграмм и аппаратного воспроизведения речи.

**Положения, выносимые на защиту:**

1. Математическая модель дискретных преобразований во временной и частотной области, реализующая анализ нестационарного сигнала.
2. Алгоритмы прореживания с помощью наложения частотно-временной маски и редуцированных предсказаний линейного предиктора;
3. Модифицированные методики сжатия ортогонального речевого потока без потерь.
4. Динамическая структура кодера как многофункциональный инструмент для разработки средства сжатия речевого потока и оценки качества воспроизведения.

**Степень достоверности и апробация результатов.** Достоверность научных результатов подтверждается:

- успешной программной имплементацией ортогонального речевого кодера;
- корректностью построения алгоритмов и выводов математических выражений для построения кодера;
- актами об успешном практическом применении результатов диссертации;
- полученные научные результаты прошли проверку на соответствие требованиям хранения и передачи информации в реальном времени математическим моделированием.

Отдельные вопросы были обсуждены на научно–технических конференциях и круглых столах: «Актуальные проблемы телекоммуникаций» (г. Курск, 24 апреля 2011г.); «Общество, современная наука и образование: проблемы и перспективы. Часть 10» (г. Тамбов, 30 ноября 2012г.); «Актуальные проблемы инфокоммуникаций» (г. Курск, апрель 2012г., 2013г., 2014г.); «Проектирование и технология электронных средств» (г. Владимир, ноябрь 2012г., декабрь 2013г.); «Роботизация вооружённых сил российской федерации» (ГНИИЦ РТ МО РФ, февраль 2016г.); «Пути повышения автономности робототехнических комплексов» (II Международный военно-технический форум «АРМИЯ-2016», 8 сентября 2016г.).

Основные результаты внедрены в учебный процесс кафедры Космического приборостроения и систем связи при преподавании дисциплин «Цифровая обработка сигналов и сигнальные процессоры в средствах подвижной радиосвязи», «Техника микропроцессорных средств в коммутации», «Основы построения инфокоммуникационных систем и сетей», «Сети и системы передачи информации».

**Публикации.** По материалам диссертации были сделаны следующие публикации: опубликовано 20 научных работ, из них 7 статей в реферированных журналах ВАК, 13 докладов на научно–технических конференциях.

**Личный вклад автора.** Модификация и адаптация процедур сжатия с потерями и без, организация экспериментальных исследований и формулирование основных научных положений сделаны автором самостоятельно.

**Структура и объем работы.** Диссертация содержит введение, 4 главы, заключение, список использованных источников, включающий 133 наименования и одно приложение. Основная часть изложена на 118 страницах. Общий объем работы – 137 страниц, включая 54 рисунка и 29 таблиц.

## 1 Анализ методик и выбор параметров для модели речевого кодирования

Рост числа цифрового оборудования для передачи данных пропорционально занимает все большее количество ресурсов связи. Если раньше особое внимание уделялось уплотнению частотных диапазонов, распределение которых по спектру строго регламентировано и заполнено, то на данный момент существенную роль в организации связи играет ограничение скорости потока данных с временным уплотнением, в первую очередь, на канальном уровне. Уменьшение пропускной способности цифрового канала противоположно росту числа подключённых абонентов, причём, вне зависимости от технологии доступа. Ввиду этого ассортимент услуг, используемых приложениями для организации потоковой связи реального времени, находится на низком уровне. Приоритет достаётся тем программным продуктам и аппаратным решениям, чей функционал остаётся на должном уровне при пониженной пропускной способности канала. Достижение приемлемых результатов возможно, в основном, за счёт сжатия канального потока данных. Таким образом, преимущество достигается теми системами, в которых показатель качества предоставляемой услуги связи балансирует с пониженными скоростями передачи данных (далее - битрейт)[68]. На примере речевого обмена, при понижении скорости битрейта первостепенной задачей является разборчивость фраз. В силу проявления подобных тенденций в развитии цифровых устройств общения реального времени, для разработки динамической функциональной системы анализа соответствия между разборчивостью речи и битрейтом необходимо определить перечень параметров и процедур цифровой обработки, её характеризующих.

Рекомендованные стандарты MPEG последних ревизий [12] провели разработку универсального контейнера для медиа информации, включающей аудио, видео и графические данные. Абсолютное большинство производителей аппаратуры и программного обеспечения приёма-передачи и обработки поддерживают данный тип транспорта и стараются не отставать от инноваций MPEG. В свою очередь, контейнеры поддерживают встраивание сторонних техник кодирования, что позволяет добиться быстрой адаптации существующего оборудования под новые алгоритмы.

Исследования интернациональной группы учёных [102] по установлению различий в распознавании и восприятии речи различных языков выявили нестационарность спектральных параметров в оцифрованных сигналах. Она заключалась в различии частотных диапазонов, содержащих основную энергетику звуков, что характеризовало выразительность и разборчивость речи при неоднородной скорости чтения дикторов на разных языках. С одной стороны, данная особенность позволила точнее присваивать звуковому потоку языковую

принадлежность, а значит распознавать его, с другой – затрудняла унификацию средств цифровой обработки сигнала.

Международные стандарты кодирования ISO не имеют структурной ориентации под различные языковые особенности и, напротив, унифицированы, а значит, содержат некоторую избыточность и не учитывают критерии выразительности и разборчивости лингвистических единиц. В силу данного предположения в диссертации проводилась работа с речевыми потоками, содержащими только русский язык.

## **1.1 Определение актуальности обеспечения вариативностью средства потокового сжатия речи**

Речевое общение – это неотъемлемая часть процесса передачи информации в различных родах деятельности человека, которая обеспечивает максимально оперативным взаимодействием многочисленные коллективы предприятий. Сложно недооценить роль прямого общения людей в управленческой деятельности. Результат выполнения поставленных задач, корректировка текущей деятельности и непосредственное вмешательство в работу одного человека или группы лиц невозможно без качественной организации передачи голосовых команд.

В рамках данной работы будут рассмотрены аспекты удалённого обмена речевой информацией, т.е. с использованием аппаратуры приёма и передачи, кодирования и сжатия, модуляции и демодуляции, шифрования, что в совокупности рассматривается как линия связи голосового общения.

Процесс разговора по линии связи можно разбить на две последовательные процедуры: взаимная идентификация абонентов и понятие сути передаваемой информации. Первое может быть достигнуто путём передачи служебной информации, например, номера телефона или текстовых определений. Но здесь возможна подмена данных, что делает определение личности говорящего по голосу первостепенным. Второе возможно при определённой выразительности (внятности) речи, тут вопрос определения личности становится менее актуальным. При этом на первый план выходят особенности восприятия русской речи, довольно подробно рассмотренные лингвистами в [20,41] и специалистами автоматической обработки в [17,40,67,123].

Анализ параметров и характеристик произношения проведён в биомедицинских научных исследованиях [22] и показывает частотно-временное распределение физического сигнала в линии передачи информации, а также математические модели его восприятия, которые

определяют ограничения на его представление и изменение. Экспертная группа по движущемуся изображению (Moving Picture Experts Group – MPEG)[96] более полувека занимается исследованиями в области психоакустики (науки о восприятии звуков человеком) и решением задач передачи аудио-информации, унифицированными результатами которых явились рекомендации семейства кодеков с неравномерным квантованием (G.71x)[46], адаптивным предсказанием (G.72x)[13] и спектральным преобразованием (MPEG-1 Audio /MP3)[55].

Предварительный обзор стандартов представления речевого сигнала в линии связи показал возможность широкого применения для пользователей, разговаривающих на различных языках, но, с другой стороны, узкую специализацию для средств передачи информации, что было продиктовано функциональными ограничениями и особенностями развития данных средств. Недостаточное внимание было уделено особенностям русского произношения. Следовательно, сопоставление лингвистических исследований, биомедицинских факторов и психоакустических характеристик русского языка предполагает получение положительных результатов сжатия при адаптации существующих техник передачи речи в условиях ограниченного ресурса линии связи.

Современные голосовые кодеры (вокодеры) обладают некоторой степенью вариативности, обусловленной пропускной способностью линии связи. Перечислим вариативные параметры и определим их влияние на качество воспроизведения:

- частота дискретизации для оцифровки аналогового первичного сигнала определяет спектральный диапазон звучания[69], и, несмотря на то, что частоты 8 КГц вполне достаточно для передачи основной энергетике [61], расширение диапазона увеличивает достоверность определения голоса говорящего и привносит комфортность в общение;

- разрядность и равномерность уровней квантования задают динамический диапазон[31], т.е. динамику изменения громкости голоса; так, обычное негромкое общение имеет небольшой диапазон – до 40 дБ, а художественное пение и громкая речь – до 70 дБ [53], что соответствует тринадцати- и двадцатитрёхразрядному коду, повышение разрядности способствует увеличению возможности различения голоса, шумов и посторонних звуков.

Минимальные параметры работы вокодеров сейчас - это 8000 Гц частоты дискретизации с разрядностью 8 бит [45]. Возникшее противоречие, почему разрядность кода не соответствует минимальным требованиям динамического диапазона голоса, будет проанализировано в следующем разделе.

В основном, современные вокодеры обладают вариативностью, лавируя между перечисленными параметрами и используя один алгоритм сжатия. Существуют редакторы обработки, хранения и записи звука [76], которые позволяют, используя библиотеки конечных

вокодеров, работать с большим набором параметров средств сжатия. Однако в процессе общения адаптивное переключение между техниками сжатия не представляется возможным, имеет место лишь уменьшение или увеличение перечисленных выше основополагающих параметров при условии внедрения служебной информации. В связи с этим разработка вариативного средства, позволяющего изменять не только параметры оцифровки, но и преобразовывать алгоритм своей работы, не вызывая при этом трудностей декодирования и обладая динамически изменяемым битрейтом, является актуальной задачей.

Для определения востребованности подобного средства сжатия необходимо обратиться к тем отраслям, в которых деятельность человека подвержена условиям быстрого изменения окружающей звуковой обстановки. К ним следует отнести: деятельность в плохих погодных условиях (крайнего севера, стихийных бедствий, штормов в море), проведение технологических работ, связанных с высоким шумом механизмов, нахождение внутри дорожного трафика, многолюдных мест, проведение специальных операций и т.д.

При установлении определённой степени корреляции в автономном режиме между полезным сигналом и динамическим шумом, возможно их разделение. Анализ непосредственно речи перед передачей в линию связи позволит оптимизировать трафик. В совокупности средство, обладающее способностью автономного анализа и структурной адаптацией к условиям внешней обстановки при динамическом изменении пропускной способности линии, является востребованным.

## **1.2 Параметрический анализ методов кодирования и алгоритмов сжатия звука для определения критичных для качества свойств речевого сигнала**

Началом обработки речи являются процессы, происходящие во входной цепи линии связи, где располагается первичное устройство преобразования (звукосниматель или микрофон), основная роль которого сводится к трансформации энергии воздушных колебаний в электрический ток, протекающий в первичных линиях [4,43,63,87]. От параметров и характеристик микрофона зависит точность передачи звучания голоса, его выразительность и разборчивость, поэтому целесообразно уделить внимание данному устройству.

Сотрудниками акустической лаборатории Microsoft [116] накоплен опыт первичной обработки звуковых и речевых сигналов путём пространственной ориентации микрофона в помещении для записи, исследованы процессы протекания и преобразования энергии голоса в первичном звуковом тракте, обусловлены процедуры аналоговой компрессии сигналов, на которых подробно остановимся в данном разделе.

### 1.2.1 Аналоговое сжатие звука

Как было отмечено в предыдущем разделе, современные вокодеры используют параметр разрядности восемь бит как минимальный вместо определённых психоакустикой тринадцати [11], что есть ключ к пониманию применения неравномерной шкалы квантования как средства достижения компрессии.

Впервые неравномерное квантование для звука было применено в стандарте G.711[66,94], где определены два алгоритма распределения уровней квантования по отношению к равномерной шкале. Первый алгоритм имеет название «Мю-закон», используется в мультиплексах E1 и представлен выражением:

$$y = \frac{\ln(1 + \mu \cdot |x|)}{\ln(1 + \mu)}, \quad (1.1)$$

где  $\mu = 255$  – коэффициент преобразования для разрядности восемь бит;

$x$  – равномерное квантованное значение;

$y$  – преобразованное неравномерное значение.

С целью наглядного представления функции преобразуем формулу 1.1 в вид, соответствующий тринадцатитбитному динамическому диапазону (0-40 дБ). Для этого определим диапазон входных значений  $x = 0 \dots 8191$ , учтём при масштабировании значения аргумента под модулем, а также адаптируем шкалу функции на восемь бит:

$$y = \frac{\ln(1 + \mu \cdot \left| \frac{x}{8191} \right|)}{\ln(1 + \mu)} \cdot 255 \quad (1.2)$$

На рисунке 1.1 изображён график преобразованного выражения 1.2. Проанализировав отдельные участки, можно сделать вывод о том, что значения нижнего диапазона квантованы гораздо точнее (т.е. отношение количества полученных отсчётов к начальным примерно 1 к 4), чем значения высокого (отношение 1 к 10).

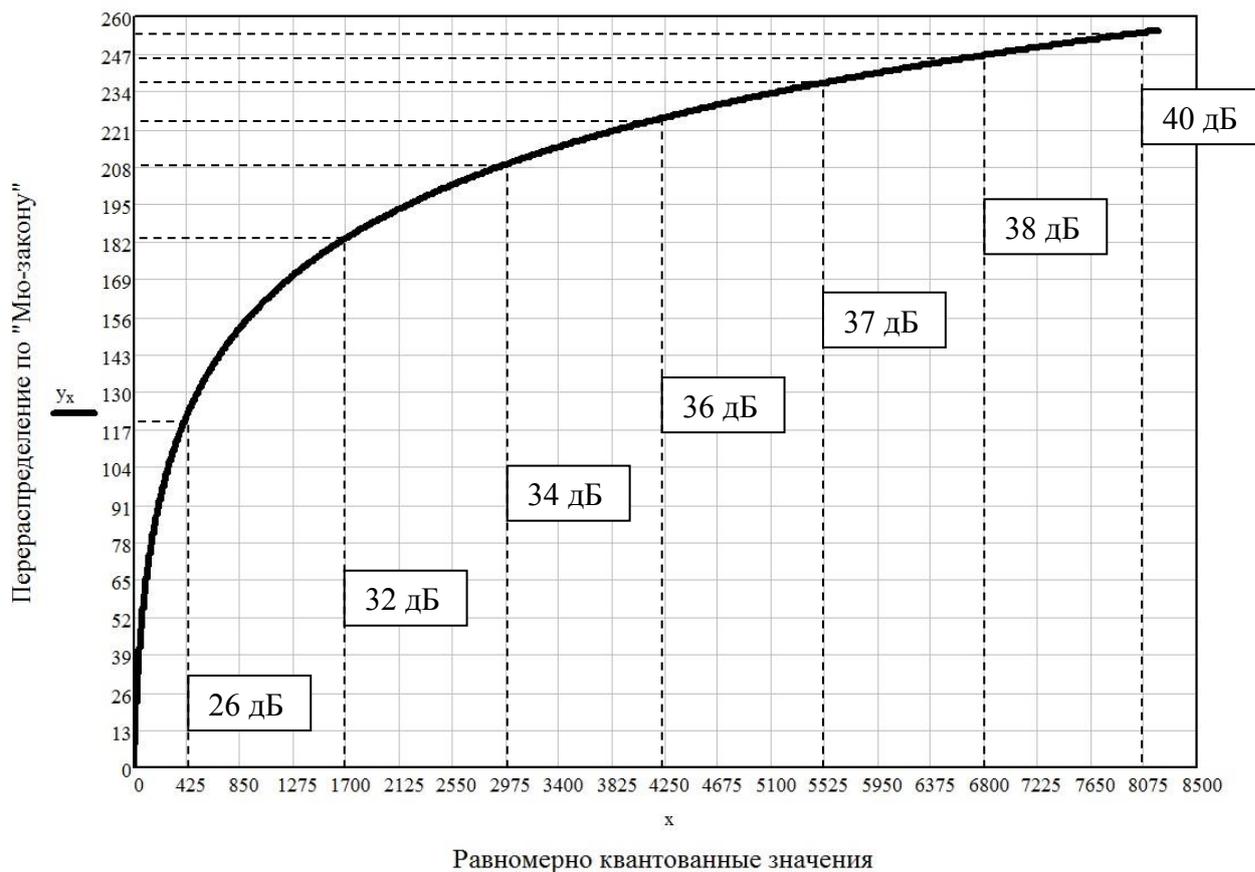


Рисунок 1.1 – График Мю-распределения для тринадцатибитного речевого потока

В данном случае тихие звуки будут воспроизведены на конечном устройстве с меньшим количеством искажений в отличие от громких.

Второй алгоритм называется «А-закон» [56,109] и используется в стандартах сотовой связи. Здесь неравномерность квантования задана двумя выражения: линейным и квадратичным:

$$y = \begin{cases} \frac{A \cdot x}{1 + \ln A}, & \text{для } x \leq \frac{1}{A} \\ \frac{1 + \ln(A \cdot x)}{1 + \ln A}, & \text{для } \frac{1}{A} < x \end{cases}, \quad (1.3)$$

где  $A = 87,6$  – коэффициент преобразования для разрядности восьми бит;

$x$  – равномерное квантованное значение;

$y$  – преобразованное неравномерное значение.

Подобно предыдущему перераспределению преобразуем формулу 1.3 для динамического диапазона 40 дБ. Для этого найдём номер коэффициента, соответствующего  $1/A$ , он будет равен

$8191/87,6 \approx 94$ . И последовательно адаптируем обе части выражения 1.3 для разрядности тринадцать бит:

$$y = \begin{cases} \frac{A \cdot \frac{x \cdot 255}{8191}}{1 + \ln A}, & \text{для } x \leq 94 \\ \frac{1 + \ln(A \cdot \frac{x}{8191})}{1 + \ln A} \cdot 255, & \text{для } 94 < x \end{cases}, \quad (1.3)$$

На рисунке 1.2 представлен график распределения по «А-закону». Стоит отметить большую крутизну характеристики, что означает увеличение точности квантования в нижнем диапазоне. На практике два алгоритма представляют собой технику аналогового сжатия. За счёт криволинейного представления диапазона громкости реализовано удаление более громких компонентов звука из всего динамического диапазона. Несмотря на высокий уровень компрессии (тринадцать бит против восьми, что в тридцать два раза отличается по битрейту), потери в энергетике передаваемого голоса также высоки. G.711 не учитывает параметры внешнего шума и адаптирован только для речи [51]. Однако подобный способ сжатия будет полезен при разработке вариативного кодера в условиях динамической адаптации статических параметров и функций данного кодера.

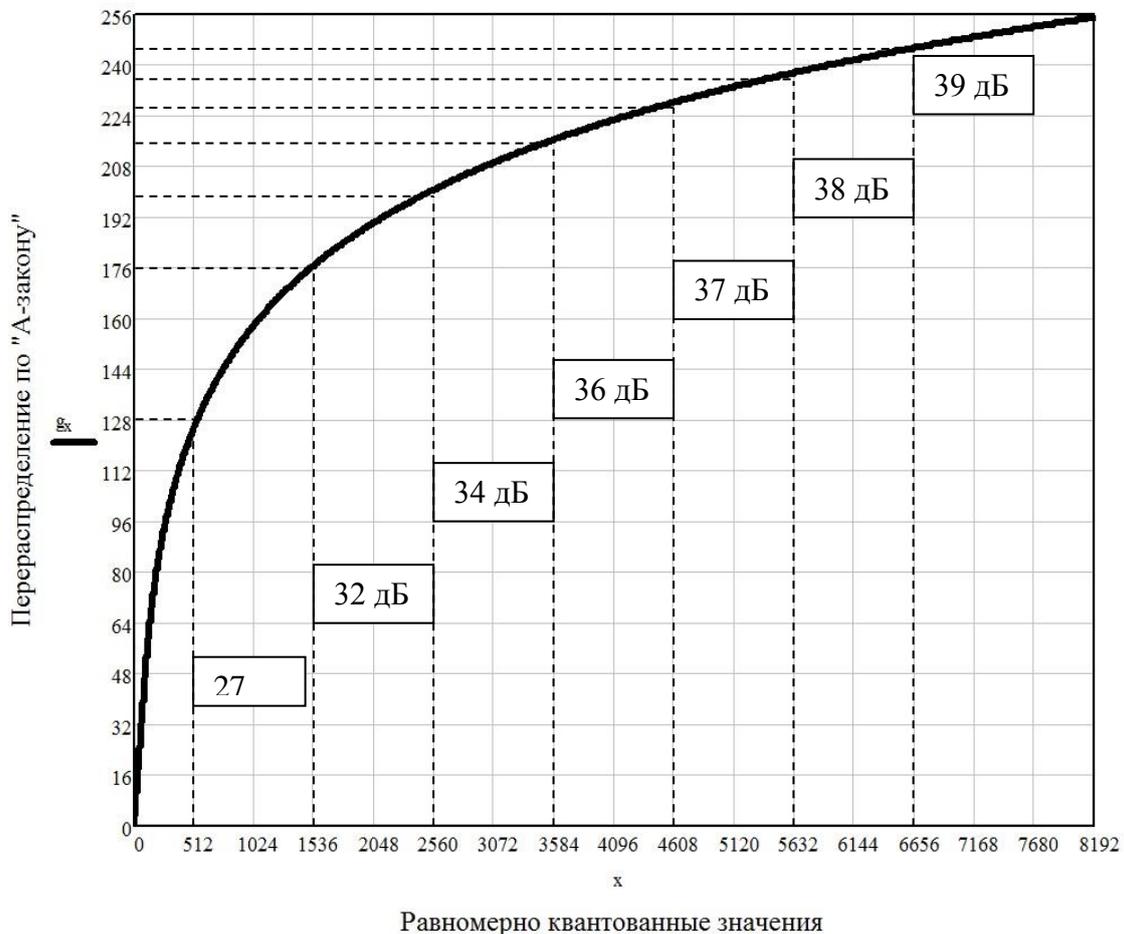


Рисунок 1.2 – Передаточная характеристика кодера G.711 с применение «А-закона»

Результатом проведённого анализа параметров аналогового сжатия следует считать выявление статических параметров, отвечающих за качество передачи речевой информации в конкретном динамическом диапазоне. Таким образом, отнесём процедуру битового распределения (как части аналого-цифрового преобразования) к одной из ключевых при синтезе устройства сжатия речи.

### 1.2.2 Компрессия за счёт адаптивного предсказания

Анализ способов предсказания позволит выявить параметры и функции кодеров, отвечающие за соответствие инерционности и степени сжатия сигнала. Параметрические подходы к предсказанию речи предпринимаются уже на протяжении полувека и их развитие хорошо представлено в труде германских учёных, синтезировавших объединяющий метод сжатия сигнала с динамическими полосовыми характеристиками [9,27]. Он основан на использовании шумового сигнала в качестве банка предсказаний, тем самым уменьшая разрядность выходного дифференциального значения, что даёт выигрыш в компрессии речи. Диапазон и характер шума выбирается исходя из фонетики диктора путём проведения анализа его произношения.

На рисунке 1.3 представлена структурная схема кодера, основанного на линейном предсказании с кодовым возбуждением (code excited linear prediction (CELP)) [104,110]. Данный кодер взят за основу для разработки кодера в области временного (линейного) кодирования. В частности, в работе будет использован стеганографический метод передачи речевой информации.

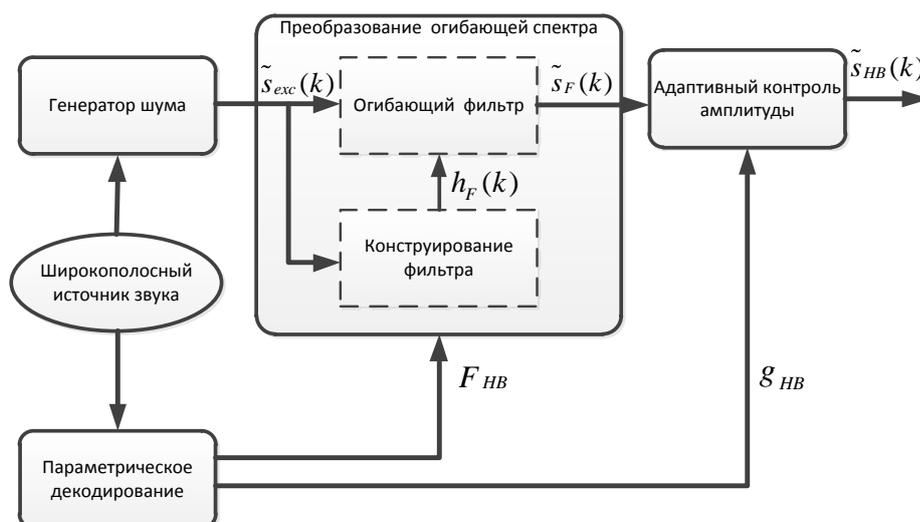


Рисунок 1.3 – Кодер CELP с шумовым кодовым возбуждением:

$F_{HB}$  - субполосные коэффициенты,  $g_{HB}$  - усиливающий фактор,  $\tilde{s}_{exc}(k)$  - шумовое возбуждение,  $\tilde{s}_F(k)$  выход огибающего фильтра,  $h_F(k)$  - сигнал фильтра,  $\tilde{s}_{HB}(k)$  - выходной сигнал кодера

В основу потокового сжатия временных отсчётов положен функционал кодера с адаптивной дифференциальной импульсно-кодовой модуляцией (АДИКМ) [16,65,79]. Особенностью данного кодера является «предсказание» последующего временного отсчёта на основе анализа предыдущих. Сжатие выходного потока происходит за счёт вычисления разницы между предсказанным отсчётом и реальным, полученное значение квантуется кодом с меньшей разрядностью. Структурная схема и математическое выражение представлены ниже.

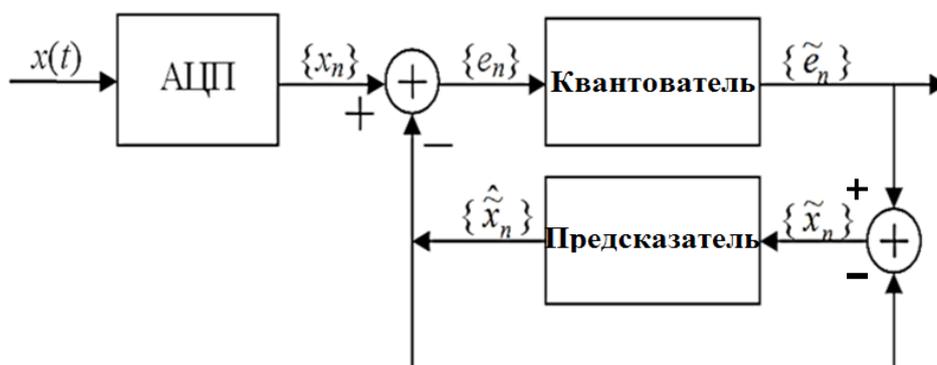


Рисунок 1.4 - Структура дифференциального кодера с обратными связями

В работу средства сжатия внедрена отрицательная обратная связь, описанная простой формулой:

$$e_n = x_n - \hat{x}_n,$$

где  $e_n$  - входной цифровой сигнал квантователя;

$x_n$  - входной вектор данных от АЦП;

$\hat{x}_n$  - выходной вектор предсказателя.

Под вектором здесь понимается последовательность одномерных дискретных отсчётов. Подразумевается, что результат вычитания предсказанного вектора из вектора, сформированного АЦП, окажется много меньшим  $x_n$  и может быть переквантован меньшей разрядностью, что и обеспечит сжатие.

Функционал предсказателя многообразен и является отличительной чертой семейств потоковых кодеков. В зависимости от временных характеристик линии передачи данных изменяется структура и последовательность процедур предсказания. В системах с коммутацией каналов [78] векторы цифровых отсчётов делятся на равные длины и обрабатываются автоматами сжатия, нередко имея в своей структуре заранее заготовленные сэмплы (отрезки речи)[42] и адаптивно-настраиваемые фильтры (вокодеры, подробно рассмотренные в [81]). В системах с коммутацией пакетов [77] существует возможность варьировать длину сэмпла и распараллеливать потоки данных кодера, что позволяет одновременно со сжатым аудио передавать служебные данные, способствующие автономной настройке декодера относительно кодера. Однако в таких линиях задержка сигнала является далеко не постоянной, что накладывает ограничение на их применение. Распространённость систем с коммутацией пакетов во многом предопределила развитие перцептивных кодеров, которые будут проанализированы в следующем разделе.

В результате анализа различных средств сжатия звука, использующих предсказание, было составлено математическое выражение, обеспечивающее перестройку режима работы предиктора наподобие рекурсивного цифрового фильтра с обратной связью:

$$\hat{\tilde{x}}_n = \sum_{k=1}^K b_k \cdot \tilde{x}_{n-k} - \sum_{m=1}^M a_m \cdot \tilde{x}_{n-m}, \quad (1.4)$$

где  $n$  - индекс временного отсчёта;

$m$  - индекс коэффициента рекурсии;

$b$  - коэффициент усиления предсказания с индексом  $k$  ;

$a$  - коэффициент усиления рекурсии;

$K$  и  $M$  - общее количество коэффициентов предсказания и рекурсии.

Под рекурсией в данном выражении определена скорость подстройки работы предсказателя относительно скорости нарастания или изменения входного оцифрованного сигнала.

С целью повышения эффективности сжатия в адаптивных предсказателях используют кодовые книги вместо генераторов шума. В работах [11,121,122] показана модель адаптивного построения подобного буфера сигналов для предсказания, однако в нём будут содержаться форманты наречий языка Хинди ввиду этнической принадлежности авторов. Тем не менее, в данной статье также изложен подход к изменению скорости воспроизведения потока, тем

самым сокращая его длительность, и проведение обратного изменения в декодере. Представленная процедура не относится к индивидуальным свойствам того или иного языка, поэтому может быть использована и в нашем исследовании. Выражение ниже содержит краткую результирующую выдержку питча, формализованную для текущей работы:

$$y[n] = x[n] + y[n - N],$$

где  $x[n]$  – входные временные отсчёты;

$y[n]$  – выходные линейные отсчёты;

$N$  – длина ускоренного (вырезанного) сэмпла.

Определим степень инерционности выходного сигнала в зависимости от режима работы предсказателя. В первом случае будут установлены параметры, обеспечивающие глубокую обратную связь и быструю перестройку фильтра, во втором – напротив, неглубокую обратную связь и медленную скорость. В качестве аналитического сигнала выберем функцию  $f(x) = \sin(x^2)$ , которая увеличивает собственную циклическую частоту по параболическому закону. Выясним скорость адаптации предсказателей, реализовав два варианта по формуле 1.4.

Выберем вектор усиления  $b = [1, 0, 1, 0, 1, 0, 1, 0]$ , вектор медленной рекурсии  $a = [1, 1, 1, 1]$  и быстрой рекурсии  $a = [1, 1, 1]$ , выполнив масштабирование, выражение 1.4 в первом случае примет вид:

$$X1_x := 0.2 \cdot \sum_{k=0}^K (b_k \cdot f_{x-k}) - 0.25 \cdot \left[ \sum_{m=0}^3 (a_m \cdot f_{x-m}) \right]$$

Отобразим графически полученное выражение для медленного предсказания

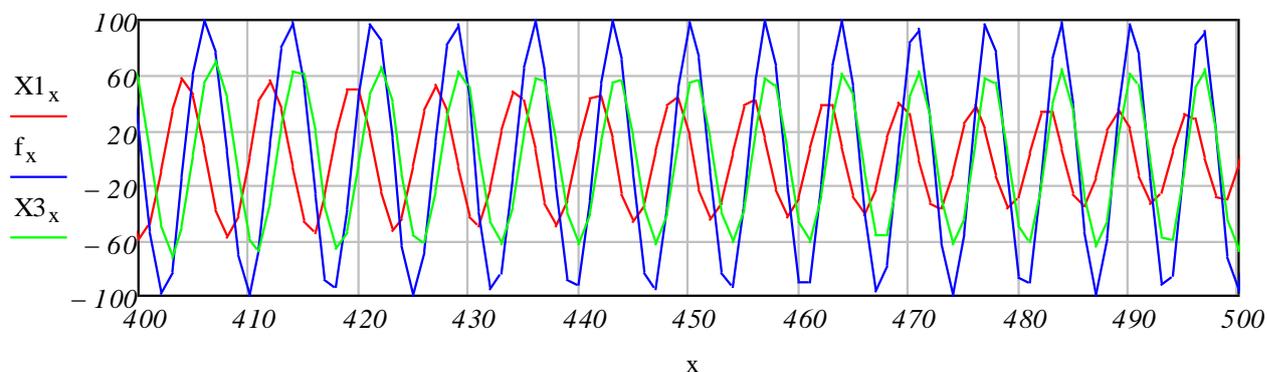


Рисунок 1.5 – Медленное предсказание,  $f_x$  – исходный сигнал,  $X1_x$  – предсказанный сигнал,  $X3_x$  – сигнал на выходе кодера

Во втором случае функция 1.4 примет вид:

$$X2_x := 0.2 \cdot \sum_{k=0}^K (b_k \cdot f_{x-k}) - 0.5 \cdot \left[ \sum_{m=0}^I (a_m \cdot f_{x-m}) \right]$$

График взаимодействия процесса предсказания будет выглядеть:

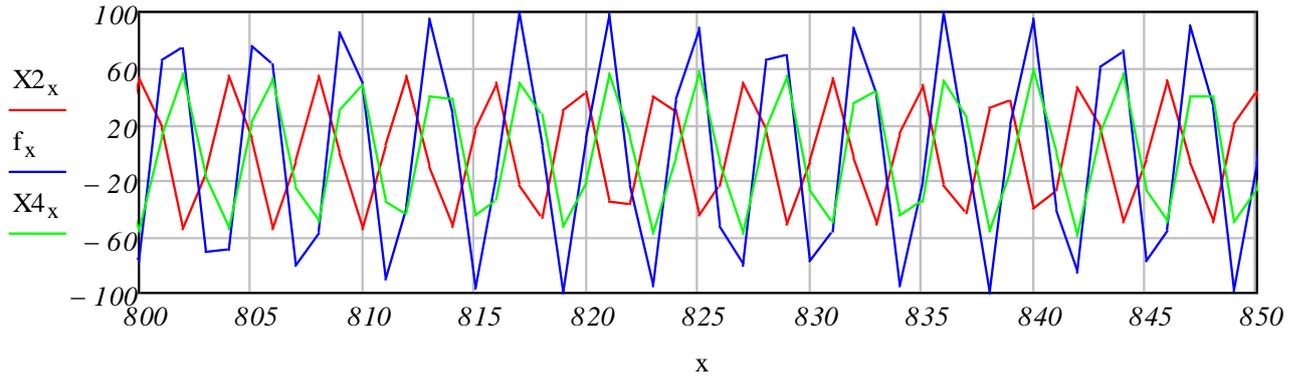


Рисунок 1.6 – Быстрое предсказание,  $f_x$  – исходный сигнал,  $X2_x$  – предсказанный сигнал,  $X4_x$  – выходной сигнал кодера

Как видно из двух графиков, оба процесса получают эффективность не сразу, а по прошествии определённого числа отсчётов. Чем выше частота исследуемого сигнала, тем больше коэффициент сжатия, но с ростом инертности, вносимой рекурсивной обратной связью, уменьшается время начала процесса предсказания.

Таким образом, в результате проведённого исследования установлена параметрическая взаимосвязь элементов адаптивного дифференцированного взаимодействия, в совокупности осуществляющего сжатие временного потока без потери качества воспроизведения.

### 1.2.3 Перцептивное кодирование

Медицинские исследования активности коры головного мозга доказывают способность адаптации восприятия звуковой информации человеком [7]. Именно в коре мозга происходит анализ и осмысление потока нейронных импульсов от звукового нерва, расположенного в раковине слухового аппарата [112,129]. Так, с одной стороны, поток звука в природном его проявлении является избыточным для восприятия и, наоборот, если звучание является прерывистым или зашумлённым, внутренний предиктор, основанный на мышлении,

достаивает недостающие фрагменты речи, используя опыт или собственную книгу предсказаний [5,6].

Таким образом, слух человека обладает свойствами инертности и адаптации, что также реализовано при построении семейств перцептивных кодеров [34]. Если человек долгое время находился в шумной обстановке и оказался в тихом месте, то некоторое время ему сложно будет воспринимать тихие звуки типа шёпота. И наоборот, при долгом нахождении, например, в тихом читальном зале, резкое воздействие громким звуком может привести к временной потере слуха, или в тяжких случаях – к контузии. Впоследствии, как правило, после такого поражения слуховой системы человек не может вспомнить услышанное до поражающего звука. Данная особенность, впрочем, распространяется и на другие органы чувств и является особенностью всего организма.

Приведём другой абстрактный пример, когда во время негромкого разговора неожиданно появляется монотонный, но громкий звук, наподобие клаксона автомобиля, гудка поезда или тестового телевизионного сигнала, означающего конец эфира. Подобные звуки не только причиняют дискомфорт, как описано в предыдущем абзаце, но и делают невозможным общение во время их звучания. Хотя с точки зрения спектрального представления эти помехи могут рассматриваться как узкополосные, напротив, наша речь – это сверхширокополосный сигнал [82] с диапазоном частот от 300 до 3400 Гц. Однако под действием громкой, но узкой помехи мы перестаём слышать.

Учитывая приведённые негативные эффекты, специалисты в области психоакустики установили математические взаимодействия для слуха и различных звуков [1,18,35,92]. Так, на текущий момент в биомедицине и цифровой обработке звуковых сигналов используются модели временного и частотного маскирования. Под маскированием понимается невосприимчивость к звукам определённой частоты и громкости ниже определённого порога – маски.

Но и без посторонних шумов диапазон слуха человека не линеен, т.е. какие-то звуки мы слышим хорошо, а какие-то – только если они громкие. И тут не должно быть сомнения, что именно речевой диапазон воспринимается нами лучше всего, а нижние и верхние частоты – намного хуже. Отсюда вытекает желание у большинства людей при воспроизведении музыкального контента повисить в эквалайзере регуляторы басовых и верхних частот и понизить «середину».

Если рассматривать диапазон слуха более подробно, то можно разделить его на критичные зоны, совместная корреляция которых приводит к определению тонов и октав. На слух человеку хорошо слышно разделение на восемь нот, что в совокупности образует октаву, и это технически означает увеличение частоты в два раза. Здесь тоже установлены

математические значения и определены критичные диапазоны – барки [62,133]. При определении звукового давления используют единицы высоты тонального сигнала, измеряемого в мелах. Здесь привязка шкалы мелов осуществляется непосредственно к звуковому нерву – волосковой клетке внутреннего уха, которая преобразует сигнал в нервный импульс для мозга.

В качестве анализирующего речь средства в современных кодерах используются преобразования временных дискретных отсчётов в частотные. Базовым тождеством в этой сфере является дискретное преобразование Фурье (ДПФ) [60,105]:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i k n}{N}},$$

где  $x_n$  - входной вещественный временной сигнал,  $n = 0, \dots, N-1$ .

Обратная форма преобразования имеет вид:

$$x_n = 1/N \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i k n}{N}},$$

где  $X_k$  - входной комплексный частотный сигнал,  $k = 0, \dots, N-1$ .

Главным преимуществом ДПФ можно считать полную обратимость процедуры, а значит и отсутствие искажений в восстановленном сигнале декодера. Но ввиду комплексности частотного сигнала поток данных увеличивается в два раза, следовательно, данная избыточность негативно повлияет на процесс сжатия. Поэтому целесообразно применять ДПФ для автономных процедур кодера.

Для того чтобы избавиться от комплексных чисел и оперировать только вещественными необходимо использовать чётные преобразования, коими являются функции косинуса. Формулы прямого и обратного косинусного преобразования имеют вид [113,114]:

$$X_k = \sum_{n=0}^{N-1} x_n \cos \left[ \pi k \frac{(2n+1)}{2N} \right]$$

$$x_n = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X_k \cos \left[ \pi k \frac{(2n+1)}{2N} \right]$$

Косинусное преобразование не является обратимым, следовательно, вносит определённые [62] искажения в восстановленный сигнал. Основное их проявление заключается в зеркальном наложении спектра высоких частот относительно низких, что, в свою очередь, может быть компенсировано использованием весовых окон и фазовых поправок [3]. В результате использования процедуры чётного ортогонального преобразования появляется возможность повысить степень компрессии звукового потока.

Подводя итог обзора перцептивных особенностей слуха человека, выделим ключевые модели и их вариативные параметры для синтеза кодера:

- модель временного маскирования с параметрами длительности временного промежутка неслышимых звуков до и после громкого воздействия;

- модель временного маскирования с функцией маскирования по спектру относительно сильного тона;

- модель абсолютного порога слуха человека с функцией неравномерного распределения слышимого диапазона;

- распределение по критическим частотным диапазонам – тональностям.

Для реализации представленного перечня в качестве основного инструмента выбрано чётное дискретное косинусное преобразование.

### **1.3 Постановка задачи структурно-параметрического синтеза для реализации ортогонального речевого кодера адаптивного определённым критериям степени сжатия и разборчивости речи**

В условиях непостоянной пропускной способности канала, воздействия звуковых шумов и перцептивных свойств голоса, основной задачей средства связи является адаптивная цифровая обработка сигнала [54,111]. В определение адаптивности вложено положение о возможности выбора методов кодирования, перестройки их параметров и управления разборчивостью воспроизводимой речи.

Оценка качества может быть выполнена двумя способами: экспертной группой слушателей, согласно ГОСТ-51061[24] и ГОСТ-53801-2010[28] по заданной шкале оценок, и спектральным анализом эталонного сигнала от диктора и принимаемым от аудитора.

Анализ и исследование методов и процедур преобразования широкополосного звукового сигнала проводится с целью определения наиболее значимых операций для повышения степени его сжатия. В результате выявлен перечень перспективных операций преобразования оцифрованного потока.

Имея перечень актуальных средств преобразования, ставится задача их подготовки и адаптации для математической реализации. Здесь идёт речь о разработке модели процессов цифровой обработки, обеспечивающей оперативную алгоритмическую и параметрическую адаптивность устройств кодирования и сжатия при высокой степени нестационарности свойств звукового сигнала, что включает в себя две части работы. Первая - подготовительная:

- развёрнутое представление алгоритмов подготовки широкополосного сигнала и их преобразование для параллельных процедур наложения временной маски;
- адаптация функциональных представлений спектральных параметров для временного потока при реализации алгоритмов частотного маскирования;
- методическое приспособление алгоритмов потокового сжатия без потерь для компрессии речевого потока и формирования выходного кадра.

Вторая содержит синтез элементов модели:

- построение процедур аналитического разделения широкополосного сигнала для определённых критических диапазонов речи;
- определение параметров и математическая реализация спектральных преобразований при проведении процедуры создания частотной маски.

Реализованный математический аппарат позволит создать методики и алгоритмы формирования процедур сжатия в устройствах хранения и передачи сигнала, позволяющих сократить объем данных. В этот процесс входит проведение компрессии с потерями (lossy) и без (lossless), что будет достигнуто при:

- разработке алгоритма временного предсказания с учётом особенностей произношения русского языка;
- создании алгоритма построения частотно-временной маски для использования перцептивных свойств слуха при сжатии.

Математический и методологический набор средств подготовки и сжатия отдельных свойств сигнала даёт возможность для синтеза структурно-функциональной схемы кодера, который будет обладать адаптивностью к требуемым условиям передачи речевой информации в реальном времени. По итогам проектирования поставлена задача по определению средств для экспериментальной оценки степени сжатия данных.

Приведённый перечень математического и методического инструментария позволит создать многофункциональный аппарат для системного анализа свойств восприятия преобразованной в линии связи речи слухом человека. А также на их базе реализовать семейство кодеров по заранее заданным условиям качества связи.

## 1.4 Выводы по первой главе

Основной задачей реализации удалённого общения между абонентами является построение линии связи, в состав которой входят средства подготовки и оптимизации потока данных. В условиях постоянно растущего трафика и, как следствие, уплотнения каналов связи неотъемлемой частью оптимизации потока является его сжатие. Для речевых данных реализуются две техники – с допустимыми потерями и без них. С учётом особенностей произношения и восприятия звуков, при неоднородности внешних и внутренних шумовых факторов разработка средства анализа свойств речевых преобразований на базе адаптивного кодера является актуальной задачей.

Параметрический анализ методов кодирования и алгоритмов сжатия звука выявил критичные для качества воспроизведения речи аспекты:

- аналоговое сжатие на стадии квантования сигнала позволит повысить эффективность использования динамического диапазона и является начальным этапом работы кодера. В качестве ключевых параметров определены функции битового распределения, разрядность квантования и частота дискретизации;

- установлена взаимосвязь элементов адаптивного дифференцированного взаимодействия, включающая в себя алгоритмы предсказания, в совокупности осуществляющего сжатие временного потока без потери качества воспроизведения, которая показала возможность увеличения степени компрессии за счёт повышения скорости адаптации и количества аналитических блоков;

- психоакустические ограничения слуха человека позволяют добиться большей эффективности сжатия путём установления неслышимого порога для звуков. Здесь выделены четыре критерия для перцептивного сжатия. Это модели временного и частотного маскирования, характеристика абсолютного порога слуха и параметры критичных диапазонов слуха.

Учитывая перечисленные результаты первичных исследований, была сформулирована расширенная задача структурно-параметрической реализации устройства для анализа эффективности сжатия и воспроизведения. Приведены определения для понятий адаптации, эффективности и маскирования при проведении компрессии.

## 2 Математическое и методическое приспособление средств компрессии

Первым этапом стыковки трёх совершенно разнородных предметных областей – психоакустики, лингвистики и цифровой обработки сигналов – представлен выбор и приспособление дискретных математических инструментов во временной и частотной области для процессов воспроизведения звуков и их интерпретации слухом человека [93].

### 2.1 Адаптация ортогонального преобразования время-частота

Ортогональные преобразования – это мощное спектральное средство, показывающее распределение широколиственного сигнала в амплитудно- и фазочастотной области. В результате обзора из целого ряда методов [3,87] выбраны два алгоритма: быстрое преобразование Фурье с прореживанием по времени и дискретное косинусное преобразование.

Преобразование дискретного вектора одномерного массива, содержащего звуковые отсчёты, сводится к его представлению в виде суммы ортогональных базисных сэмплов. В простом приближении это тригонометрические функции, взятые с одним периодом и с определённой частотой. Мы используем базовые функции косинуса в силу их чётности и свойства уплотнения энергии звука, о которых пойдёт речь в следующем подпункте.

#### 2.1.1 Применение дискретного синусного преобразования

В обобщенном виде ортогональные функции  $X(t)$  и  $Y(t)$  в некотором диапазоне изменения аргумента  $t_2 - t_1$  удовлетворяют условию:

$$\int_{t_1}^{t_2} X(t) \times Y(t) dt = 0. \quad (2.1)$$

Определим функцию  $X(t)$  как временное представление звукового потока. После проведения процедуры дискретизации во времени на отрезке  $t_2 - t_1$  получим вектор цифровых отсчётов  $X = [x_0, x_1, x_2 \dots x_{2N-1}]$ , где  $2N$  - количество отсчётов в сэмпле, а аналоговое значение  $\Delta x = x_2 - x_1$  - есть период дискретизации и  $1/\Delta t$  - её частота. По теореме Котельникова [44,117] максимальный неискажённый тон быть воспроизведён с частотой  $1/2\Delta t$ .

Положим, что  $Y(t)$  - это ортогональная  $X(t)$  функция и является её спектральным частотным представлением. Тогда вектор  $Y = [y_0, y_1, y_2 \dots y_{(N-1)}]$  будет дискретным спектром сигнала  $X(t)$  с максимальной неискажённой частотой  $y_{(N-1)} = 1/(2(x_n - x_{n-1}))$ , где  $n \in (0 \dots 2N-1)$ . Заменяем представление отсчётов вектора  $Y$  на тождественные аналоговые частоты для  $X$  и получим:

$$Y_x = \left[ 1/(2(x_{2N-1} - x_{2(N-1)-(2N-2)})), 1/(2(x_{2N-1} - x_{2(N-1)-(2N-3)})), 1/(2(x_{2N-1} - x_{2(N-1)-n})), \dots, 1/(2(x_{2N-1} - x_{2(N-1)-1})) \right]$$

Для преобразования дискретных временных отсчётов  $X_x$  в дискретные частотные  $Y_x$  применим классическое дискретное преобразование Фурье (ДПФ):

$$Y_x = \sum_{n=0}^{2N-1} X_x \cdot e^{-i \frac{2\pi n \cdot x}{(2N-1)}} \quad (2.2)$$

Преобразуем тригонометрически выражение 2.2 по формуле Эйлера, чтобы выделить частотные составляющие тонов и полутонов:

$$Y_x = \sum_{n=0}^{2N-1} X_x \cdot e^{-i \frac{2\pi n \cdot x}{2N}} = \sum_{n=0}^{2N-1} X_x \cdot \left[ \cos\left(\frac{2\pi n \cdot x}{2N}\right) - i \sin\left(\frac{2\pi n \cdot x}{2N}\right) \right] \quad (2.3)$$

Учитывая то, что мы используем только те частоты, которые ниже частоты Найквиста, преобразуем формулу 2.3 в вещественное ДПФ [119]:

$$Y_x = \sum_{n=0}^{2N-1} X_x \cdot \cos\left(\frac{2\pi n \cdot x}{2N}\right) + \sum_{n=0}^{2N-1} X_x \cdot \sin\left(\frac{2\pi n \cdot x}{2N}\right), \quad (2.4)$$

В 2.4 сумма содержащая косинусы называется прямым косинусным преобразованием, вторая сумма – прямое синусное преобразование. Остановимся на первом в силу его свойств «уплотнения энергии» и некоррелированности коэффициентов [72]. Дискретное косинусное преобразование (ДКП) будет эквивалентно вещественному при условии выполнения масштабирования. Функции синуса и косинуса эквивалентны между собой при значении угла

$45^\circ$ :  $\left| \sin\left(\frac{\pi}{4} + \frac{\pi n}{2}\right) \right| = \left| \cos\left(\frac{\pi}{4} + \frac{\pi n}{2}\right) \right| = \frac{\sqrt{2}}{2}$ . Базовый вектор косинуса при нулевой частоте будет

равен постоянной составляющей звука, поэтому выведем его за знак суммы и будем соблюдать масштаб:

$$Y_x = \frac{X_0}{\sqrt{2N}} + \sqrt{\frac{2}{2N}} \cdot \sum_{n=1}^{2N-1} X_x \cdot \cos\left(\frac{2\pi(x+1) \cdot n}{4N}\right) \quad (2.5)$$

В итоге получим прямую и обратную формулу адаптивного ортогонального преобразования для узкополосного сигнала с ограниченным кадром:

$$b(k) = \sqrt{\frac{1}{N}} \cdot \sum_{n=1}^{2N-1} x(n) \cdot \cos\left(\frac{2\pi(k+1) \cdot (n+1)}{4N}\right),$$

$$x(n) = \sqrt{\frac{2}{N}} \cdot \sum_{k=1}^{N-1} b(k) \cdot \cos\left(\frac{2\pi(k+1) \cdot (n+1)}{4N}\right) \quad (2.6)$$

где  $k \in (1 \dots N)$ .

Вектор  $b(k)$  содержит спектр узкополосного сигнала, подготовленный для психоакустического анализа. Учитывая особенности распределения тонов внутри одного критичного диапазона, их количество будет варьироваться относительно числа тонов и полутонов. Следует учитывать, что размер входного временного вектора  $b(n)$  должен содержать не меньше  $n = k \cdot 2 + 1$  отсчётов, чтобы избежать эффекта алиасинга. Результат применения 2.6 наглядно демонстрирует график на рисунке 2.1.

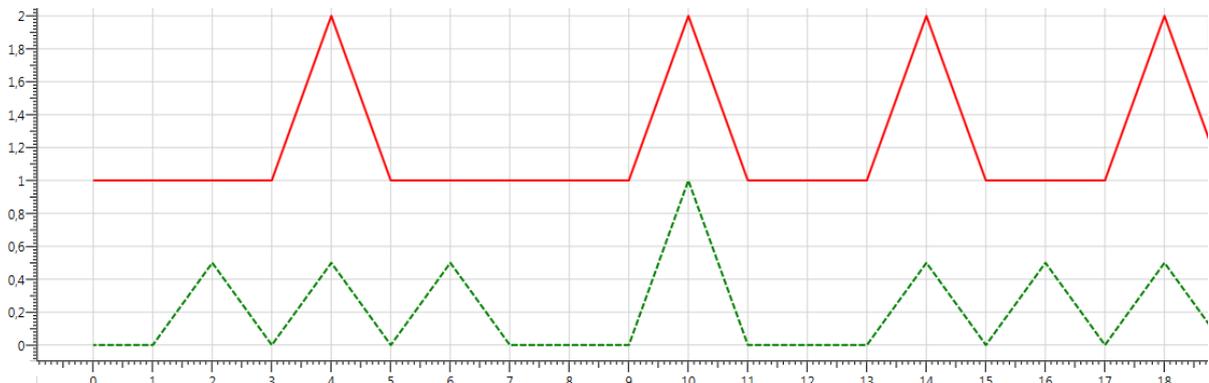


Рисунок 2.1 – график входного (-) и выходного (--) сигнала косинусного преобразования

Как видно из графика, восстановленный сигнал содержит искажения, проявляющиеся на третьем, шестом отсчётах и зеркально им – на четырнадцатом и восемнадцатом. Данный эффект вносит высокочастотные искажения в речевой сигнал, проявляющийся в виде «железных шумов». Модифицируем синусное весовое окно MPEG-H [100] к преобразованию:

$$h(n) = \sin^4 \left[ \frac{\pi \cdot \sin(\pi / 2) \cdot (n + 0.5)}{2N} \right] \quad (2.7)$$

В данном случае восстановленный сигнал будет иметь меньшее число наложений спектра:

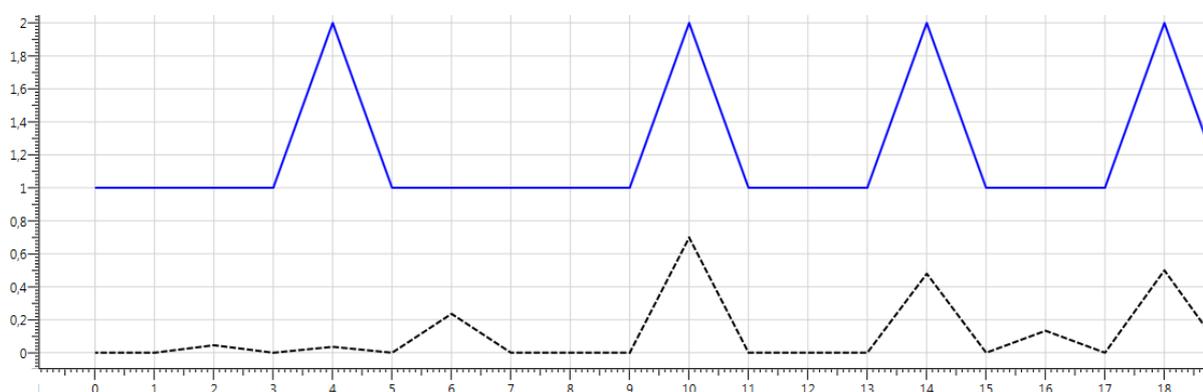


Рисунок 2.2 – график входного (-) и выходного (--) сигнала косинусного преобразования с применением синусного окна

Применение оконной функции даёт положительный эффект во второй половине восстановленного сигнала, однако имеет искажения в первой половине. Преобразим входной сигнал зеркально относительно нулевого отсчёта по формуле 2.8, тем самым увеличив сэмпл в два раза, и проведём преобразования 2.6 и 2.7:

$$x'(j) = \begin{cases} x(N - n - 1), & \text{if } j < N \\ x(n), & \text{otherwise} \end{cases}, \quad (2.8)$$

где  $j \in (1..2N)$  - количество полученных зеркальных отсчётов.

На рисунке 2.3 на среднем графике проиллюстрирован результат зеркального преобразования, при котором первая половина отсчётов симметрична второй половине относительно центрального отсчёта.

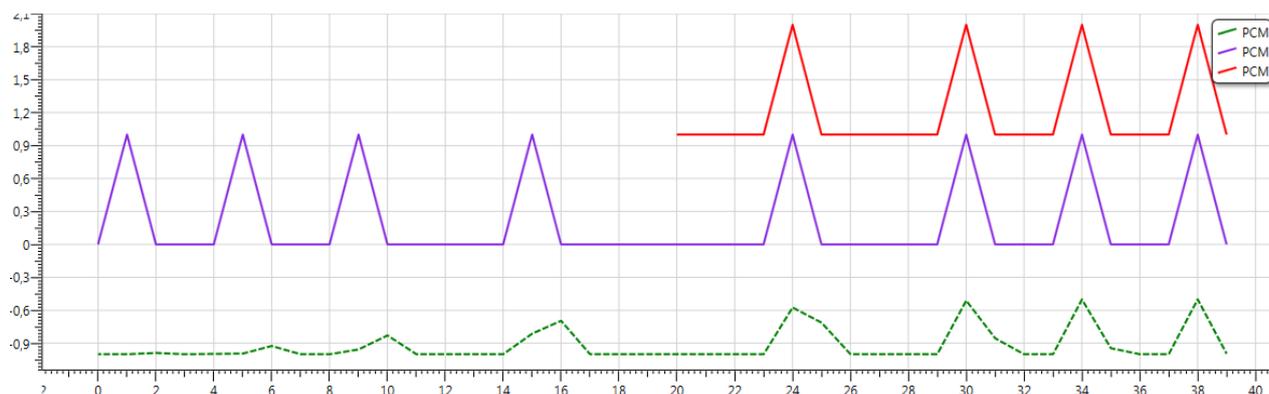


Рисунок 2.3 – график входного (—), зеркального (—) и выходного (—) сигнала косинусного преобразования с применением синусного окна

Тем самым мы добились приемлемого результата во второй половине восстановленного сигнала. Стоит отметить, что процедура восстановления сигнала в декодере должна будет включать в себя наложение сэмплов сигналов через каждую половину окна, или, как показано на рисунке 2.3, накладка произойдет с каждого двадцатого отсчета.

### 2.1.2 Быстрое преобразование Фурье с прореживанием по времени

В тех случаях, когда размер входного массива превышает третий порядок, имеет смысл применять алгоритмы быстрых ортогональных преобразований. Они основаны на тригонометрических упрощениях функций двойного угла и поэтому могут быть эффективны, только если количество временных отсчетов  $2N$  удовлетворяет условию  $2N = 2^s$ , где  $s \in \mathbb{N}$  [25,58]. Быстрое преобразование Фурье (БПФ) позволяет сократить число операций в  $\log_{10}(2N)$  раз по сравнению с преобразованиями, показанными выше, благодаря чему БПФ получил широкое применение в кодерах и стандартах [30,59]. В нашем исследовании необходимо использовать представление длинного временного массива для получения спектра всего диапазона речи. Учитывая, что его верхняя частота более чем в десять раз больше нижней, для его единовременного преобразования целесообразно использовать быстрый алгоритм, приспособленный к выбранным условиям. Количество операций уменьшается за счёт перестроения пар временных элементов по чётности и нечётности.

Проанализировав два алгоритма БПФ (с прореживанием по времени и по частоте) для структурной адаптации был выбран вариант с прореживанием по времени. Оба алгоритма имеют одинаковую эффективность, но их главным отличием является своевременность

ортогональных преобразований, а именно операций перемножения и суммирования комплексных чисел. Функционал алгоритма с прореживанием по частоте предполагает сначала проведение ДПФ от значений, кратных двум, вплоть до элементарного ДПФ-II, а затем перестановку (прореживание) полученных частотных отсчетов до необходимого последовательного порядка. Представим графически адаптированный вариант перестановки на рисунке 2.1

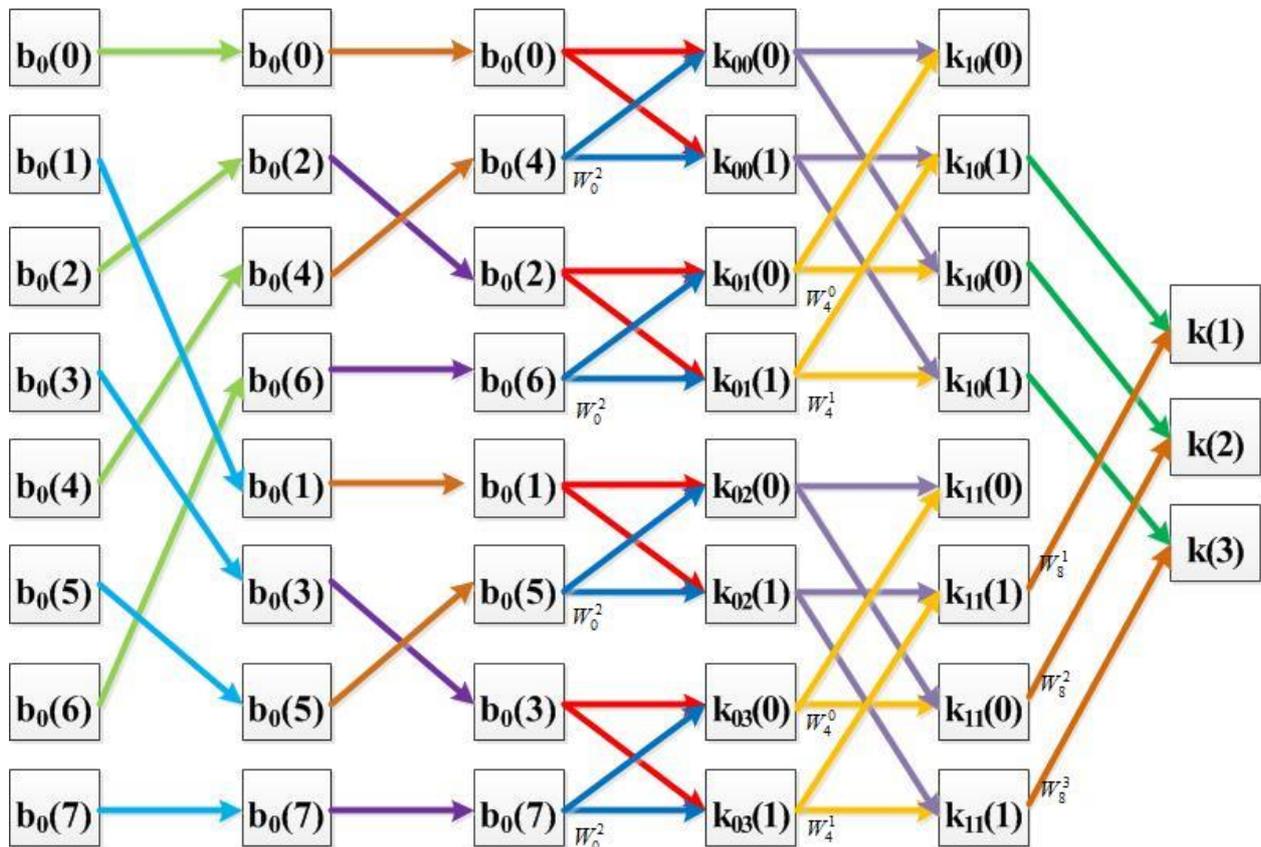


Рисунок 2.4 – Граф адаптированного алгоритма БПФ с прореживанием по времени

Следует помнить, что в рамках исследования нас интересуют действительные элементы спектра, удовлетворяющее теореме Котельникова, а значит их число в два раза меньше, чем количество временных элементов. Алгоритм БПФ с прореживанием по времени сначала предполагает прореживание элементов, а затем ортогональные преобразования ДПФ-(II, IV, VIII...). На выходе каждого преобразования получают частотные элементы, по количеству равные входным временным, что избыточно в нашем случае. Имеет смысл сократить выходные элементы, а значит операции по их вычислению, вдвое. И не будем забывать про нулевой частотный элемент, который показывает постоянную составляющую сигнала ( $f = 0$ ), его также следует исключить из вычисления.

Обратимся к последовательности действий при реализации представленного ортогонального преобразования:

- 1) производим три стадии прореживания временных отсчётов по принципу двоично-инверсионной перестановки, как это показано на графе 2.1;
- 2) вычисляем простые ДПФ-II от четырёх пар отсчётов по формулам:

$$k_{00}(0) = b_0(0) + W_2^0 \cdot b_0(4),$$

$$k_{00}(1) = b_0(0) - W_2^0 \cdot b_0(4),$$

$$k_{01}(0) = b_0(2) + W_2^0 \cdot b_0(6),$$

$$k_{01}(1) = b_0(2) - W_2^0 \cdot b_0(6),$$

$$k_{02}(0) = b_0(1) + W_2^0 \cdot b_0(5),$$

$$k_{02}(1) = b_0(1) - W_2^0 \cdot b_0(5),$$

$$k_{03}(0) = b_0(3) + W_2^0 \cdot b_0(7),$$

$$k_{03}(1) = b_0(3) - W_2^0 \cdot b_0(7),$$

где  $W_2^0$  - поворачивающая функция для двухточечного ДПФ = 1.

- 3) вычисляем перекрестные ДПФ-IV от двух четырёхточечных отсчётов по формулам:

$$k_{10}(0) = k_{00}(0) + W_4^0 \cdot k_{01}(0),$$

$$k_{10}(1) = k_{00}(1) + W_4^1 \cdot k_{01}(1),$$

$$k_{10}(2) = k_{00}(0) - W_4^0 \cdot k_{01}(0),$$

$$k_{10}(3) = k_{00}(1) - W_4^1 \cdot k_{01}(1),$$

$$k_{11}(0) = k_{02}(0) + W_4^0 \cdot k_{03}(0),$$

$$k_{11}(1) = k_{02}(1) + W_4^1 \cdot k_{03}(1),$$

$$k_{11}(2) = k_{02}(0) - W_4^0 \cdot k_{03}(0),$$

$$k_{11}(3) = k_{02}(1) - W_4^1 \cdot k_{03}(1),$$

где  $W_4^0 = 1$  и  $W_4^1 = -i$ .

- 4) вычисляем укороченный список конечных выходных отсчётов:

$$k(1) = k_{10}(1) + W_8^1 \cdot k_{11}(1);$$

$$k(2) = k_{10}(2) + W_8^2 \cdot k_{11}(2);$$

$$k(3) = k_{10}(3) + W_8^3 \cdot k_{11}(3),$$

где  $W_8^1 = e^{(-j\pi/4)}$ ,  $W_8^2 = -j$ ,  $W_8^3 = e^{(-j3\pi/4)}$ .

Для общего случая адаптированного БПФ выражение примет вид:

$$k(i) = k_0(i) + W_N^k \cdot k_1(i), \quad i = 1 \dots \frac{N}{4} - 1;$$

$$k(i + \frac{N}{4}) = k_0(i) - W_N^k \cdot k_1(i), \quad i = 1 \dots \frac{N}{4} - 1;$$

$$W_N^k = e^{-j\frac{2\pi k}{N}}$$

Произведём оценку эффективности полученного результата. Количество операций умножения равно  $\prod_{\text{АБПФ}} = 2N \cdot (\log_2(2N) - 2) - N - 1$ , где  $N - 1$  - это количество невоображённых операций вычисления мнимых и нулевого частотных отсчётов. Число операций суммирования  $\sum_{\text{АБПФ}} = 2N \cdot \log_2(2N) - N - 1$ .

Отношение количества операций нашего преобразования и ДПФ:

$$\prod \frac{\text{АБПФ}}{\text{ДПФ}} = \frac{2N \cdot (\log_2(2N) - 2) - N - 1}{4N^2} \quad (2.9)$$

$$\sum \frac{\text{АБПФ}}{\text{ДПФ}} = \frac{2N \cdot \log_2(2N) - N - 1}{4N^2} \quad (2.10)$$

Из 2.9 и 2.10 следует, что эффективность адаптированного преобразования будет расти прямо пропорционально увеличению количества преобразуемых отсчётов.

## 2.2 Адаптация методик потокового сжатия без потерь информации

В развитии методов полилогического анализа текстовой информации [26,91,93] здесь предложено использовать конструкции криптографической обработки для компрессии частотно-временных элементов, определяющих энергетические составляющие звукового потока.

Принцип компактной упаковки текстовых данных заключается в анализе контейнера (окна или буфера) определённого размера на предмет нахождения в нём повторных элементов, которыми являются символьные конструкции переменной длины. Различаются методы компрессии между собой, в основном, способами кодирования самих символов и их наборов (алфавитов), в состав последних дополнительно включают служебные символы, содержащие параметры обнаруженных повторов. Функциональным конечным этапом является построение специфических конструкций - кодовых деревьев, составленных по лексикографическому принципу[71,98,103].

Далее будут приведены методики адаптированных дескрипторов для внедрения в процессы ортогонального кодирования речи [83-86,88].

### 2.2.1 Словарное сжатие без потерь в кодеке

Для структуризации упорядочения сжатой информации внутри структуры длинного архивного файла или потока необходимо выделять служебные байты и непосредственно блок (в некоторых источниках - словарь). Каждый архиватор добавляет в начало и конец сжатого файла собственные заголовки и окончания. В случае, если размер сжатого файла больше, чем допустимый словарь, блоки разделяются. Так, например, максимальный размер блока в ZIP (Deflate) [77,118] равен 32 Кбайт. В общем виде структуру архива представим так:

заголовок	блок 1 (32Кб)	Разделитель	блок 2 (32Кб)	...	окончание
-----------	------------------	-------------	------------------	-----	-----------

В заголовке обычно указывают: версию архиватора, название файла или директории, количество блоков, тип разделителя, размер исходных и сжатых данных; в окончании присутствует контрольная сумма и т.д. Пример синтеза служебной информации и блока приведён ниже:

00000000	50	4b	03	04	14	00	00	00	08	00	30	16	87	44	33	28	PK.....0.#D3 (
00000010	34	e0	1a	00	00	00	23	00	00	00	07	00	00	00	31	32	4a....#.....12
00000020	33	2e	74	78	74	7b	f5	ee	cd	fb	17	af	ff	7f	7a	f7	3.txt{хоны.їя zч
00000030	41	e1	23	84	fd	e2	c3	bb	47	0f	5e	2b	20	24	00	50	АБ#,,эвГ»G.^+ \$.Е
00000040	4b	01	02	1f	00	14	00	00	00	08	00	30	16	87	44	33	К.....0.#D3
00000050	28	34	e0	1a	00	00	00	23	00	00	00	07	00	24	00	00	(4a....#.....\$. .
00000060	00	00	00	00	00	20	00	00	00	00	00	00	00	31	32	33	.....123
00000070	2e	74	78	74	0a	00	20	00	00	00	00	00	01	00	18	00	txt.. ..
00000080	1e	39	d8	79	ea	51	cf	01	e9	da	59	6e	ea	51	cf	01	9ШукQP.ЙЪYнкQP.
00000090	e9	da	59	6e	ea	51	cf	01	50	4b	05	06	00	00	00	00	ЙЪYнкQP. PK.....
000000a0	01	00	01	00	59	00	00	00	3f	00	00	00	00	00	..	..	...Y...?.....

Рисунок 2.5 – Структура архива ZIP версии 6.3. Красным показан сжатый блок, остальное – служебная информация.

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000007	6c	00	45	05	11	06	1c	03	10	09	01	00	5f	29	2c	00	l.Е..._. ....),..
00000010	00	06	a8	04	89	01	03	05	d2	d0	9d	42	d8	d3	21	a6	..Ё.%....ТРКВШУ!!
00000020	55	50	2c	09	5d	5d	55	d4	04	0e	8d	39	60	9c	4b	9a	UP,.)]УФ..К9`њКЪ
00000030	d8	14	0f	5c	98	56	6c	32	67	52	09	9d	a8	06	86	9e	Ш..\ V12gR.кЁ.†h
00000040	82	b9	04	85	69	5e	41	88	bc	4f	74	cb	95	6f	40	b1	,№...i^АЕјОтЛ.о@±
00000050	eb	00	cd	b7	63	d7	26	04	c5	bd	75	30	04	ea	89	e9	л.Н.сЧ&.ESu0.к%Й
00000060	e0	1d	62	c1	63	9c	ce	65	64	68	7f	29	55	2b	3d	00	a.бЕсьOedh) U+=.
00000070	45	01	11	06	1c	04	10	09	01	00	30	3d	2c	00	00	00	Е.....0=,...
00000080	18	04	82	9d	18	06	50	28	46	b7	66	75	ad	b5	8d	07	..,к..Р(F·fu-µК.
00000090	ba	c5	77	b4	b9	d0	04	c8	a1	06	d1	da	62	f3	be	5e	eЕwгТР.ИЎ.СЪbys^
000000a0	dc	4e	15	a5	61	a3	c7	1b	59	c6	cd	75	11	53	00	45	ЪН.ГаЈЭ.УЖHu.S.Е
000000b0	01	11	06	1c	05	10	09	01	00	46	40	79	2c	00	01	01	.....F@y,...
000000c0	10	04	97	a1	e8	02	29	9e	12	83	72	a1	64	b9	a6	73	..-Ўи.)h.fгrЎd№!s
000000d0	c1	cc	ac	3a	e6	28	04	db	11	02	d5	a8	a6	39	fd	10	ЕМ-:ж(.Ы..ХЁ!9э.
000000e0	91	9a	fb	3c	f6	79	6c	b2	60	45	ef	04	cf	7d	64	f9	`льк<цyLI`Еп.П)дщ
000000f0	29	4f	1a	fe	9e	2b	9b	da	8a	29	b7	53	ad	3b	39	4c	)O.юh+>Ъь) ·S-;9L
00000100	24	74	a8	00	45	05	11	06	1c	06	10	09	01	00	80	9a	\$tЁ.Е.....Ъь
00000110	40	51	2c	02	14	00	c3	04	5a	a9	05	95	84	0a	38	c3	@Q,...Г.Z@.·„.8Г
00000120	c3	8c	bb	31	b5	2f	2b	d4	61	a7	ec	85	04	ca	5d	38	ГЪ>лµ/+Фa\$м...К]8
00000130	4e	84	60	3f	a4	a4	7c	e5	c8	50	5f	01	3a	a3	0b	2f	N,,`?°° еИР_.:J./
00000140	8c	04	07	49	95	01	78	9d	68	1e	25	ff	21	27	be	18	Ъ..Г.хКh.Ѕя!'s.
00000150	c1	12	3b	fc	84	29	04	c3	4d	1e	9e	40	80	b3	ec	da	Б.;ь,,).ГМ.h@ЪимЪ
00000160	33	4a	ba	1a	b5	a8	ef	19	51	19	d7	04	12	31	29	80	ЗJe.µЁп.Q.Ч..1)Ъ
00000170	e8	5c	db	f3	89	3e	ca	5d	01	70	45	07	66	21	3d	15	и\Ыу%>К].pЕ.f!=-.
00000180	04	e6	f9	18	35	be	3b	e3	3a	40	b1	0c	f3	5b	61	51	.жщ.5s;r:@±.y[aQ
00000190	38	11	58	ba	f7	04	de	31	93	71	50	86	d6	54	10	d2	8.Хеч.Ю1`qP†ЦТ.Т
000001a0	65	14	d5	85	c0	4d	2d	aa	26	93	d6	9c	3e	00	45	05	е.Х...АМ-Є&`Цъ>.Е.
000001b0	11	06	1c	07	10	09	01	00	31	40	65	2c	01	00	01	00	.....1@e,....
000001c0	04	c2	a1	01	0d	fc	20	53	01	e5	c1	94	53	f0	01	cf	.ВЎ..ьь S.eВ"Sp.П
000001d0	da	a2	c6	51	01	04	51	e1	20	98	99	f4	ef	45	16	f5	ЪўЖQ..Qб`тфпЕ.х
000001e0	5b	a8	ea	dc	80	01	42	29	b8	48	84	6c	6c	00	45	05	[ЁкЪ.В)єН,,ll.Е.
000001f0	11	06	1c	..	..	..	..	..	..	..	..	..	..	..	..	..	....

Рисунок 2.6 – Сжатый поток с выделенным заголовком

Исходя из представленной структуры, целесообразно обосновать способ выделения заголовков, окончаний и разделителей. Заголовок уникален для каждого нового архива и размер

его вариативен, то же и у окончания. Но разделитель, как правило, остается постоянным и часто содержится в заголовке.

Главным отличием синтеза поточных данных является неопределённость точки входа в архив, т.е. для нахождения служебных байтов необходимо накопить количество информации из потока равное большему, чем один блок, размеру. В данном случае поиск блока возможен по разделителю. Часто в качестве него выступает заголовок канального уровня (см. рисунок 2.3).

Таким образом, методика синтеза разделителя блоков будет следующей:

1. провести поиск повторяющихся конструкций байтов (от 3х и более подряд);
2. выстроить найденные конструкции в порядке убывания размера и указать частоту повторений в файле;
3. проанализировать количество информации, заключенной между первым найденным повтором. В случае, если оно меньше или равно 32 Кб, сделать вывод о присутствии разделителя. В обратном случае – провести анализ для меньшей байтовой конструкции.

Результатом синтеза является как первая конструкция, удовлетворяющая условию максимального размера блока, так и последующие.

Для окончательного определения разделителя необходимо синтезировать первый байт сжатого блока.

### 2.2.2 Методика сжатия кадров по статическому словарю

Рассмотрим методику компрессии статического Deflate (первый и второй бит первого байта блока – 01). В данном случае набор символов в словаре строго определен и не меняется в различных блоках. Размер и диапазон кодов для набора литералов алфавита и длин повторов расположены следующим образом [42]:

Литерал	Размер в битах	Диапазон кодов
0 – 143	8	00110000 - 10111111
144 - 255	9	110010000 - 111111111
256 - 279	7	0000000 - 0010111
280 - 285	8	11000000 - 11000111

Так код 0000000 принадлежит диапазону 0000000 – 0010111, является символом 256 и означает конец блока.

Все коды литералов находятся друг за другом, коды символов следуют в обратном порядке. Для синтеза сжатых литералов в диапазоне 00110000-10111111 необходимо из исходного кода Хаффмана вычесть 00110000, в диапазоне 110010000-111111111 из исходного

кода вычесть 100000000. Коды для дистанций переводятся в десятичное счисление и вычисляются соответственно литералам. Окончанием блока всегда является литера 256.

Отметим, что коды Хаффмана расположены последовательно ортогонально и лексикографически, т.е. установление каждого последующего бита, начиная со старшего, однозначно определяет размер и диапазон литералов. Например, считывание первых двух битов – 00 означает возможную длину кода 7 или 8, третий однозначно определяет длину и диапазон. Данное свойство присуще всей логике нахождения кодов словарей Хаффмана (в т.ч. и динамического).

Методику восстановления исходных байтов можно представить следующим образом:

1. Чтение первых двух бит;

2. Если 00, то это диапазоны литералов 0-143 или 256-279. Если 11, то это диапазоны 144-255 или 280-285. Переход к пункту 3;

а) Если следующий бит 0, то это семибитный код диапазона 256-279. Это диапазон длин повторов. Читать 7 бит. Найти значение по формуле  $256 + \{7 \text{ бит}\}$ . Определить количество дополнительных бит. Считать дополнительные биты. Определить длину повтора. Если это литерал 256, закончить обработку блока - выход. Если не 256, то считать 5 бит для дистанции в обратном порядке. Определить количество дополнительных бит. Читать дополнительные биты дистанций. Вычислить дистанцию. Отсчитать дистанцию от последнего найденного байта влево, копировать число байтов, равное длине повтора, вставить содержимое в конец, вслед за последним найденным;

б) Если следующий бит 1, то это восьмибитный код диапазона 0-143. Определить исходный байт по формуле  $\{8 \text{ бит}\} - 00110000$ ;

3. Если 11, то это диапазоны литералов 144-255 или 280-285. Читать следующий бит;

а) Если бит равен 0, то это девятибитный диапазон 110010000-110011111. Читать 9 бит. Вычислить исходный байт по формуле  $\{9 \text{ бит}\} - 100000000$ ;

б) Если бит равен 1, то это восьмибитный диапазон 11000000 – 11000111. Это литерал длины повтора. Читать 8 бит. Определить количество дополнительных бит по таблице 2. Читать дополнительные биты. Определить длину. Читать 5 бит для дистанции. Определить дополнительное количество бит для дистанции по таблице 3. Читать дополнительные биты. Вычислить дистанцию. Отсчитать дистанцию от последнего найденного байта влево, копировать число байтов, равное длине повтора, вставить содержимое в конец, вслед за последним найденным.

4. Повторить пункт 1.

Таким образом, алгоритм компрессора является цикличным, условием выхода из которого становится код 0000000.

### 2.2.3 Методика сжатия кадров по динамическому словарю

Синтезирование динамически сжатого Deflate (первый и второй бит первого байта – 10) [40] происходит посредством построения трёх деревьев Хаффмана, каждое из которых определяет последующий код символов и, главное, код окончания. В данном случае, он в каждом блоке различен и, не определив деревья, невозможно определить конец блока.

Особенностью динамического словаря является не только уникальность кодов символов для каждого блока, но и двойное кодирование по Хаффману. Сжатию подвержена также и размерность кода для литералов длин/символов и дистанций. С целью более компактного размещения кодов словаря неиспользуемые литералы и дистанции не входят в систему деревьев. Их точное количество определяют первые байты блока.

№ байта	1								2								3									
№ бита	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
алфавиты	HLIT					тип		б	HCLEN				HDIST												HCLEN	
№ бита	4	3	2	1	0				2	1	0	4	3	2	1	0										3

Методика нахождения размера алфавитов следующая:

1. Читать пять бит в прямом порядке. Вычислить размер алфавита длин/символов по формуле  $HLIT = 257 + \{5 \text{ бит}\}$ ;
2. Читать пять бит в прямом порядке. Вычислить размер алфавита дистанций по формуле  $HDIST = 9 + \{5 \text{ бит}\}$ ;
3. Читать 4 бита в прямом порядке. Вычислить размер алфавита длин кодов Хаффмана  $HCLEN = 4 + \{4 \text{ бита}\}$ .

Первыми необходимо определить размеры кодов для дерева длин кодов Хаффмана. Последовательность нахождения следующая:

1. Размеры кодов считать подряд по три бита (количество трёхбитных чисел определено в HCLEN, максимум 19), переводить в декартовое счисление и последовательно вносить в порядке, показанном в таблице. Максимальное значение длины кода равно 7;

Символ алфавита длин	16	17	18	0	8	7	9	6	10	5	11	4	12	3	13	2	14	1	15
Длина кода символа	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}	{3}

2. Выполнить поиск значений кодов Хаффмана в лексикографическом порядке: значению младшего кодового значения соответствует высший литерал алфавита.

Методика нахождения кодов по известным длинам происходит следующим образом:

1. Найти самый длинный код и задать все его разряды единицами. Например, если длина кода 5, то – 11111;
2. Назначить все последующие коды данной длины, декрементировав каждый относительно предыдущего (количество кодов взять из таблицы);
3. При переходе на более короткую длину кода произвести побитный сдвиг вправо на один бит;
4. Перейти в пункт 2, если не все коды длин найдены, иначе – перейти в пункт 5.

Пример нахождения 4 кодов дерева по длине 5:

Длина кода 5	1	1	1	1	1	Декрементировать
Длина кода 5	1	1	1	1	0	Декрементировать
Длина кода 5	1	1	1	0	1	Декрементировать
Длина кода 5	1	1	1	0	0	Сдвинуть на бит вправо и декрементировать
Длина кода 4		1	1	0	1	Декрементировать...

5. Перестроить таблицу 1 в лексикографическом порядке: младшей длине кода соответствует меньший код, но не нарушая соответствия столбцов таблицы 1 и длины кода. Это и есть первое дерево.

Нахождение кодов размера большого алфавита

Символ алфавита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Длина кода символа	DEC																		
Код Хаффмана	BIN																		

Интерпретация 19ти значений длин кодов:

0 - 15: Представляют длины кодов 0 – 15 для большого алфавита литералов.

16: Копировать предыдущую длину кода от 3 до 6 раз. Следующие 2 бита в прямом порядке указывают длину повторения.

17: Повторить длину кода 0 от 3 до 10 раз. Следующие 3 бита в прямом порядке указывают длину повторения.

18: Повторить длину кода 0 от 11 до 138 раз. Следующие 7 бит в прямом порядке указывают длину повторения.

Теперь, зная коды Хаффмана для длин большого алфавита литералов, находим размеры их кодов:

№ литерала длины/символа	Размер кода из табл.п.6	Код Хаффмана по дереву
0	DEC	{BIN}
1	DEC	{BIN}
2	DEC	{BIN}
3	DEC	{BIN}
...	...	...
...	...	...
285	DEC	{BIN}

1. Читать первый бит и последующие (согласно дереву Хаффмана каждый последующий бит точно определяет сам код и его размер), определить код длины. Если символ оказывается 16, 17 или 18, читать дополнительные биты. Установить соответствующие нулевые (если 17 или 18) и повторные значения длин кода (если 16) для большого алфавита литералов;
2. Читать следующие биты и устанавливать соответствия литералов длинам их кодов до последнего, определенного НЛИТ. Максимальное значение 285;
3. Найти все коды Хаффмана для НЛИТ, зная их длины, по методике, описанной выше;
4. Установить лексикографическое соответствие между литералами и кодами Хаффмана, найденными в пункте 3. Получим дерево для основного алфавита литералов;
5. Читать следующий бит, найти длины кодов алфавита дистанций. Максимальное значение HDIST 29;
6. Установить лексикографическое соответствие для кодов Хаффмана по их длинам. Получим третье дерево алфавита дистанций:

Символ алфавита дистанций	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Длина кода символа	DEC														
Код Хаффмана	BIN														
Символ алфавита дистанций	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Длина кода символа	DEC														
Код Хаффмана	BIN														

Первое дерево для длин кодов Хаффмана, состоящее из 4-19 значений, больше не понадобится. Далее последовательно читаем в обратном порядке коды исходных символов их повторений:

- 7 Читать в обратном порядке биты и искать соответствие в таблице литералов/длин.
- а) Если найден литерал (0-255), записать в результат;
  - б) Если длина повтора (257-285), читать дополнительные биты в прямом порядке. Определить длину повтора. Далее читать в обратном порядке биты и искать соответствие в алфавите дистанций, читать дополнительные биты для дистанций. Определить дистанцию. Отсчитать влево от последнего найденного байта дистанцию, копировать количество символов, соответствующее длине повтора, и вставить в конец, вслед за последним найденным;
  - в) Если найден код Хаффмана для символа 256 – это означает конец блока.

#### 2.2.4 Методика сжатия кадра по принципу скользящего окна

Алгоритм архивирования Stack LZS (скользящее окно) представлен подробно в описании [41]. Сжатие LZS и декомпрессия используют алгоритм типа LZ77. Последние два килобайта несжатых данных являются словарём с изменяющимся скользящим окном. Компрессор LZS ищет соответствия между данными, которые будут сжаты, и предшествующими 2 Кб данных. Если он находит совпадение, то оно кодируется сочетанием *Смещение/Длина* для поиска совпадения в словаре. Если совпадений не найдено, следующий байт данных кодируется как «литерал» – исходный байт. Сжатый поток данных заканчивается *маркером окончания*.

Данные кодируются в поток переменной длины. Литерал кодируется как бит «0», за которым следуют 8 битов *исходного байта*.

*Смещение/Длина* кодируются как бит «1», далее идет код смещения, а затем код длины. В конце устанавливается исключительный код – *Маркер окончания*.

*Смещение* может иметь минимальное значение 1, максимальное – 2047. Значение 1 указывает на самый последний байт в буфере истории, текущий байт данных есть начало повтора. Смещение кодируется следующим образом:

- если смещение меньше 128: бит «1» предшествует 7-битному значению смещения.
- если смещение больше или равно 128: бит «0» предшествует значению смещения с 11-битной длиной.

*Маркер окончания* кодируется как 9-битный маркер 110000000. После конечного маркера, по мере необходимости, нулевые биты в количестве от 0 до 7 дополняют поток до границы байта.

Транскрипция сжатого потока данных:

$\langle \text{Сжатый поток} \rangle := [\langle \text{Сжатая строка} \rangle] \langle \text{Маркер окончания} \rangle$

$\langle \text{Сжатая строка} \rangle := 0 \langle \text{Исходный байт} \rangle \mid 1 \langle \text{Сжатый байт} \rangle$

<Исходный байт>:= <b><b><b><b><b><b><b><b> (8-bit byte)

<Сжатый байт>:= <Смещение> <Длина>

<Смещение >:= 1 <b><b><b><b><b><b><b> | (7-bit смещение)  
0 <b><b><b><b><b><b><b><b><b><b><b>(11-bit)

<Маркер окончания >:= 110000000

<b>:= 1 | 0

<Длина>:=

00	= 2	1111 0110	= 14
01	= 3	1111 0111	= 15
10	= 4	1111 1000	= 16
1100	= 5	1111 1001	= 17
1101	= 6	1111 1010	= 18
1110	= 7	1111 1011	= 19
1111 0000	= 8	1111 1100	= 20
1111 0001	= 9	1111 1101	= 21
1111 0010	= 10	1111 1110	= 22
1111 0011	= 11	1111 1111 0000	= 23
1111 0100	= 12	1111 1111 0001	= 24
1111 0101	= 13	...	

Методика компрессора и дескриптора для LZS:

- 1) Искать четырёхбайтный заголовок: первый байт - 108dec, второй – 00hex, третий и четвертый – вариативные, для определения заголовка IPComp.
- 2) Искать слово (WORD) по каждому из вариантов:

№	1 байт	2 байт
1	11000000	00000000
2	X1100000	00000000
3	XX110000	00000000
4	XXX11000	00000000
5	XXXX1100	00000000
6	XXXXX110	00000000
7	XXXXXXX11	00000000
8	XXXXXXXX1	10000000

где X – любой возможный бит, для поиска Маркера окончания.

- 3) Читать следующий бит текущего байта сжатого потока:
- а) Если бит равен 0, то это несжатая строка, вставить в результат следующие восемь бит, это *Исходный байт*.
  - б) Если бит равен 1, провести проверку на код 110000000 (если да, то завершить процесс – это *Маркер окончания*), в обратном случае – это сжатая строка, читать *Смещение/Длина*.
- 4) Читать первый бит *Смещения*:
- а) Если первый бит смещения равен 0, то размер смещения 7 бит. Читать 7 бит, перевести результат в декартовое счисление.
  - б) Если первый бит смещения равен 1, то размер смещения 11 бит. Читать 11 бит, перевести результат в декартовое счисление.
- 5) Читать первый бит *Длины*. Определить полный код длины лексикографически (см. транскрипцию). Перевести результат в декартовое счисление.
- 6) Отсчитать *Смещение* побайтно от конца словаря, копировать число найденных ранее байт согласно *Длине*. Вставить в конец. Перейти к пункту 3.

### 2.3 Выводы по второй главе

В ходе проведённой адаптации были сформированы предварительные для данного исследования результаты цифрового представления и обработки звука в сочетании с особенностями его восприятия человеком для проведения процедур сжатия. В промежуточном итоге вынесены следующие положения:

- исследование и адаптация известных ортогональных методик преобразования временной функции в частоту станет инструментом анализа речевого сигнала для выявления избыточности в условиях нестационарности его восприятия. Здесь алгоритм БПФ будет использован как средство выявления громких тонов критичных диапазонов, а использование МДКП ускорит процесс вычисления, выполняя требования по фазовому сдвигу;

- адаптированные для звука методики сжатия без потерь информации увеличат коэффициент компрессии при неизменности качества, однако внесут задержку в процедуру передачи данных, что может повлиять на комфортность общения. Производными от оригинальных параметров и объектов сжатия выступают значения частот и амплитуд каждого тона отдельного барка.

### 3 Синтез математической модели для функциональной и алгоритмической реализации устройства динамического кодера

В соответствии с поставленной в работе научно-технической задачей, в текущей главе будут разработаны процедуры цифрового воздействия на звуковой сигнал, которые в совокупности составляют математическую модель кодера, реализующую процедуры подготовки и преобразований речевого потока данных.

#### 3.1 Синтез псевдо-квадратурного зеркального банка фильтров в перцептивном кодировании

Первым этапом подготовки входного потока для ортогонального преобразования явилась цифровая фильтрация. Для определения структуры блока полифазной фильтрации проведены исследования по типу «анализ через синтез» [9]. Количество параллельных полосовых фильтров было окончательно установлено путём добавления последующих с анализом получившейся структуры на устойчивость к фазовым сдвигам. Одновременно велась работа по определению вариативных параметров каждого фильтра в банке. В результате была синтезирована модель полифазного фильтра [15,36,37], состоящего из нерекурсивных фильтров, с уникально организованными весовыми оконными функциями.

В силу того, что различные реализации цифровых фильтров подразумевают множество неидентичных фазочастотных характеристик (ФЧХ), и, как следствие, частотно-временных искажений, был синтезирован банк с однородными ФЧХ.

Вариативными параметрами для Банка фильтров были определены сглаживающая функция со своими коэффициентами, весовой коэффициент и коэффициент передачи фильтра.

Разностное уравнение импульсной характеристики ЦФ для проверки расчёта коэффициентов:

$$y(k) = b_0x(k) + \dots + b_mx(k-m) - a_1y(k-1) - \dots - a_ny(k-n),$$

где  $a_i$  и  $b_j$  – вещественные коэффициенты;

$k$  – отчет на выходе фильтра  $y(k)$ .

Сворачивающий вектор дискретных значений фильтра:

$$h_n = \begin{cases} \frac{\psi \times p_n \times (\sin(\omega_2 \times n) - \sin(\omega_1 \times n))}{\pi \times n}, & \text{если } n \geq 1 \\ \omega_0, & \text{в остальных случаях} \end{cases},$$

где  $h_n$  – n-ый множитель коэффициента  $b$  фильтра;

$\Psi$  – коэффициент передачи фильтра;

$\omega_0, \omega_1, \omega_2$  – круговые частоты постоянной составляющей, верхней и нижней частоты;

$P_n$  – весовая функция, противодействующая явлению Гиббса:

$$P_n = 1 - \frac{n}{\theta \times N},$$

где  $n$  – количество входных временных отсчетов;

$N$  – сглаживающий коэффициент;

$\theta$  – весовой коэффициент.

Синтез всего банка полифазного фильтра в работе представлен методически. Результатом построения данного блока явился ряд временных отсчётов импульсных характеристик для всех четырнадцати полосовых фильтров. Способ расчёта коэффициентов для голоса автора приведён ниже. Нижняя и верхняя частоты с учётом масштаба определяются по формуле (3.1) [16]:

$$f_1 = f_n / 8000 \text{ Гц}, \quad f_2 = f_b / 8000 \text{ Гц}. \quad (3.1)$$

Для расчёта необходимо знать циклическую частоту, которая определяется для нижней и верхней частоты формулой (2.4) [16]:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_2 := 2 \times \pi \times f_2 \quad (3.2)$$

Нулевой отсчёт импульсной характеристики фильтра определяется выражением  $\omega_0 = (\omega_2 - \omega_1) / \pi$  [16].

Для каждого из фильтров методом «анализ через синтез» выбраны следующие коэффициенты:  $\Psi$  – коэффициент передачи фильтра,  $N$  – сглаживающий коэффициент,  $\Theta$  – весовой коэффициент.

Проведём расчёт коэффициентов цифрового нерекурсивного фильтра для критических интервалов частот представленных ниже [48]:

Диапазон (Барк) 4, центральная частота 350 Гц, ширина 300 – 400 Гц,

$f_1 = 300/8000$  Гц, нижняя частота с учётом масштаба,  $f_2 := 400/8000$  Гц, верхняя частота с учётом масштаба, тогда определим циклические частоты для  $f_1$  и  $f_2$  согласно формуле (2.4):

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.38 = 0.236 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.05 = 0.314 \text{ Рад/с}.$$

Нулевой отсчёт импульсной характеристики фильтра определяется согласно выражению  $\omega_0 = (\omega_2 - \omega_1)/\pi$ , а частота дискретизации определяется формулой:  $f = \frac{1}{\Delta t}$ ,  $\Delta t = 125 \times 10^{-6}$  с., тогда  $f = 8 \times 10^3$  Гц;  $n=0..31$  – диапазон отсчётов. Для данного фильтра индивидуально выбираем следующие коэффициенты, которые равны соответственно:  $\Psi = 38$ ,  $N = 61$ ,  $\Theta = 0,6$ .

Весовая функция, противодействующая явлению Гиббса [12]  $p_n$  определяется выражением  $p_n = 1 - n/\theta \times N$ . Импульсная характеристика и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.1.

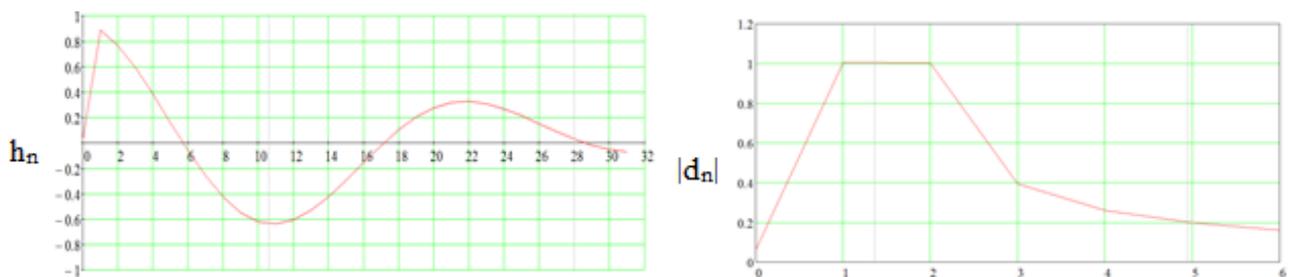


Рисунок 3.1 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведённой импульсной характеристики, представленной на рисунке 3.1, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.1.

Таблица 3.1 – Значения коэффициентов КИХ фильтра для диапазона частот 300 – 400 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.025	0.889	0.765	0.591	0.383	0.159	-0.062
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	-0.062	-0.063	-0.429	-0.551	0.622	-0.64	0.608
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	-0.532	-0.424	-0.294	-0.155	-0.019	0.104	0.203
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	0.274	0.315	0.324	0.307	0.267	0.212	0.149
№ п/п	28	29	30	31	-	-	-
Значение коэффициента	0.086	0.028	-0.018	-0.069	-	-	-

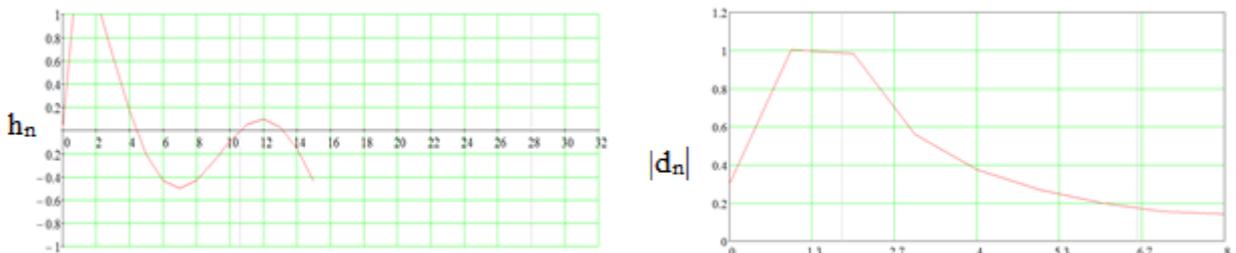
Диапазон (Барк) 5, центральная частота 450 Гц, ширина 400 – 510 Гц.  $f_1 = 400/8000$  Гц, нижняя частота с учётом масштаба;  $f_2 = 510/8000$  Гц, верхняя частота с учётом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.38 = 0.236 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.063 = 0.401 \text{ Рад/с}.$$

В данном случае диапазон отсчётов составляет  $n := 0 \dots 15$ . Для данного фильтра выбираем следующие коэффициенты, которые соответственно равны:  $\Psi = 68$ ,  $N = 23$ ,  $\Theta = 0,46$ .

Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 2.2.

Рисунок 3.2 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведённой импульсной характеристики, представленной на рисунке 3.2, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.2.

Таблица 3.2 – Значения коэффициентов КИХ фильтра для диапазона частот 400 – 510 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.028	1.586	1.144	0.639	0.163	-0.21	-0.434
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	-0.5	-0.429	-0.272	-0.09	-0.05	0.099	0.027
№ п/п	14	15	-	-	-	-	-
Значение коэффициента	-0.163	-0.439	-	-	-	-	-

Диапазон (Барк) 6, центральная частота 570 Гц, ширина 510 – 630 Гц.  $f_1 = 510/8000$  Гц, нижняя частота с учётом масштаба;  $f_2 = 630/8000$  Гц, верхняя частота с учётом масштаба.

Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.063 = 0.401 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.08 = 0.495 \text{ Рад/с}.$$

В данном случае количество отсчётов составляет  $n = 0 \dots 31$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 40$ ,  $N = 50$ ,  $\Theta = 0,49$ .

Импульсная характеристика и АЧХ (БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.3.

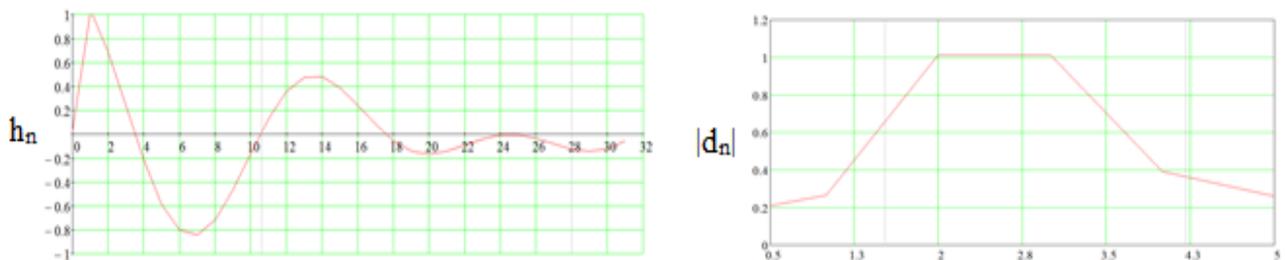


Рисунок 3.3 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведённой импульсной характеристики, представленной на рисунке 3.3, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.3.

Таблица 3.3 – Значения коэффициентов КИХ фильтра для диапазона частот 510 – 630 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0,03	1.037	0.688	0.237	-0.218	-0.586	-0.803
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	-0.842	-0.714	-0.465	-0.16	0.133	0.356	0.473
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	0.478	0.388	0.214	0.079	-0.057	-0.142	-0.169
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	-0.145	-0.093	-0.039	$-4.873 \times 10^{-3}$	$-3.747 \times 10^{-3}$	0.034	-0.082
№ п/п	28	29	30	31	-	-	-
Значение коэффициента	-0.126	-0.144	-0.122	-0.056	-	-	-

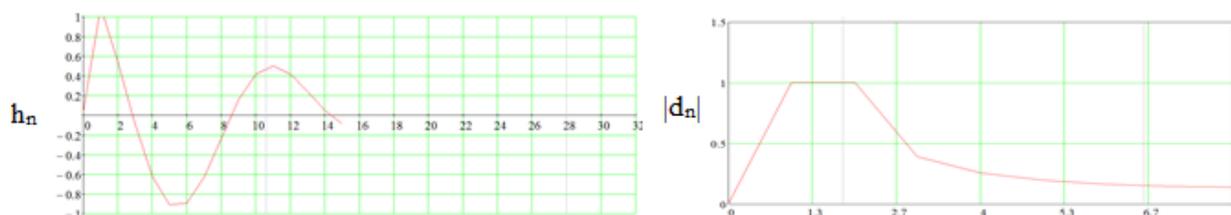
Диапазон (Барк) 7, центральная частота 700 Гц, ширина 630 – 770 Гц.  $f_1 = 630/8000$  Гц, нижняя частота с учётом масштаба;  $f_2 = 770/8000$  Гц, верхняя частота с учётом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.08 = 0.495 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.096 = 0.605 \text{ Рад/с}.$$

В данном случае диапазон отсчётов составляет  $n := 0 \dots 31$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 39$ ,  $N = 40$ ,  $\Theta = 0,46$ .

Импульсная характеристика  $h_n$  и АЧХ (БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.4.

Рисунок 3.4 – Импульсная характеристика  $h_n$  и АЧХ (БПФ)  $|d_n|$

Для приведенной импульсной характеристики, представленной на рисунке 3.4, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.4.

Таблица 3.4 – Значения коэффициентов КИХ фильтра для диапазона частот 630 – 770 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.035	1.1	0.551	-0.086	-0.623	-0.907	-0.892
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	-0.627	-0.231	0.156	0.419	0.502	0.419	0.239
№ п/п	14	15	-	-	-	-	-
Значение коэффициента	0.046	-0.086	-	-	-	-	-

Диапазон (Барк) 8, центральная частота 840 Гц, ширина 770 – 920 Гц.  $f_1 = 770/8000$  Гц, нижняя частота с учётом масштаба,  $f_2 = 920/8000$  Гц, верхняя частота с учётом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.096 = 0.605 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.115 = 0.723 \text{ Рад/с}.$$

В данном случае диапазон отсчётов составляет  $n := 0 \dots 31$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 32$ ,  $N = 64$ ,  $\Theta = 0,39$ .

Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.5.

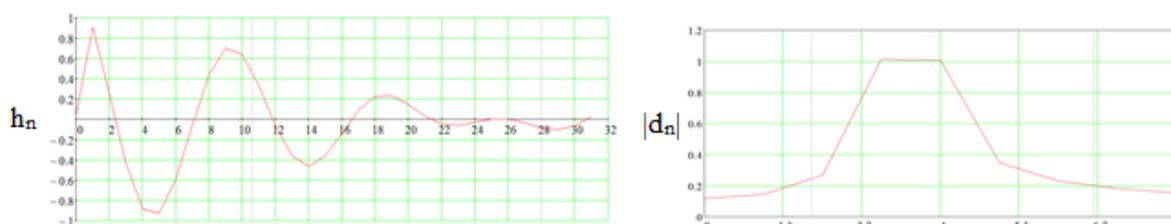


Рисунок 2.5 – Импульсная характеристика  $h_n$  и АЧХ (БПФ)  $|d_n|$

Для приведённой импульсной характеристики, представленной на рисунке 3.5, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.5.

Таблица 3.5 – Значения коэффициентов КИХ фильтра для диапазона частот 770 – 920 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.038	0.907	0.265	0.237	-0.882	-0.931	-0.596
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	-0.056	0.442	0.697	0.636	0.329	-0.063	-0.364
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	-0.465	-0.361	-0.136	0.091	0.224	0.23	0.142
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	0.029	-0.047	-0.061	-0.031	$8.245 \times 10^{-4}$	$7.686 \times 10^{-4}$	-0.037
№ п/п	28	29	30	31	-	-	-
Значение коэффициента	-0.085	-0.104	-0.066	0.023	-	-	-

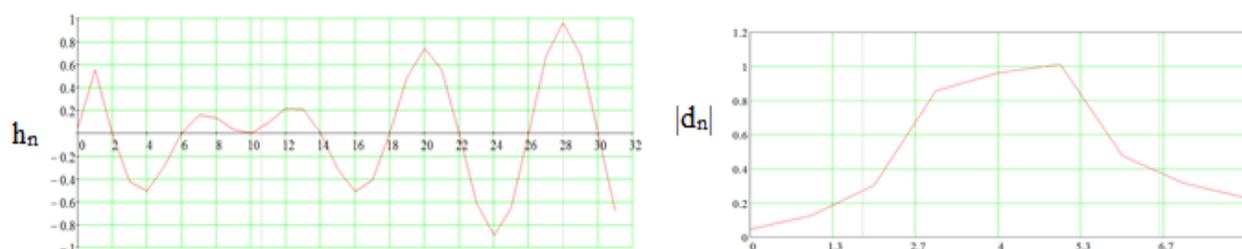
Диапазон (Барк) 9, центральная частота 1000 Гц, ширина 920 – 1080 Гц.  $f_1 = 920/8000$  Гц, нижняя частота с учетом масштаба;  $f_2 = 1080/8000$  Гц, верхняя частота с учетом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.115 = 0.723 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.135 = 0.848 \text{ Рад/с}.$$

В данном случае диапазон отсчётов составляет  $n := 0 \dots 31$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 22$ ,  $N = 22$ ,  $\Theta = 0,43$ .

Импульсная характеристика  $h_n$  и АЧХ (БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.6.

Рисунок 3.6 – Импульсная характеристика  $h_n$  и АЧХ (БПФ)  $|d_n|$ 

Для приведённой импульсной характеристики, представленной на рисунке 3.6, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.6.

Таблица 3.6 – Значения коэффициентов КИХ фильтра для диапазона частот 920 – 1080 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.04	0.556	0	-0.422	-0.503	-0.289	0
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	0.157	0.13	0.029	0	0.093	0.215	0.208
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	0	-0.313	-0.511	-0.407	0	0.489	0.742
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	0.557	0	-0.611	-0.895	-0.651	0	0.675
№ п/п	28	29	30	31	-	-	-
Значение коэффициента	0.963	0.683	0	-0.676	-	-	-

Диапазон (Барк) 10, центральная частота 1175 Гц, ширина 1080 – 1270 Гц.  $f := 1080/8000$  Гц, нижняя частота с учетом масштаба;  $f_2 = 1270/8000$  Гц, верхняя частота с учетом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.135 = 0.848 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.16 = 0.997 \text{ Рад/с}.$$

В данном случае диапазон отсчетов составляет  $n := 0 \dots 31$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 15$ ,  $N = 19$ ,  $\Theta = 0,44$ .

Импульсная характеристика  $h_n$  и АЧХ (БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.7.

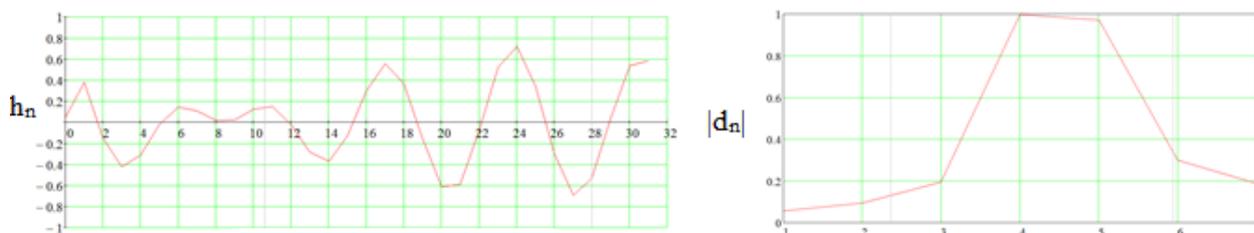


Рисунок 3.7 – Частотная характеристика фильтра (БПФ)

Для приведенной импульсной характеристики, представленной на рисунке 3.7, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.7.

Таблица 3.7 – Значения коэффициентов КИХ фильтра для диапазона частот 1080 – 1270 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.047	0.378	-0.147	-0.422	-0.312	-0.027	0.143
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	0.109	0.013	0.022	0.125	0.15	-0.021	-0.283
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	-0.373	-0.132	0.298	0.554	0.369	-0.16	-0.612
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	-0.593	-0.083	0.519	0.718	0.343	-0.303	-0.696
№ п/п	28	29	30	31	-	-	-
Значение коэффициента	-0.529	0.04	0.538	0.581	-	-	-

Диапазон (Барк) 11, центральная частота 1370 Гц, ширина 1270– 1480 Гц.  $f_1 = 1270/8000$  Гц, нижняя частота с учетом масштаба;  $f_2 = 1480/8000$  Гц, верхняя частота с учетом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле :

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.16 = 0.997 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.19 = 0.162 \text{ Рад/с}.$$

В данном случае диапазон отсчетов составляет  $n := 0 \dots 31$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 17$ ,  $N = 20$ ,  $\Theta = 0,44$ .

Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.8.

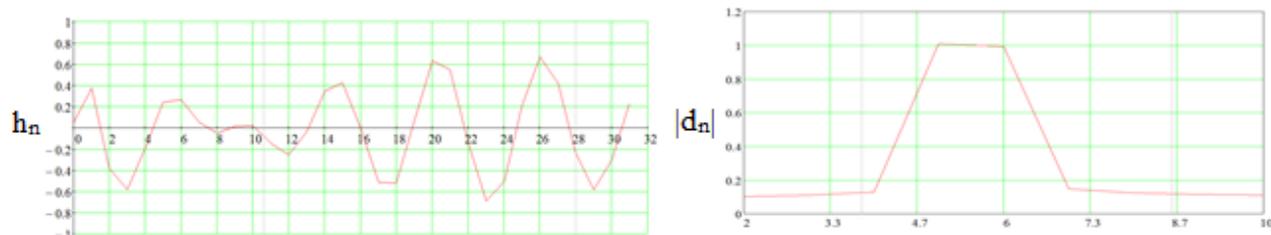


Рисунок 3.8 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведенной импульсной характеристики, представленной на рисунке 3.8, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.8.

Таблица 3.8 – Значения коэффициентов КИХ фильтра для диапазона частот 1270 – 1480 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.053	0.372	-0.381	-0.579	-0.183	0.238	0.267
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	0.05	-0.053	0.018	0.021	-0.15	-0.253	-0.034
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	0.347	0.424	0	-0.516	-0.521	-0.065	0.634
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	0.545	-0.14	-0.688	-0.505	0.204	0.67	0.417
№ п/п	28	29	30	31	-	-	-
Значение коэффициента	-0.239	-0.581	-0.299	0.229	-	-	-

Диапазон (Барк) 12, центральная частота 1600 Гц, ширина 1480– 1720 Гц.  $f_1 = 1480/8000$  Гц, нижняя частота с учетом масштаба;  $f_2 = 1720/8000$  Гц, верхняя частота с учетом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.19 = 1.162 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.22 = 1.353 \text{ Рад/с}.$$

В данном случае диапазон отсчетов составляет  $n := 0...31$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 15$ ,  $N = 22$ ,  $\Theta = 0,34$ .

Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.9.

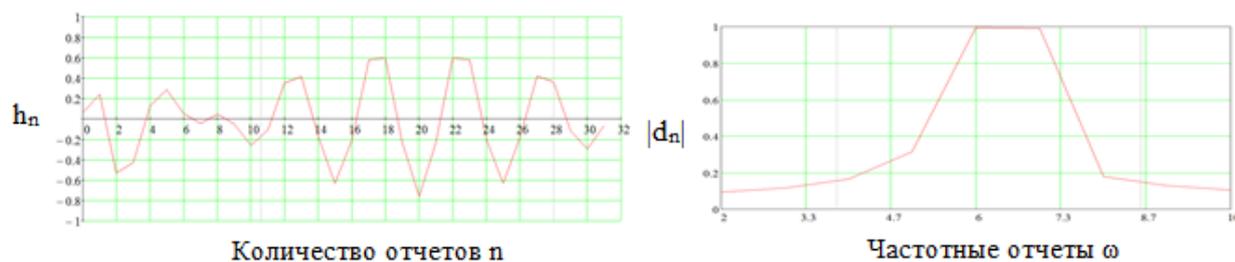


Рисунок 3.9 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведенной импульсной характеристики, представленной на рисунке 3.9, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.9.

Таблица 3.9 – Значения коэффициентов КИХ фильтра для диапазона частот 1480 – 1720 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.06	0.241	-0.53	-0.43	0.126	0.287	0.052
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	-0.043	0.046	-0.05	-0.26	-0.109	0.352	0.413
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	-0.178	-0.632	-0.21	0.578	0.599	-0.233	-0.76
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	-0.233	0.597	0.576	-0.209	-0.633	-0.179	0.42
№ п/п	28	29	30	31	-	-	-
Значение коэффициента	0.365	-0.116	-0.296	-0.065	-	-	-

Диапазон (Барк) 13, центральная частота 1850 Гц, ширина 1720 – 2000 Гц.  $f_1 = 1720/8000$  Гц, нижняя частота с учетом масштаба;  $f_2 = 2000/8000$  Гц, верхняя частота с учетом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.215 = 1.351 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.25 = 1.571 \text{ Рад/с}.$$

В данном случае диапазон отсчетов составляет  $n := 0 \dots 31$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 15$ ,  $N = 22$ ,  $\Theta = 0,34$ .

Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.10.

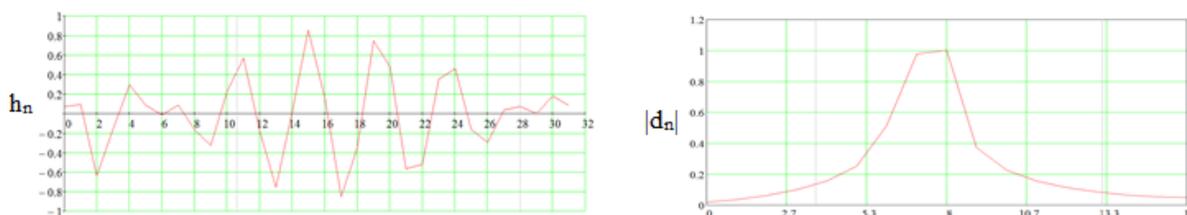


Рисунок 3.10 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведенной импульсной характеристики, представленной на рисунке 3.10, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.10.

Таблица 3.10 – Значения коэффициентов КИХ фильтра для диапазона частот 1720 – 2000 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.07	0.09	-0.639	-0.159	-0.297	0.089	-0.014
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	0.089	-0.169	-0.326	0.229	0.565	-0.172	-0.755
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	0.026	0.857	0.166	-0.851	-0.35	0.745	0.481
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	-0.565	-0.521	0.353	0.458	-0.161	-0.298	0.033
№ п/п	28	29	30	31	-	-	-
Значение коэффициента	0.071	$-2.55 \times 10^{-3}$	0.179	0.081	-	-	-

Диапазон (Барк) 14, центральная частота 2150 Гц, ширина 2000 – 2320 Гц.  $f_1 = 2000/8000$  Гц, нижняя частота с учетом масштаба;  $f_2 = 2320/8000$  Гц, верхняя частота с учетом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.25 = 1.571 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.29 = 1.822 \text{ Рад/с}.$$

В данном случае диапазон отсчётов составляет  $n := 0 \dots 31$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 2$ ,  $N = 20$ ,  $\Theta = 0,07$ .

Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.11.

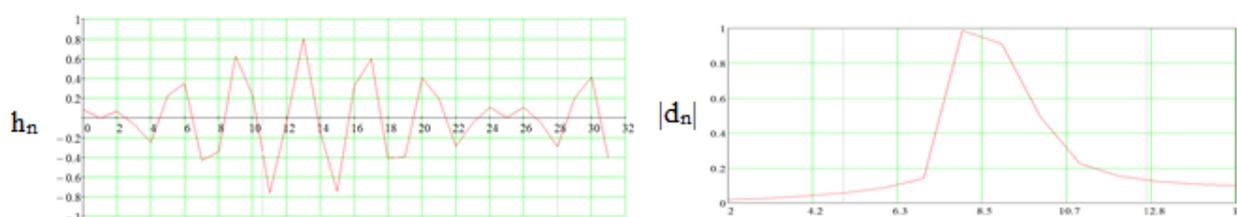


Рисунок 3.11 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведенной импульсной характеристики, представленной на рисунке 3.11, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.11.

Таблица 3.11 – Значения коэффициентов КИХ фильтра для диапазона частот 2000 – 2320 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.08	$-5.714 \times 10^{-3}$	0.066	-0.066	-0.25	0.266	0.348
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	-0.432	-0.339	0.629	0.23	-0.766	-0.05	0.808
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	-0.151	-0.746	0.32	0.595	-0.412	-0.395	0.402
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	0.197	-0.291	-0.053	0.106	0	0.107	-0.053
№ п/п	28	29	30	31	32	33	34
Значение коэффициента	-0.296	0.201	0.412	-0.407	-	-	-

Диапазон (Барк) 15, центральная частота 2500 Гц, ширина 2320 – 2700 Гц.  $f_1 = 2320/8000$  Гц, нижняя частота с учетом масштаба;  $f_2 = 2700/8000$  Гц, верхняя частота с учетом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.29 = 1.822 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.34 = 2.121 \text{ Рад/с}.$$

В данном случае диапазон отсчётов составляет  $n := 0 \dots 31$ . Для данного фильтра выбираем коэффициенты, равные соответственно:  $\Psi = 21$ ,  $N = 20$ ,  $\Theta = 0,5$ .

Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.12.

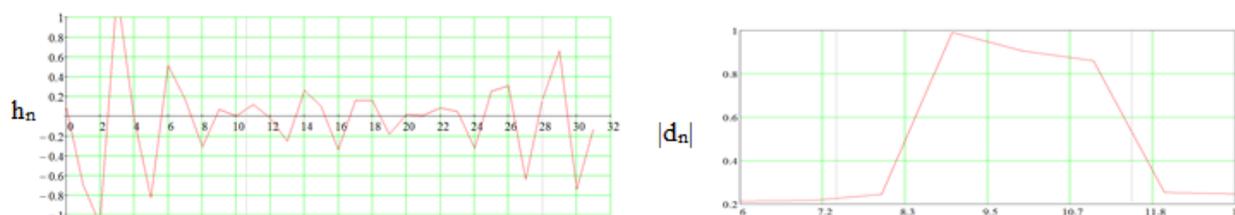


Рисунок 3.12 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведенной импульсной характеристики, представленной на рисунке 3.12, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.11.

Таблица 3.12 – Значения коэффициентов КИХ фильтра для диапазона частот 2320 – 2700 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.095	-0.698	-1.094	1.259	-0.035	-0.824	0.514
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	0.164	-0.31	0.065	0	0.116	-0.02	-0.253
№ п/п	14	15	16	17	18	19	20
Значение коэффициента	0.259	0.095	-0.34	0.157	0.157	-0.185	0.016
№ п/п	21	22	23	24	25	26	27
Значение коэффициента	$4.667 \times 10^{-3}$	0.084	-0.046	-0.326	0.248	0.304	-0.642
№ п/п	28	29	30	31	-	-	-
Значение коэффициента	0.161	0.661	-0.739	-0.134	-	-	-

Диапазон (Барк) 16, центральная частота 2900 Гц, ширина 2700 – 3150 Гц.  $f_1 = 2700/8000$  Гц, нижняя частота с учетом масштаба;  $f_2 = 3150/8000$  Гц, верхняя частота с учетом масштаба. Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.34 = 2.121 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.4 = 2.474 \text{ Рад/с}.$$

В данном случае диапазон отсчётов составляет  $n := 0 \dots 15$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 18$ ,  $N = 22$ ,  $\Theta = 0,51$ .

Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.13.

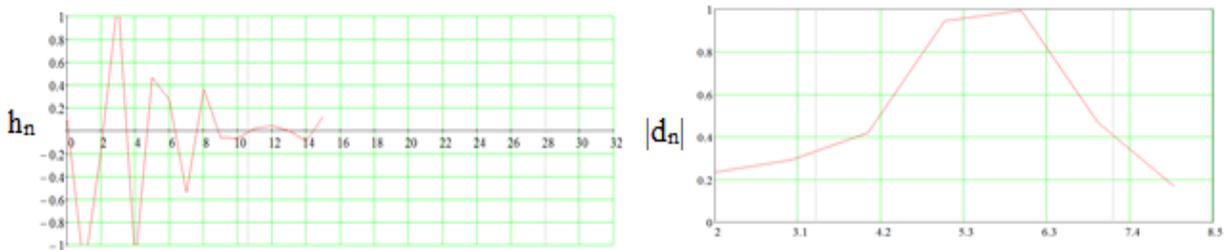


Рисунок 3.13 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведённой импульсной характеристики, представленной на рисунке 3.13, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.13.

Таблица 3.13 – Значения коэффициентов КИХ фильтра для диапазона частот 2700 – 3150 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.112	-1.219	-0.192	1.161	-1.164	0.463	0.268
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	-0.542	0.362	-0.064	-0.068	0.019	0.043	$-2.052 \times 10^{-3}$
№ п/п	14	15	-	-	-	-	-
Значение коэффициента	-0.092	0.121	-	-	-	-	-

Диапазон (Барк) 17, центральная частота 3400 Гц, ширина 3150 – 3700 Гц.  $f_1 := \frac{3150}{8000}$  Гц,

нижняя частота с учетом масштаба;  $f_2 := \frac{3700}{8000}$  Гц, верхняя частота с учетом масштаба.

Определим циклические частоты для  $f_1$  и  $f_2$  по формуле:

$$\omega_1 := 2 \times \pi \times f_1, \quad \omega_1 := 2 \times \pi \times 0.4 = 2.474 \text{ Рад/с},$$

$$\omega_2 := 2 \times \pi \times f_2, \quad \omega_2 := 2 \times \pi \times 0.5 = 2.906 \text{ Рад/с}.$$

В данном случае диапазон отсчётов составляет  $n := 0 \dots 15$ . Для данного фильтра выбираем следующие коэффициенты, равные соответственно:  $\Psi = 15$ ,  $N = 21$ ,  $\Theta = 0,51$ .

Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$  для данного фильтра приведена на рисунке 3.14.

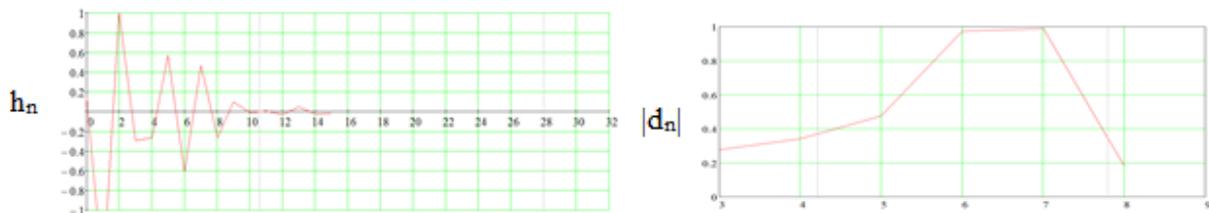


Рисунок 3.14 – Импульсная характеристика  $h_n$  и АЧХ(БПФ)  $|d_n|$

Для приведённой импульсной характеристики, представленной на рисунке 3.14, количество коэффициентов КИХ фильтра и их значения приведены в таблице 3.14.

Таблица 3.14 – Значения коэффициентов КИХ фильтра для диапазона частот 3150 – 3700 Гц.

№ п/п	0	1	2	3	4	5	6
Значение коэффициента	0.138	-1.669	1.006	-0.296	-0.266	0.57	-0.612
№ п/п	7	8	9	10	11	12	13
Значение коэффициента	0.472	-0.266	0.095	-0.01	$4.114 \times 10^{-3}$	-0.033	0.047
№ п/п	14	15	-	-	-	-	-
Значение коэффициента	-0.025	-0.022	-	-	-	-	-

На рисунке 3.15 показаны речевые сигналы до и после проведения фильтрации:

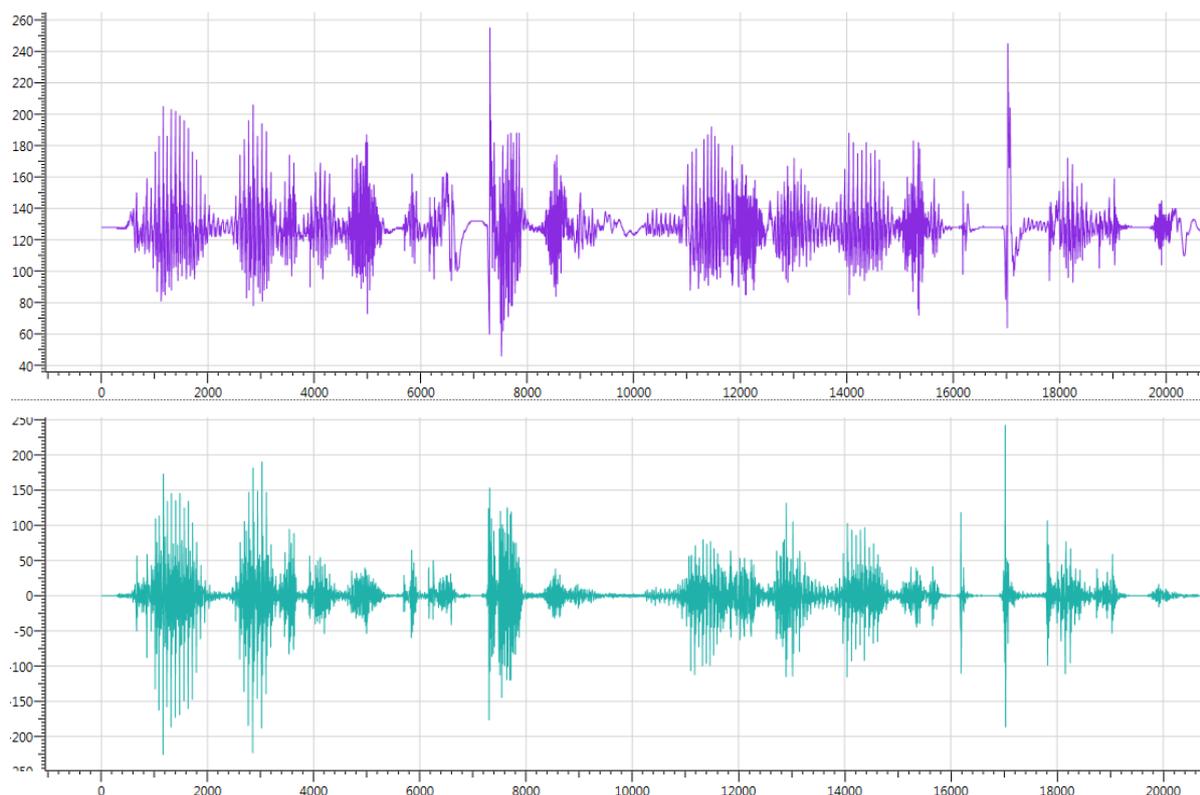


Рисунок 3.15 – Речевой сигнал до (верхний график) и после прохождения банка фильтров (нижний график)

Нижний график рисунка 3.15 содержит преобразованный биполярный сигнал, что является неотъемлемой частью подготовки входного унополярного сигнала к дискретным ортогональным преобразованиям.

Проверка каждого фильтра в синтезированном банке проводилась по методике вычисления минимальной дисперсии банка фильтров, используя неунифицированное ДПФ [13,95]. Если величина дисперсии остается в пределах входных параметров математической модели психоакустического анализа, то можно сделать вывод о приемлемости использования синтезированной структуры.

В данном разделе был проведён расчёт импульсной характеристики с помощью системы Mathcad, приведён способ, с помощью которого определено количество коэффициентов КИХ фильтра, необходимое для обеспечения заданной переходной полосы, а также определена оконная функция для обеспечения заданного подавления в полосе заграждения.

### 3.2 Обоснование и разработка алгоритма предиктора в ортогональном кодере

Русская речь отличается от остальных множеством динамических параметров, в т.ч. частотным диапазоном и временным слуховым восприятием. Особенно наглядно последнее объясняется фонетической транскрипцией русских слов.

Запись устной речи в полном соответствии с её звучанием не может быть осуществлена обычным орфографическим написанием. При орфографическом письме отсутствует полное соответствие между звуками и буквами, в графике отсутствуют знаки, необходимые для записи всех звуков устной речи. Так, например, редуцированные гласные в безударных слогах не произносимы, а значит, не слышимы [16,64].

Учитывая данную особенность русской речи, можно предположить возможность внедрения в стандартизированные речевые кодеры устройства, обеспечивающего более плотное сжатие без серьёзных осложнений разборчивости.

Ввиду акустических особенностей слуха человека в целом [11,26,49], в речевом кодировании (сжатии) получили широкое распространение вокодеры с адаптивным речевым предсказанием. Их главная отличительная особенность – автокорреляционный анализ ансамбля предыдущих дискретных отсчётов сигнала с текущим для предсказания последующих [90]. Результатом вычисления являются дифференциальные значения, которые содержат кодовую последовательность с меньшей разрядностью. Обобщённая структурная схема кодера с линейным предиктором изображена на рисунке 3.15:

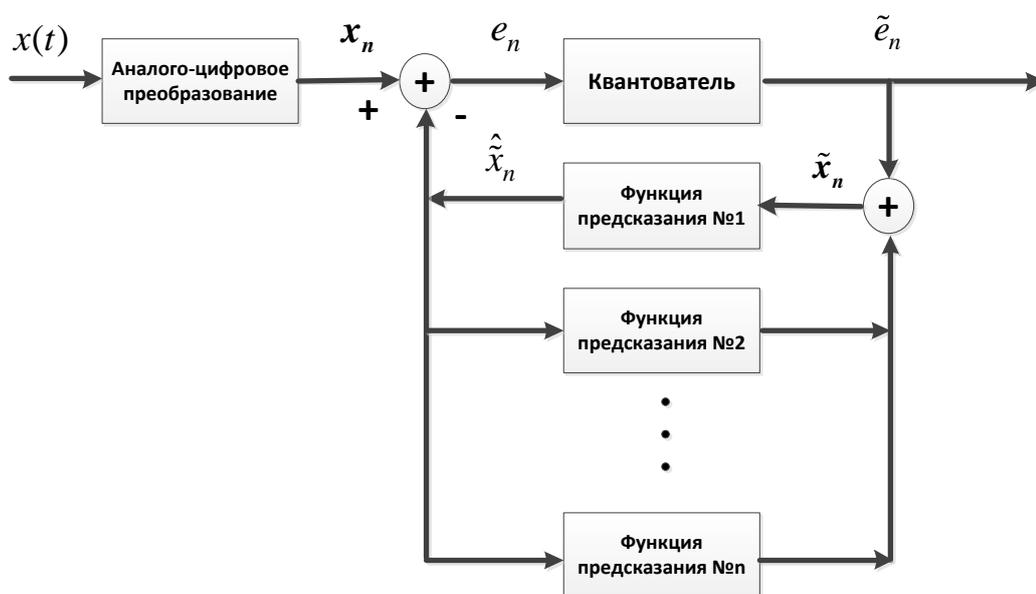


Рисунок 3.16 – Структура дифференциального кодера с обратной связью

В представленной схеме процесс предсказания реализован в цепи обратной связи, охватывающей устройство квантования (квантователь). Входной сигнал в формате ИКМ для предсказателя обозначен  $\tilde{x}_n$ . Он представляет собой сигнальный отсчёт  $x_n$ , дифференцированный в результате вычисления сигнала ошибки в квантователе. Математическое выражение для выходного дифференцированного сигнала, проиллюстрированного адаптивного кодера, в котором последующее входное значение сравнивается только с одним предыдущим, представлено в формуле [32]:

$$\hat{\tilde{x}}_n = \sum_{k=1}^K b_k \cdot \tilde{x}_{n-k} - \sum_{m=1}^M a_m \cdot \tilde{x}_{n-m}, \quad (3.3)$$

- $x(t)$  - аналоговый речевой сигнал;
- $e_n$  - входной сигнал квантователя;
- $\tilde{e}_n$  - выходной сигнал квантователя;
- $x_n$  - входной сигнальный отсчёт;
- $\tilde{x}_n$  - входной отсчёт предсказателя;
- $\hat{\tilde{x}}_n$  - выходной отсчёт предсказателя;
- $n$  - индекс временного отсчёта;
- $b$  - коэффициент предсказания;
- $a$  - коэффициент рекурсии;
- $k$  - индекс коэффициента предсказания;
- $m$  - индекс коэффициента рекурсии;
- $K$  и  $M$  - количество коэффициентов предсказания и рекурсии.

Показанная на схеме (рисунок 3.15) разность  $e_n = x_n - \hat{\tilde{x}}_n$  является входным сигналом квантователя, далее по схеме сигнал  $\tilde{e}_n$  обозначает его выход. Сжатие реализуется путём кодирования выходного сигнала ошибки предсказания  $\tilde{e}_n$  последовательностью двоичных символов с меньшей разрядностью квантования, что и является результатом работы всего кодера. Полученная ошибка  $\tilde{e}_n$  также суммируется с предсказанной величиной  $\hat{\tilde{x}}_n$ , с целью получения  $\tilde{x}_n$ .

Основной же идеей применения ортогонального звукового кодирования является использование неидеальности слуха человека в частотном диапазоне [23,50]. Основанием для построения подобных кодеров стала психоакустическая модель звукового восприятия, по структуре которой происходит построение маски в кодере для «неслышимой» информативной составляющей [74,126]. На рисунках 3.16 и 3.17 изображены результаты эксперимента по определению границ маскирования для частотной области реального речевого сигнала помехой с частотой 1 КГц.



Рисунок 3.17 – Частотная маска для реального речевого сигнала

- максимум для неискаженного сигнала;
- результат быстрого преобразования Фурье;
- граница частотного маскирования.

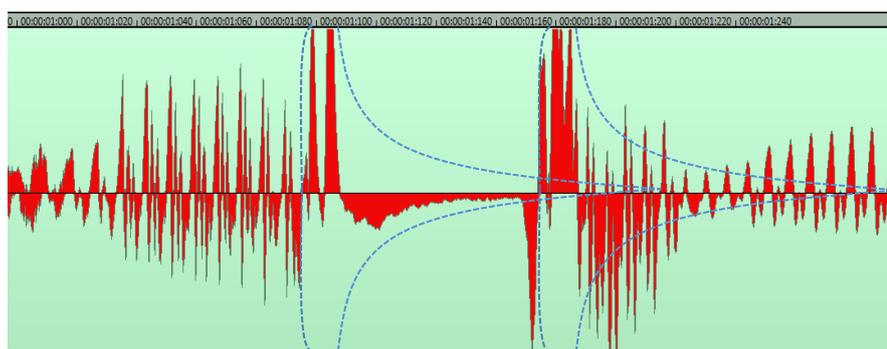


Рисунок 3.18 – Временная маска для реального речевого сигнала:

- дискретные временные значения в формате РСМ;
- граница временного маскирования.

Функция психоакустической модели, отображающая порог частотного маскирования (выделенная белым пунктиром на рисунке 3.18), определяется по формуле [32,34]:

$$SF_{об}(x) = 15,81 + 7,5 \times (x + 0,474) - 17,5 \times \sqrt{1 + (x + 0,474)^2} \quad (3.4)$$

Аппроксимированные пороговые значения для временной области, показанные на рисунке 3.17 синими пунктирами, были вычислены с помощью двойной округло-экспоненциальной функции Плака–Мура [16]. Пороговые значения для симметричных и асимметричных входных сигналов установлены одновременно сверху и снизу. Каждая сторона временного окна ( $W$ ) определяется по формуле 3.5 [19,14]:

$$W(t) = (1 - \omega) \left( 1 + \frac{2t}{T_p} \right) \exp\left( -\frac{2t}{T_p} \right) + \omega \left( 1 + \frac{2t}{T_s} \right) \exp\left( -\frac{2t}{T_s} \right) \quad (3.5)$$

Параметр  $T_p$  – постоянная времени, связанная с нарастающим к пику действующим значением амплитуды дискретного сигнала;  $T_s$  – постоянная времени, связанная с окончанием (спадом) маски; и  $\omega$  определяет точку перехода между пиковыми и конечными частями функции.

В обоих случаях для временного и частотного маскирования сжатие исходного сигнала происходит путём удаления дискрет, находящихся ниже порога маскирования.

В развитие известных техник ортогонального речевого кодирования, представленных группой MPEG [32,33,34,35], автором предложен вариант интеграции функции линейного предсказания в структуру модифицированного кодера МР3. По своей сути главной отличительной чертой ортогональных кодеров от линейных является преобразование «время-частота». Сжатие потока в MPEG достигается путём определения факторов маскирования в частотной области, удаления низких по амплитуде частотных коэффициентов и переквантования оставшихся кодом с меньшей длиной [102,106-108,130-132].

Основная идея более плотного сжатия заключается в преобразовании (переквантовании, удалении) частей временного потока перед ортогональным преобразованием за счет внедрения блока линейного предиктора в процесс полифазной линейной фильтрации. Модифицируемая область ортогонального кодера представлена на рисунке 3.19.



Рисунок 3.19 – Линейный предиктор в структуре ортогонального кодера

На вход кодера приходит сигнал  $X(t)$  в ИКМ (PCM) формате. Далее происходит разделение на 14 критических полос  $X_1(t) - X_{14}(t)$  при помощи полифазного КИХ фильтра. Каждый временной поток попадает в блок «Линейного предиктора», в котором осуществляется процедура наложения временной маски. Результатом является параллельный выходной поток  $E_1(t) - E_{14}(t)$ , который затем подвергается частотному кодированию посредством модифицированного косинусного преобразования (MDCT) [14,75].

Реальный ортогональный кодер работает в цифровых устройствах кодирования/квантования, работа которых труднообъяснима математическими выражениями в декартовой системе счисления. Ввиду трудности представления принципа работы предиктора в виде математической модели, как вариант способа отображения его работы предложен функциональный алгоритм (рисунок 3.20).

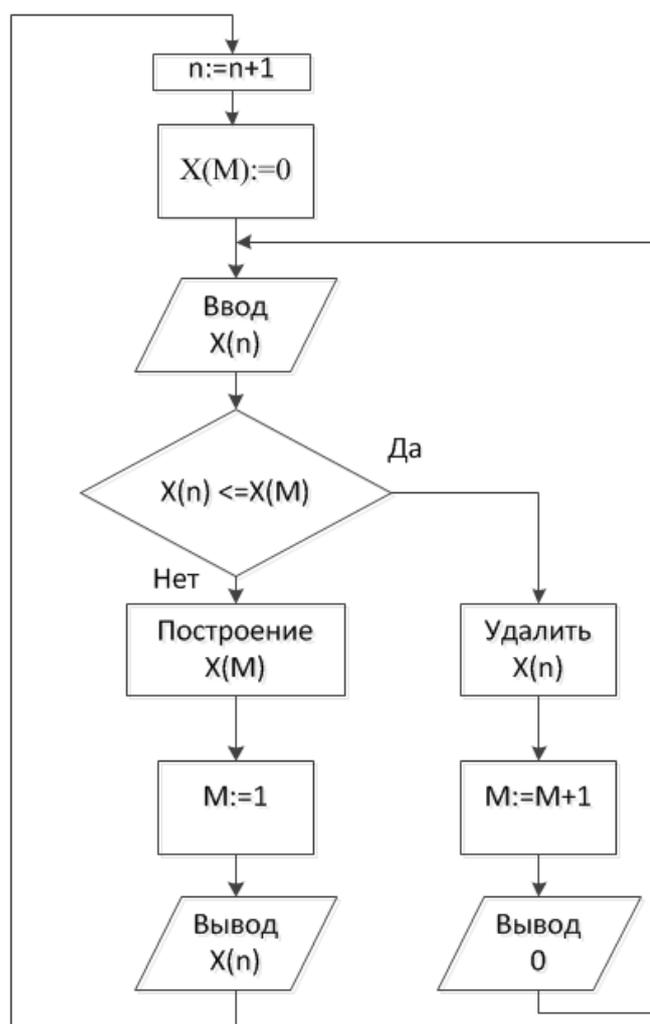


Рисунок 3.20 – Алгоритм работы блока «Линейный предиктор/маскирование»

В данном алгоритме  $X(n)$  – входной сигнал от полифазного фильтра,  $X(M)$  – построенная временная маска,  $n$  – номер временного отсчета входного сигнала,  $M$  – номер временного отсчета маски.

Пришедший на вход устройства отсчёт сравнивается с уровнем маски, построенной ранее. В случае, если он оказывается меньшим, на выходе формируется нулевое значение, в обратном - происходит перестройка маски соответственно новому пиковому значению, и на вывод  $X(n)$  попадает в неизменном виде. Длина маски увеличивается прямо пропорционально увеличению количества уровней квантования (уменьшения шума квантования), и вычисляется по формуле 3.15. Изменение длины маски в зависимости от частоты дискретизации и разрядности показано в таблице 3.15.

Таблица 3.15 – Соответствия количества отсчетов временной маски

Разрядность кода	Максимальное затухание, дБ	Максимальная длина, мс	Количество М		
			8КГц	22,05КГц	44,1КГц
8 бит	24	40	320	882	1764
12 бит	36	60	480	1323	2646
16 бит	48	100	800	2205	4410
24 бит	72	>160	1280	3308	6615

В результате вычислений блока «Линейного предиктора/маскирования» на вход преобразователя «время-частота» подаётся поток данных с удалёнными в результате маскирования временными отсчётами. Данный процесс проиллюстрирован на рисунке 3.20:

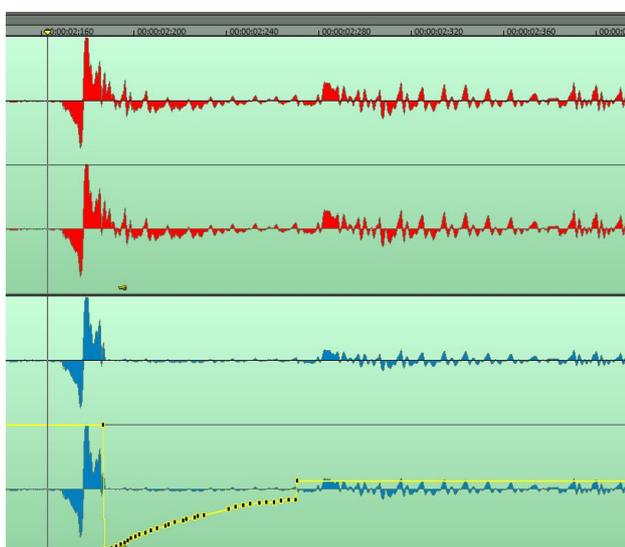


Рисунок 3.21 – Иллюстрация работы блока «Линейного предиктора/маскирования»

- входной РСМ сигнал;
- наложенная функция временной маски;
- выходной сигнал

При частоте дискретизации речевого сигнала 8 КГц и разрядности кода 8 бит количество удалённых отсчётов в представленном результате эксперимента ориентировочно составит 640, количество маскированной (непередаваемой) информации будет равно  $640 \times 8 = 5120$  бит. Соответственно, при дальнейшем ортогональном кодировании путём использования модифицированного дискретного косинусного преобразования частотные коэффициенты также будут нулевыми на всём удалённом участке.

Стоит отметить следующее: разборчивость речи при таком перераспределении временного потока остаётся на приемлемом уровне, что подтверждено экспериментально; данный подход нецелесообразно применять для сжатия музыкальных, либо иных звуковых сигналов.

Комплекс предложенных процедур позволяет увеличить степень плотности сжатия речевого сигнала, однако остаётся открытым вопрос о реальной скорости битрейта. Данный расчёт должен быть аппроксимирован при помощи результатов углублённого фонетического анализа наиболее употребляемых слов и словосочетаний в русском языке.

### **3.3 Определение параметров ортогональных преобразований**

Поступающий на вход кодера оцифрованный сигнал распараллеливается на два потока для частотного и временного перцептивного анализа. Первый поток проходит подготовительный этап разделения на узкие полосы, и затем происходит выделение основных энергетических составляющих звука. Второй подвергается широкополосному ортогональному преобразованию с целью анализа сигнала на частотное маскирование.

Таким образом, для дальнейшего синтеза кодера необходимо конкретизировать параметры адаптированных средств ортогональных преобразований. В качестве широкополосного средства в 2.1.2 выбрано АБПФ, и узкополосного – АДКП (п.2.1.1).

#### **3.3.1 Гибридная реализация адаптивного преобразования Фурье**

Как было сказано в пункте 1.2.3, преобразования время-частота нашли широкое применение в ЦОС и является сильным инструментом в любых алгоритмах фильтрации и распознавании сигналов.

Человеческим ухом воспринимаются звуки, лежащие в диапазоне частот от 20 Гц до 20кГц, что составляет примерно 500 полутонов. И в этом случае целесообразно вычисление спектра по алгоритму ДПФ 1000, который требует  $1000^2$  операций комплексного умножения и сложения [76,96].

Так как эффективно передаваемая полоса частот, канала тональной частоты, составляет 0,3 – 3,4 КГц, то для расчёта количества частотных отсчётов необходимо определить число полутонов (или тонов, в зависимости от выбранного качества) внутри данного диапазона. Согласно шкале барков, в рамках спектра речевого сигнала находятся 14 критических

диапазонов, и, следовательно, число спектральных элементов должно быть  $G = T \cdot note$ , где  $T$  - количество барков и  $note$  - число тонов или полутонов.

В случае решения с выделением тонов как качественных частотных показателей получим  $14 \cdot 7 = 98$  элементов спектра, что определяет 196 временных отсчётов. В другом случае, при выделении полутонов получим  $14 \cdot 12 = 168$  элементов спектра, что определяет 336 временных отсчётов.

Теперь необходимо определить, каким образом сгруппировать и прореживать временные отсчёты АБПФ, определённые в 2.1.2. Для начала вычислим количество операций комплексного умножения и суммирования. Адаптированный алгоритм работает только с  $2N = 2^s$  количеством входных элементов, где  $s \in \mathbb{N}$ , поэтому доставим необходимое количество и получим АБПФ(256) и АБПФ(512).

$$\sum_{\text{АБПФ}(256)} = 2N \cdot \log_2(2N) - N - 1 = 2 \cdot 128 \cdot \log_2(256) - 128 - 1 = 1919,$$

$$\prod_{\text{АБПФ}(256)} = 2N \cdot (\log_2(2N) - 2) - N - 1 = 2 \cdot 128 \cdot (\log_2(256) - 2) - 128 - 1 = 1407.$$

$$\sum_{\text{АБПФ}(512)} = 2N \cdot \log_2(2N) - N - 1 = 2 \cdot 256 \cdot \log_2(512) - 256 - 1 = 4351$$

$$\prod_{\text{АБПФ}(512)} = 2N \cdot (\log_2(2N) - 2) - N - 1 = 2 \cdot 256 \cdot (\log_2(512) - 2) - 256 - 1 = 3327$$

Очевидно, что делить последовательности звуковых сэмплов можно по-разному, однако от этого зависит, сможем ли мы при объединении получить неискажённый спектр сигнала и чего, с точки зрения энергетических потерь, это будет стоить. То есть можно сказать, что эффективность алгоритма АБПФ полностью зависит от способа разбиения и объединения последовательности временных элементов.

Чтобы определиться с количеством объединений и разбиений, необходимо ввести выборку длиной  $N = 2^L$ , где  $L$  - положительное целое число ( $L \in \mathbb{N}$ ). Для достижения максимального эффекта необходимо разделить последовательность пополам  $L-1$  раз, таким образом получив  $N/2$  групп по два отсчёта.

Для адаптированного БПФ (3.6), где  $N = 256$  ( $L = 8$ ) такое разбиение представлено на рисунке 3.21.

$$b_x = \sum_{n=1}^{256-1} X_x \cdot \cos\left(\frac{2\pi n \cdot x}{256}\right) + \sum_{n=1}^{256} X_x \cdot \sin\left(\frac{2\pi n \cdot x}{256}\right). \quad (3.6)$$

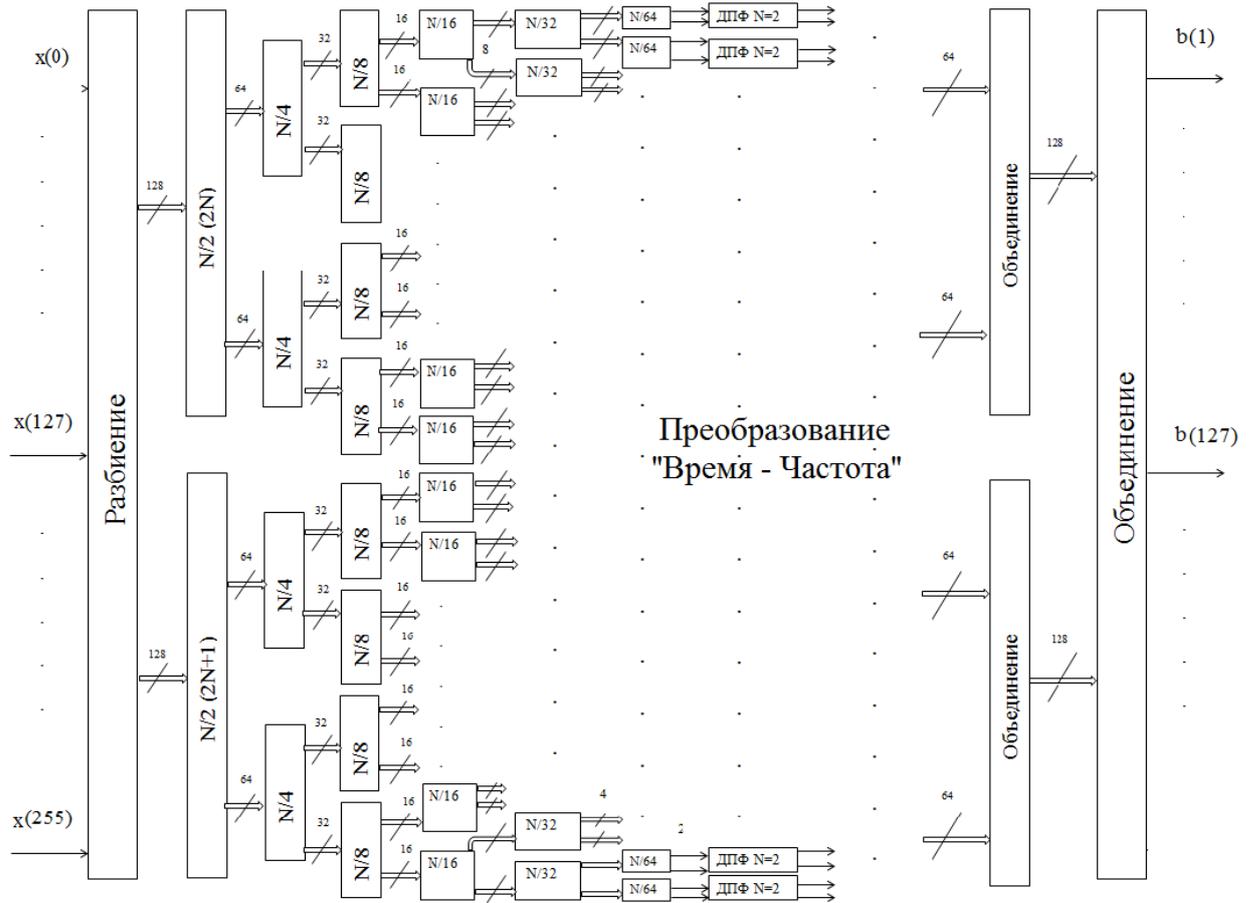


Рисунок 3.22 – Прореживание и объединение последовательностей при  $N = 256$

Как было показано в пункте 2.1.2, адаптированный метод БПФ заключается в прореживании временных отсчётов и укороченной процедуре вычисления-объединения частотных отсчётов.

Для начала введём два вектора:  $\{x_{[even]}\}$  и  $\{x_{[odd]}\}$  как частные составляющие  $\{x\}$  и определим их в выражении (3.7) и (3.8):

$$X_{[even]n} = X_{2n}, \quad (3.7)$$

$$X_{[odd]n} = X_{2n+1}, \quad (3.8)$$

где  $n = 0, 1, \dots, N/2 - 1$ .

Далее для представленных векторов применено ДПФ, результатом чего являются два новых массива  $\{X_{[even]}\}$  и  $\{X_{[odd]}\}$ , в каждом из которых количество элементов составляет  $N/2$ .

Из конечных реализаций алгоритмов БПФ следует в силу тригонометрических свойств двойного угла вектор  $\{X\}$  можно представить парой векторов  $\{X_{[\text{even}]}\}$  и  $\{X_{[\text{odd}]}\}$  в виде 3.9:

$$X_k = X_{[\text{even}]k} + W_N^k X_{[\text{odd}]k}, \quad X_k = X_{[\text{even}]k-N/2} - W_N^{k-N/2} X_{[\text{odd}]k-N/2}, \quad (3.9)$$

где  $k = 0, 1, \dots, N/2-1, k = N/2, \dots, N-1$ .

Параметр  $W_N^k$ , в данном случае, как и в 2.2, определяет поворачивающую функцию для операций определения перестроенных частотных элементов и  $W_N^k = e^{-j2\pi k/N}$ .

Проведём проверку выражения 3.9 путём подстановки обратного преобразования Фурье

$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \times e^{j2\pi kn/N}$  в выражение определения чётности:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \times e^{j2\pi kn/N},$$

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{kn} = \sum_{n=0}^{N/2-1} (x_{2n} W_N^{k2n} + x_{2n+1} W_N^{k(2n+1)}) =$$

$$= \sum_{n=0}^{N/2-1} x_{2n} W_N^{k2n} + \sum_{n=0}^{N/2-1} x_{2n+1} W_N^{k(2n+1)} =$$

$$= \sum_{n=0}^{N/2-1} X_{[\text{even}]n} W_N^{k2n} + \sum_{n=0}^{N/2-1} X_{[\text{odd}]n} W_N^{k(2n+1)} =$$

$$= \sum_{n=0}^{N/2-1} X_{[\text{even}]n} W_N^{k2n} + W_N^k \sum_{n=0}^{N/2-1} X_{[\text{odd}]n} W_N^{k2n}.$$

Используем свойство поворачивающей функции  $W_N^{k2n} = e^{-j2\pi k2n/N} = e^{-j2\pi kn/N/2} = W_{N/2}^{kn}$ :

Применив  $W_N^{k2n} = e^{-j2\pi k2n/N} = e^{-j2\pi kn/N/2} = W_{N/2}^{kn}$  для определения  $W_N^k = e^{-j2\pi k/N}$ , имеем (3.10):

$$\mathbf{X}_k = \sum_{n=0}^{N/2-1} X_{[\text{even}]n} \mathbf{W}_{N/2}^{kn} + \mathbf{W}_N^k \sum_{n=0}^{N/2-1} X_{[\text{odd}]n} \mathbf{W}_{N/2}^{kn}. \quad (3.10)$$

Произведём сравнение полученного выражения с формулами для чётного и нечётного элементов  $X_{[\text{even}]k}$  и  $X_{[\text{odd}]k}$ , где  $k=0,1,\dots,N/2-1$ :

$$X_{[\text{even}]k} = \sum_{n=0}^{N/2-1} X_{[\text{even}]n} \mathbf{W}_{N/2}^{kn}, \quad X_{[\text{odd}]k} = \sum_{n=0}^{N/2-1} X_{[\text{odd}]n} \mathbf{W}_{N/2}^{kn}. \quad (3.11)$$

Для проверки представленного выражения снова преобразуем функцию поворота  $\mathbf{W}_{N/2}^{(k-N/2) \times n} = \mathbf{W}_{N/2}^{kn}$ . Внедрим это выражение в структуру тождества  $\mathbf{W}_N^{k2n} = e^{\frac{-j2\pi k2n}{N}} = e^{\frac{-j2\pi kn}{N/2}} = \mathbf{W}_{N/2}^{kn}$  и изменим  $\mathbf{W}_N^k$ . Таким образом, получим (3.12):

$$\begin{aligned} \mathbf{X}_k &= \sum_{n=0}^{N/2-1} X_{[\text{even}]n} \mathbf{W}_{N/2}^{kn} + \mathbf{W}_N^k \sum_{n=0}^{N/2-1} X_{[\text{odd}]n} \mathbf{W}_{N/2}^{kn} = \\ &= \sum_{n=0}^{N/2-1} X_{[\text{even}]n} \mathbf{W}_{N/2}^{(k-N/2) \times n} - \mathbf{W}_N^{k-N/2} \sum_{n=0}^{N/2-1} X_{[\text{odd}]n} \mathbf{W}_{N/2}^{(k-N/2) \times n}. \end{aligned} \quad (3.12)$$

Для  $k=N/2,\dots,N-1$  по тождеству  $\mathbf{W}_N^k = e^{\frac{-j2\pi k}{N}}$  определим (3.13):

$$X_{[\text{even}]k-N/2} = \sum_{n=0}^{N/2-1} X_{[\text{even}]n} \mathbf{W}_{N/2}^{(k-N/2)n}, \quad X_{[\text{odd}]k-N/2} = \sum_{n=0}^{N/2-1} X_{[\text{odd}]n} \mathbf{W}_{N/2}^{(k-N/2)n}. \quad (3.13)$$

Для проверки утверждения 3.13 выполним следующую проверку. Применяя тождество 3.12 для 3.11, получим тождество 3.13, тем самым доказав достоверность положенного утверждения о том, что элементы вектора  $\{\mathbf{X}\}$  можно представить через элементы массивов  $\{X_{[\text{even}]}\}$  и  $\{X_{[\text{odd}]}\}$ .

Рассмотрим реализацию адаптированного алгоритма БПФ при прореживании N/128, N/64, N/32, N/16, N/8, N/4, N/2, для этого необходимо обратиться к формулам (3.9):

Формула объединения для N/128 имеет вид :

$$X_{\{128\}0} = X_{\{256\}[\text{even}]0} + W_{128}^0 X_{\{256\}[\text{odd}]0}, X_{\{128\}1} = X_{\{256\}[\text{even}]0} - W_{128}^0 X_{\{256\}[\text{odd}]0}. \quad (3.14)$$

Формула объединения для N/64 имеет вид (3.15):

$$X_{\{64\}0} = X_{\{128\}[\text{even}]0} + W_{64}^0 X_{\{128\}[\text{odd}]0}, X_{\{64\}2} = X_{\{128\}[\text{even}]0} - W_{64}^0 X_{\{128\}[\text{odd}]0}, \\ X_{\{64\}1} = X_{\{128\}[\text{even}]1} + W_{64}^1 X_{\{128\}[\text{odd}]1}, X_{\{64\}3} = X_{\{128\}[\text{even}]1} - W_{64}^1 X_{\{128\}[\text{odd}]1}. \quad (3.15)$$

Формула объединения для N/32 имеет вид (3.16):

$$X_{\{32\}0} = X_{\{64\}[\text{even}]0} + W_{32}^0 X_{\{64\}[\text{odd}]0}, X_{\{32\}4} = X_{\{64\}[\text{even}]0} - W_{32}^0 X_{\{64\}[\text{odd}]0}, \\ X_{\{32\}1} = X_{\{64\}[\text{even}]1} + W_{32}^1 X_{\{64\}[\text{odd}]1}, X_{\{32\}5} = X_{\{64\}[\text{even}]1} - W_{32}^1 X_{\{64\}[\text{odd}]1}, \\ X_{\{32\}2} = X_{\{64\}[\text{even}]2} + W_{32}^2 X_{\{64\}[\text{odd}]2}, X_{\{32\}6} = X_{\{64\}[\text{even}]2} - W_{32}^2 X_{\{64\}[\text{odd}]2}, \\ X_{\{32\}3} = X_{\{64\}[\text{even}]3} + W_{32}^3 X_{\{64\}[\text{odd}]3}, X_{\{32\}7} = X_{\{64\}[\text{even}]3} - W_{32}^3 X_{\{64\}[\text{odd}]3}. \quad (3.16)$$

Формула объединения для N/16 имеет вид (3.17):

$$X_{\{16\}0} = X_{\{32\}[\text{even}]0} + W_{16}^0 X_{\{32\}[\text{odd}]0}, X_{\{16\}8} = X_{\{32\}[\text{even}]0} - W_{16}^0 X_{\{32\}[\text{odd}]0}, \\ X_{\{16\}1} = X_{\{32\}[\text{even}]1} + W_{16}^1 X_{\{32\}[\text{odd}]1}, X_{\{16\}9} = X_{\{32\}[\text{even}]1} - W_{16}^1 X_{\{32\}[\text{odd}]1}, \\ X_{\{16\}2} = X_{\{32\}[\text{even}]2} + W_{16}^2 X_{\{32\}[\text{odd}]2}, X_{\{16\}10} = X_{\{32\}[\text{even}]2} - W_{16}^2 X_{\{32\}[\text{odd}]2}, \\ X_{\{16\}3} = X_{\{32\}[\text{even}]3} + W_{16}^3 X_{\{32\}[\text{odd}]3}, X_{\{16\}11} = X_{\{32\}[\text{even}]3} - W_{16}^3 X_{\{32\}[\text{odd}]3}, \\ X_{\{16\}4} = X_{\{32\}[\text{even}]4} + W_{16}^4 X_{\{32\}[\text{odd}]4}, X_{\{16\}12} = X_{\{32\}[\text{even}]4} - W_{16}^4 X_{\{32\}[\text{odd}]4}, \quad (3.17)$$

$$X_{\{16\}5} = X_{\{32\}[\text{even}]5} + W_{16}^5 X_{\{32\}[\text{odd}]5}, \quad X_{\{16\}13} = X_{\{32\}[\text{even}]5} - W_{16}^5 X_{\{32\}[\text{odd}]5},$$

$$X_{\{16\}6} = X_{\{32\}[\text{even}]6} + W_{16}^6 X_{\{32\}[\text{odd}]6}, \quad X_{\{16\}14} = X_{\{32\}[\text{even}]6} - W_{16}^6 X_{\{32\}[\text{odd}]6},$$

$$X_{\{16\}7} = X_{\{32\}[\text{even}]7} + W_{16}^7 X_{\{8\}[\text{odd}]7}, \quad X_{\{16\}14} = X_{\{32\}[\text{even}]7} - W_{16}^7 X_{\{32\}[\text{odd}]7}.$$

Выражения, объединяющие временные отсчёты  $N/8$ ,  $N/4$ ,  $N/2$ , имеют тот же вид и выражены соотношением (3.18) для значений с индексами  $2n$  и  $2n-1$ :

$$X_{\{N\}k} = X_{\{2N\}[\text{even}]k} + W_N^k X_{\{2N\}[\text{odd}]k}, \quad X_{\{N\}k-1} = X_{\{2N\}[\text{even}]k-1} - W_N^k X_{\{2N\}[\text{odd}]k-1}, \quad (3.18)$$

где  $k = 0, 1, \dots, N/2-1$ .

В том случае, при котором необходимо преобразование от 512 отсчётов, действия по прореживанию будут аналогичны АБПФ-256, отличием будут лишь процессы объединения и вычисления с поворачивающей функцией  $W_{128}^k = e^{-j\frac{2\pi k}{128}}$  в завершении представления конечных 255 отсчётов.

Представленная реализация адаптивного алгоритма АБПФ в кодере предназначена для создания математической основы при построении психоакустической модели частотно-временного маскирования неслышимых для уха человека тонов под влиянием на них более громких.

### 3.3.2 Реализация быстрого дискретного косинусного преобразования

Косинусное преобразование не может быть полностью обратимым. Его результатом является выделение энергетического вещественного спектра сигнала, однако при этом теряется изначальная фаза различных частотных элементов. Вот почему так важно использовать техники полифазной фильтрации при подготовке к перцептивному ортогональному кодированию.

Любое ортогональное преобразование несёт в себе шумы (помехи), возникающие вне диапазона самого преобразования, создавая при этом так называемые боковые лепестки. Поэтому важно обеспечить два условия выполнения данной процедуры. Первое – это

использовать максимально возможное прямое окно преобразований, что выполняется блоком банка фильтров с синтезированным набором АЧХ. Второе – не допустить проникновения инородных для спектра преобразования частотных элементов в результат вычисления. Данное требование выполняют ограничения, наложенные на вычисление выходных отсчётов самого косинусного преобразования.

АДКП разработано для выполнения операций над блоками, где происходит наложение и базируется на ДКП – IV [47,99]. АДКП представляет собой линейную функцию (3.19), которая преобразует действительные числа  $2N$  к реальному, согласно уравнениям  $F: \mathbb{R}^{2N} \Rightarrow \mathbb{R}^N$ , где  $\mathbb{R}$  – множество реальных.

$$X_k = \sum_{n=0}^{2N-1} x_n \cos \left[ \frac{\pi}{N} \left( n + \frac{1}{2} + \frac{N}{2} \right) \times \left( k + \frac{1}{2} \right) \right], \quad (3.19)$$

где  $N$  – некоторое реальное число,  $n$  – количество временных отсчётов,  $k$  – количество частотных отсчётов.

Блок АДКП ограничивает количество спектральных составляющих аудиосигнала до процедуры применения маскирования. С помощью обратного АДКП возможно восстановить исходный образ аудиосигнала в декодере (3.20):

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cos \left[ \frac{\pi}{N} \left( n + \frac{1}{2} + \frac{N}{2} \right) \left( k + \frac{1}{2} \right) \right] \quad (3.20)$$

где,  $n = 0, \dots, 2N-1$ ,  $k = 0, \dots, N-1$ ,  $N$  – количество входных (выходных) отсчётов.

Ввиду определённого ранее количества критичных диапазонов, сделаем вывод о том, что количество параллельных процедур АДКП составляет 14. На входе процесса размещены временные отсчёты, и на выходе – частотные отсчёты, которые в дальнейшем необходимы для обработки спектра аудиосигнала при помощи перцептивного анализа.

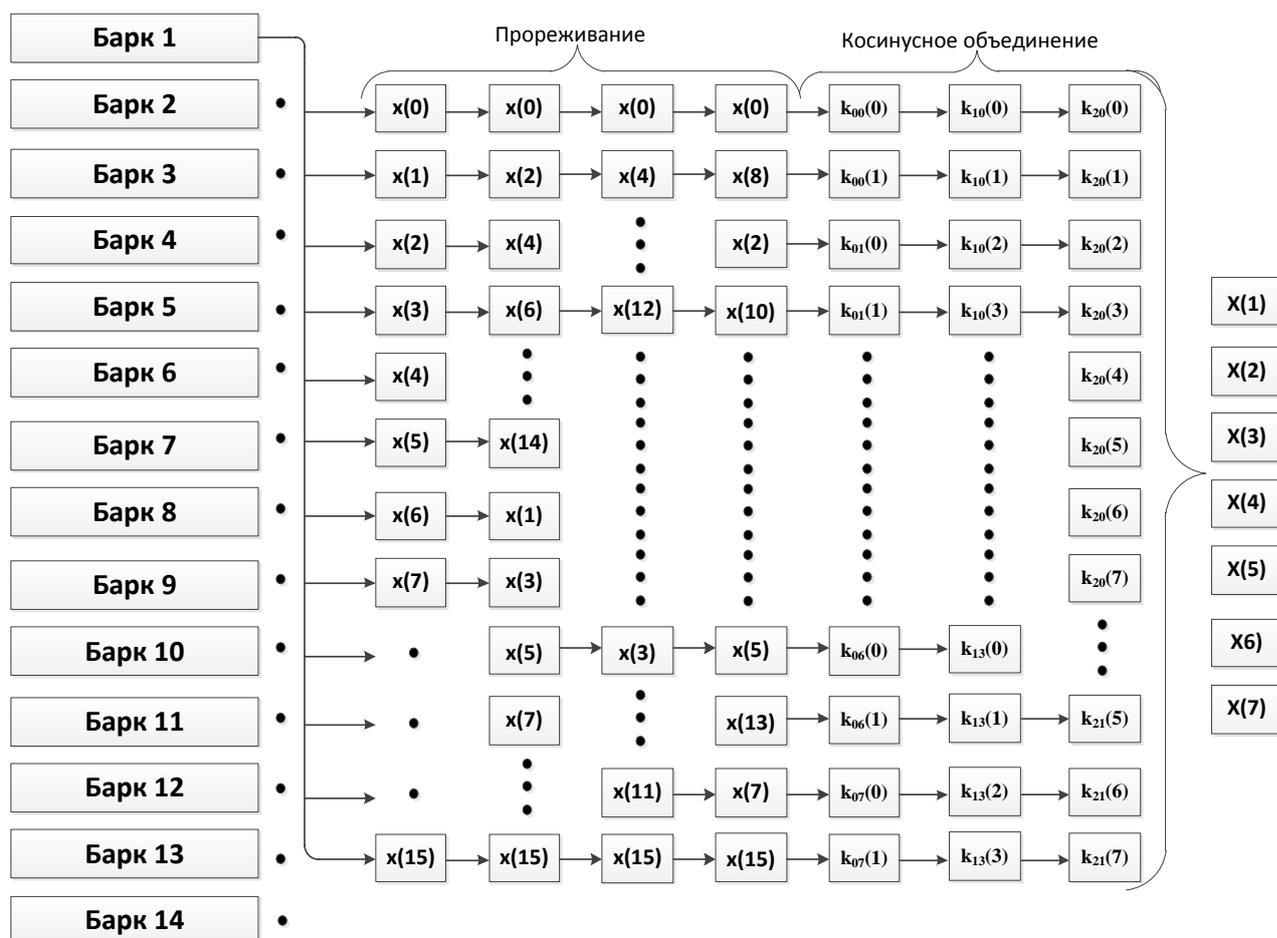


Рисунок 3.23 – Функциональная схема адаптивного ДКП

Ортогональное преобразование, представленное на рисунке 3.22, выполняется для каждого критичного диапазона. В силу необратимости процесса, на данном этапе происходят первые потери при сжатии речевого сигнала. Однако, ввиду того, что слух человека не замечает фазовые шумы на частотах разного диапазона, передача основной энергии сигнала является достаточным условием кодирования.

### 3.4 Построение частотно-временной маски для ортогонального речевого кодирования на основе психоакустической модели слуха

Входными данными для данной процедуры является результат адаптивного быстрого преобразования Фурье, вычисленный из потока битов с аналого-цифрового преобразователя в виде сигнала с импульсно-кодовой модуляцией (ИКМ). На выходе процедуры формируется матрица частотных коэффициентов, являющихся входными параметрами для процессов масштабирования и квантования в речевом кодере [72,22]. Построение маски – это

динамический процесс, реализующийся как в частотной, так и во временной плоскости. Процесс заключается в использовании эффекта маскирования «тихих» тонов более «сильными» [70,24,52]. Исходными данными является ряд частотных коэффициентов Фурье, представленный формулой (3.21) [80,124]. В зависимости от быстродействия кодера (количества входных отсчётов для БПФ), количество одновременно загружаемых значений может достигать 4096 [89].

$$X_m = \sum_{n=0}^{M-1} x_{2n} a_M^{nm} + \exp(-2\pi i \frac{m}{N}) \sum_{n=0}^{M-1} x_{2n+1} a_M^{nm} \quad (3.21)$$

Рассмотрим построение дискретной математической модели маски и алгоритм построения коэффициентов, влияющих на точность маски в дискретном исполнении. Определим модели уровня слуха при абсолютной тишине и его маскирование сильным тональным сигналом. В частотной области происходит наложение двух дискретных амплитудно-частотных характеристик (АЧХ) вида, представленного формулой (3.22): первой является модель слуха человека в абсолютной тишине (рисунок 3.23а), коэффициенты для которой определяются формулой (3.23); второй – психоакустическая модель (рисунок 3.23 б), отображающая порог частотного маскирования, модель определяется по формуле (3.23) [72,70].

$$T(f) = \alpha \times \left( \frac{f}{1000} \right)^{-0.8} - \beta \times e^{-0.6 \times \left( \frac{f}{1000} - 3.3 \right)^2} + \gamma \times \left( \frac{f}{1000} \right)^4, \quad (3.22)$$

$\alpha$  - коэффициент восприимчивости басовых звуков;

$\beta$  - коэффициент восприимчивости речевого диапазона;

$\gamma$  - коэффициент восприимчивости высоких звуков.

$$SF(x) = \varepsilon + \Omega \times (x + \mu) - \Theta \times \sqrt{1 + (x + \mu)^2}, \quad (3.23)$$

$\varepsilon$  - уровень пика маски для громкого звука;

$\Omega$  - плотность маскирования соседних тонов;

$\mu$  - линейность спадов маски;

$\Theta$  - ширина диапазона маски.

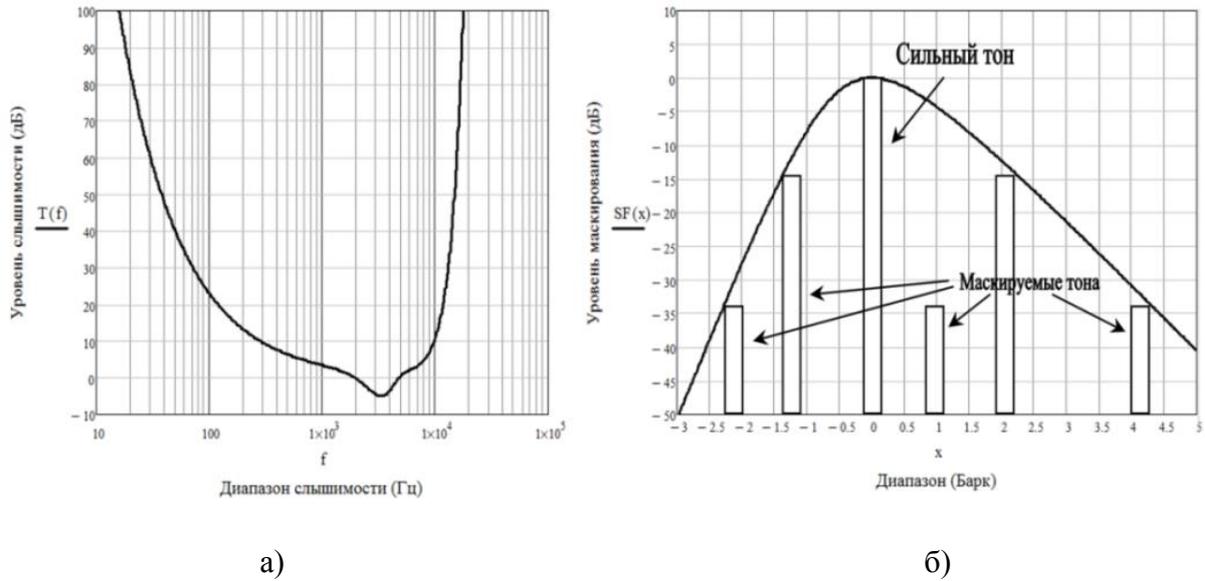


Рисунок 3.24 – График АЧХ слуха – а, и график модели порога частотного маскирования – б.

Как показано на графике 3.23а, чувствительность слуха максимальна в диапазоне от 1 до 4 КГц. Порог маскирования «сильным» тоном (график 3.23б) показывает распространение маски по диапазону, т.е. неслышимыми становятся все тона, уровень которых находится ниже функции 3.23б.

$$W(t) = (1-\omega)\left(1+\frac{2t}{\varphi}\right)\exp\left(-\frac{2t}{\varphi}\right) + \omega\left(1+\frac{2t}{\sigma}\right)\exp\left(-\frac{2t}{\sigma}\right), \quad (3.24)$$

$\varphi$  - постоянная времени, связанная с нарастающим к пику действующим значением амплитуды звука, определяет пред-маску;

$\sigma$  - постоянная времени, связанная с окончанием (спадом), определяет пост-маску;

$\Omega$  - определяет точку перехода между пиковыми и конечными частями функции.

Модель маскирования во временной плоскости изображена на графике рисунка 3.34. Имеет место маскирование до появления «сильного» сигнала в интервале времени  $\Delta t \sim 20$  мс и после него в интервале времени  $\Delta t \sim 160$  мс [40, 21].



Рисунок 3.25 – График модели временного маскирования

Опираясь на неоднородную зависимость значений порога слышимости во всём диапазоне, в процессе кодирования речи используют критичные интервалы частот (барки). В соответствии с распределением диапазона частот по критичным областям, меняется частотная маска. Распределение интервалов по частоте для речевого диапазона (300-3400 Гц) определяется выражением (3.25) [72, 24], пример распределения критичных диапазонов которого показан на графике, представленном на рисунке (3.24).

$$z(f) = 13 \times \tan^{-1}(0,00076f) + 3,5 \times \tan^{-1} \left[ \left( \frac{f}{7500} \right)^2 \right]. \quad (3.25)$$

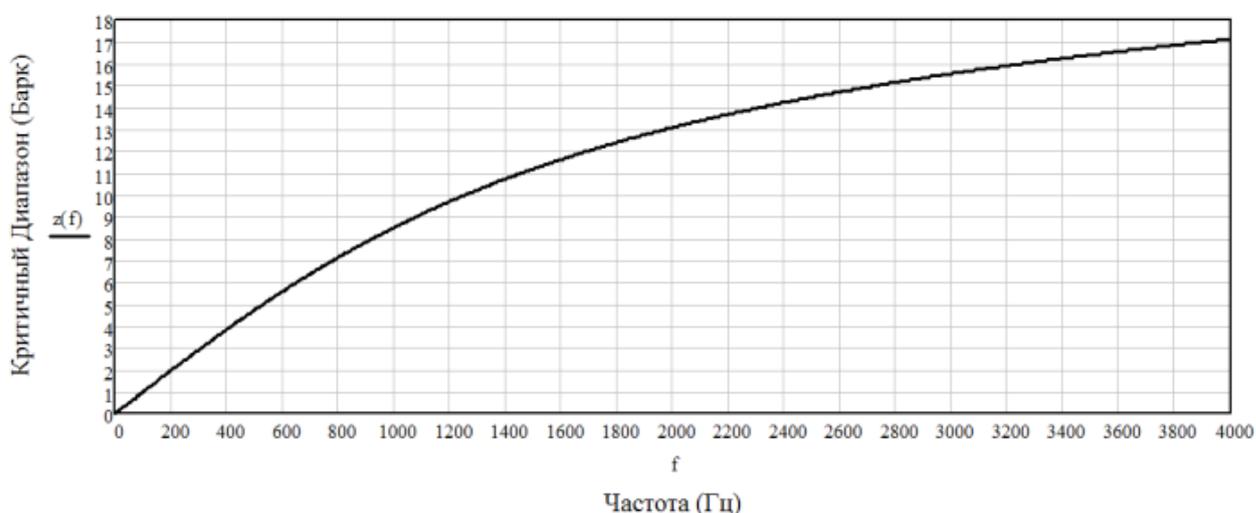


Рисунок 3.26 – График распределения критичных интервалов по частоте

На основании выражения (3.25) с использованием зависимости средней частоты барка от его частотного диапазона (3.25) [40] и с учётом частотного диапазона речевого сигнала, могут быть определены значения критичных диапазонов частот. Пример значений барков приведён в таблице 3.16. Учитывая то, что данные значения напрямую влияют на разборчивость речевого сигнала и его битрейт, определение данных параметров является необходимым элементом общей процедуры анализа выбора параметров кодера.

$$BW_c(f) = 25 + 75 \times \left[ 1 + 1,4 \left( \frac{f}{1000} \right)^2 \right]^{0,69}. \quad (3.26)$$

Таблица 3.16 - Критичные интервалы частот речи

Диапазон (Барк)	Центральная частота (Гц)	Ширина (Гц)	Диапазон (Барк)	Центральная частота (Гц)	Ширина (Гц)
1	50	-100	10	1175	1080-1270
2	150	100-200	11	1370	1270-1480
3	250	200-300	12	1600	1480-1720
4	350	300-400	13	1850	1720-2000
5	450	400-510	14	2150	2000-2320
6	570	510-630	15	2500	2320-2700
7	700	630-770	16	2900	2700-3150
8	840	770-920	17	3400	3150-3700
9	1000	920-1080	18	4000	3700-4400

Используя неоднородные зависимости характеристик слуха и его маскирования при распределении критичных диапазонов (Барков), представленные в таблице 3.16, построим матрицу результата вычисления маски  $M(n,m)$  (формула 3.27), где  $k_{ij}$  - весовые коэффициенты БПФ,  $i=1(1)n$  – номер частотного отсчёта,  $j=1(1)m$  – номер временного отсчёта. Размерность матрицы определяется количеством коэффициентов БПФ (по строкам) и временем преобразования БПФ (по столбцам), зависящим от частоты дискретизации ИКМ сигнала.

$$M = \begin{vmatrix} k_{11} & k_{12} & \dots & k_{1(n-1)} & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2(n-1)} & k_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ k_{(m-1)1} & k_{(m-1)2} & \dots & k_{(m-1)(n-1)} & k_{(m-1)n} \\ k_{m1} & k_{m2} & \dots & k_{m(n-1)} & k_{mn} \end{vmatrix}. \quad (3.27)$$

Элементы матрицы  $k_{ij}$  взаимосвязаны как по частоте, так и по времени. Точность построения дискретной маски зависит от значений и количества коэффициентов  $k_{ij}$ . Для частотных коэффициентов маски определяется их зависимость от «соседних» коэффициентов, т.е. выявляется, подвержен ли данный тон маскированию или нет. В первом случае в этом месте матрицы ставится нулевой коэффициент  $k_{ij}$ , в противном – коэффициент остаётся прежним. Во временной области взаимодействие определяется по столбцам, принцип выявления нулевых  $k_{ij}$  тот же, что и по строкам.

Вычисления представляют собой два цикла: в первом определяются элементы строк, т.е. коэффициенты БПФ с учётом частотного маскирования (вычисляются «слева-направо»), во втором – элементы столбцов с учётом временного маскирования (вычисляются пред- и пост-маска для каждого элемента построчно «сверху-вниз»). Параметры  $T_i$  и  $T_j$  являются значениями порога маскирования по строкам и столбцам соответственно.

Процедура наложения временной маски является аналогичной для временных коэффициентов  $k_{ij}$ . Следующим этапом кодирования является наложение полученной маски на результат дискретного косинусного преобразования входного сигнала ИКМ.

Таким образом, происходит построение матрицы вида, определённого формулой (3.26), состоящей из горизонтальных рядов значений масок. В результате те коэффициенты, которые оказываются ниже порога маскирования (рисунки 3.23, 3.24, 3.25), будут обращены в ноль.

Процедура определения элементов матрицы в общем виде для ряда БПФ (формула 3.27) и распределения критических частот (таблица 3.16) представлена в виде алгоритма (рисунок 3.26).

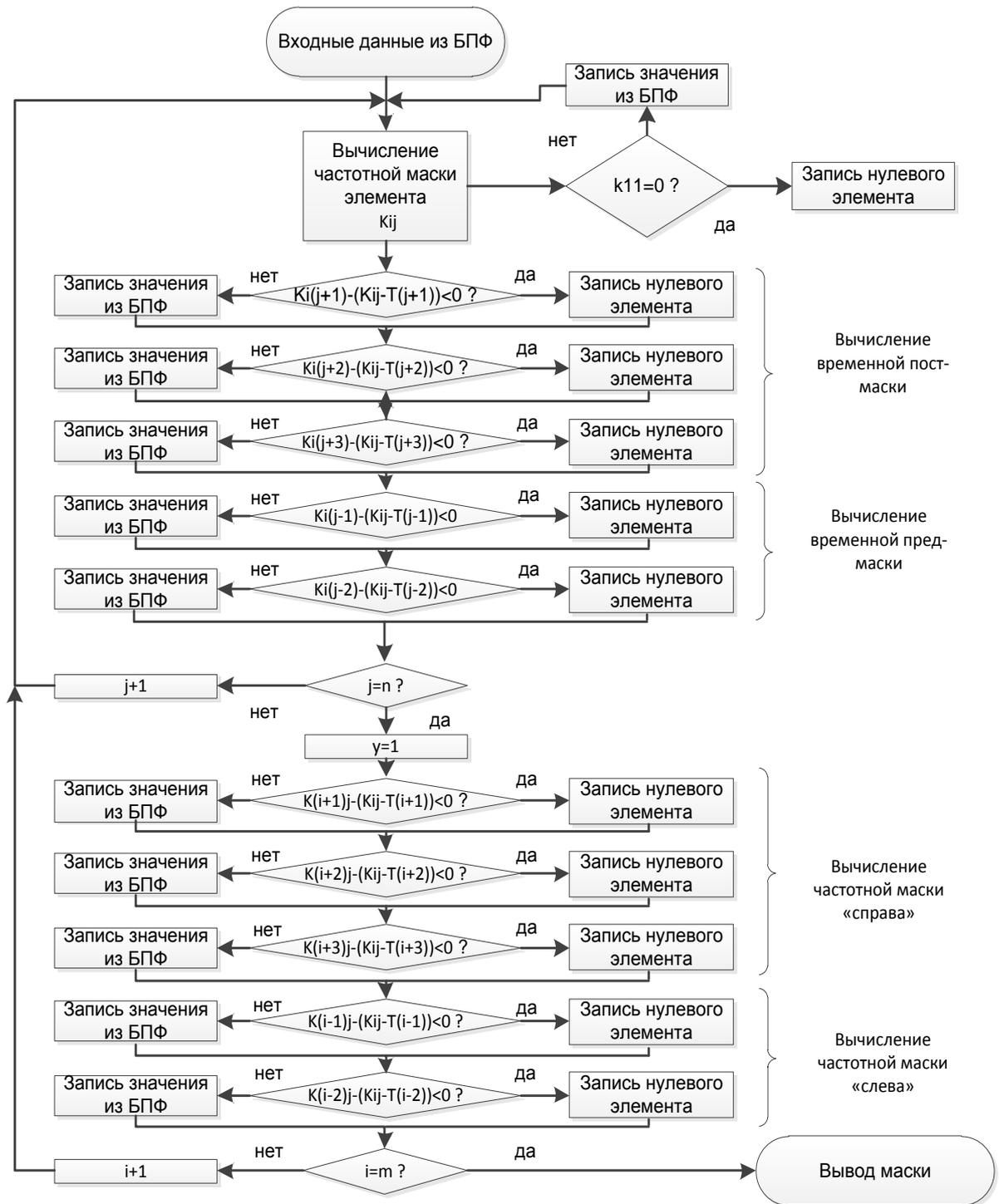


Рисунок 3.27 – Алгоритм вычисления выходной матрицы частотно-временной маски

Таким образом, сделаем вывод о том, что данный комплекс процедур позволяет управлять процессом сжатия речевого сигнала, а именно влиять на его качество (разборчивость) и битрейт. Вариативными параметрами для построения являются интервалы деления по критичным диапазонам, параметры вычисления коэффициентов (уровней маскирования по частоте и времени) и размерность матрицы маски.

Стоит отметить, что точность определения коэффициентов маскирования зависит от количества отсчётов БПФ и уровней квантования входного сигнала ИКМ. Процедура наложения, по сути, схожа с процессом прореживания по частоте при дискретном преобразовании Фурье. Графически процесс проиллюстрирован на рисунке 3.27.

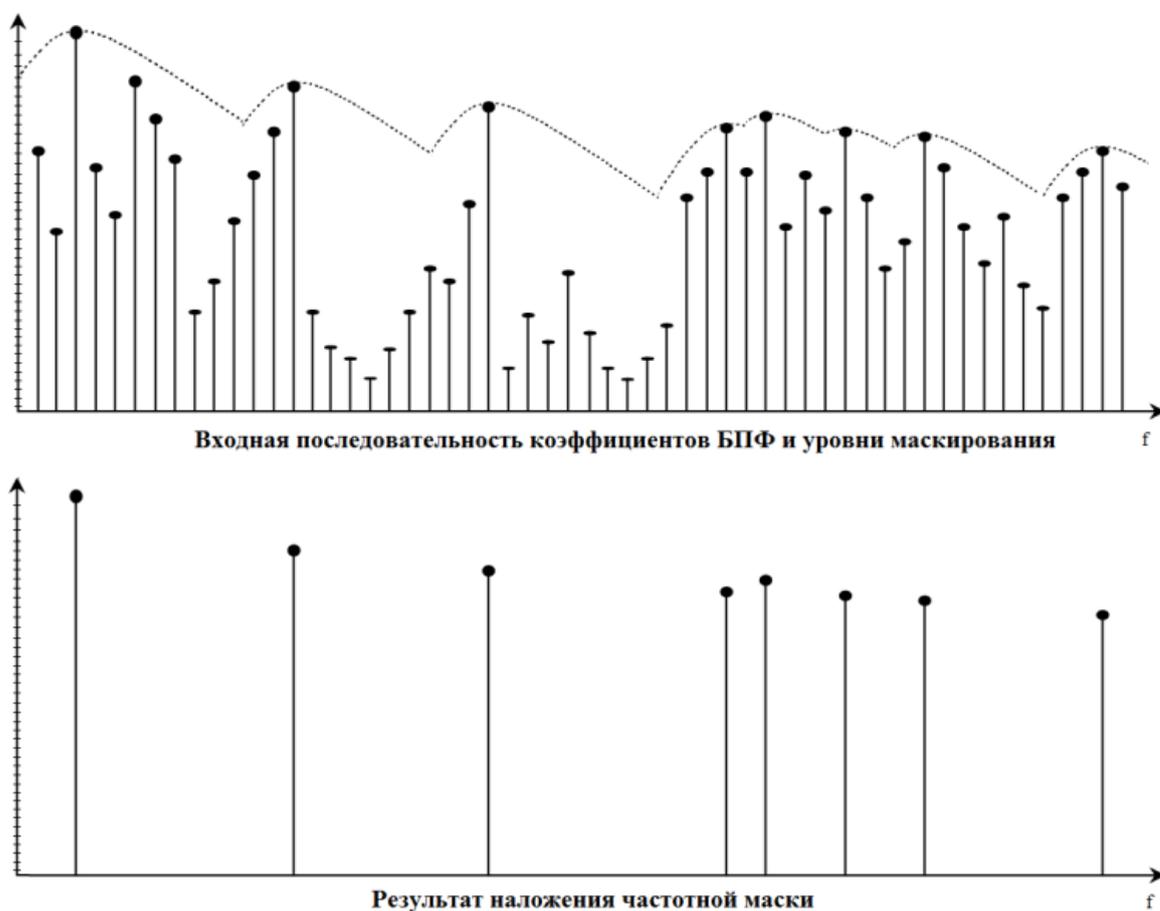


Рисунок 3.28 – Определение маски для дискретного сигнал БПФ. Пунктиром показан уровень маскирования

Результатом наложения маски является спектр, содержащий основные форманты речи в текущем временном отрезке, определяющие выразительность и уникальность голоса.

Оценка выразительности речевого сигнала при анализе громких тонов речи, используя двумерное БПФ,[1,14] наиболее актуальна при определении качественных критериев работы системы кодирования речи по принципу «время-частота».

### 3.5 Выводы по третьей главе

Построенная математическая модель наделяет средство сжатия параметрическими и функциональными взаимосвязями, что позволяет говорить о её динамичности и, при изменении методической и алгоритмических операций, – о её вариативности.

Выделим свойства составных частей модели как результат работы по данной главе:

- совокупность операций цифровой обработки сигналов по разделению широкополосного сигнала на критичные диапазоны должна приносить минимальные фазовые сдвиги, что обеспечит незначительные звуковые искажения и подготовит сигнал к реализации сжатия посредством удаления маскируемых тонов;

- выявленные особенности произношения русского языка создают основу для повышения плотности информативных составляющих при передаче речевого потока путём удаления редуцированных звуков;

- ортогональные преобразования временного речевого потока реализуют спектр тональностей в широком и узком диапазонах и осуществляют организацию спектральных массивов для получения и наложения психоакустической маски;

- вариация математических функций восприятия звука задаёт его амплитудно-частотные характеристики, которые содержат основные форманты речи в текущем временном отрезке, определяющие выразительность и уникальность голоса;

- модель абсолютного порога слуха является входной частотной характеристикой с динамическими параметрами диапазона восприятия речи среди всего спектра слуха и будет использована для анализа выразительности голоса, а также возможности удаления информационно невосприимчивых тональностей звука;

- наложение частотно-временной маски выполняет основную роль в процедуре сжатия с потерями, от вариации их параметров зависит как степень компрессии, так и качество воспроизводимого звука.

## 4 Структурный синтез адаптивного кодера и средства оценки его качественных свойств

Наличие результатов исследований математических моделей и набора методик сжатия позволяет осуществить синтез структуры кодера в целом. А также определить в ней вариативные параметры и степень их влияния на изменение речевого потока, установить степень динамичности цепей внутри структуры.

### 4.1 Функциональное взаимодействие моделей масок

С целью объединения перцептивных методов обработки звукового потока разделим поток на параллельные ортогональные преобразования. В первом будут происходить процессы подготовки к вычислению и наложению временной маски, во втором – подготовка и вычисление частотной маски. Как показано на рисунке 4.1, в результате реализуется взаимодействие трёх моделей: частотной, временной и абсолютной.

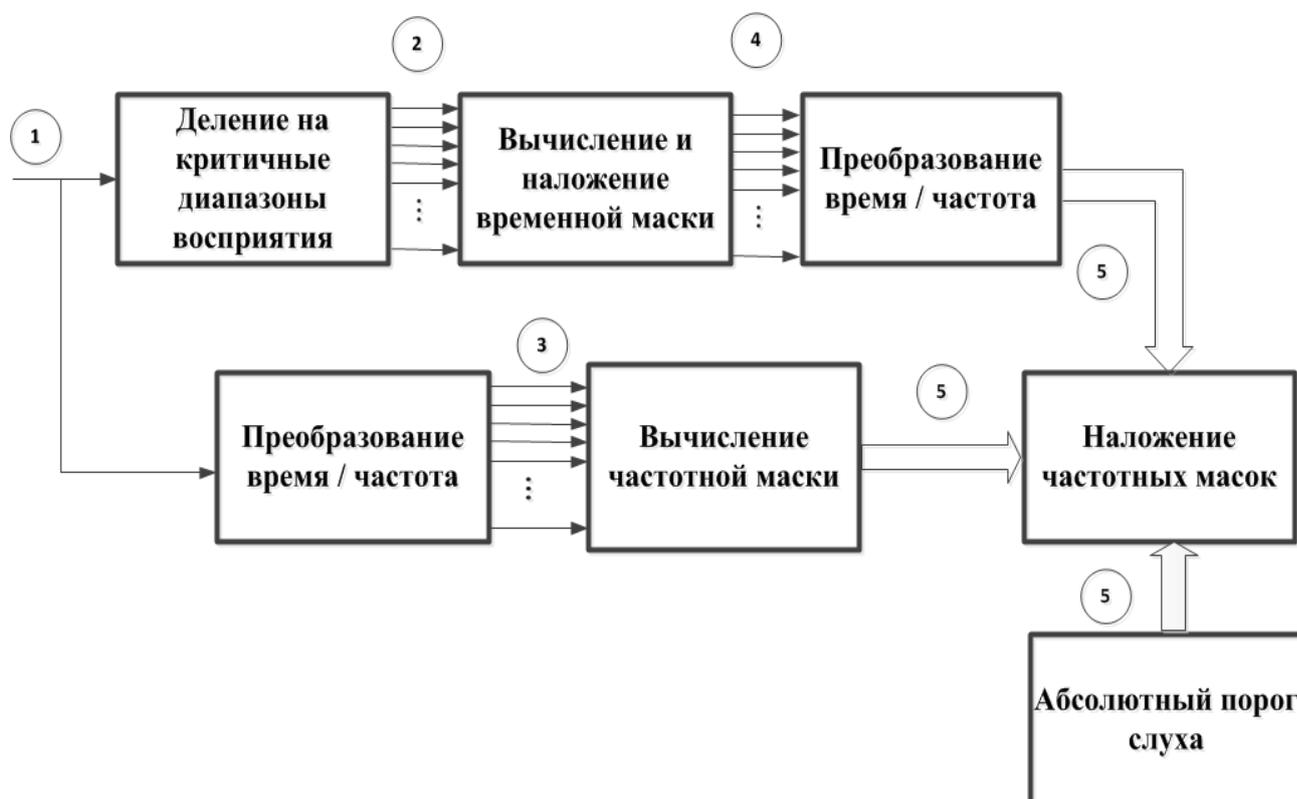


Рисунок 4.1 - Функциональная схема взаимодействия моделей масок

Для последующей программной реализации выделим отдельно входные и выходные потоки в каждом блоке преобразований.

Поток №1 представляет собой вектор отсчётов аналого-цифрового преобразования вида  $b = [b_0, b_1, b_2, b_3, \dots, b_n]$ . Каждый отсчёт  $b_i$  имеет одинаковую разрядность, определённую квантованием.

Поток №1 представляет собой вектор дискретных отсчётов аналого-цифрового преобразования вида  $b = [b_0, b_1, b_2, b_3, \dots, b_n]$  с разрядностью  $n$ , определённой точкой изгиба  $q$  (9) передаточной характеристики неравномерного квантования с вычисленной разрядностью  $m$  при заданном динамическом диапазоне  $D$ .

$$b_i = \begin{cases} \frac{A \cdot x_i \cdot 2^n}{1 + \ln A}, & \text{для } x \leq \frac{1}{A} \\ \frac{1 + \ln(A \cdot \frac{x_i}{2^m})}{1 + \ln A} \cdot 255, & \text{для } x > \frac{1}{A} \end{cases}; \quad A = 2^m \div q; \quad m = \log_2 10^{D/10}$$

Блок «Деление на критичные диапазоны» выполняет функцию разделения широкополосного сигнала на узкие полосы, соответствующие выбранным баркам. По сути, реализуется полосовая фильтрация с параметрами фильтров, установленными в разделе 3.1. На рисунке 4.2 показана программная реализация процедуры разделения.

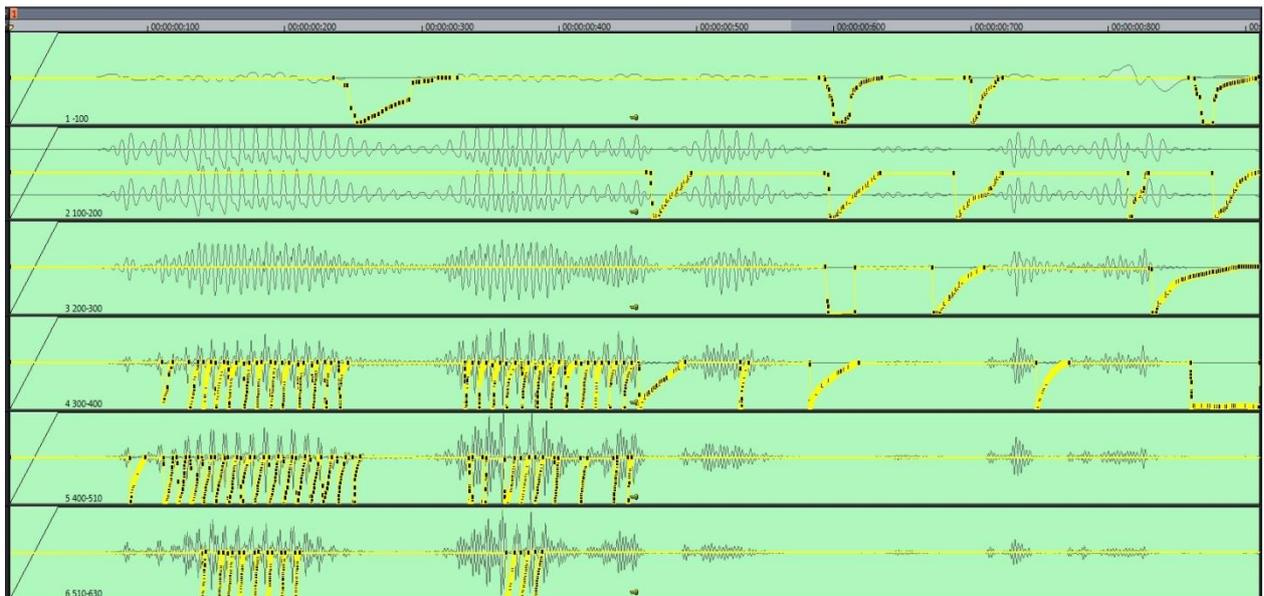


Рисунок 4.2 – Разделённый на критичные полосы (1-6) речевой поток с удалёнными неслышимыми дискретами

Поток №2 представляет собой параллельные векторы  $\hat{b}$  (матрицу), отображающие критичные полосы:

$$\hat{b} = \begin{bmatrix} \hat{b}_{00} & \hat{b}_{01} & \hat{b}_{02} & \dots & \hat{b}_{0n} \\ \hat{b}_{10} & \hat{b}_{11} & \hat{b}_{12} & \dots & \hat{b}_{1n} \\ \hat{b}_{20} & \hat{b}_{21} & \hat{b}_{22} & \dots & \hat{b}_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{b}_{m0} & \hat{b}_{m1} & \hat{b}_{m2} & \dots & \hat{b}_{mn} \end{bmatrix}$$

Разрядность данных сохраняется, а разделение на критичные полосы посредством полифазной нерекурсивной фильтрации обеспечивает минимальные фазовые искажения (сдвиги).

«Преобразование «время/частота» происходит по гибриднему алгоритму АДКП (ДПФ-IV), описанному в 2.1.1. На вход поступает разделённый на кадры узкополосный параллельный поток, проиллюстрированный ниже.

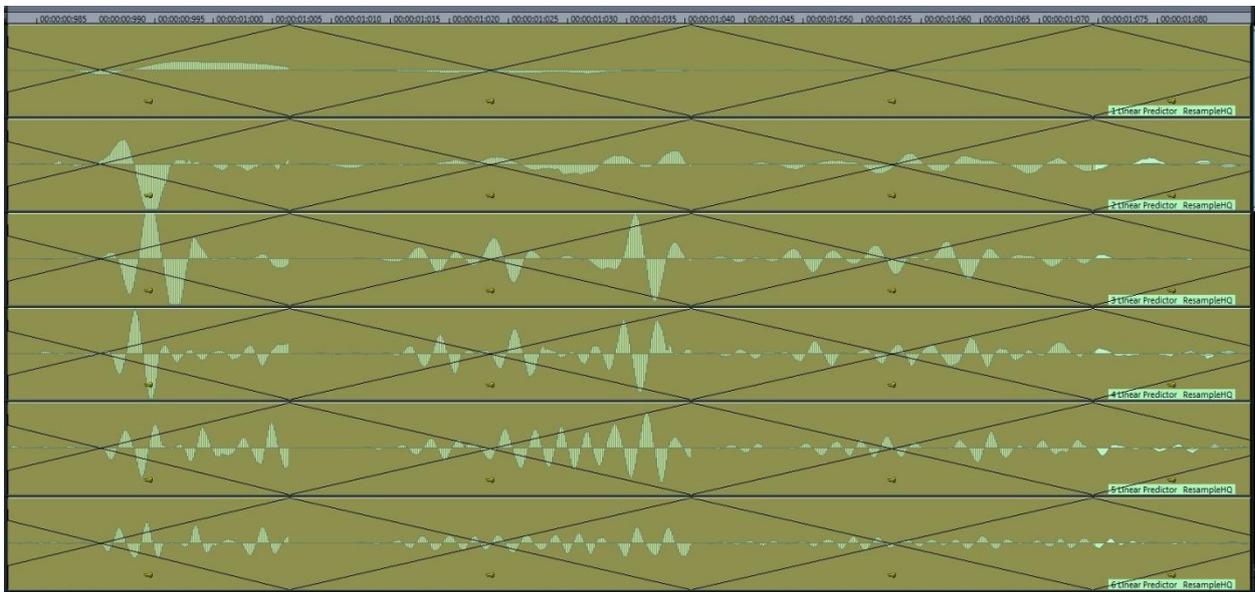


Рисунок 4.3 – Подготовка параллельного потока к ортогональному преобразованию

Результатом спектрального преобразования №4 является матрица, количество столбцов которой уменьшено вдвое по сравнению с  $\hat{b}$  :

$$\hat{f} = \begin{bmatrix} \hat{f}_{00} & \hat{f}_{01} & \hat{b}_{02} & \cdots & \hat{f}_{0(n/2)} \\ \hat{f}_{10} & \hat{f}_{11} & \hat{f}_{12} & \cdots & \hat{f}_{1(n/2)} \\ \hat{f}_{20} & \hat{f}_{21} & \hat{f}_{22} & \cdots & \hat{f}_{2(n/2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{f}_{m0} & \hat{f}_{m1} & \hat{f}_{m2} & \cdots & \hat{f}_{m(n/2)} \end{bmatrix}$$

Второй блок «преобразование «время/частота» представляет преобразованное БПФ, в структуру которого встроено косинусное преобразование от четырёх отсчётов, синтезированные в 3.3.1. Выходные данные №3 представляют собой массивы спектров для анализа сэмпла на маскирование по частоте и в совокупности выстраиваются в матрицу 3.4.

$$b = [b_0, b_1, b_2, b_3, \dots, b_n] \Rightarrow \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ \vdots \\ k_{n/2} \end{bmatrix} \Rightarrow \begin{bmatrix} k_0 \cdot z^{-m} \\ k_1 \cdot z^{-m} \\ k_2 \cdot z^{-m} \\ \vdots \\ k_{n/2} \cdot z^{-m} \end{bmatrix} + \dots + \begin{bmatrix} k_0 \cdot z^{-1} \\ k_1 \cdot z^{-1} \\ k_2 \cdot z^{-1} \\ \vdots \\ k_{n/2} \cdot z^{-1} \end{bmatrix} + \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ \vdots \\ k_{n/2} \end{bmatrix} \Rightarrow$$

$$\Rightarrow M = \begin{vmatrix} k_{11} & k_{12} & \cdots & k_{1(n-1)} & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2(n-1)} & k_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ k_{(m-1)1} & k_{(m-1)2} & \cdots & k_{(m-1)(n-1)} & k_{(m-1)n} \\ k_{m1} & k_{m2} & \cdots & k_{m(n-1)} & k_{mn} \end{vmatrix}$$

В конце преобразований три матрицы со спектральными данными накладываются по алгоритму 3.26, тем самым осуществляя перцептивное сжатие lossy.

Результатом наложения маски является спектр, который содержит основные форманты речи в текущем временном отрезке, определяющие выразительность и уникальность голоса.

## 4.2 Внедрение техники сжатия без потерь для ортогонального кодирования

Следующим этапом является процедура сжатия без потерь, в ходе которой происходят следующие действия: структуризация ортогонального потока, выбор техники сжатия, введение алфавита и словаря кодирования, формирование выходного потока.

Принцип сжатия без потерь для звука был заимствован из архиваторов [91] файлов, которые позволяют оптимизировать пространство хранения информации. В качестве минимальной единицы анализа в них выступают символы (Characters), в своей совокупности образующие алфавит, т.е. полный набор символов присущих данной системе, в который входят заглавные и прописные буквы, цифры, знаки пунктуации и служебные конструкции («перевод каретки», стирание и удаление, «поднять регистр» и т.д.).

В нашем исследовании определены два основных параметра для цифрового речевого сигнала – это его частота и амплитуда (громкость). Количество первых  $m$  определяется путём вычисления количества тонов  $t$ , содержащихся во всех выбранных критических диапазонах  $T$ :

$$m = \sum_{l=1}^L (T_l \cdot \sum_{i=1}^I t_i),$$

где  $i$  и  $l$  - индексы тона и диапазона,  $T$  и  $t$  принимают значение «0» или «1».

В рамках одного алфавита добавлены символы, определяющие параметры для ещё одного алфавита, содержащего определения для других свойств метода сжатия, в плюс к которому могут быть добавлены параметры для третьего и т.д. В текущей реализации структуры будет использовано три дерева.

Входными данными для организации алфавита (см. таблицы ниже) были определены частотные отсчёты из вычисленной в квантователе матрицы вида 4.1.

Таблица 1 - Построение дерева кодов для сжатия без потерь

Символ алфавита длин	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
Длина кода символа	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC
Код дерева	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN
№ литерала частоты						Размер кода из дерева размера							Код Хаффмана по дереву							
0 350 Гц						DEC							{BIN}							
1 450 Гц						DEC							{BIN}							
2 570 Гц						DEC							{BIN}							
3 700 Гц						DEC							{BIN}							
...						...							...							
...						...							...							
256 4000 Гц						DEC							{BIN}							
Символ алфавита амплитуд	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14					
Длина кода символа	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC					
Код дерева	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN					
Символ алфавита амплитуд	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29					
Длина кода символа	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC	DEC					
Код дерева	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN					

Здесь литералы – это частотные коэффициенты, длины кодов которых определены путем построения трёхуровневых кодовых деревьев. Ввиду того, что большинство частотных коэффициентов после наложения маски оказываются нулевыми, результирующий код будет иметь меньшую размерность.

Каждый ненулевой частотный коэффициент имеет дискретное значение амплитуды, которое также кодируется двухуровневым деревом. Этот процесс позволяет сократить кодовое значение минимум на 40%. Рисунок 4.4 иллюстрирует процесс построения основного дерева.

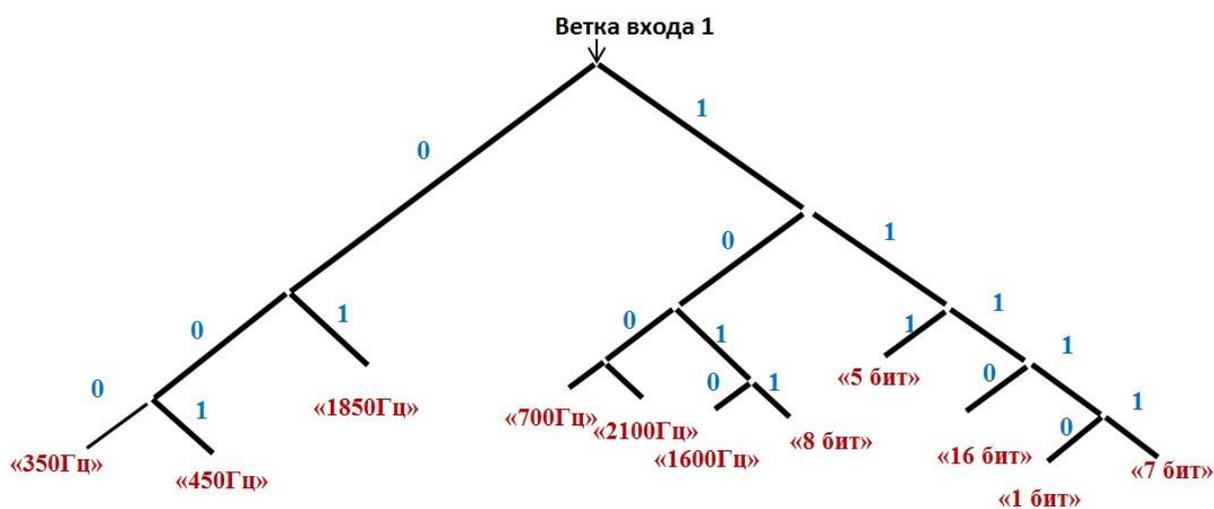


Рисунок 4.4 – кодовое дерево для алфавита, содержащего символы частот и длины кодов амплитуд

На следующем рисунке 4.5 представлено построение деревьев второго и третьего уровня. Второе содержит размеры кодов литералов и символьные дистанции до последнего повтора.

Если дистанция находится вне значений алфавита данного дерева, обращаемся к третьему дереву, где ветвление укажет нужный код для дистанции в основном дереве.

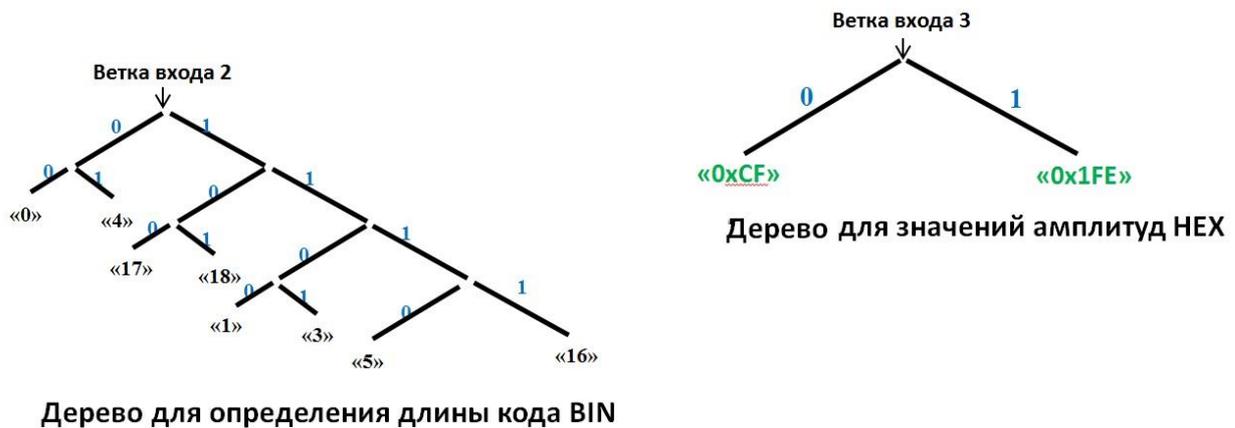


Рисунок 4.5 - Процедура построения дерева для кодов длин (слева) и дистанций до повторов (справа)

Вся процедура сжатия без потерь происходит посредством построения оговорённых трёх кодовых деревьев, каждое из которых определяет последующий код символов литералов. Сжатию подвержена также и размерность кода для литералов частотных отсчётов их амплитуд. С целью более компактного размещения кодов словаря, неиспользуемые литералы и дистанции не входят в систему деревьев. Выходной кадр состоит из двух частей: словаря и кодовых значений деревьев для частот и амплитуд.

### 4.3 Формирование выходного кадра и декодирование

Проанализированные стандартизированные перцептивные кодеки обладают минимальной частотой дискретизации 22,5 КГц [7]. Этот показатель явно избыточен в случае их применения для кодирования речи человека.

При кодировании канала тональной частоты было уменьшено количество полифазных фильтров. За счёт этого снижено количество потоков на выходе блока, производящего адаптированное дискретное косинусное преобразование.

Приемлемо использовать частоту дискретизации равную 8000 Гц, которая рекомендована для речевого сигнала.

В случае обработки монофонического сигнала с постоянным битрейтом и параметрами кодера можно полностью отказаться от блока кодирования информации о состоянии канала. Блок формирования выходного битового потока необходимо модифицировать для расстановки стартовых и стоповых последовательностей каждого фрейма и не учитывать синхронизирующую служебную информацию.

Так, на вход первого блока декодера будет поступать поток битов, который в стандартах ITU-T состоит из следующих частей: заголовок, сторонняя информация, главные данные, вспомогательные данные [69,115,127]. Структура кадра выходного потока мультиплексируется и поступает в линию передачи в виде 4.6 [10]:

Заголовок	Сторонняя информация	Главные данные	Вспомогательные данные
-----------	----------------------	----------------	------------------------

Рисунок 4.6 — Структура кадра выходного потока

В данном блоке также происходит идентификация и синхронизация потока битов, первый процесс способствует тому, что в последующих блоках полученная информация будет правильно интерпретирована, а второй процесс в данном случае заключается в поиске положения начала первого и последующих заголовков.

Затем следуют блоки «расшифровка сторонней информации», «определение частотных коэффициентов», «расшифровка данных кодовых деревьев». В блоке «определение частотных коэффициентов» происходит процесс выделения частотных полос и декодирование первой части блока битов, который содержит в себе стороннюю информацию.

В блоке «повторное квантование» происходит инверсное квантование и упорядочение спектра, а также определение масштаба частотных коэффициентов, что способствует восстановлению частотной линии.

В блоке «переупорядочение спектра» будет происходить процесс поиска коротких окон (оконных функций) в каждой из 14 подгрупп, и в том случае, если они будут найдены, то данный блок произведёт их переупорядочение в бит-реверсивном порядке. Далее каждый из них проходит процедуру обратного ортогонального преобразования, где частотные элементы при непосредственном участии оконных функций преобразуются в полифазный узкополосный временной поток. Данный процесс протекает параллельно для всех критичных диапазонов.

В завершающем блоке происходит синтез полифазного банка фильтров, который преобразовывает 14 групп в 1 кадр временного интервала.

На следующем рисунке 4.7 проиллюстрирован полный кадр из речевого потока данных. Идентификатор показывает порядок следования бит, затем следует номер версии исполнения или семейства перцептивного кодека, следующей структурой являются защитные биты контрольной суммы (CRC), длина которых составляет 16 бит [120,128].

16 бит	16 бит	16 бит	16 бит
Идентификатор		Синхронизация	
Слой		Защитные биты	
Скорость передачи			
Частота		Основные биты	Дополнительные биты
Метод		Расширенный метод	
Копия	Размещение	Внесение дополнительных данных	
Аудиоданные			

Рисунок 4.7 — Полная структура стандартизированного ITU-I фрейма [101]

Завершающими структурами являются типы копирования данных, размещение, а также внесение дополнительных данных или так называемой выразительности.

Организация битов в кадре потока представлена в таблице 4.1 [77,125].

Таблица 4.1 — Организация битов в полной структуре кадра

№ п/п	Положение бит в заголовке	Число бит	Определение
1	1-12	12	Синхронизация
2	13	1	Идентификатор (ID)
3	14-15	2	Обозначение слоя
4	16	1	Защитные биты
5	17-20	4	Битрейт
6	21-22	2	Частота: для MP3 32, 44,1 48 кГц.
7	23	1	Дополнительные биты
8	24	1	Основные биты
9	25-26	2	Метод
10	27-28	2	Расширенный метод
11	29	1	Авторское право: указывается защищено или нет
12	30	1	Копия: указывает является ли файл копией или оригиналом
13	31-32	2	Акцент (выразительность)

Структурная схема декодера коррелирована со многими известными перцептивными кодеками [38], поэтому условимся о том, что процедура распаковки обратна описанным выше процедурам.



Рисунок 4.8 — Блочная диаграмма ортогонального декодера

Возможность использования стандартизированного декодера ITU-I для разработанной модели кодера обоснована ввиду того, что структура изменённого файла подобна структуре ITU-I формата и будет безошибочно декодирована в ИКМ формат.

#### 4.4 Результирующая структурная схема кодера и предварительная оценка сжатия

Итогом структурного проектирования кодера является схема его функционального взаимодействия, представленная на следующей иллюстрации.

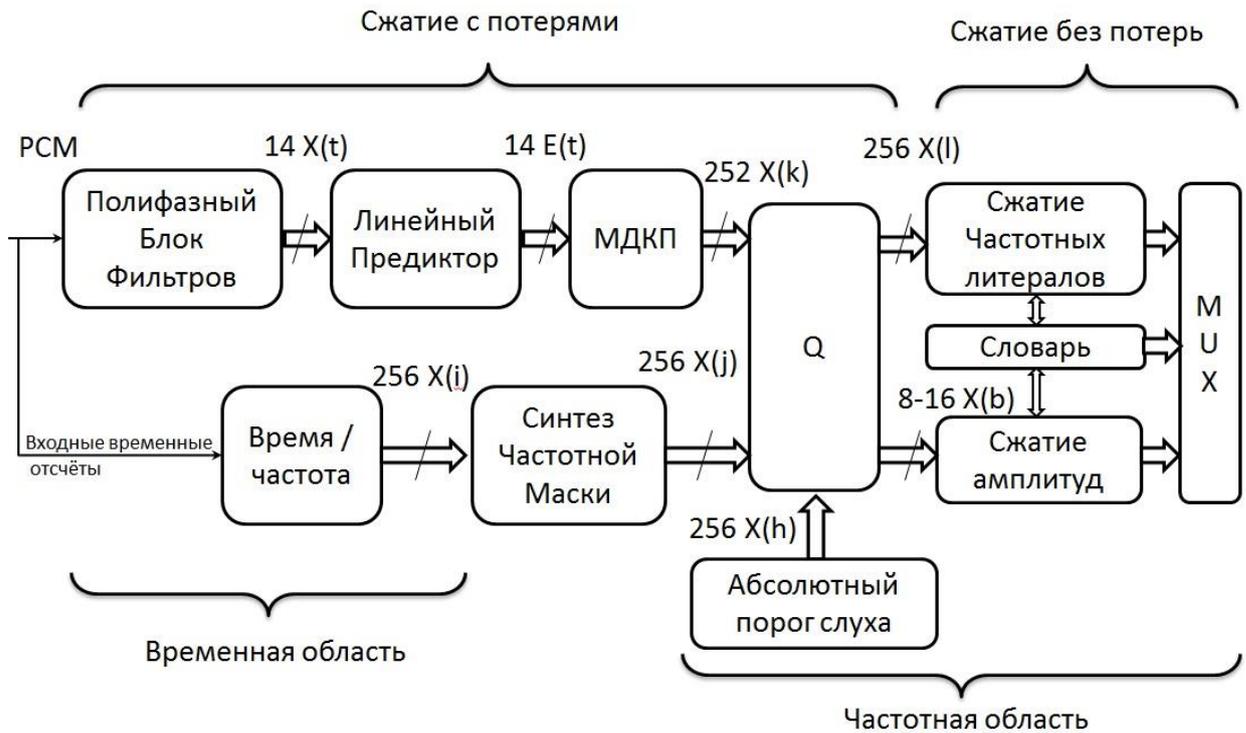


Рисунок 4.9 – Вариативная структура синтезированного кодера,  $X(t)$  – Временные массивы данных,  $E(t)$  – Предсказанные массивы,  $X(i)$ ,  $X(j)$ ,  $X(k)$ ,  $X(h)$ ,  $X(l)$ ,  $X(b)$  – частотные массивы

В синтезированной функциональной модели поток входных оцифрованных данных попадает параллельно на модифицированный полифазный фильтр, имеющий динамическую структуру, в нашем случае – 14 полосовых фильтров, и на блок БПФ от 256 отсчётов. В блоке «линейного предиктора» происходит наложение временной маски. Модифицированное дискретное косинусное преобразование производит ортогональное преобразование «время-частота». Квантователь обеспечивает взаимодействие трёх спектров: модели абсолютного порога слуха, психоакустической маски, вычисленной из БПФ и входных частотных значений. Далее параллельный поток подвергается сжатию без потерь.

Вариативными блоками в кодере определены: банк полифазных фильтров (с динамикой выбора сглаживающего окна и количества полос); линейный предиктор (с различной

длительностью выборки и функцией временного маскирования); функционал квантователя (с изменяемой глубиной частотного маскирования); блок сжатия без потерь (с выбором техники сжатия между статическим и динамическим словарём, а также методом скользящего окна).

Эффективность применения линейного предиктора экспериментально показана в таблице ниже.

Таблица 4.2 Соотношение длины маски, разрядности АЦП и эффективности сжатия

Разрядность кода	Динамический диапазон, дБ	Максимальная длина, мс	Компрессия, бит удалено		
			8КГц	22,05КГц	44,1КГц
8 бит	24	40	320	882	1764
12 бит	36	60	480	1323	2646
16 бит	48	100	800	2205	4410
24 бит	72	>160	1280	3308	6615

Сжатие путём наложения частотной маски позволило осуществить компрессию с соотношением до 10:1 для речевого сигнала, что также доказано работоспособностью ортогональных кодеров MPEG.

Соответствие процентного отношения сжатых данных по технике сжатия без потерь с алгоритмом Deflate показано в таблице ниже.

Таблица 4.3 Эффективность сжатия без потерь в различных условиях

Частота дискретизации, Гц	Код, бит	Эффективность сжатия, %		
		Статический Хаффман	Динамический Хаффман	Скользящее окно Лемпеля–Зива
8 000	8	99%	127%	96 %
	12	99%	110%	94%
	16	98%	107%	90%
	24	97%	106%	84%
22 500	8	92%	106%	95%
	12	90%	101%	92%
	16	89%	95%	91%
	24	88%	89%	90%
44 100	8	90%	89%	95%
	12	90%	87%	92%
	16	90%	85%	91%
	24	90%	85%	90%

Спектральные искажения при работе кодека определены сопоставлением БПФ для сэмплов исходного и переданного потоков. В случае, если спектральная плотность мощности

шума не превышает граничное значение, заданное проектировщиком, оговаривается приемлемость разборчивости речевого сигнала.

#### 4.5 Программное обеспечение имплементации речевого кодера

Для проверки спроектированных структур и методик сжатия речевого потока реализовано программное обеспечение, которое позволяет произвести анализ выходного звука на разборчивость и выразительность. Внешний облик программы представлен на рисунке 4.10.

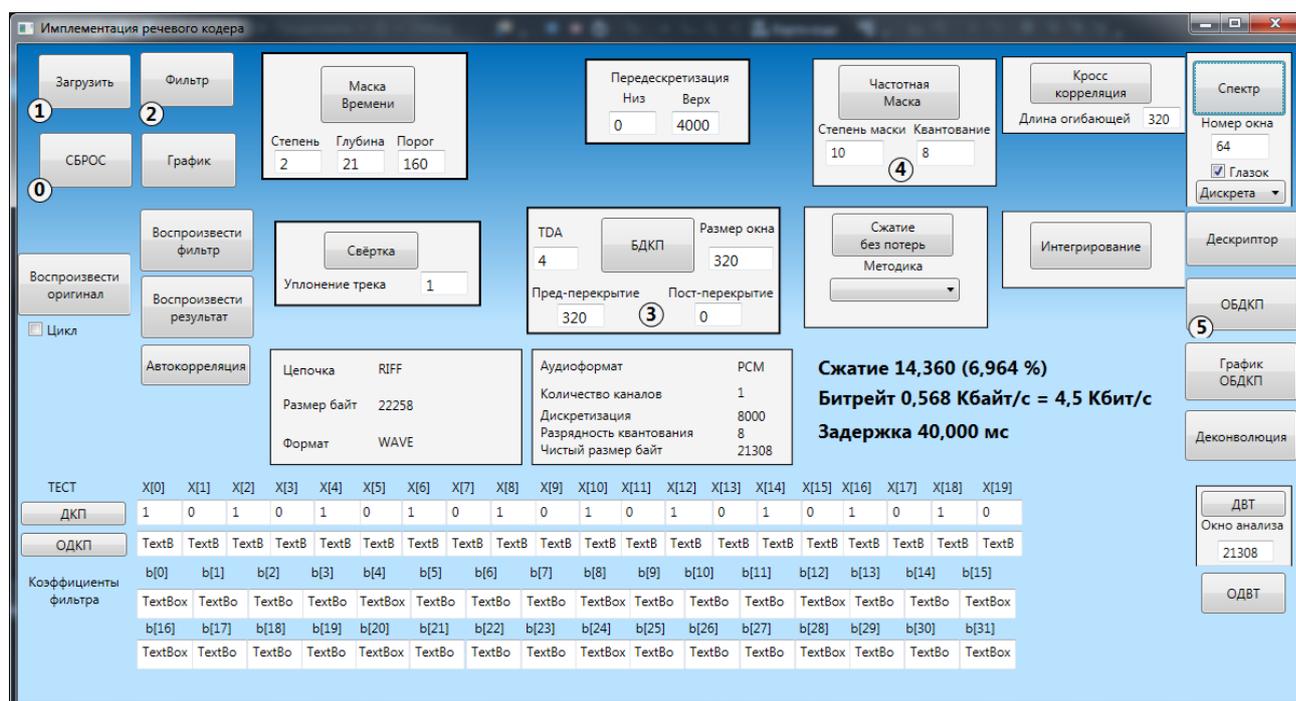


Рисунок 4.10 – Имплементация речевого кодера

В функционал программы входят возможность звукового воспроизведения входного речевого сигнала в формате Windows Media Audio с частотой дискретизации от 8 КГц до 44,1 КГц и отображение его параметров и графика (см. п. 3.1).

Как было показано в п. 2.1, при помощи наборного поля входных дискрет  $X[0] - X[19]$  и коэффициентов фильтра  $b[0] - b[16]$  возможно провести анализ функции фильтрации и дискретного косинусного преобразования с использованием зеркального сигнала и синусного окна.

Поле «БДКП» и «Маскирование» позволяют осуществить ортогональное преобразование и наложение масок с выбранными параметрами перекрытия, весовым коэффициентом TDA

(Time-domain aliasing – временного спектрального наложения), степенью маски и коэффициентом переквантования, а также отобразить результат:

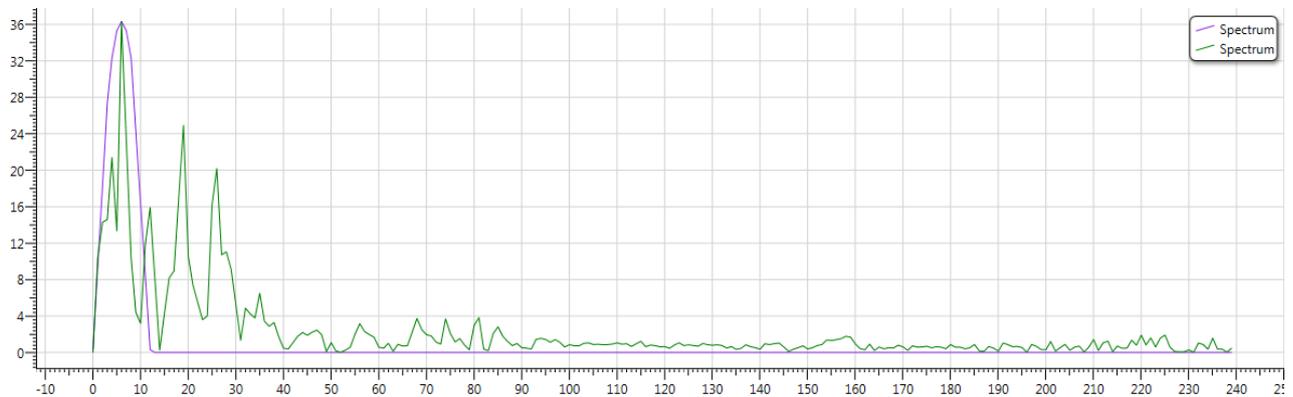


Рисунок 4.11 – Спектр анализируемого (-) сигнала и наложенная на него маска (-)

Основной процесс прореживания (сжатия с потерями) сигнала происходит в спектральной плоскости. На рисунке 4.12 изображены глазковые диаграммы спектров речи до и после масштабирования:

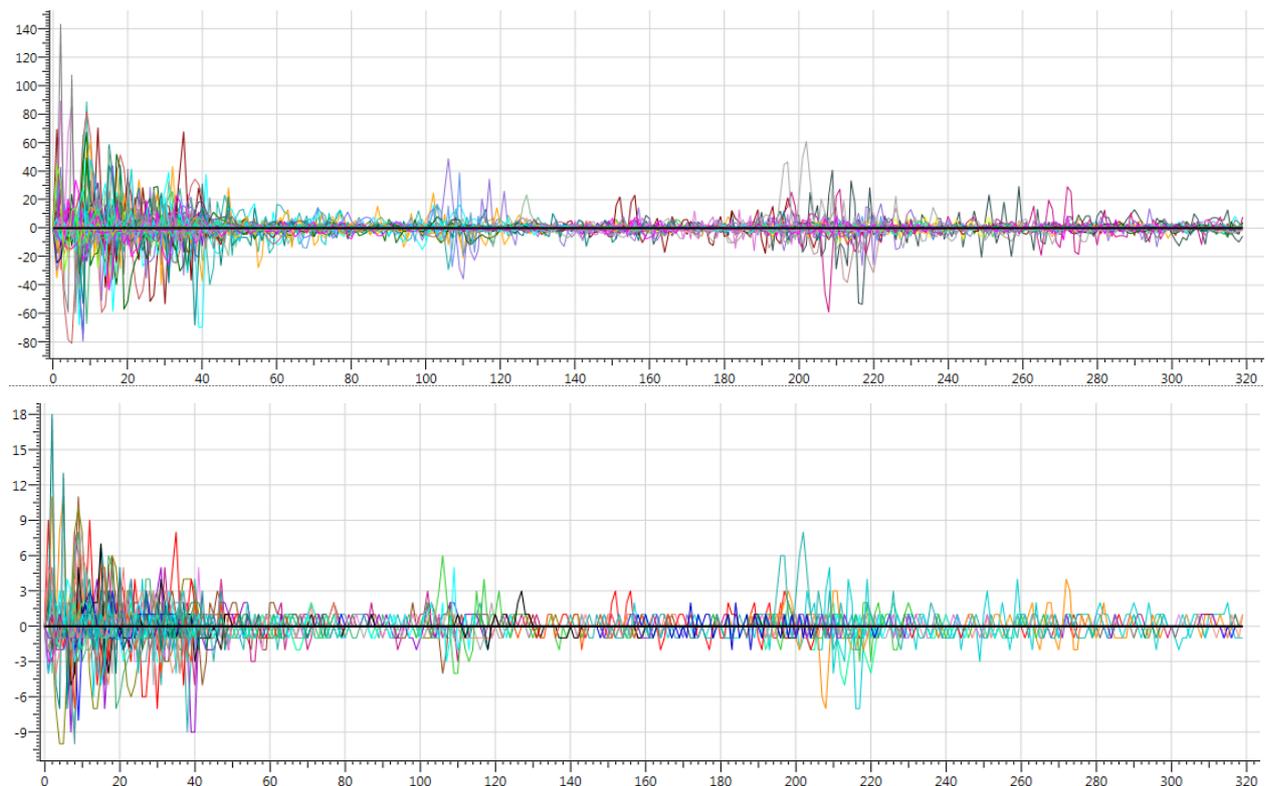


Рисунок 4.12 – Сравнение спектров речи до и после масштабирования

Кнопки «ОБДКП» и «График ОБДКП» являются функциями декодирования и предоставляют возможность провести обратное ортогональное преобразование и оценить визуально результат на графике, содержащем временное представление сигнала:

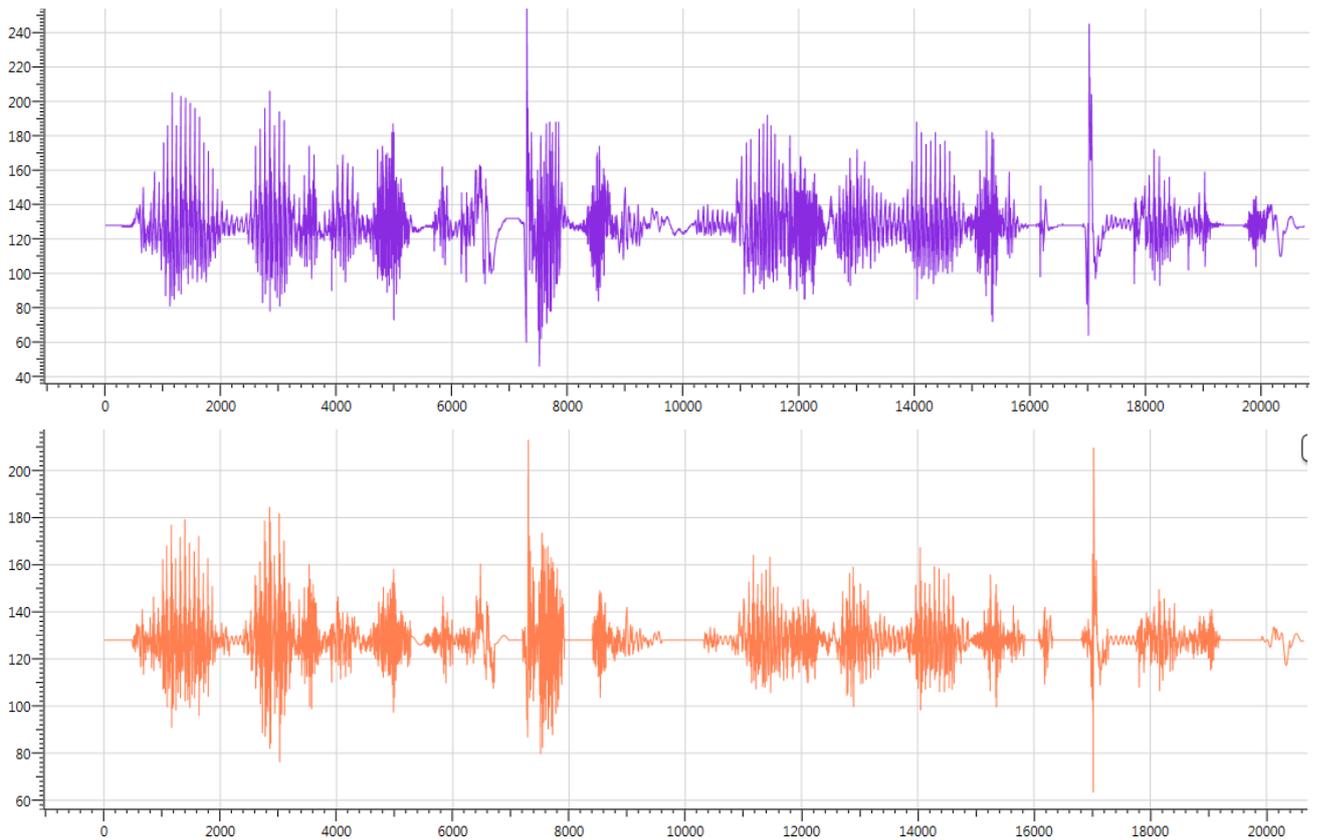


Рисунок 4.13 – Сравнение сигнала до сжатия (-) и сигнала после декодирования (-)

Программное обеспечение написано на языке C# с использованием классов WPF, входящих в состав .Net. Исходный код приложения представлен в ПРИЛОЖЕНИИ А.

#### 4.6 Оценка временных задержек сигнала в линии связи

Определение временной сложности цифровых преобразований состоит из этапов анализа времени преобразований ортогональных блоков и цифровой фильтрации.

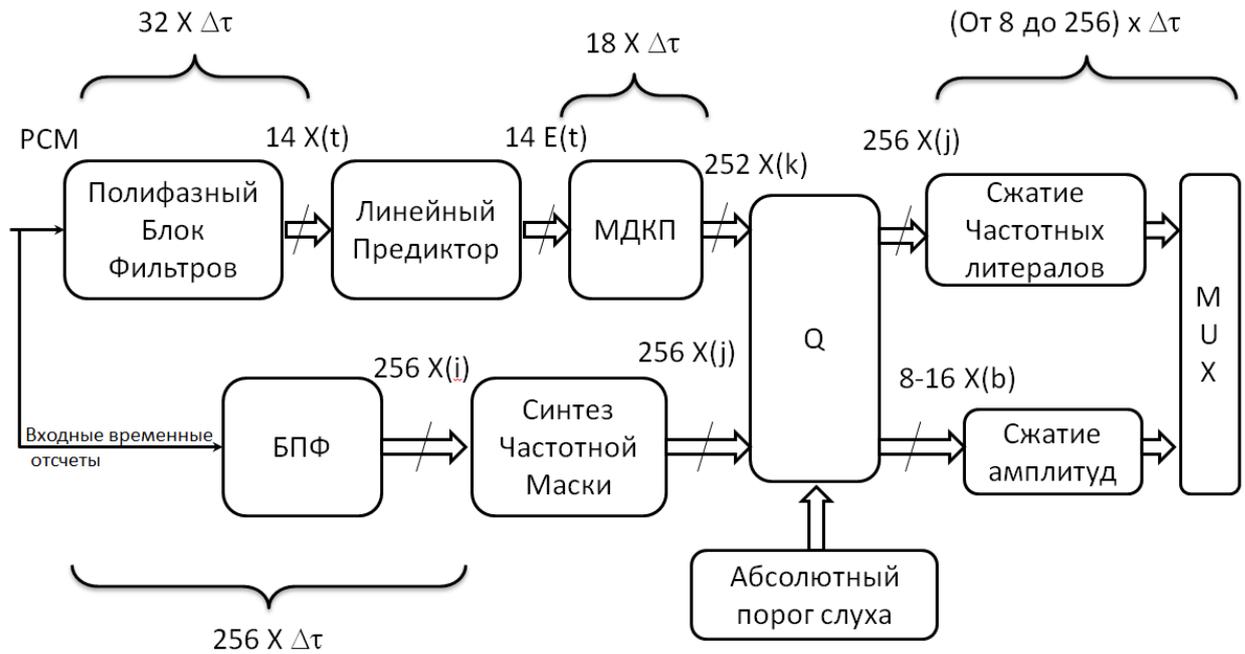


Рисунок 4.14 – Синтезированная структура ортогонального кодирования

При распространении сигнала в данной схеме каждый блок вносит задержку равную  $\tau$ . При поступлении сигнала на вход схемы происходит накопление данного сигнала в 14 фильтрах, которые находятся в блоке цифровых полосовых фильтров, где вносимая задержка не оказывает существенного воздействия на сигнал, по этой причине мы не будем здесь ее рассматривать.

В блоке БПФ происходит накопление 256 отсчётов, следовательно, для их накопления необходимо некоторое время. Для того чтобы установить связь между значением отсчётов и их эквивалентному значению во времени необходимо произвести расчёт того, какой временной интервал эквивалентен одному отсчёту. Так как одна секунда содержит 8000 отсчётов, то можно сделать вывод о том, что 1 отсчёт равен 125 мкс,  $1/8000 = 125$  мкс, тогда для накопления 256 отсчётов необходимо  $125 \text{ мкс} \times 256 = 32 \text{ мс}$ , таким образом, мы определили, что задержка сигнала для блока БПФ составляет 32 мс.

В блоке МДКП происходит накопление 18 отсчётов, а значит для их накопления необходимо  $125 \text{ мкс} \times 18 = 2,25 \text{ мс}$ . Все последующие блоки вносят примерно те же временные задержки, в силу компрессии схожего потока данных, поэтому их берём в расчёт как эквивалентные.

Из приведённых выше данных, можно сделать вывод о том, что вносимая блоком МДКП задержка не так существенна, как вносимая блоком БПФ, где происходит накопление 256

отсчётов, а временная задержка составляет 32 мс, здесь необходимо разобраться, является ли данная задержка приемлемой для речевого сигнала.

Суммарная задержка кодера составляет 64 мс, а для декодера она составляет 1,75 мс (блок ОМДКП), поэтому можно сказать, что общая задержка сигнала для кодера и декодера составляет 65,75 мс, а как было сказано выше, нормальная слуховая система способна различить временные интервалы в 200 мс. Данная задержка временного сигнала будет приемлема, так как она не повлияет на комфортность общения в дуплексном режиме.

#### **4.7 Выводы по четвёртой главе**

Создание условий для математической реализации отдельных структурных блоков цифрового взаимодействия позволило синтезировать средство кодирования и сжатия на параметрическом уровне, что означает построение структурно-параметрического аппарата для исследования свойств речевого сигнала во время его перцептивной обработки.

Результатами работы по текущей главе представлены:

- построенное на основе математических и программных комплексов средство речевого кодирования, состоящее из блоков сжатия с потерями и без них;
- определены форматы данных, протокол и кадры для передачи, обозначены особенности декодирования;
- проведена предварительная оценка степени компрессии при использовании различных вариаций структурных блоков сжатия;
- оценены временные задержки для синтезированного варианта из семейства кодеров в каждом блоке преобразований, установлена приемлемость полученных результатов для дуплексной линии связи.

## Заключение

Основным результатом работы является многофункциональный инструмент цифровой обработки речевого сигнала с изменяемым уровнем сжатия и битрейтом, а также с функциями оценки задержки, который является базой для построения семейства кодеров с параметрами, необходимыми для внедрения в системы хранения и дуплексного обмена речевой информацией реального времени. По результатам исследований можно сделать следующие выводы:

- анализ техник речевого кодирования показал основные преимущества ортогонального и линейного метода компрессии, в результате было выбрано гибридное решение для синтеза структуры кодера. Процедуры были разделены на две части: блок сжатия с потерями и без. Блок сжатия с потерями, в свою очередь, разделён на линейные и ортогональные преобразования;

- в процессе линейной цифровой обработки были синтезированы математические модели блока полифазной фильтрации и линейного предиктора, построены алгоритмы их реализации. Выявлено, что данные блоки позволяют управлять критериями разборчивости и выразительности речи;

- проведена разработка модели и алгоритма наложения психоакустической маски – основного ортогонального блока, реализующего сжатие с потерями. Данный блок позволил варьировать битрейт на основе метода удаления неслышимых частотных компонентов, сжатие – до 4 раз;

- структурирована система сжатия без потерь по словарным и оконным методам, основной задачей которой является компрессия частотных выборок. Данная процедура в совокупности с предыдущим преобразованием позволила сжать поток до 14 раз;

- реализована программная имплементация речевого кодера, позволяющая применить методы и алгоритмы диссертационного исследования к записанному голосовому потоку, вычислить степень сжатия и задержку и оценить качество воспроизведения речи.

Полученные результаты соответствуют п. 4 «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации», п. 7 «Методы и алгоритмы структурно-параметрического синтеза и идентификации сложных систем» паспорта специальности 05.13.01 «Системный анализ, управление и обработка информации (технические системы)».

В развитие итогов исследования перспективными задачами представляются: нахождение составных базовых функций ортогональных преобразований, содержащих универсальные энергетические сочетания голоса диктора; исследование путей автоматического выбора

процедур сжатия и их параметров при изменении условий передачи сигнала; создание голосовых лексикографических словарей, увеличивающих плотность сжатия.

## Список сокращений и условных обозначений

АБПФ	- Адаптированное быстрое преобразование Фурье
АДИКМ	- Адаптивная дифференцированная импульсно-кодовая модуляция
АДКП	- Адаптированное косинусное преобразование.
АЧХ	- Амплитудно-частотная характеристика
БПФ	- Быстрое Преобразование Фурье.
ДКП	- Дискретное косинусное преобразование.
ИКМ	- Импульсно – кодовая модуляция.
КИХ	- Конечная импульсная характеристика.
МДКП	- Модифицированное дискретное косинус – преобразование.
ОЦК	- Основной цифровой канал.
УЗД	- Уровень звукового давления.
ФЧХ	Фазочастотная характеристика
ЦТВ	- Цифровое телевидение.
CRC	- (cyclic redundancy check) Циклический избыточный код.
DIT	- (decimation – in – time) – 8 – точечное ДПФ с прореживанием по времени.
IS	- (International standard) – Международный стандарт.
ISO	- (International Organization for Standardization) Международная Организация по Стандартизации.
MDCT	- (Modified discrete cosine transform) модифицированное дискретное косинусное преобразование – МДКП.
MPEG	- Motion Picture Expert Group (экспертная группа по вопросам движущегося изображения).
MP3	- (MPEG Audio Layer III) – Третий слой формата кодирования звуковой дорожки MPEG.
ITU-I	- (International Telecommunication Union) - Международный союз электросвязи

## Список терминов

**Алфавит:** в техниках сжатия без потерь, а также в двоичном кодировании текстовой информации означает совокупность всех символов, применяемых для передачи информации. Чаще всего алфавит построен иерархично и определяет лексикографию.

**Битрейт:** усреднённая скорость потока данных, вычисляемая в канале передачи данных. Является качественной мерой степени сжатия информации при сравнении различных кодеков (единица измерения бит(байт) в секунду).

**Вариативность** речевого кодера: способность средства сжатия адаптироваться условиям передачи цифрового сигнала (битрейту, задержке, качеству и т.д.), при этом максимально возможно сохраняя заданные оператором параметры.

**Качество** речи: параметр оценки речевого сигнала при его воспроизведении, показывающий степень узнаваемости голоса диктора.

**Лексикографический** способ построения кодовых деревьев в техниках сжатия без потерь информации: представляет собой вариант организации и распределения символов алфавита согласно первичности их расположения в нём при равенстве длины кода.

**Линейный предиктор:** автомат математического предсказания будущих во времени элементов происходящий путём многоуровневого анализа предыдущих элементов, обладающий разнородной обратной связью и хранилищем.

**Нестационарность** речевого сигнала: определение сложной для математического описания, предсказания и анализа функции, представляющей оцифрованную речь диктора.

**Перцептивное** кодирование: процесс преобразование звуковой информации, при котором происходит удаление не воспринимаемых слухом человека компонентов звука, за счёт чего, в основном, происходит сжатие с потерями.

**Полиграфический анализ:** средство исследования текстовой информации, основанное на сопоставлении слов и выявлении повторов как самих слов, так наборов символов. Является основным методом при сжатии информации без потерь.

**Психоакустика:** область науки, в рамках которой происходят исследования восприятия, влияния и понимания звуковых сигналов при участии слуха человека. Является фундаментальной сферой, результаты опытов в которой становятся базой для построения средств перцептивного кодирования.

**Разборчивость речи:** параметр оценки речевого сигнала при его воспроизведении, определяющий возможность беспрепятственного понимания всех слов и сути передаваемой диктором информации.

**Редуцированные звуки:** в лингвистике непронизносимые составляющие слов, находящиеся в безударных слогах, но занимающие определённое время на произношение, по-другому – невнятные звуки речи.

**Символ:** минимальная единица передаваемой между двумя средствами связи информации, уникальная и имеющая одинаковую битовую длину в рамках одного алфавита.

**Скользящее окно:** поле анализа (буфер) набора символов при потоковом сжатии, которое обеспечивает сравнение своего содержимого на предмет повторов. Смещается пошагово на единицу символа.

**Степень сжатия речевого потока:** относительная величина, показывающая процентное соотношение битрейта кодера к несжатому потоку звука ИКМ-формата.

## Список литературы

1. **Авдеев, Л.В.** Математическая модель восприятия звукорядов. / Л. В. Авдеев, П. Б. Иванов Препринт Р5-90-4 Объединенного Института Ядерных Исследований — Дубна, 1990
2. **Аграновский А.В., Леднов Д.А.** Теоретические аспекты алгоритмов обработки и классификации речевых сигналов. / А.В. Аграновский, Д.А. Леднов. - Москва: Изд-во «Радио и связь», 2004. 164 с.
3. **Айфичер, Э., Джервис, Б.** Цифровая обработка сигналов. Практический подход. / М., Вильямс, 2004 . - 992 с.
4. **Алаябьев, С.И.** Радиовещание и электроакустика: учебник для вузов / С. И. Алаябьев, А. В. Выходец, Р. Гермер ; ред. Ю. А. Ковалгин. - Москва: Радио и связь, 1998.
5. **Алдошина, И. А.** Основы психоакустики, часть 9: слуховые пороги, часть 2 // Звукорежиссёр. – 2003. - №10.
6. **Алдошина, И.** Основы психоакустики (подборка статей с сайта <http://www.625-net.ru>) / Ирина Алдошина. – Режим доступа: <http://www.keklab.ru/buf/ai/psychoacoustics.pdf>
7. **Ануфриев, А.** Кодеры и декодеры MPEG // Информационно-технический журнал 625. – изд. ООО «Издательство 625», №7, 2003 – с.
8. **Артюшенко, В.М., Шелухин, О.И.** Цифровое сжатие видеоинформации и звука. Учебное пособие. / Под ред. В. М. Артюшенко. — М.: Издательско-торговая корпорация «Дашков и К», 2003. — 426 с. ISBN 5-74978-258-7
9. **Бахвалов, И. В.** Численные методы / И.В. Бахвалов. – М: . БИНОМ., 2008. — 636 с.
10. **Белугин, И.Н., Шиленков, Е.А.** Повышение структурной устойчивости речевого ортогонального кодера / И.Н. Белугин, Е.А. Шиленков // Материалы Всероссийской научно-практической конференции «Инфокоммуникации и информационная безопасность: состояние, проблемы и пути решения», Курск, 25-26 апреля 2014. – С. 232 - 236.
11. **Беранек, Л.** Акустические измерения / Л. Беранек.; Пер. с англ., М., 1952
12. **Бибииков, Н.Г.** Оценки параметров периферических нейронных элементов // в сб. «Акустика речи и слуха» Л., Наука, 1986, С.25-55.
13. **Блейхут Р.** Теория и практика кодов, контролирующих ошибки / пер. Зигангирова К. Ш. - Москва: Мир, 1986. – 576 с.
14. **Блейхут, Р.** Быстрые алгоритмы цифровой обработки сигналов. / Р. Блейхут. - М.: Мир, 1989. — 448 с.: ил.

15. **Богнер, Р., Константи́нидис, А.** Введение в цифровую фильтрацию. / Р. Богнер, А. Константи́нидис. Пер. с англ. - М.: Мир, 1976. - 216 с. - Режим доступа: <http://www.dsplib.ru/content/filters/firwin/firwin.html>
16. **Борискевич, А. А., Холев, О. В.** Цифровая обработка речи и изображений [текст] // А.А. Борискевич А. А., О.В. Холев, В.Ю. Цветков. - Мн.: БГУИР, 2004. – 183 с.
17. **Борискевич, А.А.** Цифровая обработка речи и изображений. / А.А. Борискевич: Минск, 2003. – 383с.
18. **Борискевич, И.А.** Алгоритм интеграции изображений в вейвлет-области для медицинских систем визуализации / И.А. Борискевич // Медэлектроника-2010. Средства медицинской электроники и новые медицинские технологии : Сборник научных статей. - Минск, 2010. С. 39-41.
19. **Вадутов, О.С.** Математические основы обработки сигналов: учебное пособие. / О.С. Вадутов. - Томск: Изд. ТПУ, 2011. - с. 212
20. **Валгина, Н.С.** Современный русский язык: Учебник / Валгина Н.С., Розенталь Д.Э., Фомина М.И. // Под редакцией Н.С. Валгиной. - 6-е изд., перераб. и доп. – Москва: Логос, 2002. – 528 с.
21. **Вартанян, И. А.** Звук — слух — мозг. / И.А. Вартанян. - Л.: Наука, 1981.
22. **Вахитов, Ш.** Акустика. / Шакир Вахитов, Юрий Ковалгин, Александр Фадеев, Юрий Шевьев. - Специальность. Для высших учебных заведений. – М: Горячая Линия – Телеком. – 2009. 664 с. ISBN 978-5-9912-0093-6
23. **Виноградов, А.** Психоакустика на службе у людей / А. Виноградов. – Режим доступа: <http://www.hifinews.ru/article/details/6072.htm>.
24. **Винцюк, Т.К.** Анализ, распознавание и интерпретация речевых сигналов / Т.К. Винцюк. - Киев: Наук. думка, 1987. – 264 с.
25. **Войнаровский, М.** БПФ, вывод основных формул. / М. Войнаровский. – Режим доступа: <http://psi-logic.narod.ru/fft/fft2.htm>
26. **Гельфанд, С. А.** Слух: введение в психологическую и физиологическую акустику / С.А. Гельфанд.; Пер. с англ. — М.: Медицина, 1984
27. **Гибсон, Джерри Д.** Digital Compression for Multimedia. / Джерри Д. Гибсон, Тони Бергер, Том Лукэбог — Morgan Kaufmann, 1998. — ISBN 978-1-55860-369-1.1
28. **ГОСТ Р 53801-2010.** Связь федеральная. - М.: Стандартинформ, 2011. – 23 с.
29. **Двайт, Р.** Расширения для сжатия и архивации (The Unarchiver provides). / Рон Двайт, Александр Рошал, Дмитрий Ерохин. - win. rar GmbH., - 2012
30. **Ермилов, А. В.** Математическое обеспечение вычислительных машин, комплексов и компьютерных сетей. [Текст]: автореф. дис. на соиск. учен. степ. канд. техн. наук.

2014 / Ермилов Алексей Валерьевич – 2014 НИУ «Высшая Школа Экономики» - Москва, 2014. – 24 с.

31. **Ефимов, А.П.** Акустика: Справочник / А. П. Ефимов, А. В. Ноконов, М. А. Сапожников, В. И. Шоров // Под ред. М. А. Сапожникова. 2-е изд., перераб. и доп. М.: Радио и связь, 1989.- 336 с.

32. **Ефимова, О.В., Шиленков, Е.А.** Банк фильтров ортогонального речевого кодирования / О.В. Ефимова, Е.А. Шиленков // Материалы Всероссийской научно-практической конференции «Инфо-коммуникации и информационная безопасность: состояние, проблемы и пути решения», Курск, 25-26 апреля 2014. – С. 113 - 118.

33. **Зигангиров, К. Ш.** Процедуры последовательного декодирования. / К.Ш. Зиганкиров. - М.: Связь, 1974.- 208 с.

34. **Злотник, Б. М.** Помехоустойчивые коды в системах связи. / Б.М. Злотник. - М.: Радио и связь, 1989.- 232 с.

35. **Казимилова Е.А.** Речевой сигнал как отражение изменений функционального состояния при депрессивном и тревожно-депрессивном расстройствах. [Текст]: автореф. дис. на соиск. учен. степ. канд. биол. наук. 2016 / Казимилова Евдокия Алексеевна; ИВНДН РАН. – Москва, 2016. – 24 с.

36. **Канаков, В.А., Миронов, Н.А.** Выделение речевого сообщения из помех от распределенных в пространстве источников // Труды XIX научной конференции по радиофизике, посвященной 70-летию радиофизического факультета. – Н.Новгород, 2015. – С. 141-142

37. **Карпухин, В.А.** Применение цифровой фильтрации в медико – биологической практике. Общие понятия цифровой фильтрации / В.А. Карпухин – М: Издательство МГТУ им. Баумана, 2007. - Режим доступа: <http://do.gendocs.ru/docs/index-214103.html>

38. **Квинт, И.** Sound Forge 9. / Игорь Квинт. – СПб: Питер., 2009. – 167 с. ISBN: 978-5-388-00378-2

39. **Кестер, У.** Аналого-цифровые преобразователи для задач цифровой обработки сигналов. / Уолт Кестер. - М: Техносфера - 39 с. ISBN: 978-5-94836-146-8

40. **Кипяткова, И.С., Ронжин, А.Л.** Автоматическая обработка разговорной русской речи / И.С. Кипяткова, А.Л. Ронжин, А.А. Карпов. - СПб.: ГУАП, 2013. – 314 с.

41. **Киркоров С. И.** Цифровая обработка речи и изображений. / С.И. Киркоров. – Минск: Высший государственный колледж связи. 2010.

42. **Кларк Д., Кейн Д.** Кодирование с исправлением ошибок в системах цифровой связи. / Д. Кларк, Д. Кейн. - Москва: Радио и связь, 1987.- 392 с.

43. **Котельников, В. А.** О пропускной способности эфира и проволоки в электросвязи — Всесоюзный энергетический комитет. // Материалы к I Всесоюзному съезду по вопросам технической реконструкции дела связи и развития слаботочной промышленности, 1933. Репринт статьи в журнале УФН, 176:7 (2006), 762—7701
44. **Крухмалев, В.В., Гордиенко, В.Н.** Основы построения телекоммуникационных систем и сетей. / Крухмалев В.В., Гордиенко В.Н., Моченов А.Д., Иванов В.И. и др. М: Горячая линия – Телеком. – 2006. 750 с. ISBN: 5-93517-202-X
45. **Куликовский, Л.Ф. Мотов, В.В.** Теоретические основы информационных процессов. / Л.Ф. Куликовский, В.В. Мотов. - Москва: Высшая школа 1987. – 241с.
46. **Куликовский, Л.Ф.** Теоретические основы информационных процессов. / Л.Ф. Куликовский. - Москва: Высшая школа, 1987. – 247с.
47. **Куприянов, М. С. Матюшкин, Б. Д.** Цифровая обработка сигналов. / М.С. Куприянов. Б.Д. матюшкин.— 2-е изд. — СПб: Политехника, 2000. — 592 с.
48. **Лазарев Ю.** Моделирование процессов и систем в MATLAB. / Ю. Лазарев. Учебный курс. – СПб.: Питер; Киев: Издательская группа BHV, 2005. – 512 с. ISBN 5-469-00600-X. ISBN 966-552-144-6
49. **Лебедева Н.Н., Каримова Е.Д.,** Анализ речевого сигнала в исследованиях функционального состояния человека / Н.Н. Лебедева, Е.Д. Каримова. Е.А. Казиминова // Биомедицинская радиоэлектроника. – 2015. – № 2. – С. 3-11.
50. **Линдсей, П., Норман, Д.** Переработка информации у человека / П. Линдсей, Д. Норман — М: Мир, 1974
51. **Лукманова, О.Р.** Математическое моделирование акустоэлектрического канала утечки речевой информации при использовании пассивного средства защиты. // МИЭТ, - М.; Зеленоград, 20 - 22 апреля 2016 г.
52. **Миронов, М.А.** Два акустических эффекта в окружающей среде / М.А. Миронов // Акустика среды обитания – 2016. - № 1. – С. 5 -13.
53. **Морозов, В.П.** Восприятие речи: Вопросы функциональной асимметрии мозга. / В. П. Морозов, И. А. Вартанян, В. И. Галунов и др.- Ленинград.: Наука, 1988. -135 с.
54. **Мухин И.Е., Шиленков Е.А.** Метод параметрического синтеза антенно-фидерных, радиоприёмных и демодуляторных средств сигнально-приёмного тракта современных систем телекоммуникаций / И.Е. Мухин И.Е., Е.А. Шиленков // Известия ЮЗГУ. – 2012. - №2 (Ч.3). Курск – С. 110-115.
55. **Никитин Г. И.** Сверточные коды. / Г.И. Никитин. - Санкт-Петербург, 2001. – 80 с.
56. **Никитин, Г. И.** Корректирующие коды: Метод. указ. / Г.И. Никитин. - ЛИАП. Ленинград, 1989.- 32 с.

57. **Никитин, Г. И.** Эффективные коды: Метод. указ. / Г.И. Никитин. - ЛИАП. Ленинград, 1987.- 28 с.
58. **Нуссбаумер Г.** Быстрое преобразование Фурье и алгоритмы вычисления свёрток. / Г. Нуссбаумер. - М.: Радио и связь, 1985. — 248 с.
59. **Олифер, В. Г.** Компьютерные сети. Принципы, технологии, протоколы [Текст] : учебное пособие для студентов вузов / В. Г. Олифер, Н. А. Олифер. - 3-е изд. - Москва [и др.] : Питер, 2007. - 957 с. : ил. - (Учебник для вузов). - Библиогр.: с. 919-921 (47 назв.). - Алф. указ.: с. 922-957
60. **Оппенгейм А., Шафер Р.** Цифровая обработка сигналов / А. Оппенгейм, Р Шафер. - Пер.с англ. Кулешова С. А., - Мир цифровой обработки. М: Техносфера, 2008. – 858 с. 5-94836-077-6
61. **Охорзин, В.М.** Построение каскадных кодов на основе кодов Боуза – Чоудхури – Хоквингема и Рида – Соломона. / В.М. Охорзин, Д.С. Кукунин, М.С. Новодворский - Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича. - 2010.
62. **Пат. 2 451 998** Российская Федерация, МПК G06F 17/14, G10L 19/02. Эффективный способ проектирования набора фильтров для mdct/imdct в приложениях для кодирования речи и аудиосигналов / Чивукула Р.К., Резник Ю.Ж Заявитель и патентообладатель Квэлкомм Инк. – № 2010115283/08; заявл. 19.09.2008; опубл. 27.05.2012, Бюл. №15. – 27с.
63. **Перелыгин, С. В.** Разработка методов пространственной обработки речевых сигналов с помощью микрофонной антенной решетки. // «Санкт-Петербургский политехнический университет Петра Великого». - 2016. – 5 с.
64. **Пинчук, Н.В.** Психоакустика и воздействие шума [текст] // Н. Пинчук – С. Пб.: Речь, 2007. – 128 с. ISBN 5-9268-0557-0
65. **Питерсон, У., Уэлдон, Э.** Коды, исправляющие ошибки. / У. Питерсон, Э. Уэлдон. - Москва: Мир, 1976.- 600 с
66. **Полманн, Кен. К.** Principles of Digital Audio. / Кен К. Полман — McGraw-Hill Professional, 2005. — ISBN 978-0-07-144156-8.1
67. **Попов, А.В.** Методы и устройство формирования сигналов в цифровых видеоинформационных системах. [Текст]: автореф. дис. на соиск. учен. степ. канд. техн. наук. 2015 / Попов Андрей Владимирович; МГУСИ. – Москва, 2015. – 79 с.
68. **Потапенко, А.М., Шиленков, Е.А.** Методика построения динамических сигнально-кодовых конструкций с целью повышения пропускной способности канала связи /

А.М. Потапенко, Е.А. Шиленков // Проектирование и технология электронных средств. – 2012. - №1. Владимир – С. 12-16.

69. **Рекомендация ITU-R BT.601** 25th Anniversary and still «in force» - мост, который объединяет аналоговые и цифровые сигналы (1982-2007), ITU-R, 2007.
70. **Ричардсон, Я.** Видеокодирование. H.264 и MPEG-4 - стандарты нового поколения // Москва: Техносфера, 2005. - 368 с. ISBN 5-94836-041-5
71. **Самофалов, К.Г.** Прикладная теория цифровых автоматов. / К.Г. Самофалов, А.М. Романкевич, В.Н. Валуйский, Каневский, Ю.С. Пиневиц М.М. — К.: Вища школа, 1987. — 375 с.
72. **Сергеенко, А.Б.** Цифровая обработка сигналов [Текст]: учебник для вузов / А.Б. Сергеенко. – 2-е изд. – СПб.: Питер, 2007. – 751 с.
73. **Скляр, Б.** Цифровая связь. Теоретические основы и практическое применение / Б. Скляр . – 2-е изд. – М. : Вильямс, 2003 . – 1104 с. - ISBN 5-84590-386-6 .
74. **Скучик, Е.** Основы акустики / Е. Скучик.; Пер. с нем., т. 2, М., 1959
75. **СТО 70238424.17.220.20.005-2011** Системы связи для сбора и передачи информации в электроэнергетике. Условия создания. Нормы и требования. СТАНДАРТ ОРГАНИЗАЦИИ НП «ИНВЭЛ» Режим доступа: <http://docs.cntd.ru/document/1200097077>
76. **Стретт, Док. В.** Теория звука / Пер. с англ. П. Н. Успенского и С. А. Каменецкого. Т. 1. М.: Государственное издательство технико-теоретической литературы, 1955.- 500 с.
77. **Терхард, Э.** Расчет виртуальных полей и исследование слуха / Эрнст Терхард - Xenharmonicon 17, 1998
78. **Финк, Л. М.** Теория передачи дискретных сообщений. / Л.М. Финк. - Москва: Сов. радио, 1963.- 576 с.
79. **Фланаган, Л. Джеймс.** Adaptive quantization in differential PCM coding of speech. / П. Каммиски, Н. С. Джаянт, Л. Джеймс Фланаган — Bell Syst. Tech. J., 1973. — Vol. 52. — P. 1105—1118.1
80. **Чесебиев, И. А.** Компьютерное распознавание и порождение речи / И. А. Чесебиев. - Спорт и культура-2000, 2008. – 128 с. 978-5-901682-44-9
81. **Шварцман В. О., Емельянов Г. А.** Теория передачи дискретной информации: Учебник для вузов связи. / В.О. Шварцман, Г.А. Емельянов. - Москва: Связь, 1979.- 424 с.
82. **Шиленков, Е.А.** Кодирование и сжатие речевого сигнала в ТФОП / Е.А. Шиленков, А.М. Потапенко // Общество, современная наука и образование: проблемы и перспективы: сб. науч. тр. по мат-лам Междунар. заоч. науч.-практ. конф. - 2012: (Ч.10). Тамбов, - С. 159-161.

83. **Шиленков, Е.А.** Методика дескриптора LZSTACKER / Е.А. Шиленков // Материалы Всероссийской научно-практической конференции «Инфокоммуникации и информационная безопасность: состояние, проблемы и пути решения», Курск, 25-26 апреля 2014. – С. 18 - 22.
84. **Шиленков, Е.А.** Методика дескриптора данных по динамическому словарю / Е.А. Шиленков // Материалы Всероссийской научно-практической конференции «Инфокоммуникации и информационная безопасность: состояние, проблемы и пути решения», Курск, 25-26 апреля 2014. – С. 118 - 122.
85. **Шиленков, Е.А.** Методика дескриптора данных по динамическому словарю / Е.А. Шиленков // Материалы Всероссийской научно-практической конференции «Инфокоммуникации и информационная безопасность: состояние, проблемы и пути решения», Курск, 25-26 апреля 2014. – С. 285 – 287.
86. **Шиленков, Е.А.** Методика поиска сжатых блоков в поточных данных / Е.А. Шиленков // Материалы Всероссийской научно-практической конференции «Инфокоммуникации и информационная безопасность: состояние, проблемы и пути решения», Курск, 25-26 апреля 2014. – С. 22 - 24.
87. **Шиленков, Е.А.** Нетривиальность анализа сигналов в дискретном канале / Е.А. Шиленков, К.Ю. Рюмшин, А.Ю. Богомазов // Известия ЮЗГУ. – 2012. - №2 (Ч.2). Курск – С. 72-76.
88. **Шиленков, Е.А.** Определение дескриптора и формата словаря сжатых данных / Е.А. Шиленков // Материалы Всероссийской научно-практической конференции «Инфокоммуникации и информационная безопасность: состояние, проблемы и пути решения», Курск, 25-26 апреля 2014. – С. 287-289.
89. **Шиленков, Е.А.** Способ построения частотно-временной маски для речевого кодирования на основе психоакустической модели слуха / Е.А. Шиленков // Проектирование и технология электронных средств. – 2013. - №4. Владимир – С. 16-20.
90. **Шиленков, Е.А., Хотынюк, С.С.** Линейный предиктор в ортогональном речевом кодере / Е.А. Шиленков, С.С. Хотынюк // Материалы Всероссийской научно-практической конференции «Инфокоммуникации и информационная безопасность: состояние, проблемы и пути решения», Курск, 25-26 апреля 2014. – С. 11 - 17.
91. **Шушарин, А.С.** Полилогия современного мира (Критика запущенной социологии) / А.С. Шушарин – Режим доступа <http://www.shusharin.ru>
92. **Щерба, Л. В.** Опыт общей теории лексикографии // Языковая система и речевая деятельность / Л. В. Щерба. — Л.: Наука, 1974. — 428 с. — 5500 экз. (в пер.)

93. **Якубинский, Л.П.** Избранные работы: Язык и его функционирование. / Л.П. Якубинский - М., 1986. - С. 17-58
94. **ANSI-C Implementations of CCITT G.711, G.721 and G.723 Voice Compressions** by Sun Microsystems, Inc.1
95. **Bäckström, Tom.** Decorrelating MVDR filterbanks using the non-uniform discrete fourier transform // IEEE Signal Process. Lett., 22(4): 479—483, 2014.
96. **Bernstein, L. R.** The effects of signal duration on NoSo and NoSpi thresholds at 500 Hz and 4 kHz / L. R. Bernstein, C. Trahiotis // The Journal of the Acoustical Society of America. – New York, 1999. – P. 105.
97. **Bober, M.** MPEG-7 Visual Shape Descriptors. IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, Issue 6, pp.716-719, 2001
98. **Deutsch, P.** GZIP file format specification version 4.3 / Peter Deutsch // Режим доступа: <ftp://ftp.uu.net/graphics/png/documents/zlib/zdoc-index.html>
99. **Dr. Rao, K. R.** Discrete Transforms and their Applications : курс лекций : [учебное пособие]. Т. 2 / S. Jayaraman, S. Esakkirajan [и др.] отв. ред. Dr. K. R. Rao, Department of electrical engineering, the University of Texas at Arlington, - Paper: ByRadhika, - 2015. – 13с.
100. **Dr. Valin, Jean-Marc. Maxwell, Gregory, Dr. Terriberry, Timothy B.** CELT: A Low-latency, High-quality Audio Codec / Dr. Jean-Marc Valin, Gregory Maxwell, Dr. Timothy B. Terriberry. - 2014. – 62с. Режим доступа: <http://www.celt-codec.org/presentations/misc/lca-celt.pdf>
101. **Edström, B.** Let's build an MP3-decoder! / Bjorn Edström. – C++ Imlementation. – 2009. P 12
102. **G Kiss, M G Tulics, D Sztahó, A Esposito, K Vicsi** Language independent detection possibilities of depression by speech / Recent Advances in Nonlinear Speech Processing. 294 p. Cham (Németország): Springer International Publishing, (Smart Innovation, Systems and Technologies; 48.), - 2016. pp. 103-114. ISBN:978-3-319-28107-0
103. **Geraldine, C.** Data Compression Method – Adaptive Coding with Sliding Window for Information Interchange / C. Geraldine ; NY – 2004
104. **Ghaderi, M. Savoji, M.H.** «Wideband Speech Coding using ADPCM and a New Enhanced Bandwidth Extension Method» / M. Ghaderi, M.H. Savoji // 7 th IEEE International Symposium on Intelligent Signal Processing (WISO), - :Florina -2014. - pp. 1-4.
105. **Guckert, J.** The use of FFT and MDCT in MP3 audio compression / J. Guckert // Math. – 2012. – №2270. – P. 6-8.
106. **ISO/IEC 11172-3** «Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3», 1993 P. 10

107. **ISO/IEC 11172-3. Stage: 90.60** (2010-06-17). Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio.
108. **ISO/IEC 23008-1-2013**. MPEG media transport. Digital container format for Audio, video, data.
109. **ITU-T G.711.0 (2009)** for applications requiring the lossless encoding of frames consisting of 40, 80, 160, 240 and 320 A-/ $\mu$ -law log-PCM samples conforming to Recommendation ITU-T G.711. - 11.2009
110. **ITU-T G.729** : Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP). - 06.2012
111. **Jagtap, S.K. Mulye, M.S.** Speech Coding Techniques / S.K. Jagtap. M.S. Mulye. M.D. Uplane. // Procedia Computer Science. Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15), Volume 49. - 2015. - pp.253–263.
112. **Kuruvilla-Mathew, A. Purdy, Sc. Welch, D.** Cortical encoding of speech acoustics: Effects of noise and amplification / Int J Audiol. – 2015. Vol 54(11), pp. 852-864.
113. **Lee, S., Lee, I.** «A Low-Delay MDCT/IMDCT» / S. Lee, I. Lee // ETRI Journal. - October 2013. - Vol 35, Issue 5 - pp 935-938.
114. **Loeffler, C.** Practical Fast 1-D DCT Algorithms with 11 Multiplications / C. Loeffler, A. Ligtenberg and G. Moschytz. // Proc. Int'l. Conf. on Acoustics, Speech, and Signal Processing 1989 (ICASSP '89), pp. 988—991.
115. **Manas, Ray, Mahesh, Chandra.** Speech Coding Techniques for VoIP Applications: A Technical Review / Ray Manas. Chandra Mahesh. B.P. Patil. // World Applied Sciences Journal 33 (5). - DOI: IDOSI Publication. – 2015. – 8p.
116. **Mark R. P. Thomas, Hannes Gamper, and Ivan J. Tashev,** BFGUI: An Interactive Tool for the Synthesis and Analysis of Microphone Array Beamformers, in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), IEEE – Institute of Electrical and Electronics Engineers, Shanghai, China, March 2016.
117. **Nyquist, H.** Certain topics in telegraph transmission theory. H. Nyquist. - Trans. AIEE, vol. 47, pp. 617—644, Apr. 1928
118. **PKWARE, Inc.** ZIP File Format Specification // Режим доступа: [www.pkware.com/documents/casestudies/APPNOTE.TXT](http://www.pkware.com/documents/casestudies/APPNOTE.TXT)
119. **Rassol, Raissi.** The Theory Behind Mp3. / Raissi Rassol. - December 2002. P 45
120. **Ross N., Williams.** Элементарное руководство CRC алгоритмам обнаружения ошибок / Williams Ross N. - М: Техносфера. - 1993. - 36 с.

121. **Saeid, Rahim and Alku, Paavo.** Feature extraction using power-law adjusted linear prediction with application to speaker recognition under severe vocal effort mismatch / Rahim Saeid. Paavo Alku. Tom Bäckström. // IEEE Trans. Audio, Speech, Lang. Process., 24(1): 42—53, 2016.
122. **Satyapriya, K. Dasari, Y.** Performance analysis of speech coding techniques. / K. Satyapriya, Y. Dasari. // International Journal of Advanced Research in Electrical., Electronics and Instrumentation Engineering. - November 2013; vol.2; issue 11: 5725-32
123. **Saveliev A.I, Basov O.O., Ronzhin A.L., Ronzhin Al.L.** Algorithms for Low Bit-Rate Coding with Adaptation to Statistical Characteristics of Speech Signal. Speech and Computer. Springer, LNAI, Vol. 9319, Proc. SPECOM-2015, Athens, Greece 2015, pp. 65-72.
124. **Schroeder, M. J.** Acoust. Soc. Am. / M. Schroeder. - Binghamton, NY 1979. P. 66, 1647
125. **Shanmuga Raju, S. Karthik, R.** Modified imdct-decoder based mp3 Multichannel audio decoding System / Raju S. Shanmuga, R. Karthik., Pradeep K.P Sai, E. Varadharajan // International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. - Vol. 2, Issue 3, - March 2013. – p8.
126. **Spiegel, M.F.** Simultaneous and nonsimultaneous masking within natural speech / Murray F. Spiegel, M. Schroeder // The Journal of the Acoustical Society of America. – New York, 1979. – P. 66.
127. **Sripada, P.** MP3 DECODER in Theory and Practice. / Praveen Sriada. - Masters Thesis Report.; Blekinge Tekniska Högskola. - March 2006. P 43.
128. **Supurovic, P.** MPEG script / Predrag Supurovic, 10th January 2006, - Режим доступа <http://backpan.perl.org/authors/id/C/CN/CNANDOR/MPEG-MP3Info-0.52.readme>
129. **Terhardt, E.** On the perception of periodic sound fluctuations (roughness) / E. Terhardt // Hearing Research. – 1979. – № 1. – P. 155.
130. **Wallach, D.S.** Accelerated MPEG compression of Dynamic poligonal scenes / D.S Wallach, S Kunapalli, M.F Cohen. - ACM Jun 1991
131. **Yang, X.** MP3 Coding Scheme / X. Yang, C. Waxman. – Philadelphia : University of Pennsylvania, 2012 – 18 p.
132. **Yang, X., Waxman, C.** MP3 Coding Scheme. / Xiang Yang, Corey Waxman Proceedings of the Conference on Design, Automation and Test in Europe 2012 P 652-657. ISBN: 978-3-9810801-5-5
133. **Zwicker, E.** Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen) / E. Zwicker // The Journal of the Acoustical Society of America. – New York, 1961. – P. 33, 248.

## ПРИЛОЖЕНИЕ А Исходный код приложения «Имплементация речевого кодера»

Язык программирования C#, приложение классов WPF

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Microsoft.Research.DynamicDataDisplay.DataSources;
using System.Windows.Threading;
using Microsoft.Research.DynamicDataDisplay;
using System.Globalization;
using System.Media;
namespace wav
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        Int32 chunkSize;
        UInt32 SampleRate;
        UInt16 Quantize;
        UInt32 RawSize;
        int N = 22050; //Ширина окна ДКП
        List<byte> Final = new List<byte>();
        List<byte> Suffix = new List<byte>();
        List<byte> list = new List<byte>();
        List<byte> Decode = new List<byte>();
    }
}
```

```

ChartPlotter plotter = new ChartPlotter();
double phase = 0;
SoundPlayer Play = new SoundPlayer();
EnumerableDataSource<byte> animatedDataSource = null;
EnumerableDataSource<double> animatedDataSource1 = null;
EnumerableDataSource<double> animatedDataSource2 = null;
EnumerableDataSource<double> animatedDataSource3 = null;
List<double> list2 = new List<double>();
/// <summary>Programmatically created header</summary>
Header chartHeader = new Header();
/// <summary>Text contents of header</summary>
TextBlock headerContents = new TextBlock();
List<double[]> Massive_Sample = new List<double[]>();//Коллекция окон сэмлов
List<double[]> Massive_Spectr = new List<double[]>();//Коллекция окон сэмлов
List<double[]> Massive_ODCT = new List<double[]>();//Коллекция окон ОДКП
List<double> Final_ODCT = new List<double>();//Коллекция окон ОДКП
double pi = Math.PI;
double d = 4;//TDMA - коэффициент
//Переменные для тестов
double[] a_Vect = new double[40];
double[] x_Vect = new double[20];
double[] y_Vect = new double[20];
public MainWindow()
{
    InitializeComponent();
    DCT_0.Text = "1";
    DCT_1.Text = "0";
    DCT_2.Text = "1";
    DCT_3.Text = "0";
    DCT_4.Text = "1";
    DCT_5.Text = "0";
    DCT_6.Text = "1";
    DCT_7.Text = "0";
    DCT_8.Text = "1";
    DCT_9.Text = "0";
    DCT_10.Text = "1";
    DCT_11.Text = "0";
    DCT_12.Text = "1";
    DCT_13.Text = "0";
    DCT_14.Text = "1";
    DCT_15.Text = "0";
    DCT_16.Text = "1";
}

```

```

DCT_17.Text = "0";
DCT_18.Text = "1";
DCT_19.Text = "0";
}
private void Button_Click(object sender, RoutedEventArgs e)//Загрузить
{
    using (var file = System.IO.File.OpenRead("8x8.wav"))
    {
        byte[] buffer = new byte[file.Length];
        file.Read(buffer, 0, buffer.Length);
        Final.AddRange(buffer);
        Final.RemoveRange(44, Final.Count - 44);
        Decode.AddRange(buffer);
        Decode.RemoveRange(44, Decode.Count - 44);
        chunkID_text.Text = Encoding.ASCII.GetString(buffer, 0, 4);
        chunkSize = BitConverter.ToInt32(buffer, 4) + 8;
        chunkSize_text.Text = chunkSize.ToString();
        format_text.Text = Encoding.ASCII.GetString(buffer, 8, 4);
        UInt16 format=BitConverter.ToUInt16(buffer, 20);
        if (format == 1) audioFormat_text.Text = "PCM";
        UInt16 chanel = BitConverter.ToUInt16(buffer, 22);
        chanel_text.Text = chanel.ToString();
        SampleRate = BitConverter.ToInt32(buffer, 24);
        sample_rate_text.Text = SampleRate.ToString();
        Quantize = BitConverter.ToUInt16(buffer, 34);
        quant_text.Text = Quantize.ToString();
        RawSize = BitConverter.ToInt32(buffer, 40);
        rawsize_text.Text = RawSize.ToString();
        list.AddRange(buffer);//Записываем в копилку (list) данные из буфера
        Suffix.AddRange(buffer);
        Suffix.RemoveRange(0, (int)RawSize - 1);
        list.RemoveRange((int)RawSize - 1, (chunkSize)-((int)RawSize-1));
        list.RemoveRange(0, 44);
    }
    SystemSounds.Asterisk.Play();
}
private void Button_Click_1(object sender, RoutedEventArgs e)
{
    if (Application.Current.Windows.Count < 2)//Если это окно еще не существует
    {
        Window win = new Window();//Создаём окно
        win.Height = 400;
    }
}

```

```

win.Width = 1200;
win.Show();//Показать окно
win.Content = plotter;
headerContents.FontSize = 24;
headerContents.Text = "Phase = 0.00";
headerContents.HorizontalAlignment = HorizontalAlignment.Center;
chartHeader.Content = headerContents;
plotter.Children.Remove(chartHeader);
plotter.Children.Add(chartHeader);
win.ContentRendered += win_ContentRendered; //Событие по окончании создания окна
}
}
void win_ContentRendered(object sender, EventArgs e)
{
    UInt32[] animatedX = new UInt32[list.Count];
    byte[] animatedY = new byte[list.Count];
    for (int i = 0; i < list.Count; i++)
    {
        animatedX[i] = (UInt32)i;
        animatedY[i] = list[i];
    }
    EnumerableDataSource<UInt32> xSrc = new EnumerableDataSource<UInt32>(animatedX);
    xSrc.SetXMapping(x => x);
    animatedDataSource = new EnumerableDataSource<byte>(animatedY);
    animatedDataSource.SetYMapping(y => y);

    // Adding graph to plotter
    plotter.RemoveUserElements();
    plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource), new Pen(Brushes.BlueViolet, 1),
new PenDescription("PCM"));
    plotter.FitToView();
}
private void Button_Click_2(object sender, RoutedEventArgs e)// Воспроизведение
{
    Play.SoundLocation = "8x8.wav";
    Play.Load();
    Play.Play();
}
private void Button_Click_3(object sender, RoutedEventArgs e)//Фильтрация
{
    double[] b = new double[16] { 0.035, 1.1, 0.551, -0.086, -0.623, -0.907, -0.892, -0.627, -0.231, 0.156, 0.419, 0.502,
0.419, 0.239, 0.046, -0.086 }; //Фильтр

```

```

double[] x = new double[list.Count];
for (int i=0; i<list.Count; i++)
{
    x[i] = (double)list[i];
    x[i] -= 128;//делаем сигнал биполярным
}
int M = b.Length;
int n = list.Count;
//y[n]=b0x[n]+b1x[n-1]+...bmx[n-M]
double [] y = new double[n];
for (int yi = 0; yi < n; yi++)
{
    double t = 0.0;
    for (int bi = M-1; bi >=0; bi--)
    {
        if (yi - bi < 0) continue;
        t += b[bi] * x[yi - bi];
    }
    y[yi] = t;
}
list2.AddRange(y);
byte[] store = new byte[list2.Count];
for (int i=0; i<list2.Count;i++)
{
    store[i]=(byte)(list2[i]/2 + 128);
}
Final.AddRange(store);
Final.AddRange(Suffix);
byte[] save = new byte[Final.Count];
for (int i = 0; i < Final.Count; i++)
{
    save[i] = Final[i];
}
File.WriteAllBytes(@"filter.wav", save);
Window win2 = new Window();//Создаём окно
win2.Height = 400;
win2.Width = 1200;
win2.Show();//Показать окно
win2.Content = plotter;
headerContents.FontSize = 24;
headerContents.Text = "Phase = 0.00";
headerContents.HorizontalAlignment = HorizontalAlignment.Center;

```

```

plotter.Children.Remove(chartHeader);
chartHeader.Content = headerContents;
win2.ContentRendered += win2_ContentRendered; //Событие по окончанию создания окна
}
void win2_ContentRendered(object sender, EventArgs e)//Открываем окно
{
    UInt32[] animatedX = new UInt32[list2.Count];
    double[] animatedY = new double[list2.Count];
    for (int i = 0; i < list.Count; i++)
    {
        animatedX[i] = (UInt32)i;
        animatedY[i] = list2[i];
    }
    EnumerableDataSource<UInt32> xSrc = new EnumerableDataSource<UInt32>(animatedX);
    xSrc.SetXMapping(x => x);
    animatedDataSource2 = new EnumerableDataSource<double>(animatedY);
    animatedDataSource2.SetYMapping(y => y);

    // Adding graph to plotter
    plotter.RemoveUserElements();
    plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource2), new Pen(Brushes.BlueViolet, 1),
new PenDescription("PCM"));
    plotter.FitToView();
}
private void Button_Click_4(object sender, RoutedEventArgs e)//Воспроизведение результата
{
    Play.SoundLocation = "filter.wav";
    Play.Load();
    Play.Play();
}
private double[] FDCT(double[] x, int N)//Функция ДКП от сэмпла
{
    double[] y = new double[N / 2];
    int q = Convert.ToInt32(Quant.Text);
    for (int k = 0; k < N / 2; k++)
    {
        double summ = 0;
        double D = N / (Math.Pow(2, d - 1));
        double[] h = new double[N];
        for (int n = 0; n < N; n++)
        {
            double cos = (pi * (2 * n + N + 1) * (2 * k + 1) / (2 * N));

```

```

    h[n] = Math.Pow(Math.Sin((pi / 2) * Math.Sin((pi / 2) * (n + 0.5) / (N))), 2);
    summ += x[n]*h[n]*Math.Cos(cos);
}
double sqrt = (double)1/N;
y[k] = summ/21;
y[0] = 0;
}
return y;
}
private void Button_Click_5(object sender, RoutedEventArgs e)//Выполнение ДКП
{
    d = Convert.ToDouble(TDA.Text);
    N = Convert.ToInt32(win_width.Text);
    int AA = Convert.ToInt32(AntiAlias.Text);
    int BB = Convert.ToInt32(Overlap.Text);
    int NumberOfSpect = list.Count/ N - 1;//Число окон сэмплов
    //y[n]=b0x[n]+b1x[n-1]+...bmx[n-M]
    double[] y = new double[list.Count];
    for (int i = 0; i < list.Count; i++)
    {
        y[i] = list[i];
        y[i] -= 128;
    }
    for (int i = 0; i < NumberOfSpect; i++)//Разделяем поток на сэмплы
    {
        double[] dct = new double[N+AA+BB];
        {
            for (int n = 0; n < N; n++)
            {
                dct[n+BB] = y[(N * i) + n];//Выполняем перекрытие
            }
            for (int n = N; n < BB; n++)
            {
                //Выполняем перекрытие
                dct[BB-n-1] = y[(N * (i)) + n];
            }
            Massive_Sample.Add(dct);
        }
    }
    for (int i = 0; i < NumberOfSpect; i++)//Выполняем БДКП для каждого сэмпла
    {
        double[] fdct = new double[N+AA+BB];

```

```

    fdct = FDCT(Massive_Sample[i], N+AA+BB);
    Massive_Spectr.Add(fdct);
}
}
private void Button_Click_6(object sender, RoutedEventArgs e)//Показать спектр
{
    if (Application.Current.Windows.Count < 2)//Если это окно еще не существует
    {
        Window win3 = new Window();//Создаём окно
        win3.Height = 400;
        win3.Width = 1200;
        win3.Show();//Показать окно
        win3.Content = plotter;
        headerContents.FontSize = 24;
        headerContents.Text = "Phase = 0.00";
        headerContents.HorizontalAlignment = HorizontalAlignment.Center;
        chartHeader.Content = headerContents;
        plotter.Children.Remove(chartHeader);
        plotter.Children.Add(chartHeader);
        win3.ContentRendered += win3_ContentRendered; //Событие по окончании создания окна
    }
}
void win3_ContentRendered(object sender, EventArgs e)//Открываем окно
{
    N = Convert.ToInt32(win_width.Text);
    UInt32[] animatedX = new UInt32[N];
    double[] animatedY = new double[N];
    double[] spectr = new double[N];
    Int32 n = Convert.ToInt32(Number.Text);
    spectr = Massive_Spectr[n];
    for (int i = 0; i < N; i++)
    {
        spectr[i] = Math.Abs(spectr[i]);
    }
    double max = spectr.Max();
    for (int i = 0; i < N; i++)
    {
        animatedX[i] = (UInt32)i;
        animatedY[i] = -Math.Pow(i - Array.IndexOf(spectr, max), Convert.ToInt32(MaskPow.Text)) + max;
        if (animatedY[i] < 0) animatedY[i] = 0;
    }
    EnumerableDataSource<UInt32> xSrc = new EnumerableDataSource<UInt32>(animatedX);

```

```

xSrc.SetXMapping(x => x);
animatedDataSource3 = new EnumerableDataSource<double>(animatedY);
animatedDataSource3.SetYMapping(y => y);
// Adding graph to plotter
plotter.RemoveUserElements();
plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource3), new Pen(Brushes.BlueViolet, 1),
new PenDescription("Spectrum"));
plotter.FitToView();
UInt32[] animatedX1 = new UInt32[N];
double[] animatedY1 = new double[N];
for (int i = 0; i < N; i++)
{
    animatedX1[i] = (UInt32)i;
    animatedY1[i] = spectr[i];
}
EnumerableDataSource<UInt32> xSrc2 = new EnumerableDataSource<UInt32>(animatedX1);
xSrc2.SetXMapping(x => x);
animatedDataSource2 = new EnumerableDataSource<double>(animatedY1);
animatedDataSource2.SetYMapping(y => y);
// Adding graph to plotter
plotter.AddLineGraph(new CompositeDataSource(xSrc2, animatedDataSource2), new Pen(Brushes.Green, 1), new
PenDescription("Spectrum"));
plotter.FitToView();
}
private double[] OFDCT(double[] x, int N, double[] a)//ОДКП
{
    int q = Convert.ToInt32(Quant.Text);
    double[] y = new double[N];
    for (int n = 0; n < N; n++)
    {
        double summ = 0;
        double[] h = new double[N];
        double D = N / (Math.Pow(2, d - 1));
        for (int k = 0; k < N / 2; k++)
        {
            double cos = (pi * (2*n + N + 1) * (2*k + 1) / (2*N));
            summ += x[k] * Math.Cos(cos);
        }
        h[n] = Math.Pow(Math.Sin((pi / 2) * Math.Sin((pi / 2) * (n + 0.5) / (N))), 2);
        y[n] = h[n] * summ*120/N;
    }
    return y;
}

```

```

}
private void Button_Click_7(object sender, RoutedEventArgs e)//Выполнение ОБДКП
{
    d = Convert.ToDouble(TDA.Text);
    N = Convert.ToInt32(win_width.Text);
    int q = Convert.ToInt32(Quant.Text);
    int AA = Convert.ToInt32(AntiAlias.Text);
    int BB = Convert.ToInt32(Overlap.Text);
    using (var file = System.IO.File.OpenRead("Coded.cod"))
    {
        byte[] buffer = new byte[file.Length];
        file.Read(buffer, 0, buffer.Length);
        Massive_Spectr.RemoveRange(0, Massive_Spectr.Count);
        for (int n = 0; n < list.Count / (N) - 1; n++)
        {
            double[] sample = new double[N];
            for (int i = 0; i < N; i++)
            {
                sample[i] = ((double)(buffer[N * n + i]) - 128) * q;
            }
            Massive_Spectr.Add(sample);
        }
    }
    int NumberOfSpect = list.Count / (N)-1;//Число окон сэмплов
    for (int i = 0; i < NumberOfSpect; i++)//Выполняем ОБДКП для каждого сэмпла
    {
        double[] fdct = new double[N+AA+BB];
        fdct = OFDCT(Massive_Spectr[i], N+AA+BB, Massive_Sample[i]);
        Final_ODCT.AddRange(fdct);
        Final_ODCT.RemoveRange(Final_ODCT.Count - fdct.Length, BB);
        Final_ODCT.RemoveRange(Final_ODCT.Count - AA - 1, AA);
    }
    byte[] store = new byte[list.Count];
    for (int i = 0; i < Final_ODCT.Count; i++)
    {
        Final_ODCT[i] += 128;
        store[i] = (byte)(Math.Round(Final_ODCT[i]));
    }
    Decode.AddRange(store);
    Decode.AddRange(Suffix);
    byte[] save = new byte[Decode.Count];
    for (int i = 0; i < Decode.Count; i++)

```

```

    {
        save[i] = Decode[i];
    }
    File.WriteAllBytes(@"decode.wav", save);
}
void win4_ContentRendered(object sender, EventArgs e)
{
    UInt32[] animatedX = new UInt32[Final_ODCT.Count];
    double[] animatedY = new double[Final_ODCT.Count];
    for (int i = 0; i < Final_ODCT.Count; i++)
    {
        animatedX[i] = (UInt32)i;
        animatedY[i] = Final_ODCT[i];
    }
    EnumerableDataSource<UInt32> xSrc = new EnumerableDataSource<UInt32>(animatedX);
    xSrc.SetXMapping(x => x);
    animatedDataSource2 = new EnumerableDataSource<double>(animatedY);
    animatedDataSource2.SetYMapping(y => y);

    plotter.RemoveUserElements();
    // Adding graph to plotter
    Pen pen = new Pen(Brushes.Coral, 1);
    pen.DashCap = PenLineCap.Square;
    pen.DashStyle = DashStyles.Solid;
    plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource2), pen, new
PenDescription("PCM"));
    plotter.FitToView();
}
private void Button_Click_8(object sender, RoutedEventArgs e)//Tect
{
    double[] a = new double[20] { Convert.ToDouble(DCT_0.Text), Convert.ToDouble(DCT_1.Text),
Convert.ToDouble(DCT_2.Text), Convert.ToDouble(DCT_3.Text), Convert.ToDouble(DCT_4.Text),
Convert.ToDouble(DCT_5.Text), Convert.ToDouble(DCT_6.Text), Convert.ToDouble(DCT_7.Text),
Convert.ToDouble(DCT_8.Text), Convert.ToDouble(DCT_9.Text), Convert.ToDouble(DCT_10.Text),
Convert.ToDouble(DCT_11.Text), Convert.ToDouble(DCT_12.Text), Convert.ToDouble(DCT_13.Text),
Convert.ToDouble(DCT_14.Text), Convert.ToDouble(DCT_15.Text), Convert.ToDouble(DCT_16.Text),
Convert.ToDouble(DCT_17.Text), Convert.ToDouble(DCT_18.Text), Convert.ToDouble(DCT_19.Text), };
    double[] x = new double[40];
    for (int i=0; i<x.Length/2; i++)
    {
        x[i] = a[a.Length-i-1];
        x[i + x.Length / 2] = a[i];
    }
}

```

```

}
double[] y = new double[40];
int N = x.Length;
for (int k = 0; k < N / 2; k++)
{
    double summ = 0;
    double D = N / (Math.Pow(2, d - 1));
    double[] h = new double[N];
    for (int n = 0; n < N; n++)
    {
        double cos = (pi * (2 * n + N) * (2 * k + 1) / (2 * N));
        h[n] = Math.Pow(Math.Sin((pi / 2) * Math.Sin((pi / 2) * (n + 0.5) / (N))), 2);
        summ += x[n] * h[n] * Math.Cos(cos);
    }
    y[k] = summ;
}
double[] x1 = new double[40];
double[] y1 = new double[40];
N = y.Length;
for (int n = 0; n < N; n++)
{
    double summ = 0;
    double[] h = new double[N];
    for (int k = 0; k < N / 2; k++)
    {
        double cos = (pi * (2 * n + N) * (2 * k + 1) / (2 * N));
        summ += y[k] * Math.Cos(cos);
    }
    h[n] = Math.Pow(Math.Sin((pi / 2) * Math.Sin((pi / 2) * (n + 0.5) / (N))), 2);
    y1[n] = summ * h[n] / (N/2);
}
y1[0] = 0;
//Записываем числа в поля ОДКП
DCT_0_Copy.Text = y1[0].ToString("F2");
DCT_1_Copy.Text = y1[1].ToString("F2");
DCT_2_Copy.Text = y1[2].ToString("F2");
DCT_3_Copy.Text = y1[3].ToString("F2");
DCT_4_Copy.Text = y1[4].ToString("F2");
DCT_5_Copy.Text = y1[5].ToString("F2");
DCT_6_Copy.Text = y1[6].ToString("F2");
DCT_7_Copy.Text = y1[7].ToString("F2");
DCT_8_Copy.Text = y1[8].ToString("F2");

```

```

DCT_9_Copy.Text = y1[9].ToString("F2");
DCT_10_Copy.Text = y1[10].ToString("F2");
DCT_11_Copy.Text = y1[11].ToString("F2");
DCT_12_Copy.Text = y1[12].ToString("F2");
DCT_13_Copy.Text = y1[13].ToString("F2");
DCT_14_Copy.Text = y1[14].ToString("F2");
DCT_15_Copy.Text = y1[15].ToString("F2");
DCT_16_Copy.Text = y1[16].ToString("F2");
DCT_17_Copy.Text = y1[17].ToString("F2");
DCT_18_Copy.Text = y1[18].ToString("F2");
DCT_19_Copy.Text = y1[19].ToString("F2");
a_Vect=a;
x_Vect = x;
y_Vect = y1;
Window win5 = new Window();//Создаём окно
win5.Height = 400;
win5.Width = 1200;
win5.Show();//Показать окно
win5.Content = plotter;
headerContents.FontSize = 24;
headerContents.Text = "Phase = 0.00";
headerContents.HorizontalAlignment = HorizontalAlignment.Center;
chartHeader.Content = headerContents;
plotter.Children.Remove(chartHeader);
plotter.Children.Add(chartHeader);
win5.ContentRendered += win5_ContentRendered; //Событие по окончанию создания окна
}
void win5_ContentRendered(object sender, EventArgs e)
{
    UInt32[] animatedX = new UInt32[y_Vect.Length];
    double[] animatedY = new double[y_Vect.Length];
    for (int i = 0; i < y_Vect.Length; i++)
    {
        animatedX[i] = (UInt32)i;
        animatedY[i] = y_Vect[i]-1;
    }
    EnumerableDataSource<UInt32> xSrc = new EnumerableDataSource<UInt32>(animatedX);
    xSrc.SetXMapping(x => x);
    animatedDataSource2 = new EnumerableDataSource<double>(animatedY);
    animatedDataSource2.SetYMapping(y => y);
    plotter.RemoveUserElements();
    // Adding graph to plotter

```

```

Pen pen = new Pen(Brushes.Green, 2);
pen.DashCap = PenLineCap.Square;
pen.DashStyle = DashStyles.Dash;
plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource2), pen, new
PenDescription("PCM"));
plotter.FitToView();
UInt32[] animatedX1 = new UInt32[x_Vect.Length];
double[] animatedY1 = new double[x_Vect.Length];
for (int i = 0; i < x_Vect.Length; i++)
{
    animatedX1[i] = (UInt32)i;
    animatedY1[i] = x_Vect[i];
}
xSrc = new EnumerableDataSource<UInt32>(animatedX1);
xSrc.SetXMapping(x => x);
animatedDataSource3 = new EnumerableDataSource<double>(animatedY1);
animatedDataSource3.SetYMapping(y => y);
// Adding graph to plotter
pen = new Pen(Brushes.BlueViolet, 2);
pen.DashCap = PenLineCap.Square;
//pen.DashStyle = DashStyles.Dash;
plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource3), pen, new
PenDescription("PCM"));
plotter.FitToView();
UInt32[] animatedX2 = new UInt32[a_Vect.Length];
double[] animatedY2 = new double[a_Vect.Length];
for (int i = 0; i < a_Vect.Length; i++)
{
    animatedX2[i] = (UInt32)i+10;
    animatedY2[i] = a_Vect[i]+1;
}
xSrc = new EnumerableDataSource<UInt32>(animatedX2);
xSrc.SetXMapping(x => x);
animatedDataSource1 = new EnumerableDataSource<double>(animatedY2);
animatedDataSource1.SetYMapping(y => y);
// Adding graph to plotter
pen = new Pen(Brushes.Red, 2);
pen.DashCap = PenLineCap.Square;
//pen.DashStyle = DashStyles.Dash;
plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource1), pen, new
PenDescription("PCM"));
plotter.FitToView();

```

```

}
private void Button_Click_10(object sender, RoutedEventArgs e)
{
    Play.SoundLocation = "decode.wav";
    Play.Load();
    Play.Play();
} //Воспроизведение декодера
private void SBROS(object sender, RoutedEventArgs e)
{
    list.RemoveRange(0, list.Count);
    list2.RemoveRange(0, list2.Count);
    Final.RemoveRange(0, Final.Count);
    Suffix.RemoveRange(0, Suffix.Count);
    Decode.RemoveRange(0, Decode.Count);
    Massive_Sample.RemoveRange(0, Massive_Sample.Count);
    Massive_ODCT.RemoveRange(0, Massive_ODCT.Count);
    Massive_Spectr.RemoveRange(0, Massive_Spectr.Count);
    Final_ODCT.RemoveRange(0, Final_ODCT.Count);
}
private double[] MASK(double[] x, int N)//Маскирование
{
    double pow = Convert.ToDouble(MaskPow.Text);
    int q = Convert.ToInt32(Quant.Text);
    double[] y = new double[N];
    for (int i = 0; i < N; i++)
    {
        y[i] = Math.Abs(x[i]);
    }
    double max = y.Max();
    for (int k = 0; k < N; k++)
    {
        for (int n = 0; n < N; n++)
        {
            if (y[n] != 0)
            {
                double z = -Math.Pow((n - Array.IndexOf(y,max)), pow) + max;
                if (y[n] < z) { y[n] = 0; x[n] = 0; }
            }
        }
    }
    y[Array.IndexOf(y,max)] = 0;
    max = y.Max();
}

```

```

return x;
}
private double[] QUANT(double[] x, int N,int q)//Маскирование
{
    for (int i=0;i<N;i++)
    {
        if (i<(N/10)) x[i] = Math.Round(x[i]/(3*q))+128;
        if ((i >=(N / 10))&&(i<=(N/6))) x[i] = x[i] / (q) + 128;
        if (i > (N / 6)) x[i] = x[i] / (3* q) + 128;
    }
    return x;
}
private void Masking(object sender, RoutedEventArgs e)
{
    List<double> Coded = new List<double>();
    int q = Convert.ToInt32(Quant.Text);
    N += Convert.ToInt32(Overlap.Text);
    for (int i=0;i<Massive_Spectr.Count;i++)
    {
        double[] z = new double[Massive_Spectr[i].Length];
        z = Massive_Spectr[i];
        //Massive_Spectr[i] = MASK(z, N / 2);
        z = Massive_Spectr[i];
        Massive_Spectr[i] = QUANT(z, N / 2, q);
        Coded.AddRange(Massive_Spectr[i]);//Записываем тело сжатого файла
    }
    double max = Coded.Max();
    byte[] save = new byte[Coded.Count];//Запись сжатого файла
    for (int i = 0; i < Coded.Count; i++)
    {
        //Coded[i] /= q;
        //Coded[i] += 128;
        save[i] = (byte)(Math.Round(Coded[i]));//Округляем для целого, заканчиваем неравномерное квантование
    }
    File.WriteAllBytes(@"Coded.cod", save);
    SystemSounds.Asterisk.Play();
}
private void Button_Click_11(object sender, RoutedEventArgs e)
{
    Window win4 = new Window();//Создаём окно
    win4.Height = 400;
    win4.Width = 1200;
}

```

```

win4.Show();//Показать окно
win4.Content = plotter;
headerContents.FontSize = 24;
headerContents.Text = "Phase = 0.00";
headerContents.HorizontalAlignment = HorizontalAlignment.Center;
chartHeader.Content = headerContents;
plotter.Children.Remove(chartHeader);
plotter.Children.Add(chartHeader);
win4.ContentRendered += win4_ContentRendered; //Событие по окончании создания окна
}
private void Button_Click_9(object sender, RoutedEventArgs e)
{
    List<byte> Pitch = new List<byte>();
    Pitch.AddRange(list);
    byte m = Pitch.Max();
    int max = Pitch.IndexOf(Pitch.Max());
    int deep = Convert.ToInt32(Pitch_Deep.Text);
    for (int k = 0; k < Pitch.Count; k++)
    {
        if (Pitch[k] >= Convert.ToInt32(Pitch_Bottom.Text))
        {
            for (int i=k;i<Pitch.Count-k;i++)
            {
                if (((Pitch[k + i] < 127 + 128 / Math.Pow(((Convert.ToDouble(i) - k) / Convert.ToInt32(Pitch_Deep.Text)),
Convert.ToInt32(Pitch_Mask.Text)) && (256 - Pitch[k + i] < 127 + 128 / Math.Pow(((Convert.ToDouble(i) - k) /
Convert.ToInt32(Pitch_Deep.Text)), Convert.ToInt32(Pitch_Mask.Text)))))) list[i + k] = 128;
            }
        }
    }
    Final.AddRange(list);
    Final.AddRange(Suffix);
    byte[] save = new byte[Final.Count];
    for (int i = 0; i < Final.Count; i++)
    {
        save[i] = Final[i];
    }
    File.WriteAllBytes(@"filter.wav", save);
    Window win6 = new Window();//Создаём окно
    win6.Height = 400;
    win6.Width = 1200;
    win6.Show();//Показать окно
    win6.Content = plotter;

```

```

headerContents.FontSize = 24;
headerContents.Text = "Phase = 0.00";
headerContents.HorizontalAlignment = HorizontalAlignment.Center;
chartHeader.Content = headerContents;
plotter.Children.Remove(chartHeader);
plotter.Children.Add(chartHeader);
win6.ContentRendered += win6_ContentRendered; //Событие по окончании создания окна
} //Pitch
void win6_ContentRendered(object sender, EventArgs e)
{
    UInt32[] animatedX = new UInt32[list.Count];
    double[] animatedY = new double[list.Count];
    for (int i = 0; i < list.Count; i++)
    {
        animatedX[i] = (UInt32)i;
        animatedY[i] = list[i];
    }
    EnumerableDataSource<UInt32> xSrc = new EnumerableDataSource<UInt32>(animatedX);
    xSrc.SetXMapping(x => x);
    animatedDataSource2 = new EnumerableDataSource<double>(animatedY);
    animatedDataSource2.SetYMapping(y => y);
    plotter.RemoveUserElements();
    // Adding graph to plotter
    Pen pen = new Pen(Brushes.Green, 2);
    pen.DashCap = PenLineCap.Square;
    pen.DashStyle = DashStyles.Solid;
    plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource2), pen, new
PenDescription("PCM"));
    plotter.FitToView();
    byte m = list.Max();
    int max = list.IndexOf(list.Max());
    int deep = Convert.ToInt32(Pitch_Deep.Text);
    UInt32[] animatedX1 = new UInt32[list.Count];
    double[] animatedY1 = new double[list.Count];
    for (int i = 0; i < list.Count; i++)
    {
        animatedX1[i] = (UInt32)i;
        animatedY1[i] = 127 + 128 / Math.Pow((((Convert.ToDouble(i) - max) / Convert.ToInt32(Pitch_Deep.Text)),
Convert.ToInt32(Pitch_Mask.Text));
        if (animatedY1[i]>256) animatedY1[i]=256;
    }
    xSrc = new EnumerableDataSource<UInt32>(animatedX1);

```

```

xSrc.SetXMapping(x => x);
animatedDataSource3 = new EnumerableDataSource<double>(animatedY1);
animatedDataSource3.SetYMapping(y => y);
// Adding graph to plotter
pen = new Pen(Brushes.BlueViolet, 2);
pen.DashCap = PenLineCap.Square;
//pen.DashStyle = DashStyles.Dash;
plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource3), pen, new
PenDescription("PCM"));
plotter.FitToView();
//UInt32[] animatedX2 = new UInt32[a_Vect.Length];
//double[] animatedY2 = new double[a_Vect.Length];
//for (int i = 0; i < a_Vect.Length; i++)
//{
//  animatedX2[i] = (UInt32)i + 10;
//  animatedY2[i] = a_Vect[i] + 1;
//}
//xSrc = new EnumerableDataSource<UInt32>(animatedX2);
//xSrc.SetXMapping(x => x);
//animatedDataSource1 = new EnumerableDataSource<double>(animatedY2);
//animatedDataSource1.SetYMapping(y => y);
//// Adding graph to plotter
//pen = new Pen(Brushes.Red, 2);
//pen.DashCap = PenLineCap.Square;
////pen.DashStyle = DashStyles.Dash;
//plotter.AddLineGraph(new CompositeDataSource(xSrc, animatedDataSource1), pen, new
PenDescription("PCM"));
//plotter.FitToView();
}
}
}

```