

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Е. К. ЛЕВИН

ИССЛЕДОВАНИЕ АЛГОРИТМОВ
ОБРАБОТКИ РЕЧЕВЫХ СИГНАЛОВ
В ТЕЛЕФОНИИ

Лабораторный практикум



Владимир 2024

УДК 621.395

ББК 32.882

Л36

Рецензенты:

Доктор технических наук, профессор
профессор кафедры электроники, приборостроения
и биотехнических систем

Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых
Л. Т. Сушкова

Доктор технических наук, доцент
профессор кафедры динамики электронных систем
Ярославского государственного университета им. П. Г. Демидова
А. Л. Приоров

Издается по решению редакционно-издательского совета ВлГУ

Левин, Е. К. Исследование алгоритмов обработки речевых
Л36 сигналов в телефонии : лаб. практикум / Е. К. Левин ; Владим.
гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ,
2024. – 84 с. – ISBN 978-5-9984-1821-1.

На основе функционального моделирования в системе Matlab проведено исследование алгоритмов обработки речевых сигналов в телефонии. Рассмотрены основные подходы к сжатию потоков данных речи: квантование сигнала с использованием компандирования сигналов, адаптивное и разностное квантование, кодирование сигналов на основе модели формирования речевого сигнала.

Предназначено для студентов направлений подготовки 11.03.01 «Радиотехника» и 11.03.02 «Инфокоммуникационные технологии и системы связи».

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Ил. 29. Библиогр.: 3 назв.

УДК 621.395

ББК 32.882

ISBN 978-5-9984-1821-1

© ВлГУ, 2024

ВВЕДЕНИЕ

Современная телефония характеризуется высоким качеством речевого сигнала (РС), что обусловлено представлением РС в цифровом виде. Однако непосредственное аналого-цифровое преобразование (АЦП) речевых сигналов в телефонии приводит к появлению больших потоков данных, которые значительно загружают канал связи. Существует несколько подходов к сжатию потоков данных речи. Их можно разделить на две группы.

К первой группе относят алгоритмы, которые при сжатии потока данных речи ставят цель сохранения формы речевого сигнала. Это такие методы, как использование компандирования сигналов при их квантовании, адаптивное квантование, разностное квантование. Указанные методы обеспечивают высокое качество РС, но степень сжатия потоков данных невелика.

К второй группе относят речевые кодеки (вокодеры), которые обеспечивают намного бóльшую степень сжатия потоков данных речи по сравнению с методами первой группы за счет устранения информационной избыточности РС. В этом случае по каналу связи передают не отсчеты сигнала, а параметры модели формирования РС. Поскольку параметры модели меняются во времени относительно медленно и количество параметров невелико, то удастся значительно снизить скорость потока данных. Но искажения РС на выходе кодеков больше по сравнению с искажениями при использовании методов первой группы. Чтобы обеспечить приемлемый уровень искажений при высокой степени сжатия потока данных, используют сложные алгоритмы обработки РС.

Ввиду сложности алгоритмов обработки РС целесообразно проводить их исследование на основе функционального моделирования.

В рамках учебных дисциплин «Нейросетевые алгоритмы обработки речевых сигналов в телефонии» (для магистрантов) и «Обработка сигналов звука и изображения» (для студентов бакалавриата) при выполнении лабораторных работ исследуют особенности АЦП речевых сигналов: определяют количество двоичных разрядов чисел, которыми представляются отсчеты РС при равномерном, адаптивном и разностном видах квантования; оценивают отношение сигнал-шум квантования; вычисляют размер кадра сигнала, на котором определяются параметры сигнала; исследуют особенности сжатия потоков данных РС на основе знания процесса формирования РС.

В лабораторном практикуме приводятся краткие теоретические сведения об алгоритмах обработки РС в телефонии и методические рекомендации по проведению экспериментальных исследований в среде системы Matlab. Исследуются следующие виды квантования РС: квантование с постоянным шагом квантования, квантование с использованием компандирования сигнала, адаптивное и разностное квантование. Исследуется речевой кодек на основе метода линейного предсказания.

Приводятся сценарии Matlab с подробными комментариями. В сценариях реализована возможность оценить результаты исследования путем прослушивания РС после их обработки.

Лабораторная работа № 1

ИССЛЕДОВАНИЕ КВАНТОВАТЕЛЯ С ПОСТОЯННЫМ ШАГОМ КВАНТОВАНИЯ

Цель работы: для заданных звукового файла и отношения сигнал-шум квантования рассчитать число двоичных разрядов квантователя и шаг квантования и провести проверку правильности расчета.

Краткие теоретические сведения

Квантование сигнала и его дискретизация во времени составляют суть АЦП сигнала. Кроме того, в состав АЦП входит низкочастотная фильтрация аналогового сигнала, которая подавляет высокочастотные спектральные компоненты сигнала: сверху ограничивает спектр аналогового сигнала.

Структурная схема аналого-цифрового преобразователя представлена на рис. 1.1. Она содержит фильтр нижних частот (ФНЧ), ограничивающий сверху спектр аналогового сигнала; ключевой элемент, обеспечивающий дискретизацию сигнала во времени; и квантователь, преобразующий дискретные отсчеты сигнала в двоичные числа.

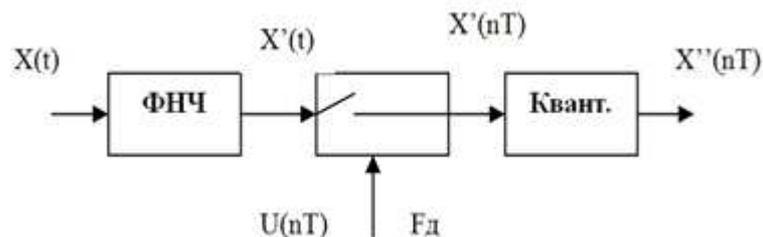


Рис. 1.1. Структурная схема аналого-цифрового преобразователя

При дискретизации ограниченный по спектру сигнал $X'(t)$ преобразуется в периодическую последовательность $X'(nT)$ амплитудно-модулированных коротких импульсов, где T – период дискретизации сигнала.

Спектр $S_{X'}$ дискретизированного сигнала образуется из спектра S_u периодической последовательности коротких импульсов $U(nT)$ путем

свертки спектров S_X и S_u – добавления боковых полос к каждому спектральному компоненту S_u .

Из рис. 1.2 следует, что установкой ФНЧ с полосой F_n после дискретизации можно вновь выделить спектр S_X , а следовательно, и восстановить $X'(t)$.

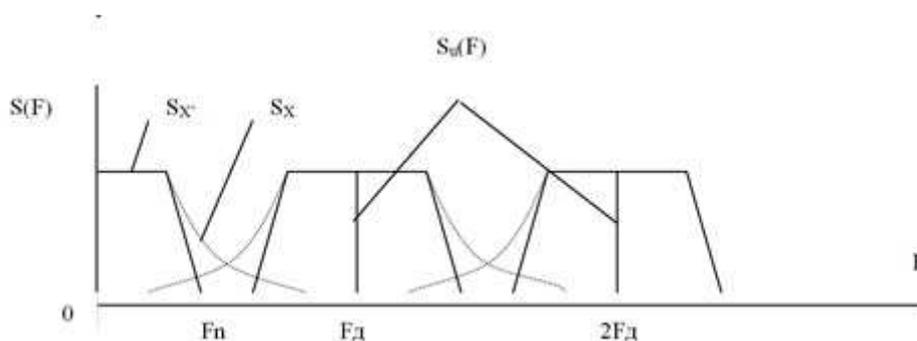


Рис. 1.2. Амплитудный спектр дискретизированного сигнала

В реальности спектры аналоговых сигналов звука и изображения имеют широкие полосы. Соседние участки спектра дискретизированного сигнала пересекаются (пунктирные линии). Следовательно, после дискретизации исходный сигнал $X(t)$ невозможно восстановить без искажений. Вот почему всегда перед дискретизацией сигнал пропускают через ФНЧ.

Однако при такой фильтрации часть высокочастотных компонентов из спектра сигнала удаляется: возникают его искажения. Желательно для повышения качества сигнала после цифроаналогового преобразования (ЦАП) повышать частоту дискретизации F_d , однако при этом растет объем потока данных.

Правильный выбор частоты осуществляется на основе теоремы Котельникова: «Если сигнал $X(t)$ имеет преобразование Фурье $X(j\Omega)$ такое, что $X(j\Omega) = 0$ при $\Omega \geq 2\pi F_n$, то сигнал $X(t)$ может быть восстановлен единственным образом по последовательности равноотстоящих отсчетов $X(nT)$, $-\infty < n < +\infty$, если частота дискретизации $F_d = 1/T > 2F_n$ ».

Если частота дискретизации в два раза больше максимальной граничной частоты спектра сигнала, то искажения сигнала после ЦАП за счет дискретизации отсутствуют.

При квантовании дискретных выборок сигнала осуществляется замена значения выборки номером ближайшего уровня квантования. За счет округления значений сигнала возникает погрешность δ округления (квантования) (рис. 1.3).

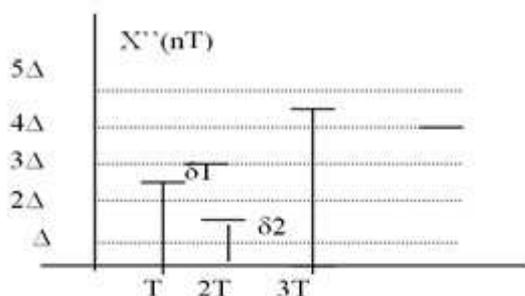


Рис. 1.3. Возникновение погрешности квантования сигнала

Квантованный сигнал X' можно рассматривать как сумму квантуемого сигнала X и ошибки δ округления

$$X' = X + \delta.$$

Если шаг Δ квантования намного меньше уровня квантуемого сигнала, то искажения сигнала, возникающие за счет округления, можно рассматривать как результат суммирования сигнала с так называемым шумом квантования, который присутствует в сигнале на выходе ЦАП. Шум квантования – белый шум с равномерным распределением плотности вероятности значений в интервале $[-\Delta/2, +\Delta/2]$. Мощность $P_{ш}$ (дисперсия $\sigma_{ш}^2$) шума квантования

$$P_{ш} = \sigma_{ш}^2 = \Delta^2 / 12.$$

Отношение q мощности сигнала P_c (дисперсии σ_x^2) к мощности шума квантования $P_{ш}$ (дисперсии $\sigma_{ш}^2$) характеризует степень искажений сигнала после ЦАП, возникающих при его квантовании,

$$q = P_c / P_{ш} = \sigma_x^2 / \sigma_{ш}^2 = 12\sigma_x^2 / \Delta^2.$$

Результат квантования – целое двоичное число, которое представляет собой номер уровня квантования. Количество разрядов числа B – важная характеристика квантователя. Если значение квантуемого сигнала меняется в пределах $(X_{\max} \dots X_{\min})$, то количество разрядов квантователя определяют с помощью выражения

$$\Delta 2^B \geq X_{\max} - X_{\min}.$$

При квантовании случайных сигналов возникает проблема определения максимального и минимального значений сигнала. В частности, речевой сигнал теоретически может принимать очень малые и очень большие значения. Плотность вероятностей значений речевого сигнала удовлетворительно описывает распределение Лапласа

$$p(x) = \frac{1}{\sigma_x \sqrt{2}} \exp\left(-\frac{|x|\sqrt{2}}{\sigma_x}\right),$$

где σ_x – среднеквадратическое отклонение. При распределении Лапласа почти все значения сигнала находятся в пределах $-4\sigma_x \leq x \leq 4\sigma_x$. Лишь 0,35 % всех значений сигнала не укладываются в данный диапазон.

Из приведенных рассуждений следует необходимость двухстороннего ограничения сигнала по максимуму и минимуму перед квантованием. Пороги ограничения сигнала по минимуму и максимуму рассматривают при расчете как минимальное и максимальное значения случайного сигнала.

Чем больше степень ограничения, тем меньше «размах» сигнала перед квантованием и, следовательно, меньше число уровней квантования. Снижение количества уровней квантования обуславливает возможность уменьшения числа разрядов квантователя и соответственного снижения скорости потока данных либо при неизменном количестве разрядов квантования обуславливает возможность уменьшения шага квантования, что повышает отношение сигнал-шум. Однако слишком сильное ограничение сигнала приводит к росту искажений сигнала после ЦАП.

Все приведенные формулы соответствуют стационарному сигналу. Речевой сигнал является нестационарным. Поэтому при расчете квантователя шаг квантования определяется заданным отношением сигнал-шум квантования при минимальном значении уровня сигнала, а число уровней квантования определяется «размахом» сигнала при максимальном значении его уровня.

Порядок расчета квантователя

В целях уменьшения трудоемкости лабораторной работы считаем заданный сигнал стационарным.

Сначала по заданному относительному количеству (0,35 %) больших и малых значений сигнала, которые находятся вне рабочего диапазона квантования, определяют уровни двухстороннего ограничения сигнала. Затем по заданному отношению сигнал-шум квантования находят число разрядов квантователя и шаг квантования. Обычно данное отношение сигнал-шум $q_{\text{зад}}$ указывают в децибелах (дБ)

$$q_{\text{зад}} = 10 \lg(q_{\text{зад}}).$$

Однако в формулах используют отношение $q_{\text{зад}}$, выраженное в «размах» по мощности. Переход от децибел к «разам» по мощности осуществляется согласно выражению

$$q_{\text{зад}} = 10^{(q_{\text{зад}}/10)}.$$

Определяют первичное значение шага квантования

$$\Delta = \sqrt{\frac{12\sigma_x^2}{q_{\text{зад}}}}.$$

Количество двоичных разрядов квантователя находят из условия

$$2^B \geq (X_{\text{max}} - X_{\text{min}}) \sqrt{\frac{q_{\text{зад}}}{12\sigma_x^2}} = \frac{X_{\text{max}} - X_{\text{min}}}{2\sigma_x} \sqrt{\frac{q_{\text{зад}}}{3}},$$
$$B \geq \log_2 \left[\frac{X_{\text{max}} - X_{\text{min}}}{2\sigma_x} \sqrt{\frac{q_{\text{зад}}}{3}} \right].$$

Величина B должна быть целым числом, поэтому полученный результат округляют в большую сторону. После определения количества разрядов квантователя рассчитывают число уровней квантования

$$N = 2^B$$

и уточняют значение шага квантования (вторичное значение шага квантования)

$$\Delta_2 = (X_{\max} - X_{\min})/N.$$

Разность (диапазон значений сигнала) $X_{\max} - X_{\min}$, заданное отношение сигнал-шум $q_{\text{зад}}$ и мощность σ_x^2 квантуемого сигнала – исходные данные для расчета квантователя. Результат расчета – целое число разрядов квантователя и шаг квантования. Номера уровней квантования определяют с помощью операции округления $n_i = \text{round}(X_i)$.

Номера n_i уровней квантования передают по каналу связи. На приемной стороне сначала проводят умножение шага квантования на номера уровней квантования, а затем полученный дискретизированный во времени сигнал пропускают через ФНЧ – формируется аналоговый сигнал, т. е. проводится ЦАП сигнала. Отсчеты дискретизированного сигнала на приемной стороне канала связи

$$X_{i \text{ ЦАП}} = n_i \Delta_2.$$

Шум квантования экспериментально определяют как разность отсчетов дискретизированных сигналов (исходного и на приемной стороне канала связи)

$$L_i = X_{i \text{ ЦАП}} - X_i.$$

Для проверки правильности расчета экспериментально определяют отношение мощности квантуемого сигнала к мощности шума квантования.

Порядок выполнения работы (реализуется строками сценария *Matlab*)

1. Читаем заданный звуковой файл, строим гистограмму значений сигнала и определяем пороги двустороннего ограничения сигнала, используя следующий сценарий.

```
[data,fs]=wavread('m5_9.wav');%чтение звукового файла
```

```

%[data,fs]=audioread('m5_9.wav');%чтение звукового файла для
%поздних версий Matlab
Procnt=0.35e-2; % относительное число отсчетов сигнала, которые
%попадают под ограничение
L=length(data);
L_Procnt=L*Procnt; %число отсчетов сигнала, которые попадают
%под ограничение
hist(data,15);grid on;title('Гистограмма входного сигнала');

```

В результате запуска сценария строится гистограмма значений сигнала (рис. 1.4) и определяется максимально допустимое количество (L_Procnt) отсчетов сигнала, значения которых находятся за порогами ограничения сигнала. Данное количество отсчетов следует из заданного процентного отношения отсчетов сигнала (0,35 %), находящихся за порогами ограничения, к общему количеству отсчетов сигнала.

Гистограмма – это графическое представление функции плотности вероятности значений сигнала, построенное по выборке его значений. Для построения гистограммы используют функцию $hist(data, n)$. Здесь $data$ – анализируемые данные; n – число интервалов, на которые разбивается весь диапазон значений анализируемых данных. Гистограмма значений сигнала показывает зависимость количества отсчетов сигнала, значения которых находятся в заданных интервалах значений, от этих интервалов.

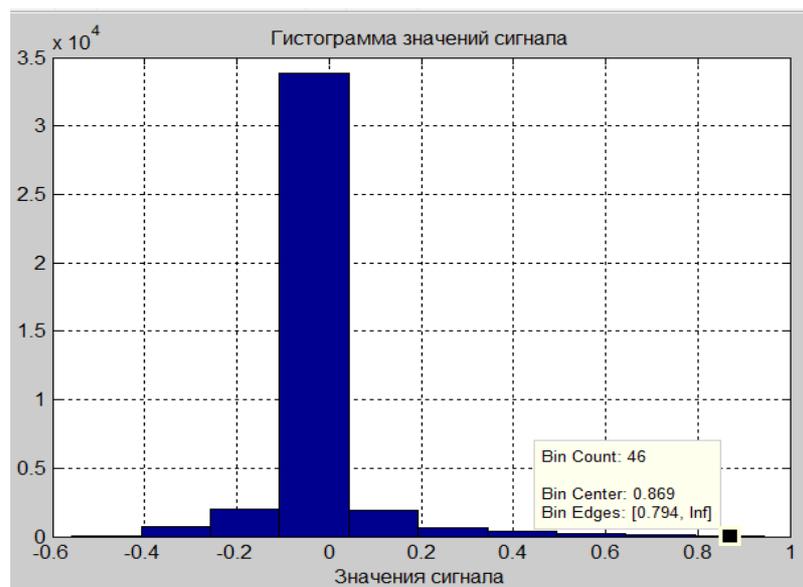


Рис. 1.4. Гистограмма значений сигнала

2. По гистограмме находим нижний и верхний пороги ограничения сигнала следующим образом. С помощью инструмента *Tools/Data Cursor* определяем суммарное количество отсчетов сигнала, которые находятся в нескольких крайних интервалах слева и справа от центра гистограммы (переменная *Bin Count*).

В приведенном рисунке *Bin Count* = 46. Таким образом, в самый правый интервал гистограммы попадает 46 отсчетов сигнала. Левая граница интервала *Bin Edges* = 0,794, а правая *Bin Edges* = inf – не определена, так как данный интервал крайний. Если верхний порог ограничения установить равным 0,794, то 46 самых больших отсчетов сигнала станут равными данному порогу ограничения.

После определения порогов ограничения по гистограмме их в небольших пределах корректируем, чтобы точно задать число отсчетов сигнала, которые находятся вне порогов ограничения. С этой целью число *n* интервалов функции *hist(data, n)* увеличиваем.

3. В соответствии с найденными порогами ограничиваем сигнал, используя следующий сценарий.

```
porog_max=0.644;
porog_min=-0.5;
data_ogr=Ogrnich(data,porog_max,porog_min);%Ограничение сигнала
function [ data_ogr ] = Ogrnich( data,porog_max,porog_min )
%Функция ограничивает сигнал снизу и сверху
%data - входной сигнал
%porog_max,porog_min - верхний и нижний пороги ограничения
% data_ogr - выходной сигнал

L=length(data);
data_ogr=zeros(L,1);
for k=1:L
    if data(k)>=porog_max
        data_ogr(k)=porog_max;
    elseif data(k)<=porog_min;
        data_ogr(k)=porog_min;
    else
        data_ogr(k)=data(k);
    end
end
end
```

4. Проводим расчет квантователя в соответствии с приведенными выше выражениями, используя соответствующую функцию.

```

q_dB=20;
[B0_ogr, B_ogr,delta_ogr,data_q_ogr]=RaschetQvant4(data_ogr,q_dB);
function [B0,B,delta,data_q]=RaschetQvant4(data,q_dB)
% Расчет квантователя
% data - квантуемый сигнал
% q_dB - заданное отношение с/ш в дБ
% B0 - дробное число разрядов
% B - число разрядов квантователя
% delta - уточненный шаг квантования
% data_q - поток данных после квантования (выход АЦП)

q=10^(q_dB/10);%заданное отношение с/ш в "размах" по мощности
razmakh=max(data)-min(data);%размах сигнала
P=var(data);%средняя мощность сигнала
delta0=sqrt(12*P/q);%первичный шаг квантования
B0=log2(razmakh/delta0);%дробное число разрядов
B=ceil(B0);%целое число разрядов
N=2^B;%число уровней квантования
delta=razmakh/N;%уточненный шаг квантования
data_q=round(data/delta);%номера уровней квантования

```

5. Проводим проверку правильности расчета, определяя отношение сигнал-шум квантования после ЦАП сигнала на приемной стороне линии связи.

```

dataDACogr=data_q_ogr*delta_ogr;%сигнал после ограничения
и квантования
shum_ogr=dataDACogr-data_ogr;%шум квантования ограничен.
сигнала
P_shum_ogr=var(shum_ogr);%дисперсия шума квантования
P_s=var(data);%дисперсия сигнала
q_shum_ogr=P_s/P_shum_ogr;%фактическое значение с/ш
q_dB_ogr=10*log10(q_shum_ogr);%фактическое значение с/ш (дБ)

```

6. Оцениваем качество сигнала на приемной стороне линии связи путем сравнения его с исходным сигналом на передающей стороне.

Сравниваем «осциллограммы» сигналов, а также результаты их прослушивания. После прослушивания сигнала на приемной стороне линии связи оцениваем разборчивость речи и узнаваемость диктора по голосу по шкале: «хорошо», «удовлетворительно», «плохо».

```
%оценка качества сигнала по «осциллограммам»  
t=0:1/fs:(length(data)-1)/fs;  
figure;plot(t,data);grid on;title('Сигнал до ограничения');  
figure;plot(t,data_ogr);grid on;title('Сигнал после ограничения');  
figure; plot(t,dataDACogr);grid on;title('Сигнал после ограничения и  
квантования');  
  
%прослушивание входного сигнала  
soundsc( data,fs);  
%прослушивание входного сигнала после ограничения  
soundsc( data_ogr,fs);  
%прослушивание сигнала после ограничения и квантования  
%на приемной стороне линии связи  
soundsc( dataDACogr,fs);
```

7. Чтобы оценить влияние процедуры ограничения сигнала на результаты расчета квантователя, повторяем пункты 4 и 5 для сигнала при отсутствии процедуры ограничения.

Содержание отчета о работе

- Номер варианта задания и исходные данные для расчета.
- Цель работы.
- Методика расчета квантователя с расчетными формулами.
- Результаты расчета квантователя (скриншот окна переменных): целое и дробное значения числа разрядов, уточненный шаг квантования, пороги ограничения, фактическое значение отношения сигнал-шум.
 - Гистограмма значений исходного сигнала, фактическое число отсчетов сигнала, попадающих под ограничение, полученное по результатам анализа гистограммы.
 - «Осциллограммы» сигналов: исходного, ограниченного, ограниченного и квантованного, по которым можно судить об искажениях сигнала за счет его ограничения и квантования.

- Выводы о влиянии шума квантования и порогов ограничения сигнала на результаты прослушивания сигналов и формы их «осциллограмм».
- Выводы о причинах отличия заданного и фактического значений отношения сигнал-шум.
- Листинги используемых сценариев и функций.

Контрольные вопросы

1. С какой целью проводят ограничение сигнала?
2. Как следует выбирать пороговые значения ограничений?
3. Каким образом с помощью гистограммы определяют уровни ограничения сигнала?
4. Почему нельзя устанавливать слишком большой и слишком малый шаг квантования?
5. Как правильно выбрать частоту дискретизации при АЦП?
6. Укажите причины искажений сигнала, возникающих при АЦП.
7. Укажите строки сценариев и функций, которые соответствуют расчетным формулам.

Варианты задания для выполнения работы

Номер варианта	Имя звукового файла	Отношение сигнал-шум, дБ
1	m0_4.wav	5
2	fm17.wav	6
3	m16.wav	7
4	m7.wav	8
5	m0_4.wav	9
6	fm17.wav	10
7	m16.wav	11
8	m7.wav	12
9	m0_4.wav	13
10	fm17.wav	14
11	m16.wav	15
12	m7.wav	16
13	m0_4.wav	17
14	fm17.wav	18
15	m16.wav	19
16	m7.wav	20
17	m0_4.wav	21

Лабораторная работа № 2

ИССЛЕДОВАНИЕ КВАНТОВАТЕЛЯ, ИСПОЛЬЗУЮЩЕГО КОМПАНДИРОВАНИЕ СИГНАЛОВ

Цель работы: для заданных звукового файла и отношения сигнал-шум квантования определить число двоичных разрядов квантователя и оценить эффективность использования компримирования сигналов при квантовании; сравнить результаты использования компримирования для A , μ – параметров компрессии.

Краткие теоретические сведения

Если уровень сигнала меняется во времени, то выбор шага квантования осуществляют с учетом минимального уровня, чтобы обеспечить требуемое отношение сигнал-шум квантования, а выбор количества уровней квантования – с учетом «размаха» сигнала максимального уровня. При этом величина шага квантования относительно невелика, а число уровней квантования становится очень большим, и номера уровней квантования представляются многоразрядными двоичными числами.

Стремление сократить число двоичных разрядов квантователя привело к идее увеличения шага квантования при повышении уровня сигнала. Если определенным образом увеличивать шаг квантования с ростом уровня сигнала, то число уровней квантования не будет меняться по сравнению с квантованием сигнала малого уровня. Останется неизменным и отношение сигнал-шум квантования.

Наиболее просто в вычислительном отношении идея увеличения шага Δ при росте уровня сигнала реализуется с помощью нелинейного преобразования сигнала с последующим равномерным его квантованием. Если функция нелинейного преобразования близка к логарифмической, то отношение сигнал-шум изменяется мало при изменении уровня сигнала. На рис. 2.1 показано, что равномерное квантование преобразованного сигнала эквивалентно неравномерному квантованию входного сигнала.

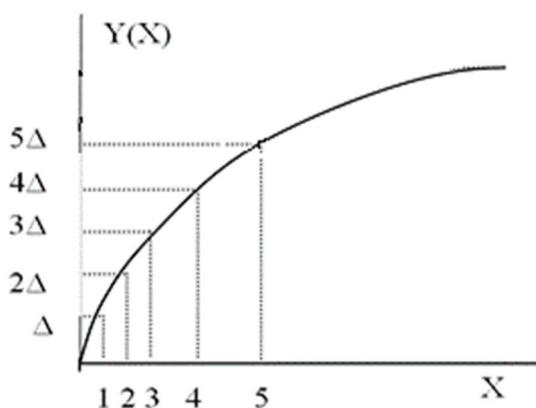


Рис. 2.1. Закон компрессии сигнала

Данное нелинейное преобразование превращает большие изменения входного сигнала X в малые изменения выходного сигнала Y (сигнал «сжимается» по уровню), поэтому оно получило название *компрессии сигнала*. На практике применяют законы компрессии, близкие к логарифмическому закону. В частности, в телефонии используют так называемый μ -закон (μ -law) компрессии

$$y[n] = F\{x[n]\} = \frac{\ln\left(1 + \mu \frac{|x[n]|}{X_{\max}}\right)}{\ln(1 + \mu)} X_{\max} \text{sign}(x[n]),$$

где $\text{sign}(x[n]) = \begin{cases} 1, & x[n] \geq 0 \\ -1, & x[n] < 0 \end{cases}$.

Здесь функцию $\text{sign}(x[n])$ используют, чтобы распространить данный метод преобразования не только на положительные, но и на отрицательные значения $x[n]$.

После квантования сжатого сигнала уменьшенный по объему цифровой поток данных можно использовать либо для передачи по

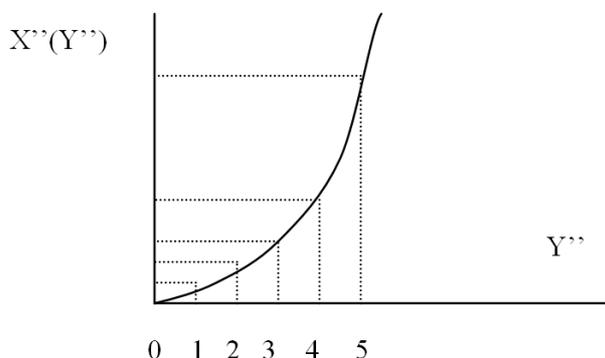


Рис. 2.2. Закон экспандирования сигнала

каналу связи, либо для записи на носитель информации. При необходимости восстановления исходного сигнала проводят процедуру обратной компрессии – *экспандирование сигнала*. На рис. 2.2 изображена характеристика закона экспандирования.

Закон экспандирования для μ -закона компрессии определяется выражением

$$z[n] = \frac{Y_{\max}}{\mu} \left[\exp\left(\frac{|y[n]|}{Y_{\max}} \ln(1 + \mu)\right) - 1 \right] \text{sign}(y[n]).$$

На практике также используют A -закон компрессии

$$y[n] = \left\{ \begin{array}{ll} \frac{A|x[n]|}{1 + \ln(A)} \text{sign}(x[n]), & |x[n]| < \frac{1}{A} \\ \frac{1 + \ln(A|x[n]|)}{1 + \ln(A)} \text{sign}(x[n]), & \frac{1}{A} \leq |x[n]| < 1 \end{array} \right\}.$$

Соответствующий закон экспандирования

$$z[n] = \left\{ \begin{array}{ll} |y[n]| \frac{1 + \ln(A)}{A} \text{sign}(y[n]), & 0 \leq |y[n]| < \frac{1}{1 + \ln(A)} \\ \frac{\exp(|y[n]|(1 + \ln(A))) - 1}{A} \text{sign}(y[n]), & \frac{1}{A} \leq |y[n]| < 1 \end{array} \right\}.$$

В приведенных выражениях величины A , μ – параметры компрессии.

Совокупность операций компрессии и экспандирования называют *компандированием*. Если звуковой сигнал необходимо сохранить или передать в цифровом виде, то процесс его обработки с использованием компандирования выглядит следующим образом: компрессия сигнала, квантование с постоянным шагом, передача по линии связи (или сохранение данных), умножение числовых данных на шаг квантования, экспандирование.

Использование компандирования позволяет уменьшить число разрядов квантователя и диапазон изменения отношения сигнал-шум для различных сегментов (фрагментов) сигнала.

Порядок выполнения работы (реализуется строками сценария *Matlab*)

1. Анализируем влияние параметра μ компрессии на формы законов компрессии и экспандирования (рис. 2.3), используя малое и большие значения параметра $\mu_1 = N_0 \times 2$ и $\mu_2 = N_0 \times 40$, где N_0 – номер варианта задания. Нижеследующий сценарий необходимо запустить два раза с заданными значениями параметра μ . Следует сделать вывод о зависимости степени нелинейности закона компрессии от параметра μ .

```
mu=200;%параметр компрессии  
data=-100:100;%тестовый сигнал  
dataCmrps = compress2( data,mu );%компрессия
```

```
function dataCmrps = compress2( data,mu )  
%компрессия входного сигнала  
%data входной сигнал  
%mu параметр компрессии  
%dataCmrps выходной сигнал  
X_max=max(abs(data));  
dataCmrps=X_max*log(1+mu*abs(data)/X_max)/log(1+mu).*sign(data);  
end  
dataExpnd = expnd2( data,mu );%экспандирование
```

```
function dataExpnd = expnd2( data,mu )  
%экспандирование входного сигнала  
%data входной сигнал  
%mu параметр компрессии  
%data_Expnd выходной сигнал  
X_max=max(abs(data));  
dataExpnd=X_max/mu*(exp(abs(data)/X_max*log(1+mu))-  
1).*sign(data);  
end
```

```
subplot(2,1,1);plot(data,dataCmrps);title('mu-закон компрессии');grid on;  
subplot(2,1,2);plot(data,dataExpnd);title('mu-закон экспандирования');  
grid on;
```

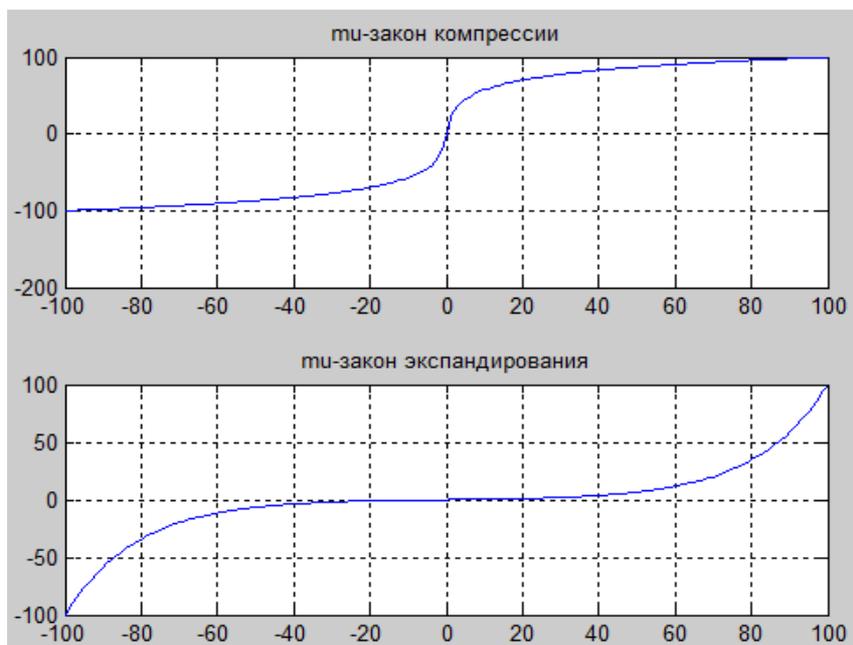


Рис. 2.3. Законы компрессии и экспандирования

2. Проводим расчет квантователя при использовании и без использования компрессии сигнала. Процедуру ограничения сигнала не применяем, чтобы оценить влияние только компрессии на результаты квантования. Функцию *RaschetQvant4* используем из лабораторной работы № 1.

```
%Определяем с/ш после квантования без компрессии
[B0,B,delta,data_q]=RaschetQvant4(data,q_dB);
dataDAC=data_q*delta;
shum=dataDAC-data;%шум квантования
P_shum=var(shum);% дисперсия шума квантования
P_s=var(data);%дисперсия сигнала
q_shum=P_s/P_shum;%фактическое значение с/ш
q_dB_m=10*log10(q_shum);%фактическое значение с/ш (дБ)
```

```
%Квантование с компрессией
mu=150;%параметр компрессии
dataCmrps = compress2( data,mu );
[B0Cmrps,BCmrps,deltaCmrps,data_qCmrps]=RaschetQvant4(dataCmrps,
q_dB);%квантование
data_outCmrps=data_qCmrps*deltaCmrps;%сигнал на выходе квант-ля
```

```

dataExpnd = expnd2( data_outCmrps,mi );%сигнал после экспандирования
shumExpnd=dataExpnd-data;%шум квантования
P_shumExpnd=var(shumExpnd);%дисперсия шума квантования
P_s=var(data);%дисперсия сигнала
q_shumExpnd=P_s/P_shumExpnd;%фактическое значение с/ш (в "разах")
q_dB_expExpnd=10*log10(q_shumExpnd);%фактическое
%(экспериментальное) среднее значение с/ш (дБ)

```

3. Оцениваем изменение отношения сигнал-шум с течением времени при наличии и отсутствии компандирования (рис. 2.4). С этой целью делим сигнал на отдельные сегменты (кадры) и для каждого кадра определяем отношение сигнал-шум. Анализируем также изменения формы сигнала.

```

n_sgmnt=200;%размер сегмента
[q_dB_sgmnt] = SNRsgmnt4(n_sgmnt,data,dataDAC);
[q_dB_Expnd_sgmnt] = SNRsgmnt4(n_sgmnt,data,dataExpnd);
figure; subplot(2,1,1); stem(q_dB_sgmnt);title('С/Ш при равномерном
квантовании (по сегментам)'); grid on;
subplot(2,1,2); stem(q_dB_Expnd_sgmnt);title('С/Ш при компандировании
(по сегментам)'); grid on;
figure; subplot(2,1,1); stem(dataDAC);title('Сигнал при равномерном
квантовании'); grid on;
subplot(2,1,2); stem(dataExpnd);title('Сигнал при квантовании
с компандированием'); grid on;

```

```

function [q_dB_sgmnt] = SNRsgmnt4(n_sgmnt,data,data_out)
%определяется с/ш для каждого сегмента сигнала
%n_sgmnt - размер сегмента сигнала
%data - данные входного сигнала квантователя
%data_out - данные выходного сигнала квантователя
%q_dB_sgmnt - отношение с/ш для каждого сегмента сигнала

```

```

data_sgmnt=buffer(data,n_sgmnt);%сегменты квантуемого сигнала
data_out_sgmnt=buffer(data_out,n_sgmnt);%сегменты сигнала на вых.
квант-ля
shum_sgmnt=data_out_sgmnt-data_sgmnt;%шум квантования

```

$P_s_sgmnt = \text{var}(data_sgmnt);$ % мощность квантуемого сигнала по сегментам

$P_shum_sgmnt = \text{var}(shum_sgmnt);$ % дисперсия шума квантования

$q_sgmnt = P_s_sgmnt / P_shum_sgmnt;$ % фактическое значение с/ш

$q_dB_sgmnt = 10 * \log_{10}(q_sgmnt);$ % фактическое значение с/ш (в дБ)

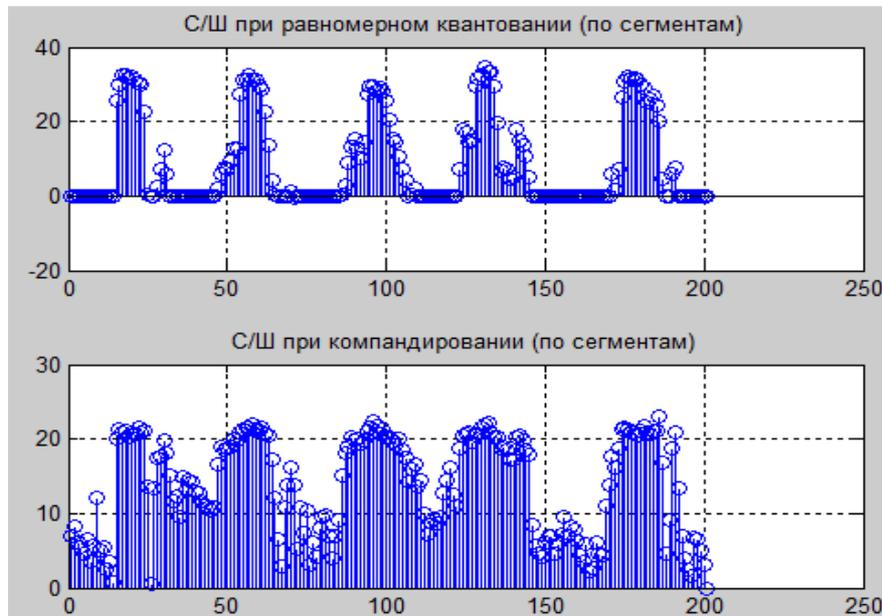


Рис. 2.4. Изменение отношения сигнал-шум во времени

По результатам эксперимента определяем максимальное и минимальное значения отношения сигнал-шум, а также изменения формы сигнала при наличии и отсутствии компрессирования.

4. Повторяем пункты 2 и 3 для другого значения параметра компрессии.

5. Сравниваем результаты использования A - и μ -законов компрессии. С этой целью повторяем пункты 1 – 4 для A -закона компрессии, используя соответствующие функции компрессии и экспандирования.

```
function dataCmrps = compressA2(data,A)
```

```
%компрессия входного сигнала
```

```
%data - входной сигнал
```

```
%A - параметр компрессии
```

```
%dataCmrps - выходной сигнал
```

```

X_max=max(abs(data));
data1=data/X_max;
C1=abs(data1)<(1/A);
C2=1-C1;
dataCmrps1=X_max*abs(data1)*A/(1+log(A)).*sign(data);

```

```

dataCmrps2=X_max*(1+log(abs(A*data1+1e-
10)))/(1+log(A)).*sign(data);
dataCmrps=C1.*dataCmrps1+C2.*dataCmrps2;

```

```

function dataExpnd = expandA2(data,A)
%Экспандирование входного сигнала
%data_log входной сигнал
%A параметр компрессии
%dataExpnd выходной сигнал
%
X_max=max(abs(data));
data1=data/X_max;
C1=abs(data1)<(1/(1+log(A)));%Матрица C содержит единицы, когда
неравенство справедливо, и нули – в противном случае
dataExpnd1=X_max*(abs(data1)*(1+log(A))/A).*sign(data);
C2=1-C1;
dataExpnd2=(X_max*(1/A)*exp(abs(data1)*(1+log(A))-1)).*sign(data);
dataExpnd=C1.*dataExpnd1+C2.*dataExpnd2;

```

Содержание отчета о работе

- Номер варианта задания и исходные данные для расчета.
- Цель работы.
- Результаты экспериментов (скриншот окна переменных): целое и дробное значения числа разрядов, фактическое значение отношения сигнал-шум.
 - «Осциллограммы» сигналов: квантованного, квантованного с использованием компандирования, по которым можно судить об искажениях сигнала за счет его ограничения и квантования.
 - Графики зависимости отношения сигнал-шум от номера сегмента.

- Результаты сравнения A -, μ – законов компрессии.
- Выводы по результатам экспериментов.
- Листинги функций и сценариев.

Контрольные вопросы

1. С какой целью используют компрессию сигналов при квантовании?
2. Как зависит сегментное отношение сигнал-шум от параметра компрессии?
3. Как следует менять шаг квантования в зависимости от уровня сигнала, чтобы стабилизировать сегментные значения сигнал-шум?
4. Почему используются разные законы компрессии?

Варианты задания для выполнения работы

Номер варианта	Имя звукового файла	Отношение сигнал-шум, дБ
1	m0_4.wav	10
2	fm17.wav	12
3	m16.wav	14
4	m7.wav	16
5	m0_4.wav	18
6	fm17.wav	20
7	m16.wav	22
8	m7.wav	24
9	m0_4.wav	26
10	fm17.wav	28
11	m16.wav	30
12	m7.wav	32
13	m0_4.wav	34
14	fm17.wav	36
15	m16.wav	38
16	m7.wav	40
17	m0_4.wav	42

Следует использовать два значения параметра μ компрессии для μ -закона компрессии: $\mu_1 = N_0 \times 2$ и $\mu_2 = N_0 \times 40$, где N_0 – номер варианта задания.

Лабораторная работа № 3

ИССЛЕДОВАНИЕ АДАПТИВНОГО КВАНТОВАТЕЛЯ

Цель работы: для заданных звукового файла и отношения сигнал-шум квантования определить число двоичных разрядов адаптивного квантователя, отношение сигнал-шум квантования; сравнить результаты использования адаптивного квантователя с результатами использования квантователя с командированием.

Краткие теоретические сведения

Квантователь с постоянным шагом квантования обеспечивает большее отношение сигнал-шум по сравнению с командированием при условии согласования его с уровнем входного сигнала. Под согласованием понимают точное соответствие шага квантования уровню сигнала и заданному отношению сигнал-шум. Отсюда следует целесообразность изменения шага квантования в соответствии с уровнем сигнала. Квантователи с изменяемым шагом называют *адаптивными квантователями*.

Наиболее целесообразно использовать такие квантователи в случае с нестационарными сигналами, когда уровень сигнала меняется во времени. В этом случае отношение сигнал-шум, наоборот, мало меняется во времени, следовательно, различные фрагменты нестационарного сигнала имеют примерно одинаковый уровень искажений в отличие от квантователей с постоянным шагом квантования.

При использовании адаптивных квантователей сигнал разбивается на отдельные фрагменты (сегменты). Каждый сегмент содержит M отсчетов дискретизированного сигнала. Шаг квантования $\Delta(n)$ устанавливают прямо пропорционально оценке $\hat{\sigma}_x$ среднеквадратического значения сигнала, определяемой на сегменте. Оценка $\hat{\sigma}_x$ – это квадратный корень из оценки дисперсии, которую определяют на интервале времени в зависимости от степени стационарности сигнала. Оценку дисперсии находят по формуле

$$\hat{\sigma}_x^2(n) = \frac{1}{M} \sum_{m=n}^{n+M-1} x^2(m),$$

где M – количество отсчетов сигнала, по которым определяют оценку; $x(m)$ – значения сигнала в дискретные моменты времени.

Чем быстрее меняется дисперсия сигнала во времени, тем меньше должен быть интервал времени (меньше значение M), на котором производят оценку.

Следует отметить, что в данном случае необходимо передавать или хранить информацию не только об отсчетах сигнала, но и о шаге квантования Δ : так формируется дополнительный поток данных, который объединяется с основным потоком в кодере канала связи (рис. 3.1). Но поскольку оценка $\hat{\sigma}_x$ меняется намного медленнее значений самого сигнала, то и информация о шаге передается с бóльшим периодом дискретизации, нежели информация об уровне сигнала.

Передача данных о шаге $\Delta[n]$ осуществляется через каждые M отсчетов. Следовательно, поток данных о шаге $\Delta[n]$ намного меньше потока, соответствующего $x(n)$. Следует также учесть, что после дискретизации шага квантования проводится его квантование. Выбор параметров квантования шага приводит к дополнительному снижению потока данных.

Помимо увеличения отношения сигнал-шум адаптация шага квантования обеспечивает и обработку сигнала с широким динамическим диапазоном при мало изменяющемся отношении сигнал-шум. Схема адаптивного квантователя с изменяющимся шагом квантования представлена на рис. 3.1.

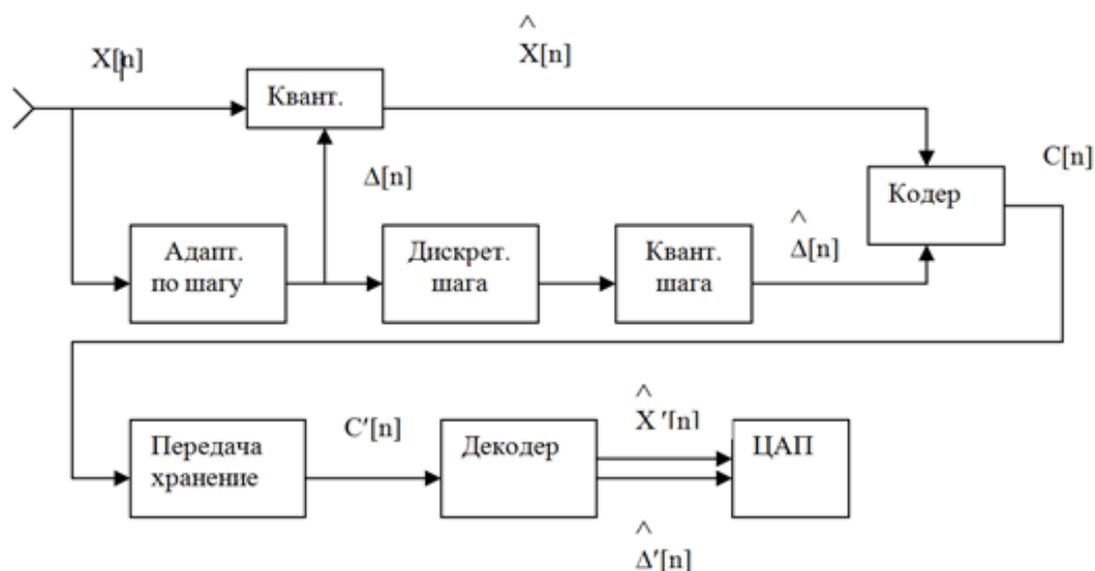


Рис. 3.1. Схема передачи данных при использовании адаптивного квантователя

Использование адаптивных квантователей позволяет сократить число разрядов квантователя по сравнению с компандированием при равных значениях отношения сигнал-шум квантования. Кроме того, адаптивный квантователь обеспечивает примерно одинаковый уровень искажений как для сегментов с большой мощностью, так и для мало-мощных сегментов.

Для исследования степени искажений целесообразно оценивать отношение сигнал-шум на каждом сегменте сигнала. В качестве численной меры искажений сигнала можно использовать среднее значение сегментного отношения сигнал-шум и его среднеквадратическое отклонение от среднего значения.

Порядок выполнения работы (реализуется строками сценария *Matlab*)

1. Читаем заданный звуковой файл.

```
[data,fs]=wavread('m5_9.wav');
```

2. Полученные данные разбиваем на сегменты.

```
n_sgmnt=200;%размер сегмента сигнала,  
%на котором определяется шаг квантования  
%сегментированные данные  
data_sgmnt=buffer(data,n_sgmnt);
```

3. Для каждого сегмента с учетом заданного отношения сигнал-шум проводим расчет квантователя с постоянным шагом квантования. В результате расчета для каждого сегмента определяем: количество разрядов квантователя, шаг квантования и номера уровней квантования.

Поскольку изменять разрядность квантования для каждого сегмента нельзя, то определяем среднее по всем сегментам количество разрядов.

```
q=10^(q_dB/10);%заданное отношение с/ш в "разах" по мощности  
razmakh=max(data_sgmnt)-min(data_sgmnt);%размах сигнала  
%по сегментам
```

```

P=var(data_sgmnt);%мощность сигнала
%по сегментам
delta0=sqrt(12*P/q);%первичный шаг квантования
%по сегментам
B0=log2(razmakh./delta0);%дробное число разрядов
%по сегментам
B0m=mean(B0);%Среднее по всем сегментам число разрядов
B=ceil(B0m);%целое число разрядов
N=2.^B;%число уровней квантования
delta=razmakh./N;%уточненный шаг квантования
%по сегментам
delta_rep= repmat(delta,n_sgmnt,1);%размер матрицы
%delta_rep равен размеру матрицы data_sgmnt
data_q=round(data_sgmnt./delta_rep);%номера уровней квантования
%по сегментам

```

4. Определяем сигнал на выходе ЦАП.

```

dataDACsgmnt=data_q.*delta_rep;%сегментированный сигнал на
выходе ЦАП
dataDAC=dataDACsgmnt(:);%объединение сегментов

```

5. Анализируем отношение сигнал-шум сигнала на выходе ЦАП. Устанавливаем длительность сегмента для анализа отношения сигнал-шум. Для каждого сегмента определяем отношение сигнал-шум.

```

n_sgmnt2=200;%размер сегмента сигнала
%для определения с/ш

%сегментированный входной сигнал
data_sgmnt2=buffer(data,n_sgmnt2);
%сегментированный сигнал на выходе ЦАП
dataDAC_sgmnt2=buffer(dataDAC,n_sgmnt2);

%сегментированный шум
shum_sgmnt2=(dataDAC_sgmnt2(:,1:length(data_sgmnt2))-
data_sgmnt2);
%мощность сегментированного сигнала

```

```

Ps_sgmnt2=var(data_sgmnt2);
%мощность сегментированного шума
Pshum_sgmnt2=var(shum_sgmnt2);
%посегментно определяем отношение с/ш
q_sgmnt02=Ps_sgmnt2./Pshum_sgmnt2;
q_sgmnt2=q_sgmnt02(1:end-1);
qdB_sgmnt2=10*log10(q_sgmnt2); %с/ш (дБ)

```

6. Определяем среднее значение и среднеквадратическое отклонение для совокупности сегментных отношений сигнал-шум для случаев, когда размер сегмента $n_sgmnt=400$ и $n_sgmnt=600$.

```

mean_qdB_sgmnt2=mean(qdB_sgmnt2);
std_qdB_sgmnt2=std(qdB_sgmnt2);

```

7. Для сравнения результатов адаптивного квантования с квантованием при постоянном шаге квантования определяем отношение сигнал-шум общее и сегментное для постоянного шага квантования.

```

%При заданном отношении с/ш определяем
%число разрядов квантователя с постоянным шагом
q=10^(q_dB/10);%заданное отношение с/ш в "разах" по мощности
razmakh00=max(data)-min(data);%размах сигнала
P00=var(data);%средняя мощность сигнала
delta00=sqrt(12*P00/q);%первичный шаг квантования
B00=log2(razmakh00/delta00);%дробное число разрядов
V00=ceil(B00);%целое число разрядов
N00=2^V00;%число уровней квантования
delta000=razmakh00/N00;%уточненный шаг квантования
data_q00=round(data/delta000);%номера уровней квантования
dataDAC00=data_q00*delta000;

%Определяем отношение с/ш для
%квантователя с постоянным шагом
shum00=dataDAC00-data;%шум квантования
P_shum00=var(shum00);% дисперсия шума квантования
P_s00=var(data);%дисперсия сигнала
%q_shum=P_s/P_shum;%фактическое значение с/ш
q_dB_m00=10*log10(P_s00/P_shum00);%фактическое значение с/ш (дБ)
%Определяем отношение с/ш для каждого сегмента

```

```

%для квантователя с постоянным шагом
data_sgmnt2=buffer(data,n_sgmnt2);
shumSgmnt00=buffer(shum00,n_sgmnt2);
%Определяем мощность сигнала для каждого сегмента
Ps_sgmnt2=var(data_sgmnt2);
%Определяем мощность шума для каждого сегмента сигнала
Pshum_sgmnt200=var(shumSgmnt00);
qdB_sgmnt200=10*log10(Ps_sgmnt2./Pshum_sgmnt200);

```

8. Строим графики:

- зависимости мощности сигнала от номера сегмента;
- зависимости шага квантования от номера сегмента;
- зависимости отношения сигнал-шум (дБ) от номера сегмента;
- квантованного сигнала при постоянном шаге квантования;
- квантованного сигнала при адаптивном шаге квантования.

```

subplot(4,1,1);stem(Ps_sgmnt2);grid on;title('Мощность сигнала
посегментно');
subplot(4,1,2);stem(delta);grid on;title ('Изменение шага квантования
во времени');
subplot(4,1,3);stem(qdB_sgmnt2);grid on;title('Отношение с/ш
посегментно для адапт. квант-я');
subplot(4,1,4);stem(qdB_sgmnt200);grid on;title('Отношение с/ш
посегментно для квант. с пост. шагом');

```

Результаты адаптивного квантования представлены на рис. 3.2.

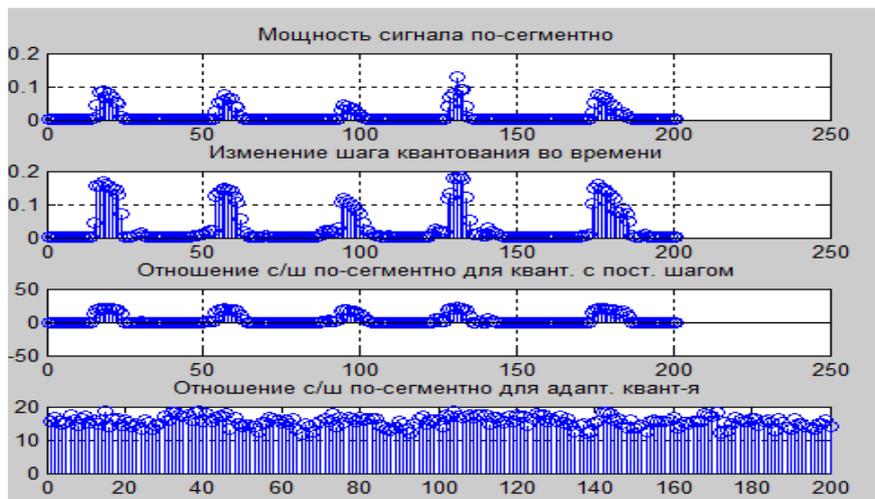


Рис. 3.2. Результаты адаптивного квантования

```

figure;
subplot(2,1,1);stem(dataDAC00);grid on;title('Сигнал после квантования
с постоянным шагом');
subplot(2,1,2);stem(dataDAC);grid on;title('Сигнал после квантования
с изменяемым шагом');

```

Результаты сравнения форм сигнала после адаптивного квантования и квантования с постоянным шагом представлены на рис. 3.3

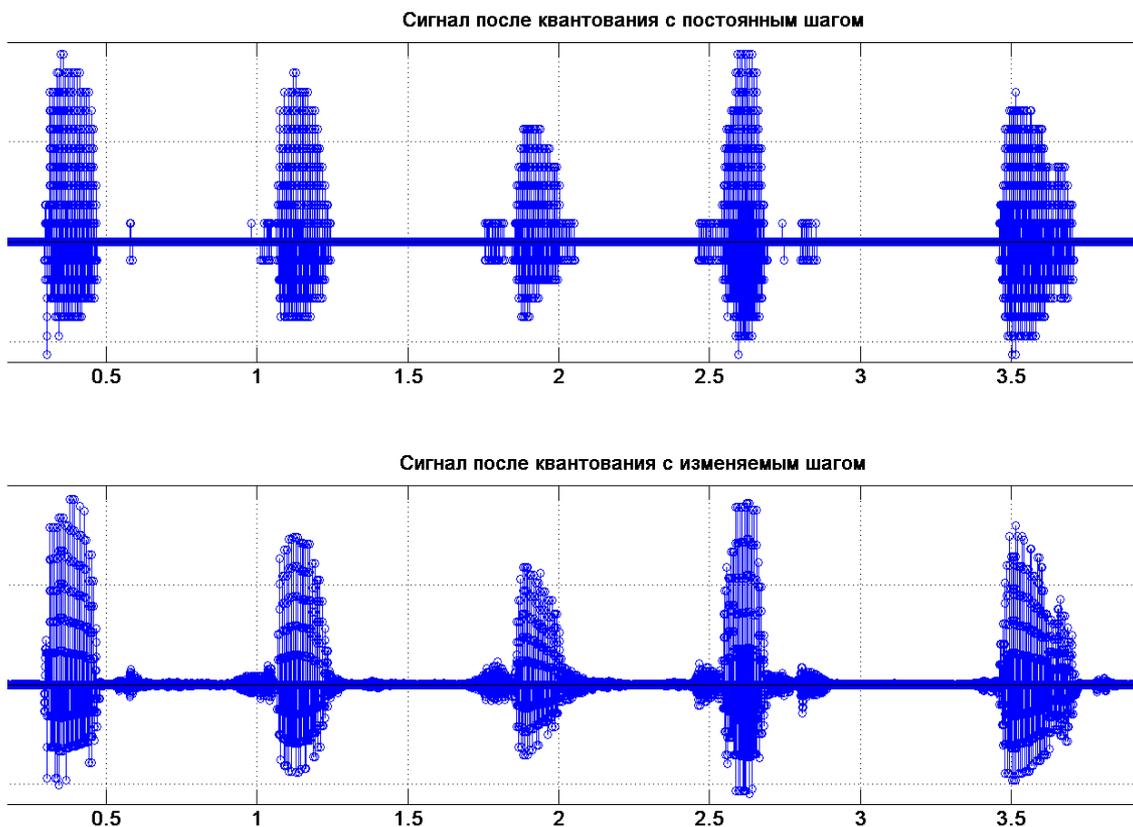


Рис. 3.3. Сравнение форм сигнала после адаптивного квантования и квантования с постоянным шагом

9. Осуществляем контрольное прослушивание входного и квантованного сигналов.

```

soundsc(dataDAC);%контрольное прослушивание
soundsc(data);

```

10. Используя результаты работы по исследованию квантования с компандированием, сравниваем результаты адаптивного квантования и квантования с компандированием.

Содержание отчета о работе

- Номер варианта задания и исходные данные для расчета.
- Цель работы.
- Структурная схема адаптивного квантователя.
- Результаты экспериментов (скриншот окна переменных): целое и дробное значения числа разрядов, фактическое значение отношения сигнал-шум.
 - График изменения шага квантования во времени.
 - Графики зависимостей отношения сигнал-шум (дБ) от номера сегмента для случаев равномерного квантования, квантования с компандированием и адаптивного квантования.
 - «Осциллограммы» сигналов для случаев равномерного квантования и адаптивного квантования.
 - Выводы по работе (сравнение качества сигналов при использовании равномерного квантования, квантования с компандированием и адаптивного квантования).

Контрольные вопросы

1. В чем заключается преимущество адаптивного квантователя по сравнению со случаями использования равномерного квантования и компандирования при квантовании?
2. Как меняется шаг квантования при изменении уровня квантуемого сигнала? (Показать результаты эксперимента).
3. Почему нельзя устанавливать размер сегмента, на котором определяется шаг квантования, очень большим и очень малым?
4. Какова скорость потока данных на выходе адаптивного квантователя (сумма потоков данных о сигнале и шаге квантования)?

Варианты заданий для выполнения работы

Номер варианта	Имя файла	Отношение сигнал-шум, дБ
1	m0_4.wav	10
2	fm17.wav	12
3	m16.wav	14
4	m7.wav	16
5	m0_4.wav	18
6	fm17.wav	20
7	m16.wav	22
8	m7.wav	24
9	m0_4.wav	26
10	fm17.wav	28
11	m16.wav	30
12	m7.wav	32
13	m0_4.wav	34
14	fm17.wav	36
15	m16.wav	38
16	m7.wav	40
17	m0_4.wav	42

Лабораторная работа № 4

ИССЛЕДОВАНИЕ РАЗНОСТНОГО КВАНТОВАТЕЛЯ

Цель работы: для заданных звукового файла и отношения сигнал-шум квантования определить зависимость коэффициента усиления квантователя от порядка предсказания предсказывающего фильтра.

Краткие теоретические сведения

Обычно уровень сигнала меняется достаточно медленно на интервале периода дискретизации. Следовательно, между соседними отсчетами существует значительная корреляция. Зная предыдущие отсчеты, можно с некоторой погрешностью предсказать значение следующего отсчета. Поэтому разность между реальным отсчетом и его

предсказанным значением, равная погрешности предсказания, значительно меньше значений самих отсчетов.

Если по каналу связи вместо отсчетов сигнала передавать значения погрешности предсказания, то можно сократить число уровней квантования, а следовательно, и количество разрядов квантователя. Такой подход используют при построении разностных квантователей (рис. 4.1).

В состав квантователя входит предсказывающий фильтр, который на основе ряда предыдущих значений последовательности $x^{\wedge}[n]$ предсказывает (формирует оценку) последующее значение $x^{\sim}[n]$. Квантованию подвергается разность между истинным значением $x[n]$ и предсказанным $x^{\sim}[n]$ – погрешность предсказания. Поскольку разность $d[n] = x[n] - x^{\sim}[n]$ меньше самих значений $x[n]$, то и разрядность квантователя при том же отношении сигнал-шум также меньше.

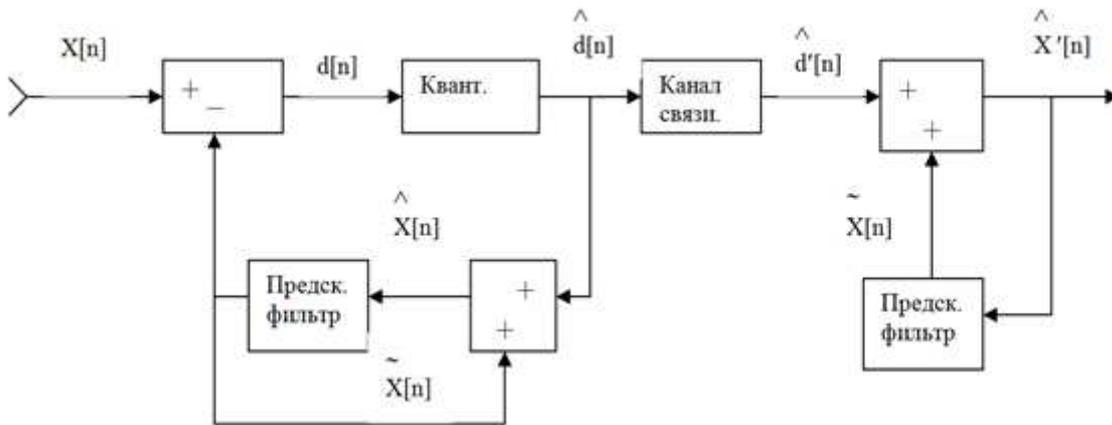


Рис. 4.1. Структура разностного квантователя

Пусть ошибка квантования равна $e[n]$, тогда квантованная разность

$$d^{\wedge}[n] = d[n] + e[n].$$

Покажем, что $x^{\wedge}[n] = x[n] + e[n]$, т. е. погрешность квантования $x[n]$ равна погрешности квантования $d[n]$. Квантованное значение

$$\begin{aligned} x^{\wedge}[n] &= x^{\sim}[n] + d^{\wedge}[n] = x^{\sim}[n] + d[n] + e[n] = x^{\sim}[n] + x[n] - x^{\sim}[n] + e[n] = \\ &= x[n] + e[n]. \end{aligned}$$

Если предсказание осуществить достаточно точно, то дисперсия разности $\sigma_d^2 \ll \sigma_x^2$. Определим отношение сигнал-шум квантования

$$P_c / P_{ш} = \sigma_x^2 / \sigma_e^2 = \sigma_x^2 / \sigma_d^2 \times \sigma_d^2 / \sigma_e^2 = G_p \times \sigma_d^2 / \sigma_e^2,$$

где $G_p = \sigma_x^2 / \sigma_d^2$ – коэффициент усиления системы, обусловленный разностным квантованием. Он определяет выигрыш в отношении сигнал-шум. А величина σ_d^2 / σ_e^2 обусловлена лишь характеристиками квантователя, который может быть равномерным, адаптивным, с компандированием.

Предсказывающий фильтр формирует выходной сигнал согласно следующему алгоритму:

$$\tilde{x}[n] = \sum_{k=1}^p a_k \hat{x}[n - k],$$

где p – порядок предсказания.

Используя оптимальные значения коэффициентов a_k линейного предсказания, обеспечивают минимум σ_d^2 . Оптимальные значения коэффициентов a_k полностью определяются видом нормированной автокорреляционной функции $\rho(\tau)$ сигнала $x[n]$.

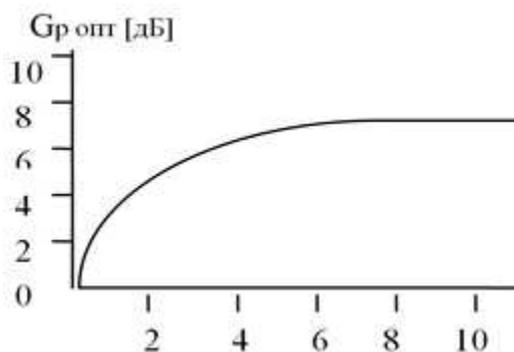


Рис. 4.2. Зависимость коэффициента усиления системы от порядка предсказания

При увеличении порядка предсказания выигрыш от применения разностного квантователя растет сначала быстро, а затем медленнее. При порядке предсказания $p > 5$ величина G_p практически не меняется (рис. 4.2).

Из графика $G_{p\text{опт}}(p)$, усредненного по нескольким речевым сигналам, видно, что при оптимальных коэффициентах предсказывающего фильтра выигрыш от применения разностного квантования достигает примерно 6 дБ, что эквивалентно добавлению одного разряда квантователю.

Поскольку корреляционная функция при смене звуков речи меняется, то меняются и оптимальные коэффициенты предсказывающего фильтра. Отсюда следует целесообразность использования адаптивного фильтра с подстройкой под конкретные звуки речи.

Особенности моделирования разностного квантователя

Поскольку обработка сигнала в предсказывающем фильтре проводится последовательно с каждым отсчетом сигнала, то необходимо получить доступ к внутреннему состоянию фильтра. При этом обработка последующего отсчета происходит с учетом результатов обработки предыдущего отсчета.

Функция *filter* позволяет задавать и считывать внутреннее состояние дискретного фильтра.

$$[y, z2] = \text{filter}(b, a, x, z1);$$

Здесь $z1$ – начальное внутреннее состояние фильтра (входной параметр); $z2$ – конечное внутреннее состояние фильтра (выходной параметр). Вектор коэффициентов $b(i)$ относится к нерекурсивной части фильтра, а вектор $a(i)$ – к рекурсивной части; n – порядок фильтра; x, y – входной и выходной сигналы соответственно.

Функция реализует рекурсивный дискретный фильтр в транспонированной форме (*direct transposed form II*), которая представлена на рис. 4.3.

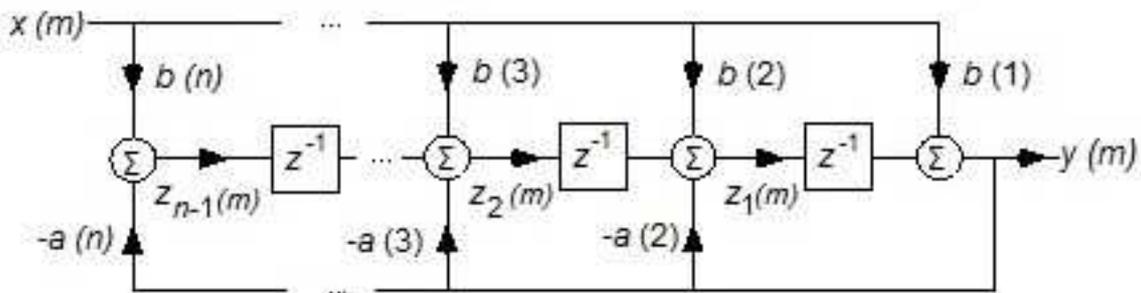


Рис. 4.3. Транспонированная форма рекурсивного дискретного фильтра

Возможность доступа к внутреннему состоянию фильтра позволяет реализовать обработку сигнала по частям.

% 1-й блок данных

```
[y(1:N), z] = filter(b, a, x(1:N));
```

% 2-й блок данных

```
[y(N+1:2*N), z] = filter(b, a, x(N+1:2*N), z);
```

По умолчанию предполагается нулевое внутреннее состояние фильтра.

Порядок выполнения работы (пример выполнения задания)
(реализуется строками сценария *Matlab*)

1. Устанавливаем заданные значения размеров сегментов сигнала, перекрытия соседних сегментов, порядка предсказания и отношения сигнал-шум квантования. Читаем заданный файл. При необходимости выделяем из него заданные участки аудиозаписи.

```
q_dB=20;  
ds=200;%размер сегмента (кол-во отсчетов)  
do=100;%размер перекрытия соседних сегментов  
pre_order=10;%порядок предсказания  
name='один.wav';% name-имя звукового файла  
[V,Fs]=wavread(name);
```

2. Используя функцию *RaschetQvant4* (см. лабораторную работу № 1), проводим расчет квантователя для определения шага *delta* квантования, который используют при разностном квантовании.

```
[B0, B, delta, data_q]=RaschetQvant4(V,q_dB);
```

3. Определяем параметры адаптивного предсказывающего фильтра – коэффициенты линейного предсказания (КЛП). Для повышения точности определения КЛП сигнал сначала проходит через предсказывающий фильтр. Далее сигнал разбивается на отдельные сегменты (блоки). Для каждого сегмента определяют корреляционную функцию, и по функции находят КЛП.

```
%предсказывающий фильтр  
xf=filter([1 -.95],1,V);  
%Разбиваем сигнал на блоки с указанным размером перекрытия  
xb=buffer(xf,ds,do);  
%каждый блок сигнала "взвешивается" окном Хэмминга  
[m,n]=size(xb);  
w=hamming(m);  
xh= repmat(w,1,n).*xb;
```

```

%Вычисляем корреляционные функции отдельных
%блоков (столбцов) сигнала
%вычисляем корреляционную функцию первого столбца
xk1=xcorr(xh(:,1),pre_order,'biased');
%вычисляем корреляционные функции других столбцов
if n>1
    for i=2:n
        xk1=cat(2,xk1,xcorr(xh(:,i),pre_order,'biased')); %объединяем
корреляционные функции
    end;
end
xk=xk1(pre_order+1:end,:);
%с помощью алгоритма Левинсона - Дарбина вычисляем
%вектор коэффициентов линейного предсказания - A0
%вектор коэффициенты частной корреляции - K
%средний квадрат ошибки предсказания - E
[A0,E,K]=levinson(xk);
%преобразуем коэфф. линейн. предсказания
%согласно стандарт. формы фильтра (нужен доступ к его
состоянию)
A=-A0;
A(:,1)=0;

```

4. Для определения разностного сигнала разбиваем входной сигнал на сегменты без перекрытия соседних сегментов, устанавливаем нулевыми начальное состояние предсказывающего фильтра и первого отсчета сигнала на входе фильтра. Последовательно, отсчет за отсчетом, пропускаем входной сигнал через адаптивный предсказывающий фильтр (каждому сегменту сигнала соответствует свой набор КЛП фильтра). Формируется разностный сигнал.

```

%определяем разностный сигнал
dataSegmnt=buffer(V,ds-do);
[m0,n0]=size(dataSegmnt);
[m1,n1]=size(A);
%Задаем начальное состояние
% предсказывающего фильтра

```

```

Z1=zeros(n1-1,1);
%Задаем значение первого отсчета
%сигнала на входе фильтра
start=0;
%сегменты квантованного речевого сигнала
quantv=zeros(m0,m1);
%сегменты квантованного разностного сигнала
s=zeros(m0,m1);
%числа, соответствующие результатам квантования
%сегментов разностного сигнала
dQuant=zeros(m0,m1);

for k3=1:m1
    segmnt=dataSegmnt(:,k3);
    sgmntA=A(k3,:);
    v=sgmnt;
    for k2=1:m0
        if k2==1
            [predictv,Z]=filter(sgmntA,1,start,Z1);
        else
            [predictv,Z]=filter(sgmntA,1,start2,Z);
        end
        d=v(k2)-predictv;
        dQuant(k2,k3)=round(d/delta);
        s(k2,k3)=dQuant(k2,k3)*delta;
        quantv(k2,k3)=predictv+s(k2,k3);
        start2=quantv(k2,k3);
    end
    start=quantv(k2,k3);
    Z1=Z;
end
end

```

5. Далее объединяем все сегменты квантованного разностного сигнала и получаем массив квантованного разностного сигнала (рис. 4.4). Аналогично формируем массив квантованного входного сигнала, образованного на передающей стороне из квантованного разностного сигнала (рис. 4.5).

$OUT_s = s(:)$; %квантованный разностный сигнал
 $OUT_v = quantv(:)$; %квантованный входной сигнал,
% сформированный на передающей стороне
%из квантованного разностного сигнала

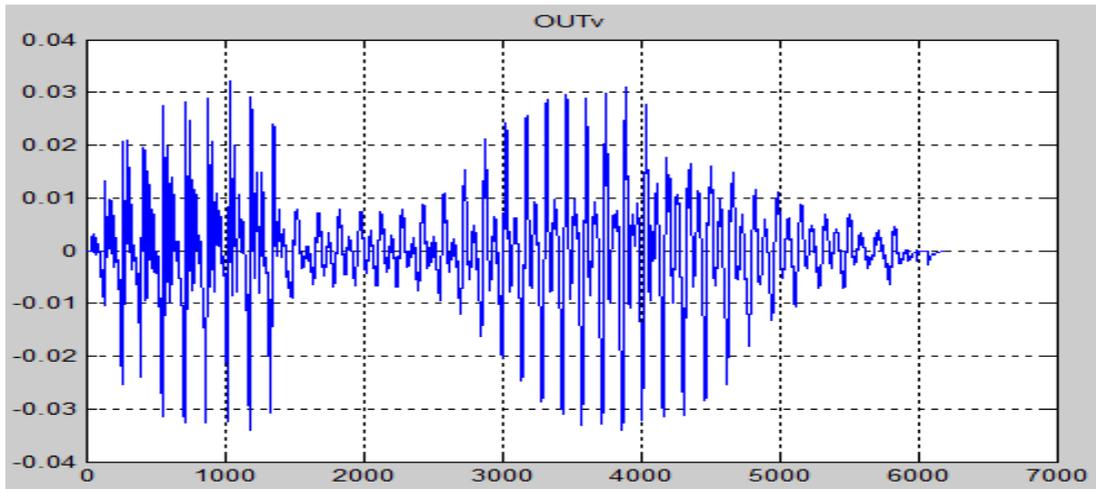


Рис. 4.4. Сформированный квантованный сигнал

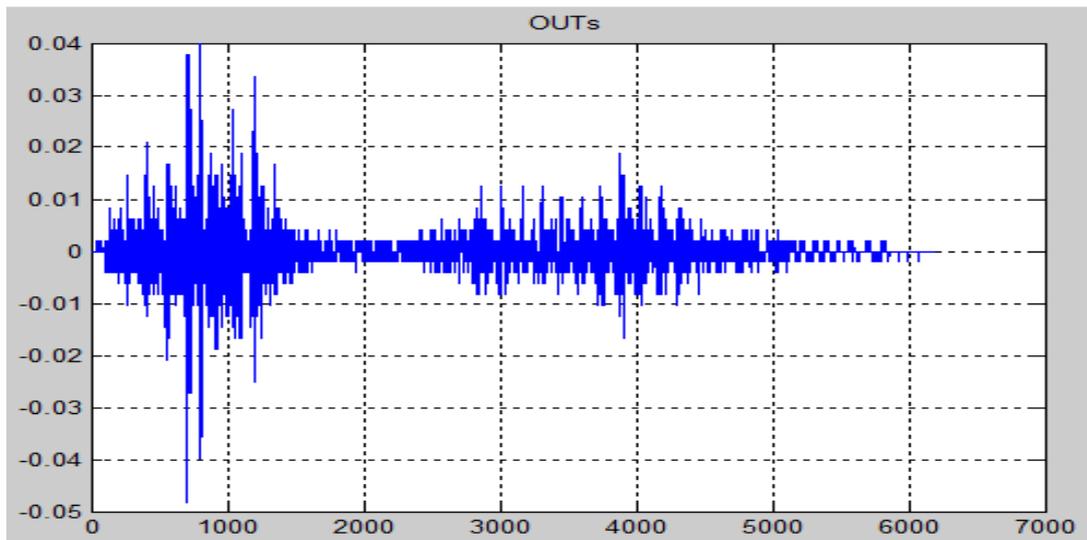


Рис. 4.5. Квантованный разностный сигнал

7. На приемной стороне канала связи по переданным квантованному разностному сигналу и коэффициентам линейного предсказания предсказывающего фильтра формируем квантованный входной сигнал.

```

%Задаем нулевое начальное состояние
%предсказывающего фильтра
ZZ1=zeros(n1-1,1);
%Задаем нулевое значение первого отсчета
%сигнала на входе фильтра
sstart=0;

%Формируем сегменты квантованного сигнала
quantvv=zeros(m0,m1);
for k33=1:m1
    ssegmnt=dataSegmnt(:,k33);
    ssgmntA=A(k33,:);
    vv=ssegmnt;
    for k22=1:length(vv)
        if k22==1
            [predictvv,ZZ]=filter(ssgmntA,1,sstart,ZZ1);
        else
            [predictvv,ZZ]=filter(ssgmntA,1,sstart2,ZZ);
        end
        quantvv(k22,k33)=predictvv+s(k22,k33);
        sstart2=quantvv(k22,k33);
    end
    sstart=quantvv(k22,k33);
    ZZ1=ZZ;
end
end

```

8. Определяем коэффициент усиления системы (во сколько раз дисперсия разностного сигнала меньше дисперсии квантованного входного сигнала). Сравниваем «осциллограмму» квантованного сигнала с «осциллограммами» входного и разностного сигналов.

```

OUTvv=quantvv(:);%объединяем сегменты квант. сигнала
G=var(OUTvv)/var(OUTs);%коэффициент усиления системы
figure;plot(OUTvv);grid on;title('OUTvv');
figure;plot(OUTs);grid on;title('OUTs')

```

9. Меняя порядок предсказания от 1 до 10, повторяем все пункты, приведенные выше. Строим график зависимости коэффициента усиления от порядка предсказания.

10. Используя результаты эксперимента, формулируем выводы относительно зависимости коэффициента усиления системы от порядка предсказания.

Содержание отчета о работе

- Номер варианта задания и исходные данные для экспериментов.
- Цель работы.
- Структурная схема разностного квантователя.
- Результаты экспериментов (скриншот окна переменных): целое и дробное значения числа разрядов, значение шага квантования.
- Осциллограммы сигналов. График зависимости коэффициента усиления от порядка предсказания.
- Выводы по работе.

Контрольные вопросы

1. В чем заключается преимущество разностного квантователя по сравнению с квантователем, имеющим постоянный шаг квантования?
2. Почему порядок предсказания ограничивают сверху?
3. Почему с увеличением порядка предсказания уменьшается дисперсия разностного сигнала?
4. Опишите работу схемы разностного квантователя.

Варианты заданий для выполнения работы

Номер варианта	Имя файла	Отношение сигнал-шум, дБ
1	m0_4.wav	40
2	fm17.wav	42
3	m16.wav	22
4	m7.wav	23
5	m0_4.wav	24
6	fm17.wav	25
7	m16.wav	26
8	m7.wav	27
9	m0_4.wav	28

Окончание

Номер вариант	Имя файла	Отношение сигнал-шум, дБ
10	fm17.wav	29
11	m16.wav	30
12	m7.wav	31
13	m0_4.wav	32
14	fm17.wav	33
15	m16.wav	34
16	m7.wav	35
17	m0_4.wav	36

Лабораторная работа № 5

ИССЛЕДОВАНИЕ LPC-КОДЕКА РЕЧЕВОГО СИГНАЛА

Цель работы: для заданного звукового файла определить параметры кодека, обеспечивающие минимум искажений синтезированного сигнала.

Краткие теоретические сведения

Задача речевого кодека – уменьшение скорости потока данных речи. Основа современных кодеков – представление процесса формирования (РС) в виде реакции фильтра, синтезирующего речевой сигнал, на сигнал источника возбуждения данного фильтра. Источник возбуждения соответствует потоку воздуха, который нагнетается из легких в голосовой тракт. Синтезирующий фильтр отражает работу органов артикуляции с указанным потоком воздуха.

Упрощенно все звуки речи можно разделить на две группы: вокализованные и невокализованные. Вокализованные звуки (например, гласные) формируются при вибрации голосовых связок, а невокализованные (например, свистящие и шипящие) – путем формирования турбулентного потока воздуха, который образуется за счет сужения голосового тракта в каком-либо месте.

Вибрация голосовых связок придает импульсный характер потоку воздуха, поступающего к органам артикуляции. С турбулентным потоком воздуха можно сопоставить результат работы генератора шума.

На основе приведенных рассуждений процесс формирования звуков речи можно смоделировать схемой, представленной на рис. 5.1.

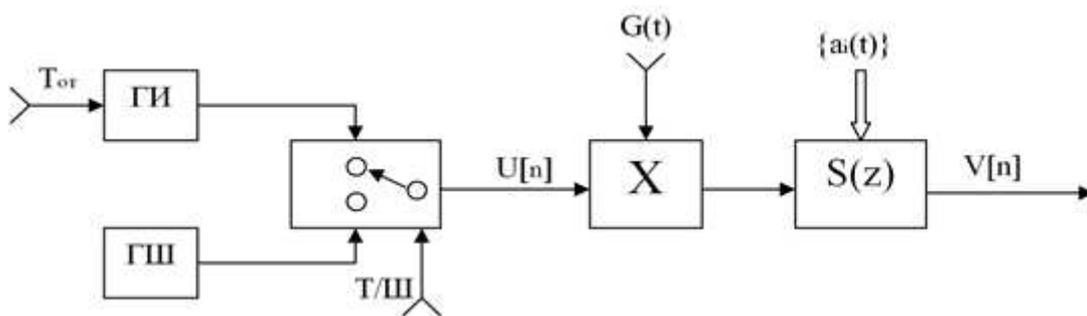


Рис. 5.1. Модель формирования речевого сигнала

На схеме обозначены: ГИ – генератор последовательности коротких импульсов, частота следования которых равна частоте вибраций голосовых связок – частоте основного тона РС; $T_{от}$ – период основного тона; ГШ – генератор шума; Т/Ш – признак «тон/шум» (принадлежность звука к группе вокализованных/невокализованных звуков); $G(t)$ – меняющийся во времени коэффициент усиления, отражающий громкость звуков; X – блок умножения; $S(z)$ – синтезирующий фильтр с данной системной функцией; $\{a(t)\}$ – набор меняющихся во времени коэффициентов, задающих системную функцию синтезирующего фильтра; $V[n]$ – речевой сигнал, меняющийся в дискретном времени n .

По каналу связи на приемную сторону передаются параметры модели – $T_{от}$, Т/Ш, $G(t)$, $\{a(t)\}$, – по которым синтезируется РС. Параметры модели определяет кодер на передающей стороне канала связи. Следует отметить, что коэффициенты $\{a(t)\}$ определяются на основе процедуры линейного предсказания текущего отсчета РС по нескольким предыдущим отсчетам. Число предыдущих отсчетов называют *порядком предсказания*.

Модель формирования РС выполняет функции декодера данных речи на приемной стороне канала связи. Поскольку параметры модели меняются медленно (примерно с частотой 100 Гц), то скорость потока данных невелика по сравнению с непосредственным аналого-цифровым преобразованием сигнала.

Рассмотренная структура декодера (модели РС) совместно с соответствующим кодером образуют простейший речевой LPC-кодек

(*Linear Prediction Coding* – кодирование на основе линейного предсказания). Достоинство кодека – реализуемая им малая скорость потока данных речи, а недостаток – сильные искажения синтезированного сигнала.

Структура кодека приведена на рис. 5.2. Здесь функции кодера исполняет «Формирователь сигнала остатка предсказания». Остальные блоки схемы относятся к декодеру.

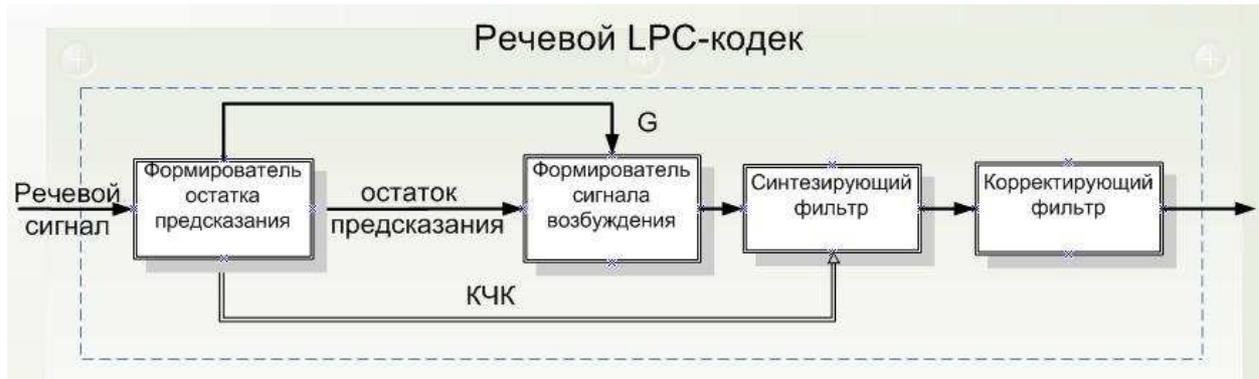


Рис. 5.2. Структура речевого LPC-кодека

Структура формирователя сигнала остатка предсказания представлена на рис. 5.3.



Рис. 5.3. Формирователь сигнала остатка предсказания

В формирователе сигнала остатка предсказания сначала РС пропускается через предсказывающий фильтр верхних частот с частотной характеристикой $h(z) = 1 - 0,95z^{-1}$, Фильтр необходим для повышения точности определения коэффициентов частной корреляции (КЧК).

Затем сигнал разбивается на ряд перекрывающихся во времени сегментов (блок «Буфер»). Перекрывание сегментов во времени обеспечивает более точное отслеживание изменения параметров сигнала во времени. Далее каждый сегмент взвешивается окном (оконной функцией) Хэмминга. Оконная функция уменьшает искажения спектра сигнала, которые возникают при его сегментации.

Определение КЧК осуществляют путем вычисления автокорреляционной функции сегмента кодируемого сигнала с последующим использованием алгоритма Левинсона – Дарбина.

Полученные коэффициенты поступают далее на управляющие входы нерекурсивного лестничного фильтра, который выполняет функцию инверсного фильтра. Его частотная характеристика обратна частотной характеристике синтезирующего фильтра. На сигнальный вход фильтра поступает сигнал с выхода предсказывающего фильтра верхних частот. На выходе инверсного фильтра формируется сигнал остатка предсказания.

Если остаток предсказания подать непосредственно на вход синтезирующего фильтра на приемной стороне канала связи, то на выходе последовательности «синтезирующий фильтр, корректирующий фильтр» формируется неискаженный речевой сигнал. Однако точная передача остатка предсказания по линии связи требует большого количества битов. Поэтому на практике остаток предсказания заменяют похожим на него сигналом возбуждения синтезирующего фильтра. Этот сигнал требует меньшего числа битов при передаче по линии связи, но обуславливает появление искажений в речевом сигнале.

На рис. 5.4 изображена структура формирователя сигнала возбуждения синтезирующего фильтра совместно с блоками из состава кодера, которые определяют параметры сигнала возбуждения.



Рис. 5.4. Формирователь сигнала возбуждения синтезирующего фильтра

Для формирования сигнала возбуждения на приемной стороне линии связи используют три параметра: результат классификации звуков на вокализованные и невокализованные (результат работы классификатора), период основного тона (результат работы измерителя основного тона (ИОТ)) и коэффициент усиления.

Если сегмент вокализованный, то формируется последовательность коротких импульсов с помощью генератора импульсов (ГИ). Период следования импульсов равен периоду основного тона. Если сегмент не вокализованный, то генератор шума (ГШ) формирует случайный сигнал. Сигнал на выходе ключевого элемента умножается на коэффициент усиления G . Чем больше сформированный сигнал возбуждения похож на сигнал остатка предсказания, тем меньше искажения синтезированного сигнала.

Работа классификатора и ИОТ осуществляется на основе анализа функции среднего значения разности (*Average Magnitude Difference Function – AMDF*).

Для строго периодической функции $X[n]$ с периодом P последовательность разностей

$$d(n) = X[n] - X[n + k]$$

равна нулю при $k = 0 \pm P, \pm 2P, \dots$. Для сегментов вокализованного речевого сигнала естественно ожидать, что последовательность $d(n)$ будет близка к нулю при величине k , кратной периоду основного тона. Кратковременная функция среднего значения разности (*AMDF*) определяется как

$$y(k) = \frac{1}{R} \sum_{n=0}^{N-1} |X[n] - X[n+k]|,$$

где R – нормирующий делитель; $X[n]$ – значение входного сигнала ИОТ в момент времени nT_d ; T_d – период дискретизации; N – число выборок в сегменте сигнала.

В общем случае $X[n]$ – сумма периодического и случайного компонентов, поэтому данная функция является случайной. Очевидно, что если $X[n]$ близка к периодической функции на интервале, выделенном с помощью временного окна, то величина $y(k)$ имеет глубокие провалы при $k = P, 2P, \dots$. На рис. 5.5 приведена упрощенная форма математического ожидания для функции *AMDF*. Пунктиром показан возможный разброс значений функции.

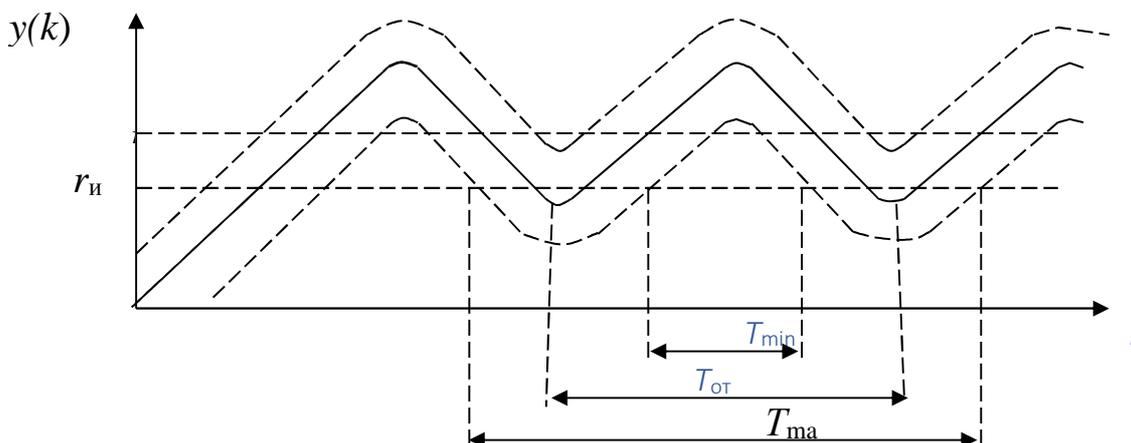


Рис. 5.5. Форма функции *AMDF*

Период $T_{от}$ основного тона определяют как расстояние между двумя минимумами функции. Измерительный порог $r_и$ служит для фиксации минимумов. Минимальные значения функции определяют лишь для тех значений k , которые обеспечивают выполнение условия $y(k) \leq r_и$. Из рис. 5.5 следует, что чем больше разброс значений $y(k)$, меньших $r_и$, тем больше разброс значений измеренного периода ($T_{min} \dots T_{max}$), а следовательно, и больше погрешность измерения периода основного тона РС.

Функцию *AMDF* используют для классификации сегментов РС следующим образом: если отношение минимума функции к максимуму будет больше заданного «классификационного» порога r_k , то

принимают решение о невокализованности сегмента; если это же отношение меньше, то сегмент считают вокализованным.

Если пороговое значение задано неверно, то возникают ошибки классификатора: по линии связи передается импульсная последовательность в то время, когда должна передаваться шумовая последовательность, и наоборот. Ошибки в определении периода основного тона, а также ошибки классификации звуков приводят к искажениям речевого сигнала на выходе.

На рис. 5.6 представлена экспериментально полученная функция $AMDF$ для вокализованного сегмента. Прямая линия соответствует пороговому значению r_k . Для вокализованных звуков она должна проходить выше минимального значения функции $AMDF$.

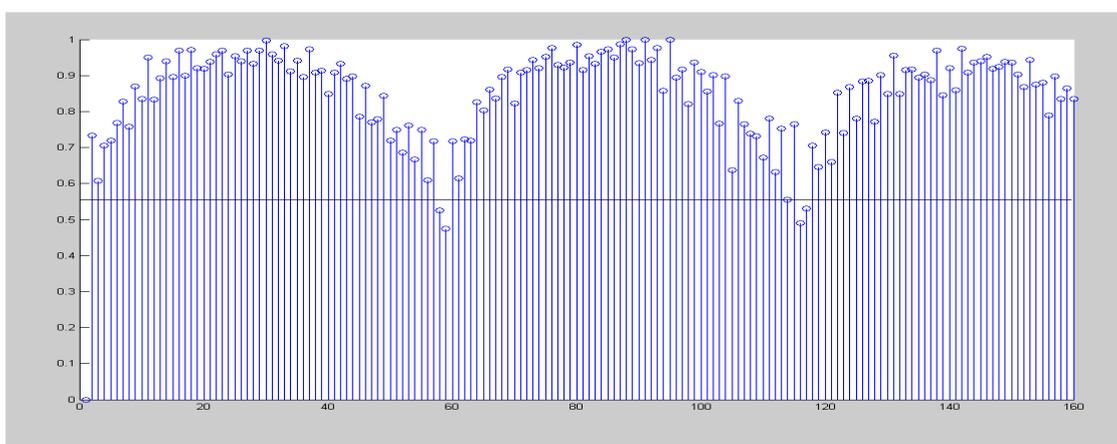


Рис. 5.6. Экспериментально полученная функция $AMDF$ для вокализованного сегмента

Основа формирователя сигнала возбуждения – измеритель основного тона, который классифицирует сегменты сигнала на вокализованные и невокализованные и определяет период основного тона для вокализованных сегментов.

Настройку измерителя основного тона осуществляют следующим образом. По «осциллограммам» РС и сигнала остатка предсказания исследователь «вручную» классифицирует сегменты на вокализованные и невокализованные. Результат классификации соответствует работе идеального классификатора.

Далее устанавливают оптимальный порог классификации путем сравнения результата классификации с результатом идеальной класси-

фикации. Меняя порог, добиваются минимального числа ошибок. После установки оптимального порога проводят измерение периода основного тона на вокализованных сегментах.

На основе результатов классификации и измерения периода основного тона формируется сигнал, состоящий из фрагментов белого шума и импульсных последовательностей. Сигнал перемножается с коэффициентом усиления и поступает на вход синтезирующего фильтра. Синтезированный сигнал прослушивают и в зависимости от результатов при необходимости корректируют значения порога.

Порядок выполнения работы (пример выполнения задания) (реализуется строками сценария *Matlab*)

1. Выделяем из заданного звукового файла *m0_4.wav* (рис. 5.7) фрагмент, соответствующий произнесению названий цифр: «три» и «четыре» (рис. 5.8).

```
V0,Fs,bJ=wavread('m0_4.wav');  
plot(V0); grid on; title('Заданный звуковой сигнал')
```

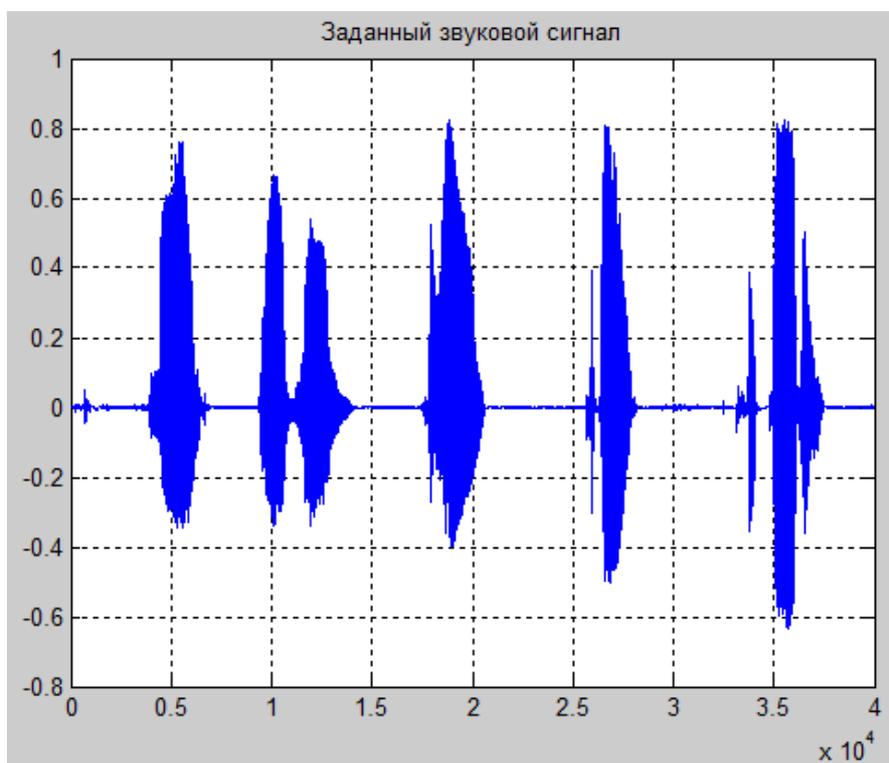


Рис. 5.7. Заданный звуковой сигнал

```
V=V0(25e3:38e3); %Выделяем заданный фрагмент сигнала
figure;plot(V);grid on;title('Заданный фрагмент сигнала')
```

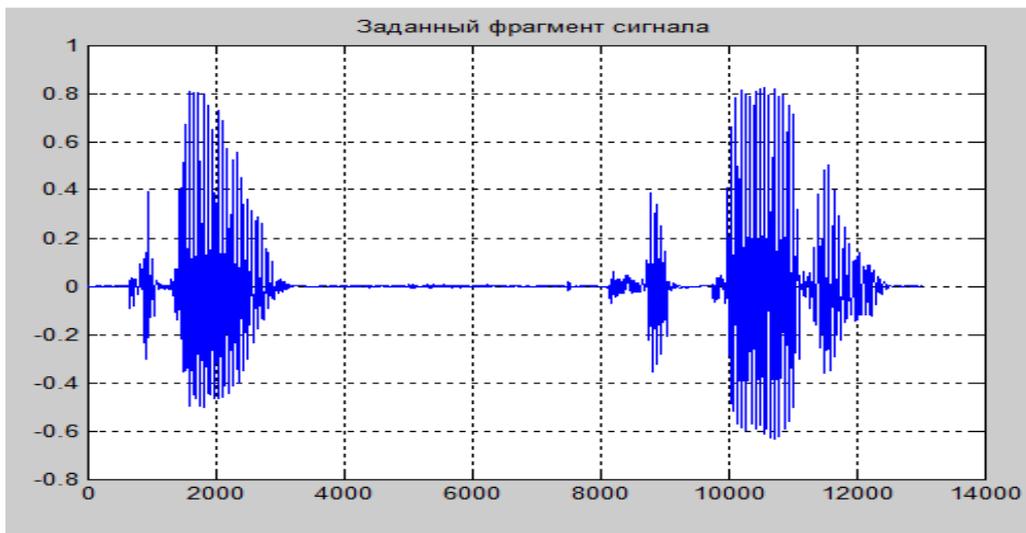


Рис. 5.8. Заданный фрагмент сигнала

2. Вводим исходные данные для исследования.

```
ds=200;%размер сегмента (кол-во отсчетов)
do=100;%размер перекрытия соседних сегментов
pre_order=10; %порядок предсказания
```

3. Определяем КЧК, коэффициент усиления G и число сегментов в выделенном фрагменте РС. Формируем сигнал $Predct_V$ остатка предсказания.

```
%вычисление  $K$  коэффициентов частной корреляции фильтра,
%энергию  $E$  остатка предсказания
[K,E,n]=analyse2(V,ds,do,pre_order);
G=E.^0.5; % коэффициент усиления
%Формирование сигнала остатка предсказания
Predct_V=filt_error2(V,ds,do,K,n);
```

4. Определяем зависимость энергии остатка предсказания от порядка предсказания (рис. 5.9) и выбираем минимальное значение порядка предсказания, при превышении которого энергия остатка предсказания меняется незначительно.

```

pre_order1max=15;%Максимальное значение порядка предсказания
EE=zeros(pre_order1max,1);
k=0;
for pre_order1=1:pre_order1max
    k=k+1;
    [K1, E1, n1]=analyse2(V,ds,do,pre_order1);
    EE(k)=mean(E1);
end
figure;stem(EE);grid on;title('Зависимость энергии остатка предсказа-
ния от порядка предсказания');

```



Рис. 5.9. Зависимость энергии остатка предсказания от порядка предсказания

Устанавливаем порядок предсказания $pre_order=10$.

5. Синтезируем сигнал по остатку предсказания и анализируем результаты синтеза (рис. 5.10).

```

Predct_V_sgmnt=buffer(Predct_V,ds-do);
[filt_synth]=synthesis2(K,Predct_V_sgmnt,n);
%Пропускаем синтезированный сигнал через корректирующий ФНЧ
Filt_xf=filter(1,[1 -.95],filt_synth);
Filt_xf_corr=Filt_xf(1:length(V));
figure;

```

```

subplot(4,1,1);plot(V0);grid on;title('Заданный сигнал');
subplot(4,1,2);plot(V);grid on;title('Заданный фрагмент сигнала');
subplot(4,1,3);plot(Filt_xf_corr);grid on;title('Сигнал, синтезированный
по остатку предсказания');
subplot(4,1,4);plot(Predct_V);grid on;title('Сигнал остатка предсказания');

```

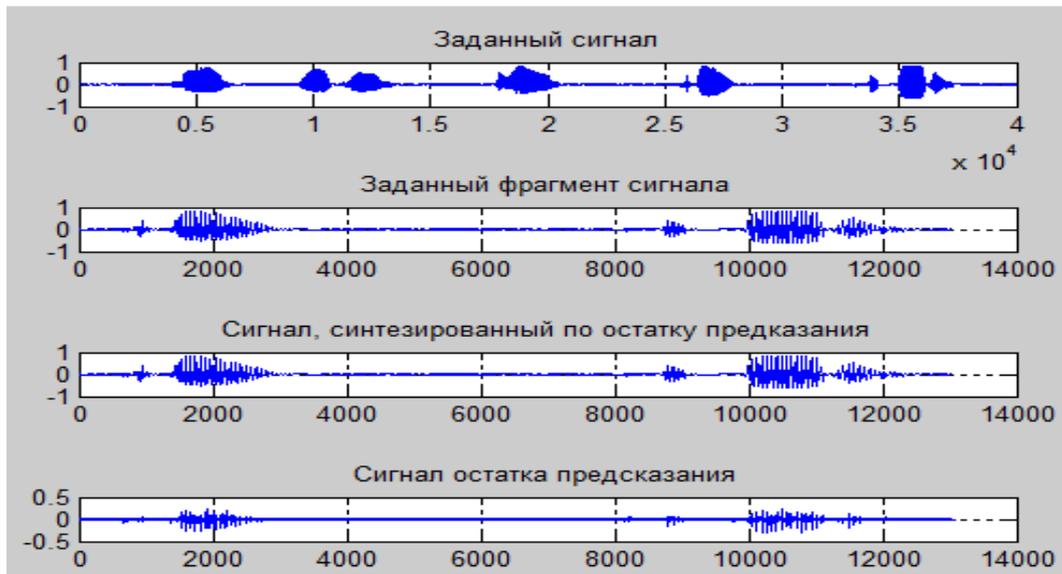


Рис. 5.10. Формы соответствия сигналов

6. Анализируя «осциллограммы» РС и остатка предсказания, формируем результат (переменная *class2*) идеальной классификации сегментов сигнала на вокализованные и невокализованные.

```

class2=ones(1,n); %матрица классификатора, заполненная единицами
%Ставим нули для сегментов пауз и невокализованных звуков
class2(1:8)=zeros(1,8);
class2(11:13)=zeros(1,3);
class2(30:87)=zeros(1,58);
class2(91:99)=zeros(1,9);
class2(124:n)=zeros(1,8);

```

7. Устанавливаем предварительное значение порога классификации и наблюдаем результат реализованной автоматической классификации сегментов РС. По результатам наблюдения корректируем значение порога. Необходимо обеспечить значение порога, которое превышает отношение y_{\min}/y_{\max} минимального значения функции *AMDF* к его

максимальному значению для вокализованных сегментов. После коррекции фиксируем результат автоматической классификации и определяем наиболее вероятное значение периода *pitch* основного тона (рис. 5.11).

```

porog=0.55; %установка порогового значения
porogGraph=zeros(n,1)+porog;%массив пороговых значений
%для графического отображения
%Определение функции AMDF и результата
%автоматической классификации
[AMDF,class1,class,pitch]=AMDF3(Predct_V,ds,do,porog);
figure; stem(AMDF(:,107)); grid on; title('AMDF');%просмотр
%нормированной по максимуму функции AMDF для выбранного сегмента
figure;
subplot(4,1,1);stem(class2);grid on;title('Результат идеальной
классификации')
subplot(4,1,2);stem(class);grid on;title('Результат реализованной
классификации')
subplot(4,1,3);stem(class1);grid on;title('Отношение umin/umax');
hold on;plot(porogGraph,'r');
hold off;
subplot(4,1,4);stem(pitch);grid on;title('Период основного тона')

```

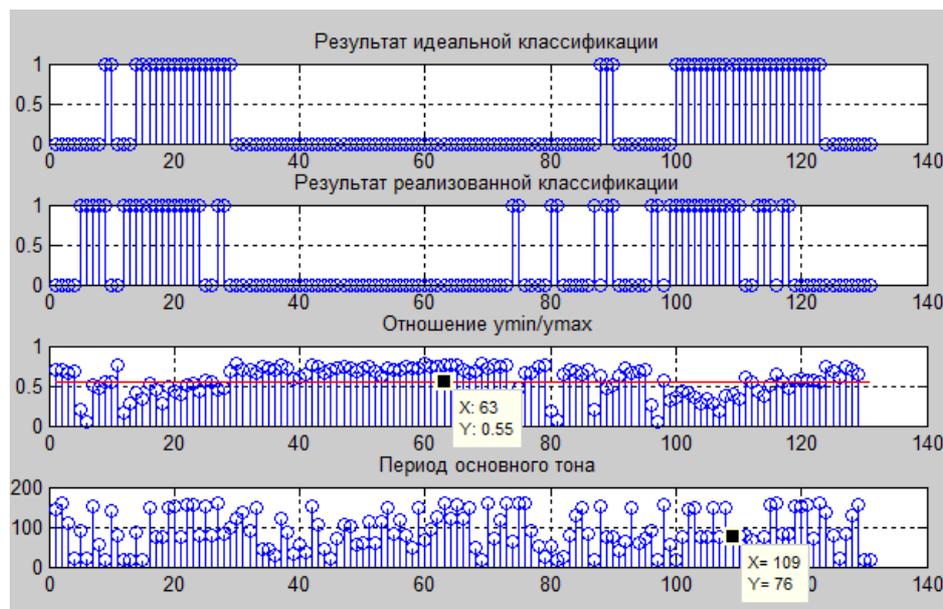


Рис. 5.11. Результаты классификации и определения частоты основного тона

Значение порога равно 0.55. Устанавливаем для дальнейшей работы значение периода основного тона равным 76 отсчетам сигнала.

Наблюдаем за формой функции *AMDF* для 107 сегмента сигнала (рис. 5.12).

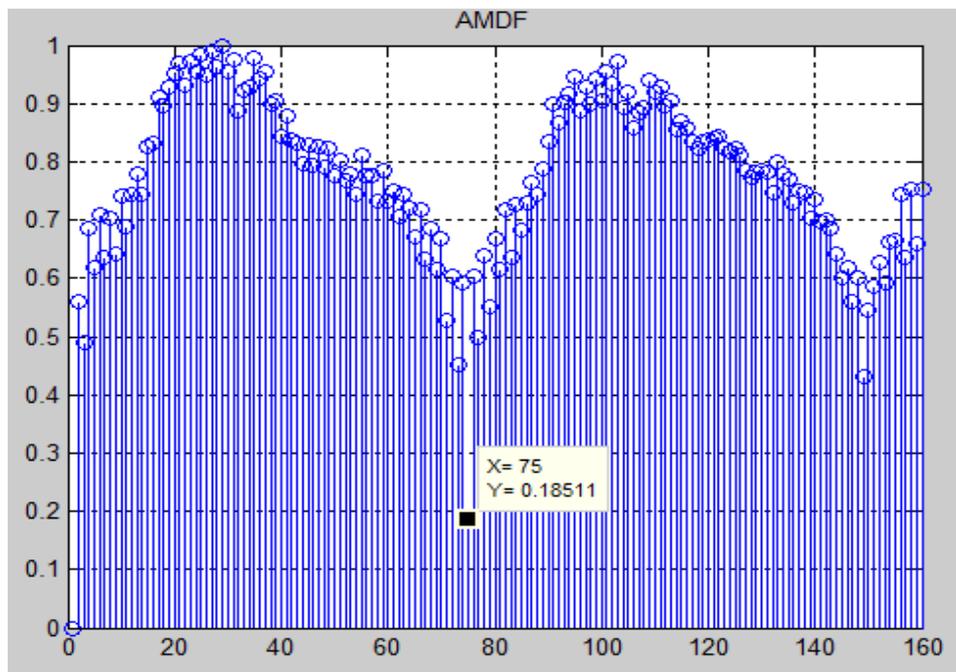


Рис. 5.12. Форма функции *AMDF*

Видно, что для данного вокализованного сегмента период основного тона равен 75 периодам частоты дискретизации сигнала.

8. С учетом выбранного значения периода основного тона ($T = 76$) формируем сигнал возбуждения синтезирующего фильтра и синтезируем речевой сигнал. По осциллограммам сигналов оцениваем уровень искажений синтезированного сигнала для случаев идеальной и реализованной классификаций (рис. 5.13). По результатам прослушивания оцениваем качество сигнала по шкале: *хорошо, удовлетворительно, плохо*.

```
T=76;%Выбранное значение периода основного тона
class=class2;%использование идеальной классификации
при формировании сигнала возбуждения
[Vozbud]=sigVozbud(class,T,G,n,do,Fs);%формирование сигнала
возбуждения
%Синтез сигнала
sgmntVozbud=buffer(Vozbud,do,0);
```

```

[fil_synth]=synthesis2(K,sgmntVozbud,n);
%Пропускаем синтезированный сигнал через корректирующий ФНЧ
Filtr_xf=filter(1,[1 -.95],fil_synth);
Filtr_xf_corr=Filtr_xf(1:length(V));
figure;
subplot(4,1,1);plot(V);grid on;title('Заданный фрагмент сигнала');
subplot(4,1,2);plot(Filtr_xf_corr);grid on;title('Синтезированный
фрагмент сигнала');
subplot(4,1,3);plot(Vozbud);grid on;title('Реализованный сигнал
возбуждения');
subplot(4,1,4);plot(Predct_V);grid on;title('Сигнал остатка предсказания');
soundsc(Filtr_xf_corr)

```

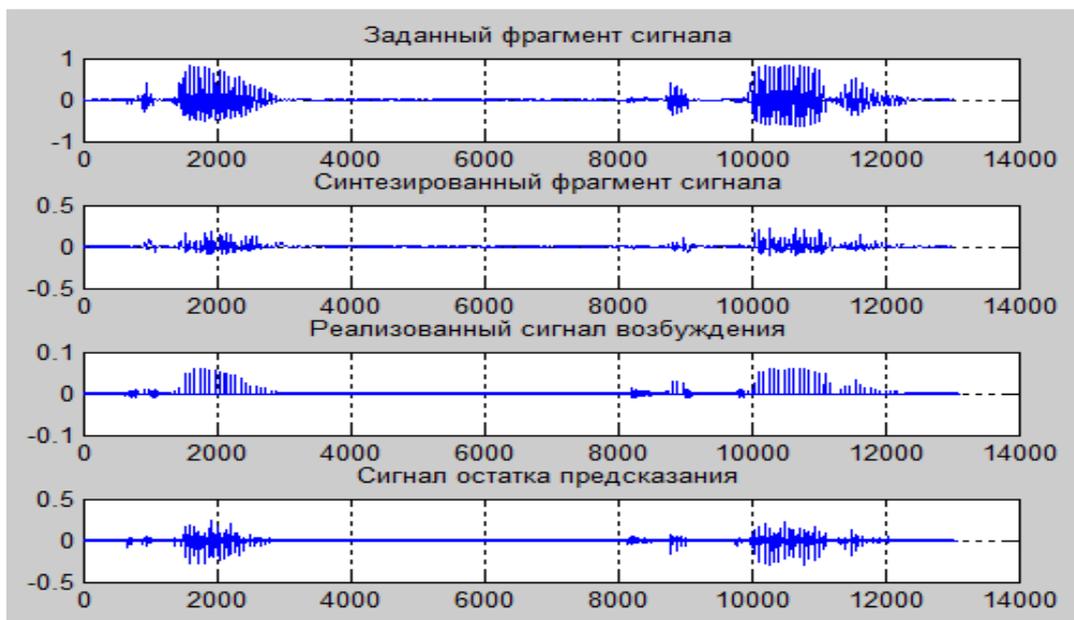


Рис. 5.13. результаты синтеза речевого сигнала и сигнала возбуждения

9. Формулируем выводы по работе

Содержание отчета о работе

- Номер варианта задания и цель работы.
- Структурные схемы кодека и его составляющих.
- «Осциллограммы» сигналов.
- Выводы по работе.
- Тексты используемых сценариев и функций.

Контрольные вопросы

1. Какова основная причина искажений синтезированного сигнала?
2. Каким образом осуществляется классификация сегментов?
3. Каким образом определяют период основного тона речевого сигнала?
4. Каковы достоинства и недостатки *LPC*-кодека?

Варианты задания для выполнения работы

Номер варианта	Имя звукового файла (название)
1	m0_4.wav(1,2)
2	fm17.wav(0,1)
3	m16.wav(3,4)
4	m7.wav(0,1)
5	m0_4.wav(3,4)
6	fm17.wav(2,3)
7	m16.wav(5,6)
8	m7.wav(7,8)
9	m0_4.wav(0,1)
10	fm17.wav(4,5)
11	m16.wav(8,9)
12	m7.wav(ошиб)
13	m0_4.wav(2,3)
14	fm17.wav(3,4)
15	m16.wav(6,7)
16	m7.wav(да,нет)
17	m0_4.wav(3,4)
18	m7.wav(0,1)
19	m0_4.wav(3,4)
20	fm17.wav(2,3)
21	m16.wav(5,6)
22	m7.wav(7,8)
23	m0_4.wav(0,1)
24	fm17.wav(4,5)
25	m16.wav(8,9)
26	m7.wav(ошиб)
27	m0_4.wav(2,3)

ЗАКЛЮЧЕНИЕ

В лабораторном практикуме представлены основные подходы к сжатию потока данных речи. Описаны следующие виды квантования РС: квантование с постоянным шагом, квантование с использованием компандирования сигнала, адаптивное квантование, разностное квантование.

Рассмотрено построение речевого кодека на основе метода линейного предсказания, в котором возбуждение синтезирующего фильтра происходит импульсами основного тона РС.

Для оценки качества РС используются объективная оценка – отношение мощности сигнала к мощности шума квантования, а также субъективная оценка по результатам прослушивания РС.

РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сергиенко, А. Б. Цифровая обработка сигналов / А. Б. Сергиенко. – СПб. : Питер, 2002. – 608 с. – ISBN 5-318-00666-3.

2. Исследование алгоритмов обработки сигналов в системе MATLAB : метод. указания к лаб. работам / Владим. гос. ун-т ; сост. Е. К. Левин. – Владимир : Изд-во Владим. гос. ун-та, 2011. – 78 с.

Электронный ресурс ВлГУ

Обработка сигналов звука и изображений в системах связи : конспект лекций по дисциплине «Обработка сигналов» для студентов ВлГУ, обучающихся по направлению 11.03.01 «Радиотехника» [Электронный ресурс] / Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых ; авт.-сост. Е. К. Левин. – URL: dspace.www1.vlsu.ru/handle/123456789/7065 (дата обращения: 01.10.2023).

ПРИЛОЖЕНИЕ

Листинги сценариев системы Matlab, используемых при выполнении лабораторных работ

Лабораторная работа № 1

ИССЛЕДОВАНИЕ КВАНТОВАТЕЛЯ С ПОСТОЯННЫМ ШАГОМ КВАНТОВАНИЯ

Листинг сценария для определения порогов ограничения сигнала (*OpredPorog3.m*)

```
% Определение порогов ограничения сигнала
clear;close all;
%[data,fs]=wavread('m5_9.wav');% чтение звукового файла
[data,fs]=audioread('m5_9.wav');% чтение звукового файла
% для поздних версий Matlab
Procnt=0.35e-2;% относительное число отсчетов сигнала,
% которые попадают под ограничение
L=length(data);
L_Procnt=L*Procnt;% число отсчетов сигнала, которые
% попадают под ограничение
hist(data,15);grid on;title('Гистограмма входного сигнала');
```

Листинг сценария для расчета квантователя и проверки правильности расчета (*RaschetProverka4.m*)

```
% Расчет квантователя и проверка расчета
clear; close all;
q_dB=20;% заданное отношение сигнал-шум
porog_max=0.644;
porog_min=-0.5;
[data,fs]=wavread('m5_9.wav');% чтение звукового файла
% [data,fs]=audioread('m5_9.wav');% чтение звукового файла
% для поздних версий Matlab
```

```

%Проводим расчет квантователя без учета
%ограничения сигнала
[B0,B,delta,data_q]=RaschetQvant4(data,q_dB);
dataDAC=data_q*delta;% сигнал на выходе квантователя

% Определяем с/ш после квантования
shum=dataDAC-data;% шум квантования
P_shum=var(shum);% дисперсия шума квантования
P_s=var(data);% дисперсия сигнала
q_shum=P_s/P_shum;% фактическое значение с/ш
q_dB_m=10*log10(q_shum);% фактическое значение с/ш (дБ)

%Проводим расчет квантователя
%с учетом ограничения сигнала
data_ogr=Ogrnich(data,porog_max,porog_min);% Ограничение сигнала
[B0_ogr,B_ogr,delta_ogr,data_q_ogr]=RaschetQvant4(data_ogr,q_dB);
dataDACogr=data_q_ogr*delta_ogr;% сигнал после ограничения и
квантования

% Определяем с/ш после ограничения и квантования
shum_ogr=dataDACogr-data_ogr;% шум квантования ограничен.
сигнала
P_shum_ogr=var(shum_ogr);% дисперсия шума квантования
P_s=var(data);% дисперсия сигнала
q_shum_ogr=P_s/P_shum_ogr;% фактическое значение с/ш
q_dB_ogr=10*log10(q_shum_ogr);% фактическое значение с/ш (дБ)

% оценка качества сигнала после ограничения
%и квантования
t=0:1/fs:(length(data)-1)/fs;
figure;plot(t,data);grid on;title('Сигнал до ограничения');
figure;plot(t,data_ogr);grid on;title('Сигнал после ограничения');
figure;
plot(t,dataDACogr);grid on;title('Сигнал после ограничения и квантования');

% прослушивание входного сигнала
soundsc( data,fs);

```

```

% прослушивание входного сигнала после ограничения
soundsc( data_ogr,fs);
% прослушивание входного сигнала
% на приемной стороне после ограничения и квантования
soundsc( dataDACogr,fs);

```

Функции, используемые при исследовании квантователя

```

function [ data_ogr ] = Ogrnich( data,porog_max,porog_min )
% Функция ограничивает сигнал снизу и сверху
% data - входной сигнал
% porog_max,porog_min - верхний и нижний пороги ограничения
% data_ogr - выходной сигнал

L=length(data);
data_ogr=zeros(L,1);
for k=1:L
    if data(k)>=porog_max
        data_ogr(k)=porog_max;
    elseif data(k)<=porog_min;
        data_ogr(k)=porog_min;
    else
        data_ogr(k)=data(k);
    end
end
end

function [B0,B,delta,data_q]=RaschetQvant4(data,q_dB)
% Расчет квантователя
% data - квантуемый сигнал
% q_dB - заданное отношение с/ш в дБ

%B0 - дробное число разрядов
%B - число разрядов квантователя
%delta - уточненный шаг квантования
%data_q - поток данных после квантования(выход АЦП)

q=10^(q_dB/10);% заданное отношение с/ш в "размах" по мощности

```

```

razmakh=max(data)-min(data);% размах сигнала
P=var(data);% средняя мощность сигнала
delta0=sqrt(12*P/q);% первичный шаг квантования
B0=log2(razmakh/delta0);% дробное число разрядов
B=ceil(B0);% целое число разрядов
N=2^B;% число уровней квантования
delta=razmakh/N;% уточненный шаг квантования
data_q=round(data/delta);% номера уровней квантования

```

Лабораторная работа № 2

ИССЛЕДОВАНИЕ КВАНТОВАТЕЛЯ, ИСПОЛЬЗУЮЩЕГО КОМПАНДИРОВАНИЕ СИГНАЛОВ

Листинг сценария для проверки соответствия друг другу законов компрессии и экспандирования (*TstComprExpnd4*)

```

% Проверка правильности реализаций
% функций компрессии и экспандирования (mu-закон)
clear; close all;
path(path,'bibl');
mu=200;% параметр компрессии
data=-100:100;% тестовый сигнал

dataCmrps = compress2( data,mu );% компрессия
dataExpnd = expnd2( data,mu );% экспандирование
subplot(2,1,1);plot(data,dataCmrps);title('mu-закон компрессии');grid on;
subplot(2,1,2);plot(data,dataExpnd);title('mu-закон экспандирования');
grid on;

% Проверка соответствия закона экспандирования закону компрессии
%(экспандирование сжатого тестового сигнала)
dataExpnd2 = expnd2( dataCmrps,mu );
figure;stem(data,dataExpnd2);title('Результат экспандирования сжатого
тестового сигнала (mu-закон)');grid on

```

**Листинг сценария для исследования квантователя
с компандированием сигналов
(comprand6.m)**

```
clear; close all;
path(path,'bibl');
q_dB=20;% заданное отношение сигнал-шум

[data,fs]=wavread('m5_9.wav');% чтение звукового файла
% [data,fs]=audioread('m5_9.wav');% чтение звукового файла
% для поздних версий Matlab

% Определяем с/ш после квантования
[B0,B,delta,data_q]=RaschetQvant4(data,q_dB);
dataDAC=data_q*delta;
shum=dataDAC-data;% шум квантования
P_shum=var(shum);% дисперсия шума квантования
P_s=var(data);% дисперсия сигнала
q_shum=P_s/P_shum;% фактическое значение с/ш
q_dB_m=10*log10(q_shum);% фактическое значение с/ш (дБ)

% Квантование с компандированием
mu=150;% параметр компрессии
dataCmrps = compress2( data,mu );
[BCmrps0,BCmrps,deltaCmrps,data_qCmrps]=RaschetQvant4(dataCmrps,
q_dB);% квантование
data_outCmrps=data_qCmrps*deltaCmrps;% сигнал на выходе квант-ля
dataExpnd = expnd2( data_outCmrps,mu );% сигнал после экспандирования

% Оценка качества сигнала после
% квантования с компандированием
shumExpnd=dataExpnd-data;% шум квантования
P_shumExpnd=var(shumExpnd);% дисперсия шума квантования
P_s=var(data);% дисперсия сигнала
q_shumExpnd=P_s/P_shumExpnd;% фактическое значение с/ш (в "разах")
q_dB_expExpnd=10*log10(q_shumExpnd);% фактическое
%(экспериментальное) среднее значение с/ш (дБ)
```

```

%сегментное отношение с/ш (дБ)
n_sgmnt=200;
[q_dB_sgmnt] = SNRsgmnt4(n_sgmnt,data,dataDAC);
[q_dB_Expnd_sgmnt] = SNRsgmnt4(n_sgmnt,data,dataExpnd);

figure;
subplot(2,1,1); stem(q_dB_sgmnt);title('С/Ш при равномерном квантовании
(по сегментам)');
grid on;
subplot(2,1,2); stem(q_dB_Expnd_sgmnt);title('С/Ш при компандирова-
нии (по сегментам)');
grid on;

figure;
subplot(2,1,1); stem(dataDAC);title('Сигнал при равномерном квантовании');
grid on;
subplot(2,1,2); stem(dataExpnd);title('Сигнал при квантовании
с компандированием');
grid on;
%soundsc(dataExpnd);
%soundsc(dataDAC);

```

Функции, используемые при исследовании квантователя

```

function dataCmrps = compress2( data,mu )
%компрессия входного сигнала
%data - входной сигнал
%mu - параметр компрессии
%dataCmrps - выходной сигнал
X_max=max(abs(data));
dataCmrps=X_max*log(1+mu*abs(data)/ X_max)/log(1+mu).*sign(data);
end

```

```

function dataExpnd = expnd2( data,mu )
%компрессия входного сигнала
%data - входной сигнал

```

```

%mu - параметр компрессии
%dataExpnd - выходной сигнал
X_max=max(abs(data));
dataExpnd=X_max/mu*(exp(abs(data)/X_max*log(1+mu))-1).*sign(data);
end

```

```

function dataCmrps = compressA2(data,A)
%компрессия входного сигнала
%data - входной сигнал
%A - параметр компрессии
%dataCmrps - выходной сигнал
X_max=max(abs(data));
data1=data/X_max;
C1=abs(data1)<(1/A);
C2=1-C1;
dataCmrps1=X_max*abs(data1)*A/(1+log(A)).*sign(data);
dataCmrps2=X_max*(1+log(abs(A*data1+1e-
10)))/(1+log(A)).*sign(data);
dataCmrps=C1.*dataCmrps1+C2.*dataCmrps2;

```

```

function dataExpnd = expandA2(data,A)
%Экспандирование входного сигнала
%data - входной сигнал
%A - параметр компрессии
%dataExpnd - выходной сигнал
%
X_max=max(abs(data));
data1=data/X_max;
C1=abs(data1)<(1/(1+log(A)));%Матрица C содержит единицы, когда
%неравенство справедливо и нули - в противном случае
dataExpnd1=X_max*(abs(data1) *(1+log(A))/A).*sign(data);
C2=1-C1;
dataExpnd2=(X_max*(1/A)*exp(abs(data1)*(1+log(A))-1)).*sign(data);
dataExpnd=C1.*dataExpnd1+C2.*dataExpnd2;

```

```

function [q_dB_sgmnt] = SNRsgmnt4(n_sgmnt,data,data_out)
%определяется с/ш для каждого сегмента сигнала

```

```

%n_sgmnt - размер сегмента сигнала
%data - данные входного сигнала квантователя
%data_out - данные выходного сигнала квантователя
%q_dB_sgmnt - отношение с/ш для каждого сегмента сигнала
data_sgmnt=buffer(data,n_sgmnt);% сегменты квантуемого сигнала
data_out_sgmnt=buffer(data_out,n_sgmnt);% сегменты сигнала на вых.
квант-ля
shum_sgmnt=data_out_sgmnt-data_sgmnt;% шум квантования
Ps_sgmnt=var(data_sgmnt);% мощность квантуемого сигнала по сегментам
P_shum_sgmnt=var(shum_sgmnt);% дисперсия шума квантования
q_sgmnt=Ps_sgmnt./P_shum_sgmnt;% фактическое значение с/ш
q_dB_sgmnt=10*log10(q_sgmnt);% фактическое значение с/ш (в дБ)

```

Лабораторная работа № 3

ИССЛЕДОВАНИЕ АДАПТИВНОГО КВАНТОВАТЕЛЯ

Листинг сценария (*AdaptQuant*)

```

close all; clear;
% Подготовка исходных данных для расчета
[data,fs]=wavread('m5_9.wav');
% [data,fs]=audioread('m5_9.wav');% чтение звукового файла
% для поздних версий Matlab
q_dB=10;% требуемое отношение С/Ш (дБ)
n_sgmnt=200;% размер сегмента сигнала
% на котором определяется шаг квантования
n_sgmnt2=200;% размер сегмента сигнала
% для определения с/ш
% сегментированные данные
data_sgmnt=buffer(data,n_sgmnt);

% При заданном отношении с/ш определяем
% число разрядов квантователя с постоянным шагом
q=10^(q_dB/10);% заданное отношение с/ш в "размах" по мощности

```

```

razmakh00=max(data)-min(data);%размах сигнала
P00=var(data);%средняя мощность сигнала
delta00=sqrt(12*P00/q);%первичный шаг квантования
B00=log2(razmakh00/delta00);%дробное число разрядов
B00=ceil(B00);%целое число разрядов
N00=2^B00;%число уровней квантования
delta000=razmakh00/N00;%уточненный шаг квантования
data_q00=round(data/delta000);%номера уровней квантования
dataDAC00=data_q00*delta000;

%Определяем отношение с/ш для
%квантователя с постоянным шагом
shum00=dataDAC00-data;%шум квантования
P_shum00=var(shum00);%дисперсия шума квантования
P_s00=var(data);%дисперсия сигнала
%q_shum=P_s/P_shum;%фактическое значение с/ш
q_dB_m00=10*log10(P_s00/P_shum00);%фактическое значение с/ш
(дБ)

%Определяем отношение с/ш для каждого сегмента
%для квантователя с постоянным шагом
data_sgmnt2=buffer(data,n_sgmnt2);
shumSgmnt00=buffer(shum00,n_sgmnt2);
%Определяем мощность сигнала для каждого сегмента
Ps_sgmnt2=var(data_sgmnt2);
%Определяем мощность шума для каждого сегмента сигнала
Pshum_sgmnt200=var(shumSgmnt00);
qdB_sgmnt200=10*log10(Ps_sgmnt2./Pshum_sgmnt200);

%При заданном отношении с/ш определяем
%число разрядов для адапт. квантователя
q=10^(q_dB/10);%заданное отношение с/ш в "размах" по мощности
razmakh=max(data_sgmnt)-min(data_sgmnt);%размах сигнала
%по сегментам
P=var(data_sgmnt);%мощность сигнала
%по сегментам
delta0=sqrt(12*P/q);%первичный шаг квантования

```

```

% по сегментам
B0=log2(razmakh./delta0);% дробное число разрядов
% по сегментам
B0m=mean(B0);% Среднее по всем сегментам число разрядов
B=ceil(B0m);% целое число разрядов
N=2.^B;% число уровней квантования
delta=razmakh./N;% уточненный шаг квантования
% по сегментам
delta_rep=repmat(delta,n_sgmnt,1);% размер матрицы
% delta_rep равен размеру матрицы data_sgmnt
data_q=round(data_sgmnt./delta_rep);% номера уровней квантования
% по сегментам

% сигнал на выходе ЦАП
dataDACsgmnt=data_q.*delta_rep;% сегментированный сигнал на выходе
ЦАП
dataDAC=dataDACsgmnt(:);% объединение сегментов

% по сегментам определяем отношение с/ш
data_sgmnt2=buffer(data,n_sgmnt2);
dataDAC_sgmnt2=buffer(dataDAC,n_sgmnt2);

% Определяем мощность шума для каждого сегмента сигнала
shum_sgmnt2=(dataDAC_sgmnt2(:,1:length(data_sgmnt2))-data_sgmnt2);
Pshum_sgmnt2=var(shum_sgmnt2);
% Определяем мощность сигнала для каждого сегмента
Ps_sgmnt2=var(data_sgmnt2);

% Определяем отношение с/ш для каждого сегмента
q_sgmnt02=Ps_sgmnt2./Pshum_sgmnt2;
q_sgmnt2=q_sgmnt02(1:end-1);
qdB_sgmnt2=10*log10(q_sgmnt2);

% определяем среднее значение отношения с/ш
noise=dataDAC(1:length(data))-data;
meanSNR=10*log10(var(data)/var(noise));
% mean_qdB_sgmnt2=mean(qdB_sgmnt2);

```

```

% определяем среднее квадратическое значение
% разброса величин отношения с/ш
std_qdB_sgmnt2=std(qdB_sgmnt2);

subplot(4,1,1);stem(Ps_sgmnt2);grid on;title('Мощность сигнала
посегментно');
subplot(4,1,2);stem(delta);grid on;title ('Изменение шага квантования
во времени');
subplot(4,1,4);stem(qdB_sgmnt2);grid on;title('Отношение с/ш
посегментно для адапт. квант-я');
subplot(4,1,3);stem(qdB_sgmnt200);grid on;title('Отношение с/ш
посегментно для квант. с пост. шагом');
figure;
subplot(2,1,1);stem(dataDAC00);grid on;title('Сигнал после квантования
с постоянным шагом');
subplot(2,1,2);stem(dataDAC);grid on;title('Сигнал после квантования
с изменяемым шагом');

% soundsc(dataDAC);% контрольное прослушивание
% soundsc(data);

```

Лабораторная работа № 4

ИССЛЕДОВАНИЕ РАЗНОСТНОГО КВАНТОВАТЕЛЯ

Листинг сценария (QuantDiff)

```

clear;close all;
name='один.wav';% name-имя звукового файла
% name='m7.wav';
q_dB=60;
ds=200;% размер сегмента (кол-во отсчетов)
do=100;% размер перекрытия соседних сегментов
pre_order=12;% порядок предсказания

[V,Fs]=audioread(name);
% [V,Fs]=wavread(name);

```

```

[B0,B,delta,data_q]=RaschetQvant4(V,q_dB);
%предыскажающий фильтр
xf=filter([1 -.95],1,V);
%Разбиваем сигнал на блоки с указанным размером перекрытия
xb=buffer(xf,ds,do);
%каждый блок сигнала "взвешивается" окном Хэмминга
[m,n]=size(xb);
w=hamming(m);
xh=repmat(w,1,n).*xb;

%Вычисляем корреляционные функции отдельных
%блоков (столбцов) сигнала
%вычисляем корреляционную функцию первого столбца
xk1=xcorr(xh(:,1),pre_order,'biased');
%вычисляем корреляционные функции других столбцов
if n>1
    for i=2:n
        xk1=cat(2,xk1,xcorr(xh(:,i),pre_order,'biased'));    %объединяем
корреляционные функции
    end
end
xk=xk1(pre_order+1:end,:);
%с помощью алгоритма Левинсона - Дарбина вычисляем
%вектор коэффициентов линейного предсказания - A0
%вектор коэффициента частной корреляции - K
%средний квадрат ошибки предсказания - E
[A0,E,K]=levinson(xk);
%преобразуем коэфф. линей. предсказания
%согласно стандарт. формы фильтра (нужен доступ к его состоянию)
A=-A0;
A(:,1)=0;

%определяем разностный сигнал
dataSegmnt=buffer(V,ds-do);
[m0,n0]=size(dataSegmnt);
[m1,n1]=size(A);
%Задаем начальное состояние

```

```

%предсказывающего фильтра
Z1=zeros(n1-1,1);
%Задаем значение первого отсчета
% сигнала на входе фильтра
start=0;
% сегменты квантованного речевого сигнала
quantv=zeros(m0,m1);
% сегменты квантованного разностного сигнала
s=zeros(m0,m1);
% числа, соответствующие результатам квантования
% сегментов разностного сигнала
dQuant=zeros(m0,m1);

for k3=1:m1
    segmnt=dataSegmnt(:,k3);
    sgmntA=A(k3,:);
    v=sgmnt;
    for k2=1:m0
        if k2==1
            [predictv,Z]=filter(sgmntA,1,start,Z1);
        else
            [predictv,Z]=filter(sgmntA,1,start2,Z);
        end
        d=v(k2)-predictv;
        dQuant(k2,k3)=round(d/delta);
        s(k2,k3)=dQuant(k2,k3)*delta;
        quantv(k2,k3)=predictv+s(k2,k3);
        start2=quantv(k2,k3);
    end
    start=quantv(k2,k3);
    Z1=Z;
end
OUTs=s(:);% квантованный разностный сигнал
OUTv=quantv(:); %квантованный входной сигнал,
% сформированный на передающей стороне
% из квантованного разностного сигнала
figure; plot(OUTs);grid on;title('OUTs')

```

```

figure; plot(OUTv);grid on;title('OUTv')
figure;plot(V);grid on;title('V');

%Приемная сторона
%Задаем нулевое начальное состояние
%предсказывающего фильтра
ZZ1=zeros(n1-1,1);
%Задаем нулевое значение первого отсчета
%сигнала на входе фильтра
sstart=0;

%Формируем сегменты квантованного сигнала
quantvv=zeros(m0,m1);
for k33=1:m1
    ssegmnt=dataSegmnt(:,k33);
    ssgmntA=A(k33,:);
    vv=ssegmnt;
    for k22=1:length(vv)
        if k22==1
            [predictvv,ZZ]=filter(ssgmntA,1,sstart,ZZ1);
        else
            [predictvv,ZZ]=filter(ssgmntA,1,sstart2,ZZ);
        end
        quantvv(k22,k33)=predictvv+s(k22,k33);
        sstart2=quantvv(k22,k33);
    end
    sstart=quantvv(k22,k33);
    ZZ1=ZZ;
end
OUTvv=quantvv(:);%объединяем сегменты квант. сигнала
G=var(OUTvv)/var(OUTs);%коэффициент усиления системы
VARvv=var(OUTvv);%дисперсия квантованного сигнала
%на приемной стороне
varV=var(V);
varOUTs=var(OUTs);%дисперсия разностного сигнала
figure;plot(OUTvv);grid on;title('OUTvv');
%soundsc(OUTvv,Fs);

```

```

%soundsc(V,16e3);
VARshum=var(OUTvv(1:length(V))-V);
VAR2shum=delta^2/12;
SNRvv=VARvv/VAR2shum;
SNRs=varOUTs/VAR2shum;
dbSNRvv=10*log10(SNRvv);
dbSNRs=10*log10(SNRs);

```

Лабораторная работа № 5

ИССЛЕДОВАНИЕ LPC-КОДЕКА РЕЧЕВОГО СИГНАЛА

Листинг общего сценария для проведения исследований (exe06_10_23_2.m)

```

close all;clear;
[V0,Fs,b]=wavread('m0_4.wav');
%[V0,Fs]=audioread('m0_4.wav');
%plot(V0);grid on;title('Заданный звуковой сигнал')
V=V0(25e3:38e3);%Выделяем заданный фрагмент сигнала

ds=200;%размер сегмента (кол-во отсчетов)
do=100;%размер перекрытия соседних сегментов
pre_order=10;%порядок предсказания

%вычисление K коэффициентов частной корреляции фильтра,
%энергии E остатка предсказания
[K,E,n]=analyse2(V,ds,do,pre_order);
G=E.^0.5;%коэффициент усиления

%Формирование сигнала остатка предсказания
Predct_V=filt_error2(V,ds,do,K,n);

%Определяем зависимость энергии
%остатка предсказания от порядка предсказания
%и выбираем нужное значение порядка предсказания
pre_order1max=15;%Максимальное значение порядка предсказания
EE=zeros(pre_order1max,1);

```

```

k=0;
for pre_order1=1:pre_order1_max
    k=k+1;
    [K1,E1,n1]=analyse2(V,ds,do,pre_order1);
    EE(k)=mean(E1);
end
figure;stem(EE);grid on;title('Зависимость энергии остатка предсказания
от порядка предсказания');

%Синтезируем сигнал по остатку предсказания
Predct_V_sgmnt=buffer(Predct_V,ds-do);
[filt_synth]=synthesis2(K,Predct_V_sgmnt,n);
%Пропускаем синтезированный сигнал через корректирующий ФНЧ
Filt_xf=filter(1,[1 -.95],filt_synth);
Filt_xf_corr=Filt_xf(1:length(V));

figure;
subplot(4,1,1);plot(V0);grid on;title('Заданный сигнал');
subplot(4,1,2);plot(V);grid on;title('Заданный фрагмент сигнала');
subplot(4,1,3);plot(Filt_xf_corr);grid on;title('Сигнал, синтезированный
по остатку предсказания');
subplot(4,1,4);plot(Predct_V);grid on;title('Сигнал остатка предсказания');

class2=ones(1,n);%предварительное формирование выходного сигнала
%идеального классификатора (матрица классификатора, заполненная
единицами,
%для последующего частичного заполнения нулями)
%Ставим нули для сегментов пауз и невокализованных звуков
class2(1:8)=zeros(1,8);
class2(11:13)=zeros(1,3);
class2(30:87)=zeros(1,58);
class2(91:99)=zeros(1,9);
class2(124:n)=zeros(1,8);
%Полученная переменная class2 - это результат идеальной классификации
%сегментов (вокализованный/невокализованный)

```

```

% Определение параметров реализуемой классификации
porog=0.55; % установка порогового значения
porogGraph=zeros(n,1)+porog;% массив пороговых значений
% для графического отображения

% Определение функции AMDF и результата
% автоматической классификации
[AMDF,class1,class,pitch]=AMDF3(Predct_V,ds,do,porog);
figure; stem(AMDF(:,107)); grid on; title('AMDF');% просмотр
% нормированной по максимуму функции AMDF для выбранного
% сегмента

figure;
subplot(4,1,1);stem(class2);grid on;title('Результат идеальной
классификации')
subplot(4,1,2);stem(class);grid on;title('Результат реализованной
классификации')
subplot(4,1,3);stem(class1);grid on;title('Отношение ymin/ymax');
hold on;plot(porogGraph,'r');
hold off;
subplot(4,1,4);stem(pitch);grid on;title('Период основного тона')

T=76;% Выбранное значение периода основного тона
class=class2;% использование идеальной классификации при формировании
сигнала возбуждения
[Vozbud]=sigVozbud(class,T,G,n,do,Fs);% формирование сигнала
возбуждения

% Синтез сигнала
sgmntVozbud=buffer(Vozbud,do,0);
[fil_synth]=synthesis2(K,sgmntVozbud,n);

% Пропускаем синтезированный сигнал через корректирующий ФНЧ
Filtr_xf=filter(1,[1 -.95],fil_synth);
Filtr_xf_corr=Filtr_xf(1:length(V));

figure;
subplot(4,1,1);plot(V);grid on;title('Заданный фрагмент сигнала');

```

```

subplot(4,1,2);plot(Filtr_xf_corr);grid on;title('Синтезированный
фрагмент сигнала');
subplot(4,1,3);plot(Vozbud);grid on;title('Реализованный сигнал
возбуждения');
subplot(4,1,4);plot(Predct_V);grid on;title('Сигнал остатка предсказания');

soundsc(Filtr_xf_corr)

```

Листинги функций

```

function [AMDF,class1,class,pitch]=AMDF3(Predct_V,ds,do,porog)
% Определяется y1 - функция AMDF
% величина задержки k меняется от 0 до 159
% Вход : Predct_V - сигнал остатка предсказания
% ds - длительность сегмента (интервал усреднения)
% do - размер перекрытия сегмента
% Выход: y1 - функция AMDF
% AMDF - переменная (для параметра классификации)
Predct_V1=buffer(Predct_V,ds,do);
% G=std(Predct_V1);% вычисляются коэффициенты усиления
[p1,p2]=size(Predct_V1);% p2-количество сегментов в сигнале
n1=p2*do;
y1=zeros(160,p2);% пустая матрица AMDF
b=1;% счетчик
for n=1:do:n1-do-159
    for k=0:159% в данном цикле происходит вычисление функции
    AMDF
        V1=Predct_V(n:(n+do-1));% вычисляется значение d, суммируется
        Vk=Predct_V((n+k):(n+do-1+k));% результат записывается в элемент
        d=abs(V1-Vk);% k-той строки b-того столбца
        y1(k+1,b)=sum(d);
    end
    b=b+1;
end
% AMDF=zeros(160,p2);

```

```

%for f=1:p2
%M1=max(y1(:,f));
%AMDF(:,f)=y1(:,f)/M1;
% end

[y11,y22]=size(y1);%y22-количество сегментов в AMDF
AMDF=zeros(160,p2);
class=zeros(1,y22);% пустая матрица классификатора
pitch=zeros(1,y22);% пустая матрица периода ОТ
class1=zeros(1,y22);% матрица отношений  $r=y_{\min}/y_{\max}$ ;
for j=1:y22
    y2=y1(16:end,:);% определение параметра классификации
    [ymin, imin]=min(y2(:,j));
    ymax=max(y2(:,j));
    r=ymin/ymax;
    class1(1,j)=r;% матрица отношений  $r=y_{\min}/y_{\max}$ ,
    pitch(1,j)=imin+15;% определение периода основного тона
    %каждый элемент которой сравнивается с пороговым значением
porog
    %если r меньше порогового, то сегмент вокализованный, и наоборот
AMDF(:,j)=y1(:,j)/ymax;%AMDF, нормированная по максимуму
    if r<porog
        class(1,j)=1;
        %pitch(1,j)=imin+15;% определение периода основного тона
    else
        class(1,j)=0;
        %pitch(1,j)=0;
    end
end

end

function[K,E,n]=analyse2(V,ds,do,pre_order)
% Данная функция производит анализ входного сигнала
% Вход : V-входной сигнал
% ds-длина сегмента
% do-длина перекрытия сегментов
% pre_order-максимальный порядок предсказания

```

```

%Выход : E-средний квадрат ошибки предсказания
%K-коэффициенты частной корреляции
%n-количество сегментов в сигнале
% xf-входной сигнал, прошедший через предсказывающий фильтр

xf=filter([1 -.95],1,V);
%буфер разбивает сигнал на блоки с указанным размером перекрытия
xb=buffer(xf,ds,do);

%"взвешиваем" сегменты сигнала окном Хэмминга
[m,n]=size(xb);
w=hamming(m);%окно Хэмминга
xh=repmat(w,1,n).*xb;

%Вычисляем корреляционную функцию отдельных блоков сигнала
%и формируем единую корреляционную функцию сигнала
%путем соединения отдельных блоков, для
%которых рассчитана корреляционная функция
%задаем максимальный порядок предсказания
%вычисляем корреляционную функцию первого столбца
xk1=xcorr(xh(:,1),pre_order,'biased');
%вычисляем корреляционные функции других столбцов
if n>1
    for i=2:n
        xk1=cat(2,xk1,xcorr(xh(:,i),pre_order,'biased'));
%объединяем корреляционные функции
    end;
end
xk=xk1(pre_order+1:end,:);
%с помощью алгоритма Левинсона - Дарбина вычисляем
%K-коэффициенты частной корреляции
%A-коэффициенты линейного предсказания
%E-средний квадрат ошибки предсказания
[A,E,K]=levinson(xk);
End

```

```

function Predct_V=filt_error2(V,ds,do,K,n)
% Данная функция пропускает входной сигнал через рекурсивный
% фильтр
% на выходе получаем сигнал остатка предсказания
% Вход : K-коэффициенты частной корреляции
% ds-длина сегмента
% do-длина перекрытия сегментов
% n-количество сегментов в сигнале
% Выход: Predct_V-сигнал ошибки предсказания

% xf-входной сигнал, прошедший через предыскажающий фильтр
xf=filter([1 -.95],1,V);
% делим сигнал на блоки без перекрытия
xb_error=buffer(xf,ds-do,0);
K_error=K(:,1);
% пропускаем первый блок сигнала и соответствующие ему коэффициенты
% частной корреляции через нерекурсивный лестничный фильтр
Hd_error=dfilt.latticemamin(K_error);
Predct_V=filter(Hd_error,xb_error(:,1));
% проделываем эту же операцию с остальными блоками
for q=(2:1:n)
    K_error=K(:,q);
    Hd_error_part=dfilt.latticemamin(K_error);
    Predct_V2=filter(Hd_error_part,xb_error(:,q));% объединяем результаты
    Predct_V3=cat(1,Predct_V,Predct_V2);
% получаем сигнал остатка предсказания
    Predct_V=Predct_V3;
end

```

```

function Vozbud=sigVozbud(class,T,G,n,do,Fs)
% функция формирует сигнал возбуждения синтезирующего фильтра
% class - результат классификации сегментов (вокал/невокал)
% T - выбранное значение периода основного тона
% G - коэффициент усиления;

```

```

%n - число сегментов сигнала
%do - размер перекрытия соседних сегментов
%Fs - частота дискретизации
%Vozbud - сигнал возбуждения синтезирующего фильтра

%Формирование шумовой составляющей сигнала возбуждения
Rand_sig=0.5*randn(n*do,1);
sgmntRand_sig=buffer(Rand_sig,do,0);
invertClass=1-class;
GinvertClass=G'.*invertClass;
sgmntGInvertClass= repmat(GinvertClass,do,1);
sgmntNoVocal=sgmntGInvertClass.*sgmntRand_sig;
NoVocal=sgmntNoVocal(:);

%Формирование импульсной составляющей сигнала возбуждения
t=(0:do*n-1)/Fs;
ImpSig0=square(2*pi*t*Fs/T,1/T*100)+1;
ImpSig=ImpSig0-mean(ImpSig0);
sgmntImpSig=buffer(ImpSig,100,0);
GClass=G'.*class;
sgmntGClass= repmat(GClass,do,1);
sgmntVocal=sgmntGClass.*sgmntImpSig;
Vocal=sgmntVocal(:);
%Объединение импульсной и шумовой составляющих сигнала
возбуждения
Vozbud=Vocal+NoVocal;

function[fil_synth]=synthesis2(K,M_signal,n);
%Данная функция пропускает входной сигнал через рекурсивный
фильтр
%на выходе получаем синтезированный сигнал
%Вход : K - кчк
%M_signal - сигнал возбуждения, помноженный на коэффициент
усиления
%Выход: fil_synth - синтезированный сигнал

```

```
% пропускаем первый блок сигнала и соответствующие ему коэффициенты
% частной корреляции через не рекурсивный лестничный фильтр
K_synth=K(:,1);
Hd=dfilt.latticear(K_synth);
fil_synth=filter(Hd,M_signal(:,1));
% проделываем эту же операцию с остальными блоками
[m11,n11]=size(M_signal);
for h=2:n11
    K_synth=K(:,h);
    Hd=dfilt.latticear(K_synth);
    fil_synth1=filter(Hd,M_signal(:,h));
    fil_synth2=cat(1,fil_synth,fil_synth1);
    fil_synth=fil_synth2;
end
```

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
Лабораторная работа № 1. Исследование квантователя с постоянным шагом квантования	5
Лабораторная работа № 2. Исследование квантователя, использующего компандирование сигналов	16
Лабораторная работа № 3. Исследование адаптивного квантователя	25
Лабораторная работа № 4. Исследование разностного квантователя	33
Лабораторная работа № 5. Исследование LPC-кодека речевого сигнала	43
ЗАКЛЮЧЕНИЕ.....	58
РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК	59
ПРИЛОЖЕНИЕ	60

Учебное издание

ЛЕВИН Евгений Калманович

ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОБРАБОТКИ РЕЧЕВЫХ СИГНАЛОВ
В ТЕЛЕФОНИИ

Лабораторный практикум

Редактор Е. А. Платонова
Технический редактор Ш. Ш. Амирсейидов
Компьютерная верстка Е. А. Кузьминой, А. Н. Герасина
Корректор О. В. Балашова
Выпускающий редактор А. А. Амирсейидова

Подписано в печать 18.04.24.
Формат 60×84/16. Усл. печ. л. 4,88. Тираж 30 экз.

Заказ

Издательство

Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых.
600000, Владимир, ул. Горького, 87.