**Московский государственный технический университет**

**имени Н.Э. Баумана**

**Калужский филиал**



**Квашенников В.В.**

**ТЕОРИЯ И ПРОЕКТИРОВАНИЕ СИСТЕМ**

**ПОМЕХОУСТОЙЧИВОЙ ПЕРЕДАЧИ ИНФОРМАЦИИ**

Методические указания к выполнению лабораторных работ

по курсу "Теория и проектирование телекоммуникационных систем"

(3-е переработанное издание)

**Калуга 2022**

**Содержание**

Введение……………………………………………………………….. 3

1. Моделирование двоичного симметричного канала с независимыми ошибками……………………………………………………………..…. 6

2. Моделирование канала с группированием ошибок. Марковская модель канала с двумя состояниями (модель Гилберта-Эллиота). Модель канала по Пуртову…………………………………..…………………….. 13

3. Оптимальный корреляционный приемник. Автокорреляционный прием. Демодулятор однократной ОФМ…………………….…………. 20

4. Частотная модуляция. Демодулятор сигналов ЧМ……………….... 28

5. Тактовая синхронизация. Векторное коррекционное устройства (КУ). Программа работы векторного КУ………………………………….. 33

6. Цикловая синхронизация. Стартстопная и кодовая цикловая синхронизация. ……. ……………………………………………………………….42

**Введение**

Проектирование перспективной телекоммуникационной аппаратуры является сложной научно – технической задачей, для решения которой используют современные инженерные методики проектирования [1,2]. Важным при проектировании имеет обеспечение высоких показателей помехоустойчивости, быстродействия, надежности и устойчивости функционирования аппаратуры.

В данном учебном пособии представлены лабораторные работы, затрагивающие вопросы создания узлов аппаратуры или контрольно-испытательного оборудования: модели дискретных каналов связи, модуляция и демодуляция сигналов, тактовая и цикловая синхронизация, кодирование и декодирование линейных помехоустойчивых кодов и расчет их помехоустойчивости. Рассмотрение этих вопросов необходимо для проектирования телекоммуникационной аппаратуры передачи и приема дискретной информации.

Укрупненная схема телекоммуникационной системы, представлена на рис.В.1. Система состоит из передающей стороны, канала связи и приемной стороны.

На передающей стороне источник сообщений генерирует биты исходной информации *a0,a1,…,ak-1*, которые далее поступают на вход кодирующего устройства помехоустойчивого кода. Кодер помехоустойчивого кода преобразует *k*-битные двоичные комбинации в *n*-битные слова помехоустойчивого кода *b0,b1,…,bn-1 (n>k).* Затем символы кодовых слов подаются на вход модулятора канала. Модулятор канала изменяет некоторые параметры переносчика сигнала (синусоида высокой частоты, импульсная последовательность) в соответствии с передаваемой информацией.

В канале связи на передаваемый сигнал воздействует помеха, которая приводит к искажениям сигнала. На приемной стороне из-за искажений сигнала могут возникать ошибки. Демодулятор преобразует аналоговый сигнал, поступающий из канала, в цифровую форму, соответствующую передаваемой двоичной информации. На выходе демодулятора сигнал дискретизирован по уровню: логический «0» либо логическая «1». Дискретизацию сигнала по времени осуществляет приемник элементов кода совместно с устройством тактовой синхронизации или коррекционным устройством (КУ). КУ определяет границы неискаженных кодовых посылок, приемник элементов кода восстанавливает длительность кодовых посылок. Устройство цикловой синхронизации или фазирования (УЦФ) предназначено для определения начала слова помехоустойчивого кода. Декодер помехоустойчивого кода выполняет декодирование принятого кодового слова. В случае успешного декодирования информация выдается получателю сообщения.

Часть системы связи, начиная с выхода кодера помехоустойчивого кода до входа УЦФ, можно рассматривать как дискретный канал связи. Кодовое слово *B=b0,b1,…,bn-1* с выхода кодера трансформируется в дискретном канале и на вход УЦФ поступает кодовое слово с ошибками *C=c0,c1,…,cn-1*, где *ci=bi xor ei i=0,…,n-1, E=e0,e1,…,en-1*–вектор ошибок.

Для оценивания параметров системы передачи данных: вероятности правильного приема, стирания, трансформации сообщения, оптимизации алгоритмов кодирования, декодирования, тактовой и цикловой синхронизации, методов модуляции и демодуляции необходимо знание характеристик канала связи. Поэтому, две первые лабораторные работы посвящены рассмотрению алгоритмов и программ дискретных моделей каналов связи. Первая работа рассматривает генерацию потока ошибок и характеристики канала с независимыми ошибками, вторая – канала с группирующимися ошибками по модели Гильберта-Эллиота.

В следующих работах исследуются методы модуляции и демодуляции и представлено моделирование модема однократной ОФМ и ЧМ.

Затем изучаются устройства тактовой синхронизации: коррекционные устройства (КУ) с авторегулированием, векторные КУ, гистограммные и другие.

Одна лабораторная работа посвящена моделированию устройств цикловой синхронизации. При этом рассмотрены старт-стопные и кодовые устройства цикловой синхронизации.

В следующих работах приведены основные алгоритмы и программы кодирования и декодирования помехоустойчивых кодов БЧХ и Рида-Соломона. Рассмотрено кодирование линейных двоичных кодов при их матричном и полиномиальном описании. Кодирование состоит в умножении исходной информации на порождающую матрицу кода или делении информационного многочлена на образующий многочлен кода. Для декодирования кодов применяют перестановочный алгоритм декодирования Касами-Рудольфа и синдромное декодирование по схеме Меггита. Кодирование недвоичных кодов Рида-Соломона в спектральной области осуществляют по схеме Горнера вычисления значений информационного многочлена в различных точках поля. Для декодирования кода Рида-Соломона с исправлением стираний используют интерполяцию по формуле Ньютона.

Описание каждой лабораторной работы состоит из краткой теоретической части, алгоритма работы рассматриваемого функционального узла, задания исходных данных и выходных данных для проверки результатов работы.

Итогом лабораторной работы должно быть составление программы и проверка результатов ее работы. Программы составляют на одном из алгоритмических языков высокого уровня, например на языке Delphi, Piton, C++ и др., и выполняются на ЭВМ.

Для проверки выполнения работы в методическом пособии приведены исходные тексты программ, с которыми студентам рекомендуется знакомиться только после самостоятельного составления программ. К каждой программе прилагаются контрольные примеры. Программы содержат диалог с пользователем. Результаты вычислений выводятся на экран монитора.

Кодер

помехоуст.

кода

Модулятор

Источник

сообщений

Передающая сторона

Канал связи

Помехи

Приемная сторона

Приемник

элементов

кода

(“1”,”0”)

Декодер

помехоуст.

кода

Получа-

тель

сообщения

Демоду-

лятор

Устройство

цикловой

синхрони-

зации

Устройство

тактовой

синхрони-

зации

Рис.В.1. Схема системы передачи дискретной информации

**1. Моделирование двоичного симметричного канала с независимыми ошибками**

Простейшей моделью канала связи является двоичный симметричный канал с независимыми ошибками.

Канал называется двоичным, если передаются двоичные символы информации 0 либо 1.

Канал симметричный, если вероятность возникновения ошибки не зависит от вида передаваемых символов:

*P(e/b=0)=P(e/b=1),*  (1.1)

где *e/b=0 и e/b=1* означают соответственно события возникновения ошибки *e* =1 при условии, что в канале передавался символ *b=0* или *b=1*.

В канале с независимыми ошибками вероятность возникновения ошибки не зависит от того, какие комбинации ошибок предшествовали данной ошибке:

*P(e0/e-1,e-2,..)=const,*  (1.2)

для любой последовательности ошибок *e-1,e-2,*…,

где *e0/e-1,e-2*,…- событие возникновения ошибки на текущей позиции, при условии, что ей предшествовала комбинация ошибок *e-1,e-2,….*

Модель канала с независимыми ошибками применяют для описания многих каналов: спутниковых, УКВ каналов в пределах прямой видимости, кабельных и волоконно-оптических линий связи. При использовании в канале с группированием ошибок перемежения символов канал приближается к каналу с независимыми ошибками.

Для генерации потока независимых ошибок с некоторой вероятностью *p* используется стандартная функция генерации случайных чисел алгоритмического языка программирования *r:=random*. Эта функция генерирует равномерно распределенные действительные числа в диапазоне от 0 до1. График плотности распределения *w(r)* случайного числа *r:=random* представлена на рис.1.1 в виде квадрата со стороной, равной 1.

*w(r)*

1

0  *r1* 1 *r*

Рис.1.1.

Вероятность того, что некоторое случайное число *r≤ r1*, равна площади заштрихованного на графике прямоугольника

*P(r≤r1)*==*r1* , (1.3)

и вероятность события *r≤ r1* есть *r1*.

Значит, для получения ошибки с вероятностью *p* достаточно проверить выполнение условия *r≤ p.*

# Алгоритм генерации массива ошибок длины n байт с вероятностью *p*.

Начало

# 

i := 1

Ввод

p - вероятность ошибки на бит

n - длина массива ошибок в байтах

b := 128

a[i] := 0

Random < p

нет

да

a[i] := a[i] + b

b = 1

нет

да

b :=b shr 1

нет

i = n

i := i+1

да

Вывод

массива ошибок a

Конец

var

i,n : integer;

b : byte;

p: real;

a: array[1..16384] of byte;

Рис. 1.2.

Входными данными программы генерации ошибок являются вероятность *p* ошибки на бит в канале и длина (количество байт) *n* генерируемого массива ошибок. Программа выводит поток ошибок на экран монитора. Часть программы, оформленная в виде пользовательского модуля unit kan1, генерирует один байт ошибок. В управляющей части modkan1 программы задаются входные данные программы и выводятся результаты работы программы на экран монитора.

unit kan1; {модуль моделирования канала с независимыми ошибками}

interface procedure er(p:real;var a:byte);

implementation

procedure er;

var

b:byte;

begin

while b<>0 do begin

b:=$80;a:=0;

if random<p then a:=a+b;

b:=b shr 1; end

end

end.

program modkan1; {программа моделирования канала с независимыми ошибками}

uses kan1;

var

i,n:integer;

a:byte;

p:real;

begin

write('Введите вероятность ошибки на бит в канале');

readln(p); write('Длина массива ошибок в байтах ?');readln(n);

randomize;

for i:=1 to n do begin

er(p,a);

b:=$80;

while b<>0 do begin

if a and b<>0 then write(1)else write(0);

b:=b shr 1; end

end;

writeln;readln;

end.

Пример.

Одна из реализаций потока ошибок для вероятности ошибок на бит *p*=0.1.

0000000000000010010000000000000000001010

0000000000100000000000000001000000100001

1010000100001000000000100000000010000000

0000000010100000000000000000001000100000

0000000100001000000000000000000010000000

0000010000000000100000000000000000100000

0000000000001000000000000000000000000000

0100000000100000000001001000000010000100

**Параметры канала с независимыми ошибками. Средняя**

**вероятность ошибок на бит в канале. Блочная статистика ошибок.**

Эффективной несмещенной оценкой средней вероятности *p* ошибки на бит в канале является частота ошибок. Если на длине *n* бит произошло *t* ошибок, то частота ошибок запишется

 (2.1)

Для независимых ошибок в канале блочная статистика ошибок подчиняется биномиальному закону распределения. Вероятность *t* ошибок в блоке длины *n* бит равна

, (2.2)

где   - биномиальный коэффициент,

*p* – средняя вероятность ошибок на бит в канале.

В программе на модели независимых ошибок оценивается частота ошибок *p1*по формуле (2.1). Точность совпадения частоты ошибок *p1* и средней вероятностью ошибки на бит *p* в определенной степени зависит от качества программного датчика случайных чисел и числа испытаний.

На модели вычисляется блочная статистика ошибок

  *t=0…n*, (2.3)

где *Q(n,t)* – количество комбинаций длины n бит с *t* ошибками,

*m* - количество испытаний.

Совпадение экспериментальной блочной статистики по формуле (2.3), с теоретической блочной статистикой, вычисленной по формуле (2.2), позволяет сделать вывод о независимости ошибок в модели канала.

Алгоритм вычисления вероятности появления *t (0<= t <= 8)* ошибок в блоке длины 8 бит(байт) на модели канала с независимыми

ошибками.

Начало

Ввод

p – вероятность ошибки на бит

n – длина массива ошибок в байтах

i:=1

p1:=p3[0]:=p3[1]:=p3[2]:=…p3[8]:=0

Генерация одного байта ошибок

er (p,a)

b :=128

j:=0

нет

(a and b) =0

j:=j+1

да

нет

b := b shr 1

b=1

да

p3[j] := p3[j] +1; p1:= p1+j

нет

i := i+1

i=n

да

Вывод блочной статистики

p3[i] / n , i = 0..8

Средняя вероятность ошибки

p1 / (8\*n)

конец

var

i,j,n:integer;

a,b:byte;

p1,p:real;

p2,p3:array[0..8] of real; Рис. 2.1.

В программе на модели канала с независимыми ошибками вычисляется частота *p1* ошибок на бит и блочная статистика *P3(n,t)* канала*.* Для генерации ошибок используется процедура er(p,a) из модуля kan1. Входными данными программы являются средняя вероятность *p* ошибки на бит в канале и длина *n* массива ошибок. В программе вычисляются теоретические вероятности *P2(n,t)* биномиального распределения ошибок.

program modkan2;

{программа моделирования канала с независимыми ошибкам}

uses kan1;

var

i,j,n:integer;

a,b:byte;

p,p1:real;

p2,p3:array[0..8] of real;

const

c:array[0..8] of integer=(1,8,28,56,70,56,28,8,1);

begin

write('Введите вероятность ошибки на бит в канале ');readln(p);

write('Длина массива ошибок в байтах ?');readln(n);

randomize;

for i:=0 to 8 do p3[i]:=0;

for i:=1 to n do begin

er(p,a);

b:=$80;j:=0;

while b<>0 do begin

if a and b <> 0 then j:=j+1;

b:=b shr 1; end;

p3[j]:=p3[j]+1;

end;

writeln;writeln('Распределение по модели');

for i:=0 to 8 do write(p3[i]/n:1:3,' ');

writeln;writeln('Теоретическое распределение');

p2[0]:=exp(8\*ln(1-p));for i:=1 to 8 do p2[i]:=p2[i-1]\*p/(1-p);

for i:=0 to 8 do p2[i]:=c[i]\*p2[i];

for i:=0 to 8 do write(p2[i]:1:3,' ');

writeln;readln;

end.

Пример.

В таблице приведены вероятности появления *t* ошибок в блоке длины один 8 бит (1 байт) при средней вероятности ошибки на бит *p=0.1*, числе испытаний *n=10000*.

Число ошибок t

Канал

0 1 2 3 4 5 6 7 8

Модель

канала 0.435 0.377 0.150 0.034 0.005 0 0 0 0

Биномиальное

распределение 0.430 0.383 0.149 0.033 0.005 0 0 0 0

**2. Моделирование канала с группированием ошибок. Марковская модель канала с двумя состояниями (модель Гилберта-Эллиота). Модель канала с группированием ошибок по Пуртову.**

Канал, который не является каналом с независимыми ошибками, называют каналом с группированием ошибок. Одной из моделей канала с группированием ошибок является марковская модель [3, 4]. В этой модели рассматривается последовательность состояний канала *S0,S1,…,Sm-1*. В каждом из состояний ошибки возникают с вероятностью соответственно *p0,p1,…,pm-1*.Последовательность состояний канала *S0,S1,…,Sm-1* образует простую однородную цепь Маркова. Цепь Маркова определяется матрицей переходных вероятностей

*P* = || *pij* || *i,j = 0..m-1*, (3.1)

где *pij*есть вероятность перехода канала в состояние *Sj*, если известно, что в предыдущий момент времени канал находился в состоянии *Si*.

Простейшей моделью канала с группированием ошибок является марковская модель канала с двумя состояниями или модель Гилберта-Эллиота. Эта модель описывается матрицей переходных вероятностей

*p00 p01*

*P* = (3.2)

*p10 p11*

и вероятностью ошибок в каждом состоянии *p0* и *p1* (*i,j*=0,1).

Обычно *p0<<p1* и состояние *S0*, в котором вероятность ошибки мала, называется “хорошим” состоянием, а состояние канала *S1* с большой вероятностью ошибки на бит - “плохим” состоянием.

Марковская модель канала позволяет достаточно просто генерировать поток ошибок канала.

Помимо марковских моделей ошибок с группированием существует много других моделей канала. Одной из широко известных моделей канала с группированием является модель Пуртова [5]. Эта модель задает блочную статистику ошибок в канале:

*P(≥t, n ) = ( n/t )1-a p* , (3.3)

где *P(≥t,n )* – вероятность *t* и более ошибок в блоке длины *n* бит,

*a* – коэффициент группирования ошибок в канале ( *0≤a≤1* ),

*p* – средняя вероятность ошибок на бит в канале.

При увеличении *а* от 0 до 1 группирование ошибок в канале возрастает. При *a*=0 модель достаточно близка к каналу с независимыми ошибками, а при *a*=1 ошибки группируются в одном пакете. Пуртовская модель канала удобна для теоретических расчетов характеристик помехоустойчивых кодов, однако генерация потока ошибок по этой модели вызывает существенные затруднения. Модель становится некорректной при больших величинах *n* и *a*. Известны модификации этой модели, свободные от перечисленных недостатков. Для канала с группированием ошибок согласно модифицированной модели канала Пуртова вероятность *t* и более ошибок (*t*≥2) в блоке длины *n* бит выражается формулой

, (3.4)

где ,

*p -*  средняя вероятность на бит в канале связи, *a* - коэффициент группирования ошибок (0 ≤ *a* ≤ 1).

Вероятность искажения слова помехоустойчивого кода длины *n* бит для этой модели равна

. (3.5)

Вероятность безошибочного приема слов помехоустойчивого кода будет иметь вид

. (3.6)

Модифицированная модель канала Пуртова является достаточно общей и универсальной моделью. При соответствующем выборе своих параметров она с той или иной степенью приближения описывает многие каналы связи:

− КВ канал – *p* = 2.2· 10-3 – 5.7·10-2; *a* = 0.24 – 0.57;

− радиорелейные – *p* = 2.2· 10-6 – 9.1·10-3; *a* = 0.27 – 0.68;

− тропосферные – *p* = 1.2· 10-4 – 4.2·10-2;  *a* = 0.44 – 0.56;

− спутниковые – *p* = 2· 10-6 – 5·10-4; *a* = 0.41 – 0.5;

− коммутируемые ТЧ каналы городской телефонной сети –

*p* = 8.2· 10-6 – 1.5·10-2; *a* = 0.1 – 0.49.

Алгоритм генерации ошибок в марковской модели

канала Гилберта-Эллиота.

Начало

Ввод

p00,p11 – переходные вероятности марковской цепи

p0,p1 – вероятности ошибок в состояниях S0 и S1

n – длина массива ошибок в байтах

i := 1

S := 0

да нет

S = 0

Генерация одного байта

ошибок er (p0,a )

Генерация одного байта ошибок er (p1,a )

Вывод младшего

бита a and 1

Вывод младшего

бита a and 1

random<p11

random<p00

да нет да нет

S := 0

S := 1

S := 0

S := 1

нет

i = 8\*n

i:=i+1

да

Конец

var

p0,p1,p00,p11:real;

a,S:byte;

i,n:integer;

Рис.3.1.

Программа генерации ошибок состоит из модуля unit kan2, процедура которого erm(*p0,p1,p00,p11,a,S*) генерирует один бит ошибок в младшем разряде байта *a* в зависимости от текущего состояния канала *S*. В основной части программы задаются переходные вероятности *p00, p11* марковской цепи и безусловные вероятности ошибки *p0, p1* в различных состояниях канала, длина *n* потока ошибок. На экран выводится поток ошибок.

{модуль моделирования канала с группирующимися ошибками}

unit kan2;

interface

uses kan1;

procedure erm(p0,p1,p00,p11:real;var a,s:byte);

implementation

procedure erm;

var

b:byte;

begin

if s=0 then begin er(p0,a);

if random<p00 then s:=0 else s:=1;

end else

begin er(p1,a);

if random<p11 then s:=1 else s:=0;

end; end;

end.

program modkan3;

{программа моделирования канала Гилберта-Эллиота}

uses kan2;

var

i,n:integer;

a,s:byte;

p0,p1,p00,p11:real;

begin

write('Введите переходные вероятности марковской цепи'); readln(p00,p11);

write('Введите вероятности ошибок в состояниях s0 и s1 канала');readln(p0,p1);

write('Длина массива ошибок в байтах ?');readln(n);

randomize;s:=0;

for i:=1 to 8\*n do begin

erm(p0,p1,p00,p11,a,s);

write(a and 1);

end;

writeln;readln;

end.

Пример.

Для параметров модели *p00=0.8;p11=0.5;p0=0.01;p1=0.3*

одна из реализаций потока ошибок будет иметь вид:

0000001000001000000000100000000000000001

0000010001110000100000000000000011000000

0000100000000000000001000000000000000000

0000010000011000001100000000000000000000

0000000000000000000000000000001000001101

0000000000000000100110000000000101010000

**Исследование канала с группированием ошибок. Определение финальных и средней вероятностей ошибок модели канала Гильберта-Эллиота .**

Безусловные вероятности нахождения марковской модели канала в состояниях *S0,S1,…,Sm-1*называются финальными вероятностями *P0,P1,…,Pm-1*.

Пусть *Pф = | P0,P1,…,Pm-1|* - вектор-строка финальных вероятностей, *P = | pij |* - матрица переходных вероятностей марковской модели, тогда справедливы соотношения [3]

*Pф · I = 1 , PФ · P = Pф* , (4.1)

где *I* – единичный вектор столбец размерности *m.*

Для модели Гильберта-Эллиота *m=2* и соотношения для финальных вероятностей запишутся

1 *p00 p01*

|*P0 P1*|  *·* 1 = 1 , | *P0 P1*|  *·* *p10 p11* = | *P0 P1*|, (4.2)

или *P0 + P1 = 1 , P0 ·p00 + P1 ·p10 = P0 , P0 ·p01 + P1 ·p11 = P1.* (4.3)

# Откуда

*P0* = , *P1* = . (4.4)

Зная финальные вероятности марковской модели, нетрудно вычислить среднюю вероятность ошибки на бит

*p = P0 ·p0 + P1 ·p1.* (4.5)

Алгоритм вычисления финальных и средней вероятности

ошибок для модели канала Гилберта-Эллиота.

Начало

p00,p11,p0,p1 – вероятности марковской модели

n – длина массива ошибок в байтах

i := 1; S := 0

j0 := 0; j1 := 0

Генерация одного бита в модели Гилберта-

Эллиота erm (p0,p1,p00,p11,a,S )

j1 := j1+a

да

S = 0

нет

j0 := j0+1

нет

i := i+1

i = 8\*n

да

pf0 := j0/8n; pf1 := 1-pf0

p := j1/8n

Вывод

эксперементальных значений pf0,pf1,p

Конец КонецНачало

var

p00,p11,p0,p1,pf0,pf1,p :real;

i,n,j0,j1 :integer;

a,S :byte;

Рис. 4.1.

В программе оценивают финальные вероятности марковской модели в состоянии *S0* и *S1*. Среднюю вероятность ошибки на бит в канале оценивают частотой ошибок. Входными данными программы являются вероятности марковской модели *p00,p11,p0,p1* и длина массива ошибок, генерируемого в модели. Программа выводит на экран экспериментальные величины финальных и средней вероятностей, а также подсчитывает теоретические значения этих вероятностей по формулам (4.1), (4.2).

program modkan4;

{программа исследования модели канала Гилберта-Эллиота}

uses kan2;

var

i,n,j0,j1:integer;

a,s:byte;

p0,p1,p00,p11,pf0,pf1,p:real;

begin

write('Введите переходные вероятности марковской цепи'); ;readln(p00,p11);

write('Введите вероятности ошибок в состояниях s0 и s1 канала’); readln(p0,p1);

write('Длина массива ошибок в байтах ?');readln(n);

randomize;s:=0;j0:=0;j1:=0;

for i:=1 to 8\*n do begin

erm(p0,p1,p00,p11,a,s);

j1:=j1+(a and 1);

if s=0 then j0:=j0+1;

end;

writeln;

pf0:=j0/(8\*n);pf1:=1-pf0;p:=j1/(8\*n);

writeln('Экспериментальные значения');

writeln('pf0=',pf0,' pf1=',pf1,' p=',p);

writeln('Теоретические значения');

pf0:=(1-p11)/((1-p00)+(1-p11));pf1:=1-pf0;p:=pf0\*p0+pf1\*p1;

writeln('pf0=',pf0,' pf1=',pf1,' p=',p);

readln;

end.

Пример.

Финальные *Pf0,Pf1* и средняя *p* вероятности ошибок для модели канала Гилберта-Эллиота. Параметры модели:*p00*=0.8;*p11*=0.5;*p0*=0.01;*p1*=0.3; число испытаний *n*=10000.

|  |  |  |  |
| --- | --- | --- | --- |
| Канал | *Pf0* | *Pf1* | *p* |
| Модель  канала | 0.718 | 0.282 | 0.090 |
| Теоретическое  значение | 0.714 | 0.286 | 0.093 |

**3. Оптимальный корреляционный приемник. Автокорреляционный прием. Демодулятор однократной ОФМ.**

Для описания работы приемника сигналов представим посылки сигнала *s0(t)* и *s1(t),* передаваемые в двоичном канале в виде векторов *s0* и *s1* функционального пространства сигналов [6]. Пространство сигналов является метрическим функциональным пространством, в котором определено расстояние Евклида между векторами

 (5.1)

Из канала на вход приемника поступают искаженные помехой посылки сигнала или вектора *x0* и *x1* функционального пространства

 *i = 0,1*, (5.2)

где *ei* – вектор помехи.

При оптимальном приеме, принимается решение о передаче сигнала *s0*, если расстояние от принятого сигнала *xi* до сигнала *s0* не более расстояния от *xi* до *s1*. В противном случае принимается решение о приеме сигнала *s1*. Математически алгоритм оптимального приема запишется:

*s0* , если *d(xi,s0) < d(xi,s1)* (5.3)

 *xi* =

  *s1* , если *d(xi,s0) ≥ d(xi,s1),*

где *d* – расстояние Евклида между векторами.

Учитывая определение расстояния между векторами (5.1), алгоритм оптимального приема после несложных математических преобразований выразится в виде:

*s0* , если 

*xi*= (5.4)

*s1* , если 

Последние уравнения определяют алгоритм приема, который называют алгоритмом оптимального корреляционного приема. В основе алгоритма оптимального корреляционного приема лежит вычисление функции корреляции между принятым сигналом и эталонными образцами передаваемого сигнала.

Для реализации алгоритма оптимального корреляционного приема необходимо знание границ посылок сигнала *xi*. Поэтому часто используется метод приема, менее чувствительный к информации о фазе принимаемых посылок – метод автокорреляционного приема [7]. Алгоритм метода автокорреляционного приема представляется в виде:

 *xj(t-T)*, если 

*xi* = (5.5)

**, если

где *xj(t-T)* – значение принятой посылки в предыдущий момент времени, ** – инверсное значение принятой ранее посылки, *T* – длительность посылки.

В основе автокорреляционного метода приема лежит вычисление функции автокорреляции принятого сигнала. При этом требуется знание отсчетов сигнала в предыдущий момент времени и величина периода, в течение которого определяется функция автокорреляции.

Аналоговый сигнал в канале передается с помощью отрезка синусоиды

*s(t) = a × sin(ω t+ϕ),* (5.6)

которая описывается тремя параметрами: амплитуда *a*, частота *ω* и фаза *ϕ*.

Модуляцией называется изменение одного или нескольких параметров сигнала в соответствии с законом изменения информации. Соответственно, модуляция по названию изменяемого параметра может быть амплитудной, частотной или фазовой. Известны методы модуляции, при которых изменяются одновременно несколько параметров сигнала, например, квадратурная амплитудная модуляция (КАМ, QAM – Quadrature Amplitude Modulation).

Если каждый сигнал переносит один бит информации, то модуляция будет двоичной или двухпозиционной, в противном случае каждый сигнал переносит несколько бит информации и модуляция будет недвоичной, многопозиционной.

В современных модемах, как правило, используются три вида модуляции: частотная, относительная фазовая (фазоразностная) и квадратурная амплитудная модуляция[8].

Выбор вида модуляции определяется помехоустойчивостью, которую обеспечивает этот вид модуляции. Помехоустойчивость зависит от минимального Евклидова расстояния между сигналами в метрическом функциональном пространстве сигналов и характера помех, действующих в канале. Возможности разнесения сигналов в функциональном пространстве ограничены мощностью сигнала и полосой пропускания канала связи. Наилучшей помехоустойчивостью обладает фазовая модуляция, затем идет частотная модуляция и на последнем месте - амплитудная модуляция.

При частотной модуляции (ЧМ) значениям “0” и “1” информационной последовательности соответствуют определенные частоты непрерывного сигнала. “1” – соответствует сигнал с нижней частотой *ω1* , а “0” – сигнал с верхней частотой *ω2*. Частотная модуляция более помехоустойчива в телефонных каналах, поскольку помехи телефонного канала искажают в основном амплитуду, а не частоту сигнала. При передаче информации, например, между мобильными средствами связи, помехоустойчивость частотной модуляции снижается за счет доплеровского смещения частот. Частотная модуляция требует значительную полосу частот, поэтому этот вид модуляции в основном применяется в низкоскоростных модемах (до 2400 бит/с) для связи в каналах с низким уровнем помех.

При относительной фазовой модуляции (ОФМ) в зависимости от значения информационной посылки изменяется только фаза сигнала. При этой модуляции исходная информационная последовательность вначале преобразуется в информационную последовательность относительной модуляции. Например, для двоичной модуляции символы относительной модуляции вычисляются по рекуррентному соотношению

 *, i = 0,1,…* (5.7)

где *v0,v1,…,vi,…* – исходная информационная последовательность, *u0,u1,…ui,…* - последовательность символов относительной модуляции, начальное условие обычно полагают *u-1 = 0.*

Тогда на приемной стороне, исходная информация восстанавливается по формуле

 (*u-1 = 0*). (5.8)

Таким образом, каждый информационный символ определяется изменением текущего значения принятого символа относительной модуляции относительно его предыдущего значения.

Модулятор каждому символу относительной модуляции сопоставляет гармонический сигнал соответствующей фазы.

Для двухфазной ОФМ, или однократной ОФМ фаза изменяется на 1800 и может принимать, например, значения 00 и 1800.

Чаще применяется четырехфазная ОФМ, или двукратная ОФМ (ДОФМ), основанная на передаче четырех сигналов, каждый из которых несет информацию о двух битах последовательности относительной модуляции. Обычно используются два набора фаз в зависимости от значения двух бит 00, 01, 10, 11: 00, 900, 1800, 2700 или 450,1350, 2250, 3150.

При количестве фаз более восьми (число кодируемых символов более трех), помехоустойчивость ОФМ резко снижается. По этой причине для высокоскоростной передачи данных ОФМ не используется (до 4800 бит/с).

При квадратурной амплитудной модуляции изменяются как фаза, так и амплитуда сигнала, что позволяет увеличить количество кодируемых бит и существенно повысить помехоустойчивость [9]. Это объясняется более равномерным расположением сигналов в функциональном пространстве сигналов и соответственно большим минимальным Евклидовым расстоянием между сигналами. В настоящее время используются способы модуляции, в которых число кодируемых информационных бит может достигать 8...9, а число позиций сигнала в сигнальном пространстве – 256…512.

КАМ позволяет повысить скорость передачи информации по каналам связи. Так, скорость передачи данных по коммутируемым телефонным каналам может достигать величины 28800 бит/с. Используется не идеальность характеристик каналообразующей аппаратуры, при которых полоса пропускания реального канала больше полосы пропускания стандартного телефонного канала - 3100 Гц. Кроме того, используются сигнально – кодовые конструкции (СКК), повышающие помехоустойчивость передачи. СКК представляет собой результат кодирования исходной информации помехоустойчивым сверточным (решетчатым) кодом и отображения полученной двоичной кодовой последовательности с помощью КАМ в пространство сигналов.

Использование СКК увеличивает минимальное Евклидово расстояние в сигнальном пространстве.

Рассмотрим более подробно алгоритм однократной ОФМ. При однократной ОФМ сигнальное пространство состоит из двух сигналов:

*x0 = a·sin(ωt), x1 = - a·sin(ωt)* (5.9)

и алгоритм автокорреляционного приема сигналов двоичной ОФМ запишется

0, если 

*xi* = (5.10)

1, если 

Структурная схема автокорреляционного приемника сигналов однократной ОФМ изображена на рис.5.1

Из канала аналоговый сигнал поступает на вход полосового фильтра. Ширина полосы пропускания полосового фильтра выбирается, исходя из разумного компромисса между точным воспроизведением входного полезного сигнала и минимизацией уровня входных помех.

Усили-тель

Полосо-вой фильтр

Линия задер-

жки

Умно-

житель

Вход

Порого-вый элемент

Интегра-

тор

Выход

Рис. 5.1

С выхода полосового фильтра сигнал поступает на вход усилителя, который должен иметь достаточно высокую чувствительность для приема слабых сигналов.

Далее сигнал подается на вход умножителя, на второй вход которого поступает сигнал, задержанный на время длительности одной посылки в линии задержки.

Автокорреляционная функция вычисляется с помощью интегратора, выход которого соединен с входом порогового элемента.

Пороговый элемент преобразует аналоговый сигнал с выхода интегратора в дискретный сигнал двоичных посылок.

Алгоритм работы модема однократной ОФМ, включающий в себя процедуры модуляции и демодуляции приведен ниже.

Алгоритм работы модема однократной ОФМ

нет да нет нет да

v0:=r0;b:=0;d1:=$8000

i=15

i:=1

i:=i+1

r0[I]:=sin(i\*pi/8);r1[i]:=-r0[i]

Вывод слова относительной

модуляции w

i:=i+1

i=16

d0:=d1 shr 1;d2:= d2 shr 1

w:=w+d1

d1:=(a and d2) xor d0

i:=1

w:=0;d2:=$8000;d0:=0

Ввод слова исходной

информации а

Начало

i:=1

d0:=w and d1

нет да

v1=r0

v1:=r1

d0=0

Вычисление автокорелляционной

функции s:=0;b1:=0

j:=0;

s:=s+v0[j]\*v1[j]

да

b1:=b1+1

s<0

нет

да

н нет

j:=j+1

j=15

да

нет

b:=b+d1

b1>8

да

нет

d1:=d1 shr 1;v0:=v1

i:=i+1

i=16

нет

да

Вывод принятого слова

b

var

i,j:integer;

Конец

a,d0,d1,d2,w,

b,b1:word;

s:real;

r0,r1,v0,v1:array[0..15] of real;

Рис. 5.2.

Входом программы модема однократной ОФМ является 16-ти битное слово информации *a*,которое преобразуется в слово относительной модуляции *w*. Каждому биту слова *w* сопоставляются отрезки синусоиды, фаза которой определяется значением этого бита. Выходом программы является полученное в результате демодуляции слово *b*, которое должно повторять введенное слово информации.

program modem1;

{Программа работы модема однократной ОФМ}

uses wincrt;

var

i,j:integer;

a,d0,d1,d2,w,b,b1:word;

s:real;

r0,r1,v0,v1:array[0..15] of real;

begin

write('Введите слово входной информации ');readln(a);

d2:=$8000;w:=0;d0:=0;

for i:=1 to 16 do begin

d1:=(a and d2) xor d0;w:=w+d1;d0:=d1 shr 1;d2:=d2 shr 1;

end;

write('Слово кода относительной модуляции=');writeln(w);

for i:=0 to 15 do begin r0[i]:=sin(i\*pi/8);r1[i]:=-r0[i];end;

v0:=r0;b:=0;d1:=$8000;

{write('Автокорреляционная функция=');}

for i:=1 to 16 do begin

d0:=w and d1;

if d0=0 then v1:=r0 else v1:=r1;

s:=0;b1:=0;

for j:=0 to 15 do begin s:=s+v0[j]\*v1[j];if s<0 then inc(b1);end;

if b1>8 then b:=b+d1;

d1:=d1 shr 1;v0:=v1;

end;

writeln;writeln('Принятое слово информации=',b);

writeln;readln;

end.

Пример

В таблице в десятичной форме записи приведены различные варианты входных слов *a* и соответствующих им слов относительной модуляции *w*

|  |  |  |  |
| --- | --- | --- | --- |
| *a* | 21845 | 65535 | 55065 |
| *w* | 26214 | 43690 | 39441 |

**4. Частотная модуляция. Демодулятор сигналов ЧМ**

Частотная модуляция (ЧМ) — вид модуляции, при котором информационный сигнал управляет частотой несущего колебания. По сравнению с амплитудной модуляцией здесь амплитуда остаётся постоянной. Частотная модуляция была предложена американцем Эдвином Армстронгом и запатентована им 26 декабря 1933 года.

Частотная модуляция применяется для высококачественной передачи [звукового](http://ru.wikipedia.org/wiki/%D0%97%D0%B2%D1%83%D0%BA) ([низкочастотного](http://ru.wikipedia.org/w/index.php?title=%D0%9D%D0%A7&action=edit&redlink=1)) сигнала в радиовещании (в диапазоне [УКВ](http://ru.wikipedia.org/wiki/%D0%A3%D0%BB%D1%8C%D1%82%D1%80%D0%B0%D0%BA%D0%BE%D1%80%D0%BE%D1%82%D0%BA%D0%B8%D0%B5_%D0%B2%D0%BE%D0%BB%D0%BD%D1%8B)), для звукового сопровождения телевизионных программ, передачи сигналов цветности в телевизионном стандарте [SECAM](http://ru.wikipedia.org/wiki/SECAM), видеозаписи на магнитную ленту, в музыкальных [синтезаторах](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B5%D0%B7%D0%B0%D1%82%D0%BE%D1%80). Высокое качество кодирования аудиосигнала обусловлено тем, что при ЧМ применяется большая (по сравнению с шириной [спектра](http://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D0%BA%D1%82%D1%80) сигнала АМ) [девиация](http://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%B2%D0%B8%D0%B0%D1%86%D0%B8%D1%8F_%D1%87%D0%B0%D1%81%D1%82%D0%BE%D1%82%D1%8B) несущего сигнала, а в приёмной аппаратуре используют ограничитель амплитуды радиосигнала для устранения импульсных помех.

Простейшим видом модуляции с постоянной амплитудой является двоичная частотная модуляция FSK. При этом типе модуляции полезный сигнал формируется из отрезков двух синусоид.

 (6.1)



где сигнал s1 используется для передачи логического нуля, а сигнал s2 — для передачи логической единицы.

Структурная схема модулятора, реализующая такой вид частотной модуляции, приведена на рисунке 6.1.

Так как начальная фаза генераторов никак не связана друг с другом, то этот вид модуляции получил название частотной модуляции с разрывной фазой. Кроме того, в этом виде модуляции начальные фазы частот *w1* и *w2* некогерентны по отношению к модулирующему сигналу, поэтому такой вид модуляции часто называют некогерентной ЧМ. Однако, справедливости ради, следует отметить, что некогерентной может быть и частотная модуляция с непрерывной фазой.

В качестве примера использования некогерентной частотной модуляции можно привести стандарт CCITT V.21 (скорость передачи данных 300 Бод). В стандарте CCITT V.21 используются частоты f1=1080 Гц и f2=1750Гц. Некогерентная модуляция применяется и для других низкоскоростных систем передачи данных, где полоса пропускания канала не является проблемой.

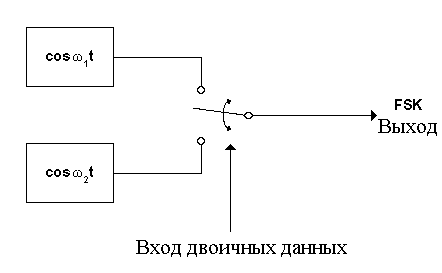


Рис. 6.1 – Схема частотного модулятора с разрывной фазой.

При демодуляции вычисляют усредненную по нескольким отсчетам вторую производную входного сигнала

. (6.2)

Квадрат частоты оценивают в виде

 (6.3)

Отсюда имеем оценку частоты сигнала и можем определить передаваемую информацию

*0*, если 

*I* = (6.4) 1, если, иначе

Алгоритм работы модулятора

Начало

нет да да нет

d1:= d1 shr1

v[i,j]:=r1[j]

v[i,j]:=r0[j]

a and d1=0

d1:=$8000

i:=15

i:=i+1

i=15

r1[i]:= sin(0.02\*i\*π/8)

r0[i]:=sin(0.01\*i\*π/8)

i:=0

Ввод слова исходной

информации а

нет

i:=i-1

i=0

Конец

да

Алгоритм работы демодулятора

i := 15

d1:=$8000; b:=0

s:=0

j:=1

r:=(v[i,j]-2\*v[i,j+1]+v[i,j+2])/v[i,j]

s:=s+r

j:=j+1

j=13

s:=s/13

b:=b+d1

s<-5e-5

d1:=d1 shr 1

i=16

var

Конец

i,j:integer;

a,,d1,b,b1:word;

s:real;

r0,r1,v0,v1:array[0..15] of real;

Рис. 6.2.

Входом программы модема ЧМ является 16-ти битное слово информации *a*.. Каждому биту слова *a* сопоставляются отрезки синусоиды, частота которой определяется значением этого бита. Выходом программы является полученное в результате демодуляции слово *b*, которое должно повторять введенное слово информации.

program modem2;

{Модем ЧМ}

uses crt;

var

i,j:integer;

a,d1,d2,b:word;

r,s,t,dt:real;

r0,r1,v0,v1:array[0..15] of real;

v:array[0..15,0..15] of real;

begin

dt:=0.05;j:=1;s:=0;for i:=0 to 31 do begin t:=i\*dt;s:=s+sin(j\*t);end;

for i:=32 to 61 do begin

t:=i\*dt;s:=s-sin(j\*(t-32\*dt))+sin(j\*(t));r:=s/sin(j\*t+pi/2);

write(r,' '); end;writeln;readln;

write('Введите слово входной информации ');readln(a);

{write('Отсчеты нуля и единицы=');}

for i:=0 to 15 do begin r0[i]:=sin((0.01)\*i\*pi/8);

r1[i]:=sin((0.02)\*i\*pi/8);end;

v0:=r0;b:=0;d1:=$8000;

{write('Вычисление массива из отрезков синусоид,

соответствующих входной информации');}

for i:=15 downto 0 do begin

if a and d1=0 then for j:=0 to 15 do v[i,j]:=r0[j]

else for j:=0 to 15 do v[i,j]:=r1[j];d1:=d1 shr 1;

end;

{Демодулятор}

b:=0;d1:=$8000;

for i:=15 downto 0 do begin s:=0;

for j:=1 to 13 do begin

r:=(v[i,j]-2\*v[i,j+1]+v[i,j+2])/v[i,j];

s:=s+r;

end;s:=s/13;

if s<-5e-5 then

b:=b+d1;

d1:=d1 shr 1; end;

writeln;writeln('Выходная информация=',b);

writeln;readln;

end.

Пример

В таблице в десятичной форме записи приведены различные варианты входных слов *a* и равные им слова выходной информации *b*.

|  |  |  |  |
| --- | --- | --- | --- |
| *a* | 21845 | 65535 | 55065 |
| *b* | 21845 | 65535 | 55065 |

**5. Тактовая синхронизация.**

**Векторное коррекционное устройство (КУ).**

С выхода демодулятора сигнал поступает на вход устройства тактовой синхронизации.

На выходе демодулятора сигнал дискретизирован по уровню: уровень логического “0” либо “1”. Вследствии воздействия помех, ослабления сигнала, а также других причин принятые посылки могут быть искажены. Искажения посылок делятся на два класса: краевые искажения и дробления. Краевыми искажениями называются сдвиги границ принятых посылок относительно границ неискаженных посылок. Также посылки могут иметь кратковременные изменения уровня - дробления.

Различают несколько видов краевых искажений: преобладания, случайные и характеристические [10]. Преобладания выражаются в том, что посылки одного знака удлиняются, а другого - укорачиваются. Случайные краевые искажения обусловлены действием в канале помех и носят случайный характер. Характеристические искажения определяются переходными процессами при смене полярности посылок и зависят от передаваемой информации.

Краевые искажения типа преобладаний и характеристические путем выбора соответствующего алгоритма работы демодулятора и регулировки параметров входящих в демодулятор узлов, во многих случаях могут быть сведены к минимуму.

Дробления, как правило, носят случайный характер и присутствуют даже при идеальной регулировке демодулятора.

Восстановление и регенерацию посылок осуществляет соответствующее устройство, включающее в себя приемник элементов кода и устройство тактовой синхронизации или коррекционное устройство (КУ). Сигнал с выхода порогового элемента демодулятора преобразуется в цифровую форму. Для этого он квантуется по времени. Время длительности *T* одной посылки делится на *m* временных интервалов длительности *τ*. В пределах каждого интервала принимается решение о его полярности. Положение интервала в пределах посылки определяется его фазой, значение которой может изменяться от 0 до 2π.

При смене полярности, посылки отделяются друг от друга границами или фронтами. Фронтом кодовой посылки является переход из “0” в “1” или наоборот. Фазы принятых фронтов составляют исходную информацию для определения границ неискаженных кодовых посылок. Фазы принятых фронтов также как и фаза неискаженных посылок отсчитываются от некоторого опорного колебания, условно принятого за нулевую точку отсчета.

Краевые искажения и дробления приводят к появлению ложных фронтов.

Перед устройством регенерации и восстановления посылок часто устанавливается дополнительный цифровой интегрирующий фильтр или фильтр Мартынова. На вход цифрового фильтра Мартынова поступают отсчеты сигнала *ai i = 1,2,… , ai = 0,1* с частотой *m* значений в течение длительности одной посылки. Обычно, для достаточно точного восстановления посылок, частота отсчетов *m* не превышает величины 16…24 отсчета в течение посылки.

Работа фильтра Мартынова задается формулой

 , *i = m,m+1,…* ,

где *ci* – выходные отсчеты фильтра Мартынова, *INT(a)* – целая часть числа *a*.

Фильтр Мартынова вычисляет вес *w* последовательности *m* подряд идущих отсчетов *ai*. Если вес *w > INT(m/2),* то на выходе фильтра будет “1”, в противном случае – “0”.

Структурная схема фильтра Мартынова изображена на рис. 7.1.

Вход *ai*

1 … m

Порого-вый

элемент

Выход *ci*

…

∑

Рис. 7.1

Фильтр Мартынова представляет собой *m* – разрядный регистр сдвига, выходы разрядов которого соединены со входами арифметического сумматора. Выход сумматора соединен со входом порогового элемента, который оценивает величину суммы. При достижении величины суммы определенного значения срабатывает пороговый элемент и на его выходе появляется сигнал лог. “1”. При уменьшении суммы - на выходе порогового элемента будет лог. “0”. Выход порогового элемента является выходом фильтра Мартынова.

Фильтр Мартынова преобразует дробления посылок входного сигнала в искажения краев посылок, создавая более благоприятные условия для работы устройства тактовой синхронизации и приема посылок.

Устройство тактовой синхронизации (КУ) определяет границы неискаженных кодовых посылок. Известны различные типы КУ:

1. Резонансное КУ

1. КУ с авторегулированием
2. Векторное КУ
3. Гистограммное КУ
4. Корреляционное КУ

Основным узлом резонансного КУ [10] является узкополосный фильтр. В резонансном КУ каждому фронту принятых посылок сопоставляется некоторый сигнал. Главной гармоникой этого сигнала является колебание, частота которого равна частоте поступления посылок. Например, этим сигналом может быть один или несколько прямоугольных импульсов соответствующей частоты. Сумма сформированных таким образом сигналов поступает на вход узкополосного фильтра, который выделяет основную гармонику. Фаза выделенной гармоники является оценкой фазы неискаженных посылок.

С уменьшением полосы пропускания узкополосного фильтра повышается точность фазирования, но одновременно повышаются требования к стабильности тактовых генераторов на передающей и приемной стороне. Недостатком этого устройства является потеря синфазности при кратковременных перерывах связи. Вместе с тем резонансные фазирующие устройства сравнительно просты в схемной реализации.

КУ с авторегулированием построено по схеме системы автоматического регулирования с отрицательной обратной связью [11, 12]. При этом датчиком сигнала управления в цепи обратной связи является фазовый дискриминатор, определяющий расхождение между текущими фазами принятых посылок и оценкой фазы неискаженных посылок на приемной стороне. Структурная схема КУ с авторегулированием представлена на рис.7.2 [13].

ФД

ИФП

Вход Выход

СФ

УУ

ДЧ

ЭУ

БОЧ

Рис. 7.2

Сигналы с выхода демодулятора поступают на измеритель фазы посылок ИФП, который в простейшем случае формирует короткие импульсы из фронтов входных посылок. Выходной сигнал ИФП подается на один из входов фазового дискриминатора ФД, на второй вход которого поступают синхроимпульсы. С выхода ФД сигналы расхождения фазы между синхроимпульсами и принимаемыми посылками попадают на усредняющее устройство УУ (как правило, реверсивный счетчик). УУ выполняет усреднение расхождения фазы по нескольким посылкам и лишь затем выдает сигнал на элемент управления ЭУ, который чаще всего представляет собой схему добавления и исключения импульсов. Через ЭУ последовательность импульсов с блока опорных частот БОЧ поступает на делитель частоты ДЧ. К выходу ДЧ подключен селектор фронтов СФ, формирующий синхроимпульсы. При рассогласовании фазы между синхроимпульсами и принимаемыми посылками усредненный по нескольким посылкам сигнал расхождения воздействует на ЭУ таким образом, чтобы ЭУ, добавляя или исключая импульсы в последовательность, поступающей на вход ДЧ, подстраивал фазу синхроимпульсов под фазу посылок.

Для описания и анализа работы КУ этого типа, определения его характеристик используется теория систем автоматического управления и контроля. Недостатком КУ этого типа является прямая зависимость между временем вхождения в синхронизм и временем поддержания синхронизации при перерывах связи. При сокращении времени вхождения в синхронизм, время поддержания синхронизма также уменьшается, что ухудшает качество работы КУ. Частично указанный недостаток может быть ослаблен, например введением переменного шага коррекции и специальной системы отключения фазирующего устройства в периоды замираний.

Резонансное КУ и устройство фазирования с авторегулированием реализуются, как правило, аппаратным путем. В процессе развития методов цифровой обработки информации на ЭВМ были разработаны алгоритмы тактовой синхронизации, в основе которых лежат программные способы реализации устройств. Определение фазы неискаженных посылок может рассматриваться как задача статистической обработки данных и нахождения статистических оценок параметров. В нашем случае обрабатываемой статистикой является множество фаз фронтов принятых посылок. Для определения фазы неискаженных посылок могут использоваться как рекуррентные оценки, так и оценки, полученные обычным способом.

Характерным для фазы фронтов является циклическая структура этой величины. Фаза, равная 0 эквивалентна фазе 2π. Поэтому средняя величина фазы (математическое ожидание) не может служить оценкой фазы неискаженных фронтов. Например, при поступлении фаз фронтов, равных π/8 и 15π/8, средняя фаза фронтов будет (π/8+15π/8)/2 = π. Однако по смыслу ясно, что фазы фронтов расположены вблизи 0, с разных сторон от 0 и равноудалены от него и оценкой фазы неискаженных фронтов в данном случае может служить величина фазы, равная 0, а не π.

Представление о фазах фронтов как о векторах на фазовой плоскости лежат в основе алгоритма работы векторного КУ.

В векторном КУ каждому фронту посылок (переходу отсчетов из “0” в “1” и обратно) сопоставляется единичный вектор на декартовой плоскости с углом поворота, равным фазе рассматриваемого фронта.

На вход КУ поступают отсчеты сигнала *ci i = 0,1,2,… , ci = 0,1* с частотой *m* значений в течение длительности одной посылки. Пусть *сi+1 = ci ⊕ 1*, т.е. при *i+1* отсчете произошло изменение уровня сигнала *.* Тогда фаза фронта запишется в виде



и прямоугольные координаты единичного вектора будут равны

*xi = cos(αi), yi = sin(αi)*

Прямоугольные координаты суммарного вектора, угол поворота которого является оценкой фазы неискаженных посылок вычисляется по формулам

и фаза суммарного вектора будет равна

.

Векторная диаграмма, поясняющая алгоритм работы векторного КУ приведена на рис. 7.3.

*Y*

*y*

*ϕ*

*yi*

*xi* *x* *X*

Рис. 7.3.

После вычисления фазы неискаженных посылок приемник элементов кода восстанавливает принятые посылки. Восстановление посылок может осуществляться путем опроса значения посылки в ее середине (метод укороченного контакта), путем оценки мягкого веса посылки (метод интегрального приема), либо возможен промежуточный вариант определения значения посылки по нескольким отсчетам в середине посылки. Все эти варианты восстановления посылок являются частными случаями метода интегрального приема с весовой функцией.

Ниже приведен алгоритм, моделирующий работу векторного КУ в канале связи, дробления в котором возникают независимо друг от друга с некоторой вероятностью *p*.

Алгоритм работы векторного КУ

Начало

Введите частоту дроблений p

Задайте длину массива посылок n

Randomize

a1[0]:=a1[1]:=0;a1[2]:=a1[3]:=$ffff

i:=0

d:=$8000;a[i]:=a1[i mod 4]

j:=1

нет

random<p

да

a[i]:=a[i] xor d

d:=d shr 1

нет

j=16

j:=j+1

да

i=n+1

i:=i+1

нет

да

s:=0;c:=0

i:=0

j=1

(d1 and $8000) <>0

да

нет

s:=s+sin(j\*pi/8);c:=c+cos(j\*pi/8)

j=15

j:=j+1

нет

да

d1:=d1 shl 1

b:=((a[i+1] shr 15) xor a[i]) and 1

b<>0

нет

да

c:=c+1

i:=i+1

i=n

нет

да

да нет

c=0

r:=arctan(s/c)

r:=pi

r<0

var да

r:=2\*pi+r

n,i,j,f:integer;

b:byte;

d,d1:word; нет

p,s,c,r:real;

a1:array[0..3] of word;

Вывод фазы неискаженных

посылок f:=trunc(8\*f/pi)

a:array[0..1025] of word;

Конец

Рис. 7.4.

На входе программы задаются частота дроблений и количество посылок. В результате работы программы вычисляется фаза неискаженных посылок, значение которой выводится на экран монитора.

program KU1;

{Программа моделирования работы векторного КУ}

uses crt;

var

n,i,j,f:integer;

b:byte;

d,d1:word;

p,s,c,r:real;

a1:array[0..3] of word;

a:array[0..1025] of word;

begin

write('Задайте среднюю частоту дроблений ');readln(p);

write('Задайте количество моделируемых посылок <1024 '); readln(n);

{Генерация потока входных посылок}

randomize;

a1[0]:=0;a1[1]:=0;a1[2]:=$ffff;a1[3]:=$ffff;

for i:=0 to n+1 do begin

d:=$8000;a[i]:=a1[i mod 4];

for j:=1 to 16 do begin

if random<p then a[i]:=a[i] xor d;

d:=d shr 1; end;

end;

{Вычисление результирующего вектора фаз посылок}

c:=0;s:=0;

for i:=0 to n do begin

d1:=(a[i] xor a[i] shl 1) and $fffe;

for j:=1 to 15 do begin

if (d1 and $8000)<>0 then begin s:=s+sin(j\*pi/8);c:=c+cos(j\*pi/8);end;

d1:=d1 shl 1;

end;

b:=((a[i+1] shr 15) xor a[i]) and 1;

if b<>0 then c:=c+1;

end;

if c=0 then r:=pi else r:=arctan(s/c);if r<0 then r:=2\*pi+r;

f:=trunc(8\*r/pi);

writeln('Фаза неискаженных посылок=',f);

readln;

end.

Пример

В таблице приведены значения фазы неискаженных посылок в 1/16 долях полного круга в зависимости от частоты дробления посылок.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Частота дроблений | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Фаза неискаженных посылок | 0 | 15 | 15 | 14 | 12 | 1 |

**6. Цикловая синхронизация. Стартстопная и кодовая цикловая синхронизация.**

С выхода устройства регенерации и восстановления посылок поступает последовательность двоичных символов, среди которых могут быть слова помехоустойчивого кода либо сообщения, не защищенные кодом. Устройство цикловой синхронизации или фазирования (УЦФ) предназначено для определения начала кодового слова или сообщения.

Цикловая синхронизация используется как в системах синхронной связи для установки и поддержания постоянного фазового соотношения между сообщениями, так и в асинхронных системах сеансной связи для поиска и выделения отдельных сообщений.

Сначала УЦФ осуществляет преобразование входной последовательности *V* в последовательность символов *U*, с помощью некоторого оператора *g: U = g(V)* [14]. В преобразованной последовательности *U* ищется некоторое множество последовательностей символов *j1,j2,… ∈ J*, называемых также синхропосылками. Местоположение синхропосылок определяет начало кодового слова или сообщения.

Простейшим методом цикловой синхронизации является стартстопная синхронизация. При стартстопной синхронизации оператор g является тождественным оператором. Синхропоследовательности расположены в начале или(и) в конце кодового слова. Элемент, стоящий перед сообщением называется стартовым, а в конце – стоповым.

Например, в телеграфии при передаче пятибитных символов для цикловой синхронизации в начале каждого символа добавляется нулевая посылка (старт), а в конце - одна или полторы единичных посылок (стоп).

Для стартстопной синхронизации кодовых слов обычно перед кодовым словом передается некоторая последовательность символов с “хорошими” синхронизирующими свойствами, например последовательность Баркера или рекуррентная последовательность максимального периода (*M* – последовательность) [15].

Считается, что некоторая последовательность *U* обладает синхронизирующими свойствами, если расстояние по Хеммингу между самой последовательностью и ее правыми (левыми) сдвигами отлично от нуля:

*dH(U,U shr z) > 0*,

где *U shr z* - сдвиг последовательности *U* на *z*  разрядов вправо, при котором *z* старших разрядов *U* замещаются произвольной *z* – разрядной кодовой комбинацией.

Для последовательности с “хорошими” синхронизирующими свойствами даже при достаточно большой величине сдвига z, кодовое расстояниеHHH *dH* будет отлична от 0. Величина *dH* определяет возможность синхронизации при наличии в синхронизирующей последовательности ошибок.

Примером синхронизации при не тождественном операторе g преобразования входной последовательности является кодовая синхронизация [16]. Кодовая синхронизация используется для синхронизации циклических помехоустойчивых кодов, а также каскадных кодов, построенных на их основе. При кодировании этих кодов проверочные символы помехоустойчивого кода суммируются по модулю два с последовательностью, обладающей синхронизирующими свойствами. Входная последовательность УЦФ пропускается через некоторый линейный цифровой фильтр (фильтр Хаффмена), в котором вычисляется синдром кода. В случае неискаженного кодового слова, синдром кода равен нулю. Таким образом, на месте синдрома в нашем случае мы будем иметь переданную синхропоследовательность. Синхронизация осуществляется путем поиска этой синхропоследовательности в последовательности символов, поступающих на вход устройства.

Наиболее эффективно использование кодовой синхронизации для синхронизации каскадных кодов. В каскадном коде передается несколько слов циклического помехоустойчивого кода и вероятность обнаружить среди них одно или несколько кодовых слов, необходимых для синхронизации всего каскадного кода, достаточно высока.

После цикловой синхронизации каскадного кода нетрудно удалить наложенные синхропоследовательности, так что эти синхропоследовательности не влияют на способность каскадного кода обнаруживать и исправлять ошибки.

Кодовая синхронизация каскадного кода, в отличие от стартстопной, не требует введения дополнительной избыточности при передаче и не снижает скорость передачи полезной информации в канале связи. Это является ее важным преимуществом по сравнению со стартстопной синхронизацией.

Рассмотрим более подробно один из вариантов стартстопной синхронизации. Предположим, что требуется передать некоторое сообщение. Для цикловой синхронизации перед сообщением вставляется двоичная синхропоследовательность G = 1110001011000011, длина которой 16 бит.

Расстояние по Хеммингу между синхропоследовательностью и последовательностями, являющимися правыми сдвигами этой синхропоследовательности, не меньше трех до величины сдвига менее восьми. Это позволяет осуществлять надежную синхронизацию при наличии не более двух ошибок в синхропоследовательности. Обнаружение синхропоследовательности в поступающей из канала информации определяет координаты местоположения передаваемого сообщения.

Ниже приведен алгоритм стартстопной цикловой синхронизации с помощью рассмотренной 16-битной последовательности.

Алгоритм стартстопной цикловой синхронизации сообщения

Начало

Ввод сообщения a

c:=$e2c3;randomize

i:=0

a1[i]:=random(256)

i=260

нет

i:=i+1

да

f:=lo(a)+1

a1[f]:=$e2;a1[f+1]:=$c3;a1[f+2]:=hi(a);a1[f+3]:=lo(a)

f:=f and 7

i:=0

a1[i]:=lo((swap(a1[i])+a1[i+1]) shr f)

i:=i+1

i=260

нет

да

i:=0

d[0]:=d[1];d[1]:=d[2];d[2]:=d[3];d[3]:=d[4];d[4]:=a1[i]

e[0]:=d[0];e[1]:=d[1];e[2]:=d[2];e[3]:=d[3];e[4]:=d[4]

i1:=255

j:=0

t:=(e[0] xor $e2) or (e[1] xor $c3)

нет

t=0

да

i1:=j

Вывод местоположения сообщения - i-2 и

сдвига сообщения в пределах байта - j

b:=swap(hi((swap(d[2])+d[3]) shl i1))+hi((swap(d[3])+d[4]) shl i1)

Вывод принятого сообщения b

e[0]:=hi((swap(e[0])+e[1]) shl 1

e[1]:=hi((swap(e[1])+e[2]) shl 1

e[2]:=hi((swap(e[2])+e[3]) shl 1

да

нет

j=7

j:=j+1

var

i,i1,j:integer;

t:byte; да

c,a,b,f:word; нет

i:=i+1

i=260

d,e:array[0..4] of byte;

a1:array[0..511] of byte;

да

Конец

Рис. 9.1

Входом программы является передаваемое сообщение – 16-ти битное слово, которое вместе с синхропосылкой случайным образом помещается в массив информации, объемом 260 байт. УЦФ отыскивает синхропосылку в массиве информации и выделяет передаваемое сообщение. На экран монитора выводятся координаты местоположения синхропосылки и принятое сообщение, которое должно совпадать с переданным.

program UCF1;

{Программа моделирования работы стартстопного УЦФ}

uses crt;

var

i,i1,j:integer;

t:byte;

c,a,b,f:word;

d,e:array[0..4] of byte;

a1:array[0..511] of byte;

begin

write('Введите передаваемое сообщение ? ');readln(a);

{Генерация входной информации УЦФ}

c:=$e2c3;

randomize;

for i:=0 to 260 do a1[i]:=random(256);

f:=lo(a)+1;a1[f]:=hi(c);a1[f+1]:=lo(c);

a1[f+2]:=hi(a);a1[f+3]:=lo(a);

f:=f and 7;

for i:=0 to 260 do a1[i]:=lo((swap(a1[i])+a1[i+1]) shr f);

{УЦФ}

for i:=0 to 260 do begin

for j:=0 to 3 do d[j]:=d[j+1];d[4]:=a1[i];

for j:=0 to 4 do e[j]:=d[j];

i1:=255;

for j:=0 to 7 do begin t:=(e[0] xor $e2) or (e[1] xor $c3);

if t=0 then begin

i1:=j;

writeln('Есть синхронизация');

writeln('Местоположение сообщения в массиве=',i-2);

writeln('Сдвиг сообщения в пределах байта=',i1);

b:= swap(hi((swap(d[2])+d[3]) shl i1))+hi((swap(d[3])+d[4]) shl i1);

writeln('Принятое сообщение ',b);

end;

e[0]:=hi((swap(e[0])+e[1]) shl 1);

e[1]:=hi((swap(e[1])+e[2]) shl 1);

e[2]:=hi((swap(e[2])+e[3]) shl 1);

end; end; readln;

end.

Пример

В таблице приведены передаваемые сообщения в десятичной форме и координаты его местоположения, полученные в результате работы программы.

|  |  |  |  |
| --- | --- | --- | --- |
| Передаваемое сообщение  a | 65535 | 1234 | 15915 |
| Координаты сообщения :  номер байта и сдвиг | 257 0 | 212 3 | 45 4 |

**Литература**

1.Кухарев А.Д. Системный подход к созданию комплексов оконечной аппаратуры систем передачи информации. Техника средств связи, сер. ТПС, вып. 4,1989, стр.3-5.

2. Кухарев А.Д. Анализ инженерных методов проектирования комплексов оконечной аппаратуры систем передачи информации. Техника средств связи, сер. ТПС, вып. 4,1989, стр. 6-10.

3.Блох Э.Л., Попов О.В., Турин В.Я. Модели источника ошибок

в каналах передачи цифровой информации. М.:Связь,1971.

4. Кухарев А.Д., Ледяков Б.П., Яковлев В.Г. Блочная модель канала, основанная на цепи Маркова. Техника средств связи, сер. ТПС, вып. 5,1990, стр. 35-41.

5. Элементы теории передачи дискретной информации, под ред. Л.П. Пуртова. М.:Связь,1972.

6.Харкевич А. А. Борьба с помехами. Физматгиз, 1965.

7.Заездный А. М., Окунев Ю. Б., Рахович Л. М. Фазоразностная модуляция и ее применение для передачи дискретной информации. М: Связь,1967.

8.Лагутенко О. И. Модемы. Справочник пользователя. Оформление А. Лурье.-Спб.: Лань, 1997.

10.Зелигер Н. Б. Основы передачи данных. Учебное пособие для вузов. М., Связь, 1974.

11.Мартынов Е.Н. Синхронизация в системах передачи дискретных сообщений. М., Связь, 1972.

12. Емельянов Г. А., Шварцман В. О. Передача дискретной информации и основы телеграфии. Учебник для вузов. М., 1973.

13.Побережский Е. С. Цифровые радиоприемные устройства. М.: Радио и связь, 1987.

14.Лосев В. В., Бродская Е. Б., Коржик В. И. Поиск и декодирование сложных дискретных сигналов. Под ред. В. И. Коржика. М.: Радио и связь, 1988.

15.Стиффлер Дж. Дж. Теория синхронной связи. Пер. с англ. Б. С. Цыбакова под ред. Э. М. Габидулина. М.: Связь, 1975.

16.Бек Г. В., Богданович В. Н., Киреев О.П. Метод синхронизации сообщений. Сб.: Построение и анализ систем передачи информации. М.: Наука, 1980.

17.Питерсон У., Уэлдон Э. Коды, исправляющие ошибки: Пер. с англ. /Под ред. Р.П. Добрушина и С.И. Самойленко. М.: Мир,1976.

18.Кларк Дж., мл., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи: Пер. с англ. М.: Радио и связь,1987.

19.Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов, исправляющих

ошибки. М.: Связь,1979.

20.Колесник В.Д., Мирончиков Е.Т. Декодирование циклических кодов. М.: Связь,1968.

21.Форни Г.Д. Каскадные коды. М.: Мир,1970.

22.Касами Т., Токура Н., Ивадари Е., Инагаки Я. Теория кодирования. М.: ир,1978.

23.Блейхут Р. Теория и практика кодов, контролирующих ошибки: Пер. с англ. М.: Мир,1986.

24.Берлекэмп Э. Алгебраическая теория кодирования. М.: Мир, 1971.

25.Квашенников В.В., Яковлев В.Г. Замечание о решении квадратных уравнений над полями Галуа. Пробл. передачи информации. 1987, т.36, #2, стр.108-111.

26.Бек Г.В. Ускоренный алгоритм декодирования кодов Рида-Соломона с исправлением стираний. Вопросы радиоэлектроники, сер.ТПС, 1975, вып.11.

27.Шабанов В.К. К оценке достижимого быстродействия декодирующих устройств кодов РС с исправлением стираний. Техника средств связи, сер.ТПС, 1983, вып.7.