МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ 'ГЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМ. Н.Э. БАУМАНА

Р.В. КОМЯГИН, А.И. СЕНИН

ИССЛЕДОВАНИЕ ПОМЕХОУСТОЙЧИВОСТИ ЦИКЛИЧЕСКИХ КОДОВ

Методические указания к выполнению лабораторной работы по курсу
"Основы теории и техники радиосистем передачи информации"

Москва, 2017

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы – теоретическое и экспериментальное исследование помехоустойчивости циклических кодов.

Задачи работы – ознакомление с теоретическим материалом по данным методическим указаниям, выполнение работы в указанном порядке.

Назначение лабораторной работы – углубление теоретических знаний, практических умений и навыков в результате проведения экспериментальных исследований помехоустойчивости циклических кодов.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Общие сведения

Кодированием и *декодированием* в широком смысле называют любое преобразование сообщения в сигнал и, обратно, сигнала в сообщение путем установления взаимного соответствия. Преобразование следует считать *оптимальным*, если в итоге производительность источника и пропускная способность непрерывного канала окажутся равными, т. е. полностью используются возможности канала. К сожалению, такая постановка задачи не дает ответа на вопрос, что для этого надо делать. Поэтому это преобразование разбивают на два этапа, а именно на этап модуляции-демодуляции, который позволяет перейти от непрерывного канала (радиоканала) к дискретному, и этап кодирования-декодирования в узком смысле, где все операции совершаются над последовательностью символов. Однако и само кодирование-декодирование имеет два противоположных по своим функциям этапа: устранение избыточности в получаемом от источника сигнале (экономное кодирование) и внесение избыточности в передаваемый по каналу цифровой сигнал (помехоустойчивое или избыточное кодирование) для повышения достоверности передаваемой информации.

Экономное кодирование направлено на то, чтобы передаваемый дискретный сигнал имел максимальную энтропию (максимальное количество информации на символ). Тогда для его передачи по радиоканалу с выбранным модемом потребуется минимальная полоса частот. Не рассматривая методы экономного кодирования, укажем только основные свойства дискретного сигнала, в котором полностью устранена избыточность, — это равная вероятность и независимость их появления в последовательности. В этом случае среднее количество информации на символ равно $\log_2 m$, где m – число различных символов.

При помехоустойчивом кодировании в поток передаваемых символов вводятся дополнительные (избыточные) символы для исправления возникающих на приемной стороне ошибок. Это требует увеличения скорости передачи по каналу, что при выбранном типе модема эквивалентно расширению полосы частот сигнала и уменьшению энергии посылки. Поэтому может возникнуть правомерный вопрос о целесообразности использования избыточного кодирования. На этот вопрос дает ответ теорема Шеннона о пропускной способности непрерывного канала связи, из которой следует, что пропускная способность непрерывного канала увеличивается с расширением его полосы, но при оптимальном в широком смысле кодировании. Поэтому следует ожидать повышения достоверности передачи при заданной скорости и отношении сигнал-шум в канале при внесении избыточности. Однако ответа на вопрос, каким должен быть оптимальный кодек, нет, да и, наверное, для сообщения, не фиксированного по длительности, не может быть. Тем не менее, избыточное кодирование широко используется для повышения верности передачи особенно в последние десятилетия, когда проблема создания сложных вычислительных устройств в малых габаритах практически решена.

Рассмотрим принципы кодирования на примере двоичного канала. Допустим, что источник обладает максимальной производительностью. Тогда обязательным условием внесения избыточности является увеличение числа переданных посылок за единицу времени по сравнению с их числом, поступающим от источника. Как вносятся избыточные символы, определяет тип кода. Наиболее просто предположить, что группе из k символов источника ставится в соответствие n символов, передаваемых по каналу. Такой код называется *блочным* и записывается условно как (n, k)-код. Возможны непрерывные коды, которые характеризуются тем, что операции кодирования и декодирования производятся над непрерывной последовательностью символов без разбиения ее на блоки.

Рассмотрим принципы помехоустойчивого кодирования на примере блочного двоичного кода как наиболее простого. Если к символам источника добавляются избыточные символы, то код называют *систематическим*. Если группе информационных символов ставится в соответствие новая группа символов, передаваемая по каналу, в которой информационных символов в явном виде нет, то код называется *неразделимым*. Теперь ответим на основной вопрос, как строить кодек, чтобы при фиксированной избыточности $\chi = r/n$ (где r — число проверочных символов для разделимого кода, n – число символов кодовой комбинации) достоверность передачи была бы максимальной,. При передаче безызбыточным примитивным кодом с числом разрядов k в каждом слове все $N_р = 2^k$ комбинаций являются разрешенными, и ошибка хотя бы в одном символе приводит к тому, что одна разрешенная комбинация переходит в другую. При этом происходит ошибка в приеме сообщения.

Введение избыточных символов приводит к тому, что полное число комбинаций увеличивается и становится равным $N = 2^n$, причем часть из них $N - N_р$ являются запрещенными и могут возникать только тогда, когда в канале происходят ошибки. Этот факт положен в основу обнаружения и исправления ошибок.

Введем понятие кодового расстояния. Предварительно отметим, что для оценки отличия одной кодовой комбинации от другой можно использовать *расстояние Хэмминга* $d(N_i, N_j)$, определяемое числом разрядов, в которых одна кодовая комбинация отличается от другой. Для двоичного кода

$$d\left(N_i, N_j\right) = \sum_{k=1}^{n} b_{ik} \oplus b_{jk},$$

где b_{ik} и b_{jk} — k-е символы кодовых комбинаций N_i и N_j соответственно; $\oplus$ — символ суммирования по модулю 2. Наименьшее расстояние Хэмминга для данного кода называется *кодовым расстоянием.* В дальнейшем его будем обозначать через d.

При независимых ошибках в канале корректирующую способность кода удается выразить через кодовое расстояние. Пусть имеется код с $d = 1$. Учитывая, что искажение одного символа изменяет расстояние Хэмминга на одну единицу, при применении кода с $d = 1$ обнаруживаются не все одиночные ошибки. Для того чтобы код мог обнаруживать любую одиночную ошибку, необходимо обеспечить кодовое расстояние, равное двум. Рассуждая аналогичным образом, получаем, что для обнаружения всех ошибок кратности l требуется код с кодовым расстоянием

$$d \geq l + 1.$$

Для исправления всех ошибок некоторой кратности требуется большее кодовое расстояние, чем для их обнаружения. Если кратность исправляемых ошибок равна l, то кодовое расстояние должно удовлетворять условию

$$d \geq 2l + 1.$$

Очевидно, что кодовое расстояние можно сделать тем больше, чем больше избыточность. Однако при этом уменьшается длительность посылок и возрастает вероятность ошибок при их приеме. Поэтому вводят понятие эффективности избыточного кодирования как отношение вероятностей ошибочного приема кодовой комбинации из k информационных символов при передаче их примитивным и избыточным кодами:

$$\xi = P(k) / P(n, k).$$

Если код примитивный, то ошибка возникает, если хотя бы в одном символе при приеме произошла ошибка. Вероятность такого события

$$P(k) = 1 - (1 - P_{\text{ош.п}})^k \approx kP_{\text{ош.п}},$$

где $P_{\text{ош.п}}$ — вероятность ошибки в приеме одного символа при передаче сообщения примитивным кодом.

Для избыточного кода ошибка в приеме кодовой комбинации будет иметь место тогда, когда число ошибок превысит исправляющую способность кода $l_{\text{и}}$, и ее вероятность

$$P(n,k) = \sum_{i=l_{\text{и}}+1}^{n} C_n^i P_{\text{ош.и}}^i (1 - P_{\text{ош.и}})^{n-i},$$

где $P_{\text{ош.и}}$ — вероятность ошибки в приеме одного символа при передаче избыточным кодом.

Различие в $P_{\text{ош.п}}$ и $P_{\text{ош.и}}$ определяется уменьшением длительности посылки при передаче избыточным кодом, пропорциональной отношению k/n. Величины $P_{\text{ош.п}}$ и $P_{\text{ош.и}}$ могут быть найдены, если известен вид модуляции и демодуляции, отношение $P_{\text{с}}/N_0$ и длительность посылок источника, где P_c – средняя мощность сигнала, N_0 – односторонняя спектральная плотность мощности белого шума.

Таким образом, задача построения кода с заданной корректирующей способностью сводится к обеспечению необходимого кодового расстояния при введении избыточности. При этом желательно, чтобы число используемых проверочных символов было минимальным. К сожалению, задача определения минимального числа проверочных символов, необходимых для обеспечения заданного кодового расстояния, не решена. Имеется лишь ряд оценок для максимального кодового расстояния при фиксированных n и k, которые часто используются при выяснении того, насколько построенный код близок к оптимальному.

Можно показать, что если существует блочный линейный код (n, k), то для него справедливо неравенство

$$r \geq \log_2\left(\sum_{i=0}^{[(d-1)/2]} C_n^i\right),$$

называемое *верхней границей Хэмминга,* где $[(d-1)/2]$ означает целую часть числа $(d-1)/2$.

Граница Хэмминга близка к оптимальной для кодов с большими значениями k/n. Для кодов с малыми значениями k/n более точной является *верхняя граница Плоткина*:

$$r \geq 2d - 2 - \log_2(d).$$

Можно также показать, что существует блочный линейный код (n, k) с кодовым расстоянием d, для которого справедливо неравенство

$$r \leq \log_2\left(\sum_{i=0}^{d-2} C_n^i\right),$$

называемое *нижней границей Варшамова—Гильберта.*

Таким образом, границы Хэмминга и Плоткина являются необходимыми условиями существования кода, а граница Варшамова—Гильберта — достаточным. Эти границы позволяют оценить эффективность блочных кодов и целесообразность их применения.

Циклические коды

Циклические коды относятся к классу линейных систематических. Поэтому для их построения, в принципе, достаточно знать порождающую матрицу [1]. Можно указать другой способ построения циклических кодов, основанный на представлении кодовых комбинаций многочленами $b(x)$ вида

$$b(x) = b_{n-1}x^{n-1} \oplus b_{n-2}x^{n-2} \oplus \ldots \oplus b_1 x \oplus b_0,$$

где b_i, $i = 0, 1, \ldots, n-1$, — символы кодовой комбинации. Над данными многочленами можно проводить все алгебраические действия с учетом того, что сложение здесь осуществляется по модулю 2.

Каждый циклический код (n, k) характеризуется так называемым порождающим многочленом. Им может быть любой многочлен $p(x)$ степени $n-k$, который делит без остатка двучлен $x^n \oplus 1$. Циклические коды характеризуются тем, что многочлены $b(x)$ кодовых комбинаций делятся без остатка на $p(x)$. Поэтому процесс кодирования сводится к нахождению по известным многочленам $a(x)$ и $p(x)$ многочлена $b(x)$, делящегося на $p(x)$, где $a(x)$ — многочлен степени $k - 1$, соответствующий информационной последовательности символов.

Очевидно, что в качестве многочлена $b(x)$ можно использовать произведение $a(x)p(x)$. Однако при этом код оказывается несистематическим, что затрудняет процесс декодирования. Поэтому на практике, в основном, применяется следующий метод нахождения многочлена $b(x)$.

Умножим многочлен $a(x)$ на x^{n-k} и полученное произведение разделим на $p(x)$. Пусть

$$a(x)x^{n-k} = m(x)p(x) \oplus c(x), \qquad (1)$$

где $m(x)$ — частное, а $c(x)$ — остаток. Поскольку операции суммирования и вычитания по модулю 2 совпадают, то выражение (1) перепишем в виде

$$a(x)x^{n-k} \oplus c(x) = m(x)p(x). \qquad (2)$$

Из соотношения (2) следует, что многочлен $a(x)x^{n-k} \oplus c(x)$ делится на $p(x)$ и, следовательно, является искомым.

Многочлен $a(x)x^{n-k}$ имеет следующую структуру: первые $n - k$ членов низшего порядка равны нулю, а коэффициенты остальных совпадают с соответствующими коэффициентами информационного многочлена $a(x)$. Многочлен $c(x)$ имеет степень меньше $n - k$. Таким образом, в найденном многочлене $b(x)$ коэффициенты при x в степени $n - k$ и выше совпадают с информационными символами, а коэффициенты при остальных членах, определяемых многочленом $c(x)$, совпадают с проверочными символами.

В соответствии с формулой (2) процесс кодирования заключается в умножении многочлена $a(x)$ на x^{n-k} и нахождении остатка от деления $a(x)x^{n-k}$ на $p(x)$ с последующим его сложением по модулю 2 с многочленом $a(x)x^{n-k}$.

Операции умножения и деления многочленов легко осуществляются линейными цепями на основе сдвигающих регистров [2]. В качестве примера на рисунке 1,*а* представлена схема умножения многочлена $b(x)$ степени $n = 6$ на многочлен $f(x) = x^3 \oplus x \oplus 1$ по модулю $x^7 \oplus 1$. Нетрудно убедиться, что после семи тактов в регистре записывается многочлен $b(x)f(x)\,\mathrm{mod}(x^7 \oplus 1)$. При делении многочлена $b(x)$ степени $n = 6$ на многочлен $f(x) = x^3 \oplus x^2 \oplus 1$ (рисунок 1,*б*) после семи тактов в регистре оказывается записанным остаток от деления.

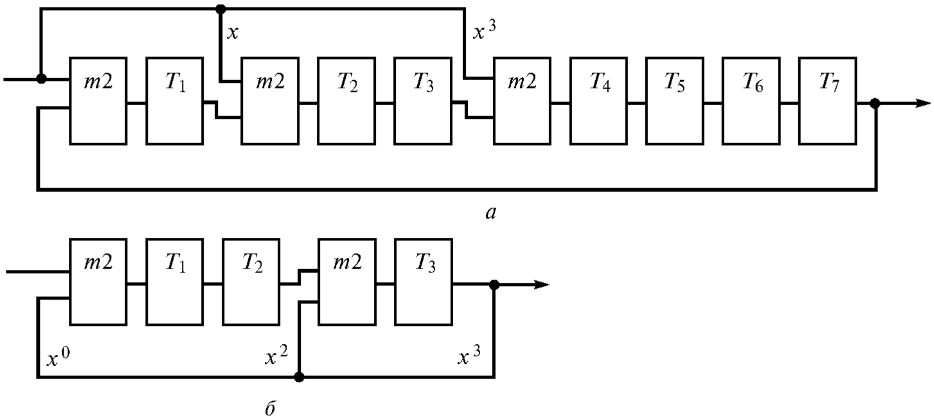

Рисунок 1 - Схемы умножения (*а*) и деления (*б*) многочленов (частный случай)

На основе приведенных схем умножения и деления многочленов строятся кодирующие устройства для циклических кодов. На рисунке 2 в качестве примера приведена схема кодера для кода (7, 4) с порождающим многочленом $p(x) = x^3 \oplus x^2 \oplus 1$. В исходном состоянии ключи $К_1$ и $К_2$ находятся в положении 1.

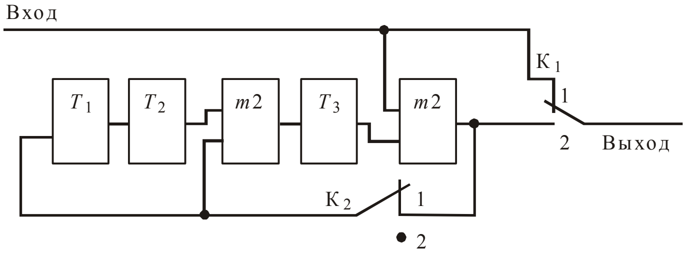

Рисунок 2 - Структурная схема кодера циклического кода с порождающим многочленом $p(x) = x^3 \oplus x^2 \oplus 1$

Информационные символы поступают одновременно на вход канала и на выход ячейки x^3 сдвигающего регистра (это соответствует умножению многочлена $a(x)$ на x^3). В течение четырех тактов происходит деление многочлена $a(x)x^3$ на многочлен $p(x) = x^3 \oplus x^2 \oplus 1$. В результате в регистре записывается остаток, представляющий собой проверочные символы. Ключи K_1 и K_2 перебрасываются в положение 2, и в течение трех последующих тактов содержащиеся в регистре символы поступают в канал.

Циклический код может быть задан проверочным многочленом $h(x)$: кодовая комбинация B принадлежит данному циклическому коду, если $b(x)h(x) = 0 \bmod (x^n \oplus 1)$. Проверочный многочлен связан с порождающим соотношением

$$h(x) = (x^n \oplus 1)/p(x).$$

Задание кода проверочным многочленом эквивалентно заданию кода системой проверочных уравнений. Характерной особенностью циклического кода является то, что все проверочные уравнения можно получить из одного путем циклического сдвига индексов символов, входящих в исходное уравнение. Например, для кода (7, 4) с порождающим многочленом $p(x)= x^3 \oplus$ $\oplus\ x^2 \oplus 1$ проверочный многочлен имеет вид $h(x) = x^4 \oplus x^3 \oplus x^2 \oplus 1$. Проверочные уравнения получаются из условия

$$b(x)h(x) = 0 \bmod (x^7 \oplus 1)\,.$$

Осуществив умножение и приравняв коэффициенты при x^4, x^5 и x^6 нулю, получим следующие уравнения, разрешенные относительно проверочных символов:

$$\begin{aligned} b_2 &= b_6 \oplus b_4 \oplus b_3, \\ b_1 &= b_5 \oplus b_3 \oplus b_2, \\ b_0 &= b_4 \oplus b_2 \oplus b_1. \end{aligned} \tag{3}$$

В качестве примера на рисунке 3 показана схема кодера циклического кода (7,4), задаваемого проверочным многочленом $h(x) = x^4 \oplus x^3 \oplus x^2 \oplus 1$, или, что то же самое, проверочными соотношениями (3).

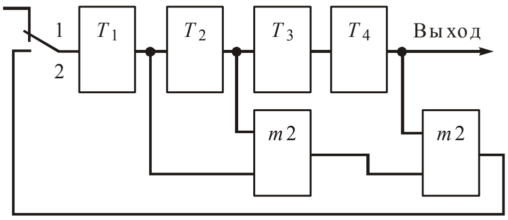

Рисунок 3 - Структурная схема кодера циклического кода, задаваемого проверочным многочленом $h(x) = x^4 \oplus x^3 \oplus x^2 \oplus 1$

В исходном состоянии ключ находится в положении 1. В течение четырех тактов импульсы поступают в регистр, после чего ключ переводится в положение 2. При этом обратная связь замыкается. Начиная с пятого такта, формируются проверочные символы в соответствии с соотношениями (3). После седьмого такта все проверочные символы оказываются сформированными, ключ вновь переключается в положение 1. Кодер готов к приему очередного сообщения. Символы кодовой комбинации поступают в канал, начиная с пятого такта.

Корректирующая способность кода зависит от порождающего многочлена $p(x)$. Поэтому его выбор очень важен при построении циклического кода. Необходимо помнить, что степень порождающего многочлена должна быть равна числу проверочных символов. Кроме того, многочлен $p(x)$ должен делить двучлен $x^n \oplus 1$.

Обнаружение ошибок при использовании таких кодов заключается в делении многочлена $b'(x) = b(x) + e(x)$, соответствующего принятой комбинации $\widetilde{B} = B \oplus e$, на $p(x)$. Если остаток $s(x)$ оказывается равным нулю, то считается, что ошибки нет, в противном случае фиксируется ошибка.

Пусть необходимо построить код, обнаруживающий все одиночные ошибки. В этом случае многочлен ошибок имеет вид $e(x) = x^i$, где $i = 0, 1, \ldots, n - 1$. Решение задачи заключается в нахождении такого многочлена $p(x)$, чтобы многочлен $e(x)$ не делился на $p(x)$. Наиболее простым, удовлетворяющим этому требованию, является многочлен $p(x) = x \oplus 1$.

Аналогично можно построить код, обнаруживающий ошибки большей кратности.

Многочлен $s(x) = (b(x) \oplus e(x)) \bmod (p(x)) = e(x) \bmod (p(x))$ зависит только от многочлена ошибок $e(x)$ и играет ту же роль, что и вектор-синдром [1]. Поэтому в принципе ошибки можно исправлять на основе таблицы соответствий между $e(x)$ и $s(x)$, хранящейся в памяти декодера, как при линейных нециклических кодах. Однако свойство цикличности позволяет существенно упростить процедуру декодирования.

Один из алгоритмов исправления ошибок основан на следующих свойствах синдрома циклического кода. Пусть имеется циклический код с кодовым расстоянием d, исправляющий все ошибки до кратности $l = [(d - 1)/2]$ включительно, где $[(d - 1)/2]$ — целая часть числа $(d - 1)/2$. Тогда можно показать, что:

— если исправляемый вектор ошибок искажает только проверочные символы, то вес синдрома будет меньше или равен l, а сам синдром будет совпадать с вектором ошибок;

— если вектор ошибки искажает хотя бы один информационный символ, то вес синдрома будет больше l;

— если $s(x)$ — остаток от деления многочлена $b(x)$ на $p(x)$, то остатком от деления многочлена $b(x)x^i$ на $p(x)$ является многочлен $s(x)x^i \bmod [p(x)]$, другими словами, синдром некоторого циклического сдвига многочлена $b(x)$ является соответствующим циклическим сдвигом синдрома исходного многочлена, взятого по модулю $p(x)$.

В качестве примера на рисунке 4 представлена схема декодера для кода (7, 4) с порождающим многочленом $p(x) = x^3 \oplus x^2 \oplus 1$. Код имеет кодовое расстояние $d = 3$, что позволяет ему исправлять все однократные ошибки. Принятая кодовая комбинация одновременно поступает в буферный регистр сдвига, служащий для запоминания кодовой комбинации и ее циклического сдвига, и на устройство деления на многочлен $p(x)$ для вычисления синдрома. В исходном состоянии ключ находится в положении 1. После семи тактов буферный регистр оказывается загруженным, а в регистре устройства деления будет вычислен синдром.

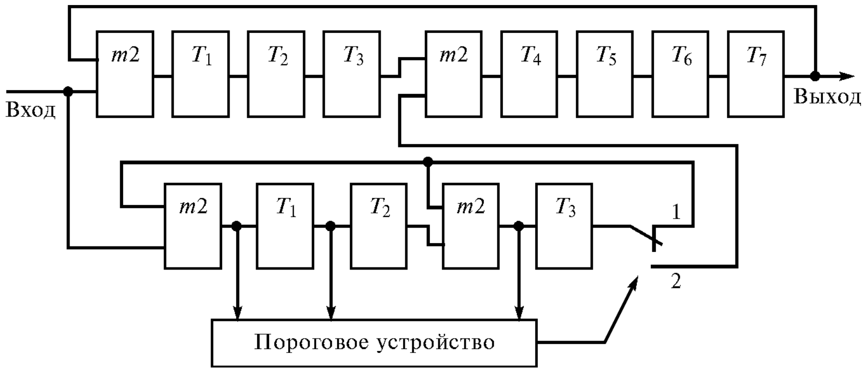

Рисунок 4 - Структурная схема декодера циклического кода с порождающим многочленом $p(x) = x^3 \oplus x^2 \oplus 1$

Если вес синдрома больше единицы, то декодер начинает проводить циклические сдвиги комбинации в буферном регистре при отсутствии новой комбинации на входе и одновременно вычислять их синдромы $s(x)x^i \bmod (p(x))$ в устройстве деления. Если на некотором i-м шаге вес синдрома окажется меньше двух, то ключ переходит в положение 2, обратные связи в регистре деления разрываются. При последующих тактах ошибки исправляются путем подачи содержимого регистра деления на вход сумматора по модулю 2, включенного в буферный регистр. После семи тактов работы декодера в автономном режиме исправленная комбинация в буферном регистре возвращается в исходное положение (информационные символы будут занимать старшие разряды).

Существуют и другие, более универсальные, алгоритмы декодирования. К циклическим кодам относятся коды Хэмминга, которые являются примерами немногих известных совершенных кодов. Они имеют кодовое расстояние $d = 3$ и исправляют все одиночные ошибки. Длина кода выбирается из условия $2^{n-k} - 1 = n$, которое имеет простой смысл: число различных ненулевых синдромов равно числу символов в кодовой последовательности. Так, существуют коды Хэмминга ($2^r - 1$, $2^r - r - 1$), в частности коды (7, 4), (15, 11), (31, 26), (63, 57) и т. д.

Заметим, что ранее использованный многочлен $p(x) = x^3 \oplus x^2 \oplus 1$ является порождающим для кода Хэмминга (7, 4).

ПРАКТИЧЕСКАЯ ЧАСТЬ

Описание программного обеспечения, используемого в работе

Лабораторный стенд состоит из персонального компьютера с установленной на нем системой компьютерной математики (СКМ) MATLAB версии не ниже R2009b.

СКМ MATLAB R2009b имеет средства для анализа характеристик систем передачи информации. Эти средства сведены в пакет **Communications System Toolbox**. В частности, в этот пакет входит мастер построения характеристик помехоустойчивости систем передачи информации, носящий название **Bit Error Rate Analysis Tool**. Этот мастер может быть запущен командой **bertool**, набранной в командном окне СКМ MATLAB. Опишем основные приемы работы с ним.

На рис. 5 показан вид окна мастера **Bit Error Rate Analysis Tool**. Окно мастера имеет три вкладки: **Teoretical**, **Semianalytic** и **Monte Carlo**, в которых при моделировании вводятся исходные данные для расчета характеристик помехоустойчивости. Лабораторная работа будет вестись во вкладке **Teoretical**.

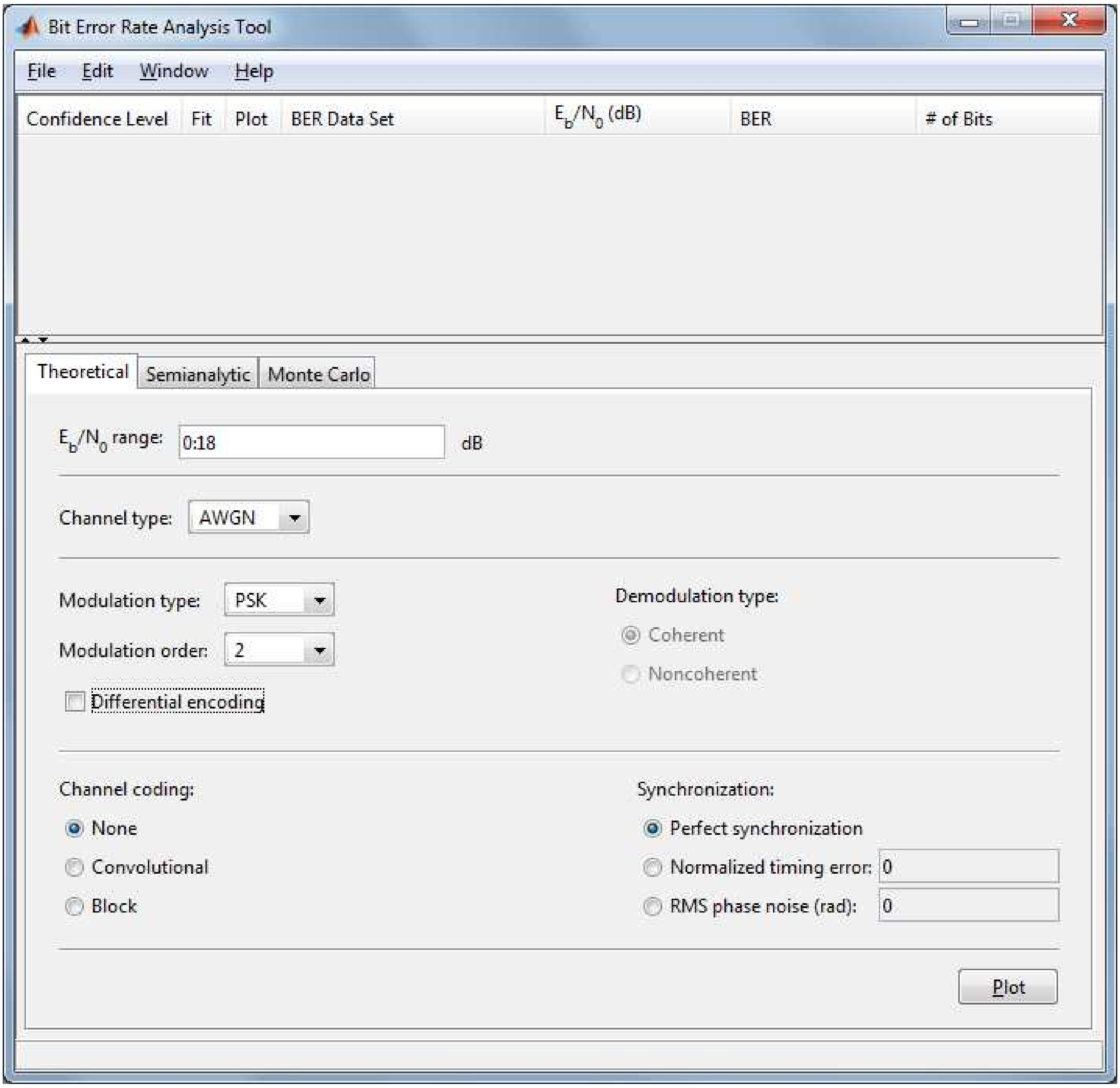

Рис. 5. Вид окна мастера построения характеристик помехоустойчивости **BERTool**

Рассмотрим назначение полей ввода исходных данных для расчета.

В поле **E_b/N_0 range (dB)** вводится диапазон изменения отношения энергии сигнала, приходящейся на один бит **E_b**, к спектральной плотности мощности белого гауссовского шума **N_0** (в децибелах), в котором необходимо построить характеристику помехоустойчивости.

В поле **Channel type** выбирается вид канала, для которого необходимо построить характеристику помехоустойчивости. Доступны следующие виды каналов: **AWGN** (**A**dditive **W**hite **G**aussian **N**oise) – канал, в котором действует только помеха типа аддитивный белый гауссовский шум; **Rayleigh**

– канал, в котором наряду с аддитивным белым гауссовским шумом действуют рэлеевские замирания; **Rician** – канал, в котором наряду с аддитивным белым гауссовским шумом действуют райсовские замирания.

Поле **Modulation type** позволяет выбрать вид модуляции сигнала (**PSK** (**P**hase **S**hift **K**eying) – фазовая манипуляция, **FSK** (**F**requency **S**hift **K**eying) – частотная манипуляция, **QAM** (**Q**uadrature **A**mplitude **M**odulation) – квадратурная амплитудная манипуляция и т.п.). При этом в поле **Modulation order** указывается порядок манипуляции. Если используется относительная ФМ, необходимо отметить поле **Differential encoding**.

При моделировании систем с частотной манипуляцией (**FSK**, **MSK**) в поле **Demodulation type** есть возможность выбрать вид приема (когерентный или некогерентный).

Ниже имеются поля, позволяющие моделировать системы, в которых используется помехоустойчивое кодирование (**Channel coding**). Так, если отмечено поле **None**, это означает, что кодирование не используется, поле **Convolutional** означает использование сверточного, а поле **Block** – блочного кодирования.

Для некоторых наборов исходных данных доступны поля **Synchronization**, позволяющие задать настройки, учитывающие неидеальность работы системы синхронизации модема. Так, отметив поле **Normalized timing error**, можно задать нормированную ошибку работы тактовой синхронизации (вводя значения от 0 до 0,5), которая представляет собой среднеквадратическое отклонение (СКО) момента начала такта, отнесенное к длительности символа. При этом считается, что ошибка синхронизации распределена по нормальному закону. В поле **RMS phase noise (rad)** задается ошибка работы системы фазовой синхронизации в виде СКО фазы опорного генератора приемной части модема, указанного в радианах. Эта ошибка также полагается нормально распределенной.

При необходимости более подробно познакомиться с различными вариантами задания исходных данных на моделирование, можно обратиться к описанию мастера **BERTool**, открыв меню **Help** в окне мастера.

Построенные зависимости вероятности ошибки на бит (**BER** — **b**it **e**rror **r**ate), появляются в отдельном окне, при этом в окне мастера в верхней части появляются строчки с описанием параметров построенных зависимостей. При построении каждой зависимости на графике присваивается имя по умолчанию. Имя зависимости можно редактировать, дважды щелкнув левой кнопкой мыши в графе **BER Data Set** нужной строчки. Далее введенное Вами имя зависимости будет отображаться и в легенде на графике.

Порядок выполнения работы

Последовательность действий при выполнении практической часть работы следующая.

1. Запустить MATLAB.
2. В командном окне (Command Window) MATLAB набрать команду **bertool** и нажать «Ввод». При этом откроется окно мастера построения характеристик помехоустойчивости **Bit Error Rate Analysis Tool** (рис. 5).
3. Руководствуясь приведенным выше описанием построить характеристики помехоустойчивости для канала без замираний и без использования кодирования для следующих видов манипуляции:

   - фазовой – ФМ-2 (с ОФМ-кодеком и без), -4, -8, -16 на одном графике;

   - амплитудно-фазовой – QAM-16, -32, -64, -256 на одном графике;

   - построить отдельный график, на котором сравнить ФМ-16 (PSK-16) и QAM-16;

   - частотной – FSK-2 (со значением коэффициента взаимной корреляции 0), -4, -16, -32 при когерентном приеме;

   - построить отдельный график, на котором сравнить FSK-2 со значениями коэффициента взаимной корреляции 0 и -0,21 при когерентном приеме и FSK-2 со значением коэффициента корреляции 0 при некогерентном приеме.

4. Изучить влияние замираний на помехоустойчивость, для чего на одном графике построить кривые помехоустойчивости для ФМ-2 при отсутствии замираний и при наличии рэлеевских замираний при одиночном приеме и с использованием разнесенного приема на 2 и на 3 ветви (поле **Diversity order**).
5. Изучить влияние канального кодирования на характеристики помехоустойчивости. Для этого на одном графике построить зависимости вероятности ошибки для ФМ-2 в канале без замираний при отсутствии кодирования, при использовании кодов Хэмминга (7,4) и (31,26), при использовании сверточного кода с относительной скоростью ½ и длиной кодового ограничения 7 при использовании жесткого и мягкого правил принятия решения (поля **Hard** и **Soft** соответственно).
6. Провести расчеты, аналогичные п. 5 для случая канала с рэлеевскими замираниями при одиночном приеме.
7. По графикам, построенным в п.п. 5 и 6, определить величину энергетического выигрыша, обеспечиваемого использованием различных вариантов помехоустойчивого кодирования.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое расстояние Хэмминга и кодовое расстояние?
2. От чего зависит корректирующая способность кода?
3. Поясните процесс кодирования при использовании циклических кодов.
4. Поясните процесс декодирования при использовании циклических кодов.
5. Что такое проверочный многочлен? Как он находится?
6. Что такое верхняя граница Хэмминга, верхняя граница Плоткина, нижняя граница Варшамова-Гильберта?
7. Что такое эффективность избыточного кодирования? Как она определяется?

ДОМАШНЕЕ ЗАДАНИЕ

1. Приведите структурные схемы кодирующего и декодирующего устройств кода Хэмминга (15,11), если его порождающий многочлен имеет вид $p(x) = x^4 \oplus x^3 \oplus x^2 \oplus x \oplus 1$.
2. При передаче информации используются четыре кодовых комбинаций: 00000, 01011, 10110, 11101. Определите кодовое расстояние. Оцените корректирующую способность кода.
3. Определите необходимое число проверочных символов в коде (15, *r*), исправляющем однократные ошибки.

ТРЕБОВАНИЯ К ОТЧЕТУ

Отчет о работе должен содержать:

1. Ответы на контрольные вопросы.
2. Решение домашнего задания.
3. Экспериментальные графики с пояснениями.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Информационные технологии в радиотехнических системах: учеб. пособие / под ред. И.Б. Федорова. – Изд. 3-е перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2011. – 846с.
2. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки: Пер. с англ. / Под. ред. Р.Л. Добрушина и С.И. Самойленко. – М.: Мир, 1976.