

**КОМПЬЮТЕРНАЯ ИМИТАЦИОННАЯ МОДЕЛЬ
ПЕРЕДАЧИ ИНФОРМАЦИИ НА ОСНОВЕ ПЕРЕМЕЖИТЕЛЯ
И ДВУМЕРНОГО ЛИНЕЙНОГО ИТЕРАТИВНОГО КОДА
С ОБЪЕДИНЕННЫМИ ДВОЙНЫМИ ДИАГОНАЛЬНЫМИ ПРОВЕРКАМИ**

The article describes the computer simulation model of the information transmissions on the basis of interleaving and two-dimensional linear iterative code with the united dual diagonal checks. This complex information transformation makes it possible to correct repeated errors with smaller expenditures of time and simplicity of the algorithm correction. In model is realized the opportunity of management by throughput of the channel, likelihood characteristics of the errors arising at data transmission, and also definition of amount of not corrected erroneous bit after decoding.

Введение. Как известно, при передаче и хранении информации все больше факторов влияют на ее качество, внося в последовательности данных различного рода ошибки. Для поиска и исправления таких ошибок используются разнообразные методы. Наиболее эффективными из таких методов являются те, которые базируются на применении помехоустойчивых корректирующих кодов, перемежении символов исходной последовательности и степени группирования [1–3].

С увеличением кратности ошибки усложняется алгоритм ее обнаружения и исправления, что влечет дополнительную задержку во времени при передаче данных.

Одним из направлений решения проблемы является комплексное преобразование передаваемой информации, состоящее из нескольких этапов. На первом этапе необходимо избавиться от группирующихся ошибок путем преобразования их в одиночные, а на втором – с помощью соответствующих кодов, которые позволяют исправить все или большинство ошибок. Отсутствие статистических данных о кратности и интенсивности ошибки, возникающей в канале передачи данных, а также трудность и относительно высокая стоимость проведения реальных экспериментов послужили причиной создания компьютерной имитационной модели.

Основная часть. Для исследования эффективности комплексного преобразования данных в разработанной компьютерной модели выполнена имитация двух схем передачи информации (рис. 1). Процесс передачи информации проходит через эти две схемы параллельно.

Компьютерная имитационная модель передачи информации реализована в среде Delphi 7. За основу исходных данных взят массив, заполненный двоичной информацией случайным образом. Кодер построен на основе ранее разработанного усеченного двумерного линейного итеративного кода с объединенными двойными диагональными проверками [4]. Кодирование информации осуществляется путем добавления проверочных бит к информационной последовательности. Формирование проверочных бит матрицы размерностью $k1$ строк на $k2$ столбцов по вертикали, горизонтали и двум диагоналям производится путем суммирования по модулю два всех информационных символов каждого из направления. По своим корректирующим способностям данный код позволяет исправить только одиночные и двойные ошибки. Декодер построен на основе синдромного декодирования, являющегося одним из наиболее распространенных методов коррекции ошибок блочными кодами. Данный декодер считается декодером максимального правдоподобия при декодировании жестких решений.

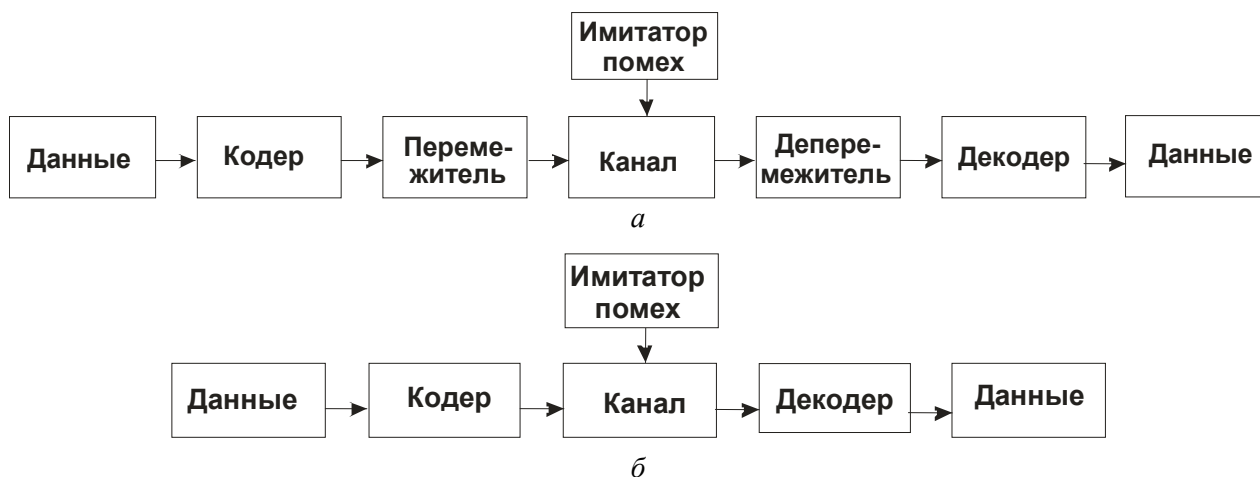


Рис. 1. Схемы передачи информации:
а – на основе кодера и перемежителя; б – на основе кодера

На основе ранее разработанного псевдо-случайного метода перемежения построены блоки перемежитель и деперемежитель [5]. Процедура перемежения состоит в перестановке символов кодированной последовательности до ее модуляции и восстановлении исходной последовательности после демодуляции. Перестановка позволяет так разнести рядом стоящие символы, чтобы они оказались разделены группой других символов, передаваемых в том же блоке данных. Данная операция не вносит избыточности, а только изменяет порядок следования символов или бит. В разработанном методе перемежение (деперемежение) основано на использовании матрицы, строки и столбцы которой формируются последовательностью входных символов. Для удобства реализации метода матричные расчеты преобразованы в линейные:

$$adr_k \leftarrow \left(\left(\text{int} \left(\frac{k}{n2-1} \right) + 1 \right)^{d1} \bmod n1 \right) - 1 + \left(\left((k \bmod (n2-1) + 1)^{d2} \bmod n2 \right) - 1 \right) \times (n2-1) = v,$$

где int – целочисленное деление; k – адрес памяти ($0 \leq k \leq (N-1)$); $n2$ – длина столбцов; $n1$ – длина строк; $d1, d2$ – степенные коэффициенты. В приведенном соотношении учитывается тот факт, что считывание символов осуществляется по столбцам. Данная формальная запись означает, что v -й бит входной последовательности будет считан на выходе под k -м номером (adr_k), что позволяет определить местоположение бит в выходной последовательности без перемежения строк и столбцов. Данная запись используется и при деперемежении с небольшой модификацией (заменой степенных коэффициентов $d1, d2$ на $e1, e2$). Реализация определения местоположения бита выполнено следующей функцией:

```
function TPsevo.Pos(const i : integer; const R : TRec) : integer;
var
  PosA, PosB : Integer;
  S : AnsiString;
begin
  PosA := (Power((i div (R.n)) + 1, R.e, (R.nr + 1)) mod (R.nr + 1)) - 1; // определение строкового местоположения бит после перемежения строк
  PosB := (Power((i mod (R.n)) + 1, R.d, (R.n + 1)) mod (R.n + 1)); // определение столбчатого местоположения бит после перемежения столбцов
  PosB := (PosB - 1) * R.nr;
  Result := PosA + PosB; //определение позиции выходного бита
end;

Power – функция, реализующая возведение в степень типа  $x^y$ :
function TPsevo.Power(const A, B : Integer; const AMod : Integer = 0) : Integer;
```

```
var i : integer;
begin
  Result := 1;
  if B > 0 then
    begin
      for i := 1 to B do
        begin
          Result := Result * A;
          if AMod <> 0 then
            Result := Result mod AMod;
          end;
        end;
    end;
end;
```

Для удобства эксплуатации модели создан интерфейс пользователя, вид которого отображен на рис. 2.

Перед началом моделирования необходимо определить начальные параметры модели. В окне *Входные данные* нужно осуществить выбор пропускной способности моделируемого канала передачи данных из наиболее распространенных на практике. Одними из таких каналов являются волоконно-оптические каналы, характеризующиеся малым ослаблением сигнала и высокой степенью защищенности от внешних электромагнитных помех [6]. Далее следует определить объем передаваемой информации в мегабайтах, выбрать длину информационного слова для усеченного двумерного линейного итеративного кода с объединенными диагональными проверками в битах. В окне *Защумленность канала* необходимо указать вероятность появления ошибки для каждого вида ошибок (одиночных, двойных, тройных и т. д.).

В начале моделирования, исходя из значения длины информационного слова кода, определяется оптимальное количество строк и столбцов проверочной матрицы с точки зрения избыточности. Оптимальной матрицей является квадратная матрица, в которой $k1$ строк равно $k2$ столбцов. Данное утверждение доказано в работе [1]. В случае, если невозможно сформировать квадратную проверочную матрицу, то необходимо воспользоваться матрицей, близкой к квадратной. Данное решение реализовано следующим образом:

```
S := $FFFFFFFF;
while i >= 1 do
  begin
    if fInputData.edBlockSize.Value mod i = 0 then
      if fInputData.edBlockSize.Value * 3 + i + i * fInputData.edBlockSize.Value < S then // определение блока с минимальным количеством проверочных символов
        begin
          k1 := i;
          k2 := fInputData.edBlockSize.Value div i;
          S := k2 * 3 + k1 + k1 * k2;
          Data_and_Likdd_k1 := k1;
          Data_and_Likdd_k2 := k2;
        end; // определение значений k1 строк и k2 столбцов
      Dec(i);
      fProcessInfo.pbProcess.Progress := fProcessInfo.pbProcess.Progress + 1;
    end;
```

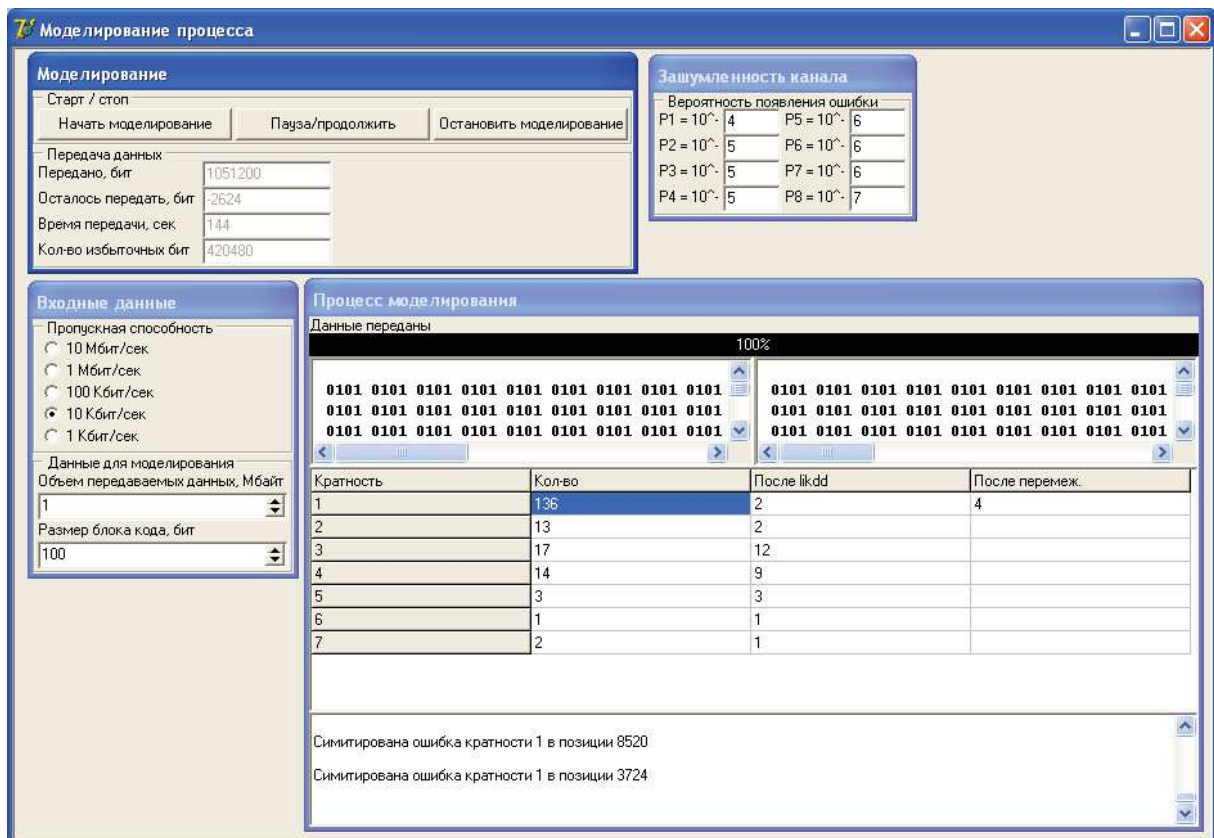


Рис. 2. Интерфейс компьютерной имитационной модели

Определяется размер передаваемого пакета данных, равный количеству кодовых последовательностей (информационное слово + проверочные символы), входящих в этот пакет, и не превышающий пропускную способность канала.

В окно *Размер блока перемежителя* заносятся все варианты размеров перемежителей и возможные степенные коэффициенты в зависимости от пропускной способности канала и размера пакета передаваемой информации. Так, первой цифрой является количество строк, второй – количество столбцов, третьей – степенной коэффициент для перемежения строк, четвертой – степенной коэффициент для перемежения столбцов, пятой – степенной коэффициент для деперемежения строк, шестой – степенной коэффициент для деперемежения столбцов (рис. 3). Выбирая из предложенных вариантов, необходимо помнить, что разница между вариантами заключается в изменении минимального расстояния разнесения бит информационной последовательности.

Процесс моделирования можно контролировать путем нажатия на кнопки *Начать моделирование*, *Пауза/продолжить*, *Остановить моделирование*. В процессе моделирования выводится в окно *Моделирование* информация о количестве переданной информации, времени, затраченном на ее передачу, количестве переданных проверочных символов. Количество и кратность ошибок, имитированных во время

эксперимента, отображается в таблице окна *Процесс моделирования*. Также в этой таблице показывается количество неисправленных ошибок и их кратность для каждой схемы моделирования (после *likdd* – на основе кодера и без перемежителя, после перемежителя – на основе кодера и перемежителя). В нижней части окна ведется история выбранных начальных параметров модели, имитированных и неисправленных ошибок, их кратность и позиция. Данная информация записывается в файл.

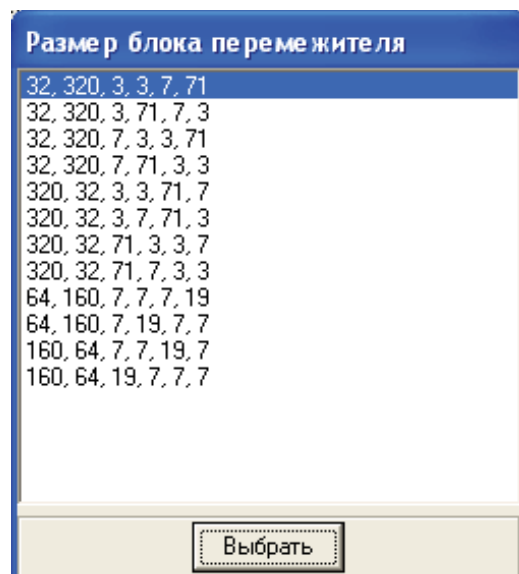


Рис. 3. Окно *Размер блока перемежителя*

В ходе проведения экспериментов выяснилось, что не всегда выбранная длина информационного слова кода и параметры процесса перемежения способны корректировать ошибки, возникающие в процессе передачи информации. Так, при средней вероятности появления ошибки, равной 10^{-5} , на 1 Мбайт передаваемой информации с размером блока перемежения 32×32 , пропускной способностью 1 кбит/с, длиной информационного слова 100 моделью имитируется в среднем 300 ошибок. Из общего количества ошибок в среднем 90 не исправляется при использовании схемы передачи данных без перемежителя и 20 ошибок – с перемежителем. Изменяя в модели только размер блока перемежения на 32×320 и пропускную способность канала на 10 кбит/с, получим уменьшение неисправленных ошибок в схеме с использованием перемежителя в среднем до 1 при неизменном объеме передаваемой информации. Другим способом сокращения неисправленных ошибок является уменьшение длины информационного слова усеченного двумерного линейного итеративного кода с объединенными диагональными проверками, но при этом увеличивается объем передаваемой информации. Данный факт объясняется не совсем оптимальным выбором начальных параметров модели.

Заключение. Разработанная компьютерная модель передачи информации показала, что выбранный вариант комплексного преобразования передаваемой информации на основе перемежителя и двумерного линейного итератив-

ного кода с объединенными двойными диагональными проверками позволяет корректировать все или практически все ошибки при меньших затратах времени и простоте алгоритма коррекции.

Литература

1. Урбанович, П. П. Избыточность в полупроводниковых интегральных микросхемах памяти / П. П. Урбанович, В. Ф. Алексеев, Е. А. Верниковский. – Минск: Наука и техника, 1995. – 263 с.
2. Мак-Вильямс, Ф. Теория кодов, исправляющих ошибки / Ф. Мак-Вильямс, Н. Слоэн; пер. с англ.; под ред. Л. А. Басалыго. – М.: Связь, 1979. – 746 с.
3. Скляр, Б. Цифровая связь. Теоретические основы и практическое применение / Б. Скляр; пер. с англ. – 2-е изд. – М.: Издательский дом «Вильямс», 2003. – 1104 с.
4. Шиман, Д. В. Метод псевдоравномерного перемежения символов турбокода / Д. В. Шиман // Труды БГТУ. Сер. VI, Физ.-мат. науки и информ. – 2005. – Вып. XIII. – С. 168–170.
5. Шиман, Д. В. Свойства и параметры линейных итеративных кодов с двойными диагональными проверками / Д. В. Шиман, Д. М. Романенко // Труды БГТУ. Сер. VI, Физ.-мат. науки и информ. – 2007. – Вып. XV. – С. 151–154.
6. Гуров, И. П. Основы теории информации и передачи сигналов / И. П. Гуров. – СПб.: ВНУ, 2000. – 97 с.