**Руководство по использованию программы**

**“Блоковый многопороговый декодер со стираниями”**

Программа позволяет исследовать эффективность блокового многопорогового декодера в канале связи со стираниями. При этом выполняется моделирование системы передачи данных, состоящей из источника данных, кодера самоортогонального кода, модулятора, канала связи, демодулятора, многопорогового декодера и приемника данных. Источник данных формирует поток данных, состоящий из 0 и 1, появляющихся с равной вероятностью. Эти данные поступают в кодер, которые выполняет их кодирование с помощью блокового самоортогонального кода, заданного порождающим полиномом из файла encoder.txt. В результате получается кодовое слово, которое передается по каналу связи со стираниями, где каждый символ с вероятностью *p* заменяется признаком стирания, а с вероятностью 1 – *p* передается без искажений. Принятое из канала сообщение поступает на вход многопорогового декодера, который исправляет стирания в принятых данных. Результат декодирования сравнивается с передаваемыми данными в приемнике данных и подсчитывается количество оставшихся стертых символов.

Для работы программы требуется три файла:

1. erasureMTD.exe – исполняемый файл программы.

2. encoder.txt – файл с параметрами используемого кодера.

3. param.txt – файл с параметрами эксперимента и декодера.

После запуска программа считывает исходные данные из файлов с параметрами и выполняет в соответствии с ними моделирование, в процессе которого оценивается вероятность стирания на выходе многопорогового декодера при заданной вероятности стирания в канале связи. Подробная статистика работы декодера выводится в файл result.txt, который создается в текущем каталоге.

Параметры файла param.txt:

Первыми тремя величинами (P0\_first, P0\_last, P0\_step) задается диапазон и шаг изменения вероятности стирания в канале связи:

P0\_first=0.43 //начальное значение вероятности стирания бита

P0\_last=0.29 //конечное значение вероятности стирания бита

P0\_step=0.01; //шаг изменения значения вероятности стирания бита

В данном случае будет выполнена оценка вероятности стирания на выходе многопорогового декодера для каждого значения вероятности стирания в канале на интервале от 0.49 до 0.29 с шагом изменения 0.01.

Следующий параметр volume задает максимальное число передаваемых битов в процессе эксперимента.

Параметр minBlockErr определяет число блоков со стираниями, после появления которых при текущем значении вероятности стирания в канале связи можно завершить моделирование (данный параметр позволяет за минимальное время обеспечить требуемую погрешность полученных результатов).

Параметр itert определяет число итераций многопорогового декодирования. Данный декодер является итеративным декодером мажоритарного типа, в котором выполняется несколько этапов (итераций) декодирования одного принятого блока. В демопрограмме установлено ограничение – число итераций декодирования должно быть не менее 3 и не более 10.

Параметры файла encoder.txt:

nk=2 nr=2

i=1 v=1 2 6 82

i=2 v=1 2 13 141

i=1 v=2 2 0 134

i=2 v=2 2 9 207

Параметр nk задает число информационных ветвей кодера (в данной демопрограмме установлено ограничение – не более четырех информационных ветвей), параметр nr задает число проверочных ветвей кодера (в данной демопрограмме установлено ограничение – не более четырех проверочных ветвей). При кодировании исходная информационная последовательность заносится в nk регистров сдвига одинаковой длины (длина регистров равна 2*p*+1, где *p* – максимальная степень образующего полинома). Далее, используя операцию суммирования по модулю 2, вычисляются nr проверочных битов. После этого выполняется циклический сдвиг информационных регистров и снова вычисляются nr проверочных символов. Данный процесс повторяется до тех пор, пока информационные регистры не вернутся в исходное состояние (2*p*+1 сдвигов).

Далее задаются полиномы, связывающие i-ю информационную и v-ю проверочную ветви. Для каждой пары информационная–проверочная ветвь (например, i=1 v=1) задается количество проверочных символов i-й информационной ветви, участвующих в формирования каждого символа v-й проверочной ветви (для кода в примере используются две проверки), затем указываются конкретные позиции используемых проверок, начиная с нуля (например, запись i=1 v=1 2 6 82 означает, что при формировании нулевого символа первой проверочной ветви будут использованы 6-й и 82-й символы первой информационной ветви).

При переборе всех пар информационная–проверочная ветвь для указания позиций проверок сначала фиксируется первая проверочная ветвь и для нее перебираются все информационные ветви, затем фиксируется вторая проверочная ветвь и для нее перебираются все информационные ветви и т.д.

i=1 v=1 2 6 82

i=2 v=1 2 13 141

i=1 v=2 2 0 134

i=2 v=2 2 9 207

В демоверсии программы значение позиции каждой из проверок должно быть меньше 1000. Если одна из позиций проверок превышает это значение, то программа выдаст сообщение об этом и сразу завершит свою работу.

Пример схемы кодера блокового кодера самоортогонального кода с одной информационной и одной проверочной ветвями представлен на рисунке. Видно, что кодер состоит только из сдвигового регистра и сумматоров по модулю 2.



Для использования такого кодера в файле encoder.txt необходимо задать:

nk=1 nr=1

i=1 v=1 4 0 1 4 6

Пример схемы двоичного блокового кодера самоортогонального кода с двумя информационными и двумя проверочными ветвями представлен на следующем рисунке.



Для использования такого кодера в файле encoder.txt необходимо задать:

nk=2 nr=2

i=1 v=1 2 0 1

i=2 v=1 2 0 4

i=1 v=2 2 0 6

i=2 v=2 2 0 15

В выходном файле result.txt выводится информация о параметрах моделирования, модуляторе, кодере и декодере, используемых при моделировании, а также для каждого значения вероятности стирания в канале связи выводится информация по оцененной вероятности стирания на выходе декодера и количеству стираний, оставшемуся после каждой итерации декодирования.