

РЕАЛИЗАЦИЯ АЛГОРИТМА ПОИСКА ОБРАЗУЮЩИХ ПОЛИНОМОВ ДЛЯ ЦИКЛИЧЕСКИХ КОДОВ, ИСПРАВЛЯЮЩИХ ПАКЕТНЫЕ ОШИБКИ

Мышко Е.А.

Научный руководитель: А.Н. Мальчуков
Томский политехнический университет
evgenvt@tpu.ru, 1man@tpu.ru

Введение

Для обеспечения надежности передачи данных применяются полиномиальные помехоустойчивые коды [1]. Для построения помехоустойчивого кода необходимо подобрать образующий полином, удовлетворяющий заданным условиям, таким как, длина информационного блока и кратность исправляемости ошибок. В работе [2,3] приведена реализация алгоритма поиска образующих полиномов для кода, исправляющего независимые ошибки. Далее будет рассмотрена реализация алгоритма поиска образующих полиномов для циклических кодов, исправляющих пакетные ошибки.

Описание алгоритма поиска образующих полиномов

Алгоритм поиска образующих полиномов для кодов, исправляющих пакетные ошибки заключается в проверке полиномов-кандидатов заданной длины и веса на выполнение необходимых и достаточных условий [4] для соответствия образуемому полиному. На рис. 1. представлена блок-схема алгоритма поиска образующих полиномов. Далее представлено пошаговое описание алгоритма поиска образуемого полинома для помехоустойчивых кодов, исправляющих пакетные ошибки.

Начало.

Шаг 1. Задаются m – разрядность информационного блока и p – желаемая корректирующая способность кода.

Шаг 2. Вычисляется минимальная длина контрольного блока (старшая степень образуемого полинома) по формуле для поиска «границы Хэмминга»[4]:

$$k = \begin{cases} t = 1, n + 1 \\ t = 2, 2 * n + 1, \\ t > 2, 2 * n + 1 + \sum_{i=1}^{p-2} n * 2^n \end{cases}$$

где k – старшая степень образуемого полинома, p – количество пакетных ошибок, исправляемых кодом, n – разрядность кодового слова.

Шаг 3. Выбирается начальное значение веса полинома $w = p$. Минимальное расстояние кода, исправляющего p пакетных ошибок, вычисляется по формуле [4]: $d = p$.

Шаг 4. Выбирается полином из множества, состоящего из полиномов со старшей степенью, равной k и весом, равным w .

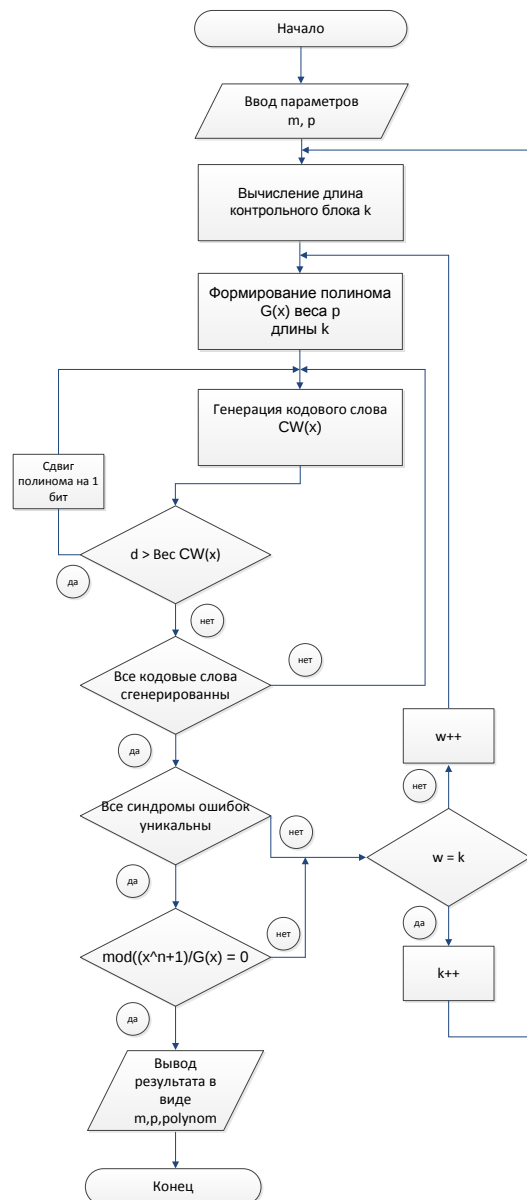


Рис. 1. Блок-схема алгоритма поиска образующих полиномов для кодов, исправляющих пакетные ошибки

Шаг 5. Строится помехоустойчивый полиномиальный код на основе выбранного образуемого полинома и генерируются все кодовые слова.

Шаг 6. Вычисляется минимальное кодовое расстояние построенного кода (путём нахождения минимального веса среди всех кодовых слов, за исключением нулевого слова).

Шаг 7. Если вычисленное кодовое расстояние удовлетворяет заданным требованиям, то переходим на шаг 8, иначе переходим на шаг 11.

Шаг 8. Вычисляются синдромы ошибок путем деления всех комбинаций пакетных ошибок на образующий полином.

Шаг 9. Если все синдромы ошибок уникальны для построенного кода, то осуществляется переход на шаг 10, иначе переходим на шаг 11.

Шаг 10. Если полином $x^n + 1$ делится на образующий полином без остатка, где n – длина кодового слова, то алгоритм завершает работу, иначе переходим на шаг 11.

Шаг 11. Если проверены все полиномы из множества (множество задано на шаге 4), то переходим на шаг 12, иначе переходим на шаг 4.

Шаг 12. Если вес w равен k , увеличиваем k на единицу и переходим на шаг 4, иначе увеличиваем вес w и переходим на шаг 4.

Конец.

Реализация алгоритма поиска образующих полиномов

Алгоритм поиска образующих полиномов для кодов, исправляющих пакетные ошибки реализован на языке C++ с применением объектно-ориентированного программирования. Структура программной реализации представлена на рис. 2.

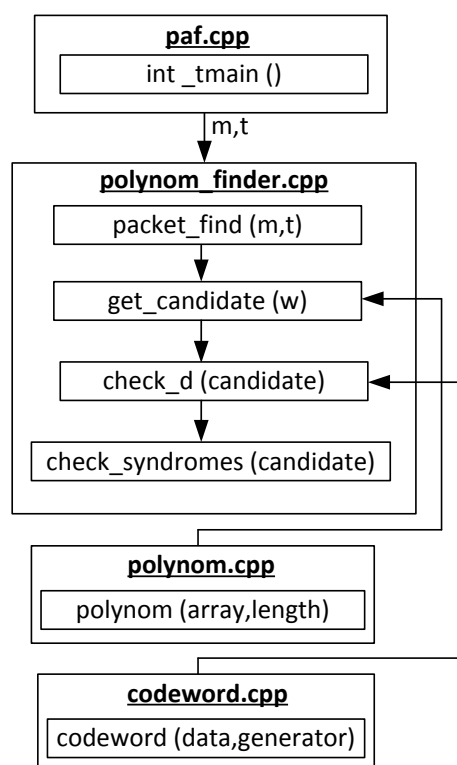


Рис. 2 Структура программной реализации алгоритма поиска

В главной программе вызывается метод `packet_find` с параметрами m – длина информационного блока, p – кратность исправляемых ошибок. В данном методе в цикле последовательно вызываются методы:

- `get_candidate(w)` – получение полинома-кандидата веса w (шаг 4 алгоритма), для проверки на выполнение необходимых и достаточных условий для образующего полинома. Метод также включает вызов конструктора `polynom`;
- `check_d(candidate)` – проверка полинома *candidate* на соответствие кодовому расстоянию d (неоходимое условие, шаг 6 –7); Метод также включает вызов конструктора `codeword`.
- `check_syndromes(candidate)` – проверка синдромов ошибок для полинома *candidate* на уникальность (достаточное условие, шаг 8 – 9) и проверка остатка от деления полинома $x^n + 1$ на *candidate*. Если все синдромы ошибок уникальны и остаток от деления равен нулю, то полином *candidate* принимается за образующий.

Заключение

В данной работе рассмотрена реализация алгоритма поиска образующего полинома для построения помехоустойчивых кодов, исправляющих пакетные ошибки. Приведена блок-схема алгоритма поиска и пошаговое описание. Рассмотрена структура программной реализации алгоритма поиска образующего полинома, представляющая последовательность вызовов методов получения полинома-кандидата и проверки на выполнение необходимых и достаточных условий для образующего полинома.

Список литературы

1. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования: методы, алгоритмы, применение: учебное пособие: пер. с англ. – М.: Техносфера, 2006. – 320 с.
2. Mytsko E. A., Malchukov A. N. Adaptation of technology MPI and OpenMP to search for the generators polynomials // 9th International Forum on Strategic Technology (IFOST-2014): Proceedings, Chittagong, October 21-23, 2014. - Chittagong: CUET, 2014 - p. 5-8.
3. Mytsko E. A., Malchukov A. N. Application of parallel computing technology openmp to search for the generator polynomials // Mechanical Engineering, Automation and Control Systems: Proceedings of International Conference, Tomsk, October 16-18, 2014. - Tomsk: TPU Publishing House, 2014 - p. 1-5
4. Мальчуков А.Н. Алгоритмическое и программное обеспечение системы для разработки кодеров помехоустойчивых кодов : диссертация кандидата технических наук : 05.13.11 / Мальчуков Андрей Николаевич; [Место защиты: Том. политехн. ун-т].- Томск, 2008.- 151 с.: ил. РГБ ОД, 61 09-5/472.