

## Ответы на ваши вопросы №№ 10 - 11

Мы просим наших читателей обратить своё внимание на то, что многие ваши вопросы достойны анализа в специальной отдельной книге. Они вполне заслуживают развёрнутой дискуссии.

В связи с этим некоторые из наших ответов будут не всегда короткими, хотя действительно глубокое обсуждение затрагиваемых вами проблем требует иногда многих десятков страниц серьёзного материала.

Если вы согласны с нами в этом важном аспекте обсуждения методов высокой достоверности передачи цифровых данных, то ваши мнения и новые вопросы о развитии теории и техники кодирования будут хорошей основой для диалога авторов с читателями нашего веб-сайта.

### 10. Чем отличаются различные корректирующие коды и методы их декодирования? Почему их так много?

Если посмотреть ответ-пример на вопрос №3 (ТЗ на коды), то легко заметить, что проект на систему кодирования содержит до трёх десятков пунктов, которые надо учесть и рассмотреть. Причина этого проста: требования к кодам по входным и выходным вероятностям, а также по многим другим параметрам могут быть очень и очень разными. И от этого сильно зависит, в свою очередь, выбор и самого метода коррекции ошибок, т.е. алгоритма декодирования, и его сложность. Неопытность заказчика системы может привести к написанию ТЗ, которое окажется невозможно выполнить. Например, нельзя исключить того, что его требования по вероятностям ошибок и избыточности будут соответствовать обратному неравенству  $R > C$  (кодовая скорость превышает пропускную способность канала), когда вообще любые системы кодирования неэффективны. Эта ситуация полностью аналогична нарушению закона сохранения энергии в механике.

Но и этого мало. Коды нужны для исправления ошибок. Сами ошибки при передаче цифровых данных могут возникать из-за очень разных причин. Но тогда они описываются и различными моделями генерации ошибок. Это тоже приводит к специфическим методам коррекции ошибок, т. е. к новым алгоритмам.

Более того, возможны и разные системы формирования передаваемых сигналов. Они также являются причиной дальнейшего изменения модели возникновения ошибок. А вот если

этого не учитывать, то результат «создания» системы кодирования по принципу «ну, вот я так хочу» может быть просто провальным. Пример подобного «достижения» мы уже приводили в ответе на вопрос №9. Там причина неуспеха была в том, что разработчики не согласовали систему сигналов и связанную с ней модель ошибок канала с кодами и методами их декодирования. Канал был двоичным, а коды - с недвоичным основанием  $q$ , причём, в том примере основание  $q$  было достаточно большим,  $q=64 \gg 1$ , что дополнительно усиливало рассогласование алгоритма и модели ошибок.

Отметим, наконец, что успехи в построении, например, алгоритмов декодирования для двоичных каналов существенно помогают продвижению и методов коррекции ошибок для многих других систем сигналов. ( Например, для многопозиционной ФМ, для системы типа АФМ-16 - сигналов на плоскости  $4 \times 4$  и т. д. . ) В этом нет ничего удивительного, так как многие параметры различных кодовых систем взаимосвязаны.

Вот и получаем, что наличие разных методов декодирования для различных сигналов – это нормально. В частности, алгоритм **МПД существует в различных модификациях**: для двоичных, недвоичных и стирающих каналов, для метрик Хемминга и Евклида, для круговых ФМ, для АФМ-N и некоторых других моделей. Опыт разработок и применения МПД свидетельствует об очень простой его адаптируемости ко многим (возможно, почти ко всем) системам сигналов и моделям шума канала.

А вот если для некоторого конкретного ТЗ различными исполнителями предлагаются совершенно разные решения, то тут надо внимательно смотреть, есть ли среди них действительно хорошие варианты. Возможно, что все они демонстрируют совершенно недостаточные знания подрядчиков в области передовых технологий декодирования. Надеемся, наши читатели уже понимают, что таких методов совсем немного.

## 11. Почему МПД гораздо проще турбо декодеров?

Рассмотрим проблему в два этапа. Когда мы не интересуемся проблемой сложности, а рассматриваем только эффективность, то в этом случае оптимальным декодером (ОД) для кода длины  $n$  с кодовой скоростью  $R$  выбирается по метрике Хемминга или Евклида ближайшее к принятому вектору  $Q$  кодовое слово из экспоненциально большого множества всех возможных  $2^{nR}$  векторов. Но даже при  $n=1000$  выбор ближайшего к  $Q$  кодового

вектора приводит всё ещё к достаточно большой вероятности неправильного декодирования. А ведь это соответствует просмотру такого огромного числа  $2^{nR} = 2^{500}$  кодовых векторов, которое даже больше числа атомов во Вселенной!

Конкретные графики для нижних оценок этих вероятностей показаны во всех презентациях нашего сайта [www.mtdbest.iki.rssi.ru](http://www.mtdbest.iki.rssi.ru) как нижние оценки вероятностей ошибки на блок и на бит для сферической упаковки. Реальные возможности методов полного экспоненциально большого перебора при использовании ОД ещё более скромны.

Именно в это мгновение и встаёт во весь рост проблема эффективного и обязательно очень простого декодирования. Ясно, что на самом деле нужно ещё больше увеличить длины кодов и одновременно снизить сложность декодирования. Причём даже сложность порядка  $n^2$  вместо экспоненты от длины кода  $n$  будет слишком большой, поскольку для этого случая разница по числу операций декодирования по сравнению с линейной сложностью составит для кода длины 10'000 также порядка 10'000 раз. Именно отсюда и следует, что эффективными и одновременно перспективными для технических приложений могут стать только алгоритмы с практически линейной сложностью реализации от длины кода  $n$ .

Ну, а вообще-то это возможно, чтобы сложность достаточно эффективного метода была бы оценена как линейная? Да, при некоторых допущениях можно считать, что как раз сложность турбо декодеров и оказалась при весьма высоком уровне шума близкой к линейной зависимости от  $n$ . Это и стало началом триумфа турбо кодов в 1993 г.. Но иначе и быть не могло. Даже небольшое усложнение алгоритма для турбо декодеров по сравнению с линейным ростом числа операций на блок никогда не привело бы к десятилетней эйфории относительно возможностей турбо кодов.

Главная же проблема открытых в 1993 г. турбо декодеров состоит в следующем. При новом для западной науки подходе к декодированию как к итеративному однородному процессу (тогда это был безусловный прогресс) авторы этого метода заменили процесс сопоставления экспоненциально большого числа потенциальных решений (т. е. допустимых кодовых слов), например, в метрике Хемминга или Евклида на процедуры не прямых косвенных оценок расстояний. Эта замена оценок основных параметров на косвенные оказалась весьма дорогой платой. Вычисления были достаточно сложными по сравнению, например, с МПД, где расстояние определяется сразу при

вычислении синдрома и только один раз. Кроме того, такую совсем непростую итеративную процедуру требовалось проводить  $I=5 - 50$  раз. Это, конечно же, тоже не очень мало. А поскольку и итераций нужно достаточно много, и сами вычисления касались других, вторичных параметров кодов, которые оценивали истинную метрику только косвенно, число операций на бит для такой «оценки с подменой целей» оказалось в итоге достаточно большим.

Во многих случаях можно считать, что число операций на бит для турбо кодов при большом шуме составляет от нескольких тысяч до примерно 10'000 операций. Это, конечно, много меньше, чем если бы мы декодировали эти же или другие коды с одинаковой эффективностью на основе, скажем, алгоритма Витерби (АВ). Но в абсолютном выражении проблема турбо кодов – это, как и раньше, именно слишком большой объём вычислений!

«Как, - спросите вы, - ведь мы только что решили, что практически их сложность – минимальная, линейная от длины кода. Это значит, что при пересчёте на каждый бит данных сложность декодеров турбо кодов фактически не растёт вообще!».

Да. Всё так. Но вот коэффициент пропорциональности  $c$  в оценке числа операций на бит  $N=c*n$  из-за неточного выбора целевой итеративной функции и большого числа самих итераций турбо декодера оказывается всё же весьма и весьма большим.

Отметим также ещё раз, что некоторые вычисления в ряде модификаций турбо декодеров оказываются качественно довольно сложными. Другой во многом важный и труднопреодолимый недостаток турбо кодов заключается ещё и в том, что некоторые из алгоритмов этого класса очень плохо распараллеливаются. Последнее обстоятельство также весьма сильно ограничивает уровни реальной производительности турбо декодеров.

Все перечисленные обстоятельства и определяют то внимательное, но скептическое отношение многих зарубежных организаций и коммуникационных фирм к турбо алгоритмам. Мы тоже полагаем, что эти коды в принципе нельзя рассматривать как перспективные для современных высокоскоростных сетей связи.

#### **А что же многопороговые декодеры (МПД)?**

Тут ситуация принципиально другая. Сама суть алгоритмов МПД состоит в том, что они всегда могут точно определять расстояние текущей гипотезы декодера до принятого сообщения. Именно это и должен измерять оптимальный декодер (ОД). Значит, МПД сразу вычисляет именно те оценки, которые и нужны для точного определения наиболее правдоподобного слова. Далее, оказывается, что МПД также относится к итеративным алгоритмам

самого классического типа. Более того, этот метод появился в 1974 году, на 20 лет раньше алгоритмов для турбо кодов. За это время все возможности МПД были хорошо изучены и в настоящий момент эти методы разработаны для очень широкого множества различных каналов и сигналов, создавая хорошую конкуренцию турбо кодам, существенно опережая последние по быстродействию при сопоставимых уровнях эффективности.

Особая предпочтительность этого метода связана с несколькими факторами. Например, решения МПД строго сходятся к решению оптимального декодера (ОД) в гауссовских и в ряде других каналов. Этим уникальным свойством МПД не обладают никакие другие методы декодирования. Очень важно также то, что в МПД все вычисления исключительно просты по сравнению с операциями турбо декодеров. Многопороговые декодеры складывают и сравнивают между собой только небольшие целые числа. Это позволяет значительно повысить производительность МПД для многих приложений.

И, наконец, ещё одно очень существенное преимущество МПД (оно тоже воистину революционно для современной технологии) состоит в том, что распараллеливание операций в МПД – чрезвычайно простая для этого алгоритма процедура. Очень легко этого достичь, в частности, при аппаратной реализации, когда можно создать сколько угодно много маленьких процессоров, которые отдельно и независимо реализуют суммирование проверок на каждом пороговом элементе. А это и значит, что если пороговые элементы спроектированы в МПД как схемы мгновенного действия (ясно, что это вполне возможно), то такой МПД декодер имеет производительность, которая в 5-30 раз превышает скорость сдвига данных в регистрах этих декодеров. Если максимальная тактовая частота для регистров сдвига в каких-либо ПЛИС ( программируемых логических интегральных схемах) составляет порядка 30 МГц, то производительность построенного на этой ПЛИС МПД будет 300 Мбит/с и более.

Ничего подобного ожидать от турбо декодеров вообще нельзя. Их производительность оказывается, наоборот, иногда в 10 - 100 раз меньшей, чем тактовая частота кристалла, на котором они созданы. Этим и определяется то чрезвычайно большое преимущество, которое имеют по производительности декодеры МПД относительно турбо алгоритмов: МПД при аппаратной реализации при некоторых реалистических параметрах системы кодирования может быть иногда примерно в 1000 раз быстрее, чем турбо подобные алгоритмы с равной эффективностью по энергетике.

«А что же программные версии? - спросите вы. – Там таких преимуществ МПД относительно турбо декодеров, наверное, нет?»

Оказывается, наоборот, тип и порядок выполнения операций в программных версиях некоторых видов декодеров легко видоизменить так, что в результате можно получить значительное уменьшение числа необходимых преобразований принятых данных. Оно полностью соответствует по своей сути повышению быстродействия программного декодера, аналогичному распараллеливанию в их аппаратных реализациях. МПД как раз и относится к таким процедурам с хорошим итоговым результатом.

Как же это снижение вычислительных затрат достигается в МПД? После вычисления всех сумм проверок они затем достаточно редко изменяются на различных следующих итерациях декодирования. Проведённый нами анализ показал, как часто надо повторно вычислять новые суммы проверок, чтобы их неправильные старые значения мало ухудшали корректирующие возможности алгоритма. Выяснилось, что при правильной программной реализации такой «экономичной» многопороговой схемы декодирования среднее число операций на бит  $N$  в МПД может составлять всего  $N=c_1*d+c_2*I$ , где  $c_1$  и  $c_2$  - небольшие целые числа,  $d$  – кодовое расстояние,  $I$  – число итераций декодирования, причем  $d$  и  $I$  - как правило, не более 30. Эта оценка, успешно подтверждённая экспериментально, соответствует для типичных кодовых параметров разнице в числе операций  $N$  с турбо кодами, составляющей от 40 до 200 раз. Это неудивительно, если обратить внимание, что в «экономичной» версии МПД на каждую итерацию обычно тратится в среднем только несколько операций декодера с небольшими целыми числами, которые всегда относятся к самым быстрым в любом микропроцессоре.

Таким образом, обычно при программной реализации МПД действительно примерно на 2 порядка быстрее, чем прочие алгоритмы.

Кроме того, в ряде случаев правильное проектирование позволяет создавать аппаратные реализации МПД декодеров, которые почти в 1000 более производительны, чем другие методы коррекции ошибок для каналов связи с высоким уровнем шума.

**Желаем вам успехов!**  
**Ждите следующую группу ответов**  
**на ваши вопросы.**  
**Пишите нам! Спрашивайте.**  
**Мы отвечаем быстро.**