The answers to your problems No. 10 - 11

We ask our readers to pay their attention that many your problems are worthy of the analysis in the special separate book. They quite merit an widely organized debate.

In this connection some of our answers will be not always short, though the really steep discussion of problems, encompassed by you, demands sometimes many tens pages of a serious stuff.

If you are agree with us in this prominent aspect of discussion about a digital data high transmission accuracy methods, your judgements and new questions on the coding theory and engineering development will be the good basis for conversation of the authors with the readers of our web-site.

10. Where is the difference between correcting codes and methods of their decoding? Why is it a lot of them so?

If they will see an answer-example on a problem No. 3 (RS for codes), it is easy to note, that the project for an coding system contains up to three tens points, which one should be taken into account and consider. The cause is a simple one: the requirements to codes by input and output probabilities, and also on many other parameters can be very much and very miscellaneous. And in turn, an error correcting method selection, i.e. decoding algorithm, and its complexity hardly depends on it. The inexperience of the system customer can result in writing RS, which one will show that it is impossible to satisfy. For example, it is impossible to eliminate that its requirements on error probabilities and redundancy will correspond to return disparity R > C (code rate exceeds a channel capacity), when in general any encoders are ineffective. This situation is completely similar to a violation of law of energy conservation in mechanics.

But also it is not enough of it. The codes are necessary for error correction. Errors after digital data transmission can arise because of the very miscellaneous causes. But then they are described also by different models of errors generating. It results in specific methods of error correction, i.e. to new algorithms too.

Moreover, the miscellaneous systems of transmitted signals creation are possible also. They also are the cause of further changes of errors generating model. And if they will leave it out, outcome of a coding system "creation" by a principle "well, I want so " can be simply totally failure. An example of similar "achievement" we have already described in the answer to a problem No. 9. There cause of a failure was that the implementators have not made agreed a signals system and bounded with it model of channel errors with codes and methods of their decoding. The channel was binary, and codes - with the non-binary basis **q**. In that example the basis **q** was large enough, **q**=64 > > 1, that follow-up strengthened misalignment of algorithm and model of errors.

We shall mark, at last, that successes in construction, for example, of decoding algorithms for binary channels essentially help progression and methods of error correction for many other signals systems. (For example, for multipositional PSK, for a system such as APSK-16 - signals on a plane 4x4 and so on). There is nothing surprising in it, as many parameters of different code systems are interdependent.

Also it is received, that the availability of miscellaneous methods of decoding for different signals is normally. In particular, the algorithm MTD exists in different modifications: for binary, non-binary and erasure channels, for the Hamming and Euclid metrics, for circular PSK, for APSK-N and some other models. The experience of developments and applications of MTD testifies to its very simple adaptability to most (probably, almost to all) of signal systems and channel noise models.

And if for some concrete RS the different producers tender the completely different solutions, then it is necessary closely to look, whether there are really good versions among them. It is possible, that all of them demonstrate completely poor knowledge of the contractors in the field of high technologies of decoding. We hope, our readers already perceive, that it is not enough for such methods.

11. Why MTD is much simpler than turbo decoders?

Let's consider a problem in two stages. When we are not interested in a complexity problem, and we consider only efficiency, in this case optimum decoder (OD) for a code of length **n** with code rate **R** selects under the metrics of the Hamming or Euclid nearest to received vector **Q** the code word from an exponentially large set of all possible $2^{\mathbf{nR}}$ vectors. But even at **n**=1000 the selection nearest code vector to **Q** results in enough high probability of incorrect decoding. And you see it corresponds to review of such huge number $2^{\mathbf{nR}}=2^{\mathbf{500}}$ of code vectors, which one is even more than number of atoms in the Universe!

The concrete charts for the lower estimations of these probabilities are shown in all presentations of our web-site www.mtdbest.iki.rssi.ru as the lower estimations of probabilities of an error per block and per bit for spherical packaging. Actual capabilities of methods of full exponentially large exhaustive search at OD usage are else more weak.

In this instant also there is in all height a problem of effective and a very simple decoding. It is clear, that actually it is necessary to increase even more the lengths of codes and simultaneously to diminish the complexity of decoding. And even the complexity about $\mathbf{n}^2$ instead of an exponent from length of a code **n** will be too large, as for this case the difference on number of decoding operations as contrasted to by linear complexity will make for a code of length 10'000 also about 10'000 times. From here it also follows, that effective and simultaneously perspective for the technical appendices algorithms with practically linear implementation complexity of code length **n** can become useful only.

Well, and generally is it possible that the complexity of effective method would be estimated as linear? Yes, at some assumptions it is possible to count, that just complexity of turbo decoders and appeared at a rather high noise level a close to linear dependence of **n**. It also became the beginning of turbo codes triumph in 1993. But and it could not be differently. Even the small complicating of algorithm for turbo decoders as contrasted to by linear increase of operations number per the block would never result in ten years' euphoria concerning turbo codes capabilities.

But the main problem of turbo decoders discovered in 1993 looks something like that. At the approach, new to western science, to decoding as to iterative homogeneous process (then it was a unconditional advance) authors of this method have changed process of confrontation exponentially large number of the potential solutions, for example, in the metrics of the Hamming or Euclid (i.e. the permissible code words) by the procedures of indirect estimations of distances. This replacement of estimations of main specifications on indirect has appeared to be a rather expensive charge. The calculus were enough composite. Besides such absolutely not simple iterative procedure was required to be fulfilled I=5 - 50 times. It is, certainly, not so little enough too. And since iterations were necessary much enough, and estimations concerned other, secondary parameters of codes, which one estimated the true metrics only indirectly, the number of operations per bit for such "estimations with substitution of the purposes " has appeared large enough in the total.

In many cases it is possible to consider, that the number of operations per bit for turbo codes at a large noise makes from several thousand up to approximately 10'000 operations. It is much less certainly than if we would decode the same or other codes with identical efficiency on the basis, say, of Viterbi algorithm (VA). But in absolute expression the problem of turbo codes is, as before, large volume of calculuss too!

" Why, - you ask, - you see we have just decided, that practically their complexity is minimum, linear of code length **n**. It means, that at recalculation per each data bit the complexity of turbo decoders actually does not grow at all! ".

Yes, it is. Everything is quite right. But the quotient **c** in an estimation of operations number per bit **N=c*n** because of unperfect selection of a target iterative function and large number of iterations decoder turbo appears rather and rather large.

Note also, that some calculus in a number of turbo decoders modifications appear qualitatively rather composite. The other disadvantage of turbo codes is that some of algorithms of this class are very bad for multisequencing. Last circumstance strongly limits actual throughput of turbo decoders too.

All listed circumstances also determine that close, but skeptical attitude of many foreign organizations and communication corporations to turbo algorithms. We suppose also, that these codes basically cannot be esteemed as perspective for modern high-velocity communications networks.

And what about multithreshold decoders (MTD)?

Here situation is in essence other. The entity of MTD algorithms is that they always "know" distance to the received massage. It also should be measured by the optimum decoder (OD). Means, MTD at once calculates those parameters, which ones and are necessary to looking for the most verisimilar word. Further, it appears, that MTD also are an iterative algorithms of the most classic type. Moreover, this method has appeared in 1974, for 20 years before algorithms for turbo codes. For this time all capabilities of MTD were well studied and in the present moment these methods are designed for very broad set of different channels and signals, creating a good competition to turbo codes, essentially advancing last in the majority of the projects, which one can be actual.

The special preferability of this method is connected to the different factors. For example, the MTD's solutions strictly converge to the solution of the optimum decoder (OD) in Gaussian and in a number of other channels. This unique property of MTD is no with any other methods of decoding. It is very important also that in MTD all calculus are extremely simple as contrasted to operations of turbo decoders. The multithreshold decoders sum and compare among themselves only small integers. It accelerates throughput of MTD for many applications very much.

And, at last, also a very essential MTD advantage (it is truly revolutionary for modern technology also) is that multisequencing of operations in MTD - is an extremely simple procedure for this algorithm. It is easy to reach a good multisequencing particularly at a hardware representation, when it is possible to create as much as a lot of necessary small processors, which ones will realize summation of checks at each threshold switch separately. And it means, that if the threshold switches are designed in MTD as the schemes of instantaneous operating (clearly, it is quite possible to do), such the MTD decoder has productivity, which one at 5-30 of time exceeds data shift rate in the registers of these decoder. If the maximum clock rate for shift registers in any PLIC (programmed logical integrated circuits) is about 30 MHz, the throughput of MTD, constructed at this PLIC will be 300 Mbit/s or more.

It is impossible to expect anything similar from turbo decoders in general. Their throughput appears, on the contrary, sometimes in 10 - 100 times smaller, than clock rate of a chip, at which one they are built. It also determines that extremely large advantage, which one have on throughput the MTD decoders rather turbo of algorithms: MTD at a hardware representation with some realistic parameters of coding system can be sometimes approximately in 1000 times faster, than turbo similar algorithms with equal efficiency on code gain.

"And what about software versions? - you ask. - There are no such advantages of MTD before turbo decoders, probably? "

It appears, on the contrary, the kind and order of fulfilment of operations in software versions of some decoders is easy to alter so, that in outcome it is possible to receive considerable decreasing of number of indispensable transformations of the received data. It corresponds completely in the essence to throughput of the softwaare decoder similar to multisequencing in their hardware representations. MTD just also falls into such procedures with good total outcome.

How this decrease of computing requirements is reached in MTD? After calculus of all check sums they are changed seldom enough at different following iterations of decoding. The analysis, conducted by us, has shown, how often it is necessary repeatedly to calculate the new sums of checks, that their incorrect older values degraded correcting capabilities of algorithm a little only. It appeared that at exact software implementation of such "economic" multithreshold decoding circuit the average numberof operations per bit **N** in MTD can make only $\mathbf{N=c_1*d+c_2*I}$, where $\mathbf{c_1}$ and $\mathbf{c_2}$ - small integers, **d** - a code distance, **I** - number of iterations of decoding, **d, I** - as a rule, no more than 30. This estimation, successfully affirmed experimentally, corresponds for representative code parameters to a difference on operations **N** with turbo codes, that is from 40 up to 200 times. It is no wonder if to pay attention, that in the "economic" version of MTD at each iteration only a few operations of the decoder with small integers are usually spent on the average, which one always fall into with most fast in any microprocessor.

Thus, it is customary at software implementation **MTD really approximately in 100 times faster,** than other algorithms.

Besides the exact designing allows to create **hardware MTD representations, which ones are almost in 1000 times more efficient in throughput**, than other methods of error correction for communication channels with high level of noise.

Best wishes !
Wait following group of the answers
on your problems.
Write to us. Ask us!
We answer quickly.