Concatenated Convolutional Codes with Interleavers

Sergio Benedetto and Guido Montorsi, CERCOM — Politecnico di Torino
Dariush Divsalar, Sequoia Communications

ABSTRACT

This article presents a tutorial overview of the class of concatenated convolutional codes with interleavers, also known as turbo-like codes. They are powerful codes, formed by a number of encoders connected through interleavers, endowed by a decoding algorithm that splits the decoding burden into separate decoding of each individual code. Refinement of successive estimates of the information sequence is obtained by iterating the procedure of passing from one decoder to the other likelihood information decorrelated by the interleaver action. The key issues of code analysis and design are covered at the level of broad comprehension, without paying attention to analytical details.

INTRODUCTION

Channel coding has become an indispensable tool in modern communication systems dominated by stringent power and bandwidth constraints. As made clear by Shannon back in 1948 [1], large coding gains (defined as the difference in the signal energy between the uncoded and coded systems required to achieve a given error probability) for a given spectral efficiency[1] can be obtained by encoding the information sequence in large blocks. The largest possible coding gain for a given rate and very large blocks stems from the capacity limit derived by Shannon. In Fig. 1 the bit error probability is plotted vs. the ratio between the energy per information bit E_b and the Gaussian noise power spectral density N_0 for several spectral efficiencies assuming very large blocks and capacity-achieving codes. The curves become vertical for bit error probability roughly below 10^{-4}, and their intercept with the abscissa constitutes the so-called Shannon limit, that is, the minimum E_b/N_0 yielding a bit error probability as low as desired.

For about five decades coding theorists have been looking in vain for codes capable of approaching the Shannon limit. From this unsuccessful search stems the folk theorem: *all codes are good, except those that we know of*. The story is more complicated though, since finding good codes is a simple task. Indeed, randomly generated codes with large block sizes will be very good with high probability. The problem lies in the fact that while encoding is always a rather simple task, the decoding complexity for a randomly generated code increases exponentially with the block size, and thus quickly becomes unmanageable. Thus, the previously mentioned theorem should be rephrased as: *all codes are good, except those that we know how to decode*. The history of the past 50 years of coding theory is about the struggle of conjugating a great structure in the code architecture to facilitate decoding with a random look in the code words distribution to enhance the code strength according to the message implicit in the proof of Shannon coding theorem.

An important step in the right direction was made by Dave Forney with his thesis work on concatenated codes [2]. Instead of making the code more and more complex in the search for larger gains, he proposed to cascade relatively simple codes (in practice, two codes, *inner* and *outer*) to obtain a powerful overall code for which the decoding complexity increased only algebraically with the block size.

Concatenated codes have become a de facto standard in many communication systems. The by far most popular structure is the cascade of an outer algebraic code, typically a Reed-Solomon code, with an inner convolutional code. The rationale is to face the "bad" channel with a code for which soft decoding

This work has been partially supported by Qualcomm, Inc.

[1] *The spectral efficiency, measured in bits per second per hertz, is defined here as the ratio between the information rate in bits per second and the bandwidth in hertz. For the bandwidth, we adopt the Shannon definition of half the product of number of channel signals per second times the number of dimensions of the signal space.*

0163-6804/03/$17.00 © 2003 IEEE

(based on the received sufficient statistics and thus better than hard decoding) is simple through the Viterbi algorithm, and then present to the outer decoder a cleaner binary channel suitable for its powerful hard decoding capabilities.

Unprecedented results very close to Shannon limit (0.5 dB) were presented in [3]. The authors of the article introduced for the first time a structure (called *turbo code* by them) in which previously adopted ingredients (two encoders and one interleaver) were organized as the parallel concatenated convolutional code (PCCC) shown in Fig. 2.

It consists of two *constituent* convolutional encoders (CEs) fed by the same information bits, in different order, as those at the input of the lower encoder are the permuted (by the interleaver) version of those entering the upper encoder. The two encoders are generally *terminated* (i.e., transformed into block codes) by appending a suitable number of dummy bits at the end of the information word to drive the encoder trellis to the zero state.

For linear CEs, the overall encoder of Fig. 2 is then a linear terminated convolutional code, and, as such, could be maximum-likelihood (ML) decoded using the Viterbi algorithm. The number of states of the encoder varies with time, and for properly designed codes reaches a value on the order of $2^{v1+v2+N}$, where 2^{v1} and 2^{v2} are the number of states of the constituent encoders, and N is the information block size, or, equivalently, the interleaver size. For the large N used in practice, the ML decoding complexity, which is proportional to the number of trellis states, soon becomes unaffordable.

The solution is to break this large complexity by means of a local decoding algorithm working on each individual CE, with a complexity on the order of $2^{v1} + 2^{v2}$, and to approach ML performance by letting each decoder take advantage of the progress made by the other through the recirculation of an enhanced reliability measure about the information symbols. The information passed from one decoder to the other is the so-called *extrinsic* information, that is, the improvement to the knowledge of each information symbol evaluated by each decoder on the basis of the statistical dependence of the coded symbols through the encoder trellis.

The decoding algorithm has important impact on what each constituent decoder should do. Indeed, since the extrinsic information output by each decoder must pass through the interleaver/deinterleaver, it has to be a quantity associated with each symbol, so sequence decoding algorithms like Viterbi decoding cannot be directly applied. Rather, one needs an algorithm capable of computing the *a posteriori* probability (APP) of *each* symbol based on the knowledge of the whole received sequence of the soft values forming sufficient statistics.

An algorithm like this was already available [4], waiting since 1974 for practical application. It is the BCJR (acronym from the authors' names) algorithm, which computes the APP of code and information symbols through two recursions on the code trellis, one in the forward and the other in the backward direction, with a computational complexity per decoded symbol that is independent of block size and increases linearly with the number of trellis states. The BCJR algorithm is between two and three times more complex than the Viterbi algorithm.

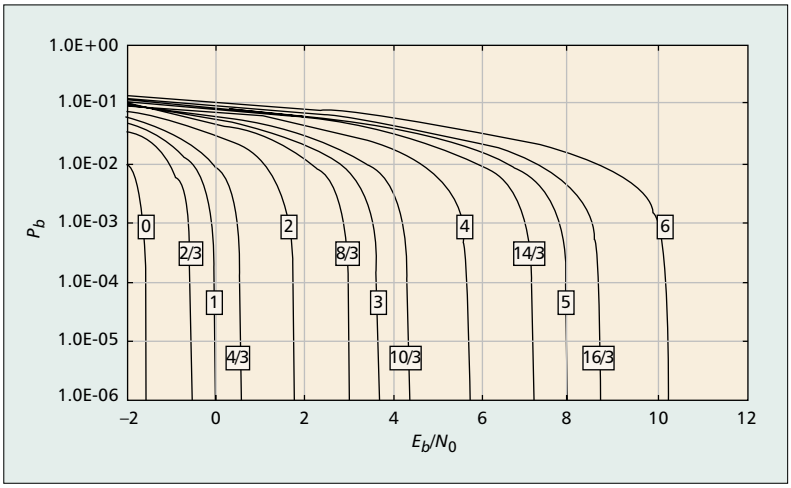

Figure 1. *A plot of the points (P_b, E_b/N_0) satisfying the capacity limits of the unconstrained additive white Gaussian noise channel for different spectral efficiencies.*

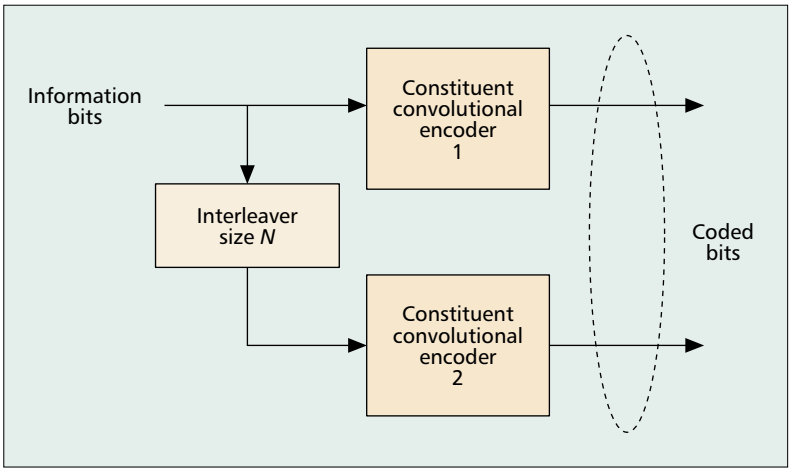

Figure 2. *A block diagram of a parallel concatenated convolutional code.*

The code structure of Fig. 2, endowed with the iterative decoding algorithm sketched above, meets nicely the two requirements of the Shannon theorem for a good code. In fact, it presents a certain degree of randomness offered by the interleaver (indeed, highly structured interleavers like row-column interleavers yield poor performance), and has a decoding complexity per decoded bit independent of block size. It should then come as no surprise that the performance of these codes closely approaches the Shannon limit.

In the following, we will guide the reader through an unpretentious tutorial tour inside the realm of concatenated codes with interleavers, aiming at heuristic explanations of the reasons for their astonishing behavior, and the main tools for the code design.

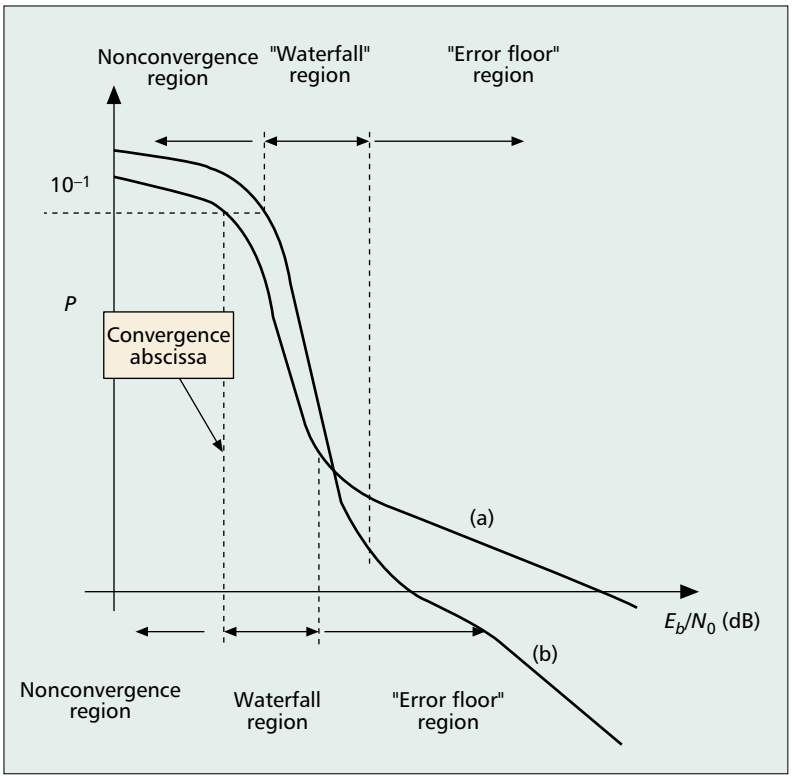

Figure 3. *Qualitative behavior of the error probability vs.* E_b/N_0 *for concatenated codes with interleavers under iterative decoding.*

Analysis and Design of Concatenated Codes with Interleavers

The error probability performance of concatenated codes with interleavers under iterative decoding is invariably represented by curves like those depicted in Fig. 3. We can identify three different regions for increasing values of E_b/N_0. The first one is the *nonconvergence* region, where the error probability keeps high nearly constant values. At a certain point, the *convergence abscissa*, the curves start a rather steep descent to medium-low values of the error probability (the *waterfall region*). Finally, in the third region (the *error floor* region), the slope of the curves decreases significantly, and performance improvements incur significant additional energy expenses. The waterfall region is dominated by the *interleaver gain* (a phenomenon to be explained in the following sections), whereas the error floor region is dictated by the minimum distance of the code. In the first region, the interleaver acts mainly through its size, whereas in the second the kind of interleaver plays a dominant role.

Any attempt to design codes like the one in Fig. 2 as a whole leads to discouraging results because of its complexity. So far, only two successful design techniques have been proposed in the literature: the first relies on maximum likelihood (ML) decoding, and leads to design rules for the constituent encoders that work nicely for medium-low error probabilities assuming a purely random interleaver. After designing the constituent encoders, one can improve the code performance by a cut and trial approach to interleaver design.

While the first approach [5, 6] is a purely analytical one, the second requires the separate simulation of the behavior of each constituent encoder, and is based on the *probability density evolution* technique (see [7–9] or the so-called *EXIT charts* [10]). This second approach leads to codes that exhibit good performance in terms of convergence abscissa, and are suited to medium-high values of the error probability.

The two design approaches converge to some common criteria, but lead in general to slightly different code designs; typically, the codes designed (or, better, found) through the second technique show faster convergence of the iterative decoding algorithm (a lower value of the converging abscissa), but reach more quickly the error floor (curve a, Fig. 3). The opposite is true for codes designed using the first technique (curve b, Fig. 3). As a consequence, the choice depends on the quality of service requirements of the system at hand. In the following, we provide a succinct survey of the two techniques with some examples.

Maximum-Likelihood Analysis and Design of PCCCs

The performance of a PCCC like that in Fig. 2 can be estimated via Monte Carlo simulation of the iterative decoding algorithm. As is easily understood, this gives no insight into the "how" and "why" of code behavior, and is of little help in code design, which should answer the questions of how to choose the constituent codes and interleaver. Moreover, it does not show how close the suboptimum iterative decoding can be to ML decoding, an important issue that requires the ability to estimate ML performance analytically.

To understand the mysteries of PCCC behavior, it is important to distinguish between codes and encoders. A code is simply the set of code words, and characterizes the output of the encoder, without any reference to the input information words. An encoder, instead, is described by the ordered set of information and code word pairs, so each code word is associated with the information word that has generated it through the encoding law.

Of the two most important performance measures of a code, the word error probability (WEP), the average probability that the decoder chooses a code word different from the transmitted one, only depends on the code, whereas the bit error probability (BEP), the average probability that the decoder makes a mistake in delivering an information bit, depends on the encoder. The fact that a given code admits many different encoders (i.e., different mappings between information words and code words) makes it evident that the minimization of the BEP requires a proper design of the input-output characteristics of the encoder. A complete description of an (n, k) encoder (k being the size of the input informa-

tion word and n the code word size) passes through the knowledge of the input-output coefficients $A_{w,d}$, which represent the number of code words with Hamming weight (number of bits equal to one) d generated by information words of weight w. Obviously, d varies from 0 to n and w from 0 to k. Upper bounds to the BEP, like the union bound, are based on the input-output coefficients. In particular, for the case of an additive white Gaussian noise channel, binary antipodal modulation and ML soft decoding, the union bound is

$$P_b \le \frac{1}{2}\sum_{w=1}^{k}\frac{w}{k}\sum_{d=d_{\min}}^{n} A_{w,d}\,\mathrm{erfc}\left(\sqrt{\frac{d r_c E_b}{N_0}}\right), \quad (1)$$

where P_b is the BEP, and $r_c = k/n$ the code rate.

Traditionally, a good code had been defined as a code with a set of large weights d (relatively to their code word block size), and, in particular, with a large *minimum distance* $d_{\min} = \min d,\ d \neq 0$, which determines the asymptotic performance for large E_b/N_0. Looking at Eq. 1 we see that the bit error probability can be reduced, even in the presence of a small minimum distance, by acting on the coefficients $A_{w,d}$. This is precisely what concatenated codes with interleavers do, as Dave Forney pointed out in his Shannon Lecture [11]: *rather than attacking error exponents, they (turbo codes) attack multiplicities, turning conventional wisdom on its head*.

To upper bound the ML BEP of a PCCC according to Eq. 1 we need to know its input-output coefficients, assuming knowledge of those of the CEs. For large interleavers such as those used in practice, computing the input-output coefficients of the overall encoder is a task with overwhelming complexity. The only way out of it is to follow the path indicated by Shannon in his proof of the channel coding theorem. In the case at hand, this means replacing the actual PCCC with the set of PCCCs employing the same CEs and all possible interleavers with a given size [5]. This move drastically simplifies the analysis, and yields the upper bound to the bit error probability of a code averaged with respect to the set of all interleavers. In practice, the performance of this average encoder can be approached by using actual interleavers generated by purely random permutations.

The "average" approach to the PCCC ML analysis proves to be a very useful tool for the design of the CEs as well, since it allows for splitting the overall PCCC design into two steps: first, the choice of CEs "optimized" for the average interleaver; second, the design of the interleaver for the chosen CEs. Referring the reader seeking mathematical rigor to [5, 6], we will now offer a heuristic and unpretentious insight into the CEs design.

In a concatenated code with interleaver, the role of the interleaver is twofold:

- First, it aims at improving the code strength by increasing the Hamming weight of code words. As an example, in the encoder of Fig. 2 the interleaver should be able to permute the information word from the input to the upper encoder to that of the lower encoder to associate a weak (low weight) upper code word with a strong (large weight) lower code word. In this respect, the interleaver improves the ML performance, which depends on the code weight spectrum.
- The second function of the interleaver is to spread the outputs from one decoder to provide the other decoders with loosely correlated inputs. This improves the behavior of the iterative decoding algorithm.

Referring to the PCCC of Fig. 2, let us consider an information word with $k = N$ bits and (Hamming) weight w that generates through the upper encoder a code word with a low weight d, corresponding to a short *error event* in the code trellis (an error event is a path leaving the all-zero state in the trellis and merging into it later). A "good" interleaver should be able to permute the position of the 1s in the information word to break the error event and generate a larger weight code word in the second encoder. The probability of finding a good interleaver by choosing it at random in the set of all distinct permutations of w objects in N positions can be estimated as 1 minus the ratio between the number of bad interleavers and the cardinality of the set of all permutations, yielding

$$P_{\text{good|w}} = 1 - \frac{N}{\binom{N}{w}} \simeq 1 - \frac{N}{N^w} = 1 - N^{1-w}, \quad (2)$$

where N is the number of bad interleavers, that is, those which rigidly shift the w ones without altering the relative positions, thus keeping the error event in the lower encoder (we neglect boundary effects for simplicity), and $\binom{N}{w}$ is the number of distinct permutations.

Equation 2 shows that the probability of randomly choosing a good interleaver increases with w and N, so the most critical information words (i.e., those for which the interleaver will find it difficult to yield through the encoders large code words weights) are those with lower weight; in particular, the information words able to generate an error event in the CEs with minimum w, $w_{\min}$. If $w_{\min} = 1$, the probability of Eq. 2 is 0 independent of N. This happens for feed forward CEs, represented as a finite impulse response filter, for which an input word with weight 1 always generates an error event due to the finiteness of the impulse response. On the other hand, choosing recursive convolutional encoders as CEs, one obtains $w_{\min} = 2$, due to the infinite impulse response. In this case, the probability of Eq. 2 increases with N as $1 - 1/N$. The mathematical analysis in [5, 6] shows that indeed the BEP with ML decoding decreases with the input block size as $1/N$, a phenomenon known as *interleaving gain*, provided that both CEs are recursive.

From the previous heuristic explanation, we also learn that the most likely error events in PCCC code words will be those generated by information words of decreasing weight w. As a consequence, the optimization of the

The "average" approach to the PCCC ML analysis proves to be a very useful tool also for the design of the CEs, since it allows for the splitting of the overall PCCC design into two steps.

Beyond offering better performance with respect to PCCCs in terms of the error floor owing to the larger interleaver gain, the SCCC structure and its iterative decoding can encompass other systems that can be interpreted as serial concatenations of individual modules.

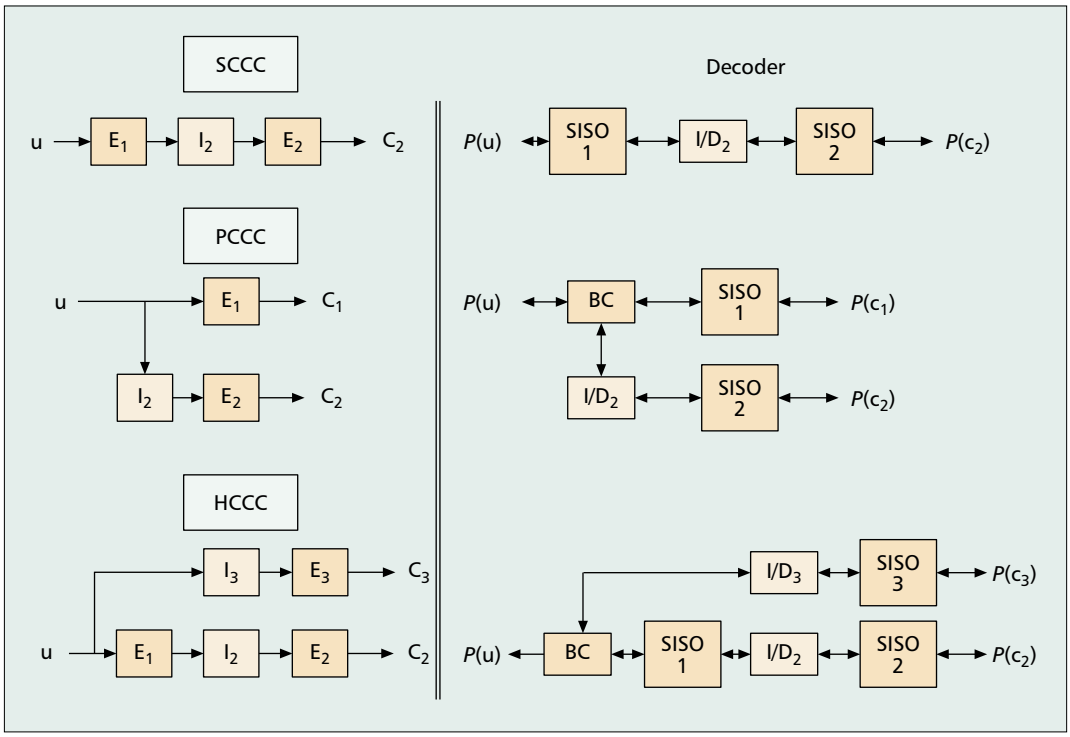

Figure 4. *Serial, parallel, and hybrid concatenated encoders and decoders. The label BC stands for broadcaster: the decoder counterpart of the one-to-two replication in the encoder.*

CEs should be aimed at maximizing the weights of code words generated, in decreasing order of importance, by information words with weight $w = 2, 3, 4, \ldots$. Once again, we notice that the classical optimality parameter for the construction of good convolutional codes (i.e., the code word weights starting from the minimum one called *free distance*) is replaced here by an optimality criterion involving the input-output characteristics of the encoder (i.e., the maximization of output weights generated by low input weights). In particular, the free distance is replaced by the so-called *effective* free distance, which is defined as the minimum weight of code words generated by information words with weight 2. Tables of CEs optimized according to the new criteria can be found in [12].

Other Forms of Code Concatenation

Besides PCCCs, other ways of concatenating encoders and interleavers are possible. The concept of a *code network* that generalizes that of concatenated codes by employing q encoders and $q - 1$ interleavers connected in various ways was introduced in [13]. Three examples of code networks are shown in Fig. 4, where **u** and **c** refer to information and code words, respectively. The first is the *serial* concatenation [14] of two convolutional encoderss (SCCC) separated by an interleaver acting on the code words of the outer code, the second is the original turbo code structure described above, and the third is a hybrid concatenation (HCCC) with $q = 3$.

Code networks are endowed with a suboptimum *distributed iterative* decoding algorithm, which generalizes the one previously described, whose complexity roughly amounts to the sum of the decoding complexities of each constituent encoder. The decoding algorithm is based on the replacement of each element of the code network by a suitably defined *soft-input soft-output* (SISO) counterpart [13], which receives reliability information from the rest of the network and broadcasts to the network an internally upgraded version of this information. In Fig. 4 we show the distributed decoders for the three code networks previously described. In it, $P(\cdot)$ represents suitably defined soft information of the corresponding symbol in the code network.

Union bounds to the error probabilities and the design tool based on ML performance can be extended to code networks, based on averaging with respect to the $q - 1$ interleavers. An important case is constituted by the SCCC. The design rules for it require that the inner encoder must be recursive, in which case the interleaver gain becomes equal to

$$N^{-\left\lfloor \frac{d^{o}_{\text{free}}+1}{2} \right\rfloor}, \tag{3}$$

where d^{o}_{free} is the free distance of the outer code. The interleaver gain of an SCCC can be significantly greater than for PCCCs.

Beyond offering better performance with respect to PCCCs in terms of the error floor due to the larger interleaver gain, the SCCC structure and its iterative decoding can encompass other systems that can be interpreted as serial concatenations of individual modules. Important examples refer to the cascade of an outer encoder and an inner modulator separated by an interleaver, to the

concatenation of an outer encoder, interleaver, and a magnetic recording channel, to coded continuous-phase modulations, and to the coded multiuser interference channel. Detailed explanations of such systems can be found in [15].

The Density Evolution Analysis and Design

Consider a PCCC or an SCCC as shown in Fig. 4. Iterative decoders for these codes are based on two SISO modules. The iterative decoder for either code construction can be viewed as a nonlinear dynamical feedback system in which extrinsic information sequences are passed from one constituent decoder to the other.

In a code using an ideal interleaver (i.e., a random interleaver with size going to infinity), the extrinsic information becomes independent and identically distributed random variables, with probability density function that can be approximated by a Gaussian probability distribution function (pdf) [8] whose expression depends only on two parameters, its mean μ and variance σ^2.

When the iterative decoding algorithm works properly, the extrinsic information improves as the iterations evolve. This improvement can be quantitatively measured in several ways [7, 8, 10]. A simple and intuitive one is given by its *signal-to-noise ratio* (SNR) defined as SNR $\triangleq \mu^2/\sigma^2$. This is equivalent to viewing each SISO as a nonlinear device with negative noise figure, yielding an output SNR larger than the input one.

Consider the input and output SNRs for each decoder, denoted $SNR1_{in}$, $SNR1_{out}$, $SNR2_{in}$, $SNR2_{out}$. A nonzero E_b/N_0 from the channel enables decoder 1 to produce a nonzero $SNR1_{out}$ for the output extrinsic information even though the initial $SNR1_{in}$ is 0. For a given value of E_b/N_0, the output SNR of each fecoder is a nonlinear function of its input SNR, denoted G_1 for decoder 1 and G_2 for decoder 2.

The functions G_1 and G_2 can be empirically evaluated by independent Monte Carlo simulations of the two SISOs fed by appropriate Gaussian random variables. Plotting G_1 and G_2^{-1} as shown in Fig. 5, we can analyze the decoder convergence by tracking the evolution of the extrinsic information's SNR from half-iteration to half-iteration.

The results refer to a rate 1/3 PCCC consisting of the two 16-state systematic recursive convolutional encoders shown in the figure. The upper curve corresponds to the input-output function G_1 for decoder 1, and the lower curve corresponds to G_2^{-1} for decoder 2.

Figure 5 graphically shows the progress of the decoder's iterations. The improvement in the SNR of the extrinsic information, and the corresponding improvement in the decoder's bit error rate, follows a staircase path reflecting at right angles between the curves corresponding to G_1 and G_2^{-1}. The steps in this staircase are large when the bounding curves are far apart, and small when they are close together. Where the curves become closer, the improvement in bit error probability slows down, as many iterations are required to get out of the narrow iterative decoding *tunnel* between the curves. When the iterative decoder successfully passes through the tunnel, convergence becomes very rapid as the two curves get farther apart at higher SNRs. This means that the bit error probability approaches zero as the number of iterations increases.

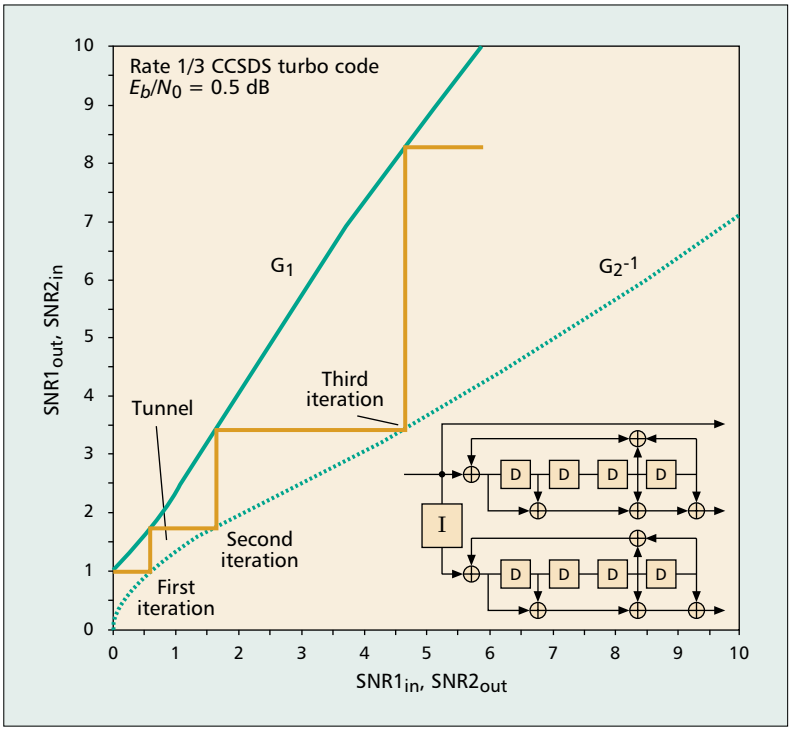

Figure 5. *Iterations and convergence of an iterative turbo decoder using recursive CEs.*

The initial displacement of the G_1 curve for $SNR1_{in} = 0$ is dependent on the E_b/N_0 due to the channel observations. If we reduce E_b/N_0 from the value of 0.5 dB used in Fig. 5, at some point the two curves will just touch each other. That value of E_b/N_0 represents the iterative decoding convergence abscissa of Fig. 3.

Performance and Applications

Using concatenated codes with interleaver in one of their various forms, keeping the same constituent encoders with interleavers of different sizes and modulators with increasing cardinality, we can obtain codes relatively simple to decode that lie within less than 1 dB[2] from the theoretical limits in a theoretically unlimited range of spectral efficiencies and block sizes. This points to one of the main characteristic of turbo-like codes that is sometimes overlooked: a *versatility* that has never been seen in any of the code classes discovered in the previous 50 years of coding theory.

As an example, in Fig. 6 we plot for various codes the code *imperfectness*, that is, the difference in dB between the SNR required to obtain a word error probability of 10^{-4} and the minimum SNR dictated by information theory vs. the information word size. The curves labeled

[2] *Actually, using a low-density parity check code with huge block size, it has been possible to approach the Shannon limit within 0.0045 dB [16].*

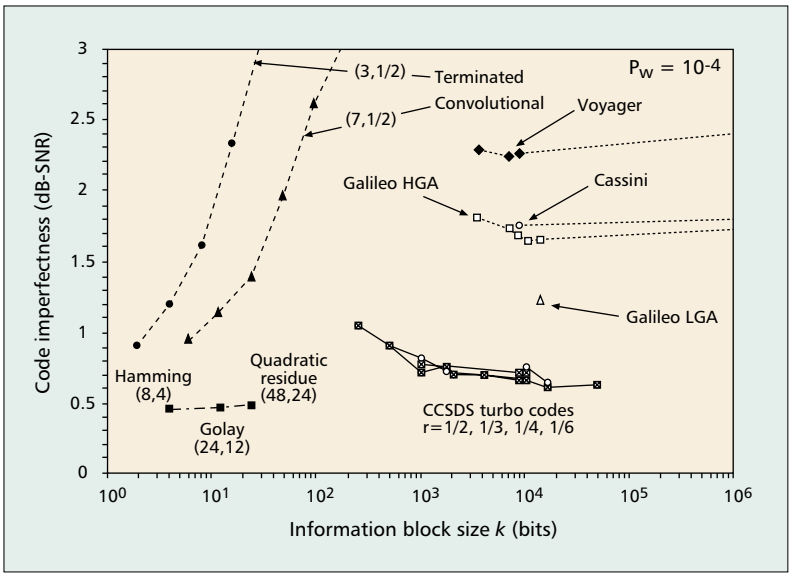

Figure 6. *Code imperfectness vs. code word size for various codes.*

CCSDS refer to the PCCCs standardized by the Consultative Committee for Space Data Systems.

The figure shows that PCCCs yields code imperfectness lower than 1 dB for block sizes ranging from a few hundred to several tens of thousand.

As described in a companion article in this issue [17], turbo-like codes have already made their way into practical applications and standards. This has occurred in spite of the fact that some unexplained features still remain in the behavior of the decoding algorithm. Indeed, although satisfactory from the point of view of engineers, more attentive to performance than to theoretical subtleties, the iterative decoding algorithm underlying the behavior of concatenated codes with interleavers (and low density parity check codes as well) still suffers from a lack of theoretical foundation. The explanation of its performance stems from heuristic arguments rather than exact theoretical results. Some important steps toward a more rigorous framework to understand the algorithm behavior have been moved recently. In [18] a geometrical interpretation of the turbo decoding algorithm has been presented, based on its formalization as a discrete-time nonlinear dynamic system, that sheds some light on the relationship between ML and iterative decoding. In the article the existence of fixed points of the decoding algorithm for a class of turbo codes has been shown, together with a set of sufficient conditions for their uniqueness and stability. An interesting follow-up describing a bifurcation analysis approach of the iterative decoding process as a dynamic system parameterized by SNR has recently appeared in [19]. However, there remain fundamental questions still unanswered, such as: Under which conditions does the iterative decoding algorithm admit a single fixed point? When there are multiple fixed points (or more generally multiple attractors, each characterized by its basin of attraction in the state space), what are the corresponding basin boundaries? What are the stability properties of different fixed points, and how do they bifurcate when the system parameters are varied? How are fixed points related to ML decoded code words?

Conclusions

A short walk into the thick jungle of results concerning turbo-like codes was the scope of this article. The approach has been necessarily tutorial, without derivations and mathematical subtleties. Our desire was to instill in readers an interest in this fascinating field that has profoundly renovated coding theory. The accurately selected bibliography (there are today more than 1000 papers on the subject, and we apologize to the authors of the good papers that were not included in the list) should provide enough "water" to quench the thirst of those who would like to delve deeper into the subject.

References

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Sys. Tech. J.*, Pts. I and II, July and Oct. 1948, pp. 379–423, pp. 623–56.
[2] G.D. Forney, Jr., *Concatenated Codes*, MIT Press, 1966.
[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," *ICC '93*, Geneva, Switzerland, May 1993.
[4] L.R. Bahl *et al.*, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Info. Theory*, vol. 2, no. 2, Mar. 1974, pp. 284–87.
[5] S. Benedetto and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Trans. Info. Theory*, vol. 42, no. 2, Mar. 1996, pp. 409–28.
[6] S. Benedetto and G. Montorsi, "Design of Parallel Concatenated Convolutional Codes," *IEEE Trans. Commun.*, vol. 44, no. 5, May 1996, pp. 591–600.
[7] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative Turbo Decoder Analysis based on Density Evolution," *IEEE JSAC*, vol. 19, no. 5, May 2001, pp. 891–907.
[8] H. El Gamal and R. Hammons, "Analyzing the Turbo Decoder using the Gaussian Approximation," *IEEE Trans. Info. Theory*, vol. 47, no. 2, Feb. 2001, pp. 671–86.
[9] T. Richardson and R. Urbanke, "The Capacity of Low Density Parity Check Codes Under Message Passing Decoding," *IEEE Trans. Info. Theory*, vol. 47, no. 2, Feb. 2001, pp. 599–618.
[10] S. ten Brink, "Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes," *IEEE Trans. Commun.*, vol. 49, Oct. 2001, pp. 1727–37.
[11] G. D. Forney, Jr., "Shannon Lecture," *1995 Int'l. Symp. Info. Theory*, Whistler, CA, June 1995.
[12] S. Benedetto, R. Garello, and G. Montorsi, "A Search for Good Convolutional Codes to be Used in the Construction of Turbo Codes," *IEEE Trans. Commun.*, vol. 46, no. 9, Sep. 1998, pp. 1101–05.
[13] S. Benedetto *et al.*, "Soft-input Soft-output Modules for the Construction and Distributed Iterative Decoding of Code Networks," *Euro. Trans. Telecommun.*, vol. 9, no. 2, Mar. 1998, pp. 155–72.
[14] S. Benedetto *et al.*, "Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding," *IEEE Trans. Info. Theory*, vol. 44, no. 3, May 1998, pp. 909–26.
[15] S. Benedetto and G. Montorsi, "Serially Concatenated Codes and Iterative Decoding," *Wiley Encyclopedia of Telecommunications* John G. Proakis,Ed., Wiley, 2002.
[16] S. Y. Chung *et al.*, "On the Design of Low-Density Parity-Check Codes Within 0.0045 dB of the Shannon Limit," *IEEE Commun. Lett.*, vol. 5, Feb. 2001, pp. 58–60.
[17] C. Berrou, "The Ten-Year-Old Turbo Codes are Entering into Service," *IEEE Commun. Mag.*, this issue.
[18] T. Richardson, "The Geometry of Turbo-Decoding Dynamics," *IEEE Trans. Info. Theory*, vol. 46, no. 1, Jan. 2000, pp. 9–23.
[19] D. Agrawal and A. Vardy, "The Turbo Decoding Algorithm and Its Phase Trajectories," *IEEE Trans. Info. Theory*, vol. 47, no. 2, Feb. 2001, pp. 699–722.

Additional Reading

[1] L. C. Perez, J. Seghers, and D. J. Costello, Jr., "A Distance Spectrum Interpretation of Turbo Codes," *IEEE Trans. Info. Theory*, vol. 42, no. 6, Nov. 1996, pp.1698–1709.

Biographies

SERGIO BENEDETTO [F] (benedetto@polito.it) received a Laurea in Ingegneria Elettronica (summa cum laude) from Politecnico di Torino, Italy, in 1969. From 1970 to 1979 he was with the Istituto di Elettronica e Telecomunicazioni, first as a research engineer, then as an associate professor. In 1980 he was made a professor in radio communications at the Università di Bari. In 1981 he rejoined Politecnico di Torino as a professor of data transmission theory in the Dipartimento di Elettronica. He spent nine months in 1980–1981 at the System Science Department of University of California, Los Angeles, as a visiting professor, and three months at the University of Canterbury, New Zealand, as an Erskine Fellow. He has coauthored two books in signal theory and probability and random variables (in Italian), and the books *Digital Transmission Theory* (Prentice-Hall, 1987) and *Optical Fiber Communications Systems* (Artech House, 1996), and *Principles of Digital Transmission with Wireless Applications* (Kluwer-Plenum, 1999) as well as about 200 papers for leading engineering journals and conferences. He has been Area Editor for Signal Design, Modulation and Detection for the *IEEE Transactions on Communications*. Active in the field of digital transmission systems since 1970, his current interests are in the field of optical fiber communications systems, performance evaluation and simulation of digital communication systems, trellis coded modulation, and concatenated coding schemes.

DARIUSH DIVSALAR [S'76,M'78,SM'90, F'97] received a Ph.D. degree from the University of California Los Angeles (UCLA) in 1978. Since 1978 he has been with Jet Propulsion Laboratory, California Institute of Technology (Caltech), Pasadena, where he is a senior research scientist. He has been working on developing state-of-the-art technology for advanced deep space communications systems for future NASA space exploration. His areas of interest are coding, digital modulation, and particularly turbo codes. He also taught digital communications and coding at UCLA from 1986 to 1996 and at Caltech since 1997. He has published over 100 papers, coauthored a book entitled *An Introduction to Trellis Coded Modulation with Applications* (MacMillan, 1991), and holds seven U.S. patents in the above areas. He is a corecipient of the 1986 Prize Paper Award in Communications for *IEEE Transactions on Vehicular Technology*. He has received over 20 NASA Tech Brief awards and a NASA Exceptional Engineering Achievement Medal in 1996. He served as Editor and Area Editor in Coding and Communication Theory for IEEE Transactions on Communications from1989 to 1996.

GUIDO MONTORSI (montorsi@polito.it) received a Laurea in Ingegneria Elettronica in 1990 from Politecnico di Torino, Italy, with a Master thesis, developed at the RAI Research Center, Turin. In 1992 he spent the year as visiting scholar in the Department of Electrical Engineering at the Rensselaer Polytechnic Institute, Troy, New York. In 1994 he received a Ph.D. degree in telecommunications from the Dipartimento di Elettronica of Politecnico di Torino. In December 1997 he became assistant professor at the Politecnico di Torino. In July 2001 he became associate professor. In 2001-2002 he spent one year in the startup company Sequoia Communications working on the innovative design and implementation of a third-generation WCDMA receiver. He is an author of more than 100 papers published on internationl journal and conference proceedings. His interests are in the area of channel coding and wireless communications, particularly on the analysis and design of concatenated coding schemes and study of iterative decoding strategies.

Turbo-like codes have already made their way into practical applications and standards. This has occurred in spite of the fact that some unexplained features still remain in the behavior of the decoding algorithm.