On the Asymptotic Input-Output Weight Distributions and Thresholds of Convolutional and Turbo-Like Codes

Igal Sason Emre Telatar Rüdiger Urbanke

EPFL – Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland
E-mails: {Igal.Sason, Emre.Telatar, Rudiger.Urbanke}@epfl.ch

December 1, 2002

Abstract

We present a general method for computing the asymptotic input-output weight distribution of convolutional encoders. In some instances, one can derive explicit analytic expressions. In general though, to determine the growth rate of the input-output weight distribution for a particular normalized input weight κ and output weight ω, a system of polynomial equations has to be solved. This method is then used to determine the asymptotic weight distribution of various concatenated code ensembles and to derive lower bounds on the thresholds of these ensembles under maximum likelihood decoding.

Index Terms: Convolutional codes, maximum-likelihood decoding, serially concatenated codes, thresholds, turbo codes, uniform interleaver, weight distribution.

1 Introduction

Consider an ensemble of turbo-like codes, defined by a concatenation of *uniformly interleaved* convolutional encoders. Such ensembles of codes include the standard parallel concatenated turbo codes [2, 3], ensembles of uniformly interleaved and serially concatenated codes [1], and other ensembles of turbo-like codes, such as the repeat-accumulate (RA) codes [8]. For a fixed decoding algorithm, these ensembles of codes typically exhibit a threshold phenomenon: most of the sufficiently long codes from such an ensemble can be used to transmit reliably up to a particular channel parameter, called the *threshold*, but will result in error rates bounded away from zero by a fixed constant above this threshold. For the case of *iterative decoders*, such thresholds have been determined, e.g., in [19, 20]. In such a setting it is then of interest to determine *how much is lost* by employing an iterative decoding technique rather than the optimal *maximum likelihood* (ML) decoder. A lower bound on the suffered loss can be found by upper bounding the error probability of an ML decoder (typically as a function of the weight distribution). This has been done, e.g., in [7, 8, 10, 14, 21, 22, 24, 25] for various classes of code ensembles. A basic ingredient of this approach is to determine the average weight distribution of the given ensemble. Although there are analytic approaches to determine this weight distribution *exactly* for relatively small lengths [5, 7, 15], none of the suggested approaches allow for a direct determination of the asymptotic growth rate. One exception is the case of repeat-accumulate codes, for which a compact analytic formula for the exact input-output weight distribution has been found [8]. Moreover, the weight distribution of a class of doubly concatenated repeat-accumulate codes (introduced in [18]), as well as the weight distribution of a rate-1 convolutional encoder with transfer function $\frac{1}{1+D+D^2}$ were derived in [12] (see Appendix B, pp. 134–137).

In this paper, we will present a simple method that allows for the determination of the asymptotic input-output weight distribution of convolutional encoders (see also [23]). In some cases, we are able to derive analytic expressions. In general, the determination of the growth rate for a fixed normalized input weight κ and output weight ω requires the (numerical) solution of a system of polynomial equations. We then apply our method to determine the growth rates of the weight distributions of various concatenated ensembles and to derive thresholds under ML decoding.[1]

The paper is organized as follows: In Section 2 we introduce our notations and give some general transformation rules for the asymptotic exponent of input-output weight distributions of convolutional encoders. In Section 3 we derive a method which is used to calculate the asymptotic growth rate of input-output weight distributions of convolutional encoders. Considerations regarding the use of this method for general convolutional encoders and some computation tasks which are involved in that respect are addressed in Section 4. Applications of this method to ensembles of uniformly interleaved parallel and serial concatenated codes are presented in Section 5. Finally, we exemplify our method to the classical turbo code which was devised by Berrou et al. [3].

2 Convolutional Encoders and General Transformation Rules

Let F be a field and let $\mathsf{F}[[D]]$ denote the associated ring of rationals. By a *convolutional encoder* of *rate* $R_{\mathrm{c}} := \frac{1}{t}$, $t \in \mathbb{N}$, we mean a t-tuple $G(D) := (G_1(D), \cdots, G_t(D))$, where $G_i(D) \in \mathsf{F}[[D]]$, $i \in [t]$.[2] All our examples will be limited to the case $\mathsf{F} = \mathsf{GF}(2)$ and $t \in \{1, 2\}$, although the basic method is not restricted to this setting. Let $\mathtt{x}$ be an input sequence of length l and let $\mathtt{y}$ be the corresponding output, where we restrict the output to the first l time steps, i.e., $\mathtt{y}$ will have total length lt. We will write $\mathtt{y} = \mathtt{x}G$ to denote this relationship. Consider the set

$$\mathcal{S}_G^l(\kappa, \omega) := \{(\mathtt{x}, \mathtt{y}) : \mathtt{y} = \mathtt{x}G; W(\mathtt{x}) = \kappa l; W(\mathtt{y}) = \omega t l\},$$

where $W(\mathtt{x})$ denotes the *Hamming weight* of $\mathtt{x}$ and κ, ω denote the *normalized* input and output weights, respectively. We are interested in the exponential growth rate of $|\mathcal{S}_G^l(\kappa, \omega)|$, which we denote by $F_G(\kappa, \omega)$. To make this growth rate well defined, consider the cardinality of the set of all input-output pairs which have normalized weight within ϵ, $\epsilon > 0$, of κ and ω, respectively. One can check that this function has a well defined growth rate. Now define $F_G(\kappa, \omega)$ as the limit of this growth rate as ϵ approaches zero. One can verify that this limit is well defined and that the limit function $F_G(\kappa, \omega)$ is continuous.[3]

[1] For iteratively decoded ensembles, there is not yet a general theory which enables one to calculate analytically the performance of these codes. However, for iteratively decoded ensembles over the binary erasure channel, one can proceed with the performance analysis in a completely similar fashion to the performance analysis of ML decoding, provided that one looks not at the weight distribution of the codewords, but rather at the weight spectrum of *stopping sets*, see [6].

[2] The generalization of our results to encoders of rate $\frac{p}{q}$ is straightforward and we will omit the details.

[3] Consider the sets of the input/output sequences that determine $|\mathcal{S}_G^{l_1}(\kappa, \omega)|$ and $|\mathcal{S}_G^{l_2}(\kappa, \omega)|$, respectively. By combining a sequence from the first set with a sequence from the second set, separated by a number of zeros which is equal to the memory length of the convolutional encoder (M), one obtains that $\log |\mathcal{S}_G^{l_1+l_2+M}(\kappa, \omega)| \geq \log |\mathcal{S}_G^{l_1}(\kappa, \omega)| + \log |\mathcal{S}_G^{l_2}(\kappa, \omega)|$, a property akin to super-additivity, which implies that the limit of the sequence $\left\{ \frac{\log |\mathcal{S}_G^{l}(\kappa, \omega)|}{l} \right\}_{l=1}^{\infty}$ exists (this serves as an outline of a proof why the function $F_G(\cdot, \cdot)$ is well defined).

Note that $F_G(\kappa, \omega)$ does not depend on the initial state of the encoder. This is true, since within a finite number of steps, one can change the state of the encoder from any given initial state to any desired final state. Therefore, by prepending a finite prefix to the input one can relate the output sequences corresponding to different initial states, [11]. The finite prefix has a negligible effect on the resulting input and output weights, from which the claim follows.

The following straightforward transformation rules allow in some cases to relate the exponential growth rates of two convolutional encoders. We say that a rational function $\frac{P(D)}{Q(D)}$ is in *minimal* form if $P(D)$ and $Q(D)$ are relatively prime. More generally, we will say that $G(D)$ is in *minimal* form if all its components $G_i(D)$ are in minimal form. Given a general encoder $G(D)$, we associate to it an encoder $\hat{G}(D)$ by eliminating from each component of $G(D)$ all common factors.

Lemma 1 (Transformation Rules)**.** The following transformation rules hold for the asymptotic growth rates of convolutional codes:

[Minimal Form] $F_G(\kappa, \omega) = F_{\hat{G}}(\kappa, \omega)$.

[Shift] For any convolutional encoder $G(D) = (G_1(D), \cdots, G_t(D))$ of rate $\frac{1}{t}$ and any t-tuple of natural numbers $(s_1, \cdots, s_t)$,

$$F_{[D^{s_1}G_1(D), \cdots, D^{s_t}G_t(D)]}(\kappa, \omega) = F_{G(D)}(\kappa, \omega).$$

[Multiplicity] For any convolutional encoder $G(D)$ and any integer d,

$$F_{G(D^d)}(\kappa, \omega) = F_{G(D)}(\kappa, \omega).$$

[Duality] Let $G(D)$ be a convolutional encoder of rate one. Then

$$F_{\frac{1}{G}}(\kappa, \omega) = F_G(\omega, \kappa).$$

Proof. The minimal and the shift rule are elementary and are only listed for completeness. For the multiplicity rule, note that the output of the encoder $G(D^d)$ at the i-th time instance only depends on the input at time instances which are congruent to i modulo d. Therefore, we can think of generating the output by first splitting the input into d separate streams and feeding each stream into an encoder $G(D)$ (rather than $G(D^d)$).[4] Finally consider the duality rule. Assume that $\mathtt{y} = \mathtt{x}G$ where $W(\mathtt{x}) = \alpha l$ and $W(\mathtt{y}) = \beta l$. Now feed $\mathtt{y}$ into the encoder $\frac{1}{G(D)}$. Then by definition, we will recover $\mathtt{x}$ (but perhaps for the last M steps, where M is the *memory length* of the encoder). This shows that $F_{\frac{1}{G}}(\beta, \alpha) \geq F_G(\alpha, \beta)$. The claim is now established by noting that by a reversal of the roles of $G(D)$ and $\frac{1}{G(D)}$, we get the converse inequality $F_G(\alpha, \beta) \geq F_{\frac{1}{G}}(\beta, \alpha)$. $\square$

Example 1 ($G(D) = \frac{D}{1+D^2}$)**.** Consider the example $G(D) = \frac{D}{1+D^2}$. By the shift rule we have $F_{\frac{D}{1+D^2}}(\kappa, \omega) = F_{\frac{1}{1+D^2}}(\kappa, \omega)$, which by the multiplicity rule is equivalent to $F_{\frac{1}{1+D}}(\kappa, \omega)$. Using finally the duality rule, we conclude that $F_{\frac{D}{1+D^2}}(\kappa, \omega) = F_{1+D}(\omega, \kappa)$.

[4] The problem that in the case of a $G(D^d)$ encoder the initial states are *linked* can be circumvented by inserting finite length guard sequences which guarantee that the encoder returns into the zero state.

3 The Basic Method

We will now discuss the basic method of determining the quantity $F_G(\kappa, \omega)$.

Assume we are given a list $\mathcal{L} := \{(k_i, w_i, l_i)\}$ of triples. Each triple represents a *pattern* of *input weight* k_i, *output weight* w_i and *input length* l_i. Let l be the given block length (length of the input stream).[5] We would like to determine in how many ways one can arrange these patterns (in a linear fashion) such that their total input weight is k, their total output weight is w and their total input length is l. More precisely, we are interested in the *growth rate* of the quantity

$$\sum_{1\le m\le l} \sum_{\substack{\sum m_i k_i = k;\ \sum m_i w_i = w \\ \sum m_i l_i = l;\ \sum m_i = m}} \binom{m}{m_1, \cdots, m_j}.$$

Define the *normalized* quantities $\kappa := \frac{k}{l}$, $\omega := \frac{w}{lt}$, $\mu := \frac{m}{lt}$ and $\mu_i = \frac{m_i}{lt}$. Note that we normalize the input weight by the *input length* l, but the output weight as well as m and m_i are normalized by the *output length* lt. Then it is easily established that this growth rate, call it $F_{\mathcal{L}}(\kappa, \omega)$, is equal to

$$F_{\mathcal{L}}(\kappa, \omega) := \max_{\substack{0\le\mu\le\frac{1}{t} \\ \sum_i \mu_i k_i = \kappa/t;\ \sum_i \mu_i w_i = \omega \\ \sum_i \mu_i l_i = \frac{1}{t};\ \sum_i \mu_i = \mu}} \mu h(\frac{\mu_1}{\mu}, \cdots, \frac{\mu_j}{\mu}), \tag{1}$$

where $h(\cdots)$ denotes the entropy function to the base of the natural logarithm.

Using Lagrange multipliers, we see that we have to find the stationary points of

$$\mu h(\frac{\mu_1}{\mu}, \cdots, \frac{\mu_j}{\mu}) + \alpha \sum_i \mu_i k_i + \beta \sum_i \mu_i w_i + \gamma \sum_i \mu_i l_i + \delta \left(\sum_i \mu_i - \mu \right).$$

Nulling the partial derivative with respect to μ yields that $\delta = 1$ (based on the relation $\sum \mu_i = \mu$).

By also nulling the partial derivatives with respect to the variables $\mu_i \quad (i = 1, 2, \cdots, j)$, one obtains that the maximizing probability distribution has the form

$$\mu_i = \mu e^{\alpha k_i} e^{\beta w_i} e^{\gamma l_i}, \quad i = 1, \cdots, j \tag{2}$$

where the quantities α, β, γ and μ are chosen such that

$$\sum_i \mu_i k_i = \frac{\kappa}{t}, \tag{3}$$

$$\sum_i \mu_i w_i = \omega, \tag{4}$$

$$\sum_i \mu_i l_i = \frac{1}{t}, \tag{5}$$

$$\sum_i \mu_i = \mu. \tag{6}$$

Assume that we specify the list $\mathcal{L}$ by means of the *formal power sum*

$$R(X, Y, Z) := \sum_i X^{k_i} Y^{w_i} Z^{l_i},$$

[5] The corresponding length of the output stream is then lt.

where the triples (k_i, w_i, l_i) are *not necessarily distinct*. Then from (2) and (6) we get the equality

$$R(e^\alpha, e^\beta, e^\gamma) = 1 \ . \tag{7}$$

Define

$$\begin{aligned}
R_X(X,Y,Z) &:= X\frac{\partial R(X,Y,Z)}{\partial X} = \sum_i X^{k_i} Y^{w_i} Z^{l_i} k_i \ ,\\
R_Y(X,Y,Z) &:= Y\frac{\partial R(X,Y,Z)}{\partial Y} = \sum_i X^{k_i} Y^{w_i} Z^{l_i} w_i \ ,\\
R_Z(X,Y,Z) &:= Z\frac{\partial R(X,Y,Z)}{\partial Z} = \sum_i X^{k_i} Y^{w_i} Z^{l_i} l_i \ .
\end{aligned}$$

Note that for μ_i of the form (2), we can express the left side of (3) as

$$\sum_i \mu_i k_i = \mu \sum_i e^{\alpha k_i} e^{\beta w_i} e^{\gamma l_i} k_i = \mu \ R_X(e^\alpha, e^\beta, e^\gamma).$$

In a similar manner, we can express the left sides of the equations (4) and (5) in terms of R_Y and R_Z, respectively. Therefore, if we let $x = e^\alpha$, $y = e^\beta$ and $z = e^\gamma$, then we can write equations (3)-(7) in the compact form

$$\begin{aligned}
R(x,y,z) &= 1 \ , & (8)\\
R_X(x,y,z) &= \frac{\kappa}{\mu t} \ , & (9)\\
R_Y(x,y,z) &= \frac{\omega}{\mu} \ , & (10)\\
R_Z(x,y,z) &= \frac{1}{\mu t} \ . & (11)
\end{aligned}$$

Note that every choice of $(x,y,z) \in (\mathbb{R}^+)^3$ corresponds to a probability distribution on the list of patterns $\mathcal{L}$ and that each such choice has an associated entropy rate, call it F_R, which can be expressed as follows (based on equations (1), (2) and (8)-(11)):

$$\begin{aligned}
F_R(x,y,z) &= -\mu \sum_i \left\{ \frac{\mu_i}{\mu} \ln\left(\frac{\mu_i}{\mu}\right)\right\}\\
&= -\mu \sum_i \left\{ x^{k_i} y^{w_i} z^{l_i} \ln\left(x^{k_i} y^{w_i} z^{l_i}\right)\right\}\\
&= -\mu \left(R_X \ln(x) + R_Y \ln(y) + R_Z \ln(z)\right)\\
&= -\frac{R_X \ln(x) + R_Y \ln(y) + R_Z \ln(z)}{tR_Z}\\
&= -\frac{\kappa}{t}\ln(x) - \omega \ln(y) - \frac{1}{t}\ln(z) \ .
\end{aligned} \tag{12}$$

Summarizing the results so far, we obtain the following theorem:

Theorem 1. Let $R(X,Y,Z) := \sum_i X^{k_i} Y^{w_i} Z^{l_i}$ be a formal power sum specifying a set of patterns where the i-th pattern has an associated input weight k_i, output weight w_i and length l_i. Let $\mathcal{S}^l_R(\kappa, \omega)$ denote the set of all linear arrangements of these patterns whose input weight, output weight and input length are κl, $\omega t l$ and l, respectively. Let $F_R(\kappa, \omega)$ denote the growth rate of $|\mathcal{S}_R^l(\kappa, \omega)|$. Then

$$F_R(\kappa, \omega) = \sup_{(x,y,z) \in \mathcal{D}} \left\{ -\frac{\kappa}{t} \ln(x) - \omega \ln(y) - \frac{1}{t} \ln(z) \right\}$$

where

$$\mathcal{D} := \left\{ (x, y, z) \in (\mathbb{R}^+)^3 : R(x, y, z) = 1, \ \frac{R_X(x, y, z)}{R_Z(x, y, z)} = \kappa \, , \frac{R_Y(x, y, z)}{R_Z(x, y, z)} = \omega t \right\}. \tag{13}$$

If the formal power sum has a finite number of terms, then the above formalism simply amounts to a compact description of the solution. The power of this approach appears when we allow formal power sums with an *infinite* number of terms, such that these formal power sums can be represented as *rational* generating functions. More precisely, we consider formal power sums of the form $R(X, Y, Z) = \sum_i X^{k_i} Y^{w_i} Z^{l_i}$ such that $R(X, Y, Z) \in \mathbb{R}[[X, Y, Z]]$, the ring of rationals, or equivalently, $R(X, Y, Z) = \frac{P(X,Y,Z)}{Q(X,Y,Z)}$ for $P(X, Y, Z)$, $Q(X, Y, Z) \in R[X, Y, Z]$, where $Q(0, 0, 0) \neq 0$ so that $Q(X, Y, Z)^{-1}$ exists.

Let us now return to our original problem. Recall that given a convolutional encoder $G(D)$ we want to determine the growth rate of all input-output pairs which have normalized input weight κ and normalized output weight ω. To accomplish this, we need to find a generating function which counts *patterns* such that *every* codeword of the encoder can be parsed in a *unique* way into a sequence of these patterns and such that *any linear* arrangement of such patterns corresponds to a codeword. Let $\tilde{R}_G(X, Y, Z)$ denote the standard generating function of a convolutional encoder which counts *detours* from the zero state, i.e., codeword segments that start and end in the zero state and do not take on the zero state in between. Then a possible choice is $R_G(X, Y, Z) = Z + \tilde{R}_G(X, Y, Z)$, since every codeword is composed in a unique way of detours and "silent periods" in which the encoder rests in the zero state. But this choice is far from unique! Another possible choice is given by $R_G(X, Y, Z) = \frac{1}{1-Z} \tilde{R}_G(X, Y, Z)$. Here we think of patterns that consist of a "silent period" *together* with a detour. Although obviously both choices must lead to the same result, the involved algebra can be essentially different for the two cases. There are many more choices. We can count code segments that start and end in a given state $\mathtt{s}$ which is not necessarily the zero state. To be precise, let $R_G^{[\mathtt{s}]}(X, Y, Z)$ denote the generating function which counts detours which start and end in state $\mathtt{s}$, where we allow an arbitrarily long prefix which corresponds to self loops at state $\mathtt{s}$. For the sake of simplicity we will in the sequel limit our attention to those choices of the generating function.

We are now ready to list our main result.

Theorem 2. For a given convolutional encoder $G(D)$, let $R = R_G^{[\mathtt{s}]}(X, Y, Z)$ have the form $R_G^{[\mathtt{s}]}(X, Y, Z) = \frac{P(X,Y,Z)}{Q(X,Y,Z)}$, where P and Q have no common factor and where the powers of X, Y and Z correspond to the input weight, output weight and input length, respectively. Then

$$F_G(\kappa, \omega) = \sup_{(x,y,z) \in \mathcal{D}} \left\{ -\frac{\kappa}{t} \ln(x) - \omega \ln(y) - \frac{1}{t} \ln(z) \right\}$$

where the set $\mathcal{D}$ is introduced in (13).

We note that in view of the function $F_G(\cdot, \cdot)$, one can obtain the growth rate of the minimum distance of the associated code if the latter grows linearly with the block length. However, if this is not the case (e.g., the increase of the minimum distance of turbo codes is at most logarithmic with the interleaver length, even with the optimal interleaver, see [4]), then the function $F_G(\cdot,\cdot)$ is not helpful for determining the asymptotic behavior of the minimum distance of these codes.

In the following, we exemplify the use of Theorem 2 for convolutional encoders whose above equations can be solved analytically. In the following two examples, we think of input sequences with an arbitrary number of leading zeros, followed by a detour. As noted above, this approach gives rise to the factor $\frac{1}{1-Z}$ appearing in the formal power series $R(X,Y,Z)$.

Example 2 ($G(D)=1+D$)**.** We start with the simplest non-trivial encoder, namely $G(D)=1+D$. It is easily established that

$$R^{[0]}_{1+D}(X,Y,Z)=\frac{XY^2Z^2}{(1-Z)(1-XZ)}.$$

After some manipulations, one gets the following solution for the set of three equations: $x^*=\frac{(2\kappa-\omega)(1-\kappa)}{\kappa(2-2\kappa-\omega)}$, $z^*=\frac{2-2\kappa-\omega}{2-2\kappa}$, and $y^*=\left(\frac{(1-x^*z^*)(1-z^*)}{x^*{z^*}^2}\right)^{\frac{1}{2}}$. With these values we obtain

$$F_{1+D}(\kappa,\omega)=(1-\kappa)h\left(\frac{\omega}{2(1-\kappa)}\right)+\kappa h\left(\frac{\omega}{2\kappa}\right).$$

Note that by the duality property in Lemma 1, it follows that

$$F_{\frac{1}{1+D}}(\kappa,\omega)=(1-\omega)h\left(\frac{\kappa}{2(1-\omega)}\right)+\omega h\left(\frac{\kappa}{2\omega}\right). \tag{14}$$

As remarked in the introduction, the case $G(D):=\frac{1}{1+D}$ is actually a non-trivial case for which $F_G(\kappa,\omega)$ was previously known. Even more, in this case it was shown in [8] that the *exact* number of input-output pairs of input weight k, output weight w and length l is equal to

$$\binom{l-w}{\lfloor k/2\rfloor}\binom{w-1}{\lceil k/2-1\rceil},$$

from which the asymptotic growth rate, as stated in (14), can be deduced easily.

Example 3 ($G(D)=(1+D^2,1+D+D^2)$)**.** Consider now the example $G(D)=(1+D^2,1+D+D^2)$. After some manipulations we get

$$R^{[0,0]}_{(1+D^2,1+D+D^2)}(X,Y,Z)=\frac{XY^5Z^3}{(1-Z)(1-XYZ(1+Z))}. \tag{15}$$

A tedious but otherwise straightforward calculation reveals that

$$\begin{aligned}
z^* &= \begin{cases} \frac{\sqrt{5\kappa^2-10\kappa+8\kappa\omega+4(1-\omega)}-\kappa}{2(1-2\kappa)} & \text{if} \quad \kappa<\frac{1}{2} \\ \frac{\sqrt{5\kappa^2-10\kappa+8\kappa\omega+4(1-\omega)}+\kappa}{2(2\kappa-1)} & \text{if} \quad \kappa>\frac{1}{2} \end{cases} \\
y^* &= \left(\frac{1-{z^*}^2}{{z^*}^2}\frac{2\omega-\kappa}{5\kappa-2\omega}\right)^{\frac{1}{4}} \\
x^* &= \frac{1}{y^*z^*(1+z^*)}\frac{5\kappa-2\omega}{4\kappa}.
\end{aligned}$$

Using these relations we can establish that

$$F_{(1+D^2,1+D+D^2)}(\kappa,\omega) = -\frac{2-\kappa-2\omega}{4}\ln(z^*) - \frac{5\kappa-2\omega}{8}\ln\left(\frac{5\kappa-2\omega}{4\kappa(1+z^*)}\right) - \frac{2\omega-\kappa}{8}\ln\left(\frac{(2\omega-\kappa)(1-z^*)}{4\kappa}\right)$$

where the expression is valid in the region where $0 < z^* < 1$ and $x^*y^* > 0$ (the last condition is equivalent to $\kappa \leq 2\omega \leq 5\kappa$), and the exponent is zero elsewhere. A plot of $F_{(1+D^2,1+D+D^2)}(\kappa,\omega)$ is depicted in Fig. 1.

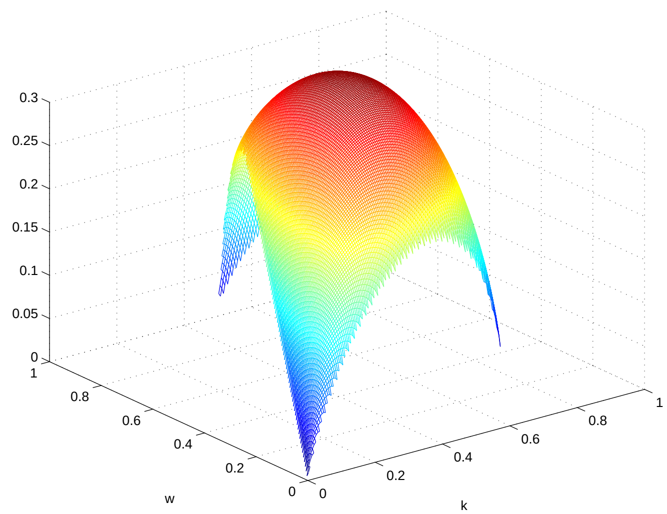

Figure 1: A plot of the asymptotic exponent $F_{(1+D^2,1+D+D^2)}(\kappa,\omega)$.

4 An Efficient Numerical Algorithm for the General Case

In general one can not hope to give explicit analytic expressions for the growth rate. Nevertheless, as we will show now, there exists an efficient numerical procedure to determine this growth rate to any desired degree of accuracy.

For the numerical determination of $F_G(\kappa,\omega)$, instead of trying to find the values of (x,y,z) that satisfy

$$R(x,y,z) = 1, \quad \frac{R_X(x,y,z)}{R_Z(x,y,z)} = \kappa, \quad \frac{R_Y(x,y,z)}{R_Z(x,y,z)} = \omega t$$

it is more advantageous to find a parametric representation of κ, ω and F_G in terms of (x,y). Then, by varying (x,y) over $(\mathbb{R}^+)^2$, we will trace the surface $(\kappa(x,y),\omega(x,y),F_G(x,y))$. To this end, observe first that for any $(x,y)\in(\mathbb{R}^+)^2$, the map $z\mapsto R(x,y,z)$ is non-decreasing for $z\geq 0$ (since all terms in the summation for R are non-decreasing in z), takes on the value zero at $z=0$, and is unbounded as z gets large. Thus, given $(x,y)\in(\mathbb{R}^+)^2$, one can find the unique $z = z(x,y)\in\mathbb{R}^+$ such that $R(x,y,z)=1$. One can now compute κ, ω and F_G via

$$\kappa = \frac{R_X(x,y,z)}{R_Z(x,y,z)}, \quad \omega = \frac{R_Y(x,y,z)}{tR_Z(x,y,z)}, \quad F_G = -\frac{\kappa}{t}\ln(x) - \omega\ln(y) - \frac{1}{t}\ln(z).$$

We have thus seen how to produce a plot of F_G as a function of κ and ω when the functions R, R_X, R_Y and R_Z are explicitly given. In most cases, however, the code is described in terms of a finite state machine that produces the codeword as its output when fed with the data sequence. To derive R explicitly from such a description becomes quickly infeasible as the memory grows. In such a case, one can still carry out the program outlined above by the following procedure: Pick a distinguished state among the states of the machine, and consider all input/output sequences that start with the machine in this state and end in the same state; let R be the formal power sum that is associated with this collection. Denote the distinguished state by 0, and let the other states be $1, \ldots, n$. Suppose the finite state machine is described by the input/output sequences that define one-step transitions between states. For $(i,j) \in \{1, \cdots, n\}^2$, let $a_{ji}(x,y,z) = x^{k_{ji}} y^{w_{ji}} z^{l_{ji}}$ if there is a one-step transition from state i to state j, where k_{ji} is the weight of the input sequence that takes the machine from state i to j, w_{ji} is the weight of the corresponding output sequence and l_{ji} is the length of the input sequence; set $a_{ji} = 0$ if there is no one-step transition from state i to state j. Also, let $b_i(x,y,z) = x^{k_i} y^{w_i} z^{l_i}$, where k_i is the weight of the input sequence that takes the machine from state 0 to state i, w_i the weight of the corresponding output sequence and l_i the length of the input sequence, and let $c_i(x,y,z) = x^{k_i} y^{w_i} z^{l_i}$, where k_i is the weight of the input sequence that takes the machine from state i to state 0, w_i the weight of the corresponding output sequence and l_i the length of the input sequence. If there is a self-loop at state 0, let $d(x,y,z)$ be the corresponding term for that loop, otherwise let $d(x,y,z) = 0$. With all these definitions, one sees that

$$R(x,y,z) = [1 - d(x,y,z)]^{-1} c(x,y,z)^T [I_n - A(x,y,z)]^{-1} b(x,y,z), \tag{16}$$

where A is the matrix whose entries are a_{ij}, and b and c are column vectors whose entries are b_i and c_i. Now, for any (x,y,z), one can evaluate the value of $R(x,y,z)$, by first evaluating a_{ij}, b_i, c_i and d, then compute the value of R via the formula above. Recall that to carry out our program we also need the values of R_X, R_Y and R_Z, the partial derivatives of R. Even though one could compute these derivatives at a point (x,y,z) numerically by evaluating R at points (x,y,z), $(x+\delta,y,z)$, etc., this leads to instability. A better way is to use (16) and write

$$\begin{aligned}\frac{\partial R}{\partial \nu} &= [1-d]^{-1} \frac{\partial d}{\partial \nu} R + [1-d]^{-1} \frac{\partial c^T}{\partial \nu} [I_n - A]^{-1} b \\ &\quad + [1-d]^{-1} c^T [I_n - A]^{-1} \frac{\partial A}{\partial \nu} [I_n - A]^{-1} b + [1-d]^{-1} c^T [I_n - A]^{-1} \frac{\partial b}{\partial \nu}\end{aligned}$$

for $\nu \in \{x,y,z\}$, so that R_X, R_Y and R_Z can be evaluated directly from the finite state machine description.

A point that needs to be recognized is that (16) is valid only in the region of convergence of the formal power sum R, and R computed via (16) is no longer non-decreasing in z. However, this does not cause any major difficulty, as the convergence of the formal power sum is equivalent to the spectral radius $\rho(A)$ of A being less than 1. For any $(x,y,z) \in (\mathbb{R}^+)^3$, one can check if $\rho(A)$ is less than 1; if not, assign the value ∞ to R, otherwise compute R via (16). Since the entries of A are non-decreasing in z, so is $\rho(A)$, and this procedure thus yields a value of R that is non-decreasing in z. In implementing this approach however, one should be careful not to try to compute $\rho(A)$ precisely for all (x,y,z) as this is computationally expensive. It is better to perform first a quick check to see if $\rho(A)$ is clearly larger than 1 or clearly less than 1, and compute ρ only in those cases where there is ambiguity.

Example 4. As an illustration, Fig. 2 shows the asymptotic exponent of the input-output weight distribution of the encoder $G(D) = \frac{1+D^4}{1+D+D^2+D^3+D^4}$ computed by this method. This encoder is of particular interest since the classical turbo code [3] comprises two such encoders as components. We will use this result later in Example 8 to compute the asymptotic weight distribution of the classical turbo code and to determine a lower bound on its threshold under ML decoding.

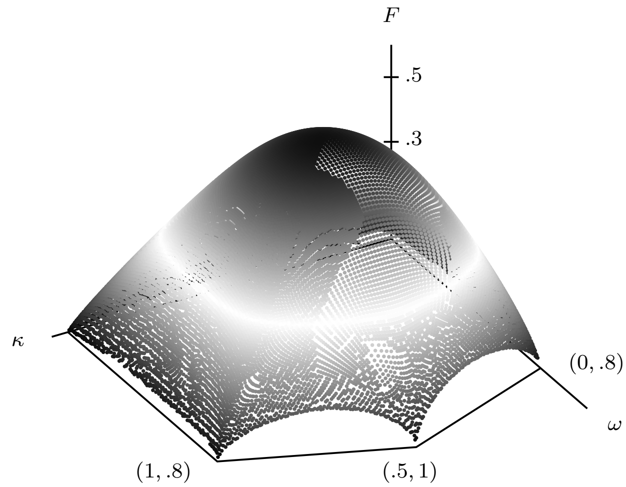

Figure 2: A plot of the asymptotic exponent $F_{\frac{1+D^4}{1+D+D^2+D^3+D^4}}(\kappa,\omega)$.

5 Applications to Uniformly Interleaved Concatenated Codes

5.1 Ensembles of Serially Concatenated Codes

We consider here the asymptotic weight distribution of ensembles of uniformly interleaved serially concatenated codes. Let $\mathcal{C}_s$, $\mathcal{C}_o$, $\mathcal{C}_i$ and N designate the serially concatenated code, the outer and inner codes, and the interleaver length respectively. The input-output weight distribution of this ensemble of codes is calculable in terms of the input-output weight distributions of its component codes [1] via

$$A_{\mathcal{C}_s}(\kappa,\omega) = \sum_{\delta} \frac{A_{\mathcal{C}_o}(\kappa,\delta)A_{\mathcal{C}_i}(\delta,\omega)}{\binom{N}{N\delta}}, \tag{17}$$

where κ, δ and ω are normalized with respect to the information block length, the interleaver length and the length of the output bits respectively; $A_{\mathcal{C}_o}(\kappa,\delta)$, $A_{\mathcal{C}_i}(\delta,\omega)$ and $A_{\mathcal{C}_s}(\kappa,\omega)$ designate the number of codewords of the outer, inner and serial concatenated code respectively, and the first and second parameters stand for the normalized input and output bits of these codes respectively. The parameter δ in the summation gets the values in the set $\left\{0, \frac{1}{N}, \ldots, 1\right\}$, and it therefore varies continuously between zero and unity in the limit where $N \to \infty$.

Let R_o, R_i and n designate the rates of the outer and inner codes and the length of the serial concatenated code, respectively. The asymptotic growth rates of the input-output weight distributions of the component codes and the serial concatenated code satisfy the equalities

$$A_{\mathcal{C}_o}(\kappa,\delta) \doteq e^{nR_iF_{\mathcal{C}_o}(\kappa,\delta)},\ A_{\mathcal{C}_i}(\delta,\omega) \doteq e^{nF_{\mathcal{C}_i}(\delta,\omega)},\ A_{\mathcal{C}_s}(\kappa,\omega) \doteq e^{nF_{\mathcal{C}_s}(\kappa,\omega)}.$$

Since $\binom{N}{N\delta} \doteq e^{Nh(\delta)}$, one gets from (17)

$$F_{\mathcal{C}_s}(\kappa,\omega) = \max_{0\le\delta\le1}\{R_iF_{\mathcal{C}_o}(\kappa,\delta) + F_{\mathcal{C}_i}(\delta,\omega) - R_ih(\delta)\}. \tag{18}$$

We will consider now the particular case where the inner code is a rate-1 differential encoder

($R_i = 1$). This ensemble of uniformly interleaved serially concatenated codes encompass the ensemble of the Repeat-Accumulate codes [8]. Based on Example 2 and (18), one gets

$$F_{\mathcal{C}_s}(\kappa,\omega) = \max_{0\leq\delta\leq\alpha_\omega}\left\{F_{\mathcal{C}_o}(\kappa,\delta) + (1-\omega)h\left(\frac{\delta}{2(1-\omega)}\right) + \omega h\left(\frac{\delta}{2\omega}\right) - h(\delta)\right\}. \tag{19}$$

where $\alpha_\omega := \min\{2\omega, 2(1-\omega)\}$. Note that since the binary entropy function is concave, by invoking Jensen's inequality in (19), one gets the inequality $F_{\mathcal{C}_s}(\kappa,\omega) \leq \max_{0\leq\delta\leq\alpha_\omega} F_{\mathcal{C}_o}(\kappa,\delta)$. This indicates the weight distribution thinning for the uniformly interleaved serially concatenated scheme which shapes the weight distribution of the outer convolutional code to resemble more closely the binomial distribution (typically of a full random code of the same rate), as was also mentioned in [16, 17].

Example 5 (Repeat-Accumulate Codes)**.** For the RA ensemble, see [8], it is well-known that $A_{\mathcal{C}_o}(\kappa,\delta)$ equals $\binom{n/q}{n\delta/q}$ if $\kappa = \delta$ and is zero otherwise, or equivalently, that

$$F_{\mathcal{C}_o}(\kappa,\delta) = \begin{cases} \frac{h(\delta)}{q}, & \text{if } \kappa = \delta, \\ -\infty, & \text{otherwise.} \end{cases} \tag{20}$$

By substituting (20) into (19) we get

$$F_{\mathrm{RA}}(\kappa,\omega) = -\left(1-\frac{1}{q}\right)h(\kappa) + (1-\omega)h\left(\frac{\kappa}{2(1-\omega)}\right) + \omega h\left(\frac{\kappa}{2\omega}\right). \tag{21}$$

From this it follows immediately that the asymptotic growth rate of the average distance spectrum of uniformly interleaved RA codes is equal to

$$r_{\mathrm{RA}}(\omega) = \max_{0\leq\kappa\leq\min\{2\omega,2(1-\omega)\}} F_{\mathrm{RA}}(\kappa,\omega), \tag{22}$$

which coincides with the analysis of Jin and McEliece [13].

Example 6. Let's next replace the rate-$\frac{1}{q}$ repetition code of the RA ensemble with a convolutional encoder whose generator is $G_{\mathcal{C}_o}(D) = [1, 1, \ldots, 1, 1+D]$. We will see that this improves the ML decoding threshold on the binary-input AWGN channel markedly. From Example 2, it can be easily verified that

$$F_{\mathcal{C}_o}(\kappa,\omega) = \frac{1-\kappa}{q}\cdot h\left(\frac{q\omega-(q-1)\kappa}{2(1-\kappa)}\right) + \frac{\kappa}{q}\cdot h\left(\frac{q\omega-(q-1)\kappa}{2\kappa}\right). \tag{23}$$

Substituting (23) into (19) gives

$$\begin{aligned} &F_{\mathcal{C}_s}(\kappa,\omega) \\ &\quad= \max_{0\leq\delta\leq 1}\left\{\frac{1-\kappa}{q}\cdot h\left(\frac{q\delta-(q-1)\kappa}{2(1-\kappa)}\right) + \frac{\kappa}{q}\cdot h\left(\frac{q\delta-(q-1)\kappa}{2\kappa}\right)\right. \\ &\qquad\qquad \left. +(1-\omega)h\left(\frac{\delta}{2(1-\omega)}\right) + \omega\, h\left(\frac{\delta}{2\omega}\right) - h(\delta)\right\}. \end{aligned}$$

Based on the bound of Divsalar [7], an upper bound on the threshold of $\frac{E_b}{N_0}$ which corresponds to ML decoding of a code or ensemble of codes is given by

$$\left(\frac{E_b}{N_0}\right)_{\text{threshold}} \leq \frac{1}{R_{\mathrm{c}}}\cdot\max_{0\leq\omega\leq1}\left\{\frac{(1-e^{-2r(\omega)})(1-\omega)}{2\omega}\right\}, \tag{24}$$

where $r(w)$ designates the asymptotic growth rate of the distance spectrum, and R_{c} is the rate (in bits per channel use) of the code (or ensemble). For the ensemble of RA codes (Example 5), then $r(w) = \max_{0\leq\kappa\leq1} F_{\mathcal{C}_s}(\kappa,\omega)$ and $R_{\mathrm{c}} = \frac{1}{q}$. This upper bound on the $\frac{E_b}{N_0}$ threshold was evaluated numerically as a function of the parameter q (which gives rise to a rate-$\frac{1}{q}$ code), and the resulting thresholds are also compared in Table 1 to the ultimate capacity limit of rate-$\frac{1}{q}$ codes.

q	RA ensemble	The ensemble of Example 6	Capacity limit
2	3.419 dB	1.020 dB	0.184 dB
3	0.792 dB	0.074 dB	-0.495 dB
4	-0.052 dB	-0.392 dB	-0.794 dB
5	-0.480 dB	-0.668 dB	-0.963 dB
6	-0.734 dB	-0.848 dB	-1.071 dB
7	-0.900 dB	-0.974 dB	-1.150 dB
8	-1.015 dB	-1.065 dB	-1.210 dB

Table 1: Upper bounds on $\frac{E_b}{N_0}$ for a binary-input AWGN channel and ML decoding, based on Divsalar's bound [7] for the ensemble of the RA codes (Example 2) and the ensemble of serially concatenated codes in Example 6.

Example 7. Let's consider next the ensemble of uniformly interleaved serially concatenated codes, where the outer code is a convolutional encoder whose generator is $G_{\mathcal{C}_o}(D) = [1+D^2, 1+D+D^2]$, and the inner code is a differential encoder. This ensemble was studied by Divsalar and Polara [9] for short block lengths. The asymptotic exponent of the input-output weight distribution of this ensemble is (based on Example 3 and (19))

$$\begin{aligned} &F_{\mathcal{C}_s}(\kappa,\omega) \\ &= \max_{0\le\delta\le\min\{2\omega,2(1-\omega)\}} \Bigg\{ \frac{4-3\kappa-2\delta-8\varepsilon}{8}\, h\left(\frac{2\delta-\kappa}{4-3\kappa-2\delta-8\varepsilon}\right) + \frac{\kappa}{2}\, h\left(\frac{2\delta-\kappa}{4\kappa}\right) + \\ &\quad \frac{5\kappa-2\delta}{8} h\left(\frac{8\varepsilon}{5\kappa-2\delta}\right) + (1-\omega)h\left(\frac{\delta}{2(1-\omega)}\right) + \omega h\left(\frac{\delta}{2\omega}\right) - h(\delta)\Bigg\} \end{aligned}$$

where

$$\varepsilon := \frac{2(1-\delta) - \sqrt{4(1-\delta) - 10\kappa + 5\kappa^2 + 8\kappa\delta}}{8}.$$

Based on the bound of Divsalar [7], an upper bound on the threshold of $\frac{E_b}{N_0}$ for a binary-input AWGN channel is given in (24), where $R_{\rm c} = \frac{1}{2}$ is the rate of the ensemble of the serially concatenated codes considered here, and $r(\omega) = \max_{0\le\kappa\le 1} F_{\mathcal{C}_s}(\kappa,\omega)$ is the asymptotic exponent of the distance spectrum of this ensemble. The threshold was calculated numerically, and is equal to 0.397 dB. For a code rate of $\frac{1}{2}$ bits per channel use, the value of $\frac{E_b}{N_0}$ which corresponds to the capacity of a binary-input AWGN channel is equal to 0.184 dB. Hence, the gap between the channel capacity and the threshold of the considered ensemble under ML decoding is at most 0.213 dB.

Fig. 3 compares the asymptotic growth rates of the distance spectrum of the ensemble of codes considered here with the ensembles of codes of rate–$\frac{1}{2}$ in Examples 2,6 (where $q=2$). The upper bounds on the thresholds of the $\frac{E_b}{N_0}$ values for the ensembles of the rate–$\frac{1}{2}$ codes referred in curves 1, 2 and 3 of Fig. 3 are 3.419 dB, 1.020 dB and 0.397 dB respectively, where the ultimate capacity limit correspond to an $\frac{E_b}{N_0}$ value of 0.184 dB (these upper bounds on the thresholds are based on the bound of Divsalar [7] for the binary-input AWGN channel and ML decoding). It was noted in [17] that the improvement in the ensemble performance associated with the optimal ML decoding is attributed to the "spectral thinning" phenomenon, and based on the thresholds above and the plots of the growth rates of the distance spectra in Fig. 3, it is reflected that the threshold of an ensemble of codes with the ML decoding is improved by narrowing the shape of the distance spectrum.

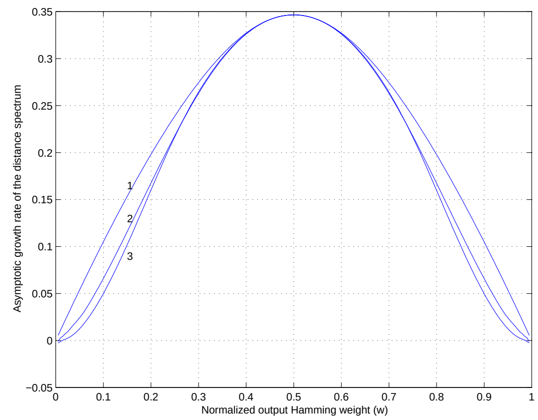

Figure 3: A plot of the asymptotic growth rates of the distance spectrum of three ensembles of serially concatenated, uniformly interleaved and differentially encoded codes of rate–$\frac{1}{2}$. Curve 1 corresponds to the ensemble of rate–$\frac{1}{2}$ RA codes, curve 2 corresponds to the ensemble with the outer convolutional encoder $G_{\mathcal{C}_o}(D) = [1, 1 + D]$, and curve 3 corresponds to the ensemble of codes with the outer convolutional encoder $G_{\mathcal{C}_o}(D) = [1 + D^2, 1 + D + D^2]$.

5.2 Ensembles of Parallel Concatenated (Turbo) Codes

We consider here the asymptotic input-output weight distribution of ensembles of uniformly interleaved *parallel* concatenated codes. We designate by $\mathcal{C}_p$ the ensemble of uniformly interleaved turbo codes, and we designate its two components (which are recursive systematic convolutional encoders) by $\mathcal{C}_1$, $\mathcal{C}_2$. We assume that no puncturing of the parity bits of these ensembles is performed (i.e., the rate of the code is $R_c = \frac{1}{3}$). The input-output weight distributions of these ensembles is calculable in terms of the input-output weight distributions of their component codes [2] via

$$A_{\mathcal{C}_p}(\kappa, \omega) = \sum_{\substack{\delta_1, \delta_2 \geq 0: \\ \kappa + \delta_1 + \delta_2 = \frac{\omega}{R_c}}} \frac{A_{\mathcal{C}_1}(\kappa, \delta_1) A_{\mathcal{C}_2}(\kappa, \delta_2)}{\binom{N}{N\kappa}}, \tag{25}$$

where the parameters κ, δ_1, δ_2 are normalized with respect to the length of the information bits (which also equals the length of the interleaver), and ω is normalized with respect to the block length of the turbo code. The first three parameters designate the normalized Hamming weights of the information bits and the normalized Hamming weight of the parity bits of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$, and the last parameter (ω) designate the normalized Hamming weight of the overall code. Since the rate of the code is one-third, then we get the equality: $\kappa + \delta_1 + \delta_2 = 3\omega$.
Similarly to the previous section, in the limit case where $N \to \infty$, we get the following equality for the asymptotic exponential behavior of input-output weight distribution of the turbo code in terms of the corresponding exponents for the two component codes

$$F_{\mathcal{C}_p}(\kappa, \omega) = \frac{1}{3} \max_{\substack{\delta_1, \delta_2 \geq 0: \\ \kappa + \delta_1 + \delta_2 = 3\omega}} \left\{ F_{\mathcal{C}_1}(\kappa, \delta_1) + F_{\mathcal{C}_2}(\kappa, \delta_2) - h(\kappa) \right\} . \tag{26}$$

Example 8 (The Berrou et al. Turbo Code)**.** Consider the classical turbo code with two recursive systematic convolutional encoders $G_1(D) = G_2(D) = \frac{1+D^4}{1+D+D^2+D^3+D^4}$ and no puncturing. Based on Example 4 and equation (25), the asymptotic exponential behavior of the input-output weight distribution for this ensemble of codes can be computed. Then, the asymptotic exponent of the distance spectrum of the ensemble of turbo codes gets the form

$$r_{\mathcal{C}_p}(\omega) = \max_{0 \leq \kappa \leq 1} F_{\mathcal{C}_p}(\kappa, \omega).$$

In Fig. 4, we compare the asymptotic exponent of the distance spectrum of the classical turbo code with the logarithm of the distance spectrum of this ensemble of uniformly interleaved turbo codes (normalized with respect to the code length of its codewords) for finite uniform interleavers of lengths $N = 100, 200, 500, 1000$. Based on the bound of Divsalar [7], an upper bound on the

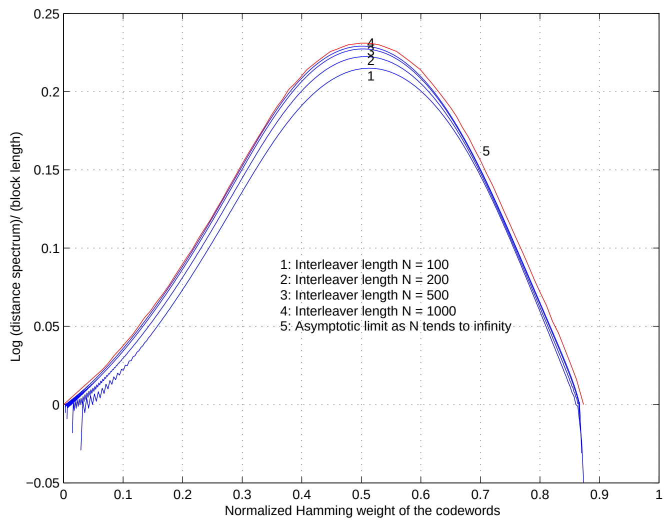

Figure 4: A comparison between the asymptotic exponent of the distance spectrum for the ensemble of uniformly interleaved classical turbo codes and the corresponding exponent for a finite length code with interleaver lengths of $N = 100, 200, 500$ and 1000.

threshold for these ensemble of turbo codes in the binary-input AWGN channel is given in (24), where $R_{\mathrm{c}} = \frac{1}{3}$ is the rate of this ensemble, and the asymptotic growth rate of its average distance spectrum as a function of the normalized Hamming weight is given by $r = r_{\mathcal{C}_p}$. A calculation of this bound yields that $\left(\frac{E_b}{N_0}\right)_{\text{threshold}} \leq -0.125$ dB, as compared to the ultimate Shannon capacity limit for a rate–$\frac{1}{3}$ code in the binary-input AWGN channel which corresponds to $\frac{E_b}{N_0} = -0.495$ dB (which yields that the gap between the channel capacity and the achievable threshold under ML decoding is at most 0.370 dB).

Acknowledgment The authors wish to thank the anonymous reviewers for their comments on an earlier version of this paper, and for their quick reviews which expedited the publication of this paper. Igal Sason wishes to thank Henry Pfister for his inputs on Table 1 in an earlier version of this paper.

References

[1] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, *Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding'*, IEEE Trans. on Information Theory, 44 (1998), pp. 909–926.

[2] S. Benedetto and G. Montorsi, *Unveiling turbo codes: Some results on parallel concatenated coding schemes*, IEEE Trans. on Information Theory, 42 (1996), pp. 409–428.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, *Near Shannon limit error-correcting coding and decoding*, in Proceedings of ICC'93, Geneva, Switzerland, May 1993, pp. 1064–1070.

[4] M. Breiling, *A logarithmic upper bound on the minimum distance of turbo codes.* submitted to IEEE Trans. on Information Theory, 2001.

[5] F. Daneshgaran and M. Mondin, *An efficient algorithm for obtaining the distance spectrum of turbo codes*, in Proceedings of the International Symposium on Turbo Codes and Related Topics, Brest, France, Sept. 1997, pp. 251–254.

[6] C. Di, D. Proietti, T. Richardson, E. Telatar, and R. Urbanke, *Finite length analysis of low-density parity-check ensembles for the binary erasure channel*, IEEE Trans. on Information Theory, 48 (2002), pp. 1570–1579.

[7] D. Divsalar, *A simple tight bound on error probability of block codes with applications to turbo codes.* TMO Progress Report 42-139, JPL, pp. 1-35, Nov. 1999.

[8] D. Divsalar, H. Jin, and R. McEliece, *Coding theorems for "turbo-like codes"*, in Proceedings of the 1998 Allerton Conference, Monticello, Sept. 1998, pp. 201–210.

[9] D. Divsalar and F. Polara, *Serial and hybrid concatenated codes with applications*, in Proceedings of the First International Symposium onTurbo Codes and Related Topics, 1997, pp. 80 – 87.

[10] T. M. Duman and M. Salehi, *New performance bounds for turbo codes*, IEEE Trans. on Communications, 46 (1998), pp. 717–723.

[11] J. Hokfelt, O. Edfors, and T. Maseng, *A survey on trellis termination alternatives for turbo codes*, IEEE Journal on Selected Areas in Communications, 19 (2001), pp. 838–847.

[12] H. Jin, *Analysis and design of turbo-like codes.* Ph.D. dissertation, Caltech, Pasadena, USA, 2001.

[13] H. Jin and R. J. McEliece, *Repeat-accumulate codes achieve awgn channel capacity*, in Proceedings of the International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 1999, pp. 10 – 18.

[14] ——, *Coding theorems for turbo code ensembles*, IEEE Trans. on Information Theory, 48 (2002), pp. 1451–1461.

[15] R. J. McEliece, *How to compute weight enumerators for convolutional codes*, in Communications and Coding, M. Darnel and B. Honary, eds., Tauton, Research Studies, 1998, ch. 6, pp. 121–141.

[16] M. Peleg, I. Sason, S. Shamai, and A. Elia, *On interleaved, differentially encoded convolutional codes*, IEEE Trans. on Information Theory, 45 (1999), pp. 2572–2581.

[17] L. Perez, J. Seghers, and D. Costello, *A distance spectrum interpretation of turbo codes*, IEEE Trans. on Information Theory, 42 (1996), pp. 1698–1709.

[18] H. Pfister and P. Siegel, *On the serial concatenation of rate-one codes through uniform interleavers*, in Thirty-seven Annual Allerton Conference on Communication, Control and Computing, Allerton, Monticello, Illinois, USA, September 1999, pp. 260 – 269.

[19] T. Richardson, A. Shokrollahi, and R. Urbanke, *Design of capacity-approaching low-density parity check codes*, IEEE Trans. on Information Theory, 47 (2001), pp. 619–637.

[20] T. Richardson and R. Urbanke, *The capacity of low-density parity check codes under message-passing decoding*, IEEE Trans. on Information Theory, 47 (2001), pp. 599–618.

[21] I. Sason and S. Shamai, *Improved upper bounds on the ML decoding error probability of parallel and serial concatenated turbo codes via their ensemble distance spectrum*, IEEE Trans. on Information Theory, 46 (2000), pp. 24–47.

[22] ——, *On improved bounds on the decoding error probability of block codes over interleaved fading channels, with application to turbo-like codes*, IEEE Trans. on Information Theory, 47 (2001), pp. 2275–2299.

[23] I. Sason, E. Telatar, and R. Urbanke, *The asymptotic input-output weight distribution of convolutional codes*, in Thirty-eight Annual Allerton Conference on Communication, Control and Computing, Allerton, Monticello, Illinois, USA, October 2000, pp. 276 – 286.

[24] S. Shamai and I. Sason, *Variations on the Gallager bounds, connections and applications*, IEEE Trans. on Information Theory, 48 (2002), pp. 3029–3051.

[25] E. Telatar and R. Urbanke, *On the ensemble performance of turbo codes*, in Proceedings IEEE International Symposium on Information Theory (ISIT '97), Ulm, Germany, June 1997, p. 105.