# Evaluation of the distance spectrum of variable-length finite-state codes

Claudio Weidmann, Michel Kieffer

# Evaluation of the Distance Spectrum of Variable-Length Finite-State Codes

Claudio Weidmann, *Member, IEEE* and Michel Kieffer, *Senior Member, IEEE*

***Abstract*—The class of variable-length finite-state joint source-channel codes is defined and a polynomial complexity algorithm for the evaluation of their distance spectrum presented. Issues in truncating the spectrum to a finite number of (possibly approximate) terms are discussed and illustrated by experimental results.**

***Index Terms*—Variable length codes, finite state machines, source coding, channel coding, communication system performance.**

## I. Introduction

JOINT source-channel (JSC) codes are being considered as a means to provide robust communication systems, *e.g.*, with strict delay constraints under time-varying channel conditions, or when no feedback channel is available for performing ARQ. An important issue is the availability of efficient performance evaluation tools that do not require simulation. The union bound on the event error probability is perhaps the most popular such tool and has been applied to convolutional codes [1, Ch. 4.4], trellis codes [2], [3], JSC variable-length codes (JSC-VLC) [4] and JSC arithmetic coding (JSC-AC) [5], as well as serially concatenated source-channel codes with iterative decoding in [6] and [7]. The union bound is computed using the code distance spectrum, which in the general case (of nonlinear, not geometrically uniform codes) requires comparing all pairs of code sequences. An exhaustive enumeration approach with exponential complexity has been proposed for JSC-VLC in [4] and applied to JSC-AC in [5].

Inspired by the polynomial algorithm for computing the free distance presented in [5], this letter presents a polynomial complexity algorithm for computing the spectrum of variable-length finite-state codes, a class of codes which includes JSC-VLC and finite-state implementations of JSC-AC. It is readily implemented on a computer, without resorting to generating functions. The main difference compared to previous work is that the rate (output bits per input symbol) is not assumed constant per transition in the state diagram. Thus it is not possible to enumerate the information (source) spectrum together with the code spectrum; only the latter will be computed.

The contributions of this letter are as follows. Section II introduces variable-length finite-state codes and computes stationary state probabilities adjusted for variable-length transitions, which are needed to average the spectrum, improving upon the approximate approach in [5]. Section III presents the union bound and the spectrum, Section IV gives a recursive spectrum evaluation algorithm, while Section V presents a direct algorithm using numerical matrix inversion. Section VI discusses the issues involved in bounding the error when approximating the spectrum by a finite number of terms, which are themselves approximations. Finally, Section VII provides experimental results for an example JSC-AC.

## II. Variable-Length Finite State Codes

A binary finite-state encoder (FSE) may be represented as a directed graph $\Gamma$ defined by a set $\mathcal{S}$ of states (vertices) and a set $\mathcal{T}$ of transitions (directed edges) between states, where each transition $u \in \mathcal{T}$ is labeled with a vector of input symbols $I(u)$ and a vector of output bits $O(u)$, whose lengths are $\ell(I(u))$ and $\ell(O(u))$, respectively. In the variable-length (VL) FSEs considered here, both input and output labels may be of variable length, generalizing fixed-length (FL) in – FL out FSEs [8] and FL in – VL out FSEs [9]. The starting point is a FSE with *nonempty* input and output labels, *e.g.*, a VLC for a $K$-ary source will be represented by the root state and $K$ VL-output transitions, while a JSC-AC will be represented by a *reduced* finite-state automaton [5] with VL input and VL output labels, all nonempty. Since these are JSC encoders, each transition will also have an associated transition probability $P(u)$, which may be computed using the source model. The sum of the outgoing transition probabilities of a state must equal one. In the following, a discrete memoryless source (DMS) is assumed, for which the transition probability is simply the product of the probabilities of the input label symbols. For a FSE to be a proper source encoder, for every state, the input labels of the outgoing transitions have to form a complete prefix set, which also implies that their transition probabilities sum to one.

Given an initial state $s_0$, the encoding of all possible (semi-) infinite input sequences can be displayed by a trellis with states aligned at equal output bit length connected by VL transitions. The VL *finite-state code* (FSC) $\mathcal{C}(\Gamma, s_0)$ is the

Paper approved by F. Alajaji, the Editor for Source and Source/Channel Coding of the IEEE Communications Society. Manuscript received April 10, 2009; revised August 1, 2009.

Claudio Weidmann is with the Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology, A-1040 Vienna, Austria (e-mail: claudio.weidmann@ieee.org).

Michel Kieffer is with LSS – CNRS – Supélec – Univ Paris-Sud, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette cedex, France (e-mail: Michel.Kieffer@lss.supelec.fr).

This work was supported by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract n. 216715).

Digital Object Identifier 00.0000/TCOMM.2010.0.000000

set of sequences of concatenated output labels corresponding to all paths through the trellis. Let $\sigma(u)$ be the originating state of a transition $u \in \mathcal{T}$ and $\tau(u)$ its target state. A path $\boldsymbol{u} = (u_1 \circ u_2 \circ \ldots \circ u_k) \in \mathcal{T}^k$ on the trellis is a concatenation (denoted by $\circ$) of transitions that satisfy $\sigma(u_{i+1}) = \tau(u_i)$ for $1 \le i < k$ (this corresponds to a walk of length $k$ on $\Gamma$). The probability of a path is $P(\boldsymbol{u}) = \prod_{i=1}^{k} P(u_i)$.

For the purpose of computing the code spectrum, it is advantageous to consider a *bit-clock* trellis [5], where each transition is labeled with exactly one output bit. The corresponding bit-clock FSE $\Gamma_b(\mathcal{S}_b, \mathcal{T}_b)$ is obtained by inserting additional deterministic transitions and states in order to split up output labels longer than one bit. A transition $t$ with such a label is replaced by a chain of $l = \ell(O(t))$ transitions $t^{(1)}, \ldots, t^{(l)}$ (and $l-1$ deterministic intermediate states), each labeled with the corresponding output bit, where $t^{(1)}$ "inherits" the input label of $t$ and the corresponding probability, while $t^{(2)}, \ldots, t^{(l)}$ have empty input labels and probability one. Define $M = |\mathcal{S}|$, then the state set $\mathcal{S}_b \supset \mathcal{S}$ of $\Gamma_b$ will be of size

$$M_b = |\mathcal{S}_b| = M - |\mathcal{T}| + \sum_{t \in \mathcal{T}} \ell(O(t)).$$

The following definitions will be used in the sequel. The set of transitions from $x$ to $y$ is $\mathcal{T}_{xy} = \{u \in \mathcal{T}_b : \sigma(u) = x, \tau(u) = y\}$, while $\mathcal{T}^*(y,z) = \{(u,v) \in \mathcal{T}_b^2 : \sigma(u) \ne \sigma(v), \tau(u) = y, \tau(v) = z\}$ is the set of all *pairs of transitions* ending in $y$ and $z$, respectively, and *not* starting from a common state. The *length of a path* will exclusively denote its output length in bits, which equals the number of transitions on the bit-clock trellis. The Hamming distance between the outputs of a pair of $n$-bit paths $(\boldsymbol{u}, \boldsymbol{v})$ is denoted by $d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v})$. The set $\mathcal{P}_n(x,y,z)$ contains all *pairs of paths* of length $n$ bits starting from state $x$ and ending in $y$ and $z$, respectively, without merging in between, *i.e.*, pairs of paths that have no state in common at bit indices $2, \ldots, n-1$. The set of all pairs of paths diverging in the starting state $x$ and converging for the first time $n$ bits later is

$$\mathcal{C}_n(x) = \bigcup_{y \in \mathcal{S}} \mathcal{P}_n(x,y,y).$$

It will be assumed that the encoder graph is *irreducible*,[1] *i.e.*, that any state can be reached from any other in a finite number of transitions, and that it is *aperiodic*,[2] *i.e.*, the state recurrence times are *not* multiples of an integer period $m > 1$. These assumptions imply that the FSE forms an ergodic Markov chain, which has a unique stationary state distribution. Thus the asymptotic steady-state performance for long (unterminated) code sequences will be independent of the initial state. It will be weighted with probabilities $\tilde{p}(x)$, $x \in \mathcal{S}$, that the current code bit belongs to a transition leaving $x$, since paths can only diverge in states $x \in \mathcal{S}$. Hence the probabilities $\tilde{p}(x)$ will depend on the lengths of the transition labels in $\Gamma$.

*Proposition 1:* The stationary probability that the current code bit belongs to a transition leaving $x \in \mathcal{S}$ is

$$\tilde{p}(x) = \frac{p_x^* \left(1 + \sum_{y \in \mathcal{S}} \sum_{l \ge 2} (l-1) p_{xy}^l\right)}{\sum_{x \in \mathcal{S}} p_x^* \left(1 + \sum_{y \in \mathcal{S}} \sum_{l \ge 2} (l-1) p_{xy}^l\right)}, \quad (1)$$

where $p_x^*$ is the stationary state probability on the FSE graph $\Gamma$,

$$p_x^* = \sum_{y \in \mathcal{S}} p_y^* \left( \sum_{l \ge 1} p_{yx}^l \right) \qquad (x \in \mathcal{S}),$$

and $p_{xy}^l = \sum_{u \in \mathcal{T}_{xy} : \ell(O(u)) = l} P(u)$ is the probability of reaching $y \in \mathcal{S}$ from $x \in \mathcal{S}$ by emitting an $l$-bit output label.

*Proof:* By its definition, $\tilde{p}(x)$ is the sum of the stationary probabilities (on the bit-clock graph $\Gamma_b$) of $x$ and of all intermediate states $x' \in \mathcal{S}_b \setminus \mathcal{S}$ corresponding to transitions leaving $x$. Let $q_x$ and $q_{xy}^{l,k}$, respectively, denote these probabilities, where the latter stands for the $k$-th intermediate state of an $l$-bit transition from $x$ to $y$ (parallel transitions of the same length are merged and their probabilities added). The equations for the stationary distribution on $\Gamma_b$ are thus

$$q_x = \sum_{y \in \mathcal{S}} q_y p_{yx}^1 + \sum_{l \ge 2} \sum_{y \in \mathcal{S}} q_{yx}^{l,l-1} \quad (x \in \mathcal{S}) \quad (2)$$

$$q_{yx}^{l,1} = q_y p_{xy}^l \quad (x, y \in \mathcal{S}) \quad (3)$$

$$q_{yx}^{l,k} = q_{yx}^{l,k-1} \quad (x, y \in \mathcal{S}, 2 \le k < l). \quad (4)$$

Combining (3) and (4) yields $q_{yx}^{l,l-1} = q_y p_{xy}^l$, which inserted into (2) leads to (1) after normalization. ∎

## III. Union Bound and Distance Spectrum

Consider a ML decoder applied to a FSC transmitted over a binary-input memoryless channel and let $P_{\mathrm{e}}$ be the event error probability at any position in the code sequence. The following union upper bound holds [1, Ch. 4.4]:

$$P_{\mathrm{e}} \le \sum_{d = d_{\mathrm{free}}}^{\infty} A_d P_d, \quad (5)$$

where

$$d_{\mathrm{free}} = \min_{n \ge 1, x \in \mathcal{S}} \min_{(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{C}_n(x)} d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v}) \quad (6)$$

is the *free distance* of the code,

$$A_d = \sum_{x \in \mathcal{S}} \tilde{p}(x) \sum_{n=d}^{\infty} \sum_{\substack{(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{C}_n(x) \\ d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v}) = d}} P(\boldsymbol{u}) \quad (7)$$

is the *spectral coefficient* that counts the average number of paths at Hamming distance $d$ from a given path, where $\tilde{p}(x)$ is the stationary probability of diverging in state $x$, and $P_d$ is the probability that the decoder selects an erroneous path at distance $d$ instead of the correct path. For BPSK signaling with energy $E_b$ per channel bit over an AWGN channel with

[1] There is little practical interest in FSEs with multiple ergodic components, however, all results apply if appropriately averaged over these components. Transient components may be eliminated, since they do not affect steady-state performance.

[2] The tools developed here can also be applied to periodic codes, either by carrying out all computations for an entire period $(b_n, \ldots, b_{n+m-1})$ at a time, instead of a single bit $b_n$, or by separate computations for each of the $m$ phases.

zero-mean Gaussian noise of variance $N_0/2$ this term is [1, Ch. 2.9]

$$P_d = \tfrac{1}{2}\operatorname{erfc}\left(\sqrt{d\tfrac{E_b}{N_0}}\right) \le \exp\left(-d\tfrac{E_b}{N_0}\right) \tag{8}$$

An important fact is that the union bound (5) is also a bound on the average bit error rate in the *code domain*, because an event error causing $d$ code bit errors has length at least $d$ bits.

## IV. Recursive spectrum evaluation

Let $A^x_{d,n}(y,z)$ be the average number of pairs of $n$-bit paths at Hamming distance $d$, starting in state $x \in \mathcal{S}$ and ending in states $y \in \mathcal{S}_b$ and $z \in \mathcal{S}_b$, respectively, without merging in between,

$$A^x_{d,n}(y,z) = \sum_{\substack{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{P}_n(x,y,z)\\ d_{\mathrm{H}}(\boldsymbol{u},\boldsymbol{v})=d}} P(\boldsymbol{u}). \tag{9}$$

When $z = y$, paths merge for the first time at length $n$. Using (9), one may rewrite (7) as

$$A_d = \sum_{x\in\mathcal{S}} \tilde{p}(x) \sum_{n=d}^{\infty} \sum_{y\in\mathcal{S}} A^x_{d,n}(y,y). \tag{10}$$

*Proposition 2:* For any $x \in \mathcal{S}$, $y, z \in \mathcal{S}_b$, $A^x_{d,n+1}(y,z)$ may be evaluated recursively as

$$\begin{aligned} A^x_{d,n+1}(y,z) = &\sum_{\substack{(u,v)\in\mathcal{T}^*(y,z)\\ d_{\mathrm{H}}(u,v)=0}} P(u) A^x_{d,n}(\sigma(u),\sigma(v)) \\ &+ \sum_{\substack{(u,v)\in\mathcal{T}^*(y,z)\\ d_{\mathrm{H}}(u,v)=1}} P(u) A^x_{d-1,n}(\sigma(u),\sigma(v)), \end{aligned} \tag{11}$$

for $d \ge 1$ and $n \ge d$. For $d = 0$ and $n \ge 1$, only the first sum in (11) is computed (*i.e.*, $A^x_{-1,n} = 0$ by definition). The recursion is initialized with

$$A^x_{d,1}(y,z) = \sum_{\substack{(u,v)\in\mathcal{T}_{xy}\times\mathcal{T}_{xz}\\ d_{\mathrm{H}}(u,v)=d}} P(u) \qquad (d = 0, 1), \tag{12}$$

*Proof:* The initialization (12) is a simple reformulation of (9) for $d = 0, 1$ and paths of length $n = 1$ bit. To prove (11), one may write (9) for pairs of paths of length $n + 1$,

$$A^x_{d,n+1}(y,z) = \sum_{\substack{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{P}_{n+1}(x,y,z)\\ d_{\mathrm{H}}(\boldsymbol{u},\boldsymbol{v})=d}} P(\boldsymbol{u}). \tag{13}$$

Consider any pair of paths $(\boldsymbol{u},\boldsymbol{v}) \in \mathcal{P}_{n+1}(x,y,z)$. Since the two paths have not merged in between, there exist a pair of states $(y',z'), y' \ne z'$, a pair of paths $(\boldsymbol{u}',\boldsymbol{v}') \in \mathcal{P}_n(x,y',z')$, and a pair of transitions $(u,v) \in \mathcal{T}^*(y,z)$ such that $\boldsymbol{u} = \boldsymbol{u}'\circ u$ and $\boldsymbol{v} = \boldsymbol{v}'\circ v$. Two cases have to be considered. First, one may have $d_{\mathrm{H}}(u,v) = 1$, in which case $d_{\mathrm{H}}(\boldsymbol{u}',\boldsymbol{v}') = d-1$. Second, one may have $d_{\mathrm{H}}(u,v) = 0$ and $d_{\mathrm{H}}(\boldsymbol{u}',\boldsymbol{v}') = d$. Hence (13) may be rewritten as

$$\begin{aligned} A^x_{d,n+1}(y,z) = &\sum_{\substack{(u,v)\in\mathcal{T}^*(y,z)\\ d_{\mathrm{H}}(u,v)=0}} \sum_{\substack{(\boldsymbol{u}',\boldsymbol{v}')\in\mathcal{P}_n(x,\sigma(u),\sigma(v))\\ d_{\mathrm{H}}(\boldsymbol{u}',\boldsymbol{v}')=d}} P(\boldsymbol{u}')P(u) \\ &+ \sum_{\substack{(u,v)\in\mathcal{T}^*(y,z)\\ d_{\mathrm{H}}(u,v)=1}} \sum_{\substack{(\boldsymbol{u}',\boldsymbol{v}')\in\mathcal{P}_n(x,\sigma(u),\sigma(v))\\ d_{\mathrm{H}}(\boldsymbol{u}',\boldsymbol{v}')=d-1}} P(\boldsymbol{u}')P(u), \end{aligned}$$

from which (11) is easily deduced. ∎

Without loss of generality, one may assume that the states in $\mathcal{S}$ are numbered $1, \ldots, M$ and the additional intermediate states in $\mathcal{S}_b \setminus \mathcal{S}$ are numbered $M+1, \ldots, M_b$. Then $A^x_{d,n}(y,z)$ may be viewed as a matrix indexed by $(y,z) \in \mathcal{S}_b^2$ and the recursion (11) written more compactly as

$$\bar{A}^x_{d,n+1} = P_0 \bar{A}^x_{d,n} + P_1 \bar{A}^x_{d-1,n}, \tag{14}$$

where $\bar{A}^x_{d,n} = A^x_{d,n}(:)$ is the column vector obtained by stacking the columns of $A^x_{d,n}$ and

$$P_h = \left(P_h\left(z,z'\right)\right)_{1\le z\le M_b, 1\le z'\le M_b} \qquad (h = 0, 1), \tag{15}$$

are two $M_b^2 \times M_b^2$ matrices. Each consists of $M_b \times M_b$ submatrices

$$P_h\left(z,z'\right) = \left(P_h(z,z',y,y')\right)_{1\le y\le M_b, 1\le y'\le M_b} \tag{16}$$

with

$$P_h\left(z,z',y,y'\right) = \begin{cases} 0, & \text{if } z' = y' \\ \displaystyle\sum_{\substack{(u,v)\in\mathcal{T}_{y'y}\times\mathcal{T}_{z'z}\\ d_{\mathrm{H}}(u,v)=h}} P(u), & \text{if } z' \ne y'. \end{cases} \tag{17}$$

Finally, let $\overline{I}_{M_b}$ be the column vector obtained by stacking the columns of the $M_b \times M_b$ identity matrix. Then (10) may be written as

$$\begin{aligned} A_d &= \sum_{x\in\mathcal{S}} \tilde{p}(x) \sum_{n=d}^{\infty} \overline{I}_{M_b}^{\mathrm{T}} \bar{A}^x_{d,n} \\ &= \sum_{x\in\mathcal{S}} \tilde{p}(x) \overline{I}_{M_b}^{\mathrm{T}} \sum_{n=d}^{\infty} \bar{A}^x_{d,n} = \sum_{x\in\mathcal{S}} \tilde{p}(x) \overline{I}_{M_b}^{\mathrm{T}} \bar{A}^x_{d}. \end{aligned} \tag{18}$$

## V. Direct spectrum evaluation

If the spectral radius $\rho(P_0) < 1$, it is possible to compute the coefficients $A_d$ exactly by matrix inversion. This assumption on $P_0$ is not very restrictive, since if $\rho(P_0) > 1$ there could be an infinite number of non-converged pairs of paths at distance $d_{\mathrm{H}} = 0$, which implies infinite decoding delay. Several JSC-AC FSEs from [5] were tested and all had $\rho(P_0) < 1$.

*Proposition 3:* The vectors $\bar{A}^x_d = \sum_{n=d}^{\infty} \bar{A}^x_{d,n}$, leading to the spectral coefficients $A_d$ via (18), may be computed as

$$\bar{A}^x_d = [SP_1]^{d-1} S \bar{A}^x_{1,1} + [SP_1]^d S \bar{A}^x_{0,1} \qquad (d \ge 1), \tag{19}$$

where $\bar{A}^x_{0,1}$ and $\bar{A}^x_{1,1}$ are the initial values (12) in vector form, and

$$S = \sum_{k=0}^{\infty} P_0^k, \tag{20}$$

which converges for $\rho(P_0) < 1$ and may be evaluated as

$$S = (I - P_0)^{-1}. \tag{21}$$

*Proof:* The proof relies on the following.

*Lemma 4:*

$$\bar{A}^x_{d+1} = SP_1\bar{A}^x_d \qquad (d \geq 1). \tag{22}$$

To prove (22), expand the recursion (14) over $k$ steps in the variable $n$, from $n = d+1$ to $n = d+k+1$, and over one step in $d$, yielding

$$\bar{A}^x_{d+1,d+k+1} = \sum_{i=0}^{k} P_0^{k-i} P_1 \bar{A}^x_{d,d+i}. \tag{23}$$

In (23), the facts that $\bar{A}^x_{d,n} = 0$ for $n < d$ and $P_0^0 = I$ have been used. Then expand $\bar{A}^x_{d+1}$ as

$$\begin{aligned}\bar{A}^x_{d+1} &= \sum_{k=0}^{\infty} \bar{A}^x_{d+1,d+1+k} = \sum_{k=0}^{\infty}\sum_{i=0}^{k} P_0^{k-i} P_1 \bar{A}^x_{d,d+i} \\ &= \sum_{k=0}^{\infty} P_0^k P_1 \bar{A}^x_{d,d} + \sum_{k=1}^{\infty} P_0^{k-1} P_1 \bar{A}^x_{d,d+1} \\ &\quad + \sum_{k=2}^{\infty} P_0^{k-2} P_1 \bar{A}^x_{d,d+2} + \dots \end{aligned} \tag{24}$$

The sum (24) may be rearranged into

$$\bar{A}^x_{d+1} = \left(\sum_{k=0}^{\infty} P_0^k\right) P_1 \left(\sum_{i=0}^{\infty} \bar{A}^x_{d,d+i}\right) = SP_1\bar{A}^x_d,$$

proving Lemma 4, thanks to which it is sufficient to show

$$\bar{A}^x_1 = S\bar{A}^x_{1,1} + [SP_1]S\bar{A}^x_{0,1} \tag{25}$$

in order to prove (19). Using (20), the first term in (25) is recognized as the recursive summation of the first term in (14). For the second term in (25), first $\bar{A}^x_0 = S\bar{A}^x_{0,1}$ is obtained in the same fashion as the first term, then the same steps as for Lemma 4 are applied to get $[SP_1]\bar{A}^x_0$. ∎

## VI. Bounding the approximation error

In practice, the spectral coefficients $A_d$ are only computed up to a maximal distance $d_{\max}$ and, if recursion (11) is used instead of matrix inversion (21), the number of terms in the sum (10) is limited to some $N > d_{\max}$. Hence $A_d$ is only approximated from below (because all terms are non-negative) and so also (5) can only be approximated. Nevertheless, it is possible to upper-bound the approximation error and thus obtain a proper upper bound. For $d \leq d_{\max}$ and $N \geq d$, one may rewrite (18) as $A_d = \alpha_d(N) + \varepsilon_d(N)$, where

$$\alpha_d(N) = \sum_x \tilde{p}(x)\bar{I}^{\mathrm{T}}_{M_b}\bar{\alpha}^x_d(N) = \sum_x \tilde{p}(x)\bar{I}^{\mathrm{T}}_{M_b}\sum_{n=d}^{N}\bar{A}^x_{d,n}, \tag{26}$$

$$\varepsilon_d(N) = \sum_x \tilde{p}(x)\bar{I}^{\mathrm{T}}_{M_b}\bar{\varepsilon}^x_d(N) = \sum_x \tilde{p}(x)\bar{I}^{\mathrm{T}}_{M_b}\sum_{n=N+1}^{\infty}\bar{A}^x_{d,n} \tag{27}$$

are the $N$-term approximation and error, respectively. The latter can be bounded with

$$\varepsilon_d(N) \leq \sum_x \tilde{p}(x)\|\bar{\varepsilon}^x_d(N)\|_1, \tag{28}$$

where the norm $\|\bar{A}\|_1 = \sum_i |\bar{A}_i|$ for vectors $\bar{A}$ and $\|P\|_1 = \max_j \sum_i |P_{ij}|$ for matrices $P$. Using reasoning similar to that leading to (25), one can show

$$\bar{\varepsilon}^x_d(N) = SP_0\bar{A}^x_{d,N} + SP_1\left(\bar{A}^x_{d-1,N} + \bar{\varepsilon}^x_{d-1}(N)\right), \tag{29}$$

where the factor $P_0$ in the first term accounts for the fact that the sum (27) starts with $\bar{A}^x_{d,N+1}$.

Because $d_{\text{free}}$ dominates the performance at high SNR, a key issue in the recursive approximation of the spectrum via (11) or (14) is to determine when $d_{\text{free}}$ has been found, *i.e.*, when no unmerged pair of $N$-bit paths could lower the smallest $d$ such that $A_d > 0$. A sufficient condition was given in [5], which adapted to the present framework states that when $N$ is such that $A_d = 0$ and $\|\bar{A}^x_{d,N}\|_1 = 0$ for $d = 0, \dots, d^* - 1$, $x \in \mathcal{S}$, while $A_{d^*} > 0$, then $d_{\text{free}} = d^*$ has been found. Let $N$ satisfy these conditions in the following, implying $\varepsilon_d(N) = 0$ for $d < d_{\text{free}}$. Then (29) may be rewritten as

$$\bar{\varepsilon}^x_d(N) = SP_0\bar{A}^x_{d,N} + \sum_{i=d_{\text{free}}}^{d-1}[SP_1]^{d-i}S\bar{A}^x_{i,N} \qquad (d_{\text{free}} \leq d \leq d_{\max}). \tag{30}$$

For $d > d_{\max}$, the error may be computed using (22) as

$$\begin{aligned}\bar{\varepsilon}^x_d(N) = \bar{A}^x_d &= [SP_1]^{d-d_{\max}}\bar{A}^x_{d_{\max}} \\ &= [SP_1]^{d-d_{\max}}\left(\bar{\alpha}^x_{d_{\max}}(N) + \bar{\varepsilon}^x_{d_{\max}}(N)\right)\end{aligned} \tag{31}$$

Since the norm $\|\cdot\|_1$ is sub-additive and sub-multiplicative, (30) in conjunction with (28) suffices to bound the approximation error for $d = d_{\text{free}}, \dots, d_{\max}$ using the quantities $\|P_0\|_1$, $\|P_1\|_1$, $\|S\|_1$, $\|\bar{A}^x_{d,N}\|_1$ and $\|\bar{\alpha}^x_{d_{\max}}(N)\|_1$. These are all readily computed during the recursive spectrum evaluation, except for $S$, which may be upper bounded as follows.

*Lemma 5:* If the spectral radius of $P_0$ satisfies $\rho(P_0) < 1$, then for any induced matrix norm $\|\cdot\|$ there exists $n_0 > 0$ such that for all $n \geq n_0$, one has

$$\|S\| \leq \frac{\left\|\sum_{i=0}^{n} P_0^i\right\|}{1 - \left\|P_0^{n+1}\right\|}. \tag{32}$$

*Proof:* Since $\rho(P_0) < 1$, for any induced matrix norm there exists $n_0 > 0$ such that $\|P_0^n\| < 1$ for any $n \geq n_0$. One may write (20) as

$$S = \sum_{i=0}^{n} P_0^i + \sum_{i=n+1}^{\infty} P_0^i = \sum_{i=0}^{n} P_0^i + P_0^{n+1}\sum_{i=0}^{\infty} P_0^i.$$

Then, for any $n \geq n_0$, the lemma follows from

$$\begin{aligned}\|S\| &\leq \left\|\sum_{i=0}^{N} P_0^i\right\| + \left\|P_0^{n+1}\right\| \left\|\sum_{i=0}^{\infty} P_0^i\right\| \\ &= \left\|\sum_{i=0}^{N} P_0^i\right\| + \left\|P_0^{n+1}\right\| \|S\|.\end{aligned}$$

∎

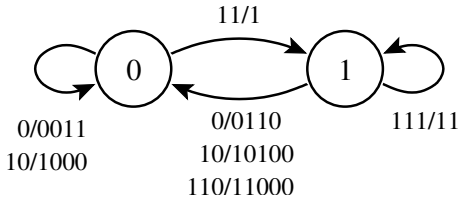


Fig. 1. Graph of a finite-state joint-source channel arithmetic encoder for a Bernoulli source with $\Pr(X=0) = 0.11$. Average rate 1.01 bits/symbol, $d_{\text{free}} = 2$, $A_2 = 0.44$.

The tail sum for $d > d_{\max}$ in the union bound (5) may also be bounded, since the error term (31) grows polynomially, while the channel error probability (8) decays exponentially. Let $\beta \geq \sum_{x\in\mathcal{S}} \left( \left\| \bar{\alpha}^x_{d_{\max}}(N) \right\|_1 + \left\| \bar{\varepsilon}^x_{d_{\max}}(N) \right\|_1 \right)$ and $\kappa \geq \|SP_1\|_1$ be upper bounds obtained as outlined above, and let $\gamma = \frac{E_b}{N_0} - \log\kappa$. Then

$$\sum_{d=d_{\max}+1}^{\infty} A_d P_d \leq \beta \sum_{d=d_{\max}+1}^{\infty} \kappa^{d-d_{\max}} \delta e^{-d\frac{E_b}{N_0}} = \frac{\beta\delta e^{-\gamma d_{\max}}}{\kappa^{d_{\max}}\left(e^{\gamma}-1\right)} \qquad (\gamma > 0), \tag{33}$$

where $\delta = \frac{1}{2}\operatorname{erfc}\left(\sqrt{d_{\text{free}}\frac{E_b}{N_0}}\right)\exp\left(d_{\text{free}}\frac{E_b}{N_0}\right)$ stems from tightening the bound (8) as in [1, Ch. 4.5]. The tail bound converges only if $\gamma > 0$, *i.e.*, if the channel SNR is larger than $\log\kappa$. In many settings this turns out to be problematic, since $\|SP_1\|_1$ and thus $\kappa$ may be quite large ($\kappa \approx 30$ for the example FSC in Sec. VII), making it impossible to obtain a useful (tight) bound at practical SNR levels. One may choose to neglect the tail sum if $d_{\max}$ is large enough; a slightly preciser method consists in reducing $\kappa$ to a value that is closer to the asymptotic growth rate $\kappa_\infty = \lim_{d\to\infty}\frac{A_{d+1}}{A_d}$. Empirically, this was found to be close to the spectral radius $\rho(SP_1)$ already for small values of $d > d_{\text{free}} + 1$ ($\kappa_\infty \approx 2.6$ for the FSC in Section VII).

## VII. Experimental results

Figure 1 shows the encoder graph of a JSC-AC for a Bernoulli source with $\Pr(X{=}0) = 0.11$, obtained as outlined in [5]. Simulation results (for blocks of 1024 symbols, BPSK over AWGN and ML sequence (Viterbi) decoding) are compared with approximate and proper upper bounds in Figures 2 and 3, confirming the empirical "rule of thumb" that relatively few terms are necessary to obtain usable approximate bounds (notice that for large enough $N$ the bound using recursive approximation can be close to that using matrix inversion). Unfortunately, the tail bound (33) provides rather gross overestimates even when using the empirical growth rate $\kappa_\infty$.

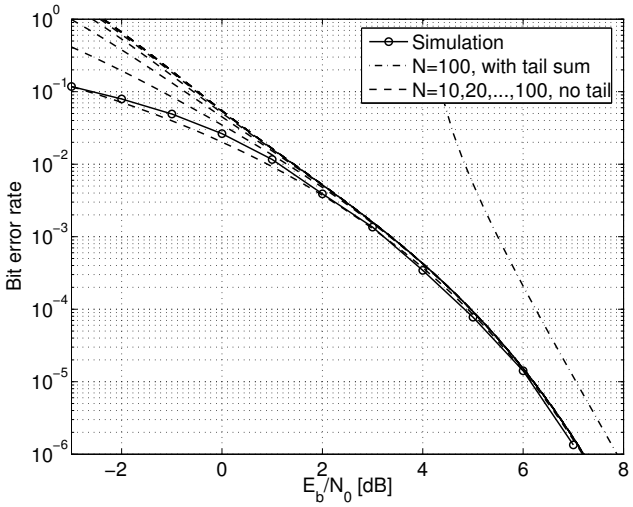


Fig. 2. Recursive approximation of the union bound: partial sum of (5) up to $d_{\max} = 10$, using partial sums of (10) up to $n = N$, where $N = 10, 20, \ldots, 100$ (evaluated with (11)); proper upper bound including bounded error terms (30) and tail sum (33) with $d_{\max}{=}10$, $N{=}100$, $\kappa_\infty{=}2.6$.

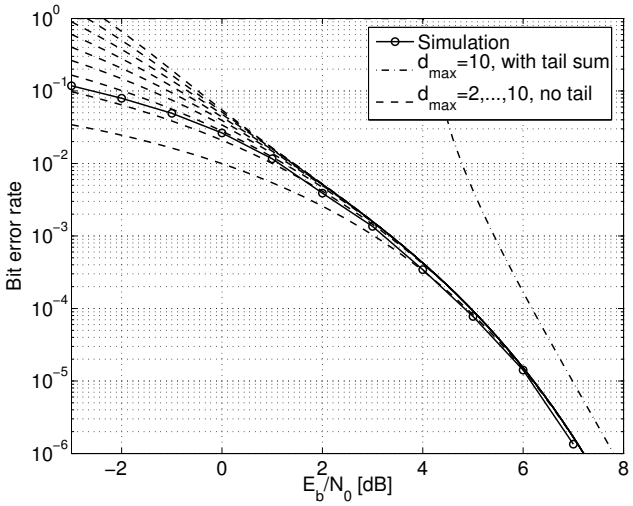


Fig. 3. Approximation of the union bound using matrix inversion: partial sums of (5) up to $d{=}d_{\max}$, where $d_{\max}{=}2, 3, \ldots, 10$; proper upper bound including tail sum (33) with $d_{\max}{=}10$, $\kappa_\infty{=}2.6$.

## References


[1] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New-York: McGraw-Hill, 1979.

[2] E. Biglieri, "High-level modulation and coding for nonlinear satellite channels," *IEEE Trans. Commun.*, vol. COM-32, no. 5, pp. 616–626, May 1984.

[3] E. Zehavi and J. K. Wolf, "On the performance evaluation of trellis codes," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 2, pp. 196–202, Mar. 1987.

[4] V. Buttigieg, "Variable-length error correcting codes," PhD dissertation, University of Manchester, Manchester, U.K., 1995. [Online]. Available: http://staff.um.edu.mt/vbut1/research/PhDThesis.pdf

[5] S. Ben-Jamaa, C. Weidmann, and M. Kieffer, "Analytical tools for optimizing the error correction performance of arithmetic codes," *IEEE Trans. Commun.*, vol. 56, no. 9, pp. 1458–1468, Sep. 2008.

[6] A. Hedayat and A. Nosratinia, "Performance analysis and design criteria for finite-alphabet source-channel codes," *IEEE Trans. Commun.*, vol. 52, no. 11, pp. 1872–1879, Nov. 2004.

[7] X. Jaspar and L. Vandendorpe, "Design and performance analysis of joint source-channel turbo schemes with variable length codes," in *Proc. ICC*, vol. 1, Seoul, Korea, May 2005, pp. 526–530.

[8] F. Pollara, R. J. McEliece, and K. Abdel-Ghaffar, "Finite-state codes," *IEEE Trans. Inform. Theory*, vol. 34, no. 5, pp. 1083–1089, Sep. 1988.

[9] P. Piret, "Comma free error correcting codes of variable length, generated by finite-state encoders," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 5, pp. 764–775, Sep. 1982.