Evaluation of the Very Low BER of FEC Codes Using Dual Adaptive Importance Sampling

R. Holzlöhner, A. Mahadevan, C. R. Menyuk, J. M. Morris, and J. Zweck

***Abstract*— We evaluate the error-correcting performance of a low-density parity-check (LDPC) code in an AWGN channel using a novel dual adaptive importance sampling (DAIS) technique based on multicanonical Monte Carlo (MMC) simulations, that allows us to calculate bit error rates as low as 10^{-19} for a $(96, 50)$ LDPC code without a priori knowledge of how to bias. Our results agree very well with standard MC simulations, as well as the union bound for the code.**

***Index Terms*— Very low BER, Multicanonical Monte Carlo, Importance sampling, LDPC codes.**

I. Introduction

THE accurate computation of very low bit error rates (BERs) for forward error correcting (FEC) codes depends on sampling the very rare noise realizations that lead to errors in the decoded bit sequence. These rare events cannot be adequately sampled using standard Monte Carlo (MC) simulations. Importance sampling (IS) [1] has been used to enhance the occurrence of these events in FEC codes. However, for IS to be effective, a biased distribution must be chosen using some knowledge of which noise realizations most likely generate errors. This task is difficult when iterative decoding algorithms are used, since codeword errors are correlated to the noise distribution among the bits in a highly complex way.

We apply the multicanonical Monte Carlo (MMC) simulation technique of [2] as the basis for a dual adaptive importance sampling (DAIS) technique to compute very low BERs. We demonstrate the DAIS technique using a $(96, 50)$ low-density parity-check (LDPC) code and sum-product decoding (SPD) [3] with up to 50 decoder iterations, achieving BER $\approx 10^{-19}$. Like standard IS [1], MMC increases the number of events in the tail of the pdf being computed by sampling from a biased pdf [4]. The advantage of MMC is that it adaptively iterates to this biased pdf with little a priori knowledge needed of how to bias. The iterative procedure uses a control quantity to update the next iteration's biased pdf so that, as the iteration number increases, there tends to be an approximately equal number of hits in each control-quantity histogram bin [2].

II. Simulation Procedure and Results

We study the performance of a regular $(96, 50)$ LDPC code with a code rate of $R = 50/96$ in an AWGN channel with BPSK using DAIS. The parity-check matrix ($\mathbf{H}$) of this code can be found at [5]. We choose this code because it was: (1) the same code studied in [1], and (2) possible to exactly compute the first two non-trivial coefficients of the code's weight enumerator function (3 and 24 codewords at Hamming weights 6 and 8, respectively), which allows us to compare our simulation results with the code's union-bound performance [6]. Note that both [1] and [5] refer to this code as a $(96, 48)$ code, but 2 of the 48 rows of $\mathbf{H}$ are linearly dependent.

We implemented SPD [3] employing the log-likelihood modification of [7], and symmetric signal levels of $+1$ and -1 for logical 1s and 0s, respectively. The pdf ρ_l of the noise in the lth bit at the receiver is zero mean Gaussian with $\sigma^2 = 1/(2R\,E_b/N_0)$ [3]. It suffices to transmit the all-zeros codeword, since the code is linear and the noise is symmetric.

Let Γ be the n-dimensional probability space of the noise in the n bits of a codeword. The noise vector $\mathbf{z} = (z_1, \ldots, z_n)$ is multivariate Gaussian with joint pdf $\rho(\mathbf{z}) = \prod_{l=1}^{n} \rho_l(z_l)$. The MMC algorithm is controlled by a scalar control quantity V defined here as $V(\mathbf{z}) = \left\{ \frac{1}{n} \sum_{l=1}^{n} \left[H(q_l z_l)\, z_l \right]^2 \right\}^{1/2}$, where $q_l = (-1)^{b_l}$ with b_l being the transmitted bit in the lth position, and $H(x) = 1$ if $x > 0$ and $H(x) = 0$ otherwise. We constructed $V(\mathbf{z})$ so that a noise component z_l contributes to V only if it may produce a bit-error at the input to the decoder. We say that a received word with a noise realization $\mathbf{z}$ generates an error, if the LDPC decoder cannot decode it to the transmitted codeword within 50 iterations.

Given a range $[V_{\min}, V_{\max}]$ for V, we partition Γ into M subsets $\Gamma_k = \{\mathbf{z} \in \Gamma \,|\, V_{k-1} \le V(\mathbf{z}) < V_k\}$, where $V_k = V_{\min} + k\Delta V$, $1 \le k \le M$, and $\Delta V = V_k - V_{k-1} = (V_{\max} - V_{\min})/M$ is the width of each bin in the partition of $[V_{\min}, V_{\max}]$. Let P_k be the probability of selecting a realization $\mathbf{z}$ from ρ so that $\mathbf{z} \in \Gamma_k$ [4], [8]. Then,

$$P_k = \int_{\Gamma} \chi_k(\mathbf{z}) \frac{\rho(\mathbf{z})}{\rho^*(\mathbf{z})} \rho^*(\mathbf{z})\, d\mathbf{z} \approx \frac{1}{N} \sum_{i=1}^{N} \chi_k(\mathbf{z}^{*,i}) \frac{\rho(\mathbf{z}^{*,i})}{\rho^*(\mathbf{z}^{*,i})}, \tag{1}$$

where $\rho^*(\mathbf{z})$ is a positive biasing pdf, $\chi_k(\mathbf{z}) = 1$ if $\mathbf{z} \in \Gamma_k$ and $\chi_k(\mathbf{z}) = 0$ otherwise, and the $\mathbf{z}^{*,i}$ are N random sample points in Γ, selected according to the pdf $\rho^*(\mathbf{z})$. The variance of the estimate of (1) is zero if the optimal biasing pdf $\rho^*_{\text{opt}}(\mathbf{z}) = \chi_k(\mathbf{z})\rho(\mathbf{z})/P_k$ is used. However, $\rho^*_{\text{opt}}(\mathbf{z})$ depends on P_k, which is initially unknown. In standard IS, one uses physical intuition to guess a biasing pdf that is close to ρ^*_{opt}. The MMC algorithm instead iterates over a sequence of biasing pdfs $\rho^{*,j}$ that approach ρ^*_{opt}. We define $\rho^{*,j}$ for the jth iteration by $\rho^{*,j}(\mathbf{z}) = \rho(\mathbf{z})/(c^j P_k^j)$, where k is such that $\mathbf{z} \in \Gamma_k$ is satisfied. The quantities P_k^j satisfy $P_k^j > 0$ and

Manuscript received 30, 2004. The associate editor coordinating the review of this letter and approving it for publication was Prof. M. Fossorier. This work was supported by the Laboratory of Telecommunications Sciences under contract MDA 904-02-C0428/Z956801.

The authors are or were with the Computer Science and Electrical Engineering Department, University of Maryland Baltimore County (UMBC), Baltimore, MD, USA (email: ronald.holzloehner@web.de, {mahadevan, morris}@umbc.edu).

Digital Object Identifier 10.1109/LCOMM.2005.xxxxxx.

$\sum_{k=1}^{M} P_k^j = 1$, and c^j is an unknown constant that ensures $\int_\Gamma \rho^{*,j}(\mathbf{z})\, d\mathbf{z} = 1$. The vector $P_k^j,\ k = 1, \ldots, M$, is the key quantity in the MMC algorithm and completely determines the bias. At the first MMC iteration, P_k^1 is usually set to $1/M,\ \forall k = 1, \ldots, M$.

Within each MMC *iteration* j, we employ the Metropolis algorithm [9] to produce a random walk of *samples* $\mathbf{z}^{*,i}$ whose pdf equals $\rho^{*,j}(\mathbf{z})$. We consider a Markov chain of transitions consisting of small steps in the noise space. Each transition goes from $\mathbf{z}^{*,i} = \mathbf{z}_a^* \in \Gamma_{k_a}$ to $\mathbf{z}_b^* = (\mathbf{z}_a^* + \epsilon^j \Delta\mathbf{z}) \in \Gamma_{k_b}$, where $\Delta\mathbf{z}$ is random and symmetric, *i.e.*, it does not favor any direction in Γ, and the transition is accepted with probability π_{ab}. If a transition from $\mathbf{z}^{*,i}$ to $\mathbf{z}_b^*$ is accepted, we set $\mathbf{z}^{*,i+1} = \mathbf{z}_b^*$, else we set $\mathbf{z}^{*,i+1} = \mathbf{z}^{*,i} = \mathbf{z}_a^*$. The ratio π_{ab}/π_{ba} equals $\rho^{*,j}(\mathbf{z}_b^*)/\rho^{*,j}(\mathbf{z}_a^*)$, which is the *detailed balance condition* that ensures that the limiting (stationary) pdf for infinitely many steps of this random walk is $\rho^{*,j}$ [9].

We consider the perturbation of the noise component in each bit $z_{a,l}^*$ of $\mathbf{z}_a^*$ separately, and accept or reject it independently with the probability $\min\left[\rho_l(z_{b,l}^*)/\rho_l(z_{a,l}^*),\, 1\right]$. We pick each perturbation Δz_l from a zero mean symmetric pdf. We obtain a trial state $\mathbf{z}_b^*$ in which only some of the components are different from their previous values in $\mathbf{z}_a^*$. Next, we compute k_b, the bin corresponding to $\mathbf{z}_b^*$, and finally accept the step from $\mathbf{z}_a^*$ to $\mathbf{z}_b^*$ with the probability $\min\left[P_{k_a}^j/P_{k_b}^j,\, 1\right]$.

In each iteration, the perturbation coefficient ϵ^j is constant for all samples. After each iteration, we adjust ϵ^j so that the *acceptance ratio* $\alpha \triangleq$ (number of accepted steps)/(total number of steps, N) is close to 0.3 (empirically chosen based on experience from previous experiments). The minimum required N for this random walk depends on the average step size $\alpha\epsilon^j \langle|\Delta\mathbf{z}|\rangle$ and hence is system-dependent. The noise realizations are recorded in the histogram $H^{*,j}$, where $H_k^{*,j} = \sum_{i=1}^{N} \chi_k(\mathbf{z}^{*,i})$ is the number of the $\mathbf{z}^{*,i}$ in iteration j that fall into Γ_k. The P_k^j are updated after each MMC iteration using the recursion relations given in [8] based on the histogram $H^{*,j}$. As j increases, the expected number of samples $\langle H_k^{*,j}\rangle$ becomes independent of the bin number k, which implies that $P_k^j \to P_k$.

Let P_{err} be the probability that a received word with noise realization $\mathbf{z}$ selected from ρ leads to an error, and $P_{\text{err},k}$ the probability that $\mathbf{z}$ leads to an error *and* falls into bin k. Then

$$P_{\text{err},k} = P_{\text{err}|k}\, P_k = P_{k|\text{err}}\, P_{\text{err}}, \tag{2a}$$

$$P_{\text{err}} = \sum_{k=1}^{M} P_{\text{err},k}, \tag{2b}$$

where $P_{\text{err}|k}$ and $P_{k|\text{err}}$ are the conditional probabilities of an error *given* that $\mathbf{z}$ falls into bin k, and vice versa. We can compute P_{err} by first running an MMC simulation as described above, where we also count the errors in bin k to produce a histogram $G_k^{*,j}$. We can then approximate $P_{\text{err}|k} \approx P_{\text{err}|k}^{j_{\max}} = \sum_{j=1}^{j_{\max}} G_k^{*,j} / \sum_{j=1}^{j_{\max}} H_k^{*,j}$ after $j_{\max}$ MMC iterations. Summing over all MMC iterations is valid since the biasing pdf at any MMC iteration only affects the total number of hits in a bin, but not the behavior of error hits relative to the total hits within a bin. Finally, we can use the left equation of (2a), and equation (2b) to get P_{err}. In Fig. 1(a) we show the approximation $(P_{\text{err}|k}^{j_{\max}}\, P_k^{j_{\max}+1})/\Delta V$ to $P_{\text{err},k}/\Delta V$ with dots for $E_b/N_0 = 11$ dB. The dashed line shows $P_k/\Delta V$, and the circles show the sum of the error histograms $\sum_{j=1}^{j_{\max}} G_k^{*,j}$. The number of sampled errors rapidly decreases to 0 as V decreases towards 0.4, which is where $P_{\text{err},k}$ tends to be largest. Consequently, the approximation $P_{\text{err}|k} \approx P_{\text{err}|k}^{j_{\max}}$ converges very slowly as the iteration number j increases. The reason is that in this *unconstrained* MMC simulation, we have not sampled enough of the higher-probability smaller-noise realizations that generate errors.

One efficient method to overcome this undersampling problem is to run a second, *constrained* MMC simulation (hence the term dual in DAIS), in which we only accept Metropolis steps that lead to errors. If a trial realization $\mathbf{z}_b^*$ does not yield an error in this simulation, we set π_{ab} to zero. The constrained simulation, hence, takes its samples from $\tilde{\rho}(\mathbf{z}) = \chi_{\text{err}}(\mathbf{z})\rho(\mathbf{z})/P_{\text{err}}$, where $\chi_{\text{err}}(\mathbf{z}) = 1$ if $\mathbf{z}$ produces an error and $\chi_{\text{err}}(\mathbf{z}) = 0$ otherwise. Note that $\tilde{\rho}(\mathbf{z})$ is proportional to $\rho(\mathbf{z})$ wherever $\chi_{\text{err}}(\mathbf{z}) = 1$. If the Metropolis random walk is ergodic in the error subset of Γ, the constrained MMC simulation approximates $P_{k|\text{err}}$. Since the $P_{\text{err},k}$ and $P_{k|\text{err}}$ estimates obtained using the two simulations are both smooth for large k, using (2a) we can obtain $P_{\text{err}} = P_{\text{err},k}/P_{k|\text{err}}$ from the data where k is large. In Fig. 1(a), the dash-dot line shows $P_{k|\text{err}}/\Delta V$ obtained from the constrained simulation, while the solid line shows the resulting $P_{\text{err},k}/\Delta V$ obtained by scaling $P_{k|\text{err}}/\Delta V$ to fit $P_k/\Delta V$ from the unconstrained simulation for $0.55 < V < 0.6$. Since MMC yields a similar number of samples in each bin, the relative statistical sampling error of $P_{k|\text{err}}$ in the constrained simulation is smaller at small V than in the unconstrained simulation. A significant advantage of running separate unconstrained and constrained simulations is that the algorithm optimizes the perturbation coefficients ϵ^j of the two simulations independently. The values of ϵ^j tend to differ strongly between the two simulations.

In our simulations, $M = 300$. In the first iteration $N^{j=1} = 5000$ samples in the unconstrained case and $N^{j=1} = 10,000$ in the constrained case, and we increase the number of samples after each iteration so that $N^{j+1} = 1.3\, N^j$. In each case, $P_k^1 = 1/M,\ k = 1, \ldots, M$, and we assume the simulation to have sufficiently converged when $\max_k |(P_k^j - P_k^{j+1})/P_k^{j+1}| < 0.1$. This convergence requires $\approx 10^6$ to 10^8 samples in total, with the samples increasing on average with increasing E_b/N_0. Also, in both cases, we initialize each MMC iteration with a $\mathbf{z}$ that gives a decoder error.

In Fig. 1(b), the $\times$ and $+$ symbols denote the decoder output BER and WER estimates, respectively, obtained via MC. The dashed curve with $\square$ and dash-dot curve with $\circ$ denote the decoder output BER and WER estimates, respectively, obtained using DAIS. Finally, the solid curve and dotted curve denote the BER and WER union bounds, respectively [6].

The union bound can be closely approximated at high E_b/N_0 by the contribution of low Hamming weight (6 and 8 in this case) codewords. The SPD for LDPC codes approximates the ML decoder [10]. Hence, we would expect the SPD to perform worse than the union bound on ML decoding at high E_b/N_0. Our results from DAIS are consistent with this expectation and indicate that DAIS can simulate WER and BER performance of codes at very low values. We also

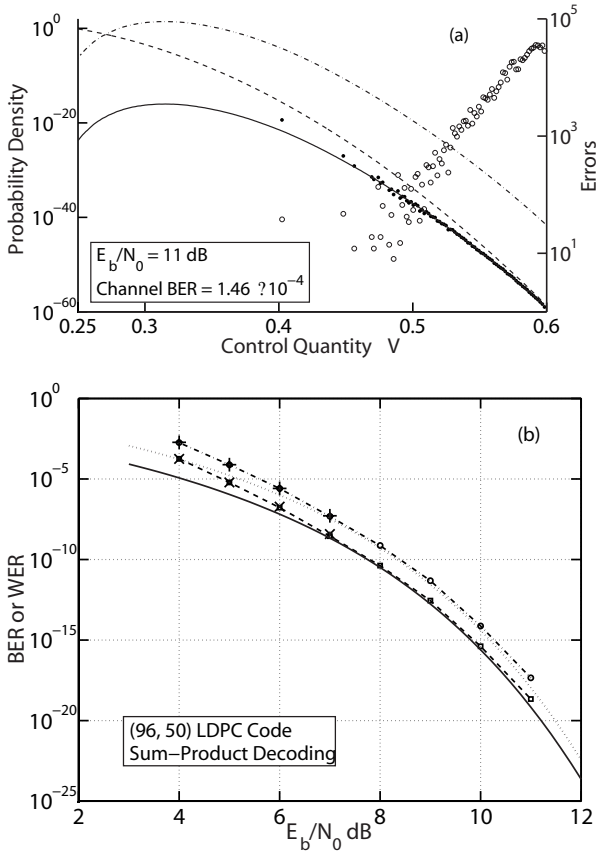

Fig. 1. (a) Dashed curve: Pdf of V. Dots: joint pdf of V and errors, both from the unconstrained simulation. Dash-dot curve: pdf of V conditioned on errors from constrained simulation. Solid curve: joint pdf of V and errors obtained by scaling dash-dot curve to fit the dots for $V > 0.55$. Circles: Number of decoder errors in the unconstrained simulation. (b) Cross and plus symbols: MC BER and WER estimates, respectively. Dashed curve with squares and dash-dot curve with circles: DAIS BER and WER, respectively. Solid and dotted curves: BER and WER union bound approximations, respectively, based on codewords at Hamming weight 6 and 8 [6].

observe excellent agreement between the results obtained by DAIS and MC, wherever MC results are available (DAIS falls within the 99% error bars for MC), which further validates DAIS.

Our argument that the true code performance should be close to the union bound at high E_b/N_0 is further bolstered by the observation that for MC simulations, as E_b/N_0 increases, the contribution of the probability of decoding-to-wrong-codewords progressively dominates the WER. For example, at $E_b/N_0 = 4\,\mathrm{dB}$, 216 of 1888 word errors recorded were due to decoding to wrong codewords (the rest were decoder failures), whereas at $E_b/N_0 = 7\,\mathrm{dB}$, the corresponding numbers were 40 of 52. Note that the BER results in [1] are farther away from the union bound than our results (by about 0.4 dB at BER $= 10^{-9}$), which may be attributed to their use of ≤ 5 iterations for the SPD, and possibly a different decoder implementation.

We note that our BER data points do not show a waterfall region since they correspond to large E_b/N_0 relative to the Shannon limit (≈ 0 dB for our code), and since the code is not very long. We earlier obtained BER estimates down to 10^{-39} for a smaller $(20, 7)$ code, but space limitations preclude presenting these results.

A measure of DAIS's gain over MC is given by the ratio of the number of samples (codewords) required to achieve a given WER at a given E_b/N_0; e.g., at $E_b/N_0 = 10$ dB, WER $\approx 10^{-14}$ is obtained by DAIS using 8×10^7 codewords (unconstrained) $+ 3 \times 10^7$ codewords (constrained) $= 11 \times 10^7$ codewords (total), whereas MC would require $\geq 10^{15}$ codewords (assuming ≥ 10 word error events). Thus the gain is $\frac{10^{15}}{11 \times 10^7} \approx 9 \times 10^6$. Our current experience suggest that DAIS's gain increases with decreasing WER, but the accuracy of DAIS as an estimator, and its dependence on the number of codewords or codeword length, is unknown at this time, and a subject of continuing research.

We are currently studying DAIS for longer codes. As code length increases, dimensionality of Γ and its partitions that map to bins of V increases. Hence, maintaining a given level of statistical accuracy in sampling each partition of Γ requires more samples for the longer code.

III. Summary and Conclusions

We presented a dual adaptive importance sampling (DAIS) technique based on multicanonical Monte Carlo (MMC) simulations, and used it to study an LDPC code in an AWGN channel. It allows us to calculate very low WERs and BERs. In contrast to standard importance sampling [1], the MMC algorithm iteratively approaches the optimal bias without a priori knowledge of how to bias. We improve the WER and BER estimates by combining the results of two MMC simulations in the large noise regions where statistical uncertainty due to sampling from the biased pdf is smallest. In one simulation, we approximate the probability of decoder errors in the large noise regions. In a second complementary simulation, we constrain the MMC random walk to the noise region that produces decoder errors. Our WER and BER results are consistent with standard Monte Carlo simulations, and the union bound on maximum-likelihood decoding [6].

References

[1] B. Xia and W. E. Ryan, "On importance sampling for linear block codes," *Proc. IEEE International Conference on Communications 2003 (ICC '03)*, pp. 2904-2908.

[2] B. A. Berg and T. Neuhaus, "The multicanonical ensemble: a new approach to simulate first-order phase transitions," *Phys. Rev. Lett.*, vol. 68, pp. 9-12, 1992.

[3] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, pp. 457-458, 1997.

[4] R. Holzlöhner and C. R. Menyuk, "Use of multicanonical Monte Carlo simulations to obtain accurate bit error rates in optical communications systems," *Opt. Lett.*, vol. 28, pp. 1894-1896, Oct. 2003.

[5] D. J. C. MacKay, "Encyclopedia of Sparse Graph Codes," http://www.inference.phy.cam.ac.uk/mackay/codes/EN/C/96.3.963.

[6] M. P. C. Fossorier, S. Lin, and D. D. Rhee, "Maximum-likelihood decoding of linear block codes and related soft-decision decoding methods," *IEEE Trans. Inform. Theory*, vol. 44, pp. 3083-3090, 1998.

[7] H. Futaki and T. Ohtsuki, "Performance of low-density parity-check (LDPC) coded OFDM systems," *Proc. ICC '02*, vol. 3, pp. 1696-1700.

[8] B. A. Berg, "Algorithmic aspects of multicanonical Monte Carlo simulations," *Nucl. Phys. Proc. Suppl.*, vol. 63, pp. 982-984, 1998.

[9] N. Metropolis *et al.*, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087-1092, 1953.

[10] H. Steendam and M. Moeneclaey, "ML-Performance of low-density parity check codes," *Proc. BENELUX-IT, 23^{rd} Symp. on Inform. Theory in the Benelux*, 2002, pp. 75-78.