CONVOLUTIONAL VERSUS LDPC AND TURBO CODES ON THE RAYLEIGH FADING CHANNEL

Graduate Students: Kristin Jagiello and Charlie Cooper
Faculty Advisors: William E. Ryan and Michael W. Marcellin
ECE Dept., University of Arizona, Tucson, AZ 85721
(kristinj, ryan, marcellin)@ece.arizona.edu

ABSTRACT

We consider the performance of low-density parity-check (LDPC) codes, turbo codes and convolutional codes over the binary-input AWGN channel with flat Rayleigh fading. LDPC and turbo codes are capacity-approaching codes for long codewords. For short and medium codewords we seek to determine if they still outperform the industry-standard memory-6, rate-1/2 convolutional code. For a fixed SNR, the probability of error for the codes of interest are plotted as a function of codelength. We find that for very short codewords, the convolutional code performs best.

1. INTRODUCTION

LDPC and turbo codes are known to be capacity-approaching codes for long codewords. As the length of the codeword is decreased, the performance gain over a convolutional code is reduced. We consider these three classes of channel codes on the binary-input AWGN channel with flat Rayleigh fading. For various codelengths we are interested in determining which code class performs best. Results show that for very short codewords (lengths $n =$ 408 or shorter), the convolutional code outperforms the LDPC and turbo code.

Section II describes the system models used in the simulations. Section III provides details of the channel codes used in this study. Section IV presents simulation results and Section V concludes the paper.

2. SYSTEM MODELS

Figure 1 depicts the system model used in our simulations. The three channel codes used were convolutional, LDPC and turbo. The interleavers for the various code lengths were designed using the "S-random" design algorithm [1]. Interleaving is done to help mitigate correlated fading. The channel model assumed here is the binary-input AWGN channel with

flat Rayleigh fading. Specifically, the model used is the slowly varying flat fading channel with a coherent receiver.

A bandpass signal can be expressed as

$$s(t) = \mathrm{Re}\left[g(t)e^{j2\pi f_c t}\right], \tag{1}$$

where $g(t)$ is the complex envelope and f_c is the carrier frequency. When a signal is broadcast from a transmitter, it undergoes multipath fading due to reflections as well as Doppler broadening if there is motion between the transmitter and receiver. For relative velocity v, wavelength λ and angle of incidence $\theta_n(t)$, the Doppler frequency is expressed as

$$f_{D,n}(t) = \frac{v}{\lambda}\cos\theta_n(t) = f_m \cos\theta_n(t). \tag{2}$$

As a result, the received signal will be the summation of different paths. For N paths, the signal is expressed as

$$x(t) = \sum_{n=1}^{N} \alpha_n(t)s(t-\tau_n(t)) = \mathrm{Re}\left[\sum_{n=1}^{N} \alpha_n(t)e^{-j\phi_n(t)}g(t-\tau_n(t))e^{j2\pi f_c t}\right], \tag{3}$$

where $\tau_n(t)$ are the time varying delays for the different paths and

$$\phi_n(t) = 2\pi\left\{[f_c + f_{D,n}(t)]\tau_n(t) - f_{D,n}(t)t\right\}. \tag{4}$$

From equation (3), the baseband equivalent channel impulse response can be written as

$$c(\tau, t) = \sum_{n=1}^{N} \alpha_n(t)e^{-j\phi_n(t)}\delta(t-\tau_n(t)), \tag{5}$$

and the complex envelope of $x(t)$ as

$$\hat{x}(t) = \sum_{n=1}^{N} \alpha_n(t)e^{-j\phi_n(t)}g(t-\tau_n(t)). \tag{6}$$

Because the path delays due to the reflections are randomly changing with time, a random process may be used to model the channel response $c(\tau, t)$. If there is no line-of-sight path, the process is zero mean and the envelope $|c(\tau, t)|$ of the channel response has a Rayleigh distribution expressed as

$$p_\alpha(x) = \frac{2x}{\Omega}e^{x^2/\Omega} \quad \mathrm{for} x \geq 0, \tag{7}$$

where $\Omega = E\{\alpha^2\}$ is the average power of the impulse response. From the channel impulse response $c(\tau, t)$, it can be seen that the signal will undergo fading. The sum of the N paths will add up constructively or destructively as the phase $\phi(t)$ varies randomly with time for each path.

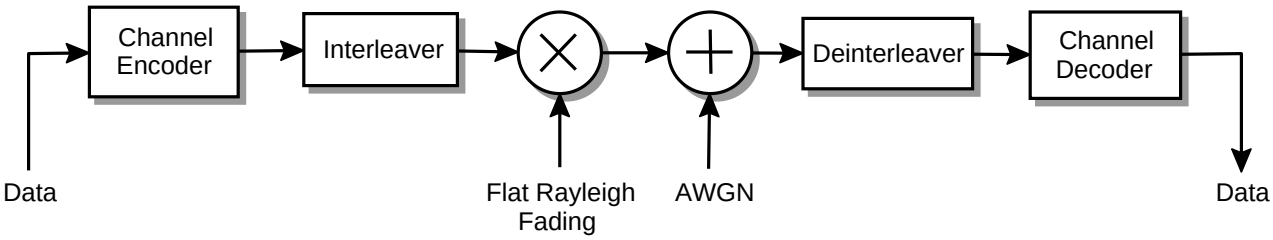

Figure 1. A block diagram of the system model.

A flat fading channel is frequency non-selective, meaning that all frequencies in the signal are affected identically. The multiplicative form of the model seen if Figure 1 comes from this fact. A channel is slowly varying if the fading is slow with respect to the duration of a single channel symbol. For this type of channel, amplitude attenuation and phase distortion are essentially constant during a symbol interval. Coherent detection can be implemented to remove phase distortion due to drift of the carrier. The resulting discrete-time channel model is expressed as

$$r_k = \alpha_k s_k + n_k$$

where $s_k \in \{\pm 1\}$ is a channel symbol conveying one code bit, α_k is the random amplitude distortion and n_k is noise. The random variables α_k are Rayleigh distributed and can be generated using various methods. Here we use the Jakes simulator [3]. It produces correlated Rayleigh fading by modeling a multipath environment as a sum of sinusoids. For our simulations correlation is maintained for 1×10^{12} bits. The parameter $f_m T$ is often used to describe this type of fading. The maximum Doppler frequency shift is f_m and T is the channel symbol interval. Smaller values of $f_m T$ correspond to worse channel conditions [2], [3]. Here we use $f_m T = 0.004$. Figure 2 is a plot of the autocorrelation function for the output of the Jakes simulator.

3. CODE SIMULATIONS

3.1. Convolutional Code

The convolutional code used was the industry standard rate-1/2, memory-6 code. The generator polynomials for this code are $g^{(1)}(D) = 1 + D + D^3 + D^4 + D^6$ and $g^{(2)}(D) = 1 + D^3 + D^4 + D^5 + D^6$. For decoding, the BCJR algorithm was used because for short codelength it is superior to the Viterbi algorithm in terms of bit error rate at the end of the codeword. Details of this algorithm can be found in many sources so a description is not given here [4], [5].

3.2. LDPC Code

The codes used were rate-1/2, regular LDPC codes available from the MacKay repository [6], the most well-known repository of LDPC codes available. Codewords are transmitted in

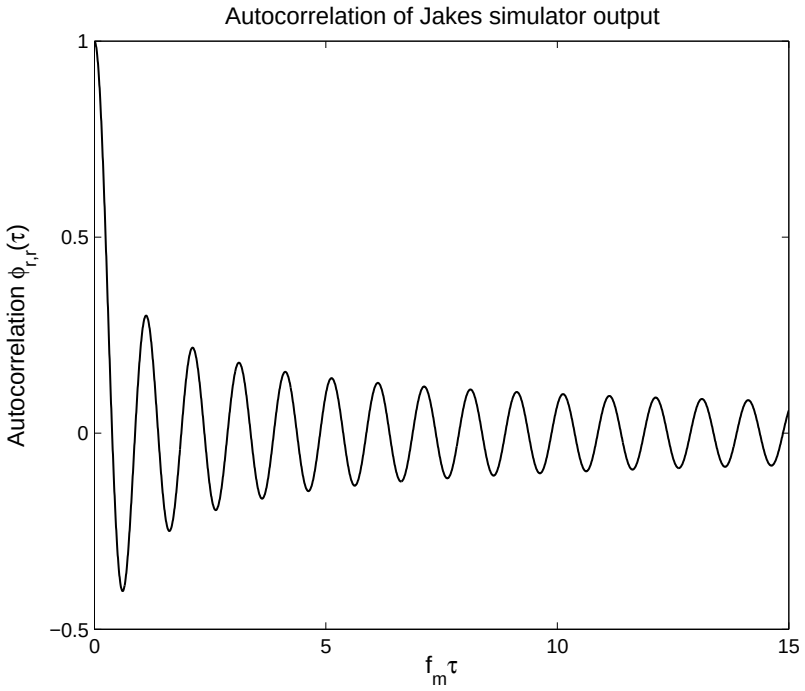

Figure 2. Autocorrelation function Jakes simulator output for $f_m T = 0.004$.

bipolar form to model BPSK modulation, where zeros are mapped to $+1$ and ones to -1. The corrupted bits were decoded using the min-sum-with-correction algorithm described below [4].

The decoder uses the code's parity check matrix $\mathbf{H}$ as a blueprint for decoding. That is, the decoder is based on a Tanner graph, which is a bipartite graph whose adjacency matrix is $\mathbf{H}$. The two sets of nodes are the check nodes (CNs) corresponding to rows of $\mathbf{H}$ and variable nodes (VNs) corresponding to columns of $\mathbf{H}$. An $(n,\ k)$ code will have n VNs and k CNs in its Tanner graph representation. A one in row i and column j of $\mathbf{H}$ corresponds to a connection between VN j and CN i.

Let VNs be indexed by j and CNs by i. Initialization of the decoder begins with the computation of log-likelihood ratios (LLRs) from the channel samples $\mathbf{y}$. L_j is the LLR computed from channel sample y_j as

$$L_j = L(v_j|y_j) = \frac{2\alpha_j y_j}{\sigma^2}, \tag{8}$$

where α_j is a Rayleigh RV and σ^2 is the noise variance of the channel, both assumed known by the decoder. Following initialization, VN-to-CN messages may be computed as

$$L_{j\to i} = L_j + \sum_{i'\neq i} L_{i'\to j}. \tag{9}$$

CN-to-VN messages are computed as

$$L_{i\to j} = \boxplus_{j'\neq j} L_{j'\to i} \tag{10}$$

where $L_1 \boxplus L_2$ is a representation of

$$L_1 \boxplus L_2 = \text{sign}(L_1)\text{sign}(L_2)[\min(|L_1|, |L_2|) + s(|L_1|, |L_2|)]. \tag{11}$$

The term $s(x, y)$ is the 'correction' term

$$s(x, y) = \log\left(1 + e^{-|x+y|}\right) - \log\left(1 + e^{-|x-y|}\right). \tag{12}$$

If the correction term in equation (11) is set to zero, the algorithm is the min-sum algorithm. Code bit decisions are made based on the signs of the total LLRs. That is, for $j = 0, 1, ..., n-1$

$$L_j^{total} = L_j + \sum_i L_{i\to j}. \tag{13}$$

The decoder decides 1 if $L_j^{total} < 0$ and 0 otherwise. Let $\mathbf{v}$ be the vector that holds code bit decisions. At this point, if $\mathbf{vH}^T = \mathbf{0}$, a codeword has been found and decoding stops; otherwise messages are computed again. Iterations proceed in this fashion until a codeword is found or a maximum number of iterations is reached [4]. For our simulations, the maximum was set to 100.

3.3. Turbo Code

The turbo codes used were parallel concatenated convolutional codes (PCCCs). The constituent codes are identical memory-4, rate-1/2 recursive systematic convolutional (RSC) codes with generator matrix $\mathbf{G}(D) = [1\ g^{(2)}(D)/g^{(1)}(D)]$ where $g^{(1)}(D) = 1 + D + D^4$ and $g^{(2)}(D) = 1 + D + D^3 + D^4$. The nominal rate of this turbo code is 1/3 as each RSC encoder will produce a parity bit for a single input bit. Let $\mathbf{u}$ denote systematic data and $\mathbf{u}'$ denote permuted systematic data. Then $\mathbf{p}$ will be the parity produced from encoding $\mathbf{u}$ and $\mathbf{q}$ will be the parity produced from encoding $\mathbf{u}'$. The codeword consists of $\mathbf{u}$, $\mathbf{p}$ and $\mathbf{q}$. In order to obtain the desired rate of 1/2, the even parity bits are punctured (deleted) prior to transmission.

The turbo decoder consists of two soft-in/soft-out (SISO) BCJR decoders and an S-random interleaver. The BCJR decoders are matched to the RSC codes used at the encoder. Also, the decoder has the same interleaver and de-interleaver used at the encoder. The received systematic data, $\mathbf{y_u}$, its corresponding parity $\mathbf{y_p}$, and extrinsic data $L_{2\to 1}$ are input to the first decoder (D1). Initially $L_{2\to 1}$ is zero as the companion decoder hasn't executed yet. The input to the second decoder (D2) is $\mathbf{y_{u'}}$, $\mathbf{y_q}$ and the extrinsic data $L_{1\to 2}$ produced by D1. $\mathbf{y_{u'}}$ is obtained by interleaving $\mathbf{y_u}$. The decoders continue to iterate until a maximum number of iterations or some other stopping criterion is reached. We set the maximum number of iterations to 15.

Code bit decisions are made based on the sign of the total LLRs where the total is computed by adding extrinsic information produced from both decoders and the likelihood information received directly from the channel [4], [5].

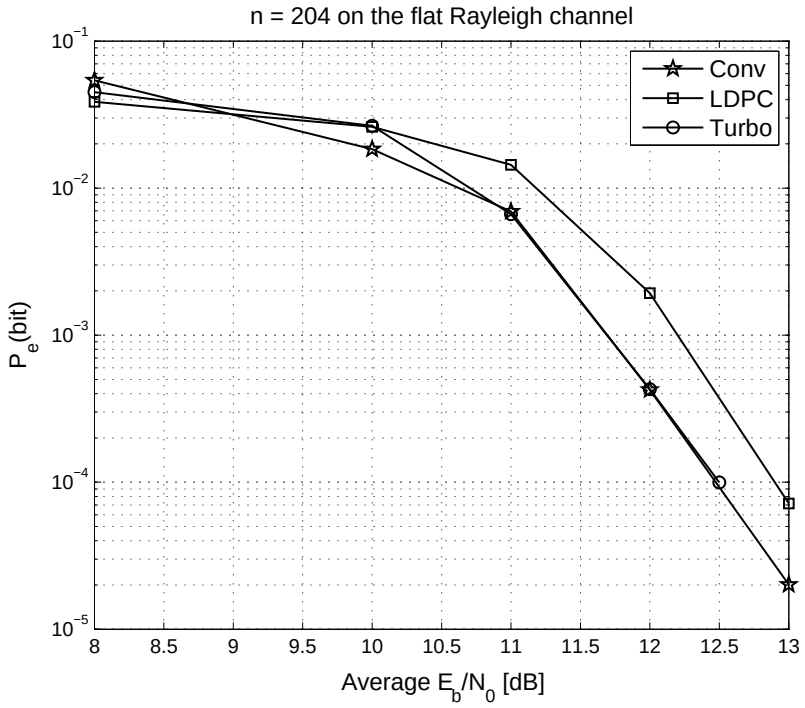

Figure 3. Rate-1/2, $n = 204$ codes over the flat Rayleigh fading channel for $f_m T = 0.004$.

4. RESULTS

For various codelengths, bit error probabilities for the three different channel codes were plotted as a function of $E\{\alpha^2\}E_b/N_0$, the average E_b/N_0. Table 1 lists the code parameters for the various codes where the notation is $R(n, k)$. n is the codelength, k is the information length and R is the code rate. The results for the $n = 204$ codes over the flat Rayleigh fading channel are shown in Figure 3. In this plot we see that the convolutional and turbo codes perfrom similarly and the LDPC code has the highest P_e(bit). For comparison, the results for the AWGN channel with no fading are shown in Figure 4. Over this channel the LDPC code has the highest P_e(bit) and the turbo code does the best. Similarly, Figure 5 shows the Rayleigh channel results for the $n = 504$ codes on the fading channel and Figure 6 is the corresponding AWGN channel results. In Figures 5 and 6 both show that for larger channel SNRs the LDPC and turbo codes do best with the turbo code attaining the lowest P_e(bit).

Table 1. Parameters for Convolutional, Turbo and LDPC Codes Used

0.5(204, 102)
0.5(408, 204)
0.5(504, 255)
0.5(816, 408)
0.5(1008, 504)

We are interested in comparing the performance of the three code types for varying code-

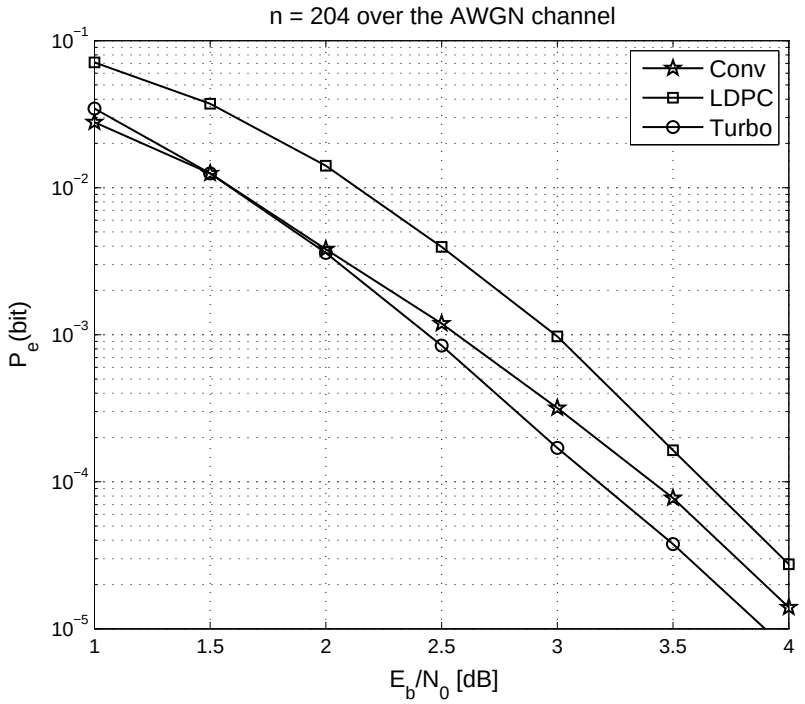

Figure 4. Rate-1/2, $n = 204$ codes over the AWGN channel.

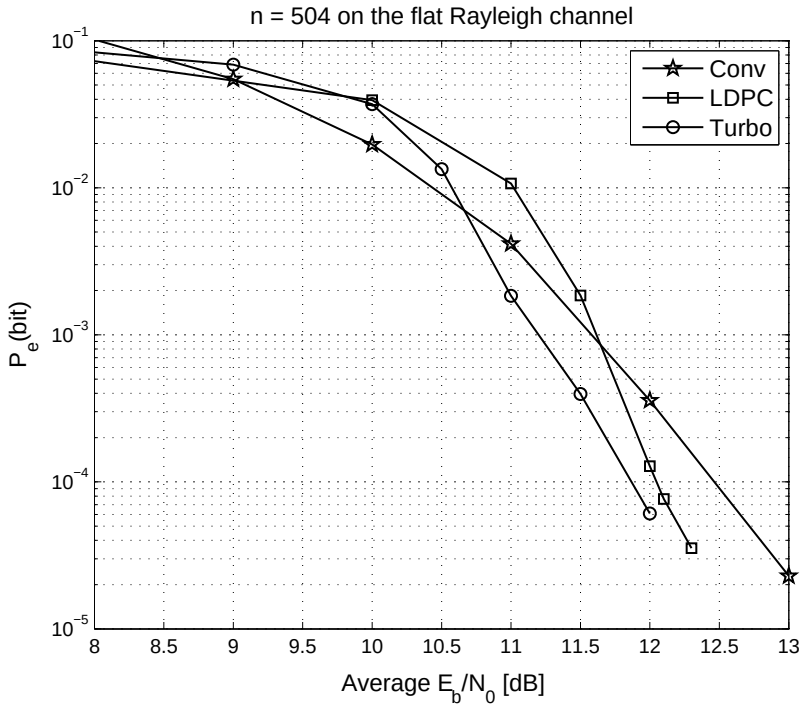

Figure 5. Rate-1/2, $n = 504$ codes over the flat Rayleigh fading channel for $f_m T = 0.004$.

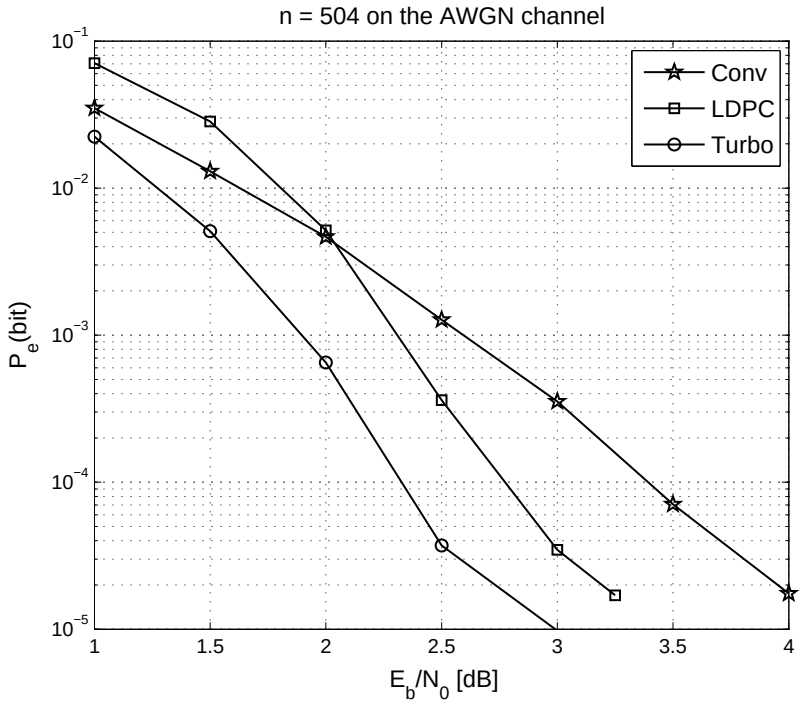

Figure 6. Rate-1/2, $n = 504$ codes over the AWGN channel.

lengths. For a fixed average channel SNR, $\mathrm{P_e(bit)}$ was plotted as a function of codelength. Figure 7 is a plot of the results for average channel SNR of 11 dB and $f_m T = 0.004$. Figure 8 is a plot of the results for average channel SNR of 12 dB and $f_m T = 0.004$. From Figure 8 it can be seen that for very short codelengths, the convolutional code and turbo code have similar performance and the LDPC code is worse. For $n = 408$, all three codes have approximately the same performance with $\mathrm{P_e(bit)} \approx 3 \times 10^{-4}$. For $n = 504$ and greater, the turbo and LDPC codes achieve a lower $\mathrm{P_e(bit)}$ than the convolutional code. The larger the codelength, the greater the discrepancy between the convolutional code and the turbo and LDPC codes. The performance of the convolutional code improves as the codelength is made larger, but this gain is small compared to the improvement achieved by the LDPC and turbo codes of the same length. From Figure 8 it can be seen that the turbo code outperforms the LDPC code. The LDPC codes considered here were not optimized (in particular they were regular) and this explains why they are inferior to turbo codes in Figure 8.

5. CONCLUSION

Results show that for all but very short codelengths, $n = 204$ and 408, the turbo and LDPC codes outperform the convolutional code on the channel considered here. It was seen that as the codelengths were increased, the gap in performance between the convolutional code and the turbo and LDPC codes increases as well. Further work could examine other channels, such as the frequency selective fading channel.

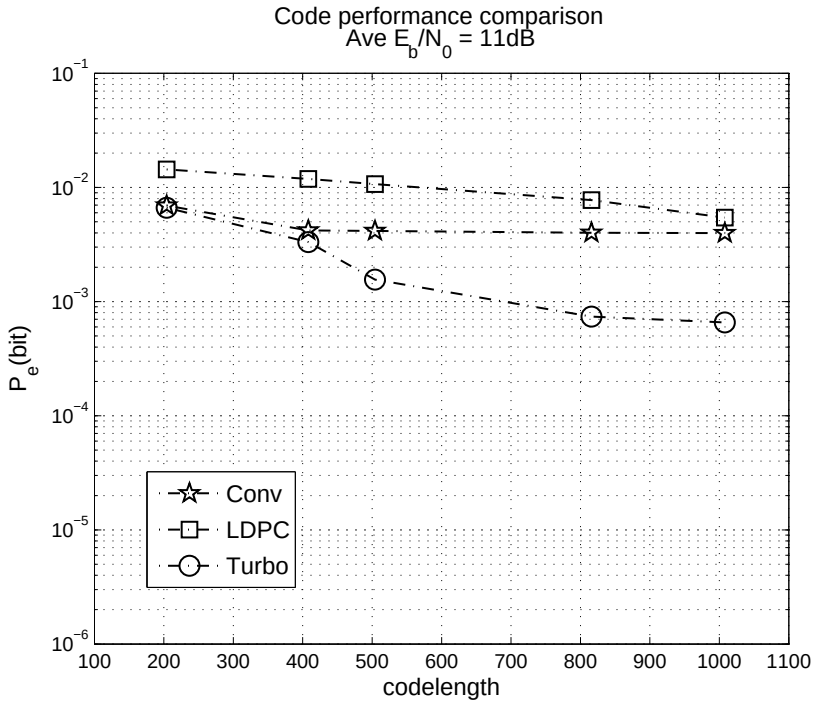

Figure 7. Rate-1/2 codes on the flat Rayleigh channel for average $E_b/N_0 = 11$ dB and $f_m T = 0.004$.

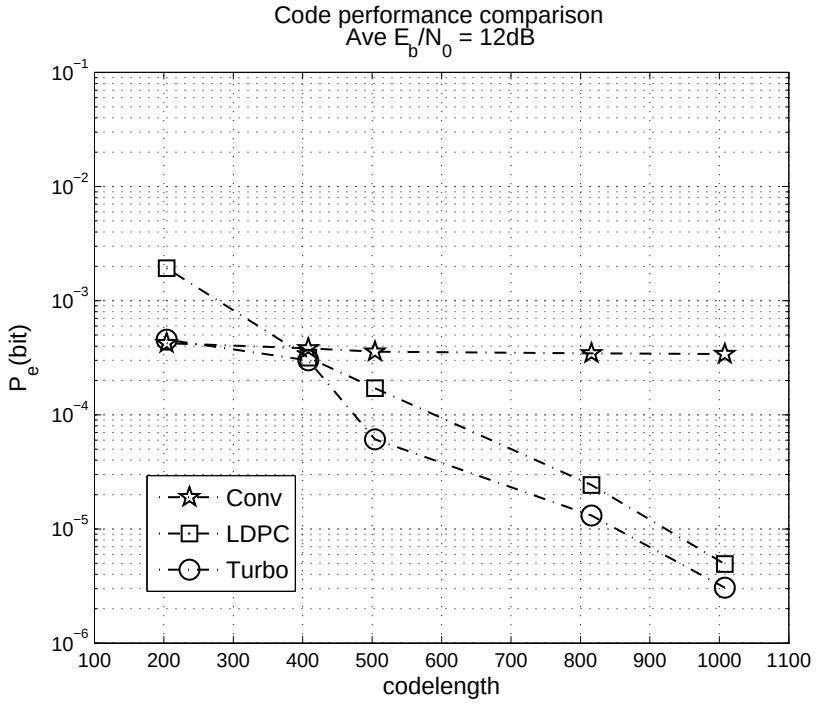

Figure 8. Rate-1/2 codes on the flat Rayleigh channel for average $E_b/N_0 = 12$ dB and $f_m T = 0.004$.

ACKNOWLEDGMENTS

This work was supported by the International Foundation for Telemetering and Rincon Research Corporation.

REFERENCES

1. D. Divsalar and F. Pollara, "Multiple turbo codes for deep-space communications," JPL TDA Progress Report, 42-121, May 15, 1995.
2. E. Biglieri, J. Proakis and S. Shamai, "Fading Channels: Information-Theoretic and Communications Aspects," *IEEE Trans. Commun.*, vol. 44, pp. 2619-2691, Oct. 1998.
3. M. Patzold and F. Laue, "Statistical properties of Jakes' fading channel simulator," *Proc. IEEE 48th Veh. Tech. Conf.*, Ottawa, Canada, pp.712-718, May 1998.
4. W. Ryan and S. Lin, *Channel Codes: Classical and Modern*, Cambridge University Press, 2009.
5. S. Lin and D. Costello, *Error Control Coding*, Pearson Prentice Hall, 2004.
6. http://www.inference.phy.cam.ac.uk/mackay/CodesFiles.html