TURBO CODES FOR COHERENT FH-SS WITH PARTIAL BAND INTERFERENCE*

Joseph H. Kang and Wayne E. Stark

University of Michigan
Dept. of Electrical Engineering and Computer Science
Ann Arbor, MI

Abstract

In this paper, turbo codes are investigated in a slow frequency-hopped spread spectrum (FH-SS) system with partial-band jamming. In addition, full-band thermal noise is present. This paper focuses on the implemention of a modified turbo decoder for this model. We consider cases of known or unknown channel state and variable number of bits per hop. Our approach is to modify the calculation of branch transition probabilities inherent in the decoder. Analytical bounds are derived and simulation is performed for coherent demodulation. The results drawn from this code are compared with a convolutional code.

1 Introduction

Turbo codes are an exciting new channel coding scheme that achieve almost reliable data communication at signal-to-noise ratios close to the Shannon limit. The results published in the inaugural paper by Berrou et al [2] were so good that they were met with much skepticism. Since then, however, these results have been reproduced and even improved. Consequently, much of the present research is applying turbo codes to different systems.

In packet data communications, the use of error correction codes plays a key role in achieving low packet error rates. When transmitting speech, excessive processing delay is unacceptable so higher error rates are tolerated. In data communication, low error rates are more important and delay at the decoder is more acceptable. Turbo codes have been shown to yield extremely low BER at low SNRs, but at the expense of computational complexity. Therefore, it would appear that turbo codes are possible candidates for packet data communications. In this paper, we will investigate the performance of turbo codes in a FH-SS system with partial band interference.

The outline of this paper is as follows. In Section 2, the system model, including details of the FH-SS model and a review of turbo codes is discussed. In Section 3, we present our modifications to the turbo decoder for FH-SS. Analytical performance bounds are derived in Section 4. Our simulation results and bound numerical results are presented in Sections 5 and 6. In Section 7, we conclude by discussing the potential of turbo codes in FH-SS systems.

2 System Model

2.1 Transmitter

The turbo encoder is formed using two constituent codes. As in the original work by Berrou *et al* [2], the constituent codes considered in this paper are recursive systematic convolutional codes. The turbo encoder is formed by concatenating the constituent codes in parallel and then separating the codes by an interleaver [5]. The encoder takes as input the data sequence $d_k \epsilon \{-1, +1\}$ and then outputs three streams: the information bits d_k, the parity bits $p_{1,k}$ of the first component encoder with input d_k, and the parity bits $p_{2,k}$ of the second component encoder with interleaved d_k as input. BPSK modulation is considered with coherent demodulation. The resulting signal is frequency hopped. The hopping patterns of the FH-SS system are modeled as sequences of independent random variables uniformly distributed over the allowable frequency range.

2.2 Channel

It is assumed that there exists a jammer that will evenly distribute its power over a fraction ρ of the frequency range. Thus, transmission occurs on a channel that includes both full-band noise with power spectral density $N_0/2$ and partial band interference with power spectral

*This research is supported by the Army Research Office under Grant DAAH04-96-1-0110

0-7803-4249-6/97/$10.00

density $N_J/2\rho$ which covers a fraction ρ of the band. As a result, there are essentially two channel states: jammed and unjammed. The probability of hopping to a jammed state is ρ and the probability of hopping to an unjammed state is $1-\rho$. Let $(y_{1,k}, y_{2,k}, y_{3,k})$ be the outputs of the channel that correspond to d_k, $p_{1,k}$, and $p_{2,k}$, respectively, and let $z_{i,k}$ represent the channel state for $y_{i,k}$. If $z_{i,k}=1$ refers to a jammed state and $z_{i,k}=0$ refers to an unjammed state, then for the case of coherent reception,

$$y_{1,k} = \sqrt{E}\, d_k + \eta_{1,k} \tag{1}$$
$$y_{2,k} = \sqrt{E}\, p_{1,k} + \eta_{2,k} \tag{2}$$
$$y_{3,k} = \sqrt{E}\, p_{2,k} + \eta_{3,k} \tag{3}$$

where $\eta_{i,k} \sim N(0, \frac{N_0}{2} + z_{i,k} \cdot \frac{N_J}{2\rho})$.

2.3 Original Turbo Decoder

Turbo decoding is an iterative procedure which makes use of the MAP algorithm. The derivation of this algorithm will not be repeated, as it has been well documented in previous papers [1] [2]. Let S_k be the state of the encoder at time k, and L_k be the log likelihood ratio (LLR) of the *a posteriori* probabilities. Then for $j = 2, 3$

$$L_k^{(j)} = \log \frac{Pr(d_k = +1|\underline{y}_1, \underline{y}_j)}{Pr(d_k = -1|\underline{y}_1, \underline{y}_j)} \tag{4}$$
$$= \log \frac{\sum_m \sum_{m'} \gamma_{+1}(y_k, m', m)\alpha_{k-1}(m')\beta_k(m)}{\sum_m \sum_{m'} \gamma_{-1}(y_k, m', m)\alpha_{k-1}(m')\beta_k(m)} \tag{5}$$

where the forward and backward recursions can be expressed as

$$\alpha_k(m) = \frac{\sum_{m'} \sum_i \gamma_i(y_k, m', m)\alpha_{k-1}(m')}{\sum_m \sum_{m'} \sum_i \gamma_i(y_k, m', m)\alpha_{k-1}(m')} \tag{6}$$

$$\beta_k(m) = \frac{\sum_{m'} \sum_i \gamma_i(y_{k+1}, m', m)\beta_{k+1}(m')}{\sum_m \sum_{m'} \sum_i \gamma_i(y_{k+1}, m', m)\alpha_k(m')} \tag{7}$$

The branch transition probabilities are

$$\begin{aligned}\gamma_i(y_k, m', m) = &\\ & p(y_{1,k}|d_k = i, S_k = m, S_{k-1} = m') \cdot \\ & p(y_{j,k}|d_k = i, S_k = m, S_{k-1} = m') \cdot \\ & q(d_k = i|S_k = m, S_{k-1} = m') \cdot \\ & Pr(S_k = m|S_{k-1} = m')\end{aligned} \tag{8}$$

where $q(d_k = i|S_k = m, S_{k-1} = m') = 1$ if bit d_k is associated with the given state transition and equals 0 if it is not. $Pr(S_k = m|S_{k-1} = m')$ depends on the *a priori* probabilities of the information bits d_k and are computed as

$$Pr(S_k = m|S_{k-1} = m') = \frac{e^{d_k \cdot L_k}}{1 + e^{L_k}}. \tag{9}$$

The combined MAP LLR for bit d_k is

$$\begin{aligned}L_k = & \log \frac{p(y_{1,k}|d_k = +1)}{p(y_{1,k}|d_k = -1)} + L_k^{(2)} + \\ & \log \frac{\sum_m \sum_{m'} \gamma'_1(y_{3,k}, m', m)\alpha_{k-1}(m')\beta_k(m)}{\sum_m \sum_{m'} \gamma'_0(y_{3,k}, m', m)\alpha_{k-1}(m')\beta_k(m)}\end{aligned} \tag{10}$$

where $\gamma'_i(y_{3,k}, m', m) = \frac{\gamma_i(y_{3,k}, m', m)}{Pr(S_k = m|S_{k-1} = m')}$.

3 Turbo Decoder for FH-SS Systems

The turbo decoding algorithm is dependent on what information is available to the turbo decoder. We examine the cases where knowledge of the channel state (i.e. jammed or unjammed) is either available or unavailable to the decoder. The case of known channel state will be referred as side information. In addition, the cases of independent and identically distributed (IID) transmission (i.e. one bit per hop) and transmission over a channel with memory (i.e. h bits per hop) are considered.

3.1 FH-SS Without Memory

In this section, we consider the case of one bit per hop. If the channel state is unknown, then the modified turbo decoder for IID FH-SS needs only to adapt the calculation of branch transition probabilities. More specifically, (8) is calculated using

$$\begin{aligned}& p(y_{i,k}|d_k = i, S_k = m, S_{k-1} = m') = \\ & \quad p(y_{i,k}|d_k = i, S_k = m, S_{k-1} = m', z_{i,k} = 1) \cdot \rho + \\ & \quad p(y_{i,k}|d_k = i, S_k = m, S_{k-1} = m', z_{i,k} = 0) \cdot (1 - \rho).\end{aligned} \tag{11}$$

If the channel state is known, then branch transition probabilities are calculated as

$$\begin{aligned}& p(y_{i,k}, z_{i,k} = z|d_k = i, S_k = m, S_{k-1} = m') = \\ & \quad p(y_{i,k}|z_{i,k} = z, d_k = i, S_k = m, S_{k-1} = m')p(z_{i,k} = z)\end{aligned} \tag{12}$$

3.2 FH-SS with Memory

In this section, we consider the case of multiple bits per hop. For our model, the number of coded bits in each dwell period will be a multiple of three. Thus, $y_{1,k}$, $y_{2,k}$, and $y_{3,k}$ are received across the same frequency. If the channel state is unknown, the approach is to calculate *a posteriori* probabilities for each channel state, $p(z_k|\underline{y}_1, \underline{y}_2)$, and send this information in addition to

$p(d_k|\underline{y}_1, \underline{y}_2)$ between decoders. Thus, information bit estimates *and* channel state estimates can be iteratively improved.

When there is no SI, the *a priori* state probabilities are replaced by the *a posteriori* state probabilities for branch transition probability calculations. As in the IID case, cases with SI use the appropriate *a priori* probability. Thus, for the *MAP1* decoder with $i = 1, 2$

$$\begin{aligned} p(y_{i,k}|d_k, S_k, S_{k-1}) \approx \\ p(y_{i,k}|d_k, S_k, S_{k-1}, z_{i,k} = 1)\, p(z_{i,k} = 1|\underline{y}_1, \underline{y}_2)) + \\ p(y_{i,k}|d_k, S_k, S_{k-1}, z_{i,k} = 0)\, p(z_{i,k} = 0|\underline{y}_1, \underline{y}_2)) \end{aligned} \quad (13)$$

when there is no side information and

$$\begin{aligned} p(y_{i,k}, z_{i,k} = z|d_k, S_k, S_{k-1}) = \\ p(y_{i,k}|z_{i,k} = z, d_k, S_k, S_{k-1})\, p(z_{i,k} = z) \end{aligned} \quad (14)$$

when there is side information.

Calculation of *a posteriori* state estimates for the *MAP1* decoder are shown below assuming there are 3 bits per hop. This result can easily be extended to h bits per hop where h is a multiple of 3.

$$\begin{aligned} & p(z_k = z|\underline{y}_1, \underline{y}_2) \\ = \; & p(y_{1,k}, y_{2,k}|z_k = z) \cdot p(z_k = z) \cdot K & (15) \\ \approx \; & p(y_{1,k}, y_{2,k}|z_k = z) \cdot p(z_k = z|\underline{y}_1, \underline{y}_3) \cdot K & (16) \end{aligned}$$

where K is a normalizing factor chosen to make the probability density function sum to 1.

In the above equation $p(z_k = m|\underline{y}_1, \underline{y}_3)$ is used in place of $p(z_k = m)$ to take advantage of the state estimate provided by the *MAP2* decoder. Equation 16 can be calculated by performing total probability on $p(y_{1,k}, y_{2,k}|z_k = m)$ over the respective coded bits.

4 Bounds

It is often impractical to generate simulation results for extremely low BERs. As a result, bounds are often calculated.

Turbo codes are linear, so the union bound can be used to form an analytical expression for the probability of error. Note that the union bound applies to the optimal decoder, while the MAP iterations of the turbo decoder are suboptimal. If A_d is the weight enumerator of the code and $P_2(d)$ is the pairwise error probability between the all-zeros codeword and a codeword of weight d, the union bound for an (n, k) block code is

$$P_{word} \leq \sum_{d=d_{min}}^{n} A_d\, P_2(d). \quad (17)$$

The only way to calculate A_d is via an exhaustive search involving all possible input sequences. One solution is to calculate an average upper bound by computing an average weight function over all possible interleaving schemes [4]. We define an average weight function as

$$\overline{A}_d = \sum_{i=1}^{k} \binom{k}{i} p(d|i) \quad (18)$$

where $p(d|i)$ is the probability that an interleaving scheme maps an input weight of i to produce a codeword of total weight d and $\binom{k}{i}$ is the number of input frames with weight i. Thus,

$$\overline{P}_{word} \leq \sum_{d=d_{min}}^{n} \sum_{i=1}^{k} \binom{k}{i} p(d|i)\, P_2(d) \quad (19)$$

$$\overline{P}_{bit} \leq \sum_{d=d_{min}}^{n} \sum_{i=1}^{k} \frac{i}{k} \binom{k}{i} p(d|i)\, P_2(d). \quad (20)$$

An algorithm for calculating $p(d|i)$ is given in [4]. Thus, in order to compute this bound, we need only calculate $P_2(d)$. Over a channel with both full-band thermal noise and partial-band jamming noise, we can calculate the pairwise error probability by conditioning on the number of jammed symbols.

$$P_2(d) = \sum_{l=0}^{d} P(\text{error} \mid E_l) \cdot P(E_l) \quad (21)$$

where E_l is the event that l symbols are jammed and

$$P(E_l) = \binom{d}{l} \rho^l (1 - \rho)^{d-l}. \quad (22)$$

The fundamental approach to calculating pairwise error probabilities is to compute log likelihood ratios. For the case where the decoder has no side information, this is difficult since the receiver does not know which bits have been jammed. Thus, for analytical purposes, we will consider the suboptimal decoder that makes bit decisions based on the sum of the channel outputs. For this suboptimal decoder,

$$P(\text{error} \mid E_l) = Q\left(\sqrt{\frac{E_b R d}{\frac{N_0}{2} + \frac{l}{d}\frac{N_J}{2\rho}}}\right). \quad (23)$$

If the decoder has side information, we calculate LLRs in the normal way and then calculate the probability of error.

$$P(\text{error} \mid E_l) = Q\left(\sqrt{E_b R \left(\frac{l}{\frac{N_0}{2} + \frac{N_J}{2\rho}} + \frac{d - l}{\frac{N_0}{2}}\right)}\right) \quad (24)$$

5 Simulation Results

For all simulations, the component encoders are rate $\frac{1}{2}$ recursive systematic convolutional encoders with memory 4 and octal generators $(37, 21)$. The packet size is 1640 bits and the number of turbo code iterations is 10. The SNR of the full-band thermal noise is set to 20 dB. Cases with memory are simulated using 3 bits per hop.

Figure 1 shows the plot of E_b/N_J needed to achieve a packet error rate (PER) of 10^{-2} for a given ρ. As would be expected, cases with side information (SI) performed better than their counterparts with no SI (NSI). The SI and NSI curves only meet when $\rho = 1.0$. In this case, all states are jammed, so side information provides no additional information. Also, the memory/NSI case yielded lower error rates than the IID/NSI case. In the memory/NSI case, we were able to calculate reliable state information. These state estimates provided useful information which in turn aided the decoding process.

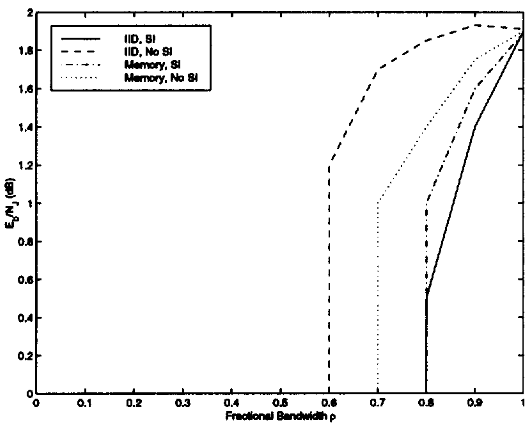

Figure 1: Simulation Results: Turbo Codes

For any memory/NSI case, where channel states are iteratively estimated, performance is upper bounded by the memory/SI case. Because channel state estimates will improve if the number of bits per hop increases, we should be able to get arbitrarily close to the corresponding memory/SI result by increasing the memory. However, as shown in Figure 1, moving from 1 bit per hop (IID case) to 3 bits per hop (memory case) resulted in degraded SI performance. Thus, the upper bound for NSI performance appears to effectively decrease. If this trend were to continue for increasing memory lengths, then there could exist a point where the NSI case does worse. Note, however, that direct comparison of the IID/SI and memory/SI cases is unfair. Because the memory/SI case uses a third the total number of hops that the IID/SI case uses, the effective block length of the memory/SI case is reduced by a factor of three. For future research, it would be interesting to further investigate these two cases as block lengths are increased.

In order to gauge our results, we refer to the application of convolutional codes to a FH-SS system with one bit per hop. Using a rate 1/3, memory 4 convolutional code with maximal free distance, this result for PER $= 10^{-2}$ is shown in Figure 2. Performance of FH-SS systems is often measured by the worst case E_b/N_J across all ρ. Using this criterion, turbo codes show a gain of $3.1 - 3.5$ dB over convolutional codes, depending on whether SI is available to the decoder.

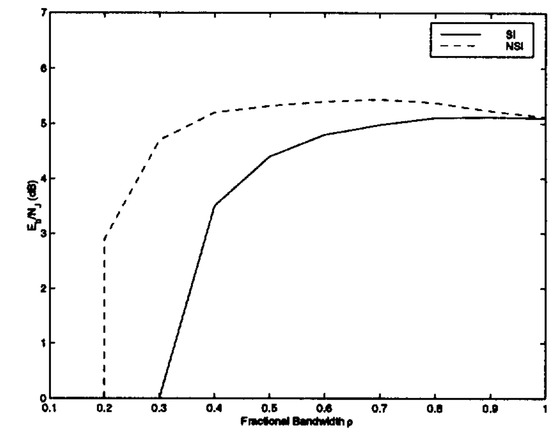

Figure 2: Simulation Results: Conv. Codes

6 Numerical Results: Bounds

The numerical results of the bounds are shown in Figures 3 and 4. The bounds are known to diverge for SNRs below 2 or 3 dB [4], a range close to the area of interest. Thus for coherent reception, we cannot concretely determine the precision of the bounds. Note, however, that for the case with SI, the general shape of the bounds conform to what is expected. At low SNRs, the jamming noise power is sufficiently high, so the code performs most poorly when $\rho = 1.0$. At high SNRs, the jamming power is so low that spreading it across more frequencies effectively makes the noise negligible. Thus, coding performance is best at $\rho = 1.0$ for high SNRs.

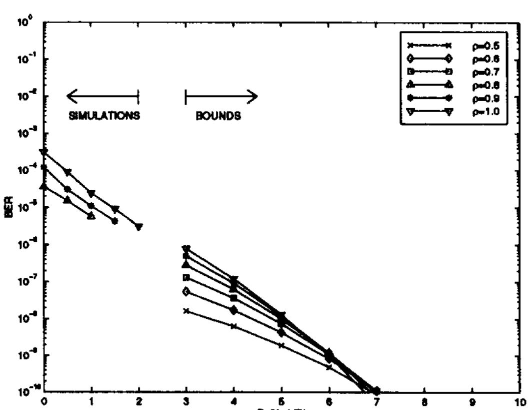

Figure 3: Bounds: Turbo Codes, With SI

For the NSI case, we get a similar result at high SNRs. At low SNRs, however, the $\rho = 1.0$ case performs the best. There are two reasons these analytical results may not be accurate. First, recall that for analysis of the NSI case, we used a suboptimal decoder to calculate pairwise error probabilities. It is possible that the bound to the optimal decoder might have a different shape. A second explanation is that bounds are generally loose at low SNRs. Thus, the results of the bounds at low SNRs are unreliable.

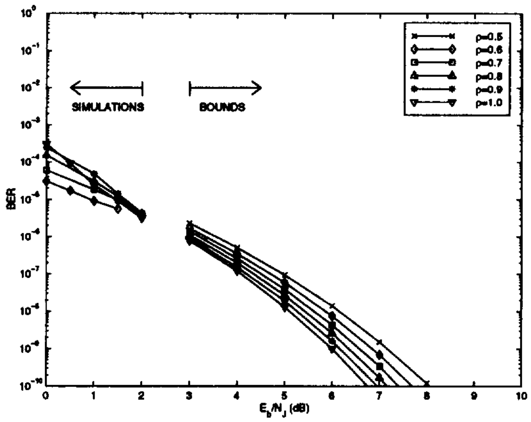

Figure 4: Bounds: Turbo Codes, No SI

Figures 5 and 6 show BER bounds for rate 1/3, memory 4 convolutional codes with maximal free distance. These were calculated by using standard techniques. The results show similar form to those of the turbo code, but yield a higher BER for a given E_b/N_J.

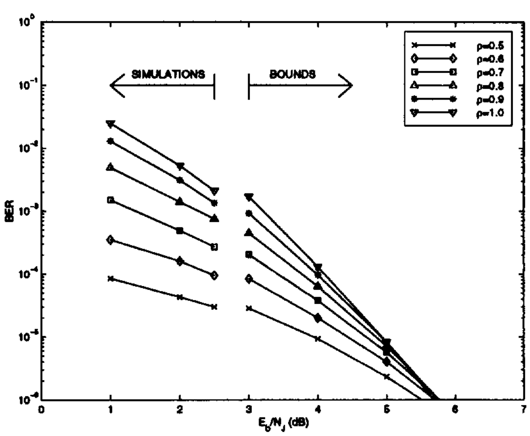

Figure 5: Bounds: Conv. Codes, With SI

7 Conclusion

We have shown that there exists great potential for turbo codes in frequency-hop spread spectrum systems by analyzing cases with 1 or 3 bits per hop, and either with or without channel state side information. The case with

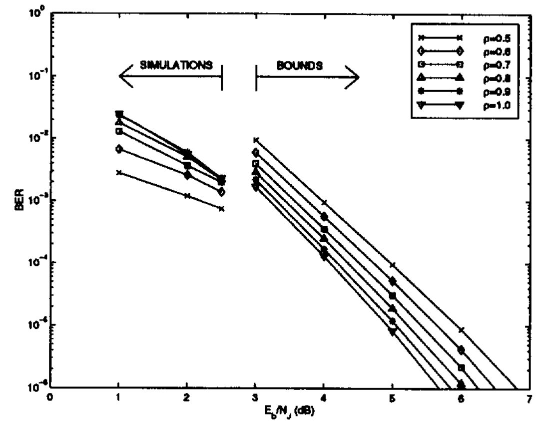

Figure 6: Bounds: Conv. Codes, No SI

1 bit/hop and side information was uniformly superior relative to the other cases, while the 1 bit/hop case without side information was uniformly inferior. We have shown that in cases with memory, channel state estimation is an effective tool for decoding. In addition, we have demonstrated the superior performance of turbo codes in FH-SS over convolutional codes.

While turbo codes are effective in reducing the E_b/N_J required to achieve a given packet error probability, they so at great computational cost. Before turbo codes can be integrated into a packet radio network or any practical data communications system, their computational complexity must be reduced while minimizing performance losses.

References

[1] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Information Theory*, vol. 20, March 1974.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," *Proc ICC '93* May 1993.

[3] M. Pursley and W. Stark, "Performance of Reed-Solomon coded frequency- hop spread-spectrum communications in partial-band interference," *IEEE Trans. Communications*, vol. COM-33, August 1985.

[4] D. Divsalar, S. Dolinar, and F. Pollara, "Transfer function bounds on the performance of turbo codes," *Telecom. and Data Aquisition Progress Report 42-122*, Jet Propulsion Laboratory, August 1995.

[5] A. Barbulescu and S. Pietrobon, "Terminating the trellis of turbo codes in the same state," *Electronics Letters*, vol. 31, Nov. 1995.