Variable Latency Turbo Codes for Wireless Multimedia Applications

Matthew C. Valenti and Brian D. Woerner

Bradley Dept. of Elect. Eng., Virginia Tech
Mobile and Portable Radio Research Group
Blacksburg VA 24061-0350
Email: mvalenti@vt.edu

Abstract — **Multimedia communication systems require different Qualities of Service (QoS) for different types of information. The QoS is generally expressed in terms of a maximum acceptable latency and Bit Error Rate (BER) or Frame Error Rate (FER). In turbo coded systems a tradeoff between latency and BER results from the choice of interleaver size. This tradeoff can be exploited in multimedia communication systems by using different interleaver sizes to achieve different QoS's. In this paper the tradeoff between latency and BER is further explored through simulation. The results are used to propose an adaptive turbo coding strategy for wireless multimedia communications that incorporates a set of interleavers of variable sizes that achieve an appropriate set of QoS's.**

1. Introduction

Research related to first and second generation mobile communication systems has focused primarily on increasing the capacity available for voice services. However, with the recent popularization of internet services such as email and Web browsing, and with the increased popularity of portable computers, it is apparent that next generation wireless systems must be capable of supporting many different types of data. In [1] many issues related to next generation wireless services are presented including the need to adapt to the dynamic variation of traffic, the need to control the quality of service (QoS) according to data type, compensation for frequency-selective fading, and the desire for high overall system capacity.

The QoS that a particular application requires is typically expressed in terms of a maximum latency and a maximum bit error rate (BER) or Frame Error Rate (FER). Applications that require low latency, such as real time voice and video communications, can generally accept higher bit error rates, while applications that require low bit error rates, such as computer data, can typically accept longer latency. There is a tradeoff in turbo codes between latency and bit error rate that is inherent in the choice of interleaver size. This tradeoff can be exploited in multimedia communications systems by using interleavers of varying sizes to offer several different qualities of service.

The organization of this paper is as follows: In section 2 the tradeoffs inherent to turbo coded systems are discussed. Section 3 presents a proposed variable frame length turbo coded system which is analyzed through simulation. Section 4 presents the results of the simulation and a discussion of the results. Finally a conclusion and discussion of future work is presented in section 5.

2. Tradeoffs in Turbo Coded Systems

Turbo codes have been shown to exhibit remarkable performance when the interleaver size is large (64 kilobits) and a sufficient number of decoding iterations is performed [2]. In [3], Benedetto and Montorsi show that the performance of turbo codes improves as the frame size is increased. However, the decoding complexity per information bit is invariant to frame size. It is this decoupling of decoding complexity with frame size that makes turbo coding such a powerful technique, particularly for large frame sizes. The tradeoff between frame/interleaver size and performance is shown for several interleaver sizes in [4], and [5]. Even when the frame size is as small as 192 bits there appears to be some benefit (in terms of performance and complexity) to using turbo codes [6]. In [5] it is shown that there is a threshold at around 200 bits; if the frame size is less than the threshold then a convolutional code will outperform the turbo code of comparable complexity, otherwise if the frame size is greater than the threshold the turbo code will perform better in terms of BER for a particular E_b/N_o. The commmunications delay or "latency" in a turbo coded system is directly proportional to the frame size. Thus there is a direct tradeoff between performance and latency in turbo coded systems.

In addition to the tradeoff between latency and performance, there are several other tradeoffs inherent to turbo coded systems. By properly choosing the code rate, a tradeoff between spectral efficiency and performance can be made. The effect of varying the code rate from 1/3 to 4/5 is shown in [7]. A tradeoff between decoding complexity and performance results from varying the number of decoding iterations or from using various constraint length elementary encoders. In [2] the effect of varying the number of decoding iterations from one to 18 is shown, and the effect of using encoders with constraint lengths three, four, and five is shown in [6]. In [7] a stop criterion is introduced which makes the number of decoding iterations dynamic. An additional tradeoff between performance and complexity is imbedded in the choice of decoding algorithm. Turbo decoders fall into two general categories: Maximum A Posteriori (MAP) algorithm

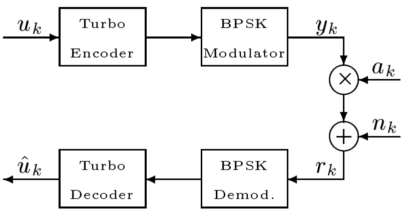

Fig. 1: System model

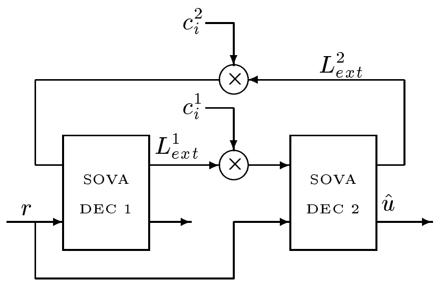

Fig. 2: Decoder diagram

based decoders and Soft Output Viterbi Algorithm (SOVA) based decoders [7]. MAP decoders generally perform better than SOVA decoders but are more computationally complex. There are several sub-optimal MAP decoders that offer reduced complexity at the price of degraded performance [6], [8]. Also there are techniques to improve the performance of SOVA decoders at the expense of a slight increase in computational complexity [9]. SOVA decoding with the normalization technique of [9] appears to offer the most practical combination of acceptable performance and computational feasability.

3. System Model

The system model is as shown in Figure 1. The system uses four frame sizes: 256, 1024, 4096, and 16,384 bits. The information bits u_k are grouped into frames, encoded and then modulated using a Binary Phase Shift Keying (BPSK) modulator. The output of the BPSK modulator y_k is multiplied by fading amplitudes a_k and added with noise n_k to produce the received value r_k. The received signal is BPSK demodulated and then turbo decoded. The output of the turbo decoder is an estimate $\hat{u}_k$ of the information bit. For the remainder of this discussion a data rate of 128,000 bits per second is assumed. In order to minimize the complexity of the decoder, a simple turbo code is utilized and decoding is performed by the SOVA algorithm with normalization performed as described in [9].

3.1 Encoder

The encoder is made up of two rate one half Recursive Systematic Convolutional (RSC) encoders, each with constraint length $K = 3$ and octal generators 5 (feedforward) and 7 (feedback) [6]. The two encoders are concatenated in parallel and separated by a random interleaver. Four interleavers were used, one for each of the four frame sizes. Each interleaver was designed by generating a random reordering vector, and no attempt was made to optimize the interleavers (although "bad" interleavers were discarded). The rate of the turbo code can be increased by puncturing the parity bits generated by the two component codes. A rate one-half turbo code was generated by using a multiplexer which selects odd indexed parity bits from the first encoder and even indexed parity bits from the second encoder. In addition, a rate one-third turbo encoder with no puncturing is considered.

3.2 Channel

The encoded bits are transmitted over the communications channel as depicted in Figure 1. The value a_k is a fading amplitude that may vary from code bit to code bit. When an Additive White Gaussian Noise (AWGN) channel model is used, a_k is unity for all code bits. The Rayleigh fading channel model assumes additional block interleaving sufficient to make the fading amplitudes independent from symbol to symbol. The fading amplitudes are normalized to have unit energy.

3.3 Decoder

Decoding is performed using eight iterations of a SOVA based decoder [7]. The performance of the SOVA decoder is enhanced using the normalization technique presented in [9]. In [9] it is noted that the SOVA algorithm suffers from two distortions — overoptimistic soft outputs and correlation between the intrinsic and extrinsic information. Performance is degraded substantially due to the first type of distortion but only mildly from the second type. For this reason, we only compenstate for the overoptimistic soft output of the SOVA algorithm. Compenstation is achieved by calculating a scale factor to be muliplied by the output of each decoding stage according to:

$$c = m_v \frac{2}{\sigma_v^2} \tag{1}$$

where m_v and σ_v^2 are the mean and variance of the SOVA output given that the original information bit is a one. Since we do not have a priori knowledge of the information bit, a satisfactory estimate of the mean and variance may be obtained by taking the absolute value of the SOVA output. The compenstation coefficients must be calculated for each decoder stage and recalculated for each new frame. The slight increase in computational complexity is justified by a significant performance improvement. A simplified diagram of the decoder that depicts the compensation operation but neglects the details of interleaving and deinterleaving is shown in Figure 2. In this figure c_i^p refers to the the scaling constant associated the i^{th} decoding iteration and the p^{th} SOVA decoder within that iteration.

For purposes of computing latency, a pipeline architecture is assumed for the decoder. Each element of the pipeline performs one iteration of decoding,

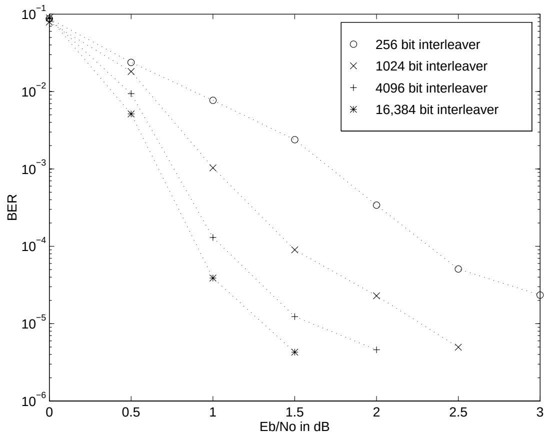

Fig. 3: Bit Error Rate vs. E_b/N_o as parameterized by frame size for turbo code with rate 1/3, constraint length 3, and 8 iterations of normalized SOVA decoding over an AWGN channel.

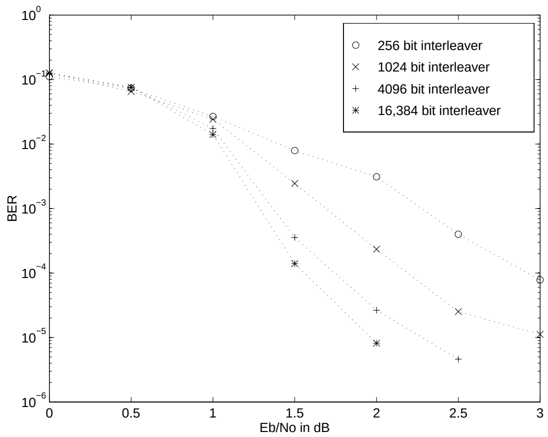

Fig. 4: Bit Error Rate vs. E_b/N_o as parameterized by frame size for turbo code with rate 1/2, constraint length 3, and 8 iterations of normalized SOVA decoding over an AWGN channel.

and thus the total latency is equal to the transmit time of the frame times the number of iterations:

$$t_d \quad = \quad \frac{k_f}{R_b} N_i \tag{2}$$

Where k_f is the frame size, R_b is the bit rate, and N_i is the number of decoding stages.

4. Simulation Results

The proposed turbo coded system was simulated using an Additive White Gaussian Noise (AWGN) channel and fully interleaved flat Rayleigh fading channel for each of the four frame sizes and two code rates. The results of the simulation for the AWGN channel are shown in Figure for the rate one-third code in Figure 3 and for the rate one-half code in Figure 4. The results of the simulated fully interleaved flat Rayleigh fading channel are shown for the rate one-third code in Figure 5 and for the rate one-half code in Figure 6. In each of these four plots there is a minimum E_b/N_o for which an acceptable range of BER's are obtained for all four interleavers. For example in Figure 3 an acceptable range of BER's can be obtained for the rate

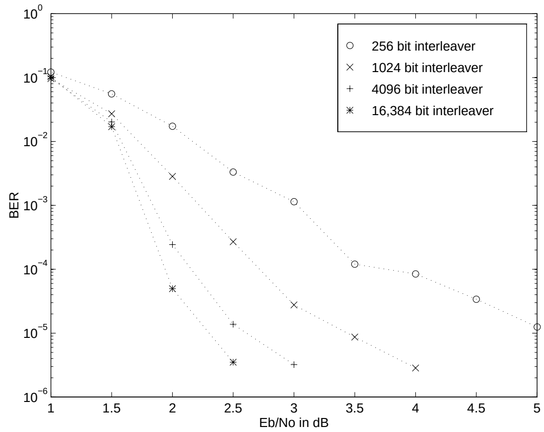

Fig. 5: Bit Error Rate vs. E_b/N_o as parameterized by frame size for turbo code with rate 1/3, constraint length 3, and 8 iterations of normalized SOVA decoding over a fully interleaved flat Rayleigh fading channel.

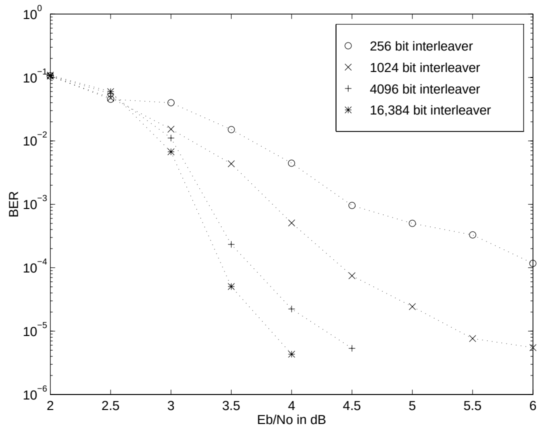

Fig. 6: Bit Error Rate vs. E_b/N_o as parameterized by frame size for turbo code with rate 1/2, constraint length 3, and 8 iterations of normalized SOVA decoding over a fully interleaved flat Rayleigh fading channel.

one-third code in AWGN by holding E_b/N_o to a constant value of 1.5 dB. The BER, FER, and latency is listed in Table 1 for the rate one-third code in AWGN with E_b/N_o set to 1.5 dB. Likewise, an acceptable range of QoS was found for the rate one-half code in AWGN at $E_b/N_o = 2$ dB (Table 2), for the rate one-third code in fully interleaved Rayleigh flat fading at $E_b/N_o = 2.5$ dB (Table 3), and the rate one-half code in fully interleaved Rayleigh flat fading at $E_b/N_o = 4$ dB (Table 4).

For each case, the 256 bit interleaver provides a latency of just 16 msec and a BER on the order of 10^{-3}. The BER and latency associated with the 256 bit interleaver is appropriate for real time voice and low-latency error-resilient image and video compression techniques such the Pyramid lattice Vector Quantization (PVQ) method of [10]. The 1,024 bit interleaver has a latency of 64 msec and a BER on the order of 10^{-4} and is appropriate for real-time video conferencing such as H.261 [11] and for JPEG images [10]. The 4,096 bit interleaver has a moderately long latency of 256 msec and a low BER on the order of 10^{-5} and is appropriate for the playback of compressed video using standards such as MPEG.

The 16,384 bit interleaver offers a very long latency of approximately one second along with extremely low BER's on the order of 10^{-6} and is appropriate for data and file transmission services. It is interesting to note that while the BER becomes smaller as the frame size increases, the Frame Error Rate (FER) remains fairly contant. Thus for systems that require different FER's there is no significant benefit to using variable frame sizes.

5. Conclusion

In turbo coded systems there is a tradeoff between performance and latency inherent in the choice of frame/interleaver size. Due to this tradeoff it is possible to achieve a variety of performance/latency points while keeping the transmitted power constant. This characteristic can be exploited in multimedia systems to allow several different QoS's at any particular value of Eb/No. A simulation study shows how a simple turbo code with 4 interleaver sizes can be used to achieve 4 different QoS's. Results were shown for both an unpunctured rate 1/3 code and a punctured rate 1/2 code in AWGN and flat fading channels.

The primary focus of this paper is the exploitation of the tradeoff between frame size and performance inherent in turbo coded systems. Similar benefits may also be obtained by matching different code rates and numbers of decoder iterations to the particular type of data that is transmitted. The conclusions of this paper are based on the simulation study of a short constraint length code with randomly designed interleavers. Further study is needed to analyze the performance of a similar system with optimized interleavers and larger constraint lengths. Furthermore, the latency that was shown in the tables assumed a pipelined decoding architecture. The decoding delays associated with other hardware implementations of the turbo decoder should be studied further.

Acknowledgements
This work was supported by the Bradley Fellowship and the MPRG industrial affiliates foundation.

References

[1] N. Morinaga, M. Nakagawa, and R. Kohono, "New concepts and technologies for achieving highly reliable and high-capacity multimedia wireless communications systems," *IEEE Commun.*, vol. 35, no. 1, pp. 34-40, Jan. 1997.

[2] C. Berrou and A. Glavieux, "Near optimum error-correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261-1271, Oct. 1996.

[3] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 409-428, Mar. 1996.

[4] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," in *Proc., IEEE GLOBECOM* (San Fransisco, 1994), pp. 1298-1303.

[5] H. Koorapaty, Y. E. Wang, and K. Balachandran, "Performance of turbo codes with short frame sizes," in *Proc., IEEE Veh. Tech. Conf.,* (Phoenix, 1997), pp. 329-333.

[6] P. Jung, "Comparison of turbo-code decoders applied to short frame transmission systems," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 530-537, Apr. 1996.

[7] J. Hagenauer, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429-445, Mar. 1996.

[8] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc., IEEE Int. Conf. on Commun.* (Seattle, 1995), pp. 1009-1013.

[9] L. Papke, P. Robertson, and E. Villebrun, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *Proc., IEEE Int. Conf. on Commun.* (1996), pp. 102-106.

[10] T. Meng, E. Tsern, A. Hung, S. Hemami, and B. Gordon, "Video compression for wireless communications," in *Proc., Virginia Tech Symp. on Wireless Personal Commun.* (Blacksburg Virginia, 1993), pp. 14.1-14.17.

[11] H. Stüttgen, "Network Evolution and Multimedia Communication," *IEEE Multimedia*, pp. 42-59, Fall 1995.

Tab. 1: QoS for rate 1/3 turbo code operating in AWGN at Eb/No = 1.5 dB

Frame Size (bits)	latency @ 128 kbps	BER	FER
256	16 msec	2.4×10^{-3}	7.3×10^{-2}
1024	64 msec	9.0×10^{-5}	2.1×10^{-2}
4096	256 msec	1.2×10^{-5}	1.5×10^{-2}
16384	1.024 sec	4.3×10^{-6}	2.0×10^{-2}

Tab. 2: QoS for rate 1/2 turbo code operating in AWGN at Eb/No = 1.5 dB

Frame Size (bits)	latency @ 128 kbps	BER	FER
256	16 msec	3.1×10^{-3}	1.0×10^{-1}
1024	64 msec	2.3×10^{-4}	3.8×10^{-2}
4096	256 msec	2.6×10^{-5}	2.8×10^{-2}
16384	1.024 sec	8.2×10^{-6}	4.5×10^{-2}

Tab. 3: QoS for rate 1/3 turbo code operating in flat Rayleigh fading at Eb/No = 2.5 dB

Frame Size (bits)	latency @ 128 kbps	BER	FER
256	16 msec	3.3×10^{-3}	1.0×10^{-1}
1024	64 msec	2.7×10^{-4}	4.1×10^{-2}
4096	256 msec	1.4×10^{-5}	1.9×10^{-2}
16384	1.024 msec	3.5×10^{-6}	1.8×10^{-2}

Tab. 4: QoS for rate 1/2 turbo code operating in flat Rayleigh fading at Eb/No = 4.0 dB

Frame Size (bits)	latency @ 128 kbps	BER	FER
256	16 msec	4.5×10^{-3}	1.7×10^{-1}
1024	64 msec	5.1×10^{-4}	8.3×10^{-2}
4096	256 msec	2.2×10^{-5}	2.5×10^{-2}
16384	1.024 msec	4.3×10^{-6}	3.0×10^{-2}