International Conference on Emerging Trends in Engineering and Technology (IETET-2010) October 14-16, 2010
Geeta Institute of Management and Technology, Kanipla, Kurukshetra, Haryana, India

High Bit Rate Turbo Coding for Wireless Communication

Kunal M. Pattani, Bharat H. Nagpara, Gaurav R. Joshi and A.C. Suthar
C. U. Shah College of Engineering and Technology, Wadhwan city, Gujarat, India
e-mail: kunalpattani@gmail.com, gaurav.joshi8386@gmail.com, acsuthar@ieee.org

Abstract*—As far as wireless communication is concerned, bit rate and bandwidth are most important parameters in terms of system design and cost effective solution. As the number of systems, users, and services increase, there must be such a transmission schemes which must provide bandwidth-efficient robust communication with low delay, supporting multiple users on broadband wire-less channels, with desired bit rate. In order to use bandwidth effectively, the error must be controlled, as it prevents the loss of data and minimize the retransmission of data and hence increases the throughput. In this paper, performance of a turbo code is investigated for high-bit-rate data transmission in a wideband DS-CDMA system. The effect of correlated multipath fading is studied by considering performance at typical high and low mobile speeds, when the size of external interleaver is varied. Performance of the turbo code is found to degrade significantly at low mobile speeds, when small external interleaver is used.

***Keywords:** Turbo Codes, Turbo Code Coder and Decoder BER, Multipath Effect.*

I. INTRODUCTION

In a communication system that employs forward error-correction coding, a digital information source sends a data sequence to an encoder. The encoder inserts redundant (or parity) bits, thereby outputting a longer sequence of code bits, called a codeword. Such code words can then be transmitted to a receiver, which uses a suitable decoder to extract the original data sequence. Codes that introduce a large measure of redundancy convey relatively little information per each individual code bit. This is advantageous because it reduces the likelihood that all of the original data will be wiped out during a single transmission. On the other hand, the addition of parity bits will generally increase transmission bandwidth requirements or message delay (or both). In Algebraic coding (also known as block coding) was the only type of forward error-correction coding the encoder intersperses parity bits into the data sequence using a particular algebraic algorithm. On the receiving end, the decoder applies an inverse of the algebraic algorithm to identify and correct any errors caused by channel corruption. Another forward error-correcting technique, known as convolutional coding process the incoming bits in streams rather than in blocks. The paramount feature of such codes is that the encoding of any bit is strongly influenced by the bits that preceded it (that is, the memory of past bits). A convolutional decoder takes into account such memory when trying to estimate the most likely sequence of data that produced the received sequence of code bits. Historically, the first type of convolutional decoding, known as sequential decoding, used a systematic procedure to search for a good estimate of the message sequence; however, such procedures require a great deal of memory, and typically suffer from buffer overflow and nongraceful degradation. In Viterbi coding, at each bit-interval, the Viterbi decoding algorithm compares the actual received code bits with the code bits that might have been generated for each possible memory-state transition. It chooses, based on metrics of similarity, the most likely sequence within a specific time frame. The Viterbi decoding algorithm requires less memory than sequential decoding because unlikely sequences are dismissed early, leaving a relatively small number of candidate sequences that need to be stored. Some types of algebraic coding are most effective in combating "bursty" errors (errors that arrive in bursts). Convolutional coding is generally more robust when faced with random errors or white noise; however, any decoding errors occurring in the convolutional decoder are likely to occur in bursts. In 1974, Joseph Odenwalder combined these two coding techniques to form a concatenated code. In this arrangement, the encoder linked together an algebraic code followed by a convolutional code. The decoder, a mirror image of the encoding operation, consisted of a convolutional decoder followed by an algebraic decoder. Thus, any bursty errors resulting from the convolutional decoder could be effectively corrected by the algebraic decoder. Performance was further enhanced by using an interleaver between the two encoding stages to mitigate any bursts that might be too long for the algebraic decoder to handle. This particular structure demonstrated significant improvement over previous coding systems and is currently being used in the Deep Space Network and Air Force Satellite Control Network as well as in commercial broadcasting services. In 1993, Claude Berrou and his associates developed the turbo code, the most powerful forward error-correction code yet. Using the turbo code, communication systems can approach the theoretical limit of channel capacity, as characterized by the so-called Shannon Limit, which had been considered unreachable for more than four decades.

II. Principles of Turbo Codes

It is theoretically possible to approach the Shannon limit by using a block code with large block length or a convolutional code with a large constraint length. The processing power required to decode such long codes makes this approach impractical. Turbo codes overcome this limitation by using recursive coders and iterative soft decoders. The recursive coder makes convolutional codes with short constraint length appear to be block codes with a large block length, and the iterative soft decoder progressively improves the estimate of the received message.

A. Coding

A specific type of convolutional coder is used to generate turbo codes. The convolutional coder shown in Figure 1a has a single input, x, outputs p0 and p1, and a constraint length K=3. Multiplexing the outputs generates a code of rate R=1/2.

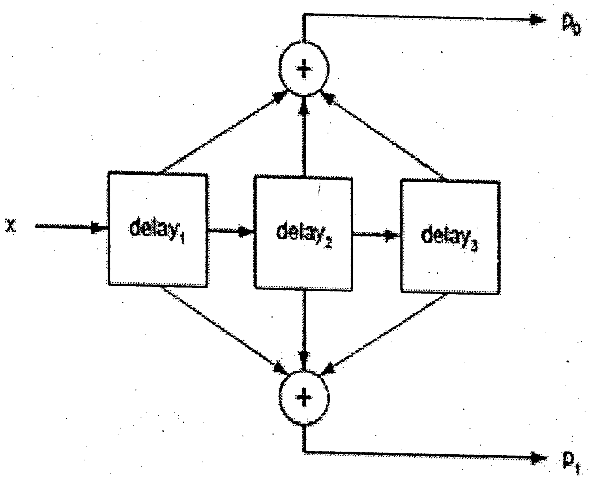

Fig. 1: a. Convolution Coders

The convolutional coder shown in Figure 1b differs in that one of the outputs, p0, has been "folded back" and is presenting one of its output sequences at the coder input, making it recursive. This has the effect of increasing the apparent block length without affecting the constraint length of the coder. The input is also presented as one outputs of the coder, making it systematic. Such coders are thus called recursive systematic convolutional (RSC) coders.In non-recursive convolutional codes it is common practice to flush the coder with zeros to bring the decoder to an end state. Flushing with zeros does not readily work with recursive coders, however relatively simple binary arithmetic can establish the input sequence that will generate a zero state RSC codes can thus be made to appear like linear block codes. A turbo code is the parallel concatenation of a number of RSC codes. Usually the number of codes is kept low, typically two, as the added performance of more codes is not justified by the added complexity and increased overhead. The input to the second decoder is an interleaved version of the systematic x, thus the outputs of coder 1 and coder 2 are time displaced codes generated from the same input sequence.

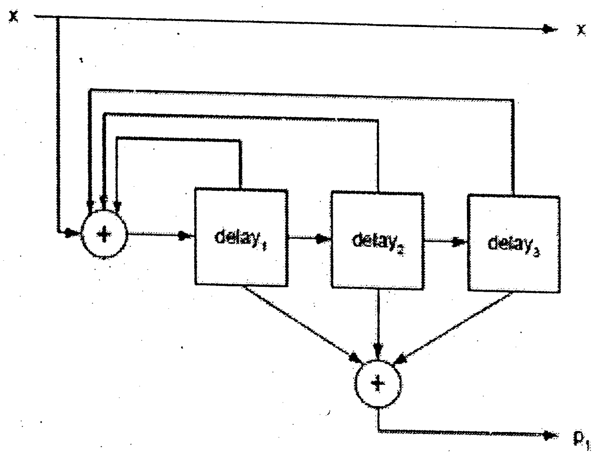

Fig. 1: b. Recursive Systematic Convolution Coders

The input sequence is only presented once at the output. The outputs of the two codersmay be multiplexed into the stream giving a rate R=1/3 code, or they may be punctured to give a rate R=1/2 code. This is illustrated in Figure 2.

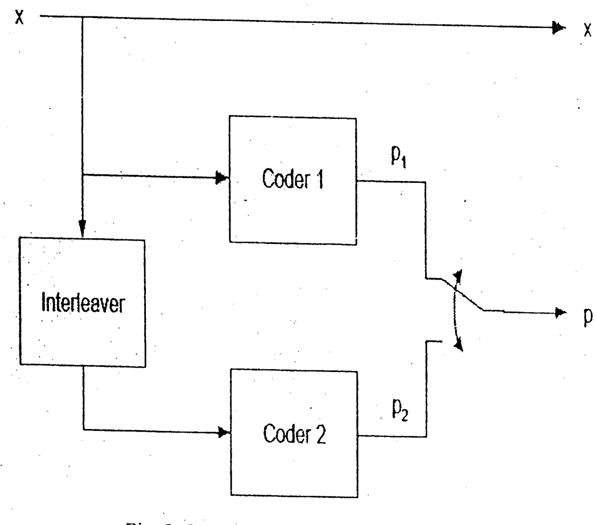

Fig. 2: Punctured Rate R= ½ Turbo Coder

The interleaver design has a significant effect on code performance. A low weight code can produce poor error performance, so it is important that one or both of thecoders produce codes with good weight. If an input sequence x produces a low weight output from coder 1, then the interleaved version of x needs to produce a codeof good weight from coder 2. Block interleavers give adequate performance, butpseudo random interleavers have been shown to give superior performance[3].

B. Decoding

At the receiver, the signal is demodulated with its associated noise and a soft outputprovided to the decoder. The soft output might take the form of a quantized value ofthe decoded bit with its associated noise, or it may be a bit with associated probability(i.e. 1 with P(1)=0.65). Most often it is the log likelihood ratio (LLR), which is defined as:

$$?_i=\ln\left(\frac{P\left\langle m_i=1\middle|y'\right\rangle}{P\left\langle m_i=0\middle|y'\right\rangle}\right)$$

The LLR is a measure of the probability that, given a received soft input y', the message bit mi associated with a transition in the trellis is 1 or 0. If the events are equiprobable then the output is 0, but any tendency for mi towards 1 or 0 will result inpositive or negative values of i.It is simplest to view the decoding process as 2 stages: initialising the decoder and decoding the sequence. The demodulator output contains the soft values of thesequence x' and the parity bits p1' and p2'. These are used to initialise the decoder, as shown in Figure 3a. The interleaved sequence is sent to decoder 2, while thesequence derived from x' is sent to decoder 1 and presented to decoder 2 through aninterleaver. This re-sequences bits from streams x' and p1' so that bits generated fromthe same bit in x are presented simultaneously to decoder 2, whether from x, p1' or p2'.

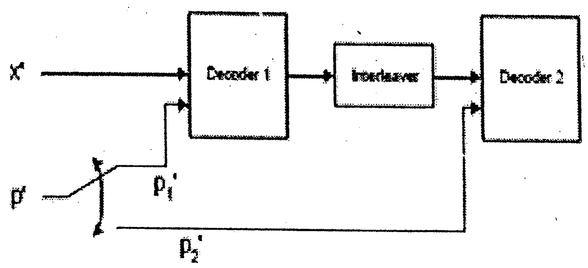

Fig. 3: a. Initialization

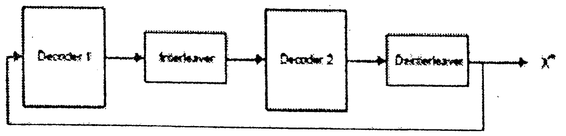

Fig. 3: b. Decoding

The decoder may have some knowledge the probability of the transmitted signal, for example it may know that some messages are more likely than others. This a prioriinformation assists the decoder, which adds information gained from the decoding process forming the a posteriori output. The decoder uses all this information tomake its best estimate of the received sequence. The output is then de-interleaved and presented back to decoder 1, which makes its best estimate. Further iterations through decoders 1 and 2, with associated interleaving and de-interleaving, refine the estimateuntil a final version of the block, x'', is presented at the output. This process is shown in Figure 3b.

.The two main types of decoder are Maximum A Posteriori (MAP) and the Soft Output Viterbi Algorithm[4] (SOVA). MAP looks for the most likely symbol received, SOVA looks for the most likely sequence. Both MAP and SOVA perform similarly at highEb/No. At low Eb/No MAP has a distinct advantage, gained at the cost of addedcomplexity.MAP was first proposed by Bahl[5] and was selected by Berrou[2] as theoptimal decoder for turbo codes. MAP looks for the most probable value for each received bit by calculating the conditional probability of the transition from theprevious bit, given the probability of the received bit. The focus on transitions, or state changes within the trellis, makes LLR a very suitable probability measure for use in MAP. SOVA is very similar to the standard Viterbi algorithm used in hard demodulators. Ituses a trellis to establish a surviving path but, unlike its hard counterpart, comparesthis with the sequences that were used to establish the non-surviving paths. Where surviving and non-surviving paths overlap the likelihood of that section being on the correct path is reinforced. Where there are differences, the likelihood of that section of the path is reduced. At the output of each decoding stage the values of the bit sequence are scaled by a channel reliability factor, calculated from the likely output sequence, to reduce the probability of over-optimistic soft outputs. The sequence and its associated confidence factors are then presented to the interleaver for further iterations. After the prescribed number of iterations, the SOVA decoder will output the sequence with the maximum likelihood.

Wang [6] has explored the performance of codes with parameters set to values that are more practical. He determined that the performance of turbo codes was influenced by four main factors: the number of iterations, constraint length, interleaver design and puncturing. His simulations were based on use of the MAP decoder and constraint lengths of 35. He concluded that:

- Ten decoding iterations are adequate.
- A constraint length of greater than 3 adds little coding gain for short block sizes (around 100)
- Random interleavers provide the best all round performance.

- The use of 1/2 rate punctured codes degrades the BER performance by only 0.5 to 0.7 dB relative to 1/3 rate unpunctured codes.

III. System Model

The lowpass-equivalent simulation model for synchronous downlink transmission used in this study is shown in the Fig. 4. Transmitter section consists of turbo code encoder, external interleaver, CDMA spreading followed by BPSK modulation. Information bit rate of 64 kbit/s is considered and frame length of 10 ms (asspecified for UMTS [7]) is used, which gives the block size of turbo code to be 640 bits. Data bits for eachblock are generated by a random data generator and passed to the turbo code encoder.After rate-1/2 turbo encoding, coded data bits at the rate of 128 kbit/s are interleaved by an external interleaver.An external row–column interleaver of 100 ms (size 1280′10) is used to interleave the codewords. A shorter interleaver (10 ms, size 128′10) is also considered to investigate the effect on performance due to increase in channel correlation. To approximate the UMTS chip rate 4.096 Mchip/s [7], a Gold code of length 31 is used for spreading. The resulting signal at chip rate of 3.968 Mchip/s is modulated using binary PSK modulation.

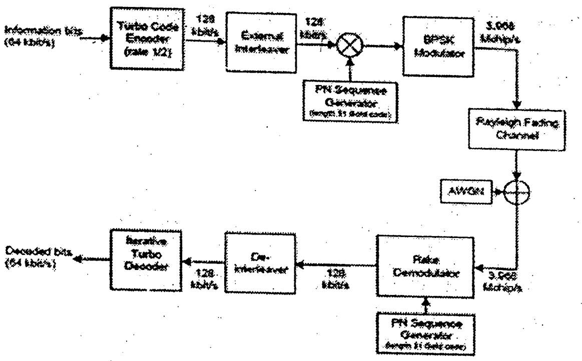

Fig. 4: System Model for Simulation

Since ideal time synchronization is assumed, bandlimiting transmit and receiver filters are omitted in the simulation model and the output signal of the transmitter is sampled at the chip rate. A six-tap Rayleigh fading channel is considered assuming a typical urban environment having an exponential power delay profile with rms delay spread of 0.4 ms. Complex Gaussian samples are filtered by a classical Doppler filter in each tap to introduce correlated fading. Typical high and low (100 and 5 km/h) mobile speeds are considered at the carrier frequency 2 GHz. Hence maximum Doppler shifts for high and low mobile speeds are 185 Hz and 9 Hz, respectively. Doppler filters are implemented as interpolation filters using polyphase network approach for efficient simulation.A three-finger Rake receiver is used for demodulation and despreading of the received signal. Coherent detection assuming ideal channel estimation is used with maximal ratio combining to extract energy from the three strongest paths. Soft channel outputs are computed in the Rake demodulator. Soft channel outputs are computed using

$$\lambda_l(C_{l,k}) = \frac{4A_sR_c}{GP_t}\left(\frac{E_b}{N_o}\right)y_{l,k}$$

which has been derived in [9] for AWGN channel. In the Equation 1, As is the average signal amplitude, Rc is code rate, G is processing gain, and Pt is average signal power of the lowpass equivalent received signal. In our simulations, values Rc=1/2, G=31, Pt=1 and As=1 have been used. Eb/N0 is calculated from the exact variance used in the noise source of the simulation model. Channel soft outputs are then passed to the turbo decoder after deinterleaving. Iterative turbo decoding is implementedusing the log-MAP algorithm in the SISO decoders as described in the previous section. We also compare the turbo code with an equal decoding complexity convolutional code. Since the computational complexity of the log-MAP algorithm is approximately twice the complexity of soft Viterbi decoding, a memory-6 convolutional code and the turbo code used in this study with 4 iterations require approximately equal decoding complexity. Hence, rate 1/2 maximum-free-distance memory-6 convolutional code with generators (133,177)octal [10] is used in place of the turbo code with the same channel model and external interleaving schemes.

IV. Simulation Results

Performance is evaluated by estimating bit error rate (BER) versus the signal-to-noise ratio Eb/N0 based on the system model described above. This study pursues two goals: first, the performance of turbo code is evaluated within normal range of mobile speeds and realistic interleaving delays to investigate the effect of correlated fading. Second, turbo codes are compared with convolutional codes under the constraint of same computational decoding cost. Figures 5 and 6 show the performance results for 100 and 5 km/h mobile speeds, respectively. Up to six decoding iterations have been simulated, but the results after 6th iteration are only shown for the 100 ms external interleaver. However, these results show that no significant improvement in performance can be observed for increasing the number of decoder iterations beyond four in any of the simulated cases. Hence, results for the 10 ms interleaver are only shown for 1st and 4th iterations.

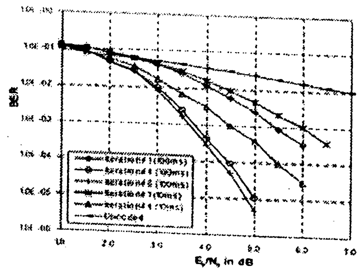

Fig 5: Performance of Turbo Code in Rayleigh Multipath Fading Channel at 100 km/h Mobile Speed

As seen from the Fig. 5, at 100 km/h mobile speed performance degrades by about 1 dB at BER 10^{-4} when the small (10 ms) external interleaver is used instead of the large (100 ms) external interleaver at 4th iteration. When smaller external interleaving delay is chosen, data symbols affected by long fades can not be anymore recovered by iterative decoding. Although BER region of 10^{-6} could not be simulated because of very long simulation time required, values of required Eb/N0 have been estimated using least square curve fitting by assuming exponential decay of the BER curves. The estimated Eb/N0 values for BER of 10^{-6} are 5.6 and 7.2 dB for large and small interleavers, respectively. For mobile speed of 5 km/h, performance degrades by about 1.4 dB when using small external interleaver instead of large interleaver as visible from the Fig. 6.

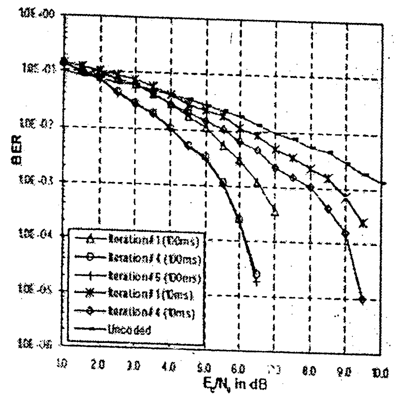

Fig. 6: Performance of Turbo Code in Rayleigh Multipath Fading Channel at 5 km/h Mobile Speed

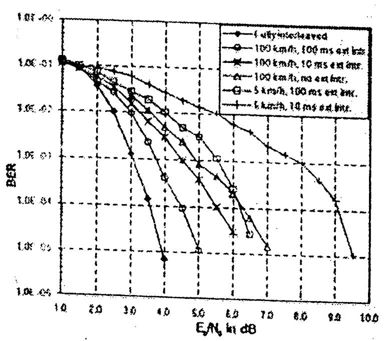

Fig. 7: Comparision of Performance after 4th Decoding Iteration, Including Results for Fully Interleaved Channel

Extrapolated values of Eb/N0 for BER of 10^{-6} from the Fig. 6 are 7.4 dB and 11.5 dB for large and small external interleavers, respectively.Effect of mobile speed on performance may be observed by combining results of Fig. 5 and 6. To see the erformance achievable in ideal conditions, performance has also been evaluated on fully interleaved Rayleigh fading channel, which has been simulated by using uncorrelated channel coefficients for successive bits. These results are presented in the Fig.7. The performance without using any external interleaver is shown as well to indicate the worst case at mobile speed of 100 km/h. Compared to ideal interleaving, about 0.9 dB and 2.0 dB degradation at BER 10^{-4} for 100 km/h mobile speed can be observed when 100 or 10 ms external interleavers are used, respectively.Use of small size interleaver (10 ms) with low mobile speed (5 km/h) is found to be the most critical case, showing a large degradation in performance of about 5.5 dB at BER 10^{-5} compared to the ideal interleaving. Performance with memory-6 convolutional code in similar system configuration is shown in the Fig. 8 and 9 for 100 km/h and 5 km/h mobile speeds, respectively. Comparison with performance of turbo code when decoding at 4th iteration is also shown. There is about 1dB improvement in using the turbo code compared to the convolutional code at BER of 10^{-4} for 100 km/h speed when 100 ms external interleaver is used.

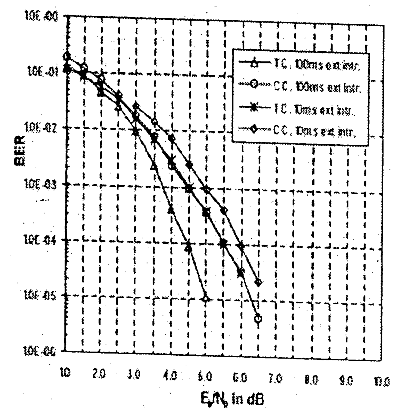

Fig. 8: Comparision Betwwen of Performance of Turbo Code after 4th Iteration and the Equal Decoding Complexity Convolution Code (CC) at 100 km/h Mobile Speed

Improvement by turbo code is reduced, when 10 ms interleaver is used, to about 0.7 dB. For the 5 km/hmobile speed, coding gains for the turbo code are found to be 0.8 dB and 0.4 dB when using 100 ms and 10 ms external interleavers, respectively. Turbo code shows lower coding gain when channel correlation is increased. However, in all of the simulated cases, performance improvement in using turbo code compared to convolutional code is still visible in the Fig. 8 and 9.

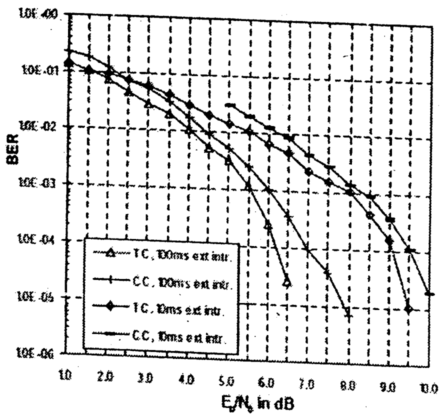

Fig. 9: Comparision betwwen of Performance of Turbo Code After 4th Iteration and the Equal Decoding Complexity Convolution Code (CC) at 5 km/h Mobile Speed

V. Conclusion

Turbo codes are a class of convolution code which exhibits the properties of large block codes through the use of recursive coders. Coder performance is heavily dependent on the design of the interleaver, which must ensure adequate weight for at least one of the codes. Soft decoders are used with turbo codes to allow the a posteriori probability to be passed between decoder iterations. The MAP decoder is generally preferred because it offers the same performance as SOVA at 0.5dB lower value of Eb/No. This performance edge is achieved at the cost of increased complexity. In this study, performance of turbo codes in correlated multipath fading channels also has been studied. Typical wideband DS-CDMA downlink transmission for information bit rate of 64 kbit/s has been considered; with system parameters closely matched to the proposed third generation system UMTS. Simulation results show significant degradation in performance of turbo code due to increase in uncompensated correlation in fading for low mobile speeds and small external interleaver size. For low mobile speeds, large-size external interleaver should be used to uncorrelate fading. Nevertheless, compared to a conventional convolutional code of equal decoding complexity, coding gain for the turbo code can be observed, even for correlated fading. Hence, turbo code is found attractive to be used with next-generation WCDMA systems - especially for high bit-rate applications that can tolerate some interleaving delay.

References

[1] Shannon CE. "Communication in the presence of Noise". Proceedings IRE, Jan 1949, reprinted Proceedings IEEE, Vol 86, no 2 Feb 98.

[2] Berrou C, Glavieux A & Thitimajshima P. "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes". IEEE 1993.

[3] Andersen JD and Zyablov VV, "Interleaver Design for Turbo Coding", Proceedings of the International Symposium on Turbo Codes, September 1997.

[4] Berrou C, Adde P, Angui E & Faudeil S. "A Low Complexity, Soft Output, Viterbi Decode Architecture". Proceedings IEEE. International Conference on Communications. May 1993.

[5] Bahl LR, Cocke J, jelinek F & Rajiv J. "Optimal Decoding of Linear Codes for Minimising Symbol Error Rate". IEEE transactions on Information Theory, March 1974.

[6] Wang CC. "On the Performance of Turbo Codes". IEEE, 1998.

[7] Nikula, E., Toskala, A., Dahlman, E., Girard, L. and Klein, A., " FRAMES Multiple Access for UMTS and IMT-2000," *IEEE Personal Communications*, April, 1998, pp. 16–24.

[8] Sklar B. "Turbo Code Concepts made easy, or How I learned to Concatenate and Reiterate". IEEE 1997.

[9] Hagenauer, J., Offer, E. and Papke, L., "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Information Theory*, Vol. 42, No. 2, March 1996, pp. 429–445.

[10] Proakis, G., *Digital Communications*, Third Edition, McGraw-Hill Inc.