CAPACITY IMPROVEMENT WITH TURBO CODES IN THE WCDMA ENHANCED UPLINK CHANNEL

Nitesh Vasava
Dr.C.H.Vithalani
Prof. K.R.Parmar
Sangramsinh Damor

M.E. Student, EC Department, L. D. College of Engineering, Ahmedabad, Gujarat, India
M.E. Student, EC Department, L. D. College of Engineering, Ahmedabad, Gujarat, India
H.O.D, EC Department, Government Engineering College, Rajkot, Gujarat India.
Prof., EC Department, L. D. College of Engineering, Ahmedabad, Gujarat, India.
nitesh_mdec@yahoo.co.in
sangram.esd@gmail.com

Abstract— Turbo coding is a very powerful error correction technique that has made a tremendous impact on channel coding. It outperforms all previously known coding schemes by achieving near Shannon limit error correction using simple component codes and large interleavers. The iterative decoding mechanism, recursive systematic encoders and use of interleavers are the characteristic features of turbo codes. The use of turbo codes enhances the data transmission efficiency in digital communications systems. This technique can also be used to provide a robust error correction solution to combat channel fading. The entire turbo coding scheme consists of recursive systematic encoders, interleavers, puncturing and the decoder. In order to obtain high coding gains with moderate decoding complexity, concatenation has proved to be an attractive scheme. Various concatenating coding scheme of different degree of complexity are finding applications in digital satellite communication system, mobile communication system, deep space mission and elsewhere. Many new concatenated coding schemes are being proposed.

Keywords- channel coding, turbo codes, interleaving, BER, low decoding complexity.

I. INTRODUCTION

Turbo codes by Berrou et al. [1], turbo coding has raised great interest in digital communication area. Turbo codes achieve near optimum error correction performance with relatively low decoding complexity compared to similar coding schemes. Due to their superior performance, turbo codes have been adopted as channel coding schemes for many applications, including the Third Generation Partnership Project (3GPP) standard. The 3GPP system address increasing demands for higher data rates in variety of services in wireless communication.

Despite the attractive performance of the turbo code in the 3GPP system, its implementation still faces different difficulties especially for mobile equipments. In these devices power consumption, memory requirements, size and cost are vital issues. While different studies have been conducted to examine low complexity and memory efficient algorithms for implementation of the turbo codes, only a few of them have presented results in the 3GPP system context [1]. The faster transmission rate in order to send the same amount of information bits per unit time implying a larger bandwidth requirement. The advantage however, is that the Signal to Noise Ratio (SNR) can be reduced significantly (also referred to as Coding Gain). In wireless systems, one of the most important performance criterion is low power transmission as that can provide a longer battery life and lesser co-channel interference.[2].

From coding theory [1], it is known that by increasing the codeword length or the encoder memory and using “good” codes, one can theoretically approach the limiting channel capacity. However finding such a code and implementing such a decoder in real-time has been an active area of research for a very long time. Turbo codes [2], is a very powerful error correcting technique, which enable reliable communication with Bit Error Rate (BER) close to Shannon limit [3]. Turbo codes are in fact a parallel concatenation of two recursive systematic convolutional codes. The fundamental difference between convolution codes and turbo codes is that while for the former, performance improves by increasing the constraint length, for turbo codes it has a small value which remains pretty much constant. Moreover, it achieves a significant coding gain at lower coding rates.

A class of parallel and serial concatenated codes, often called “turbo” codes, which achieve near-Shannon-limit error correction performance with low decoding complexity. The exceptional performance of these codes is explained, together with the basic structure of the encoders and decoders. These new codes are currently under consideration by CCSDS to define a new coding standard for space telemetry.[3].

II. TURBO CODES

Channel coding in 3GPP system includes different coding strategies depending on the type of the channel and data rate. These coding schemes are no coding, 1/2 or 1/3 convolution codes, and a turbo code. Higher date rate services require use of turbo coding. It has been estimated that roughly 300 bits should be available per Transmission Time Interval (TTI) in order to give turbo coding some gain over convolutional
coding. According to the 3GPP standard, the input data length in this system varies between 40 to 5114 bits and therefore the interleaver length must change accordingly [1].

The 3GPP FDD mode specifies a traditional structure of Parallel Concatenated Convolutional Code (PCCC) with two identical 8-state constituent Recursive Systematic Convolutional (RSC) encoders. The coding rate for the turbo code is $R = 1/3$, but after adding termination bits and rate matching, the overall rate of the code increases depending on the block size. Figure 1 shows the standard turbo encoder for 3GPP FDD mode. Output sequence from the turbo coder is $x_1, z_1, z'_1, x_2, z_2, z'_2, \ldots, x_N, z_N, z'_N$, where $x_1, x_2, \ldots, x_K$ are the input bits to the turbo encoder i.e. both the first constituent encoder and the turbo code internal interleaver, $N$ is the number of bits, $z_1, z_2, \ldots, z_N$ and $z'_1, z'_2, \ldots, z'_N$ are the output bits from the first and second constituent encoders, respectively.

The output bits from the turbo code internal interleaver are denoted by $x'_1, x'_2, \ldots, x'_N$, and these bits are the inputs to the second constituent encoder. The transfer function of both constituent encoders is

$$G(D) = \begin{bmatrix} 1 & 1 + D + D^3 \\ 1 + D^2 + D^3 \end{bmatrix}$$

The initial value of the shift registers of the 8-state constituent encoders are all zeros. In order to improve the performance of the turbo code, encoder termination is applied to both RSC encoders individually. The systematic termination bits for both RSC encoders are sent to the rate matching block.

III. INTERLEAVER

The interleaver is a very important constituent of the turbo encoder. It spreads the bursty error pattern and also increases the free distance [4]. Thus, it allows the decoders to make uncorrelated estimates of the soft output values. The convergence of the iterative decoding algorithm improves as correlation of the estimates decreases. [3]. by spreading the source bits over time, it becomes possible to make use of error control coding, which protects the source data from corruption by the channel.

An interleaver can be one of two forms: a block structure or a convolutional structure. A block interleaver formats the encoded data into a rectangular array of $m$ rows and $n$ columns, and interleaves $m \times n$ bits at a time. Usually, each row contains a word of source data having $n$ bits of degree $m$ consists of $m$ rows. The source bits are placed into the interleaver by sequentially increasing the rows number for each successive bit, and filling the columns. The interleaver source data is then read out row-wise and transmitted over the channel. this has the effect of spreading the original source bits by $m$ bit periods[4].

IV. PUNCTURING

Puncturing [4] is a technique used to increase the code rate. The multiplexer can choose the odd indexed outputs from the output of the upper encoder and its even indexed outputs from the lower one. In a more complicated system, puncturing tables are used. An important application of puncturing is to provide unequal error protection where relatively unimportant bits or during cleaner channel condition a lower rate coding is used by puncturing the coded bits while for more important bits or noisy channel conditions, higher rate coding can be used.

V.PERFORMANCE SIMULATIONS

Simulations by many researchers have shown that turbo codes can achieve small error probabilities using a signal-to-noise ratio (SNR) just slightly higher than the capacity limit, when the turbo code block is very large (104 information bits or more). Fig. 2 shows the simulated performance of a family of turbo codes of rates $r = 1/2, 1/3, 1/4$ and $1/6$, constructed for an information block length of 10200 bits. For these results, the decoder made its final decoding decisions after 10 iterations.

![Fig2. Performance curves for various turbo codes](image-url)
The turbo codes in Fig. 2 are systematic parallel concatenated codes with two recursive convolutional components. The backward connection vector for both component codes is 10011. The forward connection vector for both component codes and rates 1/2 and 1/3 is 11011. The forward connection vectors for rate 1/4 are 10101, 11111 for the first component code, and 11011, 11111 for the second component. The forward connection vectors for rate 1/6 are 11111, 11101, 10111 for the first component code, and 11011, 11111 for the second component. Puncturing of every other symbol from each component is necessary for rate 1/2. No puncturing is done for rates 1/3, 1/4, and 1/6.

To achieve a bit error rate (BER) of $10^{-6}$, threshold bit-SNRs $(E_b/N_0)$ of approximately -0.1 dB, +0.15 dB, +0.4 dB, and +1.0 dB, are required by the turbo codes of rates 1/6, 1/4, 1/3, and 1/2, respectively. These same threshold bit-SNRs achieve a codeword error rate (WER) of approximately $10^{-4}$. Turbo codes gain a significant performance improvement over the traditional Reed-Solomon and convolutional concatenated codes currently used by JPL [5].

V. LOWER BOUND ON CODE PERFORMANCE

Shannon's sphere packing bound [5] provides a lower limit to the error rate achievable by an arbitrary code of a given block size and code rate. The sphere packing bound is shown in Fig. 4, for the case of equal-energy signals applied to a continuous-input additive white Gaussian noise (AWGN) channel. The sphere-packing bound would be reached with equality only if the code were a perfect code for the continuous-input AWGN channel. Although perfectness is generally an unattainable goal, it can be a good benchmark for code performance. We define the imperfection of a given code as the difference between the code’s required $E_b/N_0$ to attain a given $P_w$, and the minimum possible $E_b/N_0$ required to attain the same $P_w$, as implied by the sphere-packing bound for codes with the same block size $k$ and code rate $r$. These differences, measured in dB, are shown in Fig. 3 for various codes, with $P_w = 10^{-4}$.

VI. UPPER BOUND ON CODE PERFORMANCE

Upper bounds on the error rate achievable by maximum likelihood decoding of a specific turbo code have been obtained by a union bounding technique [3]. These bounds are expressed in terms of the joint input and output weight distribution of the constituent codes, and they assume random, independently chosen permutations of the input data before each constituent encoding. The upper bounds on turbo code performance accurately predict the actual turbo decoder’s performance in the so-called “error floor” region above the “computational cut-off rate” threshold, below which the bounds diverge and are useless. The bounds prove that the turbo code’s performance curve does not stay steep forever as does that of a convolutional/Reed-Solomon concatenated code. When it reaches the error floor, the curve flattens out considerably and from that point onward looks like the performance curve of a weak convolutional code. The error floor is not an absolute lower limit on achievable error rate, but it is a region where the slope of the error rate curve becomes dramatically lower.

Figure 4 gives an illustration of the transition of turbo code performance curves from a steep “waterfall” region into a much flatter “error floor” region for two representative turbo codes. This figure shows the simulated performance of each code, compared with the theoretical upper bound and also with an extrapolation of the theoretical bound that gives a tighter approximation to actual performance on the error floor below the computational cut-off rate threshold. The original turbo codes developed by Berrou et al [1] had error floors starting at a BER of about $10^{-5}$. By using predictors derived from the theoretical bounds, we have been able to design good...
turbo codes that lower the error floor to possibly insignificant levels.

VII. CODE DESIGN CONSIDERATIONS
Turbo codes give the system designer vast flexibility to choose any desirable combination of code parameters without sacrificing performance more than intrinsically necessary.

1. Code Rate and Block Size — The code rate of the currently recommended CCSDS turbo encoder is selectable from 1/2, 1/3, 1/4, or 1/6, and the block size is selectable from 1784, 3568, 7136, 8920, or 16384 bits. We have seen that a single family of turbo codes encompassing this range of rates and block sizes is uniformly nearly perfect (within 0.7 dB) with respect to the theoretical lower bounds [3]. Therefore, many system tradeoffs involving code rate and block size can be adequately evaluated by reference to the universal theoretical bounds rather than by painstaking simulation of turbo codes of each rate or size.

2. Constituent Codes — Effective turbo codes can be constructed from a wide variety of constituents.
   (a) Number and Type of Constituent Codes: Turbo codes with more than two constituent codes are feasible in principle, but have not been well studied (mainly because two-component turbo codes already perform so well). The best performing and best understood constituent codes discovered thus far are recursive convolutional codes, as recommended in the original turbo code paper by Berrou et al [1].
   (b) Constraint Length: The currently recommended turbo code is formed from two recursive convolutional codes with constraint length K=5. Higher constraint lengths are more complex to decode, and seem to offer negligible performance improvement. Constituent codes with constraint lengths less than 5 may sometimes be desirable to achieve higher decoding speeds with some sacrifice of performance.
   (c) Code Generator Polynomials: Considerable theory has been developed to guide the choice of constituent code generator polynomials.

3. Permutation— The best understood permutations for turbo codes are completely random. However, pseudo-random sequences generated by simple randomizing algorithms (such as feedback shift register sequences) do not generate good turbo code permutations. Best performance is achieved by a manually optimized “S-random” permutation [1], but an optimized permutation must be stored in memory because it is infeasible to recompute for every codeword. Berrou [5] developed an algorithm for generating practically computable permutations that perform pretty well.

CONCLUSION
The advantage of turbo codes over existing coding schemes is that it attains a very low bit error rate at low signal-to-noise ratios. This makes it suitable for wireless applications where low transmission power is desired. However, the performance of turbo codes on Rayleigh and Ricean fading channel remain an active subject of research. The probability of this code to Advanced Digital System (ADS) would be very beneficial for code re-usability and also the synthesis capability of ADS would result in faster product development. Applications of turbo codes include deep space communications, coding for ATM (Asynchronous Transfer Mode) and wireless applications, fading channels, digital direct broadcast satellite services, CDMA (Code Division Multiple Access), channel equalization, combined carrier estimation and decoding, wireless LAN, digital TV, cable modem, and DSL (Digital Subscriber Line) systems. They have also been adopted in 3GPP (3rd Generation Partnership Project) WCDMA (Wideband CDMA) and DVB-RCS (Digital Video Broadcast – Return Channel Satellite) systems.

REFERENCES