

Performance Analysis of Turbo Codes In CDMA Under AWGN and Fading

M.Raja Murali Krishna¹, M.Tata Ahish², Ch.Bhuvaneshwari³, H.Vinod⁴,
R.V.Ch.Sekhar Rao⁵

Dept of ECE, Lendi Institute of Engineering and technology, Vizianagaram, Andhra Pradesh, India

Abstract: In order to have reliable communications, channel coding is often employed. Turbo code as a powerful coding technique has been widely studied and used in communication systems. Turbo code decoder algorithm is studied in details in this paper. The performance of Turbo code used in IS-2000 Code Division Multiple Access (CDMA) reverse or forward link under Additive White Gaussian Noise (AWGN) and slow fading channels is evaluated. The bit error rates (BER) of Turbo code at low signal-to-noise ratio (SNR) are obtained by simulations.

Index terms: IS-2000 CDMA, RSC encode, Interleaver, Output puncturing, Fading, AWGN channel, Rayleigh fading channel, Uncorrelated Rayleigh fading generator, Correlated Rayleigh fading generator

I. Introduction

Error corrective coding is used to enhance the efficiency and accuracy of information transmitted. In a communication transmission system, data is transferred from a transmitter to a receiver across a physical medium of transmission or channel. The channel is generally affected by noise or fading which introduces errors in the data being transferred. Error-correcting code is a signal processing technique used for correcting errors introduced in the channel. It is done by encoding the data to be transmitted and introducing redundancy in it such that the decoder can later reconstruct the data transmitted using the redundant information. Fig.1 is a block diagram of a simplified model of a coded system.

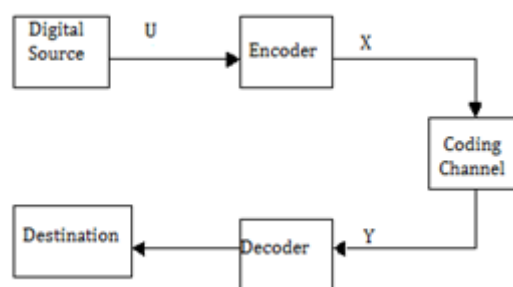


Fig 1: Simplified Model of a Coded System

The information source is a digital source. If it is not, it will be converted to digital. The digital source output u is sent to an encoder to generate encoder output x which is modulated and then transmitted over the physical channel. The decoder will make a best guess \tilde{u} of the original information u based on the received signal y which is distorted by the channel.

A major concern in coding technique is control of errors so that reliable communications can be obtained, i.e., \tilde{u} is as close to u as possible. There are many coding schemes available. Turbo code is the most exciting and potentially important development in the coding theory in recent years. This powerful code is capable of achieving near Shannon capacity performance [1].

There are many papers discussing the Turbo codes [2]. The first one is related to the Maximum A Posteriori (MAP) decoding algorithm for Convolutional codes which was proposed in 1974 by Bahl et al.[3], but initially received very little attention because of its increased complexity over alternative Convolutional decoder for a minimal advantage in bit error rate (BER) performance. Recently, however, an iterative decoder developed by Berrou et al. [1] in 1993 has enjoyed renewed and greatly increased attention. In this paper [1], they considered the iterative decoding of two Recursive Systematic Convolutional (RSC) codes concatenated in parallel through a non-uniform interleaver. For decoding the component codes they used a Soft Input/Soft Output (SISO) decoder based on the MAP algorithm. Since then, researchers around the world have investigated the performance and design of Turbo codes. Both serial and parallel concatenated Convolutional codes have been studied, e.g.. The interleaver patterns with high performance also have been evaluated, e.g. In iterative decoding, several decoding algorithms have been used, including the optimal MAP symbol estimation and its

simplification called the max-log-MAP algorithm (Additive MAP Algorithm) [4] and [6]. A further simplification of log MAP is offered by the modified soft-output Viterbi algorithm (SOVA) which works in a sliding-window SISO decoding algorithm [4].

In this paper we are evaluating the BER performance of Turbo codes used in IS-2000 CDMA reverse or forward link under Additive White Gaussian Noise (AWGN) and slow fading channels. In addition, this paper presents more accurate approximation in the modified Turbo code decoding algorithm under slow Rayleigh fading channel and corrects the error in the decoding algorithm under AWGN channel presented.

II. Turbo Encoder

In this section, a system model of Turbo code is introduced, as an example, IS-2000 CDMA system is briefly discussed, then focus on a general Turbo encoder which consists of the RSC encoder, interleaver, and puncturing. Convolutional codes can be used to encode a continuous stream of data, but in this case we assume that data is configured in finite blocks - corresponding to the interleaver size. The frames can be terminated - i.e. the encoders are forced to a known state after the information block. The termination tail is then appended to the encoded information and used in the decoder.

A. Turbo code system model:

The basic elements of a communication system with Turbo code. The source generates an information sequence of N symbols with a constant a priori probability distribution. $p(u_k) = u$ The u_k denotes the transmitted symbol at time k with value $(0, 1)$, i.e., $u_k \in \{0,1\}$. The u_k is encoded by two RSC encoders whose trellis states start at states $s_0(i)$, e.g., $s_0(1)$ and terminal at the final state, $s_k(i) = s_n(1)$ which the final state returns to the starting state for encoding the next information block. k and i denote time index and state index, respectively. The encoder generates a sequence of N output coded symbols c_N . The trellis state structure is supposed to be known at the receiver side. $C_k = \{C_k^1, C_k^2, \dots, C_k^q, C_k^{1,2}, C_k^{1,3}, \dots, C_k^{1,q}\}$ represents one coded symbol at time k with a length of $2q-1$, where $C_k^1, C_k^2, \dots, C_k^q$ are generated by the first RSC encoder, and $C_k^{1,2}, C_k^{1,3}, \dots, C_k^{1,q}$ are generated by the second RSC encoder. Each element C_k^l is binary signal, i.e. $C_k^l \in \{-1,1\}$, $l = 1, \dots, q$. After modulation, the coded symbols are mapped one by one into transmitted signals x_k .

$$x_k = \{x_k^{1,s}, x_k^{2,p}, \dots, x_k^{q,p}, x_k^{1,2,p}, x_k^{1,3,p}, \dots, x_k^{1,q,p}\}$$

represents the transmitted codeword at time k . $x_k^{1,s}$ and $x_k^{2,p}, \dots, x_k^{q,p}, x_k^{1,2,p}, x_k^{1,3,p}, \dots, x_k^{1,q,p}$ are the systematic bit and the parity check bits for the k^{th} symbol, respectively. The signal x_k is transmitted over the stationary memory less channel. At the destination, the decoder will evaluate the demodulator output y based on the statistic characteristic of the channel, i.e., the conditional probability density function (pdf) of y_k represents the received symbol at time k , and then make a decision

$$p\left(\frac{y}{c}\right) \triangleq p(y_k = y | C_k = c)$$

$$y_k = \{y_k^{1,s}, y_k^{2,p}, \dots, y_k^{q,p}, y_k^{1,2,p}, \dots, y_k^{1,q,p}\}.$$

B. IS-2000 CDMA system:

In IS-2000 CDMA reverse or forward link fundamental channel and reverse or forward link supplemental channel, the data encoder employs a Turbo encoder. The data is encoded frame by frame. The frame size is from 384 bits to 20736 bits. Tail bits are added at the end of each frame. Frame size is defined as the number of data bits plus tail bits. The transfer function for a constituent encoder is

$$G(D) = \left[1, \frac{1+D+D^3}{1+D^2+D^3}, \frac{1+D}{1+D} \right]$$

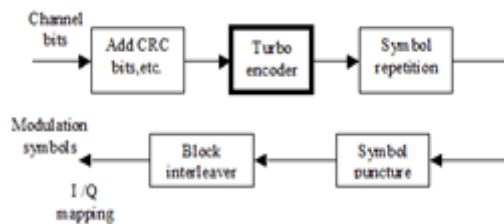


Fig 2: A part of the forward fundamental channel and forward supplement channel structure

C. General Turbo code encoder:

A general Turbo encoder is shown in Fig 3. The Turbo code encoder employs two identical systematic recursive Convolutional encoders connected in parallel with an interleaver (the “Turbo interleaver”) preceding the second recursive Convolutional encoder. The two recursive Convolutional encoders are called the constituent encoders of the Turbo encoder. The information bits are encoded by both RSC encoders. The first encoder operates on the input bits in their original order, while the second encoder operates on the input bits as permuted by the Turbo interleaver. If the input symbol is of length 1 and output symbol size is R, then the encoder is of code rate $rc=1/R$. The information bits are always transmitted across the channel. Depending on the code rate desired, the parity bits from the two constituent encoders are punctured before transmission. The tail bits will be added at the end of the transmitted frame.

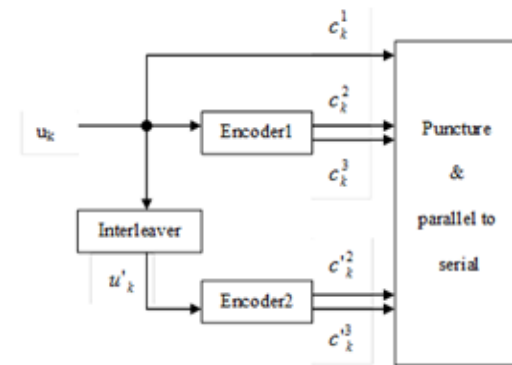


Fig 3: General Turbo code encoder

C.1 RSC encoder:

Fig 4 shows a complete RSC encoder used in IS-2000 Turbo encoder. u_k is an input information bit, c_k is the coded output symbol which is represented by $[c_k^1, c_k^2, c_k^3]$. The system constraint length K is 4. The generator connection matrix is $[1, 15/13, 17/13]$ in octal. Its trellis states are shown in fig 4.

A systematic convolution code is generated by passing the information sequences to be transmitted through a linear finite-state shift register. The input data to the encoder is shifted into the register. Hence code words are generated. The encoder is defined by a set of generator polynomials $G(D) = \left[1, \frac{g_1(D)}{g_b(D)}, \frac{g_2(D)}{g_b(D)}, \dots, \frac{g_m(D)}{g_b(D)} \right]$ and

memory size M of the shift register (or the Convolutional constraint length K which is equal to M+1). $g_m(D)$ is the mth forward generator and $g_b(D)$ is the feed back generator. The information bits are passed into the encoder frame by frame. The Shift registers starts at initialized state and end at the initial state. To ensure this happens, extra bits are needed to flush the memory, which are called encoder tail bits.

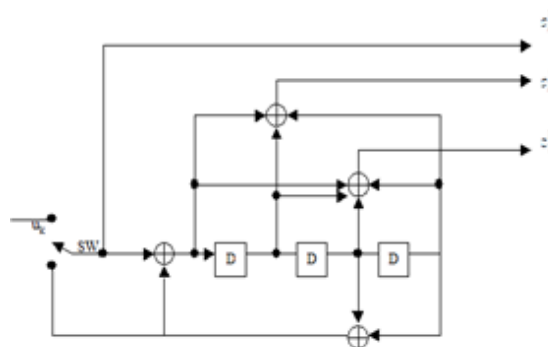


Fig 4: A RSC Turbo encoder with generators

$$G(D) = [1, 15/13, 17/13]$$

For example, switch SW connects to up when data is inputted, down when tail bit is inputted. If the encoder has code rate 1/3, it maps N data bits to 3N coded bits plus 3M tail bits. The received frame for N bit data will have 3N+3M coded bits.

C.2 Interleaver:

A burst of errors is defined as a sequence of bit errors. The method of interleaver has proved to increase the reliability in the burst error channel. In Turbo code, the structure of interleaver has been carefully chosen. It allows that input sequences for which one encoder produces low weight code words will usually cause the other encoder to produce high weight code words. Although the constituent codes are individually

weak, the combination is surprisingly powerful. The resulting code has features similar to a “random” block code with Z information bits.

Consider a simple pseudo-random interleaver with the mapping table The output data bit at position m will take the input data bit at position B(m). The above table can also be represented by an interleaver matrix B = [2 5 4 1 3]. When the input data sequence is [0 1 1 1 0], the interleaver output data sequence will be [1 0 1 0 1].

Table1: Interleaver mapping table

Input data counter m	Input data	Output data counter B(m)	Output data
1	0	2	1
2	1	5	0
3	1	4	1
4	1	1	0
5	0	3	1

Here we use s-random interleaver. The process of creating the interleaver mapping Function is as follows

1. Store the numbers 0 to N-1 in a length-N vector, denoted by $U = \{U_0, U_1, U_2, U_3 \dots U_{n-1}\}$.
2. Choose an integer value for the spread factor M spread. For a given value of N, it is important to choose in order for the interleaver construction method to be successful.
3. Set counter s=0. Repeat the following steps:
 - (a). Randomly pick a number from the list of numbers contained in U .If the chosen number differs by more than \pm when compared to the previous M spread stored in V ,store it as element V_i in the vector $V = \{V_0, V_1, \dots, V_{n-1}\}$ and mark it as unavailable for the next random selection from U. Otherwise repeat step(a)
 - (b) Increment counters i.
 - (c) Repeat (a) to (b) for $0 \leq i < N$.
4. For every i, with $0 \leq i < N$ determine the index j such that . As with random number generator interleaver, the mapping function of the interleaver is then $\pi(i) = j$.

C.3 Output puncturing:

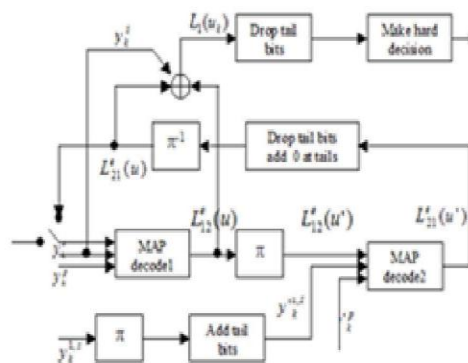
The role of Turbo code puncturing is to periodically delete selected bits to reduce coding overhead. In IS-2000 CDMA Turbo code, the systematic bits c1 in constituent encoder 1 are always transmitted for the non-tail bits, the systematic bits in constituent encoder 2 are always punctured for the non-tail bits. Other puncturing patterns depend on the code rate selected [7].

III. Turbo Code Decodingalgorithm

In this section, a Turbo code decoding algorithm under AWGN channel is reviewed. The modified decoding algorithm under fading channel is also presented.

A. Turbo code decoding algorithm under AWGN channel

AWGN channel model is a simple and common channel model in a communication system. It is easier to be studied. In this section, a Turbo code decoding algorithm under AWGN channel will be discussed. Fig 6 shows a block diagram of a Turbo decoder.



π is the notation for interleaver
 π²is the notation for de interleaver

Fig 6: Block Diagram of Turbo Decoder

A log ratio of the posteriori probability of u_k conditioned on the received signal y is defined as

$$L(u_k) \triangleq \log \left[\frac{P(u_k = 1 / y_1^N)}{P(u_k = 0 / y_1^N)} \right] \quad (1)$$

The decoding decision of \tilde{u}_k is made based on the sign of $L(u_k)$, i.e.,

$$\tilde{u}_k = \text{sign}[L(u_k)] \quad (2)$$

$L(u_k)$ is computed by three terms which are L_{apriori} , L_{channel} , and $L^e(u_k)$. L_{apriori} is a priori information based on the input bit u_k at time k . It is provided by the previous decoder. L_{channel} is the received systematic bit at time k

$$L(u_k) = \left[L^e(u_k) + Lc \cdot y_k^{1,s} \right] + \log \frac{\sum_{u^+} \tilde{\alpha}_{k-1}(s') \cdot \tilde{\beta}_k(s) \cdot \gamma_k^e(s', s)}{\sum_{u^-} \tilde{\alpha}_{k-1}(s') \cdot \tilde{\beta}_k(s) \cdot \gamma_k^e(s', s)} = L_{\text{apriori}} + L_{\text{channel}} + L^e(u_k) \quad (3)$$

Where the, L_{apriori} and L_{channel} denote $L^e(u_k)$ and $Lc \cdot y_k^{1,s}$ respectively. $\sum_{u^+}(\cdot)$ is the summation over all the possible transition branch pair (s_{k-1}, s_k) at time k given input $u_k=1$ and $\sum_{u^-}(\cdot)$ is the summation over all the possible transition branch pair (s_{k-1}, s_k) at time k given input $u_k=0$. Lc is the channel reliable factor, its computation is given as the following,

$$Lc = \frac{4 \cdot A \cdot \text{SNR}_b}{p} \quad (4)$$

Where, $A=1$ for AWGN channel, SNR_b is the uncoded bit-energy-to-noise-ratio $\left(\frac{E_b}{N_0}\right)$, p denotes $1/r_c$, r_c is code rate of the Turbo encoder.

$L^e(u_k)$ is extrinsic information based on all parity and systematic information except the systematic value at time k . It can be passed on to a subsequent decoder. It is computed using the following equations:

$$L^e(u_k) \triangleq \log \frac{\sum_{u^+} \tilde{\alpha}_{k-1}(s') \cdot \gamma_k^e(s', s) \cdot \tilde{\beta}_k(s)}{\sum_{u^-} \tilde{\alpha}_{k-1}(s') \cdot \gamma_k^e(s', s) \cdot \tilde{\beta}_k(s)} \quad (5)$$

Where

$$\gamma^e(s', s) = \exp \left[\sum_{i=2}^q \left(Lc \cdot \frac{1}{2} \cdot y_k^{i,p} \cdot c_k^i \right) \right] \quad (6)$$

$\tilde{\alpha}_k(s)$, $\tilde{\beta}_{k-1}(s')$ can be computed recursively with initial conditions described below

$$\tilde{\alpha}_0(s) = \begin{cases} 1 & \text{if } s = 1 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$\tilde{\beta}_{k-1}(s') = \frac{\sum_s \tilde{\beta}_k(s) \cdot \gamma_k(s', s)}{\sum_s \sum_{s'} \tilde{\alpha}_{k-2}(s') \cdot \gamma_{k-1}(s', s)}, \quad (8)$$

$$\tilde{\beta}_N = \begin{cases} 1 & \text{if } s = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\gamma_k(s', s) \propto \exp \left[\frac{1}{2} \cdot L^e(u_k) \cdot u_k + Lc \cdot \frac{1}{2} \cdot y_k^{1,s} \cdot c_k^1 \right] \exp \left[\sum_{i=2}^q \left(Lc \cdot \frac{1}{2} \cdot y_k^{i,p} \cdot c_k^i \right) \right] \quad (9)$$

For example, at any given iteration, decoder 1 $L_1(u_k)$ is computed as

$$L_1(u_k) = Lc \cdot y_k^{1,s} + L_{21}^e(u_k) + L_{12}^e(u_k) \quad (10)$$

$$\tilde{u}_k = \text{sign}[L_1(u_k)]$$

Where $L_1(u_k)$ is given in equation (3). $L_{21}^e(u_k)$ Is extrinsic information for decoder 1 derived from decoder 2, and $L_{12}^e(u_k)$ is the third term in equation (3) which is used as the extrinsic information for decoder 2 derived from decoder 1. The decoders are sharing the information with each other. The value of $L_1(u_k)$ decides the degree of the reliability of \tilde{u}_k .

B. Modified decoding algorithm under Rayleigh fading channel:

Fading is due to the constantly changing physical characteristics of the media communication system over radio channel. Suppose we are transmitting digital information over the channel by modulating (amplitude, phase, or frequency modulation) the basic pulse $u(t)$ at a rate $1/T$, where T is the bit or symbol interval, the time-variant channel characterized by the transfer function $C(f;t)$ distorts the signal $U(f)$, where $U(f)$ is the Fourier transform of $u(t)$. If $U(f)$ has a bandwidth W greater than the coherence bandwidth $(\nabla f)_c$ of the channel, $U(f)$ is subjected to different gains and phase shifts across the band. In such a case the channel is said to be frequency-selective. Additional distortion is caused by the time variations in $C(f;t)$. This type of distortion is evidenced as a variation in the received signal strength and has been termed fading. The effect of the fading channel on the transmitted signal $u(t)$ is a function of the choice of signal bandwidth and signal duration. If we select the signaling interval T to satisfy the condition $T \gg T_m$ (T_m is the multipath spread of the channel), the channel introduces a negligible amount of inter-symbol interference (ISI). If the bandwidth of the signal pulse $u(t)$ is $W=1/T$, the condition $T \gg T_m$ implies that $W \ll 1/T_m \approx (\nabla f)_c$, i.e., the signal bandwidth W is much smaller than the coherence bandwidth $(\nabla f)_c$ of the channel. Hence the channel is frequency non-selective. In other words, all of the frequency components in $U(f)$ undergo the same attenuation and phase shift in transmission through the channel. The transfer function $C(f;t)=C(0;t)$ for a frequency non-selective channel can be expressed in the form

$$C(0;t) = \alpha(t) e^{j\phi(t)} \quad (11)$$

where $\alpha(t)$ and $\phi(t)$ represent the envelop and the phase of the equivalent low pass channel, respectively. When $C(0;t)$ is modeled as a zero mean complex valued Gaussian random process, the envelope $\alpha(t)$ is Rayleigh distributed for any fixed value of t and $\phi(t)$ is uniformly distributed over the interval $(-\pi, \pi)$. $(\nabla f)_c$ denotes the coherence bandwidth of the channel and $(\nabla t)_c$ denotes the coherence time of the channel, B_d is the Doppler spread of the channel.

$$(\nabla f)_c \approx 1/T_m \quad (12)$$

$$(\nabla t)_c \approx 1/B_d \quad (13)$$

A slowly changing channel has a large coherence time or a small Doppler spread. If we select the signal bandwidth W to satisfy the condition $W \ll (\nabla f)_c$ and the signaling interval T to satisfy the condition $T \ll (\nabla t)_c$, the channel attenuation and phase shift are essentially fixed for the duration of at least one signaling interval since T is smaller than the coherence time of the channel. We call the channel a slowly fading channel. The fading function or channel gain is described by a probability density function and frequency dispersion measure B_d which is often referred to as the Doppler spread, or bandwidth. Figure 7 shows the model for the fading channel. In this model, Binary Phase Shift Keying (BPSK) signaling is assumed with coherent detection which implies that $\phi(t)$ is known at the receiver. The channel gain $\alpha(t)$ is described by a Rayleigh distributed random variable with probability density function $p(\alpha) = 2 \cdot \alpha \cdot e^{-\alpha^2}, \alpha \geq 0$. Aside from the channel gain, additive noise $n(t)$ is present in the channel. This additive noise is assumed to be white Gaussian noise (AWGN) with zero mean and power spectral density $N_o/2$ (W/Hz).

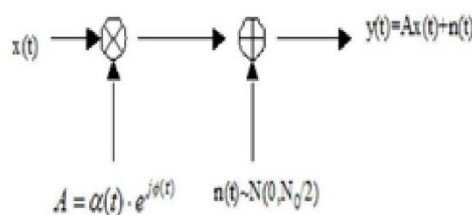


Fig 7: Block diagram of Raleigh fading channel model for a transmission system

IV. Stimulation Results

In this section, the Rayleigh fading generators are discussed, and then simulation BER results for the Turbo decoder are presented.

A. Implementation of Rayleigh fading generators:

The characteristics of the Rayleigh fading process will affect the Turbo decoder performance. Two different structures are used to generate Rayleigh fading processes in the simulation. One is to generate an uncorrelated Rayleigh fading process; the other is to generate a correlated Rayleigh fading process.

A.1 Uncorrelated Rayleigh fading generator:

Fig 8 shows a Rayleigh fading channel gain generator. This generator generates a_s and a_c which are i.i.d. Gaussian distributed variables with zero mean and unit variance, and. The auto-correlation of the random variable α is time independent (uncorrelated).

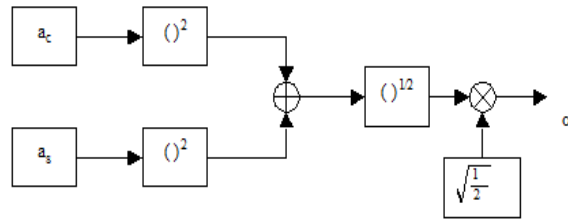


Fig 8: Rayleigh fading generator

A.2 Correlated Rayleigh fading generator:

Another simulator to generate Rayleigh fading channel gain is by using mathematical functions, which is called Jake fading generator [10]. a_c and a_s are given as the following:

$$a_c = \frac{2}{M_0} \left(\sum_{n=1}^{M_0} \cos \beta_n \cos \omega_n t + \sqrt{2} \cos \zeta \cos \omega_n t \right) \quad a_s = \frac{2}{M_0 + 1} \left(\sum_{n=1}^{M_0} \sin \beta_n \cos \omega_n t + \sqrt{2} \sin \zeta \cos \omega_n t \right)$$

$$A_1 = \frac{\left[(a_c)^2 + (a_s)^2 \right]^{1/2}}{\sqrt{2}}$$

and $M_0 = \frac{1}{2} \left(\frac{M_1}{2} - 1 \right)$, $\beta_n = \frac{\pi \cdot n}{M_0}$, $\zeta = \pi/4$, and $\omega_n = \omega_m \cos \left(\frac{2 \cdot \pi \cdot n}{M_1} \right)$ where M_0 is the number of low

frequency oscillators with frequencies equal to ω_n . a_s and a_c are approximately Gaussian random processes with

zero means and unit variances. Then $A_1 = \frac{\sqrt{(a_s)^2 + (a_c)^2}}{2}$ is Rayleigh distributed. The autocorrelation of A_1 is

given as $J_0(\omega_m \tau)$, where ω_m is the Doppler frequency and $J_0(x) = \frac{2}{\pi} \cdot \int_0^{\pi/2} \cos(x \cdot \cos \phi) d\phi$. In the

simulations, the variance of α is selected to make the average power of the received signal equal to 1 and M_0 is 8, $n \in 1, 2, \dots, M_0$. Figures 9 and 10 are two samples of the random process generated by the two generators discussed above. The curves represented by $\{A_1\}$ in solid line and $\{nr_1\}$ in dashed line are corresponding to the correlated and uncorrelated Rayleigh fading processes with $\sigma^2=1$ respectively. From the figures, it is observed that the correlation of the Rayleigh fading process decreases when the Doppler frequency increases.

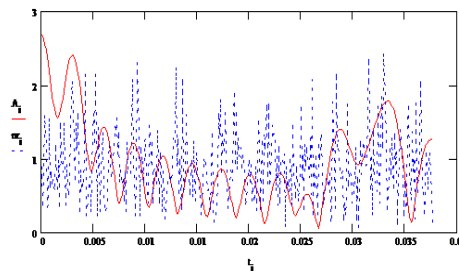


Fig 9: Realizations of correlated and uncorrelated Rayleigh fading processes with Doppler frequency $f_d = 178$ Hz

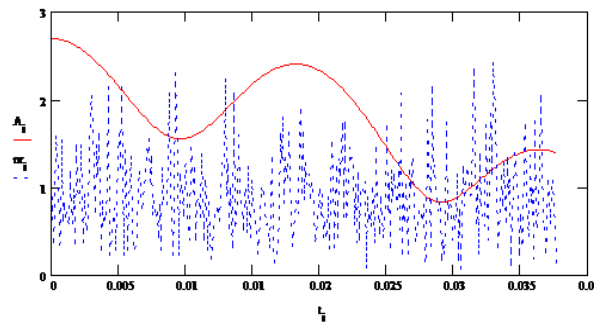


Fig 10; Realizations of correlated and uncorrelated Rayleigh fading processes with Doppler frequency $f_d=29.8$ Hz

In Fig 9, the moving speed v_c is 60 miles per hour (mph), the carrier frequency $f_c=2$ GHz, and the Doppler frequency $f_d=v_c*f_c/c=178$ Hz, where c is the light speed 3×10^8 m/sec. In Figure 10, the moving speed v_c is 10 mph, the carrier frequency $f_c=2$ GHz, and the Doppler frequency $f_d=29.8$ Hz.

B. Simulation results:

This section presents the simulation results of Turbo code performance under AWGN and fading channel environments. The results under the AWGN channel serve as a reference. The simulation uses the Turbo recursive systematic Convolutional encoder with generator matrices (1,15/13,17/13) and a shift register memory 4. The channel gain is generated by using either the Jack fading or Rayleigh fading generator. The frame size either is 384 or 570 bits/frame. It can be seen that the Turbo code’s performance becomes worse under the fading channel than that under the AWGN channel. By observing the simulation results, we have the following conclusions.

(1). Effects of the number of iterations on BER:

Increasing the number of iterations is not much help in the low regions of SNR. In the middle to high regions of SNR, when the number of iterations increases from 1 to 3, the performance of the Turbo decoder improves dramatically. In other words, BER decreases dramatically. This is due to the decoder 1 and decoder 2 share the information and makes more accurate decisions. As the number of iterations increases, the performance of the Turbo decoder improves. However, after the number of iterations reaches a certain value, the improvement is not significant. It can be explained that decoder 1 and 2 already have enough information, further iterations do not give them more information. Figure 11 shows the convergence of the decoding iterations. The correlated Rayleigh fading channel corresponds to

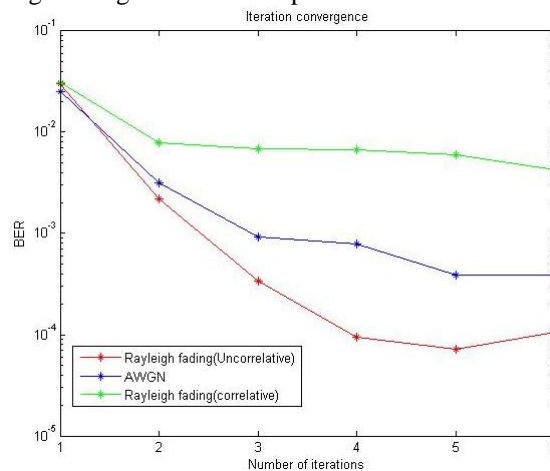


Fig 11: Iteration convergence and BER

the Doppler frequency $f_d=268.2$ Hz, with a SNR_b of 4 dB, and a code rate of 1/3. The uncorrelated Rayleigh fading corresponds to a SNR_b of 4 dB and a code rate of 1/3. The AWGN channel corresponds to a SNR_b of 1.2 dB and code rate of 1/3. The conclusion is that 3 iterations are good enough to get reasonable results in the middle region of SNR_b, and 2 iterations, for high region of SNR_b. High region is defined as SNR_b>1.4 dB, middle range 1.4 dB >SNR_b>0.3 dB in AWGN channel. In uncorrelated Fading channel, high region is defined as SNR_b>4 dB, middle range as 4 dB>SNR_b>2.5 dB. In the other words, further iterations do not gain much improvement of the BER.

In the simulations, a criterion is applied to stop the iterations. When the decoding decision in two consequent iterations is not changed, the program will stop further iteration. The simulation result shows that at SNR>3.2 dB under uncorrelated Rayleigh fading channel, the decoding errors can be reduced greatly after two to three iterations. Therefore, when five iterations are used in this region, more than half of the time the decoder is dealing errorless codes, which is a waste of time. By dynamically applying the criterion to stop the iterations greatly saves the decoding time in region of high SNR, hence, improve the decoding efficiency. The smaller the frame size, the more time it saves. The simulation results are shown in from which obviously it can be seen that some penalty is added if the iteration stops earlier. The simulation is given at the data frame size=378 bits, code rate=1/3, iteration=6. For correlated Rayleigh fading channel, Doppler frequency $f_d=178\text{Hz}$ was used. The running time totally depends on the programming language and the computer used..

(2). Larger frame size gets better performance:

The larger the frame size, the bigger the S-window. Therefore, it will produce larger distance by using an interleaver. The correlation between the two adjacent bits will become smaller. Hence the decoder gives better performance. The simulation results verified this conclusion. However, since Turbo code is a block code, it causes a time delay before getting the complete decoding output. Increasing the frame size also increases the delay time. Figure 12 shows the BERs of Turbo code under uncorrelated Rayleigh fading channel with the code rate=1/3, iteration=3, frame size $L=384$ bits and $L=576$ bits. From the figure we can see that the Turbo code with lager frame size has better performance.

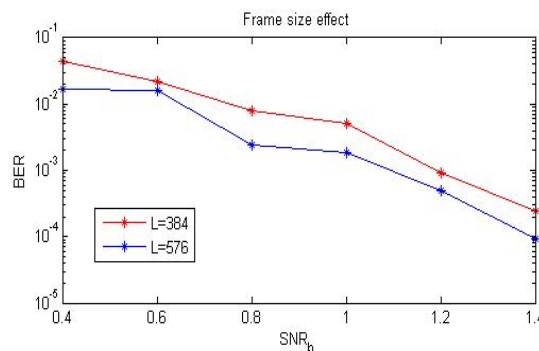


Fig 12: Effects of frame size on BER

(3). Effects of Puncturing on BER:

When the code rate is decreased, more bits have to be punctured. The bandwidth requirement is also decreased. However, some information is lost. This means that the performance of the Turbo code will also degrade in general. Fig 13 shows the effects of the punctuation on BER. The higher ,the code rate, the lower the BER. In the simulation, decode iteration=3, frame size=384, uncorrelated Rayleigh fading environment applied. The three curves are corresponding to code rate=1/2, 1/3, and 1/4, respectively.

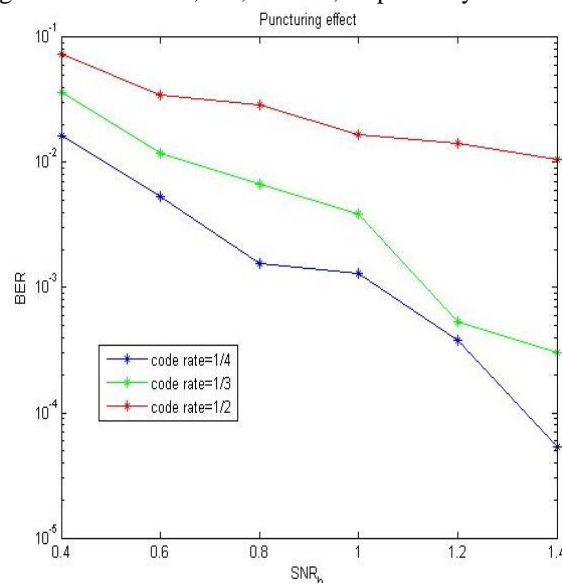


Fig 13: Effects of puncturing on BER

(4). Effects of Doppler frequency on BER:

Fig 14 shows the effects of Doppler frequency on BER of the Turbo code.

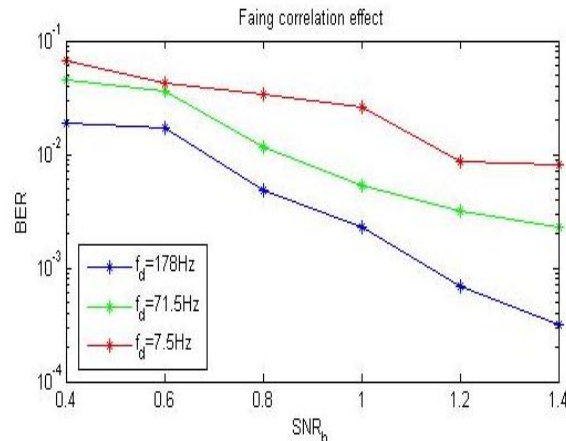


Fig 14 Effects of Doppler frequency on BER

The Jake fading generator is used in the simulations. The three curves are corresponding to $v_c=60$ mph ($f_d=178$ Hz), 24 mph ($f_d=71.5$ Hz), and 2.5 mph ($f_d=7.5$ Hz), respectively, when the carrier frequency is $f_c=2$ GHz. The code rate $1/4$, frame size 384 bits, iteration 3 are used for the three cases in the simulation. The higher the Doppler frequency, the less the correlation of the fading process, hence better the performance.

V. Conclusions And Future Work

In this paper, Turbo code was studied in details. In a Turbo code system model was introduced, then the Turbo encoder which includes the recursive systematic Convolutional encoder, interleaver, and puncturing were discussed. In Turbo code decoding algorithm under AWGN channel was studied. A modified decoding algorithm under fading channel was also presented.

In two Rayleigh fading generators were discussed and used in the simulations. Performance of the Turbo code under AWGN channel and Rayleigh fading channel environments were evaluated by simulations.

The simulation results show that Turbo code is a powerful error correcting coding technique under SNR environments. It has achieved near Shannon capacity. However, there are many factors need to be considered in the Turbo code design. First, a trade-off between the BER and the number of iterations need to be made, e.g., more iteration will get lower BER, but the decoding delay is also longer. Secondly, the effect of the frame size on the BER also needs to be considered. Although the Turbo code with larger frame size has better performance, the output delay is also longer. Thirdly, the code rate is another factor that needs to be considered. The higher coding rate needs more bandwidth.

From the simulation results, it is observed that the behavior of the Turbo decoder is quite different under different channel environments. The performance of the Turbo code is much worse under correlated Rayleigh fading channel than that of AWGN or uncorrelated Rayleigh fading channels. Another drawback of the Turbo code is its complexity and also the decoding time.

For the further work, if another interleaver is applied to the coded bits before the transmission over a fading channel, the performance of the Turbo code may be improved. How to reduce the decoding delay time is another interesting topic. Dynamically selecting the number of iteration is helpful to reduce the delay time. Another big challenge work that need to be done is theoretically to explain and analyze why the iterative decoding algorithm performs so well as it does.

Of course there are a lot of details not explained here, a lot of variation to the turbo coding scheme and a lot of things that may need a proof. Some of these can be found in the papers on the references list.

References:

- [1] G. Berrou, A.Glavieuc, and P.Thitmajshima, "Near Shannon limit error-correcting coding: Turbo codes," in Proc. 1993, Int. Conf. Com., Geneva, Switzerland, May 1993
- [2] L.R.Bahl, J.Cocke, F.Jelinek, and J.Racic, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans, Information Theory, Vol. IT-20, pp.284-287, 1974.
- [3] S.Benedetto, D.Divsalar, G.Montorsi, and F.Pollara "A soft input soft-output Maximum A Posteriori (MAP) module to decode parallel and serial concatenated codes" TDA progress report 42-127, November 15, 1996
- [4] Robert J. McEliece, Fellow, IEEE, David J. C. MacKay, and Jung-Fu Cheng Turbo Decoding as an Instance of Pearl's "Belief Propagation" Algorithm IEEE Journal on selected areas in communication, Vol. 16, No. 2, February 1998
- [5] Stephan ten Brink, Member, IEEE "Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes" IEEE Transaction on Communications, Vol. 49, No. 10, October 2001
- [6] Claude. Berrou, member IEEE and Alian Glavieux "near optimum error correcting coding and decoding: Turbo-Co