PERFORMANCE OF TURBO CODED SIGNALS OVER FADE CHANNELS

TURBO KODLANMİŞ YPARETLERİN SÖNÜMLEMELÝ KANALLARDA BAþARIM ANALÝZY

Oguz BAYAT1 Hissam ALNAJJAR2 Osman N. UCAN3 Onur OSMAN4

1,2 University of Hartford, West Hartford, CT, USA
3 Istanbul University, Engineering Faculty, Electrical & Electronics Department, 34850, Avcilar, Istanbul- Turkey
4 Istanbul Commerce University, Vocational School, Ragip Gumuspala Cad. No:84 34378 Eminonu, Istanbul-Turkey

1 e-mail: oguzbayat@hotmail.com 3 e-mail: uosman@istanbul.edu.tr 4 e-mail: onurosman@turk.net

ABSTRACT

In this paper, the performance of turbo coded signals are investigated for different frame numbers and encoder structures over Rician fading channels. The numerical results demonstrate the error performance degradation due to fading channel frame numbers and the state of the encoder.

Key Words: Turbo Coding, Rician Fading Channel

ÖZET

Bu makale de, farklý çerceve uzunluklarý, kodlayýcýý özelliklerine gore deðiþtirilerek turbo kodlanmýþ Ýparetlerin Rician ortamlarda baþarýmý incelemiþtir. Benzetim sonuçlarý, söz konusu Rician ortamda farklý parametreler için deðerlendirilmiþtir.

Anahtar Kelimeler: Turbo kodlama, Rician Sonümleme Kanalý

I. INTRODUCTION

Turbo codes are a new class of error correction codes that were introduced a long with a practical decoding algorithm in [1]. The importance of turbo codes is that they enable reliable communications with power efficiencies close to the theoretical limit predicted by Claude Shannon [2]. Since their introduction, turbo codes have been proposed for low-power applications such as deep-space and satellite communications, as well as for interference limited applications such as third generation cellular and personal communication services.

The main principle in the turbo coding scheme is to use two codes in parallel. This means that the information sequence is encoded twice, the second time after a scrambling of the information bits. The component codes are chosen as small convolutional codes in the recursive systematic form. With this encoding we are able to decode
the two encoded streams with an iterative process using two soft-in soft-out decoders, one corresponding to each of the encoders. For the simulations shown in this paper we have used a MAP decoder. The decoders exchange information as a priori probabilities for the information bits.

II. SYSTEM MODEL
The block diagram of the channel model that we use in this paper is shown in Fig. 1. The channel output \( u_k \) is

\[
u_k = a_k x_k + n_k \quad (1)
\]

\( n_k \) is Gaussian Noise where the noise variance \( \sigma^2 = N_0/2E_s \), \( a_k \) is fading amplitude for the signaling interval \( k \).

![Figure 1. Channel Model](image)

In the encoder structures we use Recursive Systematic Convolutional (RSC) encoders as Turbo encoders. Consider a half-rate RSC encoder with \( M \) memory size. If the \( d_k \) is an input at time \( k \) the output \( X_k \) is equal to

\[
X_k = d_k \quad (2)
\]

Remainder \( r(D) \) can be found using feedback polynomial \( g^{(0)}(D) \) and feedforward polynomial is \( g^{(1)}(D) \). The feedback variable is

\[
r = d_k + \sum_{j=1}^{K} a_j s^{(0)}_j \quad (3)
\]

and RSC encoder output \( Y_k \) which called parity data \([7]\) is

\[
Y_k = \sum_{j=0}^{K} r_{k-j}s^{(1)}_j \quad (4)
\]

III. TURBO DECODING
The problem of estimating the state sequence of a Markov process observed through noise has two well known trellis-based solutions- Viterbi Algorithm (VA) \([8]\) and the symbol-by-symbol maximum a posteriori (MAP) algorithm. The key difference between algorithms is that the states estimated by the VA must form a connected path through the trellis, while the states estimated by the MAP algorithm need not to be connected. When applied to digital transmission systems, the VA minimizes the frame error rate (FER), while the MAP algorithm minimizes the bit error rate (BER).

![Figure 2. Turbo Encoder for 1 ± D^3 / PRFC Model](image)

![Figure 3. Recursive Systematic Convolutional (RSC) Encoder](image)

![Figure 4. Turbo Decoder](image)
is possible to model a turbo code as a single Markov process, such a representation is extremely complex and does not lend itself to computationally tractable decoding algorithms. Turbo decoding proceeds instead by first independently estimating the individual Markov processes. Because the two Markov processes are defined by the same set of data, the estimates can be refined by sharing information between the two decoders in an iterative fashion. More specifically, the output of one decoder can be used as a priori information by the other decoder (Fig. 4). If the outputs of the individual decoders are in the form of hard-bit decisions, then there is little advantage to sharing information. However, if soft-bit decisions are produced by the individual decoders, considerable performance gains can be achieved by executing multiple iterations of decoding [6].

A. Soft-Input, Soft Output (SISO) Decoding

In the MAP decoder, the output is given by

\[ \Lambda_k = \ln \frac{P[m_k = 1 | y]}{P[m_k = 0 | y]} \tag{6} \]

\( m \) is message bit and \( y \) is received sequence. There are three inputs for Soft-Input Soft-Output (SISO) decoder. \( y_k^{(s)} \) is the systematic observation, \( y_k^{(o)} \) is the parity information and \( s_k \) is the priori information which is derived from the other decoder's output. The log-likelihood at the output of a SISO decoder using the channel model can be factored into three terms as [6]

\[ \Lambda_k = \frac{4a_k^{(s)} \cos(\Theta_k^{(s)})E_s}{N_0} y_k^{(s)} + z_k + l_k \tag{7} \]

where the term \( l_k \) is called the extrinsic information, \( a_k \) is fading amplitude, \( E_s \) is the energy per code symbol and \( N_0 \) is the noise power. The priori information at the input of one decoder is found by subtracting two values from its output to prevent “positive feedback problem” as shown in Fig. 4 of M.C. Valenti study [6]. First of all we make the following notation;

\[ L_k = \frac{4a_k^{(s)} \cos(\Theta_k^{(s)})E_s}{N_0} \tag{8} \]

which is called the reliability of the channel, and

\[ l_k = \Lambda_k - L_k^{(s)} + L_k^{(o)} - z_k \tag{9} \]

The MAP algorithm attempts to find the most likely individual state \( s_k \) given \( y \)

\[ \hat{s}_k = \arg \max_{s_k} P[s_k | y] \tag{10} \]

Before finding the a posteriori probabilities (APPs) for the message bits, the MAP algorithm finds the probability of \( P[s_k \rightarrow s_{k+1} | y] \) of each valid state transition given the noisy channel observation \( y \).

\[ P[s_k \rightarrow s_{k+1} | y] = \frac{P[s_k \rightarrow s_{k+1} | y]}{P[y]} \tag{11} \]

B. Log-MAP Algorithm

The maximum a posteriori (MAP) algorithm can calculate the a posteriori probability of each bit with a perfect performance. However there are known problems as the number of calculation depending on the memory states. These problems can be solved by performing the entire algorithm in the log domain [9][10]. To illustrate how performed in the log domain, consider the Jacobian Logarithm.

\[ \ln(e^{x}+e^{y}) = \ln(1 + \exp[|y-x|]) \tag{12} \]

This equation describes the log-MAP algorithm with a correction function \( f \).

Let \( \gamma(s_k \rightarrow s_{k+1}) \) denoted the natural logarithm of

\[ \gamma(s_k \rightarrow s_{k+1}) = \ln \gamma(s_k \rightarrow s_{k+1}) \tag{13} \]

\[ = \ln P[m_k] + \ln P[y_k | x_k] \tag{14} \]

\[ \ln P[m_k] = z_k m_k - \ln(1 + e^{c_k}) \tag{15} \]

and (14) becomes

\[ \gamma(s_k \rightarrow s_{k+1}) = \ln P[m_k] - \frac{1}{2} \ln(\pi N_0 / E_s) \]

\[ - E_s N_0 \sum_{q=0}^{N-1} y_k^{(q)} (-a_k^{(s)} x_k^{(q)} \cos(\Theta_k^{(s)}) + 1) \]

\[ = \lambda(s_k \rightarrow s_{k+1}) \tag{16} \]

Now let \( \alpha(s_k) \) be the natural logarithm of \( \alpha(s_k) \), \( \alpha(s_{k+1}) = \ln \alpha(s_k) \cdot \gamma(s_k \rightarrow s_{k+1}) \tag{17} \)

\[ = \ln \left\{ \sum_{s_k \in \Lambda} \exp \left[ \alpha(s_{k+1}) + \gamma(s_{k+1} \rightarrow s_k) \right] \right\} \tag{18} \]

\[ = \max_{s_k \in \Lambda} \left\{ \alpha(s_{k+1}) + \gamma(s_{k+1} \rightarrow s_k) \right\} \tag{19} \]

where \( \Lambda \) is the set of states \( s_k \) that are connected to the state \( s_{k+1} \). Now let \( \beta(s_{k+1}) \) be the natural logarithm of \( \beta(s_{k+1}) \),

\[ \beta(s_{k+1}) = \ln \beta(s_{k+1}) \tag{20} \]

\[ = \ln \left\{ \sum_{s_k \in \Lambda} \exp \left[ \beta(s_k) + \gamma(s_k \rightarrow s_{k+1}) \right] \right\} \tag{21} \]

\[ = \max_{s_k \in \Lambda} \left\{ \beta(s_k) + \gamma(s_k \rightarrow s_{k+1}) \right\} \tag{22} \]

where \( B \) is the set of states \( s_{k+1} \) that are connected to state \( s_k \), and we can calculate the Log Likelihood Ratio (LLR) by using

**References**

Performance Of Turbo Coded Signals Over Fading Channels

Oguz BAYAT, Hissam ALNAJJAR, Osman N. UCAN, Onur OSMAN
\[ \Lambda_k = \ln \left( \sum_{s_i} \exp \left[ \sum_{s_{i+1}} \left( \gamma(s_i \rightarrow s_{i+1}) + \beta(s_{i+1}) \right) \right] \right) \]  

where \( S = \{ s_i \rightarrow s_{i+1} : m_i = 1 \} \) is the set of all state transitions associated with a message bit of 1, and \( S = \{ s_i \rightarrow s_{i+1} : m_i = 0 \} \) is the set of all state transitions associated with a message bit of 0.

At the last iteration we make the hard decision by using the second decoder output \( \Lambda^{(2)} \),

\[ \hat{m}_i = \begin{cases} 1 & \text{if } \Lambda^{(2)} \geq \Lambda = 0 \\ 0 & \text{if } \Lambda^{(2)} < 0 \end{cases} \]  

IV. PERFORMANCE OF TURBO CODED SIGNALS

In this section, the performance of turbo coded signals are evaluated for different frame numbers and memory constellations over Rician fading channels. (Figures 5-14).

In our example, 1/2 and 1/3 rate turbo encoders with different memory configurations are investigated. Here we use the generator matrix \( g = [111:101] \), \( g = [1101:1111] \) and \( g = [1111:10001] \), a random interleaver is used and the frame size \( N = 500 \). It is clear that for a constant iteration number, as \( K \) increases, performance improves for the same SNR values. And also the frame number is the important effect for improving the performance as shown in Figure 7 and 8.

![Figure 5](image5.png)  
**Figure 5.** Bit error performance for \( K = 10 \), \( g = [111:101] \) and \( N = 500 \)

![Figure 6](image6.png)  
**Figure 6.** Bit error performance for \( K = 10 \), \( g = [111:101] \) and \( N = 192 \)

![Figure 7](image7.png)  
**Figure 7.** Bit error performance for \( r = 1/3 \) \( Eb/N0 = 4 \text{dB} \) \( g = [111:101] \) and \( N = 500 \) over \( K = 0 \) dB

![Figure 8](image8.png)  
**Figure 8.** Bit error performance for \( r = 1/3 \) \( Eb/N0 = 1.5 \text{dB} \) \( g = [111:101] \) and \( N = 500 \) over AWGN channel
Performance Of Turbo Coded Signals Over Fading Channels

Figure 9. Bit error performance for $r=1/2$, $g=[1101;1111]$ and $N=500$ over AWGN channel

Figure 10. Bit error performance for $r=1/2$, $g=[1101;1111]$, $N=500$ and $K=10$dB

Figure 11. Bit error performance for $r=1/2$, $g=[1101;1111]$, $N=500$ and $K=0$dB

Figure 12. Bit error performance for $r=1/2$, $g=[1111;10001]$ and $N=500$ over AWGN channel

Figure 13. Bit error performance for $r=1/2$, $g=[1111;10001]$, $N=500$ and $K=10$Db

Figure 14. Bit error performance for $r=1/2$, $g=[1111;10001]$, $N=500$ and $K=0$dB
V. CONCLUSION

In this paper we show how encoder structures, encoder rate and frame number can be altered the performance of turbo coded signals over Rician channel while Rician channel is under consideration. As an example, the performance of turbo coded signals are simulated over Rician channel with different fading parameter, iteration number and data block size N.

REFERENCES


Oguz Bayat: He has graduated his Bachelor of Science in Electrical Engineering from Istanbul Technical University, Istanbul, Turkey. He is currently pursuing his Master of Science education at University of Hartford, West Hartford, CT, USA. His GPA is 3.8/4. His research areas are: Mechatronics, Power electronics, statistics, multivariable control systems, Turbo coding and wireless communications. He is a member of Lions Club, Bakirkoy, Istanbul.

Osman N. Uçan: See Vol.2, Number 1, page 343.

Onur Osman: See Vol.2, Number 1, page 375.