Sequential Decoding of Convolutional Codes with Universal Metric over Bursty-Noise Channel

버스트잡음 채널에서 Universal Metric을 이용한 컨벌루션 부호의 축차복호

Byung Hyun Moon · Chae Wook Lee
(School of Computer and Communication, Taegu University)

문 병 현 · 이 채 욱
(대구대학교 정보통신공학부)

Abstract

In this paper, a new metric, universal metric, is proposed for sequential decoding of convolutional decoding. The complexity of Fano metric for Fano's sequential decoding algorithm is compared with that of the proposed universal metric. Since the Fano metric assumes that it has previous knowledge of channel transition probability, the complexity of Fano metric increases as the assumed channel error probability does not coincide with the true channel error probability. However, the universal metric dose not require the previous knowledge of the channel transition probability since it is estimated on a branch by branch basis. It is shown that the complexity of universal metric is much less than that of the Fano metric for bursty noisy channel.

I. Introduction

Massey [1] proved that the heuristic Fano metric is actually a logical choice for maximum likelihood decoding for convolutional codes. However, the Fano metric requires true channel transition probability to calculate the metric. On the other hand, the proposed universal metric does not require previous knowledge of the channel transition probability.

A sequential decoder is desirable for long constraint length codes and channel memories since the Viterbi algorithm decoder's complexity grows exponentially with the equivalent constraint length. To apply the sequential algorithm, a branch metric is computed for each branch and the branch with the largest metric is added to the path at the node. Different versions of the sequential algorithm, such as the stack algorithm or Fano algorithm can be applied to the decoding tree, as long as the metric is defined. In this paper, Fano algorithm is used to compare the decoding complexity of the Fano metric and the universal metric over the bursty error channel.

II. Preliminaries

A. Bursty-noise Channel

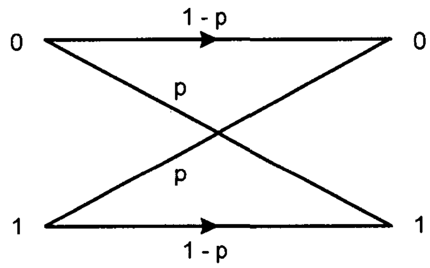

Fig. 1. Binary symmtric channel

A discrete communication channel shown in Fig. 1 is called a binary symmetric channel (BSC). If the channel input is 0, the channel output is 0 with probability (1-p) and is 1 with probability p. If the channel input is 1, the channel output is 1 with probability (1-p) and is 0 with probability p. In this paper, BSC is assumed for all the applications. However, the bit errors on real channels occur in bursts which are not well modeled by a memoryless statistical model. Error bursts can occur in military environments [2], satellite systems [3], concatenated coding schemes [4,5], and from optical storage media such as compact discs [6].

Channels exhibiting error bursts have been modeled by Markov processes, by impulse noise, and by bursty-noise. For simplicity, the bursty-noise in this paper is defined to be the background Gaussian noise plus burst noise, where burst noise is defined to be a series of finite-duration Gaussian-noise pulses with variable duration and arbitray occurance time.

B. Convolutional Encoder

For the use of BSC, convolutional codes can be generated by a convolutional encoder as shown in Fig. 2. The encoder consists of an N bit shift resisters and n modulo-2 adders. The connection between a shift register and a modulo-2 adder is specified by a set of coefficients C_{ij} where i=1,2, . . . N and j=1,2, . . . n. If the ith stage of a shift register is connected to a jth modulo 2 adder,

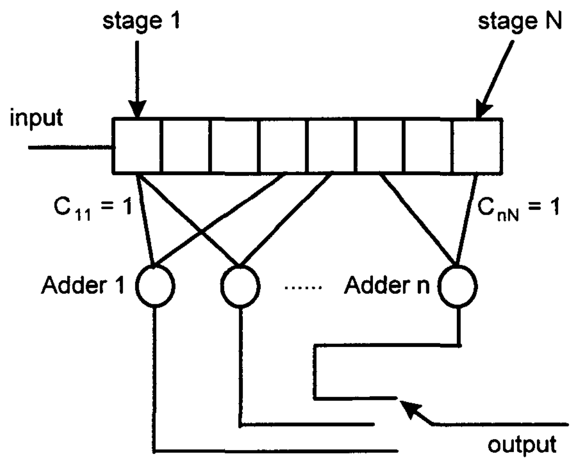

Fig. 2. Binary convolution encoder

then $C_{ij}=1$. When there is no connection between ith stage of a shift register and jth modulo 2 adder, $C_{ij}=0$.

The operation of a convolutional encoder is as follows. Assume that the output $\vec{x}_i=(x_1^i, x_2^i, \cdots, x_n^i)$ is to be encoded. First, the contents of all N stages of the shift register are set to zero. Then, the first digit, x_1^i, of $\vec{x}_i$ is shifted into the first stage of the shift registers. The n modulo-2 adders are sampled and passed to the input of the BSC for transmission. This procedure continues until the last *component of* $\vec{x}_i$ *shifts into the first stage of the shift registers.*

In a shift register encoder, an (n,k) convolutional code is equivalent to k information bits entering the encoder genarating n encoded bits. The number N is defined as constraint length of the encoder. Also, the rate of a binary (n,k) convolutional code, R, is defined as

$$R=\frac{k}{n}. \quad (1)$$

C. Tree Structure

This section considers how a convolutional encoder constructs an output sequence $\vec{y}$ from the input $\vec{x}_i=(x_1^i, x_2^i, \cdots, x_n^i)$. The first n digits of $\vec{y}$ are obtained by shifting the first bit x_1^i into the shift register and sampling n modulo-2 adders. An input tree can be obtained by adapting the convention that it corresponds to an upward branch if "0" shifts into the shift register and a downward branch if "1" shifts into the shift register. As an example, the set of 16 input sequences with 4 digits long is diagrammed in Fig. 3.

A code tree can be obtained by writing along each branch of the input tree the n digits of $\vec{y}$ associated with input sequence. Consider a particular conventional encoder with N=4, n=3 and k=1 as shown in Fig. 4. Then, the corresponding code tree can be constructed as shown in Fig. 5. If k=2, 4 branches will grow out of a node. In general, 2^k branches will grow out of a node if k information bits shift into a convolutional encoder. [7]

A code tree can be obtained by writing along each branch of the input tree the n digits of $\vec{y}$ associated with input sequence. Consider a particular conventional encoder with N=4, n=3 and k=1 as shown in Fig. 4. Then, the corresponding code tree can be constructed as

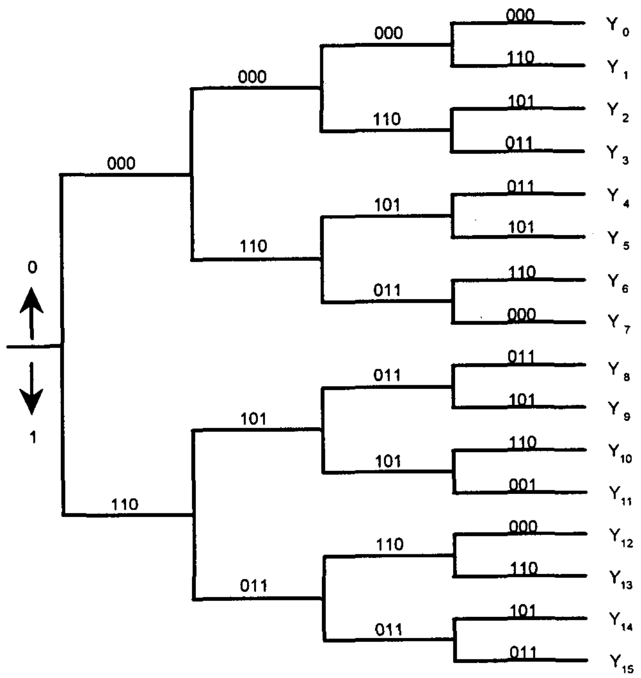

Fig. 3. A diagram of 16 input sequences

shown in Fig. 5. If k=2, 4 branches will grow out of a node. In general, 2^k branches will grow out of a node if k information bits shift into a convolutional encoder. [7]

Define an encoder matrix to be G. Then, the encoder matrix corresponds to Fig. 4 can be represented by using the notation C_{ij} introduced previously. An encoder matrix for the encorder shown in Fig. 4 is given by

$$G=\begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{32} & C_{34} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

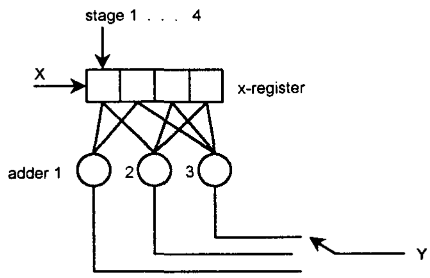

Fig. 4. A convolutional encoder with
N=4, n=3 and k=1

III. The Universal and Fano Metric

Let Y and Z be two finite sets where $Y=\{0,1\}$ and $Z=\{0,1\}$. Let $\vec{y}=(y_1, y_2, \cdots, y_n)$ and $\vec{z}=(z_1, z_2, \cdots, z_n)$ be arbitrary n sequences over Y and Z, respectively. Then, the universal metric between $\vec{y}$ and $\vec{z}$ is defined by $\vec{y}=(y_1, y_2, \cdots, y_n)$ and $\vec{z}=(z_1, z_2, \cdots, z_n)$ be arbitrary n sequences over Y and Z, respectively. Then, the universal metric between $\vec{y}$ and $\vec{z}$ is defined by

$$\Gamma(\vec{y};\vec{z}) = I(\vec{y};\vec{z}) - B \tag{2}$$

where B is bias term and $I(\vec{y};\vec{z})$ is mutual information between $\vec{y}$ and $\vec{z}$. The mutual information is given by

$$\sum_{i=0}^{1}\sum_{j=0}^{1} p_{\vec{y}\vec{z}}(i,j)\cdot\log\left(\frac{p_{\vec{y}\vec{z}}(i,j)}{p_{\vec{y}}(i)\cdot p_{\vec{z}}(j)}\right) \tag{3}$$

For each $s\in Y$ and $t\in Z$, the probability distribution $p_{\vec{y}}(s)$ is defined as the frcction of coordinates i such that $y_i=s$. Similiarly, $p_{\vec{y}}(t)$ and $p_{\vec{y}\vec{z}}(s,t)$ are defined as the fraction of coordinates i such that $z_i=t$ and $y_i=s$ and $z_i=t$ respectively.

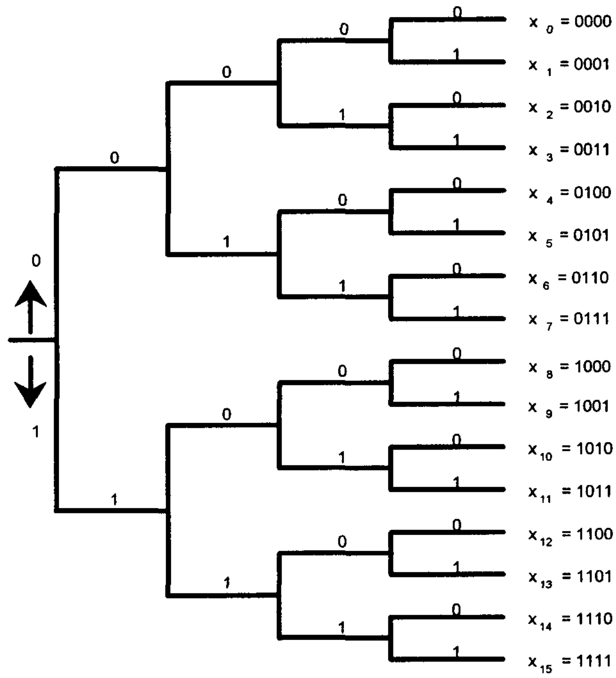

Fig. 5. Code tree for encoder of Fig. 4.

The Fano metric is defined as

$$\Gamma_F = \sum_{i=1}^{n} \left[\log\left(\frac{p(z_i|y_i)}{p(z_i)} \right) - R \right] \tag{4}$$

where $p(z_i|y_i)$ is the channel error probability and R is the rate of the convolutional code. [8]

Ⅳ. The Fano Algorithm

The Fano algorithm works as follows. At every stage, the decoder is located at some node in the code tree. From this node, the decoder looks deeper into the tree. If it finds a node with a metric greater than the threshold, T, then it moves to that node. If not, it will move backword and try to move forward along another branch. The detailes of the Fano algorithm are given in Fig. 6. [9]

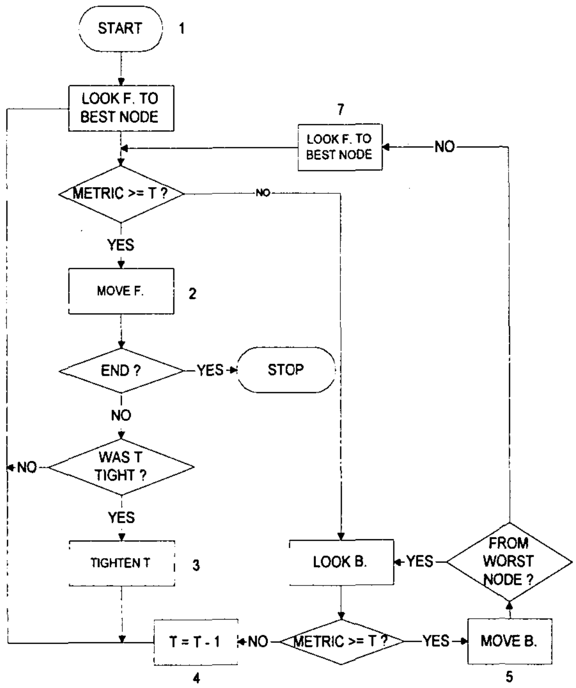

Fig. 6. A flowchart description of the Fano algorithm.

V. Example of the Fano Algorithm Using the Universal Metric

In this section, the Fano algorithm using universal metric is illustrated as an example. Consider a (8,1) convolutional encoder with encoder matrix given as below

$$G=\begin{bmatrix} 0&0&1&0&1&0&0&1&0\\ 1&0&1&1&0&1&0&1&1\\ 0&1&0&1&1&1&0&1&1\\ 0&0&1&0&0&0&1&0&0\\ 0&1&1&0&1&0&1&1&0\\ 1&0&0&1&0&0&0&1&1\\ 1&0&0&1&1&0&1&1&0\\ 1&1&1&0&0&0&1&0&1 \end{bmatrix}.$$

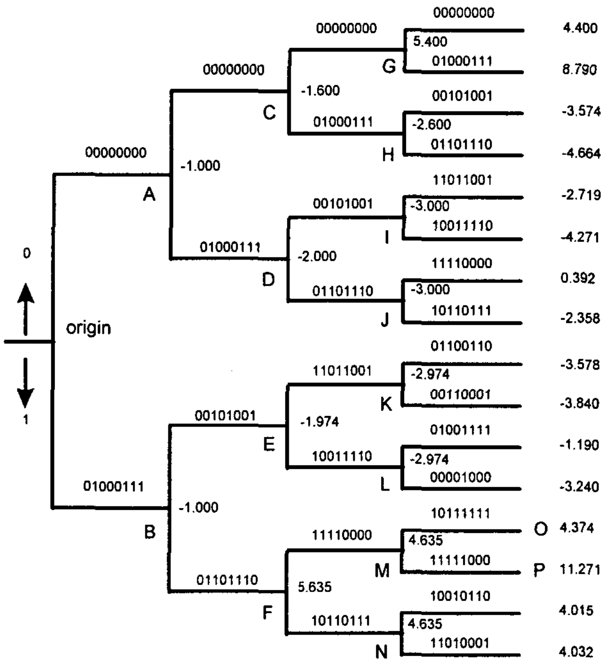

Fig. 7. Code tree with corresponding metric values for the example

Table 1. Summary of node visited and the value of threshold.

Step	Node	Threshold	Location
1	origin	0.0	1
2	origin	-1.0	4
3	A	-1.0	2
4	origin	-1.0	5
5	origin	-1.0	6
6	B	-1.0	2
7	F	-1.0	2
8	F	-1.0	3
9	F	5.0	4
10	M	4.0	2
11	P	4.0	2
STOP			

Assume the input sequence be $\vec{x}=(1, 1, 0, 1)$. Let the output sequence $\vec{y}$ = $(\vec{y_1}, \vec{y_2}, \vec{y_3}, \vec{y_4})$ .Then the first 8 digits of $\vec{y}$ is given as $\vec{y_1}$ = (0,1,0,0,0,1,1,1) corresponding to the first input. Similiarly, the rest of the output sequence is given by $\vec{y_2}$ = (0,1,1,0,1,1,1,0), $\vec{y_3}$ = (1,1,1,1,0,0,0,0) and $\vec{y_4}$ = (1,1,1,1,1,0,0,0). Assume the channel noise sequence be $\vec{n}$ = $(, \vec{n_2}\ \vec{n_1}, \vec{n_3}, \vec{n_4})$ where $\vec{n_1}$=(1,1,1,1,0,0,0,0), $\vec{n_2}$ =(0,0,0,0,0,0,0,0), $\vec{n_3}$=(1,1,1,1,0,0,0,0) and $\vec{n_4}$=(0,0,0,0,0,0,0,0). Then, the channel output sequence $\vec{z}=(\vec{z_1}, \vec{z_2}, \vec{z_3}, \vec{z_4})$ is given as $\vec{z_1}$=(1,0,1,1,0,1,1,1), $\vec{z_2}$=(0,1,1,0,1,1,1,0), $\vec{z_3}$=(0,0,0,0,0,0,0,0), and $\vec{z_4}$=(1,1,1,1,1,0,0,0). Now, the code tree with corresponding metric value calculated from universal metric using Eq. (3) is drawn in Fig. 7. The bias term, B, is set to the rate of the code which is 1/8. Table 1 summarizes the behavior of Fano algorithm by listing each of the changes in either the node visited or the threshold together with the location in the flow chart in Fig. 6 where the changes occur.

Let's define the complexity of the algorithm as the number of the nodes visited until the decoder reaches the last level of the code tree. Then, the complexity of the example above is given by 11.

VI. Simulation Results

In order to compare the complexity of the universal and Fano metric over binary symmetric channel, Monte Carlo simulations are performed. For a given channel error probability, comparisons are made for the complexity of the algorithm that will determine the efficiency of universal metric and the Fano metric. All the simulation results are basd on a fixed code rate of 0.1, a fixed constraint length of 20 with variable k. And, the length of the input sequence is fixed at 100.

The summary of the simulation results is shown in Figure 1 and 2. Figure 1 represents the complexity of universal and Fano metric for k=2 with variable channel error probbility. The complexity of the universal metric is at least twice as

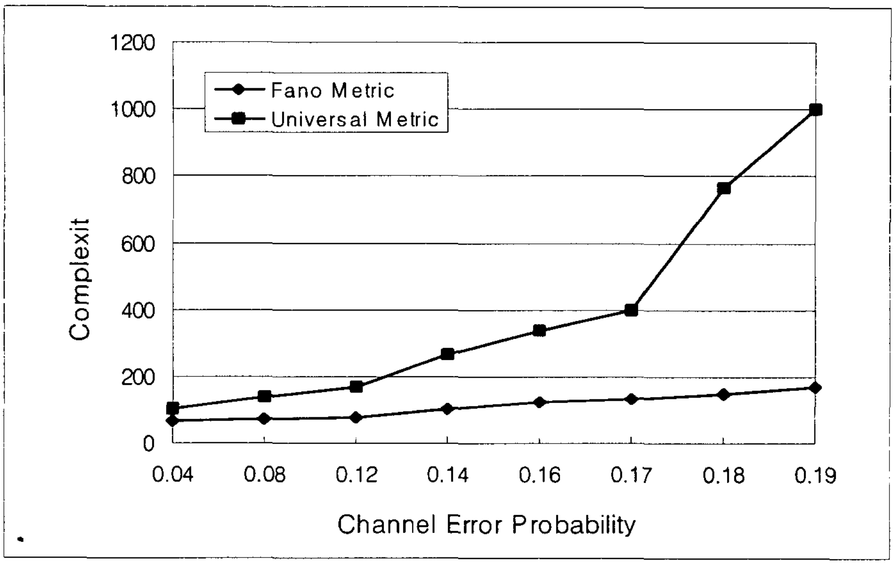

Figure 1. Complexity of Fano and Universal Metric for k=2.

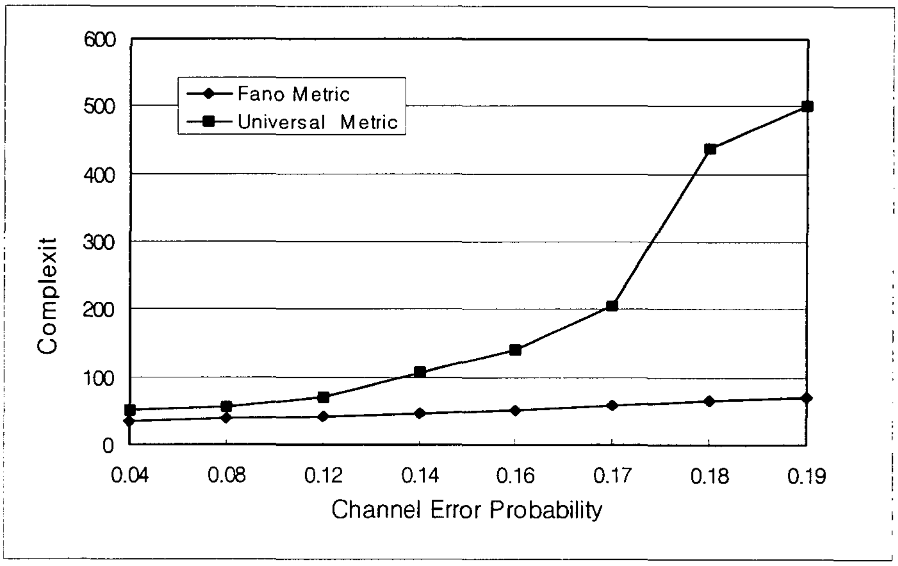

Figure 2. Complexity of Fano and Universal Metric for k=3.

large as the Fano metric. In fact, the complexity of the universal metric is almost 5 times than that of the Fano metric for the channel transition probability of 0.18. Figure 2 shows the complexity of universal and Fano metric for k=3 with variable channel error probability. The complexity of universal and Fano metric is similiar to the case with k=2. As the channel error probability increases, the complexity of universal metric increases drastically. On the other hand, the complexity of the Fano metric increases slowly.

In order to measure the performance of the universal metric and compare with Fano metric over the bursty-noisy channel, the following 2 cases of bursty-noise are are considered in the simulations. For case 1, the channel goes into burst error mode with probability 0.1 and the burst noise lasts for 3 branches of code tree with variable channel error probability. For case 2, the channel goes into burst error mode with probability 0.1 and the burst noise lasts for 4 branches of code tree with variable channel erorr probability. The background chanel error probability is deliverately chosen to be low in order to observe the effect of bursty-noise to the metric.

The simulations are based on an input sequence of length 100 with k=2 and encoder matrix of 20 by 20. For each case, ten cases of different channel error probability due to gackground Gaussian noise were tested to compare the complexity of the universal metric with Fano metric over bursty error channel

As shown in Figure 3, the complexity of the universal metric over the first case of the bursty-noise is much less than that of Fano metric. As a matter of fact, the complexity of the universal metric is less than half of the Fano's with the background channel error probability of 0.076. The similiar results are obtained for the second bursty-noise as shown in Figure 4. As shown in Figure 3 and 4, the complexity of universal metric for the second bursty-noise is slightly higher than the first bursty-noise.

VII. Conclusion

Universal metric that does not require exact channel error probability is proposed

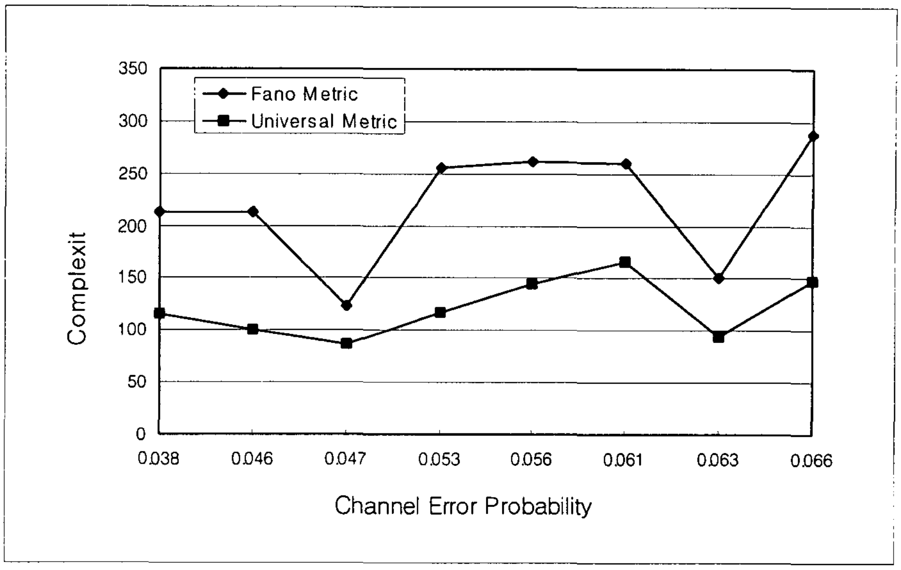

Figure 3. Complexity of Fano and Universal Metric for the 1st Bursty-noise

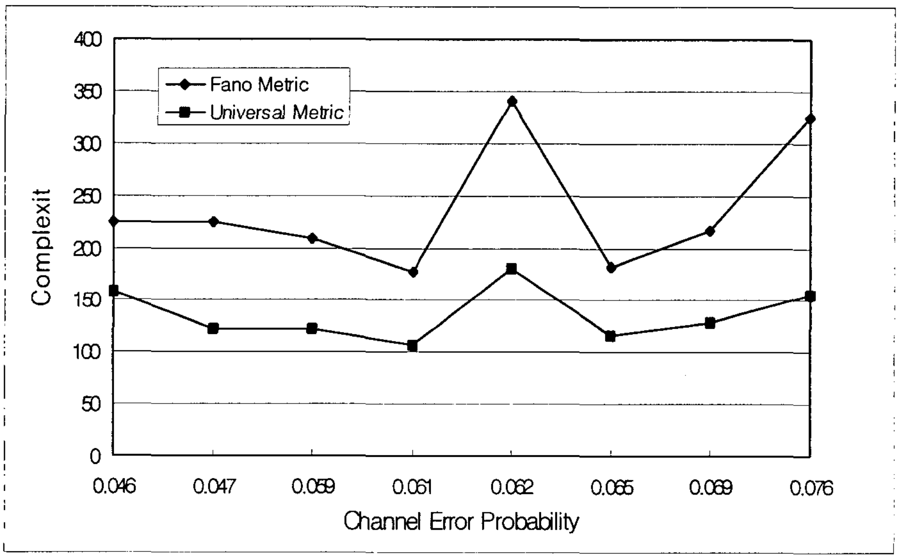

Figure 4. Complexity of Fano and Universal Metric for the 2nd Bursty-noise.

for sequential decoding. It is shown that the complexity of the universal metric is much larger than the maximum likelihood metric, Fano metric, for binary symmetric channel.

Since the channel error probability of the bursty-noise channel is different from the exact channel error probability, the complexity of the universal metric is less than the Fano metric over the bursty-noise channel. For the bursty-noise channel, the effetiveness of universal metric for the sequential decoding of convolutional codes is verified.

Reference

[1] J. L. Massey, "Variable-length codes and the Fano metric," *IEEE Trans. on Inform. Theory*, Vol IT-18, No. 1, pp. 196-198, Jan. 1972.

[2] R. E. Ziemer, and R. L. Peterson, *Digital Communications and Spread Spectrum Systems*. Macmillan, New York, 1985.

[3] T. M. McKenzie, H. Choi, and W. R. Braun, Documentation of CLASS COmputer Program for Bit Error Rate with RFI, NASA Goddard Space Flight Center, TR-0883-8214-2, August 1982.

[4] W. W. Wu, D. Haccoun, R Peile, and Y. Hirata, " Coding fir Satellite Communication", *IEEE Journal on Selected Areas in Communications*, Vol. SAC-5, No. 4, May 1987, pp. 742-748.

[5] T. Kaplan, Charization of Burst Statistics at the output of the TDRSS Viterbi Decoder, STEl 003-2TK90/M, May 2, 1990.

[6] B. Sklar, *Digital Communications Fundamentals and Applications*. Prentice Hall, 1988.

[7] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York, John WIley & Sons, Inc., 1967.

[8] R. G. Gallager, *Information Theory and Reliable Communication,* New York; John WIley & Sons, Inc., 1969.

[9] R. J. McEliece, *The Theory of Information and Coding,* Cambridge, Cambridge University Press, 1985.