MODELISATION DES ERREURS APRES DECODAGE A MAXIMUM DE VRAISEMBLANCE

MODELING OF ERRORS ISSUED FROM MAXIMUM LIKELIHOOD DECODING

K. FAZEL, Ph. SALEMBIER

*LEP : Laboratoires d'Electronique et de Physique appliquée**
3, avenue Descartes - 94451 Limeil-Brévannes Cedex (France)

** Membre de l'Organisation de Recherche Internationale de Philips*

RESUME

Le décodage à Maximum de Vraisemblance (MV) est une technique très largement utilisée dans le cadre des codes correcteurs d'erreurs, des modulations codées et des codes à réponse partielle par exemple. Les erreurs à la sortie d'un tel décodeur sont corrélées et ne peuvent pas être représentées par un modèle Binaire Symétrique. Cet article est consacré à la définition et à la présentation d'une modélisation analytique de telles erreurs. La méthode proposée est basée sur la connaissance du processus de création des erreurs lors d'un décodage à MV. Ce modèle peut être utilisé pour l'étude et la conception de "super-code", par exemple des codes concaténés ou des codes en cascades. De plus, il peut être servi de modèle générateur.

SUMMARY

Maximum Likelihood (ML) decoding has been used in many cases, such as classical error control techniques, coded modulation techniques and partial response encoding techniques. The errors issued from this decoding occur in bursts and cannot be represented by the classical Binary-Symmetric-Channel. This article is dedicated to introducing an analytical method for modeling such errors. A Markov Model is used and is based on the ML decoding error producing mechanism. This model can be used to design and study "super coding" techniques such as concatenated or cascaded codes and it can also be used to regenerate a similar error sequence without actually encoding and decoding the signal, and this can save a lot of simulation time.

INTRODUCTION

To improve the quality of a digital link, or to adapt the transmitted spectrum to the channel characteristics, several techniques involving coding have been studied. As an example, coding can be used to improve power efficiency. The redundancy introduced can then be transmitted by increasing the data rate(this is the classical error protection approach) or by increasing the size of alphabet. This last technique introduced by Ungerboeck gives rise to the so called coded modulations [1]. Coding can also be used to shape the transmitted signal spectra. This can be done by correlative level encoding techniques.

In the case of additive white gaussian noise (AWGN), the optimum decoder performs the maximum likelihood (ML) estimation of the transmitted sequence. This decoding process can be viewed as the search of the "most likely" path in the trellis representing the encoding process [3]. In searching for the most probable path, the decoder occasionaly chooses a wrong path and makes errors. This wrong path which remerges to the correct path after ℓ steps contains several errors. This effect of producing errors is such that errors occur in burst separated by fairly long error free gaps. This biased behavior of the decoder which is either "too good" or "too bad" exhibits the presence of a memory effect. The system cannot therefore be represented by the classical Binary Symmetric Channel (BSC) model.

Yet the modeling of errors at the output of a ML decoder is a very useful tool. For example, it allows the design and study of certain "super coding" techniques such as concatenated or cascaded codes which take advantage of the presence of memory (in information theory parlance, for a given bit error rate, memory increases capacity). And a model can also be used to regenerate a similar error sequence whithout actually encoding and decoding the signal (this can save a lot of simulation time).

Classical methods of modeling "channels with memory" can be used to represent the errors at the output of a ML decoder. They are often based on a Markov chain consisting of a finite number of states with defined transition probabilities [4-5]. To construct these models accurately requires the prior knowledge of the error distribution. However this information is not always available and computer simulations in order to establish it often require an excessive amount of CPU time.

Thus, the motivation to find a new method of modeling the errors issued from the ML decoder without the knowledge of error distribution beforehand. In section II, after a brief presentation of classical methods of modeling, we describe a Markov model based on the Viterbi decoder error producing mechanism. An important advantage of the proposed model is that it can be determined using the so called "Transfer Function" technique (section III). To illustrate this new method of modeling, two examples of coded 16 PSK modulation are given in section IV. The performance of this method is also discussed in this last section.

DESCRIPTION OF MODELING

As it has been said previously, errors at the output of a Viterbi decoder occur in bursts. Classical methods used for "bursty channels" can thus represent this memory effect. This approach is reviewed in the following section.

1. Classical approach of modeling channels with memory

"Bursty channels" are often modeled by a Markov chain consisting of a finite number of states with transition probabilities. The set of states are split into two subsets corresponding to the presence or the absence of errors. Various statistics about error sequences can be computed by means of these models, and they can also be used to generate error sequences with the same statistical properties. In this case, state sequences are mapped into error sequences, taking into account the states partitioning into two subsets. Note that if the two subsets include more than one state, the inverse mapping (from the error sequence to the state sequence) is in general not possible.

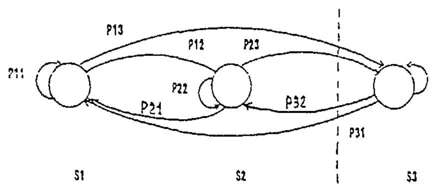
A very simple and popular model is the so called Fritchman Simplified Partitioned Markov binary model [4-5]. The states are partitioned into (N-1) error free states and a single error state. No transition is allowed between error free states. To define the transition probabilities of this model, the knowledge of the Error-Gap-Distribution* (EGD) is required. This distribution is obtained experimentally or by computer simulation. It is then expressed as the sum of (N-1) exponentials using curve fitting techniques. This last expression gives directly the transition probabilities of the model [5].

Although this approach has the advantage of being very general, its main disadvantage is that it requires a huge amount of data in order to obtain the EGD. Experimental data are often unavailable, and computer simulations require an excessive amount of CPU time, especially for low bit error rates (BER). To overcome this difficulty we introduce in the following a new approach to modeling which can be determined analytically.

* The EGD is defined by : $P(0^m/1)$ = Probability that at least m successives error free bits ("0") will be encountered next, given that an error ("1") has just occured.

2. A new approach to modeling

The model described here is a Markov model with two error free states (good states) and a single error state (bad state). These three states correspond to the Viterbi error producing mechanism. As has been said previously, the Viterbi algorithm searches the trellis for the path which is the "most likely". The decoder occasionaly chooses a wrong path. This is an error event, but all the digits corresponding to a wrong path are not necessary wrong. Thus the first state (S_1) represents the digits belonging to the correct path (of course it is an error free state). The two remaining states are devoted to the error events. The first one (S_2) corresponds to the correct digits of the wrong path (error free state) and the last one (S_3) represents the erroneous digits of the wrong path (error state). This model is illustrated in the following diagram.



where : $\sum_{j=1}^{3} P_{ij} = 1$

$i = 1, 2, 3$

In the above diagram, all the transitions between states are allowed. But very often, some transitions can be eliminated. For example, the structure of trellis can be such that the first bit of each wrong path is wrong (in this case $P_{12} = 0$), or that the last bit of each wrong path is correct (then $P_{31} = 0$).

As with the classical model, this new model can be used to compute various statistics about error sequences, or to generate the error sequences themselves (note that the knowlege of the encoder allows the mapping of an error sequence into a state sequence, which was not possible with classical models). The main advantage of this model is that it can be determined analytically. This is the subject of the following section.

ANALYTICAL DERIVATION OF THE MODEL'S PARAMETERS

The analytical derivation of the model transition probabilities is based on the "Transfer function" of the code. This function describes in a compact form all the possibilities of choosing a wrong path instead of a correct path [3]. It can be derived from the state diagram of the encoder.

Since in general the distance (and thus the probability) between an incorrect path and a correct path depends on the correct path, two methods have been developed to compute this transfer function. The first one is based on a product state diagram [6]. In this case, the product state corresponds to pairs of encoder states. This method is general, but its complexity grows with N^2, where N is the number of code states. To reduce this complexity a second method was proposed in [2]. This method is less general than the first one, but its complexity is proportional to N.

In all cases, a state diagram is defined, and each branch is labeled with two factors : the first one is the probability $P(x_k, y_k)$ of this transition occuring and depends on the distance between the correct (x_k) and wrong (y_k) received symbols. The second one $G(x_k, y_k)$ depends on the particular information we want to get, and will be described later.

The transfer function averages over all possible correct and incorrect paths, the factor $\prod_{k=0} P(x_k, y_k)\, G(x_k, y_k)$:

$$T(P,G) = \sum_{XYk}\sum \pi \; p\,(x_k,y_k)\; G(x_k,y_k)$$

where :

$X = \ldots x_k,\ x_{k+1}, \ldots$ is the correct transmitted path

$Y = \ldots y_k,\ y_{k+1}, \ldots$ is the wrong received path

$P(x_k,y_k) \simeq \frac{\alpha}{2}\ \exp\left(-\frac{(x_k - y_k)^2}{4NO}\right)$ in a AWGN case.

The computation of the transition probabilities of the model requires the knowledge of several terms : the error event probability PE, the average length of an error event L, the bit error probability after decoding Pb, the average number of pattern (a,b) (a,b ϵ 0,1) during an error event N(a,b), and the probability that the first digit of an error event will be in error PF_1. All these terms can be computed with the transfer function technique and the appropriate $G(x_k,y_k)$:

- The error event probability PE : the computation method of this parameter is very well known [2-3]. The error event is taken into account by setting $G(x_k, y_k) = 1$, if the two paths diverge, otherwise $G(x_k, y_k) = 0$. Then $PE = T(P, 1)$.

- The bit error probability Pb : this is also a classical computation [2-3]. The number of wrong bits during an error event is determined using $G(x_k, y_k) = I^n$, where n is the number of digits which differ between the two decoded error symbols corresponding to x_k and y_k. Since the useful information is the exponent of the variable I (which is itself a "dumb variable"), the bit error probability has to be derived :

  $$Pb = \frac{\delta T\,(P,I)}{m\,\delta I}\Big|\, I = 1 \text{ for a (m, n) code.}$$

- The average length of an error event L : in this case $G(x_k, s_k) = I^m$, where m is the number of information bits concerning each transition, and

  $$L = \frac{\delta T\,(P,I)}{PE\,\delta I}\Big|\, I = 1$$

- PF1 : this probability can be computed easily as previously mentioned. The transitions in the state diagram corresponding to the beginning of an error event have to be marked with $G(x_k, y_k) = I$, if the first digit is wrong, otherwise $G(x_k, y_k) = I^0$ (let us recall that this probability can often be defined without any computation, depending on the trellis structure).

- Average number of the error pattern N (a,b) : each transition in the state diagram corresponds to a decoded error symbol of m digits. It is very easy to count the number of the pair (a,b) (a,b ϵ 0,1) inside an error symbol of m digits. But the transition between the previous and current error symbols has to be taken into account. This problem can be solved if the state diagram is labeled by a 2x2 matrix instead of scalars ($G(x_k, y_k)$ as previously). Indeed let us define Ω as the set of all possible error symbols. Ω can be split into two subsets Ω_0 and Ω_1, where Ω_1 (resp. Ω_0) is the set of error symbols ending with an erroneous digits (resp.correct digits). Each transition of the state diagram is labeled by :

  $$M(P,I) = \begin{bmatrix} M\,\Omega_0\,\Omega_0\,(P,I) & M\,\Omega_0\,\Omega_1\,(P,I) \\ M\,\Omega_1\,\Omega_0\,(P,I) & M\,\Omega_1\,\Omega_1\,(P,I) \end{bmatrix}$$

  Each term $M\,\Omega_i\,\Omega_j(P, I)$ corresponds to the case where the current error symbol belongs to Ω_j and the previous error symbol belongs to Ω_i.
  $M\,\Omega_i\,\Omega_j\,(P, I)$ has the following form : $P(x_k, y_k)\, I^n$, $(G(x_k, y_k) = I^n)$, where n is the number of the pair (a,b) in the current error symbol, possibly plus one if the pair (a,b) occurs in the transition between error symbols.

The transfer function of the state diagram is then a 2 x 2 matrix :

$$T(P,I) = \begin{bmatrix} M\,\Omega_0\,\Omega_0\,(P,I) & T\,\Omega_0\,\Omega_1\,(P,I) \\ T\,\Omega_1\,\Omega_0\,(P,I) & T\,\Omega_1\,\Omega_1\,(P,I) \end{bmatrix}$$

And the average number of error pattern (a, b) is given by :

$$N(a,b) = \frac{1}{PE}\,\frac{\delta}{\delta I}\left[\sum_{i,j} T\,\Omega_i\,\Omega_j\,(P,I)\right]\Bigg|_{I=1}$$

To illustrate the computation of this last term, let's take an example. We are interested in computing the average number of the pair (1,1) in an error event : N(1,1). To this end, each transition of the state diagram is labeled by M, where :

$$M\,\Omega_i\,\Omega_j = P\,(x_k,\ y_k)\ I^n \ ;\ i = 0,1,\ j = 0,1$$

$\Omega_0 = \{$error symbols ending with "0"$\}$

$\Omega_1 = \{$error symbols ending with "1"$\}$

Error symbols$\Longleftrightarrow$symbols ϵ Error event.

Where :
- $P\,(x_k,\ y_k)$ is the probability of receiving y_k instead of x_k
- n is the number of pair (1,1) encountered in the current error symbol ($\epsilon\ \Omega_1$ or Ω_0), plus one if the previous symbol $\epsilon\ \Omega_1$ and the first digit of the current symbol is "1". If we have the following transition (between error symbols) in the state diagram :

  ..., 101, 110, ...,

  previous symbol current symbol

  where 101 $\epsilon\ \Omega_1$, 110 $\epsilon\ \Omega_0$, then n = 2.

And N(1,1) can be computed as described above by taking the derivation of the transfer function with respect of I.

Now the computation of the model transition probabilities can be carried out very easily.

Derivation of the model parameters

Transitions emanating from state S1

P_{12} (respect. P_{13}) is the error event probability knowing that the first digit is an error free (resp erroneous) digit. They are given by (PE<<1) :

$$P_{12} = \frac{PE}{m}\,(1 - PF_1), \quad P_{13} = \frac{PE}{m}\,PF_1$$

Taking into account, the fact that $\sum_{j=1}^{3} P_{ij} = 1$, i = 1,2,3, P_{11} is defined by :

$$P_{11} = 1 - \frac{PE}{m}$$

Transitions emanating from state S2

P_{22} represents the probability of a wrong path having an error free bit after an error free bit. It is the ratio between the average number of error pattern N(0,0) and the average number of error free digits N(0) along a wrong path :

$$P_{22} = P\,(0/0) = \frac{N(0,0)}{N(0)} = \frac{N(0,0)}{L - \dfrac{m\,Pb}{PE}}$$

In the same manner, we can write :

$$P_{23} = P(1/0) = \frac{N(0,1)}{N(0)} = \frac{N(0,1)\;PE}{L - \dfrac{m\,Pb}{PE}}$$

then $P_{21} = 1 - P_{23} - P_{22}$.

Transitions emanating from state S3

As described previously :

$$P_{32} = P(0/1) = \frac{N(1,0)}{N(1)} = \frac{N(1,0)\ PE}{m\ Pb}$$

then

$$P_{33} = P(1/1) = \frac{N(1,1)}{N(1)} = \frac{N(1,1)\ PE}{m\ Pb}$$

$$P_{31} = 1 - P_{32} - P_{33}.$$

EXAMPLES

In this section, the technique described above is applied to two "coded modulation" schemes. These are classicaly coded 16 PSK with two and four states convolutional codes. The transmission chain under study is shown in figure 1, and the two convolutional codes of rate (3/4) are presented in fig. 2.

As it has been seen before, the transfer function technique can be used to compute various statistics about error pairs along an erroneous path. The state diagram has to be labeled by matrices instead of scalars. To illustrate this point, fig. 3 presents P(1/1) (resp. P(1/0)), the probability of having an error knowing that the previous bit was wrong (resp. was correct) along a wrong path, as a function of the bit error rate. Results obtained by computer simulation (denoted by SIMUL) can be compared to the graph generated by the analytical estimation described here (denoted by ANALYT). It can be seen that there is a very good agrement between these two graphs.

In order to illustrate the accuracy of this new method of modeling, the error gap distribution (EGD) is presented in figure 4. For each code, four curves are given : the first one (SIMUL) represents the EGD obtained by computer simulation and can be viewed as the reference curve. The second one (FRITCH) is the EGD obtained with a classical Fritchman simplified partitioned Markov model (of course, this model was derived from the simulated EGD by a curve fitting method). The third curve (ANALYT) gives the results from the analytically derived model, the last curve presents the EGD given by a Binary Symmetric Channel (BSC). The difference between this last curve and the others reflects the presence of memory in the system. It can then be noted that the memory effect is stronger in the case of a 4 state encoder than in the case of a 2 state encoder. These results also show that the analytical model is as accurate as the Fritchman model. Of course, the main advantage of this analytical approach is its low cost in CPU time.

CONCLUSION

This new method of modeling the errors issued from Viterbi decoder is based on the "Transfer Function" technique. By mean of this technique, many statistics can be derived which permit the parameters of the model to be determined, such as the correlation between two consecutive digits. In this case the labeling of the state diagram of the code will be in the form of a 2x2 matrix instead of a scalar.

This modeling method performs as well as the classical method of modeling a channel where the errors occur in bursts. The main advantages of this model are its simplicity (only three states are needed) and the analytical derivation of its parameters. Its complexity is proportional to the complexity of deriving the transfer function.

This model can be utilized to design "super Codec" for error control, where the inner decoder will be a Viterbi decoder, and of course to regenerate error sequences for test runs.

REFERENCES

[1] G. Ungerboeck, "Channel Coding with Multilevel/ Phase Signals", IEEE Trans. on IT., vol. 28, pp. 55-66, June 1982.

[2] E. Zehavi, J.K. Wolf, "On the Performance Evaluation of Trellis codes", IEEE Trans. on IT, vol. 33, pp. 196-202, March 1987.

[3] A.J. Viterbi, "Convolutional Codes and their Performance in Communication Systems", IEEE Trans. on COM. vol. 19, pp. 751-772, Oct. 1971.

[4] L.N. Kanal, A.R.K. Sastry, "Models for Channels with Memory and their Application to Error Control", Proc. of IEEE, vol. 66, pp. 724-743, July 1978.

[5] B.D. Fritchamn, "A Binary Channel Characterization Using Partitioned Markov Chains", IEEE. Trans. on IT, vol. 13, pp. 221-227, April 1967.

[6] E. Biglieri, "High Level Modulation and Coding for Non-Linear Satellite Channel", IEEE Trans. on COM, vol. 32, pp. 616-622, May 1984.

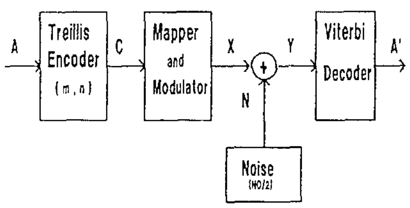


Fig.1 A Digital Transmission Chain

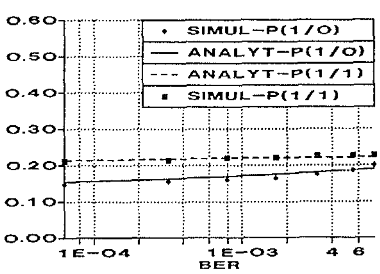


Fig.3a- 2 States 'Error Event' Statistics

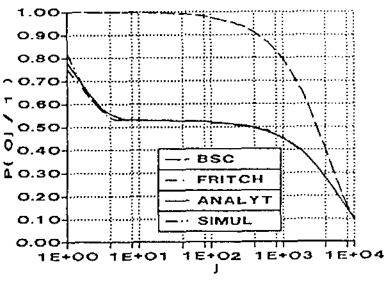
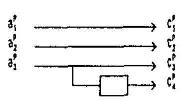
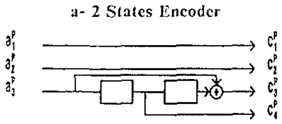


Fig.4a- 2 States Channel EGD (BER = 0.21E-03)



a- 2 States Encoder



b- 4 States Encoder

Fig.2 2 States & 4 States Encoders

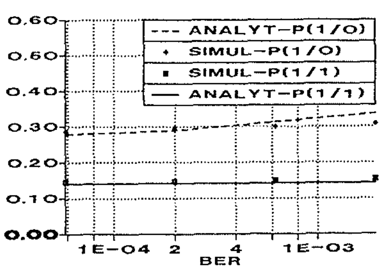


Fig.3b- 4 States 'Error Event' Statistics

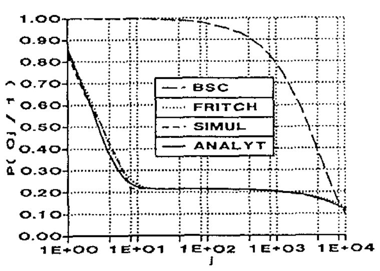


Fig.4b- 4 States Channel EGD(BER = 0.21E-03)