Performance of the Viterbi Algorithm on a Polya Channel

Néstor R. Barraza

Universidad Nacional de Tres de Febrero y Facultad de Ingeniería. UBA
nbarraza@untref.edu.ar

Abstract— **The Performance of the Viterbi algorithm with a new Polya contagion noise process is analyzed. The markov channel is the memory channel usually studied in the bibliography. The Polya channel was introduced years ago as a special case of a memory channel. In this work, the Polya contagion scheme is considered as a noise process and a Gauss-Polya noise model is introduced this way in order to perform soft decision decoding. The behavior of the Viterbi algorithm on a given recursive sequential circuit is analyzed under this special case of memory channel.**

Keywords— **Viterbi, Polya, contagion, maximum likelihood.**

1 INTRODUCTION

The interest in the study of memory channels has been brought in order to analyze correlated bits noise and burst errors introduced by several hardware recording procedures like percolation processes in high-density magnetic recording and clustering defects in silicon, see [1] and [2].

For the mentioned type of noise, sequential codes with interleaving have been demonstrated to have good performance in order to correct intersymbol interference and burst errors. Since the Viterbi algorithm is the simple maximum likelihood decoder usually used in sequential codes, its behavior and availability of updating to take into account memory channels is an interesting matter of study. The aim of this work is to introduce a Gauss-Polya noise model in order to analyze the performance of the Viterbi algorithm using either hard and soft decision rules.

This paper is organized as follows: A brief introduction to the convolutional codes and the maximum likelihood decoding procedure using Viterbi algorithm is given in Section 2. Next, a short review of memory channels considering Markov and Polya noise processes is presented in Section 3. A Gauss-Polya noise model is introduced in Section 4. Results of simulations of the Viterbi algorithm are shown and discussed in Section 5. At the end, some conclusions reviewing the contents of this work are made in Section 6.

2 CONVOLUTIONAL CODES. MAXIMUM LIKELIHOOD DECODING

2.1 Viterbi Algorithm

The simple maximum likelihood detector in the decoder is the Viterbi algorithm named after Viterbi [3]. This algorithm presents two alternatives, the hard decision rule: when the estimated output state sequence is that having the minimum Hamming distance, and the soft decision rule: when the estimated sequence is that having the minimum distance using the actual received value not quantized to either +1 or -1, it means the sequence with the minimum $||r_l - u_l||^2$ value, where r_l and u_l are the received and the possible value at instant l.

3 MEMORY CHANNELS

As it was pointed out in the introduction, in order to analyze some cases of intersymbol interference, memory channels have to be considered. In this type of channels, the noise introduced in some bit of a given codeword, depends on the noise introduced in previous bits. In more complex cases, that noise depends also on the values of the signal. Two examples of memory channels are explained below, the Markov and the Polya channel.

3.1 Markov Channel

In Markov channels, the output sequence is a Markov process where the output in a given time depends on previous output values and perhaps also on previous input values. A Markov noise model introduced in [1] and [4] considers the following density function:

$$f(z_k \quad |z_{k-1}, \cdots, z_{-\infty}, x_{-\infty}, \cdots, x_{\infty}) = f(z_k|z_{k-1}, \cdots, z_{k-L}, x_{k-1}, x_k) \quad (1)$$

where $\{z_k\}$ is the output sequence, $\{x_k\}$ is the state-channel input sequence and L is the order of the Markov process.

3.2 Polya Channel

This type of noise process based on the Polya urn model was introduced years ago in [2]. Further applications were presented in [5]. This urn model was originally introduced in order to analyze a contagion process. The model consists in making extractions from an urn containing balls of given colors. A number of c balls of the same color as that obtained are then introduced in the urn, increasing the probability of extracting a ball of that color.

In the Polya Channel, the output signal at the time i is obtained as the module-2 sum between the input and the noise process $Y_i = X_i + Z_i$. The noise process Z_i is a binary sequence corresponding to extractions of either black or white balls from a given urn composed of balls of two colors. Considering that c balls of the extracted color are introduced in the urn, the probability of getting n_w white balls after n extractions is given by [6]:

$$P_{N_w}(n_w|n) = \binom{n}{n_w} \frac{\Gamma(\frac{w}{c}+n_w)\Gamma(\frac{b}{c}+n-n_w)\Gamma(\frac{w+b}{c})}{\Gamma(\frac{w+b}{c}+n)\Gamma(\frac{w}{c})\Gamma(\frac{b}{c})} \tag{2}$$

where b and w are the initial number of black and white balls in the urn.

Let $Z_i = \{0,1\}$ be the random variable corresponding to the extraction of a white ball, it means $z_i = 1$ when the i^{th} extraction is white, and $z_i = 0$ otherwise. Then, the number of white balls obtained at the $n^{th} - 1$ extraction is given by $s_{n-1} \doteq \sum_{i=1}^{n-1} z_i$. Defining new variables,

$$p = \frac{w}{w+b} \tag{3}$$

$$\sigma = \frac{b}{w+b} \tag{4}$$

$$\gamma = \frac{c}{w+b} \tag{5}$$

the probability of z_n, i.e. the probability of error or not error at time n is as follows:

$$P(z_n|s_{n-1}) = \left(\frac{p+s_{n-1}}{1+(n-1)\gamma}\right)^{z_n} \text{x} \left(\frac{\sigma+(n-1-s_{n-1})}{1+(n-1)\gamma}\right)^{1-z_n} \tag{6}$$

The discrete Polya noise model is a stationary binary process with probability $P(Z_k = 1) = p$, see [2] and [6].

Noise introduced by the Polya channel is considered to be additive with a probability of error given by (6), then the output is obtained by a module-2 addition between the input and noise:

$$y_i = x_i \oplus z_i \tag{7}$$

4 GAUSS-POLYA CHANNEL

In order to consider soft decisions in the decoder, a continuous noise variable should be taken into account. Then, a gaussian white noise term is added to Eq. (7), resulting:

$$y_i = x_i \oplus z_i + w \tag{8}$$

the last equation is a Gauss-Markov like noise, we could name this type of noise a Gauss-Polya model.

5 SIMULATIONS

The performance of the Viterbi algorithm is analyzed for the convolutional code obtained from a simple recursive sequential circuit shown in Fig. 1. Parallel and serial concatenations with interleaving of this circuit were analyzed in a well known paper on turbocodes, [7].

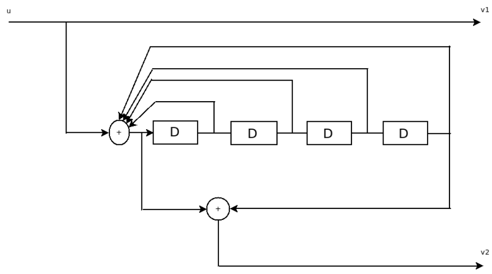

Figure 1: Simple recursive sequential circuit for an R=1/2 convolutional code

The sequential circuit depicted in Fig. 1 has the following transfer matrix:

$$G(D) = \begin{pmatrix} 1 & \frac{1+D^4}{1+D+D^2+D^3+D^4} \end{pmatrix}$$

Simulations are implemented for both rules, hard decision according to Eq. (7) and soft decision following Eq. (8). Due to dispersion in the contagion process, the bit error rate has a big variance, then, for each signal to noise ratio, bit error rates are split in two regions according they are above or lower the average. This splitting process gives two limits, un upper and a lower bound, both of them are shown. Results of simulations showing the upper and lower bounds for hard and soft decisions rules are shown in Fig. 2.

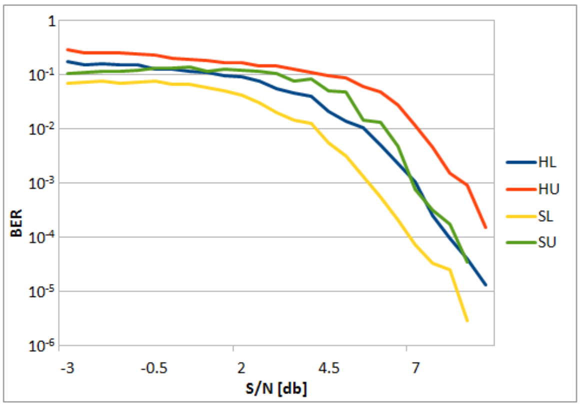

Figure 2: BER vs. SN ratio for the Polya channel showing the hard lower [HL], hard upper [HU], soft lower [SL] and soft upper [SU] bounds.

The bit error rate BER corresponds to the initial probability $p = p(z_k = 1)$ as defined in (3), see also [5]. In order to simulate the contagion process, an initial value $w = 1$ was considered in (3).

Simulations of the same convolutional code for a memoryless AWGN channel are shown in Fig. 3 for comparison.

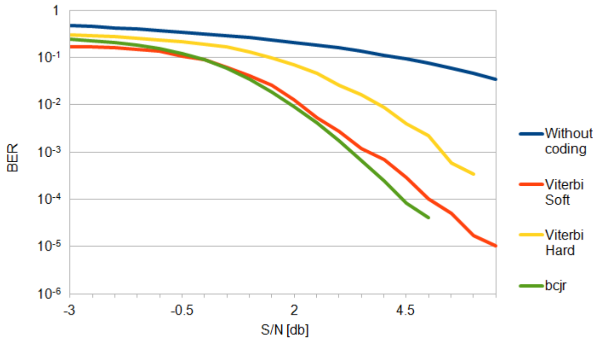

Figure 3: BER vs. SN ratio for the memoryless AWGN channel for Viterbi and bcjr [8] algorithms.

The maximum likelihood estimator was always chosen as that having the minimum Hamming distance. Even for low values of the SN ratio, we never found that the output state sequence having the maximum Hamming distance resulted in a better estimator, despite it should be considered as remarked in [2]. However, as it was theoretically demonstrated in [2] and [9], the maximum likelihood estimator not always coincides with the minimum Hamming distance, but depends on the type of channel.

Values of BER for the upper bound soft decision curve approximately coincides with the lower bound hard decision curve. Results show that for the Polya channel, the simple estimator given by the minimum Hamming distance gives worse values of BER than the memoryless AWGN channel, though better than the markov channel with the modified metrics shown in [4]. Despite BER values are worse than those for the memory AWGN channel because of the contagion process, they are sufficiently acceptable for a memory channel. An interleaving process appears to be interesting to be considered for this type of noise and it will be analyzed in a future work. The behavior of turbocodes for this type of noise is also interesting to analyze. This requires to update the transition metrics in a similar way as it was shown in [1] for a Markov channel. This issue will be also studied in a future work.

6 CONCLUSIONS

The behavior of the Viterbi algorithm with both decision rules, hard and soft, in a simple convolutional code on a Polya channel was analyzed. A contagion Gauss-Polya noise process was introduced in order to simulate the channel. Simulations show that the classical minimum Hamming distance corresponding to the maximum likelihood estimator has good performance compared to the memoryless channel. This shows that a modified metrics does not appear to be necessary at least for the Viterbi algorithm. Behavior of turbocodes and interleaving processes will be analyzed in a future work.

ACKNOWLEDGMENTS

The author wants to thank ”Universidad Nacional de Tres de Febrero” and ”Facultad de Ingeniería. UBA” for support. The author would also like to thank the anonymous reviewers for their valuable comments that helped to improve this paper.

REFERENCES

[1] A. Kavcic and J. M. F. Moura, “The viterbi algorithm and markov noise memory,” *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 291–301, 2000.

[2] F. Alajaji and T. E. Fuja, “A communication channel molded on contagion,” *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 2035–2041, 1994.

[3] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967. [Online]. Available: http://dx.doi.org/10.1109/tit.1967.1054010

[4] A. Kavcic, “Soft-output detector for channels with intersymbol interference and markov noise memory,” in *IEEE Global Telecommunications Conference (GLOBECOM’99)*, Rio de Janeiro, December 1999, pp. 728–732.

[5] A. Cohen, F. Alajaji, N. Kashyap, and G. Takahara, “Lp decoding for joint source-channel codes and for the non-ergodic polya channel,” *IEEE Communications Letters*, vol. 12, no. 9, pp. 678–680, 2008.

[6] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*, 3rd ed. Wiley, Jan. 1968.

[7] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1,” in *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*. IEEE, May, pp. 1064–1070 vol.2.

[8] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate (Corresp.),” *Information Theory, IEEE Transactions on*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[9] C. F. Caiafa, N. R. Barraza, and A. N. Proto, “Maximum likelihood decoding on a communications channel,” in *RPIC Reuniones en Procesamiento de la Información y Control*, Rio Gallegos, October 2007, pp. 728–732.