Performance of Convolutional/Single Parity Check Turbo Codes Over a Rician Fading Channel

Slava Yordanova[1], Nikolay Velikov[2] and Ginka Marinova[3]

Abstract - **Short-frame turbo product codes for real-time wireless speech communications are studied in this paper. Performance of the modeled coding system is examined through simulations of Rician fading channel. The obtained results indicate that the performance of these codes is quite exceptional given their decoding complexity.**

Keyword – **Turbo codes, iterative decoding, simulation, Rician fading channel.**

I. INTRODUCTION

The development of the turbo codes [1]-[4] is among the most significant achievement in the error control technique for the past couple of decades. The rate $R_c = 0.5$ convolutional turbo code, presented by Berrou et. al. in [1] shows a bit error rate (BER) of 10^{-5} at a signal-to-noise ratio (SNR) E_b / N_0 only 0.7 dB above the Shannon capacity limit. Although powerful the originally proposed turbo codes operate at large data frame size and number of iterations. This leads to long latency and, therefore, unsatisfactory performance for real-time communications.

Most of the further research on concatenated codes with soft-in/soft-out (SISO) iterative decoding have been dedicated to the convolutional turbo codes (CTC) or block turbo codes (BTC). A distinct approach is adopted in [5], where a hybrid convolutional/single parity check turbo code is proposed and tested through simulation over additive white Gaussian noise (AWGN) channel. The proposed scheme indicates high performance at a given low complexity and latency, which makes it suitable for real time communications. However this is not applicable for certain types of communications such as mobile communications since a more complex channel model should be used for this purpose. In this paper a short data frame hybrid turbo code (HTC) schemes, based on convolutional/single parity check codes concatenation over the more generalized Rician fading channel is studied. As it will be shown the designed scheme combines satisfactory performance and low complexity and latency and is aimed on use in real time communications over large scale types of transmission media.

[1]Slava Yordanova is with the Technical University of Varna, Department of Computer Science and Technologies, Studentska Street 1, Varna 9010, Bulgaria, E-mail: slava_y@abv.bg

[2]Nikolay Velikov is with the Naval Academy "N. Vaptzarov" of Varna, Department of Electronics, V. Drumev Street 75, Varna 9026, Bulgaria, E-mail: na_velikov@mail.bg

[3]Ginka Marinova is with the Technical University of Varna, Department of Computer Science and Technologies, Studentska Street 1, Varna 9010, Bulgaria, E-mail: gin_kaleva@abv.bg

The paper is organized as follows: In Section 2, a brief description of the system model is given. Section 3 presents the simulation results. Finally, the concluding remarks are given in Section 4.

II. SYSTEM MODEL

There are many factors that affect the performance of turbo codes. Among the most important are the interleaver size (with the number of iterations) and the decoding algorithm. CTC's with rate $R_c \le 0.5$ and 8 or more iterations perform extremely well [1]-[4]. Increasing the interleaver size increases the coding gain. However so does the encoding/decoding latency. On the other hand, downsizing the interleaver in order to improve the latency, leads to performance deterioration. At a size below a threshold value of approximately 200 bits, the convolutional code outperforms the CTC of comparable complexity [4]. Regarding the decoding algorithm the same trade-off between performance and complexity is observed. The maximum *a posteriori* (MAP)-based decoders have better performance but high complexity and slow speed, while the soft output Viterbi algorithm (SOVA)-based decoders are of a low complexity and relatively high speed. This makes the latter attractive for practical implementations.

The SISO iterative decoding of two or more concatenated block codes, known as block turbo code (BTC) is suitable for high code rates, typically greater than 0.7. The performance of BTC's doesn't depend significantly on the interleaver design and their asymptotic BER outperforms that of the CTC's due to the larger minimum code distance. Another advantage is the possibility for fast parallel decoding of the data block rows/columns since they are independent.

Both CTC's and BTC's have excellent performance for wide range of code rates and data frame sizes but more research are needed for low complexity short data frame (less than 200 bits) turbo codes over Rician fading channels.

Hybrid concatenation of convolutional/single parity check codes with non-iterative SISO decoding was considered for the first time by Hagenauer and Hoeher in [6] and elaborated by Freemen and Michelson [7] for more powerful component codes. Further study of the hybrid convolutional/single parity check codes with iterative (turbo) decoding is considered in [5]. Here we follow the approach in [5] to study the performance of hybrid convolutional/single parity check codes (HTC) over a Rician fading channel.

First consider the encoding/decoding process of the above mentioned coding scheme. Regarding the encoding phase the data bits $a_i, i = 1, 2, ..., M$ are first arranged in a rectangular array. Then, all columns are encoded with $(n, n-1)$ single parity check (SPC) outer code. Finally, the rows, including those containing the parity bits, are encoded with a $\nu = 2$ or $\nu = 3$, $R_c = 0.5$ convolutional inner code. Thus, the overall code rate R of the hybrid turbo code will be $R = R_c \cdot R_s$, where $R_s = (n-1)/n$ is the rate of the SPC code.

The Rician fading channel model can be presented the following way [8]: The fading amplitude r_i at the ith time instant can be represented as

$$r_i = \sqrt{(x_i + \beta)^2 + y_i^2}\,, \tag{1}$$

where β is the amplitude of the direct component and x_i, y_i are samples of zero-mean stationary Gaussian random processes each with variance σ_0^2. The ratio of direct to defuse energy defines the so-called Rician *K*-factor, which is given by

$$K = \beta^2 / 2\sigma_0^2\,. \tag{2}$$

The best- and worst-case Rician fading channels associated with *K*-factors of $K = \infty$ and $K = 0$ are the Gaussian and Rayleigh channels with strong LOS and no LOS path, respectively. So, the Rayleigh fading channel can be considered as a special case of a Rician fading channel with $K = 0$.

The Rician cumulative distribution function (CDF) is given by

$$C_{Rice}(r) = 1 - e^{-\gamma} \sum_{m=0}^{\infty} \left(\frac{\beta}{r}\right)^m \cdot I_m\left(\frac{r\beta}{\sigma_0^2}\right), \tag{3}$$

where $\gamma = (K + r^2 / 2\sigma_0^2)$. For practical purpose it is sufficient to increase m to the value, where the last terms contribution becomes less than 0.1 percent.

Consider the generation of uncorrelated Rician-distributed fading sequences. The mean-squared value of the Rician distribution is known to be $2\sigma_0^2(K+1)$, where σ_0^2 is the variance of the component Gaussian noise processes in (1). Further, it is often required a Rician distribution with unit mean-squared value, i.e., $E\{r^2\} = 1$ so that the signal power and the signal-to-noise ratio (SNR) coincide. In order to meet the requirement $E\{r^2\} = 1$, the equation (1) can be written in the form [8]

$$r_i = \sqrt{\frac{(x_i + \sqrt{2K})^2 + y_i^2}{2(K+1)}}\,, \tag{4}$$

where now x_i, y_i are samples of zero-mean stationary Gaussian random processes each with variance $\sigma_0^2 = 1$. So, the desired Rician fading sequences can be generated according to (4).

A simplified diagram of the HTC decoder is shown in Fig.1. The log-likelihood ratio (LLR) L_i at the output of a SISO decoder can be represented in general as [4]

$$L_i = y_i + g_i + l_i\,, \tag{5}$$

where $y_i = \frac{4E_s}{N_0} r_i$ is the weighted channel observation, g_i is the *a priori* information and l_i is the so-called extrinsic information gained by the current stage of decoding.

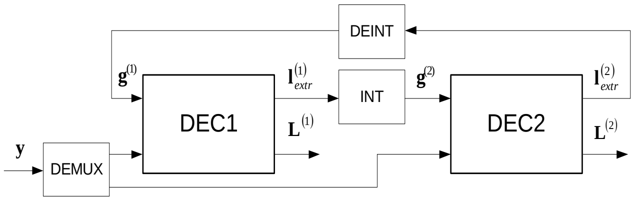

Fig.1. Turbo decoder block diagram

The first elementary decoder (DEC1) in Fig.1 uses soft output Viterbi algorithm (SOVA) to form an estimate of the LLR of each bit encoded by the convolutional code (e.g., the bits of the SPC code). The essence of SOVA is finding the most likely transmitted sequence of bits along with reliability values for the bits [5], [6]. The likelihood ratio or “soft” value of the binary path decision at time *i* can be defined as

$$\Delta_i^0 = \frac{1}{2}\left(M_i^{m_0} - M_i^{\tilde{m}_0}\right), \tag{6}$$

where $M_i^{m_0}$ and $M_i^{\tilde{m}_0}$ are the path metrics of the survivor and competitor path, respectively. Now, the SOVA output LLR of the δ-delayed decision $\hat{b}_{i-\delta}$ can be expressed as

$$L\left(\hat{b}_{i-\delta}\right) \approx \hat{b}_{i-\delta} \cdot \min_{l=0,\ldots,\delta} \Delta_i^l\,. \tag{7}$$

More detailed explanation of SOVA can be found in [6]. Once the LLR’s are obtained, the corresponding extrinsic information $\mathbf{l}^{(1)}_{extr}$ is used as *a priori* input to the second elementary decoder (DEC2). The extrinsic information $\mathbf{l}^{(2)}_{extr}$ associated with DEC2 (the SPC code decoder) can be computed according to [4]

$$l_i^{(2)} = (-1) \cdot (\min_{\substack{j=1,\ldots,n \\ j \neq i}} |L_j|) \cdot \prod_{\substack{j=1 \\ j \neq i}}^{n} \operatorname{sign}\left(L_j\right), \tag{8}$$

where $L_j = y_j + l_j^{(1)}$. According to (8) the magnitude of the extrinsic information for a particular code element is equal to the minimum magnitude of all of the other parity elements. The sign of the extrinsic information for a particular code element is equal to the sign of the element itself, if the parity of the overall equation is satisfied, and opposite to the sign of the element, if the overall parity fails. The extrinsic information $\mathbf{l}^{(2)}_{extr}$ is used as *a priori* information by the DEC1 during the next iteration as shown in Fig.1. After a predetermined number of iterations, the final estimate of the message bits $\hat{a}_i, i = 1,\ldots,M$ is found by hard-limiting the output of the DEC2:

$$\hat{a}_i = \begin{cases} 1 \; if \;\; L_i^2 \geq 0 \\ 0 \; if \;\; L_i^2 < 0 \end{cases}. \tag{9}$$

Some important notes are in order here. First, the turbo decoder operation could be improved by scaling the extrinsic information of both elementary decoders. In the present work the so-called improved SOVA will be employed in which the performance of the SOVA decoder is enhanced by scaling the extrinsic information with a factor of $c = 2\mu_s / \sigma_s^2$, where μ_s and σ_s^2 are the mean and variance of the absolute value of the SOVA output, respectively. Second, it is straightforward to decode the HTC with a variable number of iterations using a predetermined "early stopping" rule. A simple hard-decision stopping rule is to check whether identical tentative bit decisions are made at successive iterations or half-iterations. Another approach is based on comparing a metric on bit reliabilities (soft bit decisions) with a threshold. The hard-decision "early stopping" rule used in our simulations is as follows [5]: stop iterations of an *N*-bits data frame if both elementary decoders output identical sets of hard-limited extrinsic values at a given full iteration.

III. Performance Results

Performance of various HTC's with information frame sizes between 64 bits and 256 bits was studied through simulations according to the following setup. The information bits are obtained using uniformly distributed pseudorandom data. Maximum free distance memory $\nu = 2$ or $\nu = 3$, rate $R_c = 0.5$ convolutional codes are used as inner codes in the HTC scheme. The Rician fading is obtained according to (4). Iterative decoding with up to eight iterations and the above mentioned hard-decision stopping rule is used to decode the HTC schemes.

In Table I the required SNR for a BER of 10^{-4} is given. Fig.2 and Fig.3 shows simulation results of the considered HTC schemes for the case of 144 bits data frame along with performance of a reference scheme. The reference scheme, denoted as CTC, is a rate 1/2 convolutional turbo code with generators $g_0 = (07)_8$ and $g_1 = (05)_8$ (in octal notation). Iterative SOVA with up to eight iterations is used to decode the considered CTC. In Fig.2 and Fig.3 HTC1 denotes a HTC with a (13, 12) SPC outer code and a $\nu = 2$ inner convolutional code, and HTC2 denotes a HTC with a (13, 12) SPC outer code and a $\nu = 3$ inner convolutional code.

Table I

The required SNR for a $BER \approx 10^{-4}$ of HTC1 and HTC2

	HTC 1	HTC 2
K = 0 dB	7.3 dB	6.3 dB
K = 5 dB	5.7 dB	4.9 dB
K = 10 dB	4.3 dB	3.7 dB

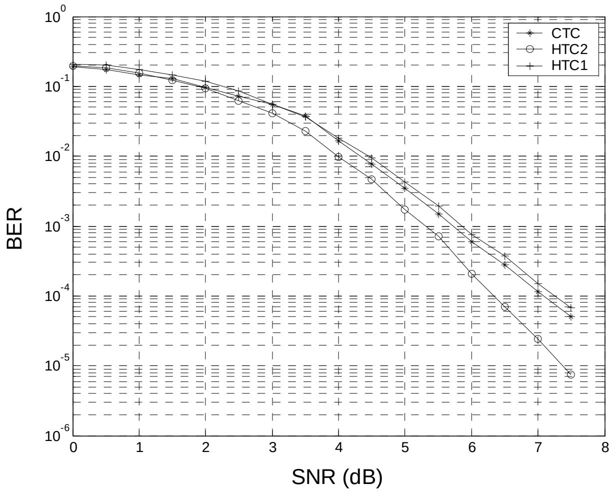

Fig.2. BER of SOVA-based turbo codes over a Rician fading channel with K = 0 dB

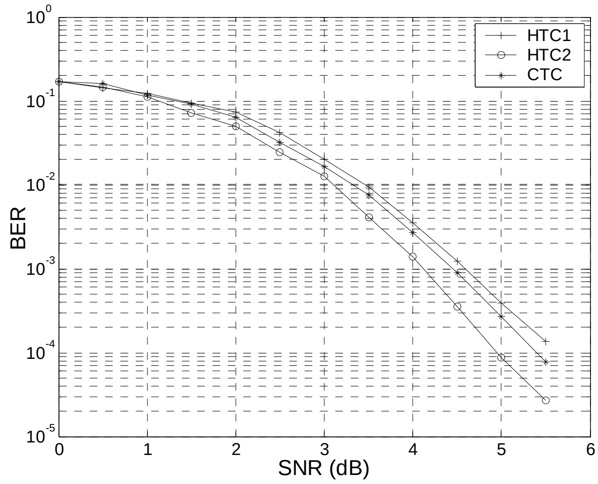

Fig.3. BER of SOVA-based turbo codes over a Rician fading channel with K = 5 dB

IV. Conclusion

In this paper, performance of hybrid convolutional/single parity check turbo codes over a Rician fading channel is studied. The results obtained through simulations indicate that the performance of these codes is quite exceptional given their decoding complexity. In fact, the performance results of HTC2 are slightly better to that of the considered CTC with SOVA decoding. Further, using the above described hard-decision autostopping rule results in a significant reduction of average number of iterations performed by the turbo decoder. Thus, it is possible to improve both the average decoding speed and power consumption of the turbo decoder. The considered hybrid turbo codes could be of interest for real-time short-frame communication services.

REFERENCES

[1] C. Berrou, A. Glavieux and P. Thitimasjshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", In Proceedings of IEEE International Conference on Communications, Geneva (Switzerland), pp. 1064-1070, 1993.

[2] C. Berrou, and A. Glavieux, "Near optimum error-correcting coding and decoding: Turbo-codes", IEEE Transactions on Communications, vol. 44, no. 10, pp. 1261–1272, 1996.

[3] S. Benedetto, and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes", IEEE Transactions on Information Theory, vol. 42, no. 2, pp. 409–428, 1996.

[4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes", IEEE Transactions on Information Theory, vol. 42, no. 2, pp. 429–445, 1996.

[5] N. Kostov, "Convolutional/single parity check turbo codes for multimedia communications", Radioengineering, vol.13, No 4, pp. 18-21, 2004.

[6] J. Hagenauer and P.Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications", In Proceedings of IEEE Global Communications Conference. Dallas (USA), pp. 47.1.1-47.1.7., 1989.

[7] FREEMAN, D., MICHELSON, A. A two-dimensional product code with robust soft-decision decoding. *IEEE Transactions on Communications*, 1996, vol. 44, no. 10, p. 1222-1226.

[8] N. Kostov, "Mobile Radio Channels Modeling in MATLAB", Radioengineering, Vol. 12, No 4, pp. 12-16, Dec. 2003.