Performance of SOVA-based Turbo Decoders with Autostopping

Nikolay T. Kostov[1], Slava M. Yordanova[2], Ginka K. Marinova[3] and Nikolay A. Velikov[4]

***Abstract* - A class of convolutional turbo codes with soft output Viterbi algorithm (SOVA) decoding is considered in this paper. Short-framed SOVA-based turbo decoders with autostopping on a Gaussian environment are studied. Bit error rate (BER), frame error rate (FER) and average number of decoding iterations versus the signal-to-noise ratio (SNR) are presented for the considered turbo decoders. The obtained results could be of interest regarding applications of these decoders in low-complexity, real-time digital communications services.**

***Keyword* – Turbo codes, soft output Viterbi algorithm (SOVA) decoding, simulations.**

I. INTRODUCTION

Turbo codes (TC) are a relatively new class of error control technique that was proposed by Berrou et. al. [1], [2]. They have been acknowledged as an extremely powerful coding scheme that achieves near-Shannon limit performance. The original rate $R_c = 0.5$ convolutional turbo code [1], [2] require a signal-to-noise ratio (SNR) of $E_b / N_0 = 0.7$ dB for bit error rate (BER) of 10^{-5} which is only 0.7 dB above the Shannon limit. However, the excellent performance of the original turbo code was achieved for a data frame size of 65536 bits and 18 decoding iterations per data frame. Because of prohibitively long latency involved with such large frame sizes, the original turbo code is not well suited for real-time multimedia communications.

There are many factors that affect the performance of TC. The most important parameters on TC's performance are the interleaver size and the decoding algorithm [1]-[3]. As the interleaver size increases, performance improves. For example, the original TC [1], [2] gains more than 1 dB at a bit error rate (BER) of 10^{-5} as the interleaver size increases from 1024 bits to 4096 bits. However, as the interleaver size increases so does the overall encoding/decoding latency. Soft output Viterbi algorithm (SOVA)-based decoders are attractive for practical implementations due to their low complexity and relatively high speed.

[1]Nikolay T. Kostov is with the Technical University of Varna, Department of Radio engineering, Studentska Street 1, Varna 9010, Bulgaria, E-mail: n_kostov@mail.bg

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

[3]Ginka K. 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

[4]Nikolay A. Velikov is with the Naval Academy "N. Vaptzarov" of Varna, Department of Electronics, Studentska Street 1, Varna 9010, Bulgaria, E-mail:na_velikov@mail.bg

However, there is a gap of 0.5 dB or even higher between their performance and that of maximum *a posteriori* (MAP)-based turbo decoders. Thus, TC's possess an inherent tradeoff between performance and latency as well as between performance and decoding complexity.

In the present paper short-framed SOVA-based turbo decoders with autostopping on a Gaussian environment are studied. The considered system model along with the obtained results is presented in the next sections.

II. SYSTEM MODEL

Let us now consider the encoding/decoding processes associated with the considered SOVA-based turbo decoders. The data frame to be transmitted is first encoded by a punctured TC using two identical recursive systematic convolutional codes. A uniform pseudorandom interleaver between convolutional codes is used. For an additive white Gaussian noise (AWGN) channel model the input/output relationship can be expressed as

$$r_i = \sqrt{E_s}\,(2b_i - 1) + n_i \,, \tag{1}$$

where r_i is the received symbol, E_s is the energy per code symbol, $(2b_i - 1)$ is the binary phase shift keying modulated code symbol and n_i is a zero-mean Gaussian variable with variance $\sigma^2 = N_0 / 2$.

The log-likelihood ratio (LLR) L_i at the output of a SOVA-based decoder can be represented as [3]

$$L_i = y_i + g_i + l_i \,, \tag{2}$$

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. Both of the two SOVA-based elementary decoders use SOVA to form an estimate of the LLR of each bit encoded by the TC. The essence of SOVA is finding the most likely transmitted sequence of bits along with reliability values for the bits [4], [5]. Let us define the likelihood ratio or "soft" value of the binary path decision at time *i* as

$$\Delta_i^0 = \frac{1}{2}\left(M_i^{m_0} - M_i^{\tilde{m}_0}\right), \tag{3}$$

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

$$L\left(\hat{b}_{i-\delta}\right) \approx \hat{b}_{i-\delta} \cdot \min_{l=0,\dots,\delta} \Delta_i^l . \quad (4)$$

For detailed explanation of SOVA see [4], [5]. The extrinsic information from the first elementary decoder is used as *a priori* information by the second elementary decoder during the next halfiteration and so on. After a predetermined maximum number of iterations, the final estimate of the message bits $\hat{a}_i, i=1,\dots,M$, is found by hard-limiting the output of one of the elementary decoders:

$$\hat{a}_i = \begin{cases} 1 \; if \;\; L_i^2 \ge 0 \\ 0 \; if \;\; L_i^2 < 0 \end{cases} . \quad (5)$$

It is possible to improve both the average decoding speed and power consumption of the turbo decoder if an “early stopping” rule is applied. Various “early stopping” rules for TCs was considered in [6] – [8]. Most of the presented results are for optimum MAP-based turbo decoders with large interleavers. In the present work we further study the performance of SOVA-based turbo decoders with small interleavers and simple “early stopping” rules.

III. The simulation code

Performance of two SOVA-based TCs with data frame sizes of N = 128 bits and N = 256 bits was studied through simulations in MATLAB. The first TC, denoted as TC1, uses convolutional codes with generators $g_0 = (07)_8$ and $g_1 = (05)_8$ (in octal notation) and the second TC, denoted as TC2, uses convolutional codes with generators $g_0 = (13)_8$ and $g_1 = (15)_8$. In fact, the considered TC2 is a punctured version of the Third Generation Partnership Project (3GPP) turbo code with overall rate $R_c = 0.5$. The information bits are obtained using uniformly distributed pseudorandom data. A uniform pseudorandom interleaver is used in both TC1 and TC2. The channel output is unquantized throughout and obtained according to (1). The turbo decoding is based on SOVA as described above. The SOVA decoding window was set to 24 bits. Soft iterative decoding with up to 12 iterations and a hard-decision or a soft-decision autostopping rule (e.g. “early stopping”) is used to decode the TC schemes. The hard-decision “early stopping” rule is as follows [9]: stop iterations if both elementary decoders output identical sets of hard-limited extrinsic values at a given full iteration, e.g. stop iterations if

$$\text{sign}(Le_{i,1}) = \text{sign}(Le_{i,2}), \quad \forall i, 1 \le i \le N \;, \quad (6)$$

where $\text{sign}(Le_{i,1})$ and $\text{sign}(Le_{i,2})$ are the *i*-th hard-limited extrinsic values at a given iteration of the first and second SOVA-based decoder, respectively. The soft-decision rule is based on comparing a metric on extrinsic values reliability (the minimum absolute value) with a threshold value θ, e.g. stop iterations if

$$\min_{1 \le i \le N} [|\, Le_{i,1} \,|, |\, Le_{i,2} \,|] > \theta \;. \quad (7)$$

The set of these threshold values as a function of the channel signal-to-noise ratio (SNR) is obtained through simulations and shown in Table I.

TABLE I
The minimum absolute values of extrinsic information for early stopping of TC1 and TC2

	N = 128 bits	N = 256 bits
SNR = 1 dB	min(|Le|) = 0.2	min(|Le|) = 0.25
SNR = 1.5 dB	min(|Le|) = 0.33	min(|Le|) = 0.5
SNR = 2 dB	min(|Le|) = 0.5	min(|Le|) = 1
SNR = 2.5 dB	min(|Le|) = 1	min(|Le|) = 1.5
SNR = 3 dB	min(|Le|) = 1.5	min(|Le|) = 2.5

The MATLAB simulation code of the considered turbo codes is given below.

```
clear; close all;
% initialization
rand('state', 123789);
randn('state', 432654);
EbN0db = 2.0;
Ntot = 256;
g = [1 1 1; 1 0 1]; [n,K] = size(g); m = K - 1; nstates = 2^m;
rate = 1/2;
nframe = 0;  frerr_max=50; niter_max = 12;
nferr = 0; final_err = 0; sum_iter = 0;
BER = 0; FER = 0; avr_iter = 0;
EbN0lin = 10^(EbN0db/10);   % convert Eb/N0 from db
Lc = 4* EbN0lin *rate;     % reliability value of the channel
sigma = 1/sqrt(2*rate* EbN0lin);
% simulation core
  while (nframe < frerr_max)
    nframe = nframe + 1;
    x = round(rand(1, Ntot-m));  % info bits
    [temp, gamma] = sort(rand(1,Ntot)); % interleaver
    enc_output = turbo_code(x, g, gamma); % encoder
    r = enc_output+sigma*randn(1,Ntot*2); % received bits
    y = demultiplex(r,gamma);
    rec = 0.5*Lc*y; % scaling the received bits
    Le = zeros(1,Ntot); % initialize the extrinsic information

    for iter = 1:niter_max
      % Decoder one
      La(gamma) = Le;  % a priori info
      Lall = sova_dec(rec(1,:), g, La, 1);
      Le = Lall - 2*rec(1,1:2:2*Ntot) - La;  % extrinsic info
      Le = Le*0.7; % scaling
      Le1 = Le(gamma);
      % Decoder two
      La = Le(gamma);  % a priori info
      Lall = sova_dec(rec(2,:), g, La, 2);
      Le = Lall - 2*rec(2,1:2:2*Ntot) - La;  % extrinsic info
      Le = Le*0.7; %scaling
      Le2 = (Le);
      % Estimate the info bits and errors
```

```
      a(gamma) = (sign(Lall)+1)/2;
      err = length(find(a(1:Ntot-m)~=x));
     % stop rule 1 – a "hard decision" rile
     % stop = nnz(sign(Le1)~=sign(Le2));
     % stop rule 2 – a "soft decision" rule
       stop1 = min(abs(Le1));  stop2 = min(abs(Le2));
       stop = min([stop1 stop2]);
       threshold = 1;

      if stop > threshold
            break
       end

   end  % iter

  if err > 0
      nferr = nferr + 1;
   end

   % BER
   final_err =  final_err + err;
   BER =   final_err/(nframe*(Ntot-m));

   % FER
   FER = nferr/nframe;

  % Average number of iterations
   sum_iter =  sum_iter + iter;
   avr_iter =  sum_iter/nframe;

 end   % while

% Save results
  save turbo_sim EbN0db Ntot g BER FER avr_iter
```

In the MATLAB code above the initialization stage includes definitions of the SNR in decibels (denoted as **EbN0db**), the interleaver size (denoted as **Ntot**), the convolutional code generators (denotes as **g**) etc. For a given SNR the simulation is executed by means of a **while-end** construction. The simulation runs until a predetermined number of frame errors (denoted as **frerr_max**) is encountered. The inner loop (e.g. the **for-end** construction) is used for iterative SOVA-based turbo decoding and runs until the "early stopping" rule is satisfied (thanks to the **if-break-end** construction) or the maximum number of decoding iterations (denoted as **niter_max**) is executed. Finally the bit error rate (BER), frame error rate (FER) and average number of decoding iterations are obtained.

IV. PERFORMANCE RESULTS

Performance results of the considered TC schemes are shown in Table II, Fig.1 end Fig.2. In Table II the required SNR for a BER of 10^{-4} is given. The BER and FER of the considered TC1 scheme are shown in Fig. 1 and the corresponding performance of the TC2 scheme is shown in Fig.2. A typical performance of the average number of decoding iterations versus the signal-to-noise ratio (SNR) for both the hard-decision and soft-decision autostopping rules applied in TC2 is shown in Fig.3 (for the case of 256 bits frame size). In Fig.3 the "soft decision+" denotes performance of the SOVA-based turbo decoding with a stopping threshold values 50% above the nominal values listed in Table I. There was no noticeable performance difference (in terms of BER and FER) between the considered hard-decision and soft-decision autostopping rules, so in Fig.1 and Fig.2 results only for the hard decision rule are presented.

TABLE II
The required SNR for a $BER \approx 10^{-4}$ of TC1 and TC2

	TC 1	TC 2
N = 128 bits	3.7 dB	3.4 dB
N = 256 bits	3.25 dB	3 dB

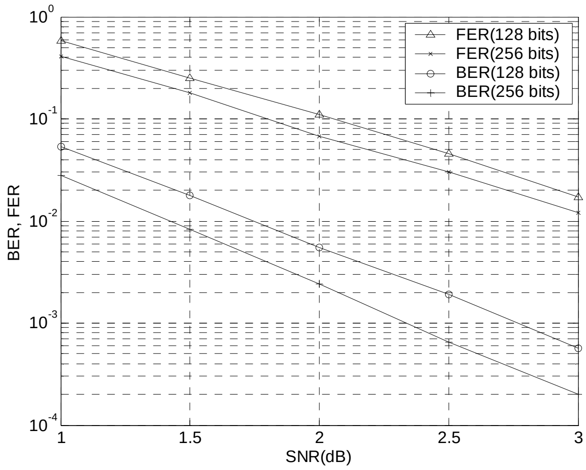

Fig.1. BER and FER of SOVA-based TC1 with 128 and 256 bits frame sizes.

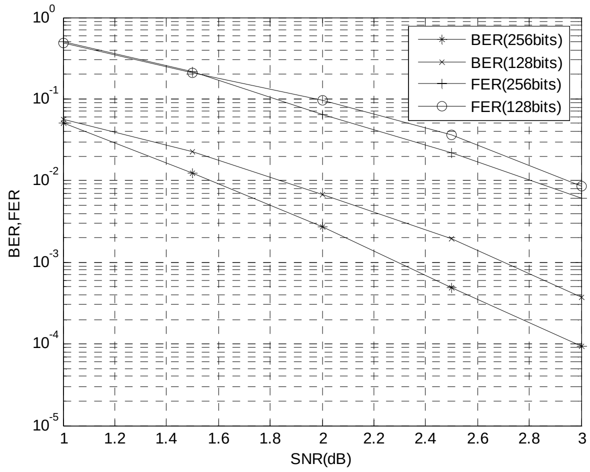

Fig.2. BER and FER of SOVA-based TC2 with 128 and 256 bits frame sizes.

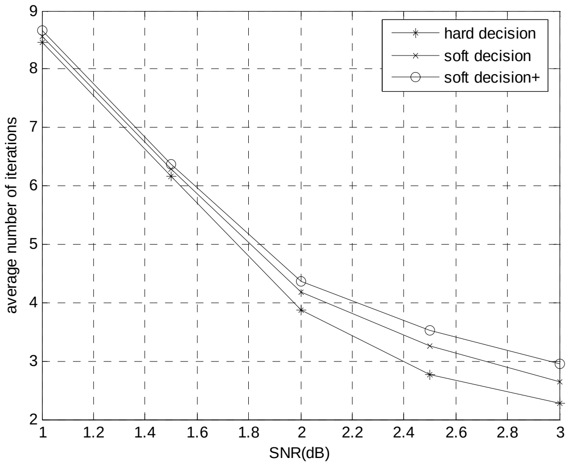

Fig.3. Average number of decoding iterations versus the SNR for TC2 with 256 bits frame size.

V. CONCLUSION

Short-framed SOVA-based turbo decoders with autostopping on a Gaussian environment are studied in this paper. Performance results in terms of BER, FER and average number of iterations versus the SNR are presented. As could be expected the performance of both TC1 and TC2 is dominated by the data frame size (e.g. by the interleaver size). The obtained results show no noticeable performance degradation of the considered "early stopped" SOVA-based decoders compared to the turbo decoding with a fixed number of iterations (at least for $\text{BER's} < 10^{-5}$). According to the Fig.3 both the hard-decision and soft-decision autostopping rules results in a significant reduction of average number of iterations performed of the turbo decoder. Thus, it is possible to improve both the average decoding speed and power consumption of the considered SOVA-based turbo decoders. The considered hard-decision autostopping rule should be preferred to soft-decision autostopping rule on non-stationary channels (like fading channels) because it is not SNR-dependent.

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] 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.

[6] A. Matache, S. Dolinar and and F. Pollara, "Stopping rules for turbo decoders", NASA TMO Progress report 42-142, pp. 1-22, August 15, 2000.

[7] R. Shao, S. Lin and M. Fossorier, " Two simple stopping criteria for turbo decoding", IEEE Transactions on Communications, vol. 47, pp. 1117–1120, 1999.

[8] A. Shibutani, H. Suda and F. Adachi, " Reducing average number of turbo decoding iterations", Electronics letters, vol. 35, pp. 701-702, 1999.

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