Turbo codes: matched MAP decoders

Zafir Popovski[1] , Tatjana Ulcar Stavrova[2]

Abstract – **In analysing wireless channels we most often assume that we exactly know the variance of the channel. However, in real applications this is usually not tha case. The variance needs to be derived from the received data and hence is only an approximation. In this paper we tackle the question on how will a non-accurate variance affect the performance of an iterative turbo decoder, test a variance estimation method and present the performances.**

Keywords – **Matched decoders, variance estimation.**

I INTRODUCTION

The turbo codes are a new class of FEC (forward error control) coding shemas that have been first intraduced by Berrou et al.[1] in 1993 and have received a lot of attention because they exhibit extremely good performance. For a bit error rate (BER) of 10^{-5}, the first TC was only 0.7 dB away from the theoretical Shannon's limit.

The turbo codes represent a parallel concatenation of two or more recursive systematic convolutional codes (RSCC), produced by a turbo encoder composed of two or more constituent RSC encoders (CEs) with input to each but one constituent encoder permuted by an interleaver of length N as shown on Fig.1.

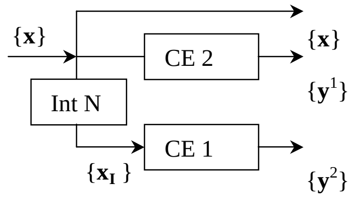

Fig.1. Turbo encoder structure

Such a composition allows for replacement of the optimal but rather complex maximum likelihood decoding algorithm by two (or more) relatively simple constituent decoders to decode corresponding constituent codes one by one.

A simple SISO (Soft-In Soft-Out) maximum a posteriori (MAP) decoder which minimise the probability of bit error appears to be a good solution for component decoders (CDs).

As an output, the MAP algorithm provide a real number which is a measure of probability of error in decoding a particular bit. Since both CDs decode the same information bit x_i, coded twice but in different order, it becomes possible this extra information, so-called extrinsic information, λ_i, to be passed as an input to the second CD allowing it to improve its own output extrinsic which will be passed to the first CD in the next iteration. The process is than iterated until reaching a satisfactory degree of confidence regarding the received noisy examples contained in length N sequence. The iterative TD shematicaly is presented on Fig.2.

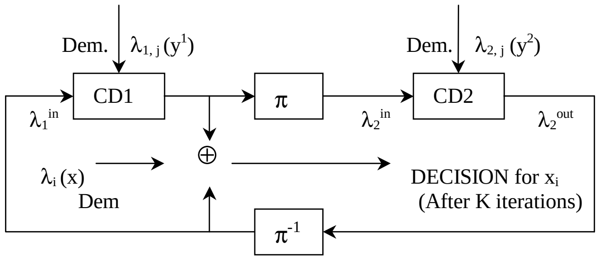

Fig.2 : Iterative TD with two MAP CDs

As the most promising one, the TC sheme is very likely to be implemented in various applications - one of which beeing the second generation of broadband satellite systems utilised on Ka band such as WEST from Matra Marconi Space, EuroSkyWay by Alenia Aerospazio, ASTROLINK by Locheed Martin, CYBERSTAR from Space Systems/Loral and SPACEWAY from Hughes Communications.

The system requirement constraints in such an application require to address the TC performances from several aspects.

First, the TC proposed by Berrou uses very large interleaving blocks (65536 bits) which would be very problematic for satellite applications since this would entail large delays and buffer sizes. The results from different researchers show that with an appropriate interleaving algorithm, good porformances can be achieved even with an interleaving length equal to the ATM-like cell. As an average of the known results, on Fig.3. we present the Eb/No required to achieve a BER of 10^{-3} for different values of the code memory and interleaving length.

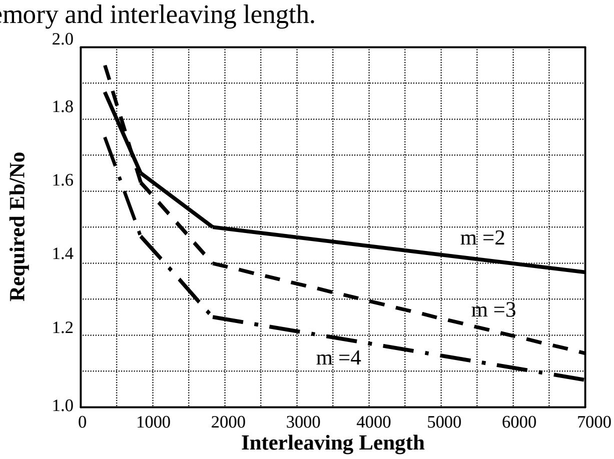

Fig.3. Eb/No to achieve a BER of 10^{-3}

[1] Zafir Popovski is with the Faculty of Electrical Engineering , "St.Cyril and Methodius" University - Skopje, E-mail : zpopovski@yahoo.com

[2] Tatjana Ulcar - Stavrova is with the Faculty of Electrical Engineering , "St.Cyril and Methodius" University - Skopje, E-mail : tanjaus@etf.ukim.edu.mk

As can be seen on Fig.3 on previous page, the performance degradation due to a reduction in the interleaving length from about 7000 to 1700 bits is lower than 0.2 dB. When the length is further reduced to about 440 bits (one ATM-like cell) the additional loss is limited to about 0.5 db which means that the TC can be successfully applied even when the interleaving length must be kept at very law value.

The second issue that needs to be addressed for the implementation of the TC is the decoding complexity which depends on both the SISO (Soft Input – Soft Output) algorithm used for the decoding of the constituent codes and the number of required iterations.

The modified *Maximum A Posteriori* (MAP) algorithm offers the best performance but is rather complex. A simplified version which does not show any performance degradation has been already proposed. Further simplification is achieved by performing the computations in the log domain and is reffered to as LogSMAP.

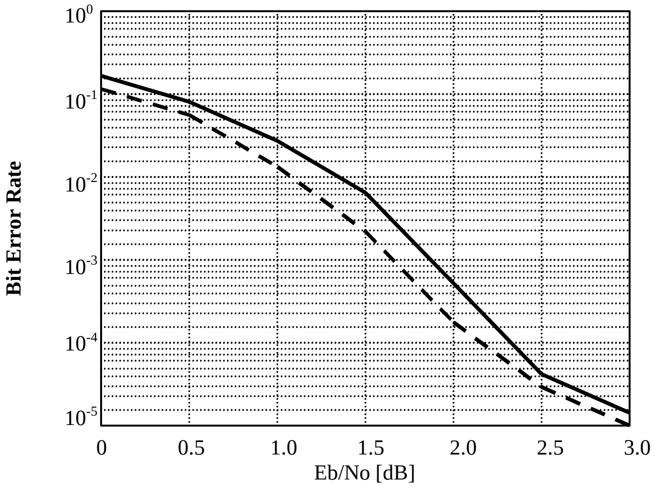

Fig.4. Log & Sub SMAP algorithm comparison

From the LogSMAP base, the SubSMAP algorithm has been developed which uses an approximation for the calculation of the logarithm of a sum of exponential terms.

Fig.4. presents the performance of the TC for different SISO decoding algorithms – LogSMAP and SubSMAP in this case, for the memory length of constituent codes equal to 2, interleaving lengths equal to 440 bits and number of iterations equal to 8. The dashed line corresponds to the performance of the LogSMAP while the plane line is associated with the SubSMAP algorithm. We can see that the performance degradation of the SubSMAP is limited to about 0.15 dB.

Better power efficiency could be achieved with constituent codes having a larger memory size. However, it was shown that increasing the memory size beyond 4 did not provide any performance gain. From the other side, increasing the memori size from 2 to 4 is very limited in view of the increased decoding complexity.

As far as the number of iterations is of concerne, this figure is set to 10 as standard. However, by implementing one of dynamic stopping rules the average number of iteration can be reduced to about from 6 to 8 iterations. Certainly, such a stopping rule should increase the decoding speed while not sacrifacing the BER performance. The soft stopping rule that is based on the evolution of the probability density function of the extrinsic information [5] shows a significant increase in the average decoding speed at almost no cost concerning the performance. The negligible computational requirements give this rule an advantage.

The speed and BER performances of this rule are shown on Figs.5 and 6, respectively.

From Fig.5 we can tell the speeds for different tresholds for the same rule to achieve the same BER.

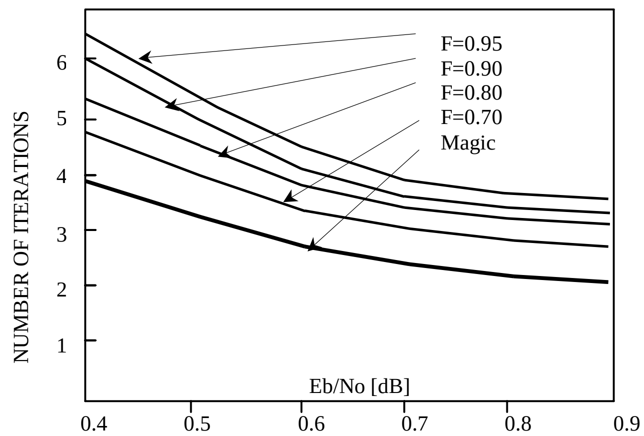

Fig.5. Average number of iterations

From the other side, Fig.6.indicates that the overall error rates are very close to each other and together to these for 10 fixed iterations. In fact they are nearly equal to the error rates achieved by a decoder using 20 iterations. In this case, 20 iterations are set as an endless loop protective treshold.

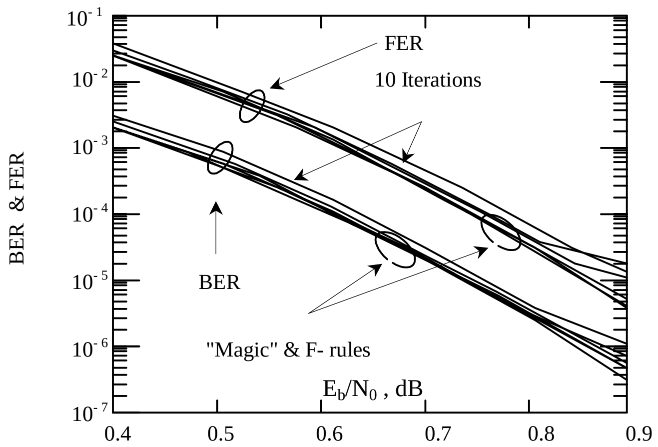

Fig.6. BER & FER with stopping rules

Finaly, the derivation of the exact a posteriori probabilities by the implemented decoding algorithm requires detailed knowledge of the channel. The problems arise when the channel varies with time requiring a continuous update of its parameters which may not be avaluable at all (some mobile applications). In practical systems, suboptimal metrics that take into account the range of channel variations are often used with the added benefit that they are often much simpler to implement than optimal metrics.

In this case, the resulting decoder is said to be mismatched with respect to the channel which results in performance reduction from that mismatch. It should be noted that asymptotically optimal decoders that operate without any knowledge of the channel statistics do exist. However, these asymptotically optimal decoders are often too complex to be used in practice.

Since the Ka band satellite channels (uplink, downlink and interlink) are proved to be slow-varying AWGN channels and that the estimates of the variance for such channels could be estimated ‘on-line’ thus resulting in a channel-matching decoder, we tried to implement a variance estimator which will be suitable in regard with the above mentioned system constraints.

In this paper we first evaluate the robustness of both the single constituent and turbo logRMAP decoders, then present some variance estimation methods and finaly implement one which we found suitable for logRMAP turbo decoder and present the result for 10 iterations performed by 4 memory length, 440 bits (ATM cell like) of interleaving length, rate 0.324 turbo code expresed by $(1,33/23,33/23)_8$

II VARIANCE SENSITIVITY

First we test the effect of variance mismatch on the performance of the constituent RSC (Recursive Systematic Convolutional) code $(1,33/23)_8$ since it is an integral part of the TC. There is a need to stres that we looked at the performance just for different values of Eb/No offset. We use the Eb/No because we can directly relate the results back to simulated BER results and actual variance since the Eb/No is inversely proportional to the variance of the channel. Note that the Eb/No of the channel can be converted to σ^2 (for an overall code rate of R) for a BPSK AWGN channel by using the following relationship :

$$\sigma^2 = (2REb/No)^{-1} \tag{1}$$

Since the En/No is inversely proportional to the variance σ^2, if the offset is negative then the variance used by the decoder is greater than that of the actual noise variance, i.e. $\sigma^2_{MAP} > \sigma^2_{noise}$, and, if the offset is positive then the variance of the decoder is less than that of actual noise : $\sigma^2_{MAP} < \sigma^2_{noise}$.

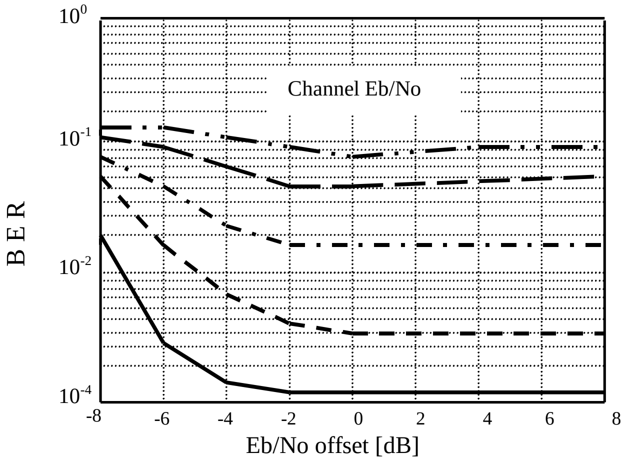

Fig.7. CC BER performance versus decoder offset

As we can see from the Fig.8 the RSC constituent performs quite well if the estimated variance is less than the actual noise variance or, in other words, when the decoder underestimate the actual condition of the channel. On the other hand the performance does not degrade until the Eb/No offset is at least 2 db under the actual Eb/No.

The robustness of the overall paralel composition is similar and is presented on fig.9.

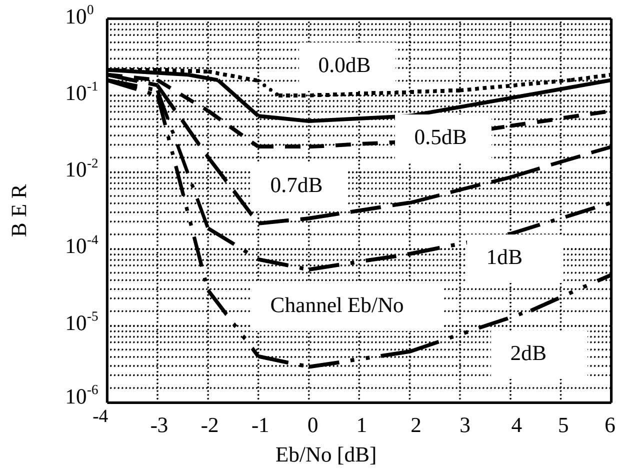

Fig.8. TC BER versus decoder Eb/No offset

We see that an overestimate of the noise power is much more problematic than an underestimate. We further see that the turbo decoder is quite robust, i.e. performance remains invariant, over an surprisingly wide range of mismatch values.

III VARIANCE ESTIMATION METHODS

Nonetheless, the issue of robustness is moot if the estimation of the channel parameters is included within the decoding process. We present here two different variance estimation methods which, although different, both are suitable for online estimation of the channel variance.

The Summer Variance Estimator estimates the variance directly from the magnitude of the received simbol r_n by finding the ratio between $E(r_n^2)$ and $E(|r_n|)^2$ which is a function of E_s/No. We have

$$E(r_n^2) = E_s + \sigma^2 \tag{2}$$

and

$$E(|r_n|)^2 = \sigma\sqrt{\frac{2}{\pi}}e^{-(E_s/2\sigma^2)} + \sqrt{E_s}\left[erf\left(\sqrt{\frac{E_s}{2\sigma^2}}\right)\right] \tag{3}$$

so, the ratio becomes :

$$\frac{E(r_n^2)}{[E(|r_n|)]^2} = \frac{1+\frac{E_s}{\sigma^2}}{\left(\sqrt{\frac{2}{\pi}}e^{-(E_s/2\sigma^2)} + \sqrt{\frac{E_s}{\sigma^2}}\left[erf\left(\sqrt{\frac{E_s}{2\sigma^2}}\right)\right]\right)^2}$$

$$= f(E_s/\sigma^2) \tag{4}$$

As we can see it is not easy to find the variance from the ratio by using Eq.(4). A practical approach is to find the ratio and its related channel variance σ^2 for the operational range of the decoder and express it as a look-up table. By using this look-up table we can find the variance of the channel by obtaining the ratio from the received data.

Another way as suggested in [4] is to find an n^{th} order polynomial that can approximate the above mentioned ratio for a fixed channel variance range that is related to the decoder's Eb/No operating range. For this purpose we can use the MatLab *polifit* function to find the polynomial that fits the function $f(x)$ in the least squares sense. The coefficients of the polynomial generated by MatLab are then used in the estimator to estimate the channel variance for the given $E(r_n^2)/E(|r_n|)^2$ ratio.

In our simulations, assuming an AWGN channel and BPSK signaling, we used the next estimate of the channel variance as developed in [2] :

$$\sigma^2 = [\frac{1}{k}\sum_{t=1}^{k} y_t^j]^2 + [\frac{1}{k}\sum_{t=1}^{k}(y_t^j - \frac{1}{k}\sum_{t=1}^{k} y_t^j)^2] - 1 \qquad (5)$$

where y_1^j, y_2^j,…, y_k^j, is a steam of soft values of the received sequence.

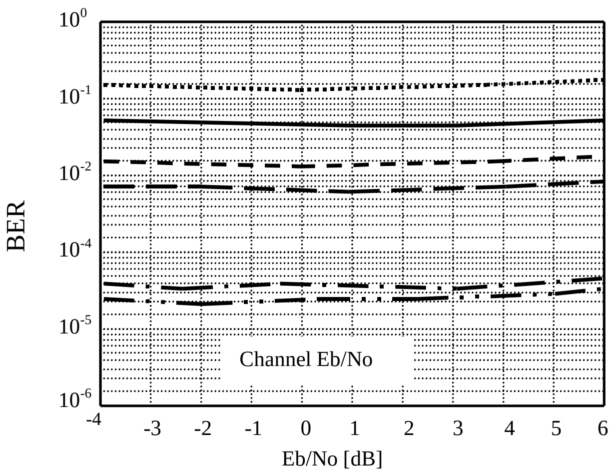

Fig.9. Performance of channel matching TC

By incorporating this estimate into the decoding process, as can be seen on Fig.9, any impact of initial error in estimating the channel variance is rapidly eliminated and the initial mismatch is no longer a factor in performance degradation.

IV Conclusion

We presented all the aspects which must be considered when the turbo codes are to be implemented into the newcoming generation of a global, Ka band satellite communications and showed that these codes are still powerfull even with small, ATM cell like interleaving length.

In this paper we emphasized the robustness of the turbo codes to the channel variations and also showed the results of a practical implemntation of a channel variance estimation model which is easy to implement and gaves very good results flatening the performance degradation due to the channel mismatching.

References

[1] C.Berrou, A.Glavieux and P.Thitimajshima,"Near Shannon Limit Error Correcting Coding: Turbo Codes", *Proceedings 1993 IEEE Int.Conf. on Comm.*,pp.1064 - 1070,Geneva, May 1993.

[2] S.Kim and S.B.Wicker, " On Mismatched and Self-Matching Turbo Decoding " - *Submitted to IEEE Transactions on Information Theory*, 1998.

[3] S.Kim, "Probabilistic Reasoning, Parameter Estimation and Issues in Turbo Decoding "-*PhD dessertation,* Cornell University, 1998.

[4] T.A.Summers and S.G.Wilson,"SNR mismatch and online estimation in turbo decoding ", *IEEE Transactions on Communications*, vol.46, pp 421-423, April 1998.

[5] Z.Popovski, T.U.Stavrova,"A New Stopping Rule for Turbo Decoders", *In Proc. XXXV International Conference on communications*, Dubrovnik 2003.

[6] M.Reed and J.Asenstorfer, " A novel variance estimator for turbo-code decoding", *in Proc. ICT'97, Melbourne, Australia, pp.173-178, April 1997.*