

Reduced Complexity DVB-RCS Turbo Decoder

Sherif Welsen Shaker

Kuang-Chi Institute of Advanced Technology
Nanshan District, Shenzhen, Guangdong, P.R. China

welsen@ieee.org

Abstract—Turbo codes are employed in many robust wireless digital communication systems. Those codes have been adopted for the satellite return channel in DVB-RCS standards. The implementation for such codes is complex and dissipates a large amount of power. For decoder hardware implementation, reduced complexity leads to low power consumption which is considered a key implementation issue. This paper investigates the decoder input quantization of low complexity decoding algorithm and proposes an algorithm for efficient decoder quantization to achieve significant improvement in the hardware implementation of the decoder architecture. Hence, leading to reduction of both area and power consumption in the whole radio receiver.

Keywords—DVB-RCS; Quantization; Turbo Decoder; Reduced Complexity.

I. INTRODUCTION

The Digital Video Broadcasting (DVB) project was founded in 1993 by the European Telecommunications Standards Institute (ETSI) with the goal of standardizing digital television services. Its initial standard for satellite delivery of digital television, named DVB-S, used a concatenation of an outer (204,188) byte shortened Reed Solomon code and an inner constraint length 7, variable rate (r ranges from $1/2$ to $7/8$) convolutional code [1].

The same infrastructure used to deliver television via satellite can also be used to deliver Internet and data services to the subscriber. Internet over DVB-S is a natural competitor against cable modem and DSL technology, and its universal coverage allows even the most remote areas to be served. Because DVB-S only provides a downlink, an uplink is also needed to enable interactive applications such as web browsing. The uplink and downlink need not be symmetric, since many Internet services require a faster downlink.

One alternative for the uplink is to use a telephone modem, but this does not allow for always-on service, has modest data rates, and can be costly in remote areas. A more attractive alternative is for the subscriber equipment to transmit an uplink signal back to the satellite over the same antenna used for receiving the downlink signal. However, given the small antenna aperture and requirement for a low-cost, low-power amplifier, there is very little margin on the uplink. Therefore, strong FEC coding is desired. For this reason, the DVB Project has adopted turbo codes for the satellite return channel in its DVB-RCS (Return Channel via Satellite) standard [2]. DVB-RCS turbo code was optimized for short frame sizes and high data rates. Twelve frame sizes are supported ranging from 12

bytes to 216 bytes, including 53 byte frame compatible with ATM and a 188 byte frame compatible with both MPEG-2 and the original DVB-S standard. The return link supports data rates from 144 kbps to 2 Mbps and is shared among terminals by using multi-frequency time-division multiple-access (MF-TDMA) and demand-assigned multiple-access (DAMA) techniques. Eight code rates are supported, ranging from $1/3$ to $6/7$. The outstanding coding performance of those codes requires the investigation of hardware implementation issues. For portable radio, low power consumption is a key implementation issue. Decoding algorithm simplification and quantization leads to reduction of power consumption in the radio receiver.

This paper applies a simplified decoding algorithm for DVB-RCS turbo codes and investigates the quantization effect on the decoder performance. The paper proposes an algorithm that can be applied on different DVB-RCS code rates and block sizes in order to achieve a reduced complexity decoder.

II. DVB-RCS TURBO CODES

DVB-RCS turbo code uses a pair of constituent Recursive Systematic Convolutional (RSC) encoders along with Log-MAP or Max-log-MAP decoding [3]. The decoder for each constituent code performs best if the encoder begins and ends in a known state, such as the all-zeros state. This can be accomplished by independently terminating the trellis of each encoder with a tail which forces the encoder back to the all-zeros state. However, forcing the encoder to a known state at the end of the encoding stage by adding tail bits, presents two major drawbacks. First, the minimum free distance d_{free} is no longer equal to the original minimum free distance for all information data. Second, the spectral efficiency of the transmission is degraded especially for small frame lengths supported by DVB-RCS [4]. As an alternative to terminating the trellis of the code, DVB-RCS uses Circular Recursive Systematic Convolutional (CRSC) encoding [5]. CRSC codes make it possible to start the encoding at known state, which is called the circular state S_c , and end the encoding in the same state without the two drawbacks mentioned above.

Unlike most of turbo codes, DVB-RCS turbo code uses duobinary constituent encoders defined over GF(4) instead of using binary encoders defined over GF(2). Compared with classical turbo codes, which use RSC single binary codes, the double binary turbo codes have several benefits [6]: (a) The performance is improved by reducing the correlation effects between the component decoders. (b) The minimum free distance is increased by introducing periodic disorder in the

symbols. (c) Puncturing can be used to increase the code rate and data rate, because duobinary codes are less sensitive to puncturing than the single binary codes. (d) The trellis contains half as many states as a binary code of identical constraint length (but the same number of edges), and therefore needs half as much memory and the decoding hardware can be clocked at half the rate as a binary code. (e) The duobinary code can be decoded with the suboptimal but efficient Max-log-MAP algorithm at a cost of only about 0.1-0.2 dB relative to the optimal log-MAP algorithm. This is in contrast with binary codes, which lose about 0.3-0.4 dB when decoded with the max-log-MAP algorithm [7]. Additionally, duobinary codes are less impacted by the uncertainty of the starting and ending states when using tail biting and perform better than their binary counterparts when punctured to higher rates.

A. DVB-RCS Encoder Structure

Figure 1 shows the block diagram of the turbo encoder that is used by DVB-RCS, while the CRSC constituent encoder is shown in Figure 2 as described in the standardized DVB-RCS. The encoder is fed blocks of k message bits which are grouped into $N = k/2$ couples. The number of couples per block can be $N \in \{48, 64, 212, 220, 228, 424, 432, 440, 752, 848, 856, 864\}$. The number of bytes per block is $N/4$. In Figure 2, A represents the first bit of the couple, and B represents the second bit. The two parity bits are denoted W and Y [8].

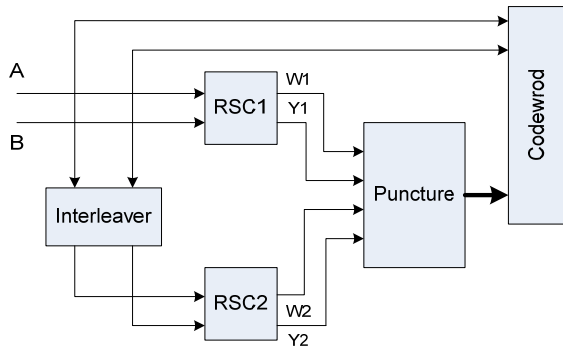


Figure 1. Block diagram of DVB-RCS Turbo encoder

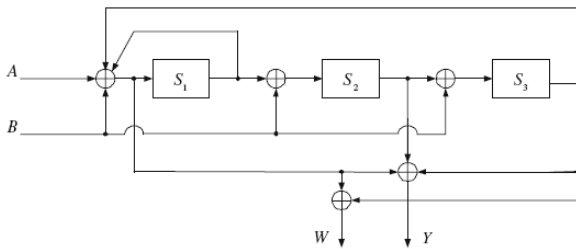


Figure 2. Duobinary CRSC constituent encoder used by DVB-RCS

Because of the tail biting nature of the code, the block must be encoded twice by each constituent encoder. During the first pass at encoding, the encoder is initialized to the all-zeros state, $S_0 = [0 \ 0 \ 0]$. After the block is encoded, the final state of the encoder S_N is used to derive the circulation state. The circulation state S_c is given by:

$$S_c = (I + G^N)^{-1} S_N \quad (1)$$

where

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (2)$$

In practice, the circulation state S_c can be found from S_N by using a lookup table [2]. Once the circulation state is found, the data is encoded again. This time, the encoder is set to start in state S_c and will be guaranteed to also end in state S_c .

B. DVB-RCS Turbo Decoder

Decoding of the DVB-RCS code is complicated by the fact that the constituent codes are duobinary and circular. As with conventional turbo codes, decoding involves the iterative exchange of extrinsic information between the two component decoders. While decoding can be performed in the probability domain, the log-domain is preferred since the low complexity max-log-MAP algorithm can then be applied [3]. Unlike the decoder for a binary turbo code, which can represent each binary symbol as a single log-likelihood ratio, the decoder for a duobinary code requires *three* log-likelihood ratios. For example, the likelihood ratios for message couple (A_k, B_k) can be represented in the form:

$$\Lambda_{a,b}(A_k, B_k) = \log \frac{P(A_k = a, B_k = b)}{P(A_k = 0, B_k = 0)} \quad (3)$$

where (a, b) can be $(0, 1)$, $(1, 0)$, or $(1, 1)$.

An iterative decoder that can be used to decode the DVB-RCS turbo code is shown in Figure 3. $\{\Lambda_{a,b}^{(i)}(A_k, B_k)\}$ denotes the set of LLRs corresponding to the message couple at the input of the decoder and $\{\Lambda_{a,b}^{(o)}(A_k, B_k)\}$ is the set of LLRs at the output of the decoder. Each decoder is provided with $\{\Lambda_{a,b}^{(i)}(A_k, B_k)\}$ along with the received values of the parity bits generated by the corresponding encoder (in LLR form). Using these inputs and the knowledge of the code constraints, the decoder can produce the updated LLRs $\{\Lambda_{a,b}^{(o)}(A_k, B_k)\}$ at its output. As with binary turbo codes, extrinsic information is passed to the other constituent decoder instead of the raw LLRs. This prevents the positive feedback of previously resolved information. Extrinsic information is found by simply subtracting the appropriate input LLR from each output LLR, as indicated in Figure 3.

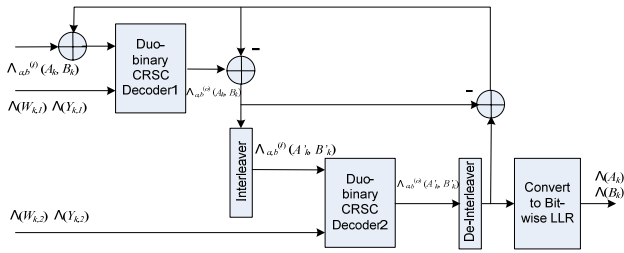


Figure 3. A decoder for the DVB-RCS code

The extension of the log-MAP and max-log-MAP algorithms [3] to the duobinary case is fairly straightforward. Each branch must be labeled with the log-likelihood ratios corresponding to the systematic and parity couples associated with that branch. Because QPSK modulation is orthogonal, the LLR of message couple (A, B) can be initialized prior to being fed into the first decoder as $\Lambda_{a,b}^{(i)}(A_k, B_k) = a \Lambda(A_k) + b \Lambda(B_k)$, where $\Lambda(C) = \log[P(C = 1)/P(C = 0)]$. Because extrinsic information about the parity bits is not exchanged, the parity bits can always be decomposed in a similar manner. However, for the systematic bits, the three likelihood ratios defined in (3) must be calculated during each iteration and exchanged between the decoders.

III. DVB-RCS FINITE PRECISION TURBO DECODER

From the hardware point of view, efficient implementation of DVB-RCS turbo decoder means to reach the best performance in terms of area, speed, and low power consumption without loss of error correction capability. In the design phase of turbo codes, good results can be achieved through floating-point software simulations. The decoding of Turbo codes is an iterative process and each iteration needs a lot of processing and calculations. This adds challenge for hardware implementation including huge processing and storage requirements. In order to achieve the most efficient implementation, there is always a tradeoff between hardware complexity and decoder capability. Some design modifications have to be done at different abstraction levels including algorithmic, architecture, and circuit levels. For this reason, floating-point representation is not practical during hardware implementation phase. Instead, fixed point representation might be realizable and preferable, and circuits then can process the data in finite precision. The primary goal when implementing quantized decoder is to find a fixed point model that has all bit-widths as small as possible under the condition of an acceptable degradation in coding performance. The smaller the bit-width of quantization, the better performance of the decoder in terms of speed, area and power consumption. Decoding performance is also affected, so optimized quantization is a key issue for the implementation complexity.

At the algorithmic level also, a lot of modifications have been done on decoding algorithms [9]. Additions instead of multiplications can be used in calculating the extrinsic metric of Log-MAP decoding algorithm, avoiding the numerical problems of MAP decoding algorithm. Also, some term can be avoided like in Max-Log-MAP decoding algorithm.

IV. SIMULATIONS AND RESULTS

A Matlab code is first driven to evaluate the performance of the floating-point simulation of DVB-RCS turbo coding on AWGN channel for all the standardized block lengths ranging from 48 to 864 message couples. Figure 4 shows the influence of the block size on the BER curve at code rate 1/3 for ten iterations and Figure 5 and Figure 6 show the influence of the code rate on the BER curve for two different frame lengths of 48, 212 message couples. Results are shown for all seven code rates when the block sizes are $N = 24$ and 212 message couples, respectively, and ten iterations of Max-Log-MAP decoding are performed.

Another code is then driven to evaluate the performance of the quantized decoder compared with the floating point representation. The simulation was run on different ranges of 4-bit quantized decoder's LLR input for different coding rates. Results showed that for 4-bit quantization, the performance is getting worst as the range is increased. The maximum and minimum of input LLR values are observed at different points of SNR, and it was found that they are linearly increasing with the SNR.

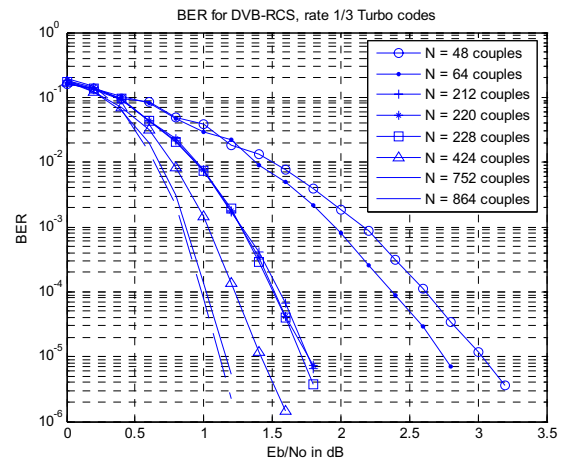


Figure 4. Influence of block size on the BER performance.

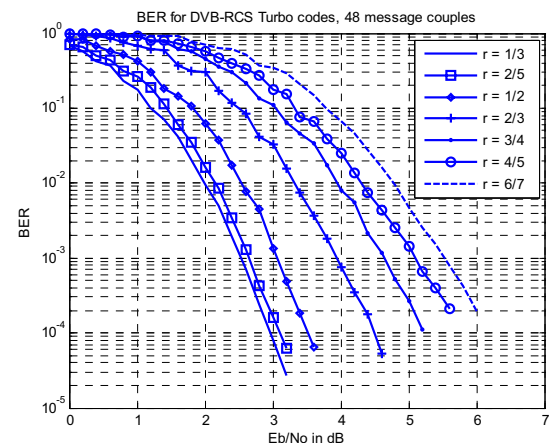


Figure 5. Influence of code rate on the BER performance of the DVB-RCS turbo code, 48 message couples.

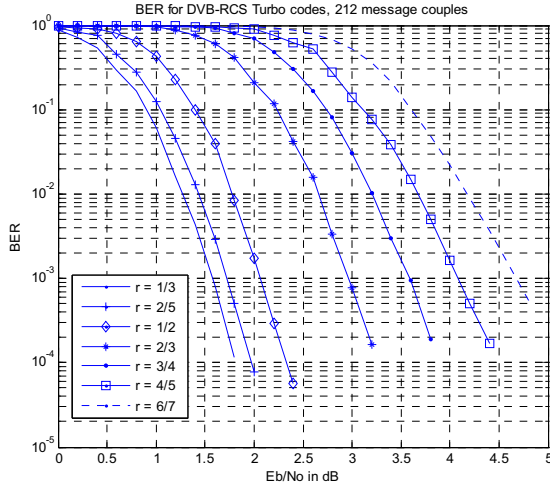


Figure 6. Influence of code rate on the BER performance of the DVB-RCS turbo code, 212 message couples.

This helped in the estimation of the quantization range based on the SNR values, by introducing SNR-dependent variable factor and multiplying that factor with the LLR input values before the decoding process. The multiplier factor should be smaller as the SNR increases. The decoding process is then applied on the input LLR values after the multiplying process.

Another technique that has been tried to estimate the appropriate quantization range by introducing a scaling factor in the decoding process. The input LLR values fed to the decoding routine are scaled by multiplying them with that scaling factor. The $\max^*$ operator within the Log-MAP decoding scheme is then computed by introducing a constant correction factor. It should be noticed that the computation of the $\max^*$ operator in the decoding algorithm constitute a significant portion of the decoding complexity. The estimate of the correction factor can be given from [10]. The computation of the $\max^*$ operator considering that correction factor is given by:

$$\max^*(x, y) = \max(x, y) + f_c(|y-x|) \quad (4)$$

where x and y , are the LLR input values, and $f_c(|y-x|)$, is the correction factor. For Log-MAP algorithm, the $\max^*$ operator can be computed as:

$$\max^*(x, y) \approx \max(x, y) + \begin{cases} 0, & \text{if } |y-x| > T \\ C, & \text{if } |y-x| \leq T \end{cases} \quad (5)$$

where C is the correction factor and T is a threshold value.

Another Matlab code was built to measure the performance of the decoding algorithm with constant correction factor for the scaled LLR input values to study the effect of the scaling. It was observed that a better performance for the decoding process can be achieved when multiplying the correction factor

with the same scaling factor in the decoding process. Accordingly, we propose a change in the approximation of the $\max^*$ operator computed by (4), to become as follows:

$$\max^*(x, y) = \max(x, y) + K(|y-x|) \quad (6)$$

where, $K = C \cdot f_c$.

The same operation has been done but this time on different code rate. Figure 7 shows the performance of the quantized 212 blocks of rate 6/7 compared with the floating-point BER. It was found that as SNR increases, the performance of the quantized code worsen. It has also been observed from the calculations that the input LLR values for the code rate 6/7 are higher than those of rate 1/3. This is an indication that varying the input LLR range as a function of SNR gives good results, but the best performance could be obtained when the code rate is also considered. Therefore, the quantization range should be scaled by a factor including both the SNR and code rate. In the same time, fixed quantization range couldn't give the best performance for all code rates.

From the hardware implementation point of view, there is a tradeoff between the performance and resolution. For a given code rate, smaller decoder input range affects the decoder performance, while higher range affects the resolution. Since the decoder performance of the quantized code is affected by both of the code rates and the value of SNR, Figure 8 shows a flow-chart of our proposed algorithm for efficient decoder quantization.

Figure 9 shows a performance comparison between variable-rate floating-point codes and quantized version of those codes for blocks of 212 message couples based on our proposed algorithm. It can be noticed that there is a very little difference in the performance between the floating-point codes and the quantized codes.

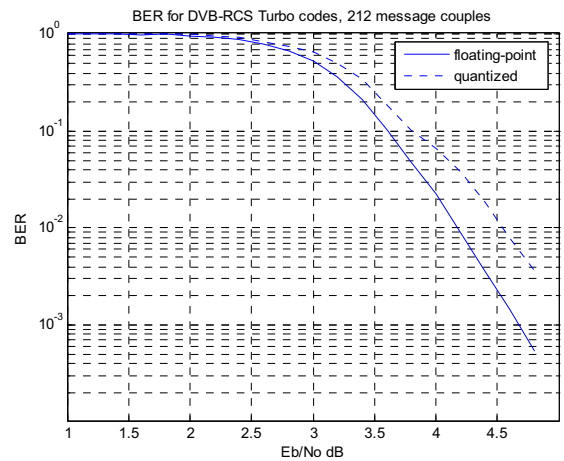


Figure 7. Floating-point vs. quantized decoder at rate 6/7

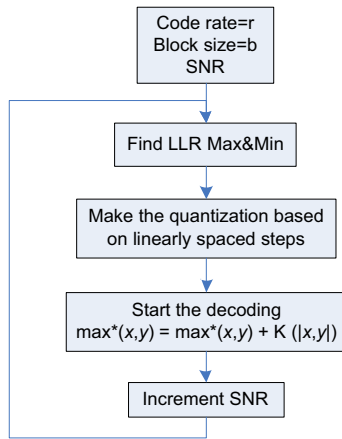


Figure 8. Proposed algorithm for efficient decoder quantization

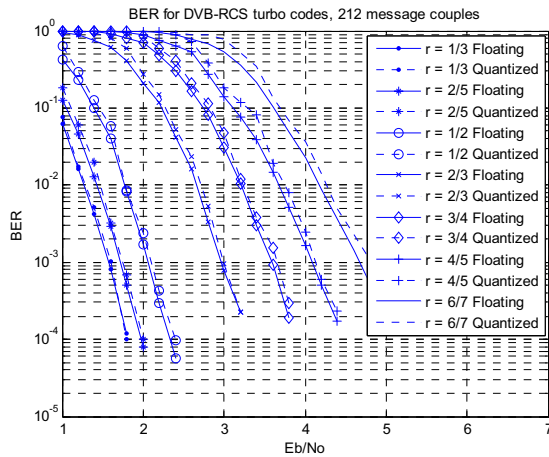


Figure 9. Influence of quantization on variable rate DVB-RCS Turbo codes

V. CONCLUSION

In this paper, an efficient decoder quantization for DVB-RCS turbo coding has been made that reduces the overall

power consumption by reducing the decoder complexity while realization of DVB-RCS radios. An algorithm that computes the quantization range for an efficient decoder quantization has been proposed. Performance simulation for varying of different supported code rates for such codes has been presented, and results showed that the quantized decoder essentially matches the performance of the floating point decoder.

ACKNOWLEDGMENT

This research was supported by grants from Shenzhen Municipal Government of China (No. KQE201106020031A, No. CXB201109210099A).

REFERENCES

- [1] *Digital broadcasting system for television, sound, and data services*, European Telecommunications Standards Institute, ETSI 200 421, 1994.
- [2] *Digital video broadcasting (DVB); interaction channel for satellite distribution systems*, European Telecommunications Standards Institute. ETSI EN 301, 790 V1.5.1, May 2009.
- [3] P. Robertson, P. Hoeher, and E. Villebrun. "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. On Telecommun.*, 8(2):119–125, Mar./Apr. 1997.
- [4] C. Douillard, M. Jezequel, and C. Berrou, "The turbo code standard for DVB-RCS," *Proc. 2nd Int. Symp. Turbo codes*, pp. 551–554, Sept. 2000.
- [5] C. Berrou, C. Douillard, and M. Jezequel, "Multiple parallel concatenation of circular recursive systematic convolutional (CRSC) codes," *Annals of Telecommunication*, 54(3–4):166–172, Mar.-Apr. 1999.
- [6] C. Berrou, M. Jezequel, and C. Douillard, "The advantages of non-binary turbo codes," *Proc. IEEE Infor. Theory Workshop*, pp. 61–63, Sept. 2001.
- [7] M. C. Valenti and J. Sun. "The UMTS turbo code and an efficient decoder implementation suitable for software defined radios," *Int. J. Wireless Info. Networks*, 8:203–216, Oct. 2001.
- [8] S.W. Shaker, S.H. Elramly, and K.A. Shehata, "Design and Implementation of Low-Power Turbo Encoder for DVB-RCS Software Radio," *Proc. 17th Telecommunications Forum TELFOR*, Belgrade, Serbia, pp. 464–467, November 2009.
- [9] Y. Ould-Cheikh-Mahmedou, P. Guinand, and P. Kabal, "Enhanced Max-Log-MAP and Enhanced Lop-APP Decoding for DVB-RCS," *Proc. Int. Symp. Turbo Codes*, Brest, France, pp. 259–262, Sept. 2003.
- [10] M. C. Valenti and J. Sun. "The UMTS turbo code and an efficient decoder implementation suitable for software defined radios," *Int. J. Wireless Info. Networks*, 8:203–216, Oct. 2001.