DVB-RCS: Efficiently Quantized Turbo Decoder

Sherif Welsen Shaker

University of Nottingham, Ningbo Campus, P.R. China

welsen@ieee.org

***Abstract*—Turbo codes have been incorporated into many important wireless communication standards including the satellite return channel in DVB-RCS standards. According to their iterative nature, the computational complexity of turbo decoder is much higher than that of convolutional FEC decoders. From the hardware implementation point of view, the complexity can be reduced by using quantized decoder. For DVB-RCS turbo coding, there are many block sizes and different code rates. In order to realize the DVB-RCS turbo decoder efficiently, an algorithm should be developed for the best computation of quantization range for each code rate and at different signal-to-noise ratios. This paper investigates the decoder input quantization of low complexity decoding algorithm and proposes an algorithm for efficient decoder quantization by introducing a scaling factor into the decoding algorithm, aiming to achieve significant improvement in the hardware implementation of the decoder architecture.**

***Index Terms*— DVB-RCS; Max-log-MAP; Quantization; Turbo Decoder; Reduced Complexity**

I. INTRODUCTION

FOUNDED by the European Telecommunications Standards Institute (ETSI) in 1993, the Digital Video Broadcasting (DVB) project intended to standardize the digital television services. DVB-S was the initial standard of digital television with satellite delivery, that 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]. DVB-S was a widely accepted standard in the forward link of broadband satellite communications. The second generation DVB-S2 includes the transmission of multimedia contents and a variety of uni-cast and multicast services. 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.

Manuscript received July, 27 2013.

S.W. Shaker, is with EEE Dpt., Faculty of Science and Engineering, University of Nottingham Ningbo, P. R. China (Tel: +86 147 149 175 34; e-mail: welsen@ieee.org).

Transmitting an uplink signal back to the satellite over the same antenna used for receiving the downlink signal, rather than using a telephone modem, became an attractive alternative for the subscriber equipment. . DVB-RCS standards have been approved for Return Channel via Satellite; it provides two-way, full IP, asymmetric communications via satellite. In this way not only the service can be quickly deployed, but the cost of the service and the quality are independent of the distance between the terminal and access point. This makes the service provided via the satellite a strong competitor in those cases where cable modems are not economically possible. 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. Turbo codes have shown great performance among forward error correction (FEC) codes; they have been used by many standards like Wideband CDMA, and Third Generation Partnership Project (3GPP) for IMT-2000. For its major advancement in channel coding area, convolutional turbo code are well-suited for mobile satellite broadcasting applications and it has been chosen for DVB-RCS standards [2]. The big advantage of turbo codes is their data transmission reliability within a half decibel of Shannon Limit.

DVB-RCS standards are open to provide interactive broadband access over satellite. It allows a central gateway or hub to broadcast IP date on the forward link in the DVB/MPEG2 format to large number of small terminals with date rates up to 48 Mbit/s. Satellite terminals can send return signals to the hub on the forward link. 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. DVB-RCS turbo code was optimized for short frame sizes and high data rates; it supports seven code rates, 1/3, 2/5, 1/2, 2/3, 3/4, 4/5, and 6/7. The outstanding coding performance of those codes requires the investigation of hardware implementation issues. For portable radio terminal, low power consumption is a key implementation issue. Decoding algorithm simplification and quantization are very important issues leading to reduction of power consumption. In the past, several algorithms have been used in order to simplify the decoding process of turbo codes.

The penalty paid for those algorithms which aim to reduce the complexity is small error rate performance degradation as compared to the performance achieved in case of using the optimal algorithm. An additional correction term is required to be added in order to minimize the performance degradation.

In this paper, our intention is to apply a simplified decoding algorithm for DVB-RCS turbo codes. It is our objective to investigate the impact of the decoder quantization on the requirement of the decoder performance. The paper proposes an algorithm for efficient decoder quantization that can be applied on different DVB-RCS code rates and block sizes in order to achieve a reduced decoder complexity. The structure of DVB-RCS turbo encoder followed by a brief review of turbo decoding algorithms is introduced, for interested reader, in section II. The structure of DVB-RCS turbo decoder is then highlighted for the Duobinary case. The quantized decoder is discussed in section III. In section IV, simulation results are presented for different frame lengths and code rates of DVB-RCS turbo codes. The effect of quantized decoder is then investigated, and the algorithm for efficient decoding quantization is proposed and tested.

II. DVB-RCS TURBO CODES

DVB-RCS turbo encoder is composed of two identical Recursive Systematic Convolutional (RSC) encoders along with Log-MAP or Max-log-MAP decoding [3]. If the encoder begins and ends at a known state, such as the all-zeros state, the decoder for each constituent code performs better. One alternative to do this is 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 dfree is no longer equal to the original minimum free distance for all information data. Second, the spectral efficiency of the transmission is degraded specially for small frame lengths supported by DVB-RCS [4]. The other alternative to terminate the trellis of the code is done by using Circular Recursive Systematic Convolutional (CRSC) encoding [5]. CRSC start the encoding at the circular state Sc, and end the encoding in the same state without the aforementioned drawbacks.

DVB-RCS turbo code uses duobinary constituent encoders defined over GF(4) instead of using binary encoders defined over GF(2). The double binary turbo codes have several benefits compared with classical turbo codes, which use RSC single binary codes, [6]: (a) Reducing the correlation effects between the component decoders, improves the performance. (b) Introducing periodic disorder in the symbols increases the minimum free distance. (c) Duobinary codes are less sensitive to puncturing than the single binary codes; hence puncturing can be used to increase the code rate and data rate. (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) 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 can be used to decode the duobinary code; 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

The block diagram of the turbo encoder that is used by DVB-RCS is shown in Figure 1, and 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 ∈ {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].

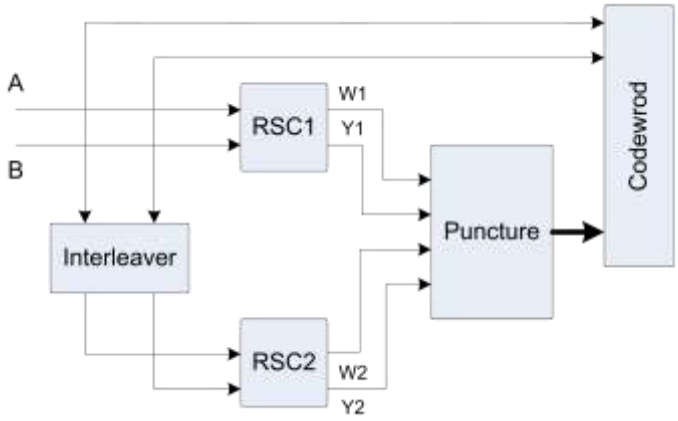

Fig. 1. Block diagram of DVB-RCS Turbo encoder.

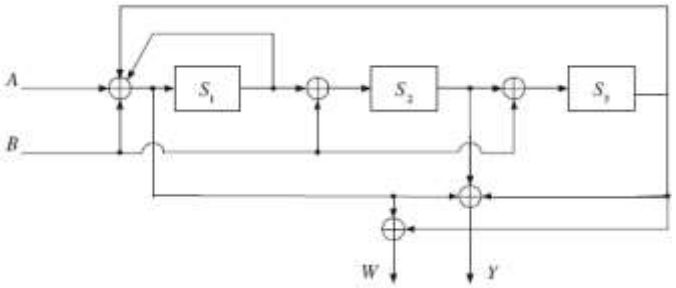

Fig. 2. Duobinary CRSC constituent encoder used by DVB-RCS.

The block must be encoded twice by each constituent encoder because of the tail biting nature of the code. First, the encoder is initialized to the all-zeros state, $\mathbf{S}_0 = [0\ 0\ 0]$. After the block is encoded, the final state of the encoder $\mathbf{S}_N$ is used to derive the circulation state. The circulation state $\mathbf{S}_c$ is given by:

$$\mathbf{S}_c = (\mathbf{I} + \mathbf{G}^N)^{-1}\mathbf{S}_N \quad (1)$$

where

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

In practice, the circulation state $\mathbf{S}_c$ can be found from $\mathbf{S}_N$ 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 $\mathbf{S}_c$ and will be guaranteed to also end in state $\mathbf{S}_c$.

B. Turbo Decoding Algorithms

In a typical turbo decoding system, two decoders operate iteratively and pass their decisions to each other after each iteration. These decoders should produce soft-outputs to improve the decoding performance. Such a decoder is called a soft-input soft-output (SISO) decoder [9]. Each decoder operates not only on its own input, but also on the other decoder's incompletely decoded output, which resembles the operation principle of turbo engines. Generally, we assume that the encoded information sequence, X_k, is transmitted over an additive white gaussian noise (AWGN) channel, and a noisy received sequence, Y_k, is obtained. In binary case, each decoder calculates the log-likelihood ratio (LLR) for the k^{th} data bit d_k, as follows:

$$L(d_k) = \log\left[\frac{P(d_k = 1 \mid Y)}{P(d_k = 0 \mid Y)}\right] \tag{3}$$

The LLR can be decomposed into three independent terms as:

$$L(d_k) = L_{apri}(d_k) + L_c(d_k) + L_e(d_k) \tag{4}$$

where $L_{apri}(d_k)$ is the a priori information of d_k, $L_c(d_k)$ is the channel measurement, and $L_e(d_k)$ is the extrinsic information exchanged between the constituent decoders. LLR computations can be performed by using one of the two main turbo decoding algorithms: Soft Output Viterbi Algorithm (SOVA) [10] and Maximum A posteriori Probability (MAP) [3]. The difference between the two algorithms is that, MAP algorithm seeks for the most likely data sequence, while SOVA seeks for the most likely connected trellis path. MAP algorithm is superior to SOVA specially at low SNR at the expense of implementation complexity.

1) MAP Algorithm

MAP is the optimal but computationally complex algorithm. According to this algorithm, LLR values for each information bit can be calculated as:

$$L(d_k) = \ln\left[\frac{\sum_{S_k}\sum_{S_{k-1}} \gamma_1(S_{k-1}, S_k)\alpha(S_{k-1})\beta(S_k)}{\sum_{S_k}\sum_{S_{k-1}} \gamma_0(S_{k-1}, S_k)\alpha(S_{k-1})\beta(S_k)}\right] \tag{5}$$

where α is the forward state metric, β is the backward state metric, γ is the branch metric, and S_k is the trellis state at time instant k. At state k, the forward state metric, $\alpha_k(S_k)$ is given by:

$$\alpha_k(S_k) = \sum_{j=0}^{1} \alpha_{k-1}(S_{k-1})\gamma_j(S_{k-1}, S_k) \tag{6}$$

The backward state metric, $\beta_k(S_k)$ is given by:

$$\beta_k(S_k) = \sum_{j=0}^{1} \beta_{k+1}(S_{k+1})\gamma_j(S_k, S_{k+1}) \tag{7}$$

The branch metric for each possible transition can be calculated as:

$$\gamma_i(S_{k-1}, S_k) = A_k P(S_k \mid S_{k-1}) \exp\left[\frac{2}{N_o}(y_k^S x_k^S(i) + y_k^p x_k^p(i, S_{k-1}, S_k))\right] \tag{8}$$

where A_k is a constant, x_k^S and x_k^p are the transmitted systematic data and parity bits at the transmitter side, and y_k^S, y_k^p are the received noisy bits at the receiver side, respectively.

2) Log-MAP Algorithm

It is a simplified version of MAP algorithm to avoid the mathematical computations complexity. Log-MAP performs the calculations in the logarithmic domain by replacing the exponential and logarithm by the max* operator as follows:

$$\max^*(x, y) = \ln(e^x + e^y) = \max(x, y) + \log(1 + e^{-|y-x|}) \tag{9}$$

where the term $\log(1+e^{-|y-x|})$ is a correction function that can be calculated by using look-up table.

3) Max-Log-MAP algorithm

It approximates the computation of max* operator in Log-MAP algorithm for the sake of simplicity by omitting the correction term, $\log(1+e^{-|y-x|})$ to become as follows:

$$\ln(e^x + e^y) \approx \max(x, y) \tag{10}$$

From the hardware implementation point of view, the complexity is reduced at the expense of decoder performance degradation [7]

C. DVB-RCS Decoder Structure

The decoding process of turbo codes involves the iterative exchange of extrinsic information between the two component decoders. In binary case, only two types of transmitted symbols are possible, 0 or 1. In the decoding process of DVB-RCS code, the case is duobinary which is more complicated. In this case, four types of transmitted symbols are possible, (00, 01, 10, or 11). The corresponding likelihood ratios are as follows:

$$\frac{P(d_k = 00/y)}{P(d_k = 00/y)} = \frac{P(d_k = 00/y) \cdot P(d_k = 00)}{P(d_k = 00/y) \cdot P(d_k = 00)} \tag{11}$$

$$\frac{P(d_k = 01/y)}{P(d_k = 00/y)} = \frac{P(d_k = 01/y) \cdot P(d_k = 01)}{P(d_k = 00/y) \cdot P(d_k = 00)} \tag{12}$$

$$\frac{P(d_k = 10/y)}{P(d_k = 00/y)} = \frac{P(d_k = 10/y) \cdot P(d_k = 10)}{P(d_k = 00/y) \cdot P(d_k = 00)} \tag{13}$$

$$\frac{P(d_k = 11/y)}{P(d_k = 00/y)} = \frac{P(d_k = 11/y) \cdot P(d_k = 11)}{P(d_k = 00/y) \cdot P(d_k = 00)} \tag{14}$$

where, d_k represents the transmitted symbol at time instant k, and y is the received continuous valued noisy symbol.

Performing the decoding in the log-domain is more preferred than in the probability domain 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:

$$\Lambda_{a,b}(A_k, B_k) = \log \frac{P(A_k = a, B_k = b)}{P(A_k = 0, B_k = 0)} \quad (15)$$

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

Figure 3 shows the iterative decoder that can be used to decode the DVB-RCS turbo code. $\{\Lambda_{a,b}^{(i)}(A_k, B_k)\}$ 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. The input LLR values are provided to each decoder along with the received values of the parity bits generated by the corresponding encoder (in LLR form). The decoder can produce the updated LLRs $\{\Lambda_{a,b}^{(o)}(A_k, B_k)\}$ at its output by using these inputs and the knowledge of the code constraints. 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.

It is fairly straightforward to extend the log-MAP and max-log-MAP algorithms [3] to the duobinary case. 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 $\Lambda(C) = \log[P(C = 1)/P(C = 0)]$. Since the 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 (15) must be calculated during each iteration and exchanged between the decoders.

Now let $\gamma_k(\mathbf{S}_i \rightarrow \mathbf{S}_j)$ denote the branch metric corresponding to state transition $\mathbf{S}_i \rightarrow \mathbf{S}_j$ at time k. The branch metric depends on the message and parity couples that label the branch along with the channel observation and extrinsic information at the decoder input. In particular, if transition $\mathbf{S}_i \rightarrow \mathbf{S}_j$ is labelled by (A_k,B_k,W_k, Y_k) = (a, b,w, y) then the branch metric $\gamma_k(\mathbf{S}_i \rightarrow \mathbf{S}_j)$ is given by:

$$\gamma_k(S_i \rightarrow S_j) = \Lambda_{a,b}^{(i)}(A_k, B_k) + w\Lambda(W_k) + y\Lambda(Y_k) \quad (16)$$

Now Let $\alpha_k(\mathbf{S}_i)$ denote the normalized forward metric at trellis stage k and state $\mathbf{S}_i$, while $\alpha'_{k+1}(\mathbf{S}_j)$ is the forward metric at trellis stage k + 1 and state $\mathbf{S}_j$ prior to normalization. The forward recursion is given by:

$$\alpha'_{k+1}(S_j) = \max_{S_i \rightarrow S_j}{}^{*}\{\alpha_k(S_i) + \gamma_k(S_i \rightarrow S_j)\} \quad (17)$$

The forward metrics are normalized with respect to the metric stored in state zero after computing the forward recursion for all $\mathbf{S}_j$ at time k+1as follows:

$$\alpha_{k+1}(S_j) = \alpha'_{k+1}(S_j) - \alpha'_{k+1}(S_0) \quad (18)$$

Again, let $\beta'_{k+1}(\mathbf{S}_j)$ denote the normalized backward metric at trellis state k+1 and state $\mathbf{S}_j$ and $\beta'_k(\mathbf{S}_i)$ denote the backward metric at trellis state k and state $\mathbf{S}_i$ prior to normalization. The backward recursion is given by:

$$\beta'_k(S_i) = \max_{S_i \rightarrow S_j}{}^{*}\{\beta_{k+1}(S_j) + \gamma_k(S_i \rightarrow S_j)\} \quad (19)$$

The backward metrics are normalized with respect to the metric stored in state zero after computing the backward recursion for all $\mathbf{S}_i$ at time k as follows:

$$\beta_k(S_i) = \beta'_k(S_i) - \beta'_k(S_0) \quad (20)$$

The sets of forward and backward metrics are then stored and used to find the LLR values according to (15). For each branch the likelihood ration can be computed as follows:

$$Z_k(S_i \rightarrow S_j) = \alpha_k(S_i) + \gamma_k(S_i \rightarrow S_j) + \beta_{k+1}(S_j) \quad (21)$$

For message pair (A_k, B_k) = (a, b) the likelihood is calculated as:

$$t_k(a,b) = \max_{S_i \rightarrow S_j:(a,b)}{}^{*}\{Z_k\} \quad (22)$$

And the possible values for (a,b) are 01, 10, or 11. At the decoder output, the LLR value is given by:

$$\Lambda_{a,b}^{(o)}(A_k, B_k) = t_k(a,b) - t_k(0,0) \quad (23)$$

After the iteration process is completed, either by fixed number of iterations or based on some convergence criterion, the LLR of each bit in the couple (A_k, B_k) is computed to take the final decision by comparing them to threshold:

$$\Lambda(A_k) = \max{}^{*}\left\{\Lambda_{1,0}^{(o)}(A_k, B_k), \Lambda_{1,1}^{(o)}(A_k, B_k)\right\} - \max{}^{*}\left\{\Lambda_{0,0}^{(o)}(A_k, B_k), \Lambda_{0,1}^{(o)}(A_k, B_k)\right\} \quad (24)$$

$$\Lambda(B_k) = \max{}^{*}\left\{\Lambda_{0,1}^{(o)}(A_k, B_k), \Lambda_{1,1}^{(o)}(A_k, B_k)\right\} - \max{}^{*}\left\{\Lambda_{0,0}^{(o)}(A_k, B_k), \Lambda_{1,0}^{(o)}(A_k, B_k)\right\} \quad (25)$$

where $\Lambda_{0,0}^{(o)}(A_k, B_k) = 0$.

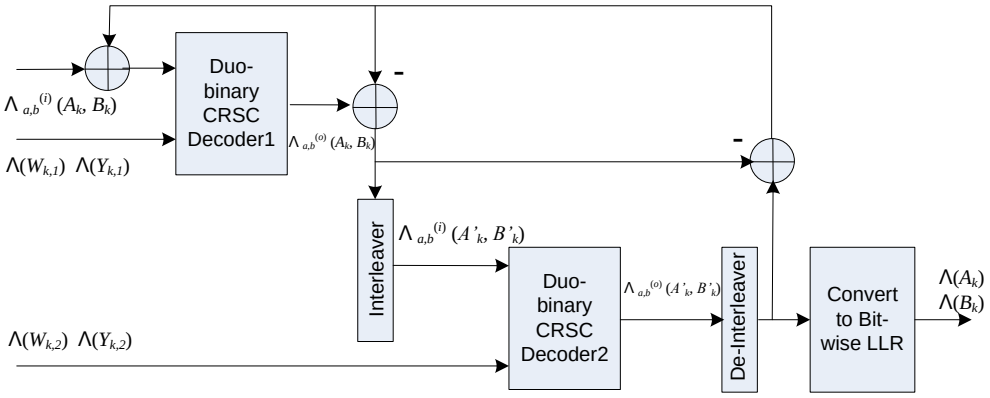

Fig. 3. A decoder for the DVB-RCS code.

III. DVB-RCS Finite Precision Turbo Decoder

In the design phase of turbo codes, good results can be achieved through floating-point software simulations. On the other hand, the efficient hardware implementation of DVB-RCS turbo decoder means, achieving the best performance in terms of area, speed, and low power consumption without loss of error correction capability. 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. There is always a tradeoff between hardware complexity and decoder capability in order to achieve the most efficient implementation of the decoder. At different abstraction levels, some design modifications have to be done including algorithmic, architecture, and circuit levels. Fixed point representation might be realizable and preferable for this reason, during hardware implementation phase; circuits then can process the data in finite precision. Finding a fixed point model that has all bit-widths as small as possible under the condition of an acceptable degradation in coding performance is the primary goal when implementing quantized decoder [11]. In terms of speed, area and power consumption the smaller the bit-width of quantization, the better is the decoder performance. In the same time the performance of the decoder is also affected, for this reason, the quantization should be optimized to control the complexity of the implementation.

For the decoding of turbo codes as it was mentioned before, at the algorithmic level also, a lot of modifications have been done already on decoding algorithms [7]. Avoiding the numerical problems of MAP decoding algorithm, additions instead of multiplications can be used in calculating the extrinsic metric of Log-MAP decoding algorithm. For the sake of more complexity reduction, quantized Max-Log-MAP decoding algorithm can be implemented.

IV. Simulations and Results

The performance of the floating-point simulation of DVB-RCS turbo codes is first evaluated using matlab computer based simulation. The performance has been evaluated in terms of bit error rate (BER) against bit energy E_b in an additive white Gaussian noise (AWGN) channel, having single-sided power spectral density N_o. All the block lengths stated by the standards ranging from 48 to 864 message couples have been included in this simulation. The modulation type is quadrature phase shift keying (QPSK). At the receiver side, a maximum of ten decoding iterations have been performed. Figure 4 shows the influence of the block size on the BER curve at code rate 1/3 while 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 on the BER. Results are shown for all seven code rates when the block sizes are N = 48 and 212 message couples, respectively, and ten iterations of Max-Log-MAP decoding are performed.

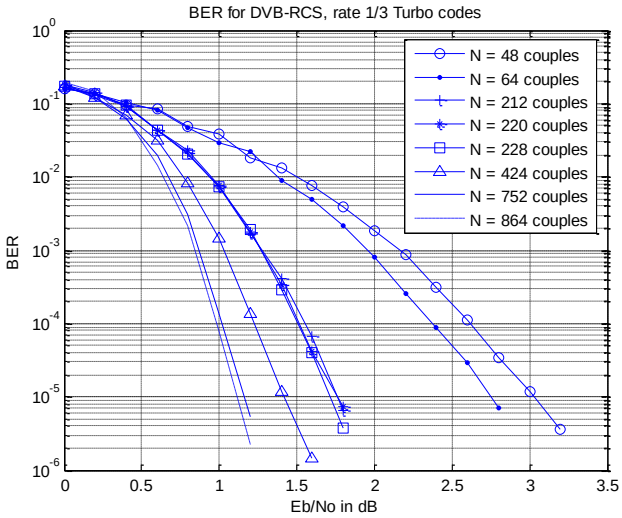

Fig. 4. Influence of block size on the BER performance.

To evaluate the performance of the quantized decoder, another matlab computer based simulation has been driven to compare the performance of the quantized decoder 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 observed that the input LLR values are linearly increasing with the SNR. This observation helped in the estimation of the quantization range. Better estimate of the quantization range can be made based on the SNR values using that observation by introducing SNR-dependent variable factor and multiplying that factor with the LLR input values before the decoding process. As the SNR increases, that multiplier factor should be smaller. The decoding process is then applied on the input LLR values after the multiplying process.

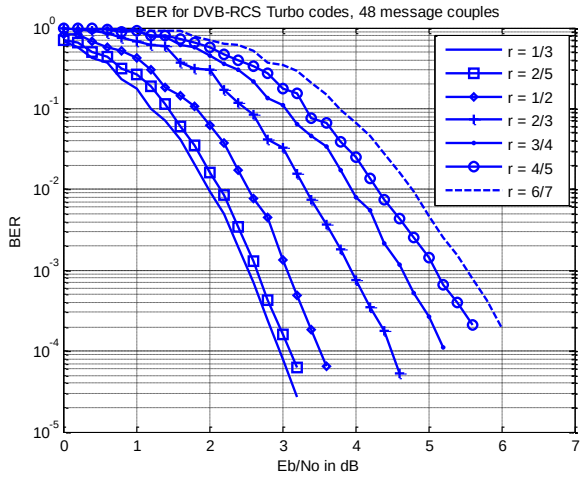

Fig. 5. Influence of block size on the BER performance.

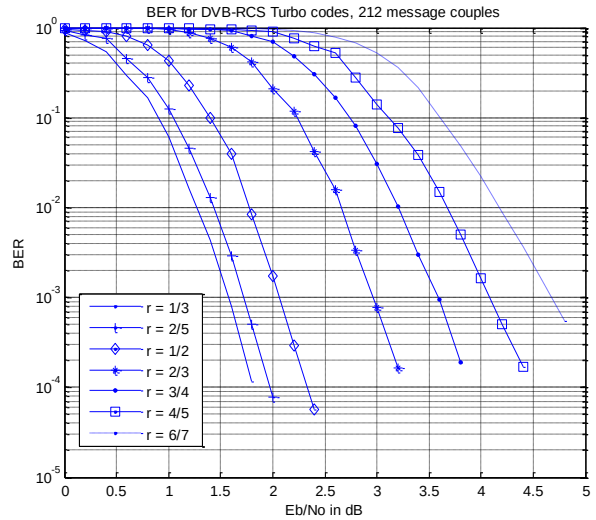

Fig. 6. Influence of block size on the BER performance.

As it was mentioned before, Max-Log-MAP simplifies the Log-MAP algorithm by omitting the correction factor. A lot of researches have been done aiming to reduce the complexity of the decoding algorithm by approximating the correction factor with different methods. Table I presents the most important reduced complexity turbo decoding algorithms. In our research, we have tried also another technique 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 first. 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 [12].

The estimate of the correction factor in our simulation is based on [16]. The computation of the max* operator considering that correction factor is given by:

TABLE I
REDUCED COMPLEXITY ALGORITHMS FOR TURBO DECODING

Decoding Algorithm	Correction factor, $f_c(\|y-x\|)$
MAX-Log-MAP [3]	0
Constant Log-MAP [13]	$\begin{cases} 3/8, if \mid y-x \mid < 2 \\ 0, otherwise \end{cases}$
Linear Log-MAP [14]	$\max(0, \ln 2 - 0.5 * \mid y-x \mid, 0)$
Average Log-MAP [15]	$\begin{cases} \ln 2 + 0.5 * (y+x), if \mid y-x \mid < 2 * \ln 2 \\ 0, otherwise \end{cases}$

$$\max^*(x,y) = \max(x,y) + f_c(|y-x|) \tag{26}$$

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

$$\max^*(x, y) \approx \max(x, y) + \begin{cases} 0, if \mid y-x \mid > T \\ C, if \mid y-x \mid \le T \end{cases} \tag{27}$$

where C is the correction factor and T is a threshold value, C = 0.5, T = 1.5 based on [16].

To measure the performance of the decoding algorithm with constant correction factor for the scaled LLR input values, another matlab code was built 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 (26), to become as follows:

$$\max^*(x, y) \approx \max(x, y) + \begin{cases} 0, if \mid y-x \mid > T \\ K, if \mid y-x \mid \le T \end{cases} \tag{28}$$

where, $K = k \,.\, C$.

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.

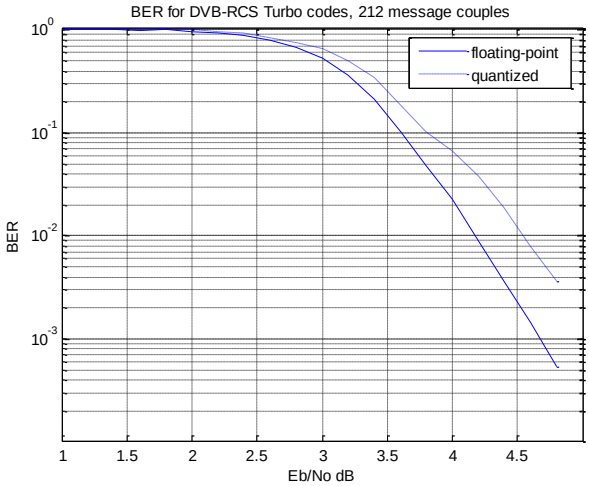

Fig. 7. Floating-point vs. quantized decoder at rate 6/7.

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. Based on the fact that the decoder performance of the quantized code is affected by both of the code rates and the value of SNR, we develop an algorithm to estimate the efficient quantization of DVB-RCS turbo decoder. Figure 8 shows a flow-chart of our proposed algorithm for the decoder quantization.

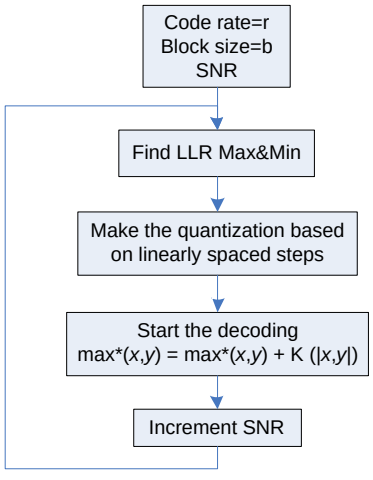

Fig. 8. Proposed algorithm for efficient decoder quantization.

The performance comparison between variable-rate floating-point codes and quantized version of those codes is shown in Figure 9. The simulation was run on blocks of 212 message couples for both of floating point codes and quantized codes using our proposed algorithm. We considered all the possible different codes rates supported by DVB-RCS standards, ranging from r = 1/3 to r = 6/7. For each code rate, the maximum and minimum LLR input values have been calculated at different SNR points. Uniform quantization has been made by using 4-bits. After that, we applied the modified MAX-Log-MAP decoding algorithm based on the proposed modification in this paper. A fixed scaling factor of 0.75 was used to perform this simulation. The performance has been evaluated in terms of bit error rate (BER) against bit energy Eb in an additive white Gaussian noise (AWGN) channel, having single-sided power spectral density No. The modulation type is QPSK. At the receiver side, a maximum of ten decoding iterations have been performed. It can be noticed that the difference in the performance between the floating-point codes and the quantized codes using the proposed algorithm in this paper is less than 0.1 dB. This gives better performance compared with [17] in which the authors obtained 0.4 dB improvement over the standard Max-Log-Map algorithm at BER of 10-4 for code rate 1/3.

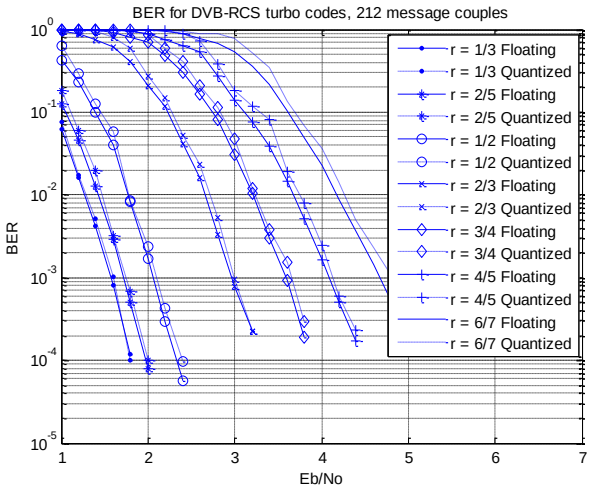

Fig. 9. Influence of quantization on variable rate DVB-RCS Turbo codes.

V. Conclusion

In this paper, an algorithm that computes the quantization range for an efficient decoder quantization has been proposed. The algorithm efficiently estimates decoder quantization of DVB-RCS turbo coding that reduces the complexity of the decoder and overall power consumption in the realization of DVB-RCS radios. The algorithm is based on the modification that improves the BER performance of Max-Log-MAP for fixed point decoder. Performance simulation for different supported code rates of such codes has been presented, and results showed that the quantized decoder essentially matches the performance of the floating point decoder.

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] Papaharalabos, S.; Benmayor, D.; Mathiopoulos, P.T.; Pingzhi Fan; , "Performance Comparisons and Improvements of Channel Coding Techniques for Digital Satellite Broadcasting to Mobile Users," *Broadcasting, IEEE Transactions on*, vol.57, no.1, pp.94-102, March 2011.
[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, "Reduced complexity DVB-RCS turbo decoder," *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, vol., no., pp.444-448, 19-22 Feb. 2012.
[9] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed. Fundamentals of Turbo Codes. 2001, NJ: Prentice Hall.
[10] Bera, D.; Sen, J., "SOVA based decoding of double-binary turbo convolutional code," *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, 2009. Wireless VITAE 2009. 1st International Conference on, vol., no., pp.757-761, 17-20 May 2009.
[11] Shaker, S.W., "DVB-S2 LDPC finite-precision decoder," *Advanced Communication Technology (ICACT)*, 2011 13th International Conference on , vol., no., pp.1383-1386, 13-16 Feb. 2011.
[12] Papaharalabos, S.; Mathiopoulos, P.; Masera, G.; Martina, M., "On optimal and near-optimal turbo decoding using generalized max operator," *Communications Letters, IEEE*, vol.13, no.7, pp.522-524, July 2009.
[13] Gross, W.J.; Gulak, P.G., "Simplified MAP algorithm suitable for implementation of turbo decoders," *Electronics Letters, IEEE*, vol.34, no.16, pp.1577-1578, 6 Aug 1998.
[14] Jung-Fu Cheng; Ottosson, T., "Linearly approximated log-MAP algorithms for turbo decoding," *Vehicular Technology Conference Proceedings, 2000. VTC* 2000-Spring Tokyo. 2000 IEEE 51st , vol.3, no., pp.2252-2256 vol.3, 2000.
[15] Classon, B.; Blankenship, K.; Desai, V., "Channel coding for 4G systems with adaptive modulation and coding" *Wireless Communications, IEEE, 2000*. vol.9, no.2, pp.8-13, April 2002.
[16] Y. Ould-Cheikh-Mahmedou, P. Guinand, and P. Kabal, "Enhanced Max-Log-MAP and Enhanced Log-APP Decoding for DVB-RCS," *Proc. Int. Symp. Turbo Codes*, Brest, France, pp. 259–262, Sept. 2003.
[17] Taskaldiran, M.; Morling, R.C.S.; Kale, I., "A comparative study on the modified Max-Log-MAP turbo decoding by extrinsic information scaling," *Wireless Telecommunications Symposium*, 2007. WTS 2007, vol., no., pp.1-5, 26-28 April 2007.

Dr. Sherif Welsen Shaker, university of Nottingham Ningbo, China, Faculty of Science and Engineering. Sherif Welsen is currently a teaching fellow in the department of Electrical and Electronics Engineering. He obtained his Masters degree of science in Electronics and Telecommunications Engineering from the Arab Academy for Science and Technology and Maritime (AAST), Cairo, Egypt. He then joined Ain-Shams University, where, he got his PhD in Electronics and Electrical Communications engineering in Jan 2010. His research was focusing on Software Defined Radios in wireless communication and in particular on "generic low power FPGA implementation of coding schemes for emerging wireless standards". Dr. Welsen worked as research fellow at Kuang-Chi Institute of Advanced Technology, Shenzhen, Guangdong, China from 2011 to 2013, and as assistant professor at Modern Academy of Engineering and Technology from 2004 to 2010. He was part-time research professor at Ain-Shams University, Cairo, Egypt from 2010 to 2011 in the project "DVBT-2 Systems and Cognitive Radio Networks Sharing the Same Frequency Band". He was the founder of many labs in the academia including microprocessors, FPGAs, and wireless communications. He is involved in Cognitive radios, SDRs, Coding Theory, Digital VLSI, VHDL, FPGA and Digital Signal Processing research.