Reliable JPEG Image Transmission Using Unequal Error Protection with Modified Non-Binary Turbo Codes

Tulsi Pawan Fowdur
Dept of Electrical and Electronic Engineering
University of Mauritius
Reduit, Mauritius
e-mail: p.fowdur@uom.ac.mu

Yashvin Beni
Dept of Electrical and Electronic Engineering
University of Mauritius
Reduit, Mauritius
e-mail: yashvinbeni@gmail.com

***Abstract*— JPEG is a widely deployed image compression standard used in several applications. However, JPEG image transmission is challenging and sophisticated strategies are required for reliable transmission. This paper proposes a JPEG image transmission using non-binary Turbo codes with Unequal Error Protection (UEP), extrinsic information scaling and iterative detection. UEP is achieved by applying a lower code-rate to protect the DC-layer of the image more efficiently and a higher code-rate for protecting the AC-layer. Additionally, the non-binary Turbo code is modified by scaling its extrinsic information to improve performance and by using a stopping criterion to limit the number of iterations required for decoding. The proposed UEP scheme provides a gain of at least 10 dB in Peak Signal to Noise Ratio (PSNR) over an Equal Error Protection (EEP) scheme when duo-binary Turbo codes are used and a gain of at least 14 dB in PSNR when triple binary codes are used, over a range of Eb/No values. Moreover, the use of triple binary turbo codes provides greater bandwidth efficiency.**

Keywords- JPEG; UEP; Non-Binary Turbo Codes.

I. Introduction

This paper builds upon the scheme proposed in [1] to develop an enhanced JPEG image transmission scheme with non-binary Turbo codes. JPEG is a Discrete Cosine Transform (DCT) based image compression algorithm, which employs Huffman coding to generate a compressed bit-stream [2]. It is a widely adopted standard and forms an integral part of several applications such as web browsing and telemedicine [3]. However, the use of Huffman coding renders the JPEG coded bit-stream very sensitive to error propagation because a single bit in error can cause a complete loss of synchronisation. As such, sophisticated coding solutions are required to ensure reliable transmission. One solution is to use powerful error-correcting codes such as Turbo codes, which are well suited to protect image data as recently demonstrated in [4]. Error resilient and concealment techniques also provide a significant improvement in transmission fidelity [5], [6]. Moreover, a highly efficient strategy for achieving robust JPEG image transmission is UEP. UEP consists of exploiting the fact that the DCT operation in JPEG, segments the image into layers of unequal importance. Hence, by allocating different levels of protection to these layers, a significant gain in the overall quality of the received image can be obtained.

Several efficient UEP schemes have been developed for JPEG image transmission using Turbo codes. For example, in [7], UEP and joint source channel decoding with a-priori statistics were combined and applied to JPEG image transmission. Both Turbo codes and Turbo and Turbo Trellis Coded Modulation were used and major gains in PSNR were obtained over conventional JPEG image transmission schemes. An error resilient wireless JPEG image transmission scheme, which employed product Turbo or Reed Solomon codes alongside an optimal UEP algorithm was proposed in [8]. In [9], an UEP scheme, which employs s-random odd-even interleaving with odd-even puncturing, as well as a new UEP scheme for the soft output Viterbi algorithm, were proposed. Improved BER and PSNR performances in JPEG image transmission were obtained with these UEP schemes [9]. In [10], a novel Turbo UEP coding scheme was proposed whereby two error protection levels of Turbo codes are achieved by a new rate-compatible puncturing mechanism. With the scheme of [10], the quality of the image transmission is improved without any additional bit rate or coding delay due to UEP. Moreover, in [11], the performance of an adaptive Wiener-Turbo system with JPEG and bit plane compressions was evaluated over Rician and Rayleigh fading channels. The scheme exploited the neighborhood relation of pixels for each color plane by employing a Turbo decoder, JPEG encoder/decoders, and adaptive Wiener filtering. It also adapted the compression ratios according to the importance of the image to be transferred and could recover high quality JPEG and bit plane compressed images [11]. Furthermore, in [12], a dichotomic technique for searching the optimal UEP strategy for the transmission of JPEG 2000 images and video over a wireless channel was proposed. A method of virtual interleaving was also adopted for the transmission of high bit rate streams over packet loss channels, guaranteeing a large PSNR advantage over a plain transmission scheme [12]. Finally, in [13], the performance of three UEP schemes for progressive JPEG image transmission using delay-constrained hybrid ARQ, with iterative bit and symbol combining was proposed. Gains of over 9 dB in PSNR were obtained with the UEP schemes as compared to their corresponding EEP schemes.

In contrast with previous works, which considered binary Turbo codes, this paper extends the work of [1] to investigate the performance an UEP scheme based on non-binary Turbo codes whereby both duo-binary and triple binary Turbo codes are used. These codes provide better convergence of iterative decoding, have reduced latency, lower sensitivity to puncturing, larger minimum distance and lower memory requirement [14]. The non-binary code is modified with a scale factor [15], [16] and stopping criterion [17] to further improve the performance of the UEP scheme. Also, triple binary Turbo codes have the advantage of providing greater bandwidth efficiency. The proposed UEP scheme allocates more protection to the DC layer, which contains the most significant part the image after the DCT operation, and less protection to the AC layer. This is achieved by using the puncturing matrices specified for the duo-binary Turbo code of the DVB-RCS standard [18] and also a puncturing pattern suitable for triple-binary Turbo codes. The UEP scheme with duo-binary Turbo codes outperforms the EEP scheme by at least 10 dB in PSNR and the one with triple binary Turbo codes outperforms its EEP counterpart by at least 15 dB in PSNR over a range of Eb/No values. Moreover, the gain in PSNR increases as the couple or triple length of the non-binary code is increased.

The organization of this paper is as follows. Section II describes the complete system model. Section III presents the simulation results and analysis. Section IV concludes the paper.

II. SYSTEM MODEL

The complete encoding process is shown in Figure 1. The input image is fed to the JPEG encoder, which operates on blocks of 8x8 pixels and performs DCT, quantization and zig-zag ordering [2]. The AC and DC coefficients are then separated into the AC and DC layers. The DC layer regroups the first coefficient from all 8x8 blocks obtained after zig-zag ordering and the AC-layer is the concatenation of the 63 coefficients from all 8x8 blocks. For example in a 256x256 image, there are 1024 blocks of size 8x8 and each block has one DC coefficient and 63 AC coefficients. The DC layer hence contains 1024 coefficients and the AC layer contains 1024x63 coefficients. To prevent error propagation, the AC and DC layers are divided into blocks of 63 and 64 coefficients respectively. The blocks of the DC layer undergo Differential Pulse Code Modulation (DPCM) and DC-Huffman coding. Each block is encoded separately and after Huffman coding a header is inserted to indicate the size in bits of the resulting DC-packet. The blocks of the AC-layer undergo Run-Length Encoding (RLE) followed by AC-Huffman coding and a header is appended to indicate the size of each AC-packet. Each DC and AC packet can be decoded independently and errors within a packet do not propagate throughout the DC or AC layer. The headers are assumed to be transmitted error-free through a side-channel. A code-rate allocation is performed to provide UEP to the DC and AC packets. The DC packets are given the lowest code-rate while the AC-packets are allocated a higher code-rate. The packets are then converted into couples or triplets of length N before being sent to the Non-binary Turbo encoder.

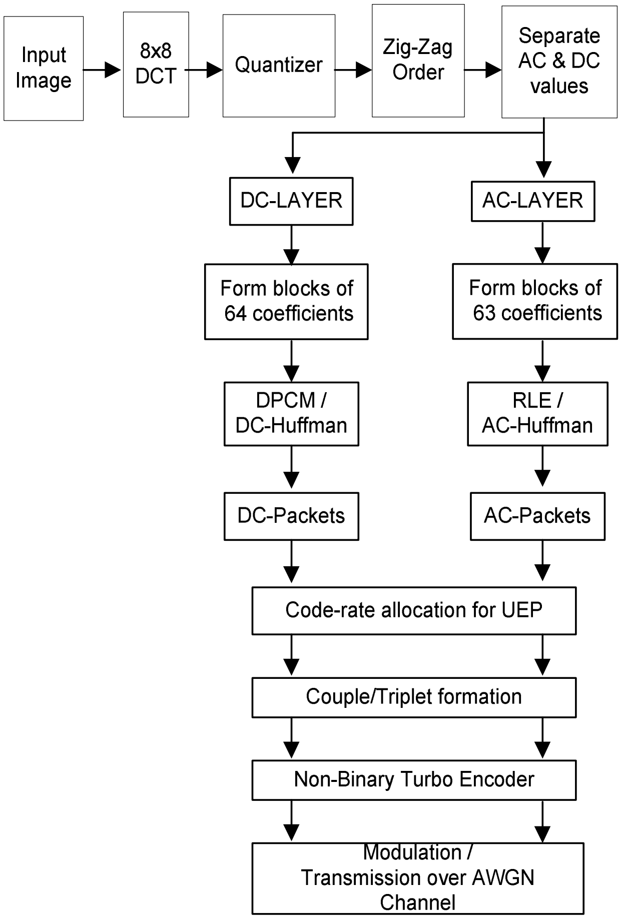


Figure 1. Complete encoding system with UEP.

The block diagram of the duo-binary encoder used is shown in Figure 2. The packets are first partitioned into blocks of length 2N bits. N is typically the couple length of the duo-binary encoder. From each block of length 2N, a couple is formed by dividing it into two streams of length N each. The upper stream is denoted as A and lower stream as B. A and B, together, form a couple of length N, which is the input unit to the encoder. D1,D2 and D3 are shift registers, A and B are the systematic output, W1 and Y1 are the parity outputs from the upper encoder and W2 and Y2 are the parity outputs from the lower encoder. The polynomials defining the connections are described in octal and symbolic notations as follows [18], [19]:

1. For the feedback branch: 15 equivalently $1 + D + D^3$ (in symbolic notation);
2. For Y1/ Y2 parity bits: 13, equivalently $1 + D^2 + D^3$;
3. For the W1/W2 parity bits: 11, equivalently $1 + D^3$.

The double binary code uses a two-level interleaving denoted by INT on Figure 2.

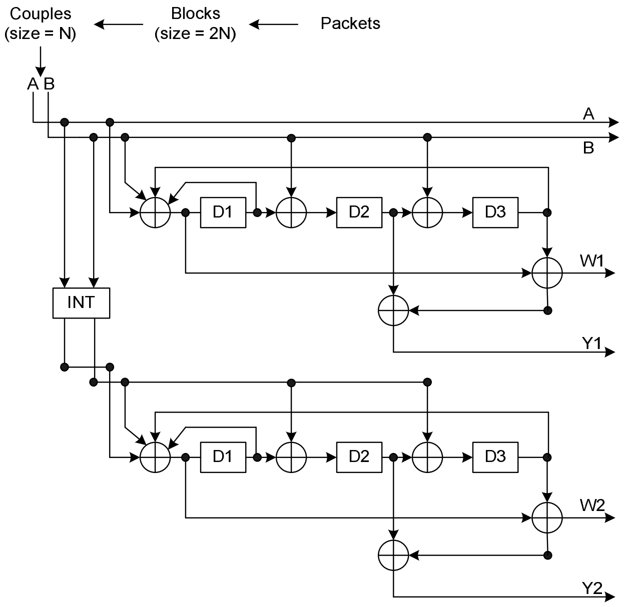


Figure 2. Duo-Binary Turbo encoder.

Let N be the number of bits per couple. The first level interleaving is an inter-symbol permutation performed between the couples and the second one is intra-symbol permutation, which is performed within the couples. The two levels of interleaving are described as follows [18], [19]:

Level 1

```
for j=0,…,N-1
    if  (j modulo 2) =0
      Invert the couple i.e., (A,B)=(B,A)
    endif
endfor
```

Level 2

```
for j=0,…,N-1

  if (j modulo 4)=0
     P=0
  eleseif ( j modulo 4)=1
     P=N/2 + P1;
  elseif (j modulo 4) =2
     P=P2
  elseif ( j modulo 4) =3
     P=N/2+P3
   endif
   Index = (P0*j + P +1) modulo N
   Permute symbol at position j to position Index

endfor
```

The default permutation parameters used for couple sizes of 64 and 212 are given in Table I [18]:

TABLE I
DUO-BINARY PERMUTATION PARAMETERS

Couple Size (N)	P0	{P1,P2,P3}
64	7	{34,32,2}
212	13	{106,108,2}

There are seven code rates that are defined in the DVB-RCS standard and they are: R=1/3, 2/5,1/2, 2/3, 3/4, 4/5 ,6/7. The rates are achieved by selectively deleting parity bits Y1,W1 and Y2, W2 . Table II shows the puncturing patterns for rates 1/3, 2/3 and 4/5 [18].

TABLE II
DUO-BINARY PUNCTURING PATTERNS

Rate	Puncturing pattern
$\frac{1}{3}$	$\begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\frac{2}{3}$	$\begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$
$\frac{4}{5}$	$\begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

After puncturing, QPSK modulation is performed as per the mapping given in Table III where (A, B), (Y1,W1) and (Y2,W2) are mapped to the complex symbols $x0_t$, $x1_t$ and $x2_t$ respectively. The symbols are multiplexed and transmitted over a complex AWGN channel and the receiver obtains $y0_t$, $y1_t$ and $y2_t$.

TABLE III
QPSK MAPPING PARAMETERS

Couple Bits	Symbol	Mapping
00	0	1/√2+1/√2i
01	1	1/√2-1/√2i
10	2	-1/√2+1/√2i
11	3	-1/√2-1/√2i

Figure 3 shows the block diagram for the triple binary encoder. The packets are first partitioned into blocks of length 3N bits. N is the triplet length of the triple-binary encoder. From each block of length 3N, a triplet is formed by dividing it into three streams of length N each. The streams are denoted as A, B and C respectively. A, B and C form a triplet of length N, which is the input unit to the encoder. The systematic outputs are A,B and C while the parity outputs from the upper encoder are W1, Z1, Y1 and the parity outputs from the lower encoder are W2, Z2 and Y2. The connections of the encoder are defined by the polynomials in octal and symbolic notation as follows [18], [20]:

1. For the feedback branch: 23, equivalently $1 + D^3 + D^4$.
2. For the Z parity bits: 35, $1 + D + D^2 + D^3$.
3. For the Y parity bits: 31, equivalently $1 + D + D^4$.
4. For the W parity bits: 21, equivalently $1 + D^4$.

Unlike duo-binary codes, interleaving for triple binary codes occurs only at one level. Thus, only inter-symbol interleaving is used [18], [20], which is described as follows:

```
for j=0,…,N-1

    if (j modulo 4=0)
            P=0
    elseif  (j modulo 4=1)
            P=N/2 +P1
    elseif ( j modulo 4=2)
            P=P2
    elseif (j modulo 4=3)
            P=N/2 +P3
    endif

    Index = (P0*j + P+1) modulo N
    Permute symbol at position j to position Index

endfor
```

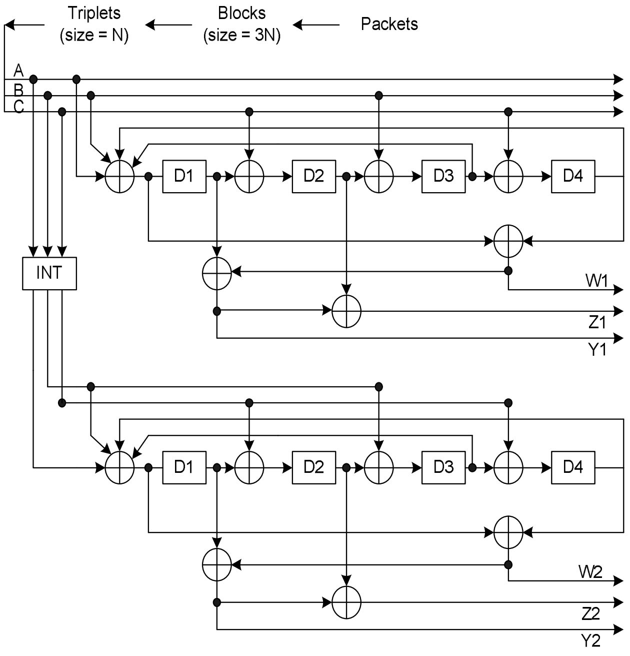


Figure 3. Triple Binary Encoder.

The default permutation parameters used for triplet sizes of 152 and 224 are given in Table IV [18], [20]:

TABLE IV
TRIPLE-BINARY PERMUTATION PARAMETERS

Triplet Size (N)	{P0,P1,P2,P3}
152	{11,34,16,2}
224	{23,114,8,118}

In the DVB-RCS duo-binary code, puncturing is carried out on each individual bit. However for the triple binary code, puncturing is performed symbol-wise. Three code rates have been employed for the triple binary code: 1/3, 2/3 and 4/5. The systematic bits (A, B, C), the parity bits (W1, Z1, Y1) and (W2, Z2,Y2) are first converted to symbols Q0, Q1 and Q2 respectively. Puncturing is then performed on the parity symbols Q1 and Q2 using the patterns given in Table V.

TABLE V
TRIPLE-BINARY PUNCTURING PATTERNS

Rate	Puncturing pattern
1/3	$Q1\&Q2$ [1 1]
2/3	$Q1\&Q2$[1 0 0 0]
4/5	$Q1\&Q2$[1 0 0 0 0 0 0 0]

After puncturing, 8-PSK modulation is performed as per the mapping given in Table VI where Q0, Q1 and Q2 are mapped to the complex symbols $x0_t$, $x1_t$ and $x2_t$ respectively.

TABLE VI
8PSK MAPPING PARAMETERS

Couple Bits	Symbol	Mapping
000	0	1+0i
001	1	1/√2+1/√2i
010	2	-1/√2+1/√2i
011	3	0+1i
100	4	1/√2-1/√2i
101	5	0-1i
110	6	-1+ 0i
111	7	-1/√2-1/√2i

The symbols are multiplexed and transmitted over a complex AWGN channel and the receiver obtains $y0_t$, $y1_t$ and $y2_t$. The decoding system is shown in Figure 4.

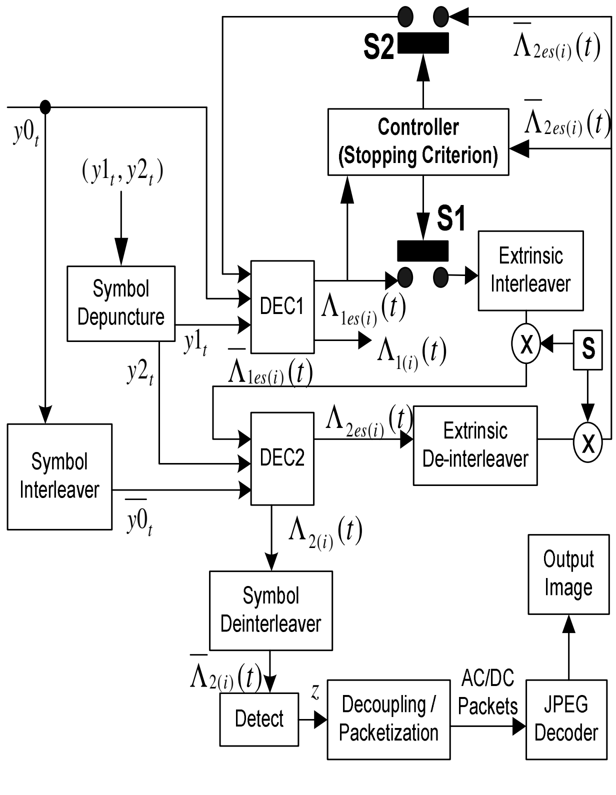


Figure 4. Decoding system with enhanced non-binary Decoder.

The received noisy parity symbols $y1_t$ and $y2_t$ are de-punctured and then sent to the decoders. DEC1 receives $y0_t$ and $y1_t$ while DEC2 receives the interleaved counterpart of $y0_t$ i.e., $\overline{y0}_t$ and $y2_t$. The decoders employ the Max-Log-MAP algorithm to compute the following parameters [18], [19], [20], [21], [22], [23]:

(a) $\overline{\gamma_t^{q(i)}}(l',l)$: The branch transition probability from state *l'* to *l* of symbol *i* at time instant *t*. For the duo-binary decoder, *i* ϵ (0,1,2,3) for decoder *q*, where *q* = 1 or 2 and for the triple binary decoder, *i* ϵ (0,1,2,3,4,5,6,7). It is computed as follows for the first decoder:

$$\overline{\gamma_t^{1(i)}}(l',l) = p(u_t^2 = i) - \begin{pmatrix} [y0_t^I - x0_t^{I(i)}(l)]^2 + [y0_t^Q - x0_t^{Q(i)}(l)]^2 + \\ [y1_t^I - x1_t^{I(i)}(l)]^2 + [y1_t^Q - x1_t^{Q(i)}(l)]^2 \end{pmatrix} \tag{1}$$

where

$p(u_t^2 = i)$ is the a-priori probability of symbol *i* obtained from the second decoder,

$x0_t^{I(i)}(l)$ and $x0_t^{Q(i)}(l)$ are the modulated in-phase and quadrature components of the complex systematic symbol $x0_t$ at time *t*, associated with the transition from state $S_{t-1} = l'$ to $S_t = l$ and input symbol *i*. $x1_t^{I(i)}(l)$ and $x1_t^{Q(i)}(l)$ represent the same for the symbol $x1_t$,

$y0_t^{I(i)}(l)$ and $y0_t^{Q(i)}(l)$ are the in-phase and quadrature components of $y0_t$ while $y1_t^{I(i)}(l)$ and $y1_t^{Q(i)}(l)$ are the in-phase and quadrature components of $y1_t$.

For the second decoder, the computation of $\overline{\gamma_t^{2(i)}}(l',l)$ is similar to equation (1) except that it uses the modulated in-phase and quadrature components of $y2_t$, $x0_t$, $x2_t$, the interleaved counterpart of $y0_t$, and the a-priori probability of symbol *i* obtained from the first decoder i.e., $p(u_t^1 = i)$.

(b) $\overline{\alpha_t^q}(l)$: The forward recursive variable at time *t* and state *l*. It is computed according to the following equation for a decoder with *Ms* states:

$$\overline{\alpha_t^q}(l) = \log \sum_{l'=0}^{Ms-1} e^{\overline{\alpha}_{t-1}^q(l') + \overline{\gamma_t^{q(i)}}(l',l)} \tag{2}$$

With the Max-log-map simplification, equation (2) can be expressed as follows:

$$\overline{\alpha_t^q}(l) = \max\left(\overline{\alpha}_{t-1}^q(l') + \overline{\gamma_t^{q(i)}}(l',l)\right) for \;\; 0 \le l' \le M_s - 1 \tag{3}$$

(c) $\overline{\beta_t^q}(l)$ is the backward recursive variable computed at time *t* as follows:

$$\overline{\beta_t^q}(l) = \log \sum_{l'=0}^{Ms-1} e^{\overline{\beta}_{t+1}^q(l') + \overline{\gamma_{t+1}^{q(i)}}(l,l')} \tag{4}$$

With the Max-log-map simplification, equation (4) can be expressed as follows:

$$\overline{\beta_t^q}(l) = \max\left(\overline{\beta}_{t-1}^q(l') + \overline{\gamma_t^{q(i)}}(l,l')\right) for \;\; 0 \le l' \le M_s - 1 \tag{5}$$

(d) $\Lambda_{q(i)}(t)$,which is the Log Likelihood Ratio (LLR) of symbol *i* where *i* ϵ(1,2,3) for the case of a duo-binary decoder and *i* ϵ (1,2,3,4,5,6,7) for the case of a triple binary decoder. The LLRs are normalized to the symbol '0'. This parameter is computed as follows [15], [16], [17], [18], [19], [23]:

$$\Lambda_{q(i)}(t) = \log \left[\frac{\sum_{l'=0}^{Ms-1} e^{\overline{\alpha_{t-1}^q}(l') + \overline{\gamma_t^{q(i)}}(l',l) + \overline{\beta_t^q}(l)}}{\sum_{l'=0}^{Ms-1} e^{\overline{\alpha_{t-1}^q}(l') + \overline{\gamma_t^{q(0)}}(l',l) + \overline{\beta_t^q}(l)}} \right] \tag{6}$$

The LLR of symbol *i* = 0 is zero. With the Max-log-map simplification, Equation (6) can be expressed as follows:

$$\Lambda_{q(i)}(t) = \max\left(\overline{\alpha}_{t-1}^q(l') + \overline{\gamma_t^{q(i)}}(l',l) + \overline{\beta}_t^q(l)\right) - \max\left(\overline{\alpha}_{t-1}^q(l') + \overline{\gamma_t^{q(0)}}(l',l) + \overline{\beta}_t^q(l)\right) for \;\; 0 \le l' \le M_s - 1 \tag{7}$$

(e) $\Lambda_{1es(i)}(t)$ and $\Lambda_{2es(i)}(t)$: The extrinsic information of symbol *i* where *i* ϵ (1,2,3) for the case of duo-binary and *i* ϵ (1,2,3,4,5,6,7) for triple binary decoding. They are generated by DEC1 and DEC2 respectively and computed as follows:

$$\Lambda_{1es(i)}(t) = \Lambda_{1(i)}(t) - \overline{\Lambda}_{2es(i)}(t) - \Lambda_{1in(i)}(t) \tag{8}$$

$$\Lambda_{2es(i)}(t) = \Lambda_{2(i)}(t) - \overline{\Lambda}_{1es(i)}(t) - \Lambda_{2in(i)}(t) \tag{9}$$

It is to be noted that $\Lambda_{1es(0)}(t) = 0$ *and* $\Lambda_{2es(0)}(t) = 0$.

$\Lambda_{1in(i)}(t)$ and $\Lambda_{2in(i)}(t)$ are the intrinsic information of symbol i where i ϵ(0,1,2,3) or (0,1,2,3,4,5,6,7). They are generated by DEC1 and DEC2 respectively. The generic equation for $\Lambda_{1in(i)}(t)$ is as follows [18], [19], [20], [21], [22]:

$$\Lambda_{1in(i)}(t) = \log\left[\frac{p(y0_t \mid u_t = i)}{p(y0_t \mid u_t = 0)}\right] \tag{10}$$

where

u_t is the systematic part of the encoded symbols and corresponds to (A_t,B_t) for the duo-binary encoder where the symbols take values i ϵ(0,1,2,3) and (A_t,B_t,C_t) for the triple binary encoder where i ϵ (0,1,2,3,4,5,6,7).

For duo-binary transmission, u_t is modulated to obtain the complex symbol $x0_t$, which takes values as per Table III. The intrinsic information of symbol i =1, from the first decoder is computed as follows:

$$\begin{aligned}\Lambda_{1in(1)}(t) &= \log\left[\frac{p(y0_t \mid x0_t = 1/\sqrt{2}, -1/\sqrt{2})}{p(y0_t \mid x0_t = 1/\sqrt{2}, 1/\sqrt{2})}\right] \\ &= -\frac{2}{\sqrt{2}\sigma^2} y0_t^Q\end{aligned} \tag{11}$$

where $y0_t^I$ and $y0_t^Q$ are the in-phase and quadrature components of $y0_t$ and σ^2 is the noise variance. Similarly, the intrinsic information for the symbols i = (2, 3) are obtained as follows:

$$\Lambda_{1in(2)}(t) = -\frac{2}{\sqrt{2}\sigma^2} y0_t^I \tag{12}$$

$$\Lambda_{1in(3)}(t) = -\frac{2}{\sqrt{2}\sigma^2}\left(y0_t^I + y0_t^Q\right) \tag{13}$$

For the symbol i = 0, the intrinsic information is zero. Regarding the second decoder, the computations are similar except that the interleaved counterpart of $y0_t$ is used.

For triple-binary transmission, u_t is modulated to obtain the complex symbol $x0_t$, which takes values as per Table VI. The intrinsic information of symbol i =1, from the first decoder is computed as follows:

$$\begin{aligned}\Lambda_{1in(1)}(t) &= \log\left[\frac{p(y0_t \mid x0_t = 1/\sqrt{2}, 1/\sqrt{2})}{p(y0_t \mid x0_t = 1,\ 0)}\right] \\ &= \frac{1}{2\sigma^2}\left(\sqrt{2}y0_t^I + \sqrt{2}y0_t^Q - 2y0_t^I\right)\end{aligned} \tag{14}$$

The intrinsic information for the symbols i = (2, 3,4,5,6,7) are obtained is a similar way and the details are given in [18], [19], [20], [21], [22]. Again for the symbol i = 0, the intrinsic information is zero and for the second decoder, the interleaved counterpart of $y0_t$ is used.

The a-priori probabilities are computed as follows by the first duo-binary decoder [18], [19], [20], [21], [22]:

$$p(u_t^1 = 0) = -\max\left(0, \Lambda_{1es(1)}(t), \Lambda_{1es(2)}(t), \Lambda_{1es(3)}(t)\right) \tag{15}$$

$$p(u_t^1 = i) = \Lambda_{1es(i)}(t) - \max\left(0, \Lambda_{1es(1)}(t), \Lambda_{1es(2)}(t), \Lambda_{1es(3)}(t)\right) \tag{16}$$

where i = (1,2,3).

Regarding the first triple binary decoder, the a-priori probabilities are computed as follows:

$$p(u_t^1 = 0) = -\max\begin{pmatrix}0,\ \Lambda_{1es(1)}(t), \Lambda_{1es(2)}(t), \Lambda_{1es(3)}(t), \\ \Lambda_{1es(4)}(t), \Lambda_{1es(5)}(t), \Lambda_{1es(6)}(t), \Lambda_{1es(7)}(t)\end{pmatrix} \tag{17}$$

$$p(u_t^1 = i) = \Lambda_{1es(i)}(t) - \max\begin{pmatrix}0,\ \Lambda_{1es(1)}(t), \Lambda_{1es(2)}(t), \Lambda_{1es(3)}(t), \\ \Lambda_{1es(4)}(t), \Lambda_{1es(5)}(t), \Lambda_{1es(6)}(t), \Lambda_{1es(7)}(t)\end{pmatrix} \tag{18}$$

where i ϵ (1,2,3,4,5,6,7).

For the second decoder, $\Lambda_{2es(i)}(t)$ is used in the above equations instead of $\Lambda_{1es(i)}(t)$.

Further details on the computation of these parameters are given in [18], [19], [20], [21], [22].

In the enhanced non-binary decoder, the extrinsic information produced by both decoders are multiplied by a scale factor S as shown in Figure 4. The application of the scale factor improves performance because the extrinsic information value output by the Turbo decoder is most of the time too optimistic, hence by scaling it, better performance is achieved [15], [16]. The controller unit accepts the extrinsic information from both decoders and uses a stopping criterion [17] to stop the iterative decoding process. At the start of the iterative decoding process, switches S1 and S2 are ON and when a given condition is met, the controller unit turns OFF both switches to stop the iterative decoding process. In this way, the decoder avoids the use of extra iterations and reduces the decoding complexity. This technique also reduces the power consumption of the decoder.

A detailed algorithm for the decoding process using duo-binary Turbo codes is now presented. In this algorithm, steps 4-12 correspond to the operations of DEC1 and steps 13-21 of DEC2. The parameters $M_{11}^{r}(t), M_{12}^{r}(t), M_{21}^{r}(t), M_{22}^{r}(t)$ are used in the stopping criterion and the function ***f*( )** counts the number of sign changes between the two arguments that are passed to it. The function ***detect( )*** determines the maximum of the LLR values and outputs either symbol 0,1,2, or 3. The variable j increases by 1 because the decoder processes one couple at a time up to a maximum of N$_c$, which is the total number of couples in the image. The variable r also increases by 1 up to a maximum limit of r$_{max}$. However, the variable num_iterations, which is used to count the number of iterations consumed by the decoder, is incremented by 0.5. This is because the stopping criterion can stop the decoding process after either DEC1 or DEC2 whereby each decoder consumes 0.5 iterations. For example, if for a given couple, the decoding process completes 2 full iterations and then at the third iteration i.e., r=3, after passing through DEC1, the stopping criterion is satisfied, then only 0.5 additional iteration is consumed and hence num_iterations will be 2.5 and not 3. The complete decoding algorithm is as follows:

```
1. num_iterations = 0
2. for  j = 1:$N_c$
3.    for  r = 1:$r_{max}$
4.          Compute: $\overline{\gamma_t^{1(i)}}(l',l)$, $\overline{\alpha_t^1}(l)$, $\overline{\beta_t^1}(l)$, $\Lambda_{1es(i)}(t)$, $\Lambda_{1(i)}(t)$
5.          num_iterations = num_iterations + 0.5.
6.          $M_{11}^r(t) = \max(\Lambda_{1es(2)}(t), \Lambda_{1es(3)}(t)) - \max(\Lambda_{1es(0)}(t), \Lambda_{1es(1)}(t))$
7.          $M_{12}^r(t) = \max(\Lambda_{1es(1)}(t), \Lambda_{1es(3)}(t)) - \max(\Lambda_{1es(0)}(t), \Lambda_{1es(2)}(t))$
8.          if (r>1)
9.              if ( $f(M_{11}^r(t), M_{11}^{r-1}(t)) \le \frac{1}{N} or f(M_{12}^r(t), M_{12}^{r-1}(t)) \le \frac{1}{N}$ )
10.                 break
11.             endif
12.         endif
13.         Compute: $\overline{\gamma_t^{2(i)}}(l',l)$, $\overline{\alpha_t^2}(l)$, $\overline{\beta_t^2}(l)$, $\Lambda_{2es(i)}(t)$, $\Lambda_{2(i)}(t)$
14.         num_iterations = num_iterations + 0.5.
15.         $M_{21}^r(t) = \max(\Lambda_{2es(2)}(t), \Lambda_{2es(3)}(t)) - \max(\Lambda_{2es(0)}(t), \Lambda_{2es(1)}(t))$
16.         $M_{22}^r(t) = \max(\Lambda_{2es(1)}(t), \Lambda_{2es(3)}(t)) - \max(\Lambda_{2es(0)}(t), \Lambda_{2es(2)}(t))$
17.         if (r>1)
18.             if ( $f(M_{21}^r(t), M_{21}^{r-1}(t)) \le \frac{1}{N} or f(M_{22}^r(t), M_{22}^{r-1}(t)) \le \frac{1}{N}$ )
19.                 break
20.             endif
21.         endif
22.   endfor
23.   Decoded couple, $z$ = detect ($\overline{\Lambda}_{2(i)}$(t)) )
24. endfor
25. Convert the received couples into AC and DC packets.
26. Perform JPEG decoding on the received packets.
```

The corresponding algorithm for triple binary Turbo codes is as follows:

```
1. num_iterations = 0
2. for  j = 1:$N_c$
3.    for  r = 1:$r_{max}$
4.          Compute: $\overline{\gamma_t^{1(i)}}(l',l)$, $\overline{\alpha_t^1}(l)$, $\overline{\beta_t^1}(l)$, $\Lambda_{1es(i)}(t)$, $\Lambda_{1(i)}(t)$
5.          num_iterations = num_iterations + 0.5.
6.          $M_{11}^r(t) = \max\begin{pmatrix}\Lambda_{1es(4)}(t), \Lambda_{1es(5)}(t), \\ \Lambda_{1es(6)}(t), \Lambda_{1es(7)}(t)\end{pmatrix} - \max\begin{pmatrix}\Lambda_{1es(0)}(t), \Lambda_{1es(1)}(t), \\ \Lambda_{1es(2)}(t), \Lambda_{1es(3)}(t)\end{pmatrix}$
7.          $M_{12}^r(t) = \max\begin{pmatrix}\Lambda_{1es(2)}(t), \Lambda_{1es(3)}(t), \\ \Lambda_{1es(6)}(t), \Lambda_{1es(7)}(t)\end{pmatrix} - \max\begin{pmatrix}\Lambda_{1es(0)}(t), \Lambda_{1es(1)}(t), \\ \Lambda_{1es(4)}(t), \Lambda_{1es(5)}(t)\end{pmatrix}$
8.          $M_{13}^r(t) = \max\begin{pmatrix}\Lambda_{1es(1)}(t), \Lambda_{1es(3)}(t), \\ \Lambda_{1es(5)}(t), \Lambda_{1es(7)}(t)\end{pmatrix} - \max\begin{pmatrix}\Lambda_{1es(0)}(t), \Lambda_{1es(2)}(t), \\ \Lambda_{1es(4)}(t), \Lambda_{1es(6)}(t)\end{pmatrix}$
9. flag = 0
```

```
10.      if (r>1)
11.         if $\begin{pmatrix} f(M_{11}^{r}(t),M_{11}^{r-1}(t))\leq\frac{1}{N} or \\ f(M_{12}^{r}(t),M_{12}^{r-1}(t))\leq\frac{1}{N} or \\ f(M_{13}^{r}(t),M_{13}^{r-1}(t))\leq\frac{1}{N} \end{pmatrix}$
12.              $Decoded\ triplet, z = detect\left(\Lambda_{1(i)}(t)\right)$
13.              flag = 1
14.              break
15.           endif
16.      endif
17.      Compute: $\overline{\gamma_t^{2(i)}}(l',l)$, $\overline{\alpha_t^{2}}(l)$, $\overline{\beta_t^{2}}(l)$, $\Lambda_{2es(i)}(t)$, $\Lambda_{2(i)}(t)$
18.      num_iterations = num_iterations + 0.5.
19.  $M_{21}^{r}(t)=\max\begin{pmatrix}\Lambda_{2es(4)}(t),\Lambda_{2es(5)}(t),\\ \Lambda_{2es(6)}(t),\Lambda_{2es(7)}(t)\end{pmatrix}-\max\begin{pmatrix}\Lambda_{2es(0)}(t),\Lambda_{2es(1)}(t),\\ \Lambda_{2es(2)}(t),\Lambda_{2es(3)}(t)\end{pmatrix}$
20.  $M_{22}^{r}(t)=\max\begin{pmatrix}\Lambda_{2es(2)}(t),\Lambda_{2es(3)}(t),\\ \Lambda_{2es(6)}(t),\Lambda_{2es(7)}(t)\end{pmatrix}-\max\begin{pmatrix}\Lambda_{2es(0)}(t),\Lambda_{2es(1)}(t),\\ \Lambda_{2es(4)}(t),\Lambda_{2es(5)}(t)\end{pmatrix}$
21.  $M_{23}^{r}(t)=\max\begin{pmatrix}\Lambda_{2es(1)}(t),\Lambda_{2es(3)}(t),\\ \Lambda_{2es(5)}(t),\Lambda_{2es(7)}(t)\end{pmatrix}-\max\begin{pmatrix}\Lambda_{2es(0)}(t),\Lambda_{2es(2)}(t),\\ \Lambda_{2es(4)}(t),\Lambda_{2es(6)}(t)\end{pmatrix}$
22.      if (r>1)
22.         if $\begin{pmatrix} f(M_{21}^{r}(t),M_{21}^{r-1}(t))\leq\frac{1}{N} or \\ f(M_{22}^{r}(t),M_{22}^{r-1}(t))\leq\frac{1}{N} or \\ f(M_{23}^{r}(t),M_{23}^{r-1}(t))\leq\frac{1}{N} \end{pmatrix}$
23.              $Decoded\ triplet, z = detect\left(\overline{\Lambda}_{2(i)}(t)\right)$
24.              flag =1
25.              break
26.           endif
27.     endif
28.   endfor
29.   if(flag = 0)
30.        $Decoded\ triplet, z = detect\left(\overline{\Lambda}_{2(i)}(t)\right)$
31.   endif
32. endfor
33. Convert the received couples into AC and DC packets.
34. Perform JPEG decoding on the received packets.
```

The algorithm for the triple binary decoder uses six parameters $M_{11}^{r}(t), M_{12}^{r}(t), M_{13}^{r}(t), M_{21}^{r}(t), M_{22}^{r}(t), M_{23}^{r}(t)$ for the stopping criterion whereas the duo-binary decoder uses only four. The computations of these parameters are also different for triple binary as compared to duo-binary. Another minor difference is that in the triple binary system the triplet can be decoded at three points. First at line 12 where the stopping criterion is met at the first decoder, second at line 23 where the criterion is met at the second decoder and third at line 28, where the stopping criterion is not met and the couple is decoded from the LLR obtained after all decoding iterations have been performed.

III. SIMULATION RESULTS AND ANALYSIS

The performances of the following four schemes for JPEG image transmission are compared for both double-binary and triple binary Turbo codes:

Scheme 1- UEP with scale factor: This scheme employs UEP to provide different levels of protection to the AC and DC packets of the image. It also uses a scale factor, S, to enhance the performance of the duo-binary Turbo code by scaling the extrinsic information, as depicted in Figure 4. The value of S is set in the range $0 < S < 1.0$.

Scheme 2 - UEP without scale factor: This scheme is similarly to Scheme 1 but the extrinsic information is not scaled and the value of S is set to 1.0 in Figure 4.

Scheme 3 - EEP with scale factor: It is similar to Scheme 1 but equal protection is given to the AC and DC packets.

Scheme 4 - EEP without scale factor: This scheme is similar to Scheme 3 but the scale factor, S, is set to 1.0.

In all simulations, the 256x256 Lena image is used as input. Moreover, it is assumed that the headers are transmitted error free over a strongly protected side channel. The overall coding rate, O_c, was limited to $O_c < 0.97$ bits/pixel and to ensure a fair comparison, the overall coding rate for UEP was less than or equal to that of EEP. However, with UEP the DC packets are more strongly protected with a code-rate of 1/3 while the AC packets are allocated a code-rate of 4/5. On the other hand the EEP schemes allocate a fixed code-rate of 2/3 to both DC and AC packets. The overall coding rate, O_c, is computed as follows:

$$O_c = \frac{1}{T}\left(\frac{T_{DC}}{R_{DC}} + \frac{T_{AC}}{R_{AC}}\right) \qquad (19)$$

where

T is total number of pixels in the image,
T_{DC} is the total number of bits in the DC packets,
R_{DC} is the code-rate allocated to the DC packets,
T_{AC} is the total number of bits in the AC packets,
R_{AC} is the code-rate allocated to the AC packets.

Parameters and results for duo-binary Turbo codes

In this simulation, the DVB-RCS standard duo-binary Turbo code [15] has been used with a stopping criterion. The encoder structure is given in Figure 2 and the generator polynomials are given in Section II. Puncturing matrices are chosen as per the DVB-RCS standard and are given in Table II. The value of S has been set to 0.75 in this simulation for Schemes 1 and 3. The source coding rate, S_c, and O_c, vary with the couple length because different numbers of padding bits are required to convert the bit stream from the JPEG encoder into couples of length N. Table VII gives the values of O_c and S_c for different couple lengths, N.

TABLE VII
CODING RATES FOR DIFFERENT VALUES OF N

N	T_{DC}	T_{AC}	S_C	O_C	
				UEP	EEP
64	5760	36096	0.639	0.952	0.958
212	5936	36040	0.641	0.959	0.961

Figure 5 shows the graph of PSNR versus Eb/No for the four schemes with N = 64.

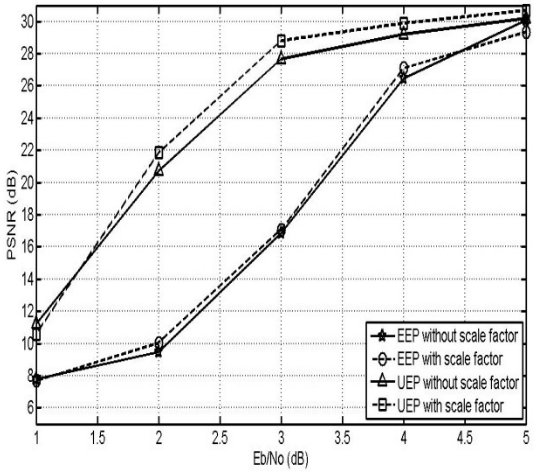


Figure 5. Graph of PSNR against Eb/No for N = 64.

The UEP scheme with scale factor provides a gain of 7dB in PSNR over the EEP schemes at Eb/No = 1.5dB and a major gain of 12dB in PSNR in the range 2dB ≤ Eb/No ≤ 3dB. It also outperforms the UEP scheme without scale factor by 1dB in PSNR in the range 2dB ≤ Eb/No ≤ 3dB. The UEP scheme outperforms the EEP schemed because with UEP the DC layer is recovered with fewer errors than the AC-layer and hence, the image can be reconstructed with much less distortions. However, it is observed that at high Eb/No values, the gain obtained with UEP over EEP decreases because the overall number of errors introduced in the image is considerably less, and convergence occurs.

The graph of number of iterations versus Eb/No for N = 64 is shown in Figure 6. The stopping criterion allows the number of iterations and hence the decoding complexity to decrease progressively as the Eb/No is increased. Interestingly, the UEP scheme with scale factor requires less iterations than the EEP schemes in the range 1dB ≤ Eb/No ≤ 3dB and provides an impressive reduction of 5.5 iterations over the EEP scheme without scale factor at Eb/No = 1dB.

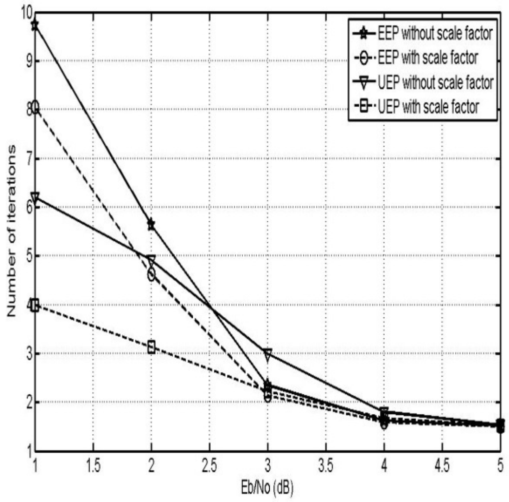


Figure 6. Number of iterations against Eb/No for N = 64.

Figure 7 shows the graph of PSNR versus Eb/No for the four schemes with N = 212.

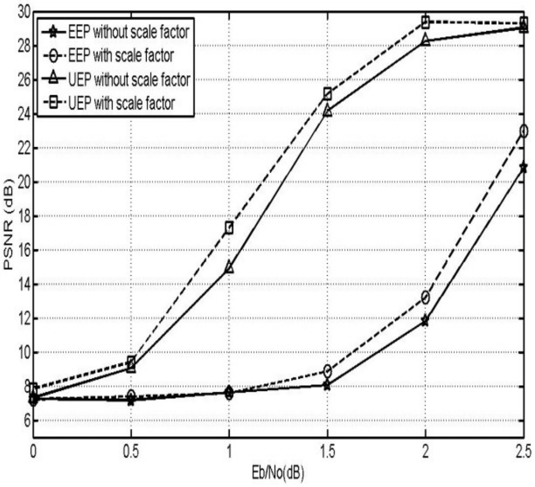


Figure 7. Graph of PSNR against Eb/No for N = 212.

The UEP scheme with scale factor provides a gain of 10 dB in PSNR over the EEP schemes at Eb/No = 1dB and a major gain of 15 dB in PSNR in the range

1.5 dB ≤ Eb/No ≤ 2dB. It also outperforms the UEP scheme without scale factor by about 1dB in PSNR in the range 1dB ≤ Eb/No ≤ 2dB. Moreover, with N = 212 the UEP scheme with scale factor outperforms the UEP scheme with scale factor for N = 64, by an average of 5 dB in PSNR. The gain is greater with a couple length of N=212 because the performance of the duo-binary Turbo code improves with increase in couple length.

The graph of number of iterations versus Eb/No for N = 212 is shown in Figure 8. It is observed that when N=212, the UEP scheme with scale factor takes less iterations than the EEP schemes only in the range 0dB ≤ Eb/No ≤ 1dB. For Eb/No > 1.5 dB the EEP scheme with scale factor requires significantly less iterations than the UEP schemes, for example, at Eb/No = 2.5dB it requires 5 iterations less than the UEP scheme without scale factor. A possible explanation for that is that the threshold used in the stopping criterion was not optimized for N = 212 and was maintained at 1/N.

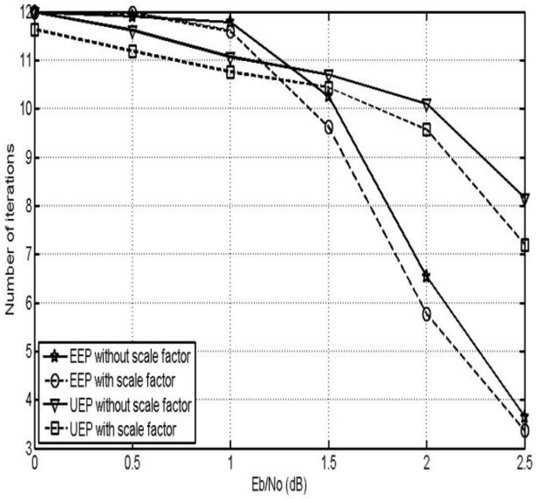
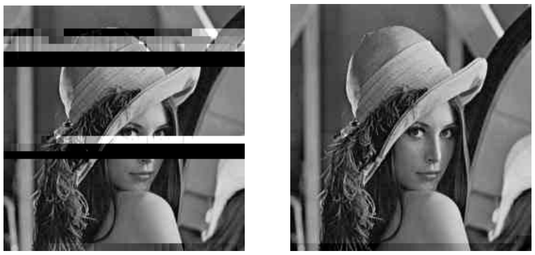


Figure 8. Number of iterations against Eb/No for N = 212.

Some decoded images with the UEP scheme with scale factor and the EEP scheme with scale factor at Eb/No = 2dB for N = 212 are shown in Figure 9.



(a) PSNR = 13.69dB
EEP Scheme 3

(b) PSNR = 29.14dB
UEP Scheme 1

Figure 9. Decoded images at Eb/No = 2dB with N = 212.

The images again confirm the superiority of the proposed UEP scheme.

Parameters and results for triple-binary Turbo codes

The encoder structure of the triple binary Turbo code used is given in Figure 3 and the generator polynomials are given in Section II. Puncturing matrices are chosen as per Table VI. The scale factor S is set to 0.75 in the range 1dB ≤ Eb/No ≤ 1.5 dB and increased to 0.9 in the range 1.7 dB ≤ Eb/No ≤ 2.5 dB for the UEP Scheme 1. This increase was required because the scale factor of 0.75 was causing a degradation in performance for Eb/No >1.5dB in the case of the UEP scheme with triple binary codes. However, for the EEP Scheme 3 it is set to S = 0.75 for the range 0 dB ≤ Eb/No ≤ 2.5dB.

The source coding rate, S_c, and O_c, vary with the triplet length because different numbers of padding bits are required to convert the bit stream from the JPEG encoder into triplets of length N. Table VIII gives the values of O_c and S_c for different triplet lengths, N.

TABLE VIII
CODING RATES FOR DIFFERENT VALUES OF N

N	T_{DC}	T_{AC}	S_C	O_C	
				UEP	EEP
152	5928	36024	0.640	0.958	0.960
224	5936	36040	0.641	0.969	0.969

Figure 10 shows the graph of PSNR versus Eb/No for the four schemes with N = 152.

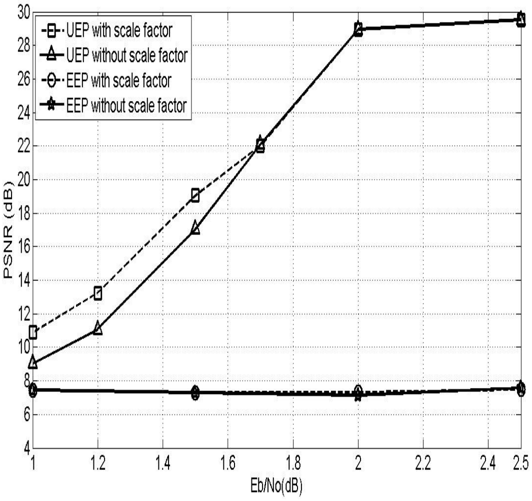


Figure 10. Graph of PSNR against Eb/No for N = 152.

The UEP scheme with scale factor provides a gain of 14 dB in PSNR over the EEP schemes at Eb/No = 1.7dB and a major gain of 20dB in PSNR in the range 2dB ≤ Eb/No ≤ 2.5 dB. In the case of triple-binary codes it is observed that the

EEP scheme shows hardly any improvement in PSNR over the range 1dB ≤ Eb/No ≤ 2.5dB because it uses a code-rate of 2/3 the convergence range for this code-rate is between 3.5dB ≤ Eb/No ≤5.5 dB. Hence the UEP schemes provide the additional benefit of converging much faster than the EEP scheme. The UEP scheme with scale factor also outperforms the UEP scheme without scale factor by 1dB in PSNR in the range 1dB ≤ Eb/No ≤ 1.5dB. However, it is observed that for Eb/No values ≥1.7dB dB, the UEP scheme with scale factor has an almost similar performance to the UEP scheme without scale factor due to convergence.

The graph of number of iterations versus Eb/No for N = 152 is shown in Figure 11. Again the stopping criterion allows the number of iterations and hence the decoding complexity to decrease progressively as the Eb/No is increased. Interestingly, the UEP scheme with scale factor requires on average 1.5 less iterations than the EEP schemes in the range 1dB ≤ Eb/No ≤ 2.5dB. At 1.7dB, there is an increase in the number of iterations for the UEP scheme with scale factor because the scale factor was increased to 0.9.

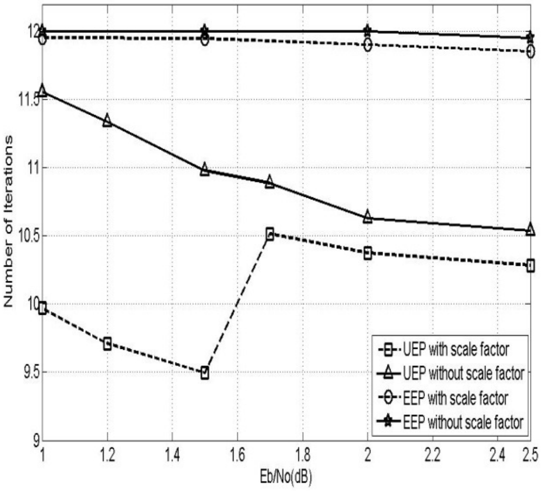


Figure 11. Number of iterations against Eb/No for N = 152.

Figure 12 shows the graph of PSNR versus Eb/No for the four schemes with N = 224. The UEP scheme with scale factor provides a gain of 14 dB in PSNR over the EEP schemes at Eb/No = 1.5dB and a major gain of 20dB in PSNR in the range 1.7dB ≤ Eb/No ≤ 2.5 dB. It also outperforms the UEP scheme without scale factor by an average of 1dB in PSNR in the range 1dB ≤ Eb/No ≤ 1.7dB. Moreover, with N = 224 the UEP scheme with scale factor outperforms the UEP scheme with scale factor for N = 152, by an average of 5 dB in PSNR at Eb/No = 1.7dB. The gain is greater with a couple length of N=224 because the performance of the triple-binary Turbo code also improves with increase in couple length.

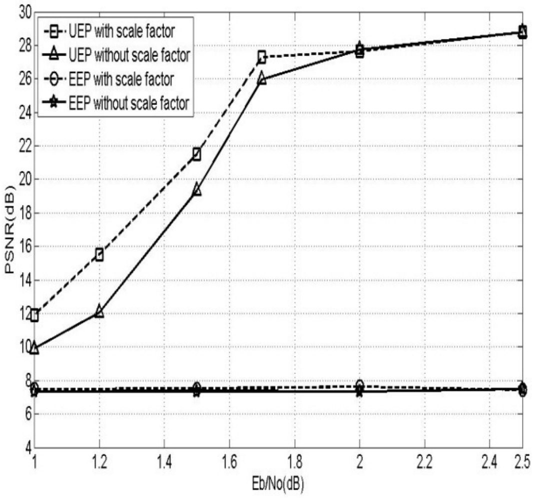


Figure 12. Graph of PSNR against Eb/No for N = 224.

The graph of number of iterations versus Eb/No for N = 224 is shown in Figure 13. It is observed that when N=224, the UEP scheme the UEP scheme with scale factor requires on average 1 iteration less than the EEP schemes in the range 1dB ≤ Eb/No ≤ 2.5dB. At 1.7dB, there is a slight increase in the number of iterations for the UEP scheme with scale factor because the scale factor was increased to 0.9.

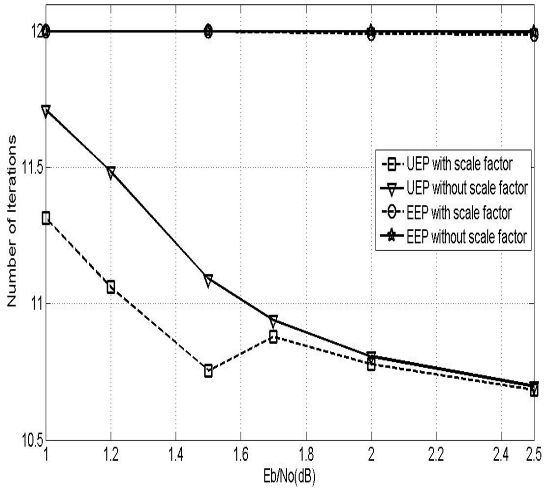
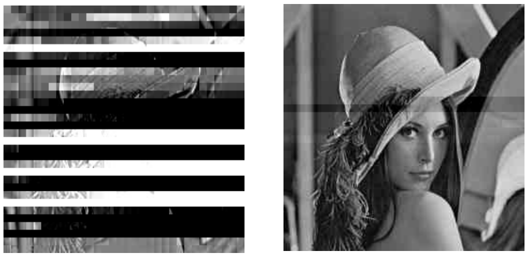


Figure 13. Number of iterations against Eb/No for N = 224.

Some decoded images with the UEP scheme with scale factor and the EEP scheme with scale factor at Eb/No = 2dB for N = 152 are shown in Figure 14. The images again confirm the superiority of the proposed UEP scheme.



PSNR = 7.15 dB
EEP Scheme 3

PSNR = 28.50 dB
UEP Scheme 1

Figure 14. Decoded images at Eb/No = 2dB with N = 152.

An interesting benefit of using triple binary Turbo codes is that it provides greater bandwidth efficient than the duo-binary Turbo codes. However it has a greater encoder and decoder complexity.

There are two ways in which the UEP scheme can lead to an increase in complexity with respect to the EEP scheme. First, in the case of duo-binary Turbo codes, over a certain Eb/No range, as observed in Figure 6, the UEP scheme requires more iterations than the EEP scheme. Second, with the UEP scheme, the non-binary Turbo encoder must treat the AC and DC packets separately and use different code-rates, hence different puncturing patterns are required for each of them. The same applies for the non-binary Turbo decoder, whereby a different de-puncturing process must be used for the AC and DC packets.

IV. CONCLUSION

This paper proposed an efficient UEP scheme for JPEG image transmission with modified non-binary Turbo codes whereby an extrinsic scale factor and a stopping criterion were incorporated. The performances of four schemes were compared with different couple and triplet lengths. The results showed that over a range of Eb/No values, major gains of the order of 10 dB in PSNR are obtained with the UEP scheme with duo-binary Turbo codes and gains of the order of 14 dB in PSNR are obtained with triple-binary Turbo codes, over conventional EEP schemes. Furthermore, the application of the scale factor coupled with the stopping criterion improved the PSNR performance and reduced the number of iterations required, hence the decoding complexity. Except for the UEP scheme with duo-binary Turbo codes with couple length of 212, all the other UEP schemes required less iterations than the EEP schemes. However, for duo-binary Turbo codes with a couple length of 212, at higher Eb/No values, the EEP schemes required less iterations. Another observation was that the performance of the UEP schemes improved with an increase in the couple or triplet length. Finally, the use of triple binary Turbo codes provided the added benefit of greater bandwidth efficiency. An interesting future work would be to optimize the threshold used in the stopping criterion for couple lengths greater than 64, so as to reduce the number of iterations required by the UEP scheme with duo-binary Turbo codes.

ACKNOWLEDGMENT

The authors would like to thank the University of Mauritius for providing the necessary facilities for conducting this research.

REFERENCES

[1] Tulsi Pawan Fowdur and Yashvin Beni, "An Unequal Error Protection Scheme for JPEG Image Transmission Using Enhanced Duo-Binary Turbo Codes", Fifth International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ 2012), Chamonix / Mont Blanc, France, April-May 2012, pp.62-67.

[2] R.C. Gonzalez, R.E Woods, and S.L Eddins, Digital Image Processing in Matlab, 2nd ed., Gatesmark Publishing, (2009).

[3] L.G. Yamamoto, "Using JPEG image transmission to facilitate telemedicine," The American Journal of Medicine, vol. 13, pp. 55-57, 2009.

[4] Ersin GOSE, "Adaptive Wiener-turbo systems with JPEG & bit plane compressions in image transmission," Turk J Elec Eng & Comp Sci, vol. 19, no.1, 2011. (doi:10.3906/elk-1003-461).

[5] L.W. Kang and J.J. Leou, "An error resilient coding scheme for JPEG image transmission based on data embedding and side-match vector quantization," J. Vis. Commun. Image, R.17, pp. 876-891, 2006.

[6] K.S. Kim, H.Y. Lee, and H.K. Lee, "Supplementary loss concealment technique for image transmission through data hiding," Multimed Tools Appl, vol. 44, pp. 1–16, 2009. (DOI 10.1007/s11042-009-0265-0).

[7] T.P. Fowdur and K.M.S. Soyjaudah, "Hybrid Joint Source Channel Decoding for Progressive JPEG Image Transmission," Circuits Systems and Signal Processing, vol. 29, pp. 857-879, 2010. (DOI 10.1007/s00034-010-9188-2).

[8] M. Padmaja, M. Kondaiah and K.S. RamaKrishna, "Error Resilient Image Transmission over Wireless Fading Channels," International Journal of Engineering Science and Technology,vol. 2(5), pp. 1242-1249, 2010.

[9] A.M. Lakhdar, R.Méliani, and M. Kandouci, "Research on Unequal Error Protection with Punctured Turbo Codes in JPEG Image Transmission System," SJEE,vol.4, no.1, pp. 95-108, June 2007.

[10] Qian Mao, Boqing Xu and Yanping Qin, "A New Scheme to Improve the Quality of Compressed Image Transmission by Turbo Unequal Error Protection Codes", Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Dalian, China, pp. 226 – 229, 2011.

[11] Ersin GOSE, "Adaptive Wiener-turbo systems with JPEG & bit plane compressions in image transmission", Turk J Elec Eng & Comp Sci, Vol.19, No.1, pp.141-155, 2011.

[12] G.Baruffa, P. Micanti, and F.Frescura, "Error Protection and Interleaving for Wireless Transmission of JPEG 2000 Images and Video", IEEE Transactions on Image Processing, Vol .18(2), pp. 346-356, 2009.

[13] T.P Fowdur and K.M.S Soyjaudah, “Highly Scalable JPEG Image Transmission with Unequal Error Protection and Optimal Feedback,” Telecommunication Systems, vol.49, Issue 4, pp. 355-377, 2012. (DOI: 10.1007/s11235-010-9379-y). (Published online: 25 June 2010)

[14] C.Douillard and C.Berrou,"Turbo codes with Rate-m/(m+1) Constituent Convolutional Codes," in IEEE Trans. Commun, pp. 1630-1638, October 2005.

[15] J.Vogt and A.Finger, “Improving the MAX-Log-MAP Turbo decoder,” Electronics letters, vol. 36, no. 23, November 2000.

[16] T.Gnanasekaran and K. Duraiswamy, “Performance of Unequal Error Protection Using MAP Algorithm and Modified MAP in AWGN and Fading Channel,” Journal of Computer Science, vol. 4 (7), pp. 585-590,2008.

[17] B.S. Shim, D.H. Jeong, and S.J. Lim, "A new stopping criterion for turbo codes," in Adv Communication Technology, Ihksan, pp. 1111-1116, 2006.

[18] M.R. Soleymani, Y.Gao and U.Vilaipornsawai,Turbo coding for satellite and Wireless Communication, Kluwer Academic Publishers, 2002.

[19] European Telecommunications Standards Institute, Interaction channel for satellite distributions systems, ETSI EN 301 790,V1.3.1 March 2003.

[20] Yingzi Gao and M R Soleymani, "Spectrally Efficient Non-binary Turbo Codes: Beyond DVB-RCS standard," Proceedings of 21th Biennial Symposium on Communcations, vol. 3, pp. 951-955, May 2002.

[21] Y. Ould-Cheikh-Mouhamedou, S. Crozier and P. Kabal, “Distance measurement method for double binary turbo codes and a new interleaver design for DVB-RCS,” Global Telecommunications Conference 2004, (GLOBECOM '04), vol. 1, pp. 172 – 178,2004 .

[22] Youssouf Ould Cheikh Mouhamedou, On Distance Measurement Methods for Turbo Codes, McGill University, Montreal, PhD Thesis 2005. Available Online:

http://www-mmsp.ece.mcgill.ca/MMSP/Theses/2005/Ould-Cheikh-MouhamedouT2005.pdf [Accessed: August 2011]

[23] Branka Vucetic., and Jinhong Yuan, Turbo Codes : Principles and Applications, Kluwer Academic Publishers, 2000.