Accumulator Aided Decoding of Low Complexity SISO Arithmetic Codes with Image Transmission Application

Saikat Majumder
Department of Electronics and Telecommunication
National Institute of Technology Raipur,
Chhattisgarh, India.

Shrish Verma
Department of Electronics and Telecommunication
National Institute of Technology Raipur,
Chhattisgarh, India.

ABSTRACT

In this paper, we address Joint Source-Channel (JSC) decoding with low decoding complexity over wireless channel. We propose a unity rate accumulator based design for soft-input soft-out decoding for low complexity Chase-like decoding of arithmetic codes. Chase-like decoding is a low complexity algorithm, where a maximum *a posteriori* sequence estimation criterion is employed for maximum likelihood decoding of variable length codes like arithmetic codes. Previous contributions propose iterative decoding SISO arithmetic codes with convolutional codes and LDPC codes. We propose application of unity rate accumulator as inner encoder and decoder in the system, which improves the bit error performance of the system by 1.25 dB with same number of decoding iterations. We evaluate the performance of the proposed scheme for image transmission application.

General Terms

Wireless communication, image transmission, source-channel coding.

Keywords

Arithmetic coding, iterative source-channel coding, Chase-like decoding, accumulator, BCJR algorithm.

1. INTRODUCTION

The current trend in communication puts emphasis on miniaturization, low power consumption, low bandwidth requirement, portability and specified minimum quality of service. Some of these goals are conflicting especially for portable battery operated devices. Source coding or data compression algorithms help achieve these goals by removing the redundancy from the data and only transmitting the minimum required number of bits only. This minimum number of bits, which is required for faithful reproduction at the receiver in a error-free channel, is related to entropy of the source. The removal of redundancy makes the compressed sequence sensitive to transmission errors. Errors occurrences, especially over wireless channels, can be reduced by using channel codes. Variable length codes (VLC), like arithmetic codes and Huffman codes, are widely used for source coding. Arithmetic codes are widely used in multimedia coding standards like JPEG 2000, JBIG2 and H.264. Although VLC codes can compress data almost to the entropy limit, they are very sensitive to channel errors, even with channel coding. Even a single bit error in the VLC coded stream can cause loss of synchronization and cause catastrophic error propagation for all the symbols in the packet.

One approach to avoid this error propagation phenomenon is to altogether avoid VLC codes and to utilize inherent redundancy in source sequence at channel decoder, as proposed in [1-4] and the references therein. These techniques utilize correlation and memory structure within the source for joint decoding at the receiver. The redundancy of the source is utilized by the channel code decoder for improving the decoding performance. But such techniques require accurate model of the source memory correlation at the decoder and are not compatible to existing multimedia coding standards. This makes us look into developing error resilient source-channel coding techniques using VLC codes. In this paper we will concentrate our attention toward arithmetic codes.

One of the first approaches towards robust arithmetic coding uses a forbidden symbol that is not in source alphabet [5]. This symbol is never transmitted and its detection in decoded sequence causes detection of error. This scheme was associated with ARQ scheme for error correction for image transmission application in [6]. For wireless communication applications, arithmetic codes can utilize the soft information out of channel decoder or matched filter [7]. This shows improvement over designs where hard decision input is used for arithmetic decoder. Performance is further improved if soft information is iteratively exchanged between source and channel decoder. The proposed schemes in [7], [8], [9], [10] consider state machine representation of arithmetic decoding to apply channel decoding techniques like Viterbi, list-Viterbi, and other sequential search techniques. In [7] stack algorithm and M-algorithm was applied for maximum a posteriori (MAP) error correction decoding of arithmetic codes with forbidden symbol. The proposed approach in [8] enables iterative decoding of arithmetic codes by means of soft-input soft-output (SISO) decoder based on suboptimal search and pruning of a binary tree. Authors in [11] propose a low complexity SISO decoder for arithmetic codes using soft-output Viterbi algorithm principle and a trellis description of arithmetic codes. Most of these discussed schemes depend on specific trellis constructions and eventually pruning the trellis or reducing it. However, as stated in [12], as the number of source sequence length and source alphabet size increases, the decoding complexity becomes intractable. A practical low complexity alternative to trellis based SISO decoders uses Chase like decoding for VLC coded sources [12], [13], [14]. These decoders show considerable improvement over classical hard decision decoding of arithmetic codes.

Our main contribution in this paper is improving the performance of Chase-like arithmetic SISO decoder system of [12] (with intermediate convolutional code stage) by the use of rate 1 accumulator. Improvement is obtained with same number of decoding iterations and overall code rate. We further analyze the convergence behavior of the system using EXIT charts [15] and show that the decoder is guaranteed to converge if SNR is greater than a particular noise threshold. The rest of the article is organized as follows. In section 2 we summarize arithmetic coding and low complexity SISO Chase-like arithmetic decoding. The proposed decoding scheme and its error performance is discussed in section 3. Section 4 analyzes the scheme using EXIT chart and section 5 compares the performance for image transmission application. Finally, section 6 is conclusion.

2. REVIEW OF ARITHMETIC CODING

2.1 Arithmetic Coding

In this section we briefly describe the encoding and decoding process of arithmetic codes as in [16], [12]. At the encoder, sequence of symbols $S = [s_1, s_2, ..., s_L]$, which may be non-binary, is mapped to a binary string based on the source probability model. In this article we will consider only binary source symbol set, $s \in \{0,1\}$, with probabilities $p_0 = P(s = 0)$ and $p_1 = P(s = 1)$. Encoding is performed recursively by computing probability interval $I(s_k) = [l_k, u_k)$ corresponding to the input string. Initially, $I(s_0) = [0,1)$, then for every recursion *k*, newer probability intervals are obtained as:

$$l_k = l_{(k-1)} + (u_{(k-1)} - l_{(k-1)})F(s_k - 1) \qquad (1)$$

$$u_k = l_{(k-1)} + (u_{(k-1)} - l_{(k-1)})F(s_k) \qquad (2)$$

where $F(s_k)$ is the cumulative probability of symbol s_k. Once this process is completed for all the symbols, a binary representation of the smallest number in the final interval is generated. This is the binary arithmetic code for the given source sequence.

At the decoder, the received binary string is again converted back to a real number called *tag*. After that decoding is just the reverse of the encoding process, as given below [16],

1. Initialize $I(s_0) = [0,1)$.
2. For each *k*, find $t^* = (tag - l_{(k-1)})/(u_{(k-1)} - l_{(k-1)})$.
3. Find symbol s_k for which $F(s_k - 1) \le t^* < F(s_k)$.
4. Update the interval $I(s_k) = [l_k, u_k)$.
5. Compute until all the symbols have been decoded or End of Block (EoB) symbol is encountered.

As can be understood from the above decoding algorithm that successively computed probability interval depend on the tag, and hence on the received bit stream. Evan a single bit error may change the probability interval and throw the decoding process out of synchronization.

2.2 Low-Complexity SISO Arithmetic Decoding

A low complexity soft-input soft-output arithmetic decoder based on Chase-like decoding [17] was proposed in [12]. The arithmetic encoder generates a binary vector C of variable length M, which is interleaved to produce U. This interleaved binary stream U is input to recursive systematic convolutioanl coder (RSCC). Our baseline system uses rate 1/2, 8-state RSCC of generator polynomial $(G_r, G) = (17, 15)_8$. Output of RSCC is BPSK modulated and transmitted over AWGN channel of noise variance σ^2. At the decoder, the received signal is iteratively decoded by exchanging extrinsic information between SISO arithmetic decoder and the BCJR (Bahl, Cocke, Jelinek and Raviv) decoder. As shown in the figure 1, extrinsic output E(U) of BCJR decoder is deinterleaved to produce a priori information A(C) for the arithmetic decoder. Extrinsic output E(C) of the arithmetic decoder is fed back as a priori information to BCJR decoder.

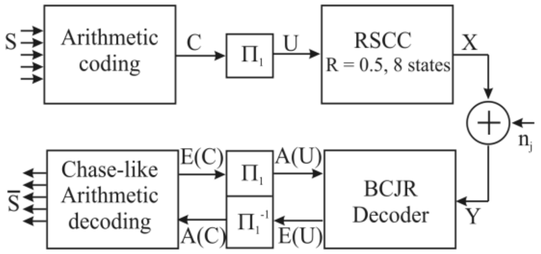


Fig 1: Soft-input soft-output decoding of arithmetic code based on Chase algorithm.

Next we summarize the algorithm behind SISO arithmetic decoder proposed in [12]. Let the *a priori* input A and extrinsic information E be in the form of log-likelihood ratios (LLR).

1. Obtain hard decision vector for A(C) as $A^h = [a_1, a_2, ..., a_M]$ and LLR values as $\alpha(A) = [\alpha_1, \alpha_2, ..., \alpha_M]$. Bit reliabilities are obtained as magnitudes of the LLR values.
2. Determine the location of *q* least reliable elements of A(C) based on reliability values $|\alpha(A)|$.
3. Generate test patterns $T^i = [t_1^i, t_2^i, ..., t_M^i], 0 < i \le 2^q$, each of length M. Each test pattern has maximum weight *q* with all the possible bit combinations in the *q* least reliable positions.
4. Form test vectors Z^i, $0 < i \le 2^q$ with $z_j^i = xor(a_j, t_j^i)$. Each test vector is decoded using classical arithmetic decoding.
5. If the decoded sequence (for *k*-th test vector) has exactly L decoded symbols and correct EoB symbol is decoded, calculate its distance metric as:

   $$M^k = \frac{1}{2\sigma^2} \sum_j \left(\frac{\alpha_j}{Lc} - h_j^{\,k} \right)^2 \qquad (3)$$

   where, $h_j^{\,k} = 1 - 2z_j^i$, and *Lc* is the channel reliability value.
6. Finally the decoded bit stream, denoted as $\overline{S}$, is the one with highest metric. To calculate the extrinsic output value, we find the bits positions which are same for all candidate sequences (test vectors for which distance metric had been calculated). For those bits constant extrinsic information is assigned as $e_j = \beta(1 - 2\hat{z}_j)$ and for all other non reliable bits we assign $e_j = 0$. The SISO module outputs extrinsic information as $E = [e_1, e_2 ... e_M]$.

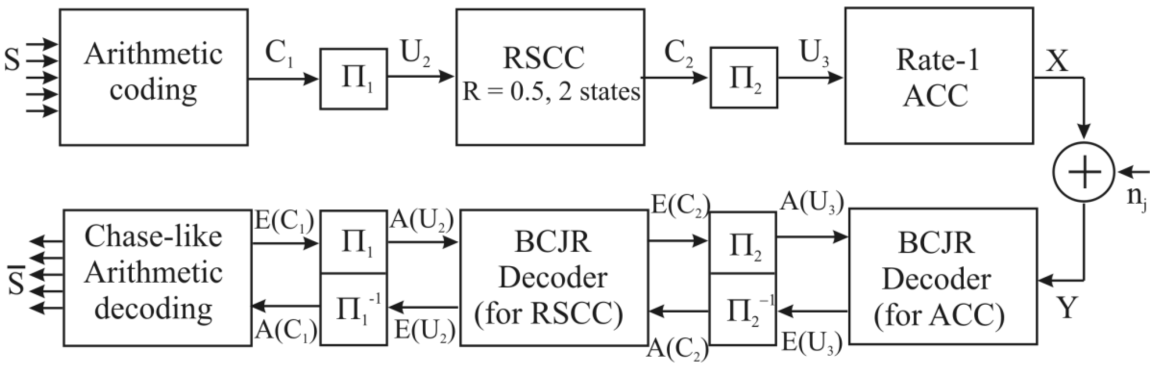


Fig 2: The proposed joint source-channel arithmetic coding system using accumulator.

3. PROPOSED SCHEME

3.1 System Model

Use of accumulators (ACC) have been proposed in many wireless communication systems for close mapping of EXIT charts of two SISO decoders. Authors in [18] use doped accumulator on the top of a bit interleaved coded modulation with iterative detection (BICM-ID) structure such that the EXIT curve matches more closely with outer decoder. In [19] authors use doped accumulator for achieving turbo-cliff. We have used this property of accumulator to obtain iteration gain in cascade of stages comprising of SISO arithmetic decoder. We next describe the proposed scheme. The block diagram of the proposed scheme using rate-1 accumulator (ACC) is shown in figure 2. The first two stages are as in the section 2.2, with the difference that RSCC is a 2 state (memory 1) encoder instead of 8-state encoder, thereby reducing the decoder complexity considerably. Binary output of RSCC is scrambled again and applied to rate-1 accumulator. The rate-1 accumulator is simply RSCC with generator polynomial $(G_r, G) = (3,\ 2)_8$ and with no systematic bits transmitted.

At the receiver noise corrupted version of the code is decoded by soft-out BCJR decoder (for accumulator) to produce extrinsic information $E(U_3)$. It is then de-interleaved and applied to RSCC decoder. Extrinsic output of RSCC decoder $E(U_2)$ is unscrambled and applied as input to SISO arithmetic decoder described in section 2.2. This process is performed iteratively until convergence is reached. The output of the arithmetic decoder $\bar{S}$ at that point is taken as decoded output.

3.2 Performance Evaluation

Several simulations were conducted to evaluate the BER performance and convergence property of proposed technique. Convergence analysis with EXIT chart is detailed in next section. In this section numerical results on packet error rate (PER) are discussed and compared to the results obtained in [12]. We consider a memoryless source with $p_0 = 0.2$ and $p_1 = 0.8$ and packet length of 512 bits, with each packet ending with an EoB symbol. The packet is encoded with conventional arithmetic encoder. As in [12], encoded bits of 4 such packets are concatenated and encoded with rate ½ RSCC with generator polynomial $(G_r, G) = (3,\ 2)_8$. Decoding is done with parameter $q = 4$ and $\beta = 1$.

Figure 3 compares the PER of the proposed scheme, with the baseline system proposed in [12]. PER are plotted as function of E_b/N_0, where E_b is the energy of each bit after arithmetic coding. We can see that for 5 decoding iterations, our scheme performs 1.25 dB better than the baseline system. With more iteration, the proposed scheme performs even better. Simulation results for 5, 10, and 20 iterations are shown. Whereas, for the scheme without accumulator, the performance saturates at 5 iterations and there is no advantage obtained for more iterations. It should be emphasized that proposed scheme has lower complexity and lower memory requirement compared to the baseline scheme in [12]. In our case two 2-state convolutional coders are required with total memory requirement of 2 bits, whereas for the system without accumulator, one 8-state convolutional encoder requires memory of 3 bits. Moreover, complexity of BCJR decoders increase exponentially with memory of the code and hence at the decoder side our scheme is less complex computationally because of using two memory 1 decoders rather than one memory 3 decoder.

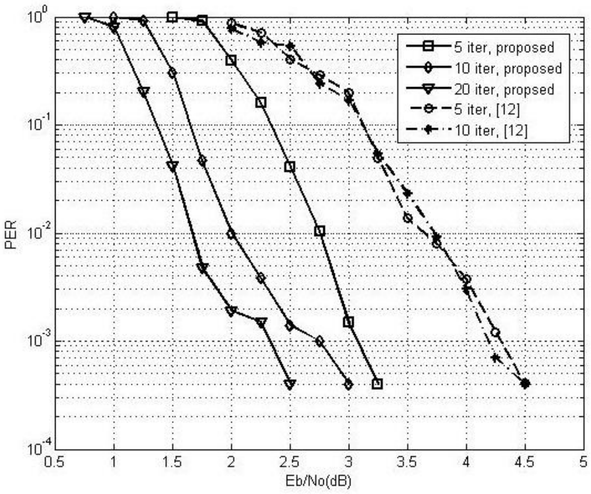


Fig 3: PER performance of the proposed system compared to decoding scheme in [12].

4. EXIT CHART ANALYSIS

Extrinsic information transfer (EXIT) chart is a successful method for predicting the convergence behavior of various concatenated and iterative systems [15]. In this section we analyze the convergence properties of the iterative decoding process using the EXIT chart technique.

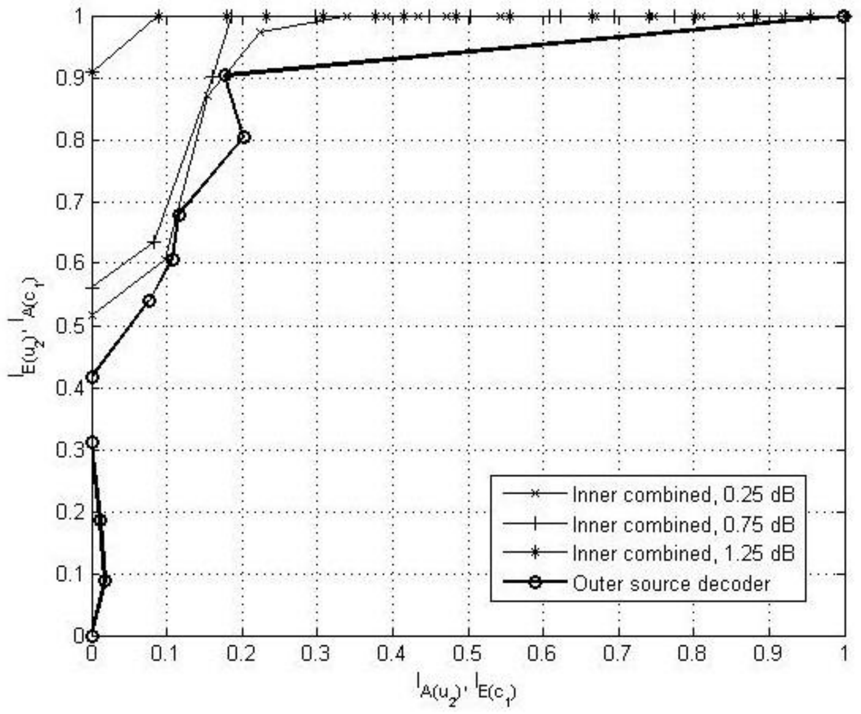


Fig 4: EXIT chart for iterative decoder in fig 2. EXIT curves for combined inner decoder are obtained for different channel conditions. An open tunnel is observed for $E_b/N_0 \geq 0.75$ dB.

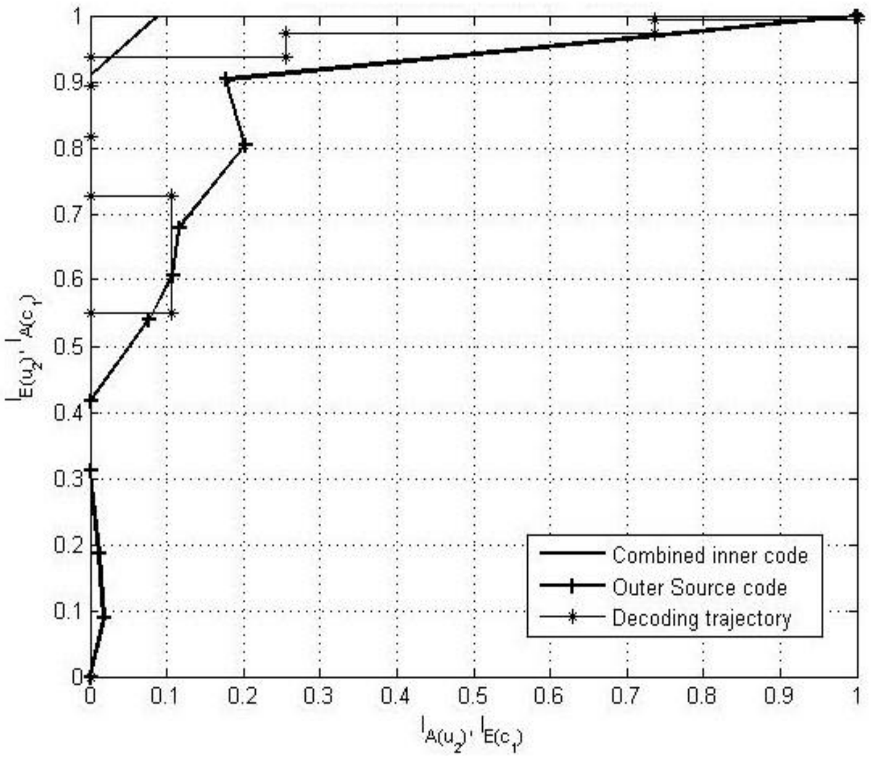


Fig 5: A decoding trajectory for the decoder in fig 2 at E_b/N_0 of 1.25 dB.

The fundamental assumption of EXIT chart is that extrinsic information passed from one SISO to other is a Gaussian random variable. The LLR *A* of *a priori* input is modeled as

$$A = \mu_A x + n_A \tag{4}$$

where *x* is binary antipodal form of information symbols C, n_A is a Gaussian random variable with zero mean and variance $\sigma_A{}^2$. The variance must satisfy the condition $\mu_A = \sigma_A{}^2/2$. The mutual information between *A* and *x* is defined as

$$I(x, A) = \frac{1}{2}\sum_{x=\pm1}\int_{-\infty}^{+\infty} f_A(\xi\,|\,x)\log_2 \frac{2 f_A(\xi\,|\,x)}{f_A(\xi\,|\,x=-1)\, f_A(\xi\,|\,x=+1)}\, d\xi \tag{5}$$

where $f_A(\xi\,|\,x)$ is conditional probability density function associated with *a priori* LLR *A*. Therefore, for a priori LLR A, mutual information is given as $I_A = I(C;A)$. Similarly, mutual information for extrinsic output E is obtained as $I_E = I(C;E)$. To obtain EXIT chart, for given values of $I_A = (0,1)$ we artificially generate the *a priori* inputs *A*, which are fed to SISO module. Then the corresponding decoding algorithm of the block is invoked to produce extrinsic output *E*. The mutual information I_E is then evaluated using equation (5). Finally, EXIT chart is obtained as the graphical plot between I_A and I_E . For decoding without any residual error, $I_E = 1$ for some value of I_A .

In figure 4 we plot the EXIT chart for combined inner decoder for various values of E_b/N_0 and inverted EXIT chart for outer arithmetic SISO decoder. Combined inner EXIT chart is obtained by considering the two inner SISO units as a single blocks and measuring the mutual information after 20 iterations. For further details about obtaining EXIT chart of two combined decoders, readers are referred to [24]. As we can see an opening in the two EXIT functions is obtained at 0.75 dB. With further increase in E_b/N_0, the opening increases even more. This guarantees convergence for all E_b/N_0 greater than 0.75 dB with sufficient number of iterations. In figure 5 we also show the EXIT chart and decoding trajectory at 1.25 dB. We can see that the decoding trajectory 'tunnels' through the two EXIT functions and finally reaching ($I_A = 1, I_E = 1$) point, which is associated with achieving near zero BER.

5. IMAGE TRANSMISSION OVER NOISY CHANNEL

In this section we evaluate the performance of the proposed scheme for an image transmission system using SPIHT coding (Set Partitioning in Hierarchical Trees) and compare with baseline system. SPIHT is a wavelet based image compression algorithm which uses arithmetic coding as last stage. SPIHT exploits the inherent similarities across wavelet subbands to compress the image efficiently. In SPIHT the image is first decomposed into number of subbands using hierarchical wavelet decomposition. The subband coefficients are then grouped into sets called spatial orientation trees, which efficiently exploits the correlation between different bands. The coefficients in each spatial orientation trees are progressively coded from most significant bit planes to least significant bit planes, starting with the highest magnitude coefficients and at lowest pyramid level. Thus SPIHT is a progressive coder in the sense that at any point during the decoding of an image, the quality of displayed image is the best that can be achieved for the number of bits input by decoder up to that moment [29].

In the considered system, the 512×512Lenna test image is compressed using the SPIHT encoder, without arithmetic coding stage, generating output of rate D_s bits per pixel. The progressive bit stream is them packetized, with each source symbol packet of length L = 512 bits and the additional last symbol being EoB (end of block). Arithmetic coded output of four such packets are combined and scrambled to form message U_2. It is then coded by rate ½, memory 1 RSCC and then scrambled again to be encoded by rate 1 accumulator. Finally, the encoded bits are BPSK modulated and transmitted over AWGN channel. At the receiver, iterative decoding as described in section 3 is applied with Chase decoding parameter q fixed to 4 bits.

Simulations were done to obtain average PSNR over 500 image transmissions for both the cases. Simulation results are given in figure 6 and 7 for source bit rates 0.4 bpp (bits per pixel) and 1 bpp, respectively. It compares the PSNR performance of the proposed image transmission system to the results obtained by the baseline system proposed in [12]. It can be seen in figure 6 that at E_b/N_0 of 3.25 dB the proposed system gives a PSNR gain of 10 dB for source rate of 0.4 bpp. On the other hand, at E_b/N_0 of 3.5 dB, there is a gain of 12 dB. All these results were obtained for 5 number of decoding iteration for both the schemes. In figure 8 we give examples of reconstructed images for both the schemes at E_b/N_0 of 3 dB which clearly shows difference in visual quality.

6. CONCLUSION

In this article we have proposed an accumulator based scheme for decoding of Chase-type SISO arithmetic coder over wireless channel. The proposed scheme consisting of cascade of unity rate accumulator, two state BCJR RSCC decoder and SISO arithmetic decoder was compared with a similar system consisting of 8 state BCJR decoder and no accumulator. Simulation results show that the proposed scheme performs better in terms of packet error rate for same overall code rate and decoding iterations compared to the baseline system. Also because of reduction in number of states in BCJR decoder, encoding and decoding complexity is reduced, even though there is an extra accumulator stage. We have studied the convergence behavior of the proposed scheme with EXIT chart analysis. Finally, the presented system has been applied for transmission of SPHIT coded images over AWGN channel. Significant improvement in terms of PSNR and visual quality were observed compared to the baseline system.

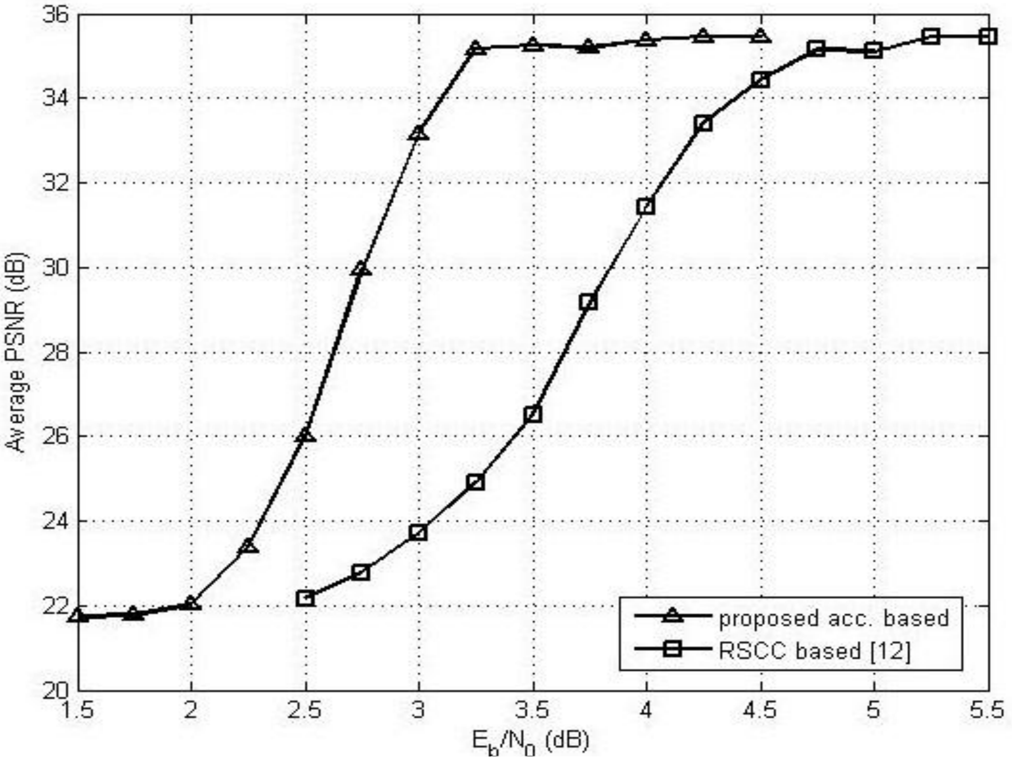


Fig 6: The average PSNR versus E_b/N_0 plot for proposed scheme applied to SPHIT image coding at D_s = 0.4 bpp.

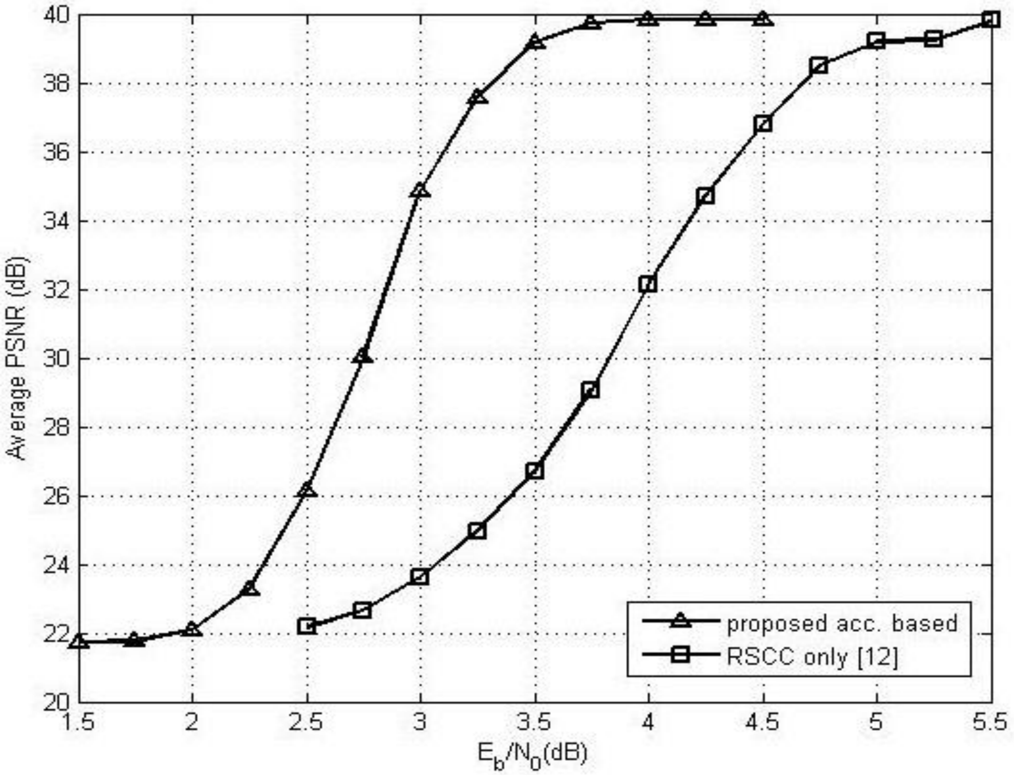
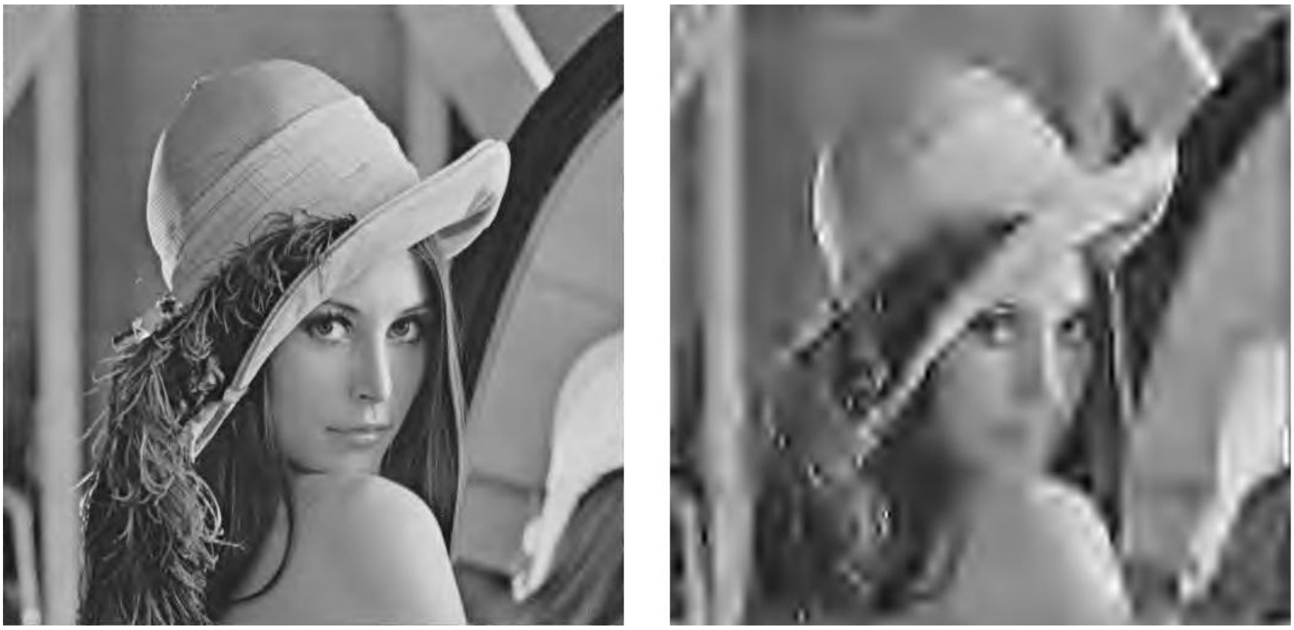


Fig 7: The average PSNR versus E_b/N_0 plot for proposed scheme applied to SPHIT image coding at D_s = 1 bpp.



(a) Proposed scheme **(b) Baseline scheme**

Fig 8: Reconstructed images in the case of E_b/N_0 of 3 dB and D_s = 0.4 bpp.

7. REFERENCES

[1] Garcia-Frias, J., "Compression of correlated binary sources using turbo codes," *Communications Letters, IEEE* , vol.5, no.10, pp.417,419, Oct. 2001.

[2] Guang-Chong Zhu; Alajaji, F., "Turbo codes for nonuniform memoryless sources over noisy channels," *Communications Letters, IEEE* , vol.6, no.2, pp.64,66, Feb. 2002.

[3] Garcia-Frias, J.; Villasenor, J.D., "Joint turbo decoding and estimation of hidden Markov sources," *Selected Areas in Communications, IEEE Journal on* , vol.19, no.9, pp.1671,1679, Sep 2001.

[4] Majumder, S.; Verma, S., "Joint source-channel decoding of IRA code for hidden Markov source," *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on* , vol., no., pp.220,223, 15-17 March 2012.

[5] Boyd, C.; Cleary, John G.; Irvine, S.A.; Rinsma-Melchert, I.; Witten, I.H., "Integrating error detection into arithmetic coding," *Communications, IEEE Transactions on* , vol.45, no.1, pp.1,3, Jan 1997.

[6] Chou, J.; Ramchandran, K., "Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission," *Selected Areas in Communications, IEEE Journal on* , vol.18, no.6, pp.861,867, June 2000.

[7] Grangetto, M.; Cosman, P.; Olmo, G., "Joint source/channel coding and MAP decoding of arithmetic codes," *Communications, IEEE Transactions on* , vol.53, no.6, pp.1007,1016, June 2005.

[8] Grangetto, M.; Scanavino, B.; Olmo, G.; Benedetto, S., "Iterative Decoding of Serially Concatenated Arithmetic and Channel Codes With JPEG 2000 Applications," *Image Processing, IEEE Transactions on* , vol.16, no.6, pp.1557,1567, June 2007.

[9] Pettijohn, B.D.; Hoffman, M.W.; Sayood, K., "Joint source/channel coding using arithmetic codes," *Communications, IEEE Transactions on* , vol.49, no.5, pp.826,836, May 2001

[10] Guionnet, T.; Guillemot, C., "Soft decoding and synchronization of arithmetic codes: application to image transmission over noisy channels," *Image Processing, IEEE Transactions on* , vol.12, no.12, pp.1599,1609, Dec. 2003.

[11] Zribi, A.; Zaibi, S.; Pyndiah, R.; Bouallegue, Ammar, "Low-complexity joint source/channel turbo decoding of arithmetic codes," *Turbo Codes and Related Topics, 2008 5th International Symposium on* , vol., no., pp.385,389, 1-5 Sept. 2008.

[12] Zribi, S., Zribi, A., Pyndiah, R., Aloui, N., “Joint source/channel iterative arithmetic decoding with JPEG 2000 image transmission application”, *EURASIP Journal on Advances in Signal Processing,* 2012

[13] Zribi, A.; Pyndiah, R.; Zaibi, S.; Guilloud, F.; Bouallegue, Ammar, "Low-Complexity Soft Decoding of Huffman Codes and Iterative Joint Source Channel Decoding," *Communications, IEEE Transactions on* , vol.60, no.6, pp.1669,1679, June 2012.

[14] Zribi, A., Zribi, S., Pyndiah, R., Bouallegue, A., “Joint source/channel Chase-like decoding of Entropy encoded sources”, *Journal of Communication and Computer,* vol. 7, no. 11, 2010.

[15] Hanzo, L., Maunder, R.G., Wang, J., Lie-Liang, Y., *Near Capacity Variable-Length Coding: Regular and EXIT Chart Aided Irregular Designs,* John Wiley & Sons, 2011.

[16] Sayood, K., *Introduction to Data Compression*, Morgan Kaufman Publishers, 2000.

[17] Chase, D., "Class of algorithms for decoding block codes with channel measurement information," *Information Theory, IEEE Transactions on* , vol.18, no.1, pp.170,182, Jan 1972.

[18] Anwar, K., Matsumoto, T., “Very simple BICM-ID using repetition code and extended mapping with doped accumulator”, *Wireless Personal Communication (Springer)*, 67: 573-584, 2012.

[19] Zhou, X., Cheng, M., Anwar, K., Matsumoto, T., “Distributed joint source-channel coding for relay systems exploiting source-relay correlation and source memory”, *EURASIP Journal on Wireless Communication and Networking*, 2012.

[20] Salomon, D., *A Guide to Data Compression Methods*, Springer-Verlag, New York, 2002.