

Forward Error Correction using Turbo Codes for Multimedia Data

Subramanya M
Department of Telecommunication Engg.
Dayananada Sagar College of Engineering,
Affiliated to VTU Belgaum
Bangalore, India
Subramanya571401@gmail.com

Shaista Khuteja
Department of Telecommunication Engg.
Dayananada Sagar College of Engineering,
Affiliated to VTU Belgaum
Bangalore, India
Khuteja21@gmail.com

Varun Kumar K C
Department of Telecommunication Engg.
Dayananada Sagar College of Engineering,
Affiliated to VTU Belgaum
Bangalore, India
Varunkc36@gmail.com

Srilatha S
Department of Telecommunication Engg.
Dayananada Sagar College of Engineering,
Affiliated to VTU Belgaum
Bangalore, India
Srilatha.s14196@gmail.com

Srividya B V
Department of Telecommunication Engg.
Dayananada Sagar College of Engineering,
Affiliated to VTU Belgaum
Bangalore, India
srividya21@gmail.com

Abstract: The swift growth in multimedia technology of wireless network has made it mandatory for the efficient transmission across erratic channel. The transmission of encoded video using error control techniques is grabbing a great attention, since it works over the recovery of the lost data and errors in the bit frames which occur as a result of congestion and physical channel fading. Turbo codes are attracting researchers because of their efficient performance. The Turbo code is made up of analogous concatenation of two Recursive Systematic Convolutional (RSC) coders parted by a non-uniform interleaver. For different code rate and information block lengths greater than 10^4 , these codes are capable of achieving low Bit-error rates (BERs) at SNRs within 1dB of Shannon's limit. Turbo codes will assist to employ Viterbi decoders. More the number of iterations, higher is the error correction capacity and hence Turbo codes act as an elucidation for obtaining large coding gains.

Keywords: Forward Error Correction codes (FEC), Turbo codes, Viterbi decoder, Multimedia data, MATLAB 2018.

I. INTRODUCTION

Error control codes are error correcting codes which form the basic element of almost every digital transmission system that are in use today.

In recent years, many multimedia services have become available and the demand for realistic multimedia systems is growing at a very fast pace. In the current scenario, the entail for dominant short and sensible length codes are getting upgraded because of the enhancement in machine-type communications in which the short data packets are often used over the channels that are accessed randomly. The coding analysts have conventionally selected the subject of designing better codes by enhancing the codes structurally which leads to flexible decoding. Coding theory recommends codes selected randomly should yield good results if they have large block size. For iterative decoding, the long channel codes are designed in a well customized way depending upon the iterative decoding threshold. The task to find the practical decoders for all random, lengthy codes has not been considered so far.

The proclamation of "Turbo codes" by Berrou et al. in 1993 has led to the most electrifying and hypothetically important growth in Coding theory. The block Turbo codes (BTC) came into picture in 1994 leading to the closure of the remaining slit to capacity. Turbo product codes is a substitute to CTC. It is applicable to either high code rates, very low error floors or low complexity decoders that are capable of operating up to several hundreds of Mbps or even more.

The introduction to Turbo codes has concreted a new method of analyzing the issue of building efficient codes and decoding them by reduced complexity. At present the systems that make use of advanced Forward Error Correction (FEC) approach near Shannon limit with an offset of few decibels. FEC codes are said to achieve near Shannon-limit error correction act by relatively simple constituent codes and huge interleavers.

II. LITERATURE SURVEY

According to information theory, Turbo codes are a class of high-performance Forward Error Correction (FEC) codes developed around 1990–91. The term "Turbo code" was coined based on the feedback that was used for decoding of the Turbo codes.

The very first paper on Turbo codes was published in IEEE International Communications Conference by Shannon in the year 1993. Turbo codes consists of comparable concatenated convolutional codes. After the Turbo codes were invented, several types of Turbo codes have been revealed which includes serial versions, serial concatenated convolutional codes and repeat-accumulate codes., In spite of the complexity of the FEC systems like Reed-Solomon codes, Iterative Turbo Decoding algorithms have been implemented on these systems.

Turbo equalization got streamed from the notion of Turbo coding. Nowadays, the Turbo codes utilize Big-Numerator-Accumulator constituent codes. The idea of rate-symmetry introduced which is used to create a doubly as-symmetric Turbo codes.

III. METHODOLOGY

As shown in Fig 3.1 the input is obtained from the user in the form of either text, image or video. The obtained data is encoded with the help of Turbo encoder. Now the encoded data is transmitted through the channel where the AWGN is introduced to it. The input data is retrieved from the encoded data with the help of Turbo decoder.

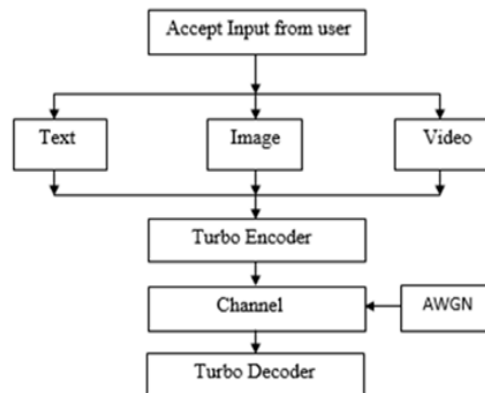


Fig 3.1: Flow diagram

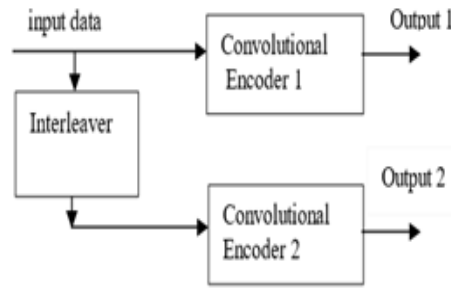


Fig 3.2: Turbo encoder

Fig 3.2 shows a block diagram of Turbo encoder consisting of two parallel concatenated convolutional encoders separated by an interleaver. The input is given to the first encoder and the same input is passed to the second encoder via an interleaver.

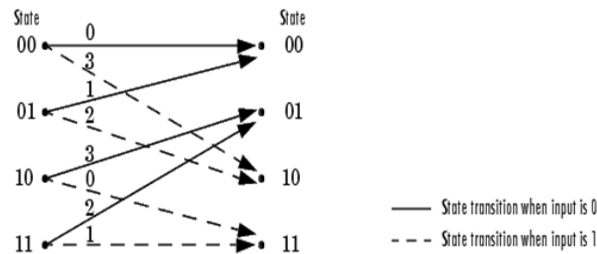


Fig 3.3: Basic Trellis structure

Fig 3.3 shows the basic structure of trellis diagram which is used for Turbo decoding process.

A complexity comparison of different decoding methods per unit time for (n,k) convolutional code with memory order v is provided in Table 3.1.

	MAP	LOG-MAP	MAX-LOG-MAP	SOVA	PROPOSED METHOD
ADDITIONS	$2 \cdot 2^K \cdot 2^V + 6$	$6 \cdot 2^K \cdot 2^V + 6$	$4 \cdot 2^K \cdot 2^V + 6$	$2^K \cdot 2^V + 9$	$2^{K+1} \cdot 2^V$
MULTIPLICATIONS	$5 \cdot 2^K \cdot 2^V + 8$	$2^K \cdot 2^V$	$2 \cdot 2^K \cdot 2^V$	$2^K \cdot 2^V$	-
MAX. OPERATIONS		$4 \cdot 2^V - 2$	$4 \cdot 2^V - 2$	$2 \cdot 2^V - 1$	-
LOOK-UPS		$4 \cdot 2^V - 2$			-
EXPONENTIATION	$2 \cdot 2^K \cdot 2^V$				-

Table 3.1: Complexity comparison table of different decoding algorithms

From the Table 3.1, it can be inferred that the number of additions required for decoding has significantly reduced when compared to the other well known algorithms. Also it can be observed that the proposed decoder does not use multiplication or exponentiation operations.

A trellis represents a graph in which the nodes are arranged into longitudinal slices and with every node at each time linked to at least one node at a previous and at least one node at future time. They are also the central data type used in Viterbi algorithm.

IV. RESULTS

The simulation results are obtained by coding the Turbo Encoder and decoder using MATLAB 2018.

```
input data is
the turbo encoder/decoder is working fine

encoded data is
PljN Po3QÜ NÜ3ÜGN3GN3ÜGN3 Äp tÜ3UÄÜÄ ÖÄÜN
noisydata = 'Y|GU5Æ µ÷-ÖâÊÜw²3÷"CöâµUivÚV}K'
```

Fig 4.1: Simulation results

Fig 4.1 depicts the simulation results for the given data. The input data to the encoder is given as “the Turbo encoder/decoder is working fine”

The encoded output is obtained as shown in Fig 4.2.

```
PljN Po3QÜ NÜ3ÜGN3GN3ÜGN3 Äp tÜ3UÄÜÄ ÖÄÜN
noisydata = 'Y|GU5Æ µ÷-ÖâÊÜw²3÷"CöâµUivÚV}K'
```

Fig 4.2: Encoded output

The decoded and retrieved data is obtained as “the Turbo encoder/decoder is working fine”.
Figures 4.3-4.6 show the results obtained for the encoding and decoding of the image as an input.

Image results

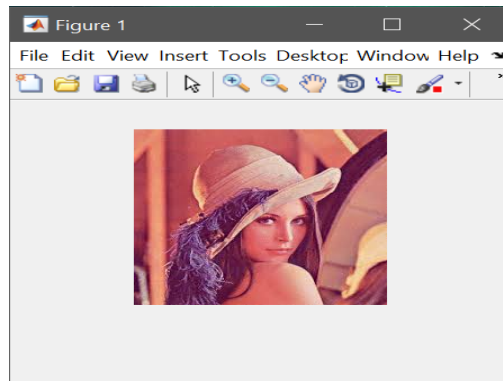


Fig 4.3: Original image

Fig 4.3 shows an input image given to the Turbo encoder.

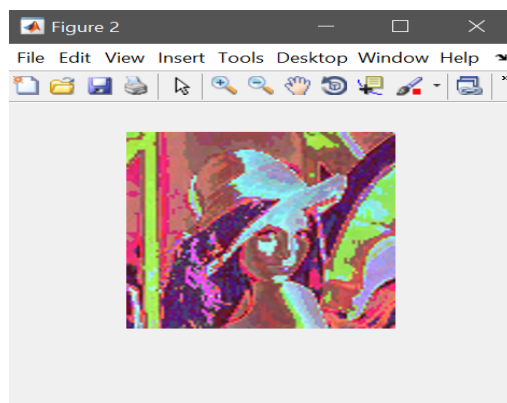


Fig 4.4: Encoded image

Fig 4.4 shows the encoded form of the image.

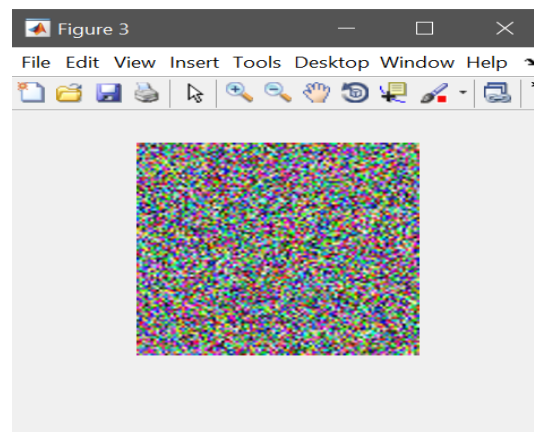


Fig 4.5: Encoded image affected by noise

Fig 4.5 shows the image to which Additive White Gaussian Noise is added for the purpose of experimentation.

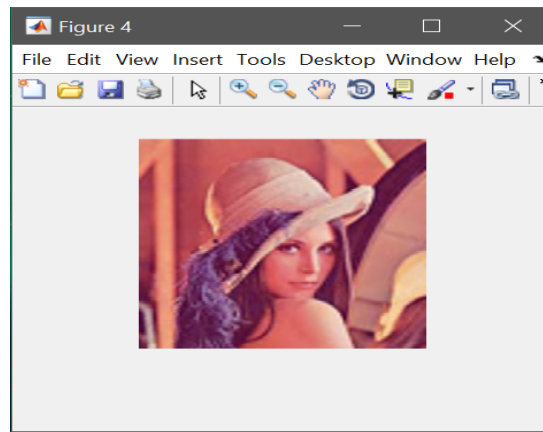


Fig 4.6: Decoded image

Fig 4.6 shows the decoded form of the image which is same as an input image.

Mean Square Error

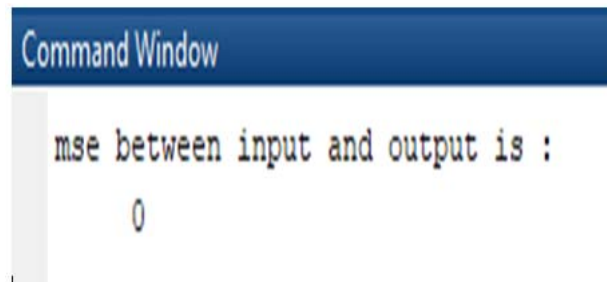


Fig 4.7: Mean Square Error

The Mean Square Error (MSE) is calculated as a ratio between the given input and the decoded output and it is observed to be zero.

Video results:

The video considered for experimentation is foreman.yuv which is for duration of 10 seconds. It consists of 300 frames each of size 352×288 pixels. Each of these RGB frames are converted to Gray Scale Video Frames, for experimentation.

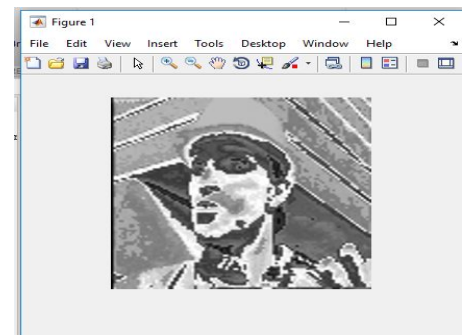
Figure 4.8 shows the sample results of 4 frames.

Figure 4.8a: Original video frames



Frame 1

Figure 4.8b: Encoded video frames



Frame 1



Frame 2

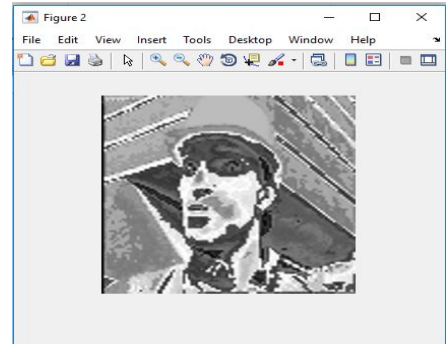


Frame 3

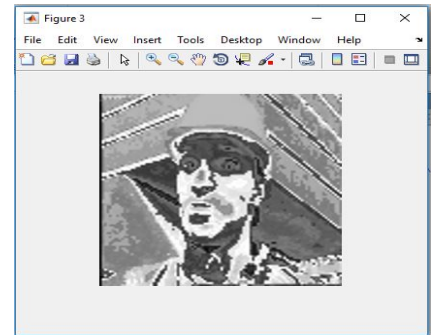


Frame 4

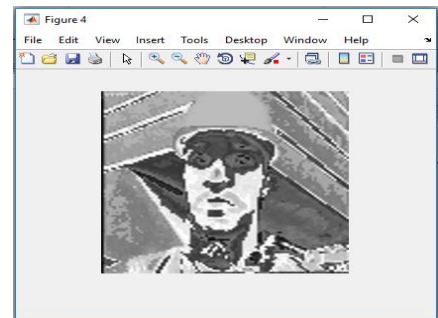
Figure 4.8c: Encoded video frames which are affected by noise



Frame 2

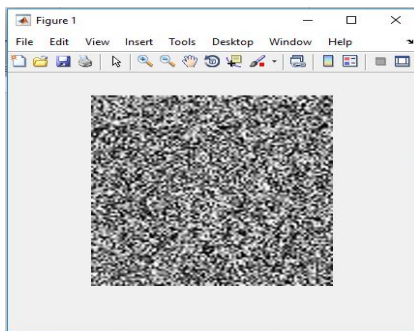


Frame 3



Frame 4

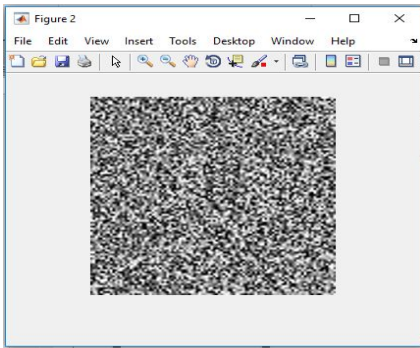
Figure 4.8d: Decoded video frames



Frame 1



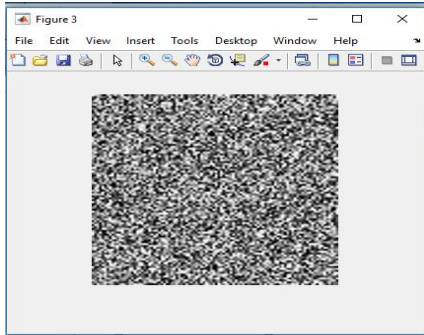
Frame 1



Frame 2



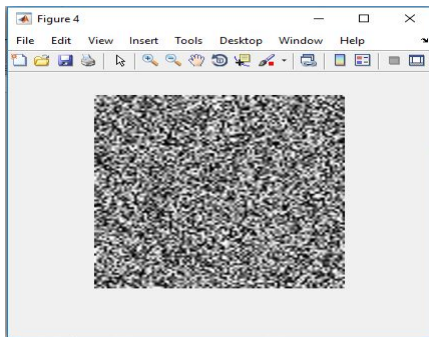
Frame 2



Frame 3



Frame 3



Frame 4



Frame 4

The Table 4.1 depicts the results for the duration of decoding time consumed by the data in the form of text, image and video frames.

Size of Data	Type of Data	No. of errors	% of errors corrected	Time taken
100 KB	Text	7372/8192	100%	0.4 sec
250×250×3 pixels	Image	46875/62500	100%	52 secs
30 video frames of size 352×288 pixels of 1 sec duration	Video	76032/101376	100%	150 secs

Table 4.1 Decoding time

V. CONCLUSION

The encoding and decoding using Turbo codes is carried out for text data and image data and video frames. In text data, the original text data is recovered from error was recovered. The original image “Lena.JPEG” considered for the purpose of experimentation is of size 250×250 pixels. The noisy image obtained after addition of Additive White Gaussian Noise (AWGN), affects 46875 pixels out of 62500 pixels. Using the proposition, the original image is recovered from the noisy image.

The video “foreman.yuv” is considered for encoding and decoding purpose. The video frames considered for experimentation is image each of size 352×288 pixels. All the images are thus encoded and decoded resulting in a sequence of moving images that is a video. The noisy video frame obtained after addition of AWGN, alters 76032 pixels out of 101,376 pixels. Using the proposed decoding technique, the original video frame is recovered from the noisy video frame.

The Mean Square Error value was obtained as zero between the original image and the recovered video frame which implies that the errors are detected and corrected.

VI. REFERENCES

- [1] Next Generation Uplink, Green Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS) Report Concerning Space Data System Standards 230.2-G-1, Jul. 2014.
- [2] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, “Improved low-density parity-check codes using irregular graphs,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 585–598, Feb. 2001.
- [3] R. Gallager, “Low-density parity-check codes,” Cambridge, MA, USA: MIT Press, 1963.
- [4] G. Liva, E. Paolini, B. Matuz, S. Scalise, and M. Chiani, “Short turbo codes over high order fields,” IEEE Trans. Commun., vol. 61, no. 6, pp. 2201–2211, Jun. 2013.
- [5] M. Mooser, “Some periodic convolutional codes better than any fixed code,” IEEE Trans. Inf. Theory, vol. 29, no. 5, pp. 750–751, Sep. 1983.
- [6] C. Weiss, C. Bettstetter, and S. Riedel, “Code construction and decoding of parallel concatenated tail-biting codes,” IEEE Trans. Inf. Theory, vol. 47, no. 1, pp. 366–386, Jan. 2001.
- [7] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” in Proc. IEEE Int. Conf. Commun., Geneva, Switzerland, May 1993.
- [8] Y. Wu and C. N. Hadjicostis, “Soft-decision decoding using ordered recordings on the most reliable basis,” IEEE Trans. Inf. Theory, vol. 53, no. 2, pp. 829–836, Feb. 2007.