

## An Integrated Forward Error Correction Scheme for Broadband Satellite Channels Using Turbo Codes

Nasser Nafaa Khamiss  
College of Information Engineering, Al Nahrian University,  
P.O. Box 64051, Jadriya, Baghdad, Iraq

---

**Abstract:** In this study, a focus on the challenging problem for the satellite data link to support Asynchronous Transfer Mode (ATM) services due to its high error rate and the onboard processing limitation is investigated. This study uses concatenated coding as a means of improving and optimizing the data link performance. The forward error correction as a solution for the burst error characteristic of satellite links and ATM cell format into account is considered. Where the applications of ATM for digital communication system on satellite networks requires system adaptation. This adaptation has to improve the overall system's performance and achieve high quality-of-service classes approaching that for fiber-optic communications. In this study, a new integrated Forward-Error-Correction (FEC) coding scheme is introduced for ATM transmission over regenerative satellite networks. This integrated coding scheme is introduced to significantly improve the cell loss ratio as compared to the standard code used in the ATM cell. Both upper and lower performance bounds for the concatenated code are obtained and check their accuracy, when compared to exact system's performance is achieved. A method of providing FEC for data services uses a parallel concatenated convolutional code, which is a Turbo Code is proposed.

**Key words:** ATM, S-UMTS, traffic parameters, FEC, parallel concatenated code, series concatenated code, soft-input soft-output, LLRs

---

### INTRODUCTION

The future generation of satellite multimedia personal communications requires dynamical control and stringent Quality of Service (QoS) guarantee. It is known that the MAC protocols developments have a dominant effect on the ensuring ability of the QoS and other breakthrough wireless technologies 4G. Customer mobility has emerged as an important catalyst triggering a future generation of the information technologies of knowledge-based society, in which computers and networks will be integrated into the everyday environment, rendering accessible a multitude of services and applications through easy-to-easy human interfaces. In order to carry out these impressing goals, the future generation of global satellite mobile (S-UMTS) and broadband personal communications systems (S-PCS) must provide the advanced multimedia services and e-applications for any mobile and geographical distributed user (personally), at any time (up to real time), at any place (globally), of any kind information (voice, data, video, image, command, positioning, etc.), of any desired Quality of Services (QoS) and of any Traffic Parameters (TP) in a low-cost and mass-market manner (Wang *et al.*, 2003). Broadband satellite communication systems play a more and more important

role in the global information infrastructure but it still remains a challenging problem for the satellite data link to support ATM services because of its high error rate and the onboard processing limitation. A concatenated coding scheme is therefore presented, as forward error correction that can not only get fiber-like performance but also has the ability to correct long burst errors.

Forward Error Correction (FEC) is required in terrestrial and satellite radio systems to provide high quality communication over the RF propagation channel, which induces signal waveform and spectrum distortions, including signal attenuation (freespace propagation loss) and multi-path induced fading. These impairments drive the design of the radio transmission and receiver equipment, the design objective of which is to select modulation formats, error control schemes, demodulation and decoding techniques and hardware components that together provide an efficient balance between system performance and implementation complexity. Differences in propagation channel characteristics, such as between terrestrial and satellite communication channels naturally result in significantly different system designs. Likewise, existing communication systems continue to evolve in order to satisfy increased system requirements for new higher rate or higher fidelity communication services

(Walaa *et al.*, 2005). In the third generation systems, the development of flexible, high-speed data communication services is of particular interest. Desirable features include the ability to perform rate adaptation and to satisfy a multiplicity of Quality Of Service (QoS) requirements (Mertzanis *et al.*, 1999).

**MATERIALS AND METHODS**

**Bent-pipe SATM interconnection architecture:** Satellite as a relay is the simplest SATM architecture, where the satellite link is treated as a communication pipe to replace a terrestrial link and to relay ATM traffic from one remote ATM user to another.

The satellite, however dose not switch ATM cells at ATM layer that is does not Virtual Path (VP) or Virtual Channel (VC) switching.

Furthermore, satellite links are static therefore, there is no need for media access, bandwidth negotiation or handoff. Such an architecture is currently used in GEO satellite (Toh and Victor, 1998). Figure 1 shows the general structure of the bent-pipe SATM interconnection architecture for fixed network, where the ATM protocol is used over broad band integrated service digital network (B-ISDN) (Conte, 2000). The link between two ATM nodes is considered reliable so the correcting error used in ATM packet is sufficient. The protocol reference model for bent-pipe SATM (relay architecture) is shown in Fig. 2, both sides implement basic ATM physical and ATM layer functions.

**Proposed system architecture description:** The usage of the original layers over satellite link (Fig. 2) faces some problems because of the difference between fiber and satellite link channels. Satellite channels have higher BER and errors are in burst manner. As a suggestion a layer for controlling error (ECL) is proposed at GES between ATM and physical layers (Fig. 3).

This new layer, which is a FEC layer is transparent for the satellite, which is proposed to be bent pipe. This layer could be considered as data link layer, it is added and removed at GES. The ATM switches at both sides did not see or deal with this layer.

The frames transmitted over satellite link will be as shown in Fig. 4. This frame consists of header and pay load, the header of 24 bits has three parts; sequence number payload, type and payload size. The pay load is of *i* ATM cells.

The mechanism of transmission is occurred as follows: the GES receive the ATM cells from the ATM node, gather them in groups of *i* cells and add header to each group. The generated frame then coded and

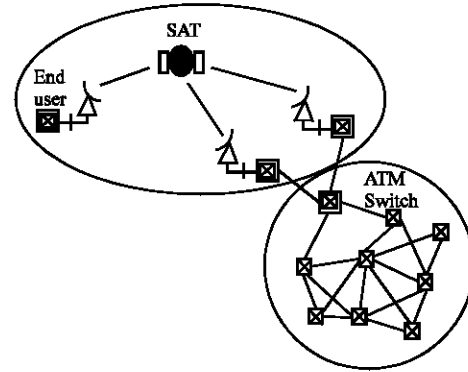


Fig. 1: ATM network architecture (Chai-Keong 1998)

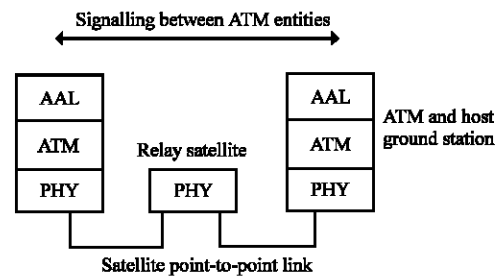


Fig. 2: Satellite ATM layer

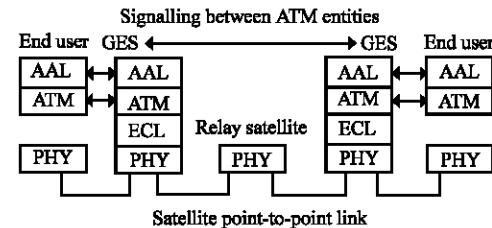


Fig. 3: Propose SATM layer

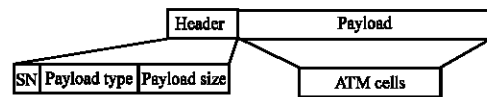


Fig. 4: SATM frame construction

transmitted over satellite link to the other side (Walid and Khamiss, 2008). The destination GES reverse the operations and forward the cells. The advantages of using this layer is reduce the path for resending, instead of making it happen between end users the GES will do the job (Fig. 5).

**System performance mathematical model:** To achieve the mathematical model of system performance, it needs to have an idea about system coder and the types of errors that follow the data transmission.

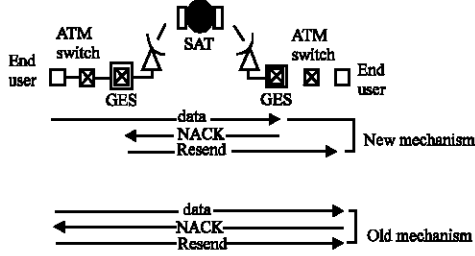


Fig. 5: Mechanism of data transfer (Walid and Khamiss, 2008)

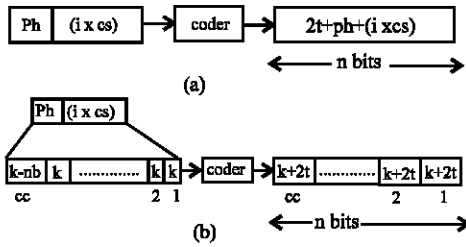


Fig. 6: Frame data coder

The coder is assumed to accept frames with  $i$  cells and their headers (Fig. 6a). Where,  $ph$  is frame header,  $i$  is number of cells,  $cs$  is cell size,  $k$  is sata size,  $t$  is correcting capability and  $P$  is packet size:

$$nb = k + \left( \left\lfloor \frac{P}{k} \right\rfloor \times k \right) - P$$

$$cc = \left\lfloor \frac{P}{k} \right\rfloor + 1$$

The coding rate will be as follows:

$$r = \frac{ph + (i \times cs)}{ph + (i \times cs) + 2 \times t} \quad (1)$$

Some time, the coder is assumed to accept data with size depends on the coding rate of the coder, where the output size is fixed of 255 bits. The frame that to be coded will be divided into small segments of  $k$ -bits of  $k = 255 - 2t$  (Fig. 6b).

Generally errors of channels are mainly of two type, channel usually has lower BER and errors are random, i.e., statistically independent in Fig. 7a and channel has higher BER and errors are in burst manner because of fading and multipath in Fig. 7b.

Even though that the used channel is satellite channel, where errors occur are burst, the start step of error analysis will be of random expression forum. It is assumed to be evenly distributed instead of statistically

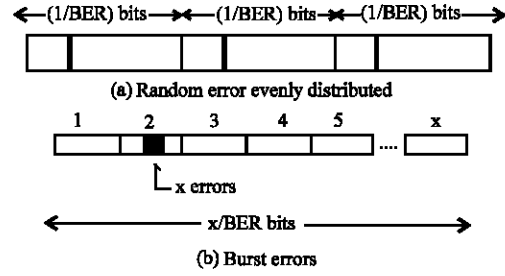


Fig. 7: Error behavior

independent. Where for each error occurs, there are  $\{(1/BER)-1\}$  correct bits. The next step in analyzing the system is the introduction of burst error of even distribution model, which describe error behavior over satellite channel. The burst errors of length  $x$  is defined as a sequence of  $x$  bit errors, the first and last of which is 1 s (Proakis, 2000). So by scaling for  $x$  bits of error and error probability of BER we will have total bits  $(x/BER)$  and the error free bits will be the difference between total bits and the error bits.

The burst scenario here assumed to be evenly distributed and the random in position will be described next. Assuming error probability (or bit error rate) equals BER. This mean that for each one error we have  $(1/BER)$  bits, the error free bits will be  $\{(1/BER)-1\}$ . So by scaling for  $x$  bits of error and error probability of BER, we will have total bits  $(x/BER)$  and the error free bits will be the difference between total bits and the error bits (Fig. 6b).

$$\text{Error free bits} = \text{Total bits} - \text{Error bits} \quad (2)$$

$$= \frac{x}{BER} - x \quad (3)$$

$$\text{Error free bits} = \frac{x \times (1 - BER)}{BER} \quad (4)$$

The random error could be considered as a special case of burst error when the length of the burst ( $x$ ) is = 1.

**System performance:** In this study, two scenarios of errors will be discussed. The coder first analyzed in present of random error then it analyzed with burst error. The final step includes system performance evaluation. It is good to know that in spite of the disadvantages of coder that it is complex and slow but it could correct more errors that results of improving the system efficiency.

**Coder with random errors:** As stated before the random error means that in each  $1/BER$ , there is only one bit error

and the word random refer to the position of the error bit. In such case, if the error occurs in one frame is less than or equal to the correcting capability the frame will correct it and passes it forward, else the frame will be discarded. The reason behind is the retransmission mechanism will not be useful because the retransmitted frame will be in error that may be continued for long time. For the 2nd case, the efficiency will degrade to zero for the link (for that frame size and that BER). The efficiency is defined as:

$$\text{Efficiency} = \frac{\text{Useful data received}}{\text{Total data transmitted}} \quad (5)$$

So, for one frame it will be:

$$\eta = \frac{i \times cs}{n} \quad (6)$$

Where:

$\eta$  = Efficiency

$n$  = Number of bits after the coder

The efficiency for the two cases could be presented as:

$$\eta = \begin{cases} 0 & \text{when } \dots NEP > t \\ \frac{cs \times i}{n} & \text{when } \dots NEP < t \end{cases} \quad (7)$$

$$NEP = P \times BER \times r \quad (8)$$

Where:

NEP = Number of error in one frame

P = Frame size

Relation between cells number and efficiency for large coder with random errors  $t = 10$  and  $BER = 10^{-2}, 10^{-3}$  and  $10^{-4}$  is shown in Fig. 8.

**Coder with burst errors:** In burst errors, efficiency limitation will be removed because the retransmission mechanism will work as will be described in this study. The efficiency for given P, BER and t is equal to the summation of the efficiencies for one, two, three etc. number of errors as follows:

$$\eta = \lim_{e \rightarrow \infty} \frac{\sum_{n=1}^e \eta_n}{e} \quad (9)$$

But since the data send is divided into frames, each one is coded alone, the errors more than one frame will do

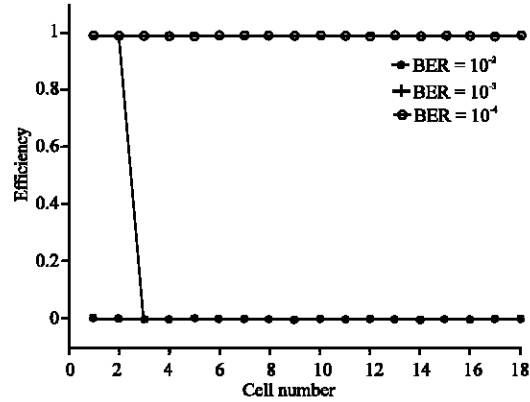


Fig. 8: Relation between cell number and efficiency

the effect of e-P errors because errors  $> 2t$  will not be seen and this model will have repetitive behavior. So,

$$\eta = \frac{\sum_{e=1}^P \eta_n}{P} \quad (10)$$

Upon the number of the errors and the value of t, the efficiency could be treated into three categories. The first category is when  $e \leq t$ , which is within the system capability of error correction. The second category is when  $t \leq e \leq 2t$ , which is within the system capability of error detection where an NACK will be transmitted to the source to retransmit that frame (Walid and Khamiss, 2008).

The third category is when  $e > 2t$ , here the errors will behave like second group, where only one frame will be not correct and the rest will be correct as shown in following analysis.

For the last two cases, retransmission techniques (ARQ) to recover from channel errors are a feasible solution, so for one frame it will be:

$$\eta = \frac{(i \times cs)}{n} \quad (11)$$

While when retransmission occurs, the efficiency is equal to the number of useful frames to the total frames sent as follows:

$$NP = \frac{1}{BER \times n} \quad (12)$$

NP is the number of frames per one error

$$\eta = \frac{\sum_{e=1}^{e=t} \eta_e + \sum_{e=t+1}^{e=2t} \eta_e + \sum_{e=(2t)+1}^n \eta_e}{n} \quad (13)$$

$$\eta = \frac{\sum_{e=1}^{e=t} \frac{(i \times cs)}{n} + \sum_{e=t+1}^{e=2t} \frac{(i \times cs)}{n} \times \frac{(e \times NP) - 1}{(e \times NP)} + \sum_{e=(2t+1)}^n \frac{(i \times cs)}{n}}{n} \quad (14)$$

$$\left( \eta = \frac{(i \times cs)}{n^2} \left[ \sum_{e=1}^{e=t} 1 + \sum_{e=t+1}^{e=2t} \frac{(e \times NP) - 1}{(e \times NP)} + \sum_{e=(2t+1)}^n 1 \right] \right) \quad (15)$$

$$\eta = \frac{(i \times cs)}{n^2} \left[ t + n - (2 \times t + 1) + 1 + \sum_{e=t+1}^{e=2t} 1 - \frac{1}{(e \times NP)} \right] \quad (16)$$

$$\eta = \frac{(i \times cs)}{n^2} \left[ n - t + 2 \times t - (t + 1) + 1 + \frac{1}{NP} \times \sum_{e=t+1}^{e=2t} \frac{1}{e} \right] \quad (17)$$

$$\eta = \frac{(i \times cs)}{n^2} \left[ n - \frac{1}{NP} \times \sum_{e=t+1}^{e=2t} \frac{1}{e} \right] \quad (18)$$

For real efficiency

$$\eta = \frac{\sum_{e=1}^{e=t} \eta_e + \sum_{e=t+1}^{e=2t} \eta_e + \sum_{e=(2t+1)}^n \eta_e}{n} \quad (19)$$

$(e \times NP)$  = Represent the number of total frames send for given number of error e.

$(e \times NP) - 1$  = Represents the correct frame where the (1) represent the retransmitted frame

In this case,  $\eta$  for the interval  $(t+1-2t)$  is equal to that for interval  $(2t-n)$  in spite of the difference in cause.

$$\eta = \frac{\sum_{e=1}^{e=t} \frac{(i \times cs)}{n} + \sum_{e=t+1}^{e=n} \frac{(i \times cs)}{n} \times \frac{(e \times NP) - 1}{(e \times NP)}}{n} \quad (20)$$

$$\eta = \frac{(i \times cs)}{n^2} \left[ \sum_{e=1}^{e=t} 1 + \sum_{e=t+1}^{e=n} \frac{(e \times NP) - 1}{(e \times NP)} \right] \quad (21)$$

$$\eta = \frac{(i \times cs)}{n^2} \left[ t + n - (t + 1) + 1 - \sum_{e=t+1}^{e=n} \frac{1}{(e \times NP)} \right] \quad (22)$$

$$\eta = \frac{(i \times cs)}{n^2} \left[ n - \frac{1}{NP} \times \sum_{e=t+1}^{e=n} \frac{1}{e} \right] \quad (23)$$

Relation between cells number and efficiency for large coder with burst errors  $t = 10$  and  $BER = 10^{-2}, 10^{-3}$  and  $10^{-4}$  of apparent and real one are shown in Fig. 9 and 10, respectively. For the purpose of comparison between apparent efficiency and real, one for the same condition of test is given in Fig. 11.

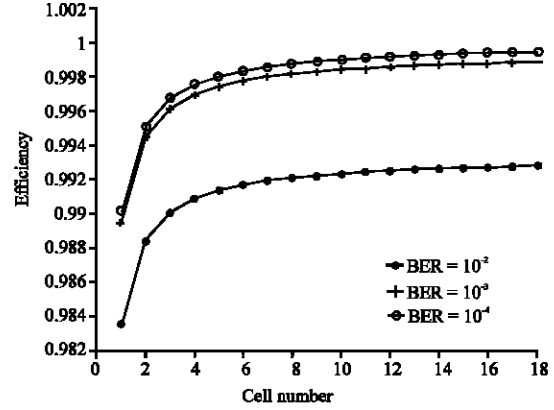


Fig. 9: Relation between cells number and apparent efficiency

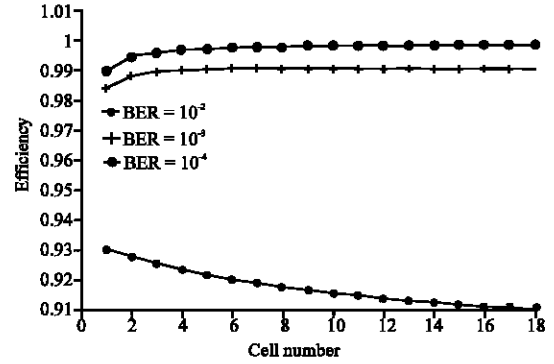


Fig. 10: Relation between cells number and real efficiency

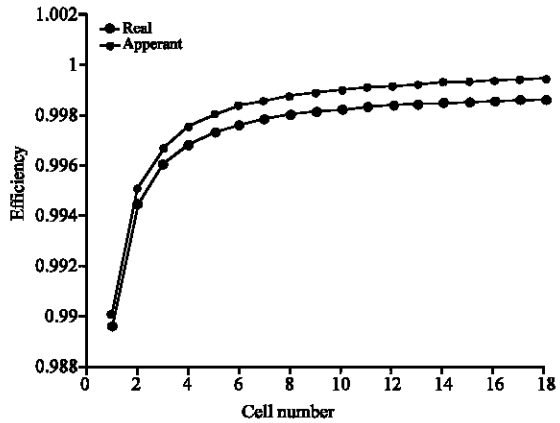


Fig. 11: Relation between cells number and apparent efficiency and real efficiency

**Encoding operation:** In this study, a method of providing forward error correction for data services uses a parallel concatenated convolutional code, which is Turbo Code is used. The present error correction in data

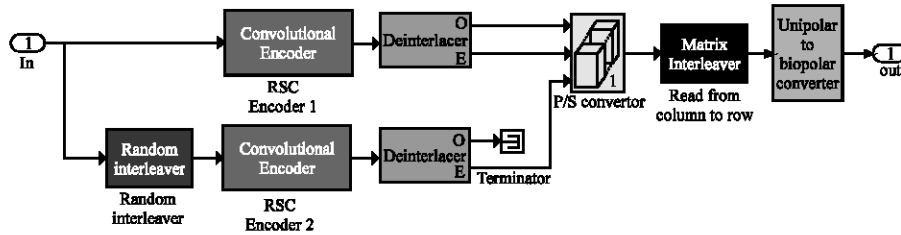


Fig. 12: Turbo code encoder

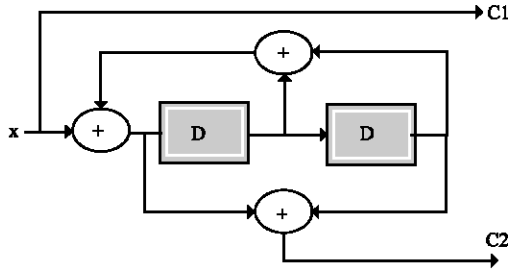


Fig. 13: The RSC encoder with  $r = 1/2$  and  $k = 3$

communications and more particularly, to Forward Error Correction (FEC) relates the selection and use of optimal Turbo Codes in high performance data communication systems, such as emerging third generation terrestrial cellular mobile radio and satellite telephone systems, for which flexibility in supporting a wide range of system requirements with respect to transmission data rates, channel coding rates, quality of service measures (e.g., latency, bit-error rate, frame error rate) and implementation complexity are highly desirable.

**Error control layer design:** A Turbo encoder consists of a Parallel Concatenation (PCCC) of typically two systematic, recursive convolutional codes (constituent codes) separated by an interleaver that randomizes the order of presentation of information bits to the second constituent encoder with respect to the first constituent encoder is shown in Fig. 12. The two encoders are identical and built based on the RSC encoder of Fig. 13. The performance of a Turbo Code depends on the choice of constituent codes, interleaver, information block size (which generally increase with higher data rates) and number of decoder iterations. For a particular Turbo Code, in which the constituent codes are fixed, one can ideally adjust the block size and number of decoder iterations to trade-off performance, latency and implementation complexity requirements. As the block size changes, however, a new interleaver matched to that block size is required

Choosing a good interleaver design, is important for obtaining a good Turbo code performance. Therefore, pseudo-random interleaver is used through the work.

Other significant parameter relating to the interleaver is its size. As the interleaver size increases, the system performance improves. However, there is a tradeoff between performance and time (delay) since both of them are directly proportional to the size. Also the key role of the interleaver is to shape the weight distribution of the code, which ultimately controls its performance. This is because the interleaver will decide, which word of the second encoder will be concatenated with the current word of the first encoder and hence what weight the complete codeword will have (Moreira and Farrell, 2006).

In this research, a code rate of  $1/2$  is considered over the ordinary non puncturing code (code rate =  $1/3$ ), which will produce a more flexible code that results in good performance particularly at low Signal to Noise Ratios (SNRs) as will be shown through the results. The information bits are always transmitted across the channel.

Depending on the desired code rate, different code rates are achieved by puncturing the parity bit sequences from the two constituent encoders. As the code rate increases, the bandwidth efficiency will be improved and the performance is degraded since the decoder has less information to use in making a decision. Therefore, a tradeoff must be made between the code rate and the performance.

The deinterlacer block accepts the input vector that has an even number of elements and alternately places the elements in each of two output vectors. Therefore, it is used to separate the systematic and the parity bit of each SC encoder. As mentioned previously, the systematic bit of the second encoder is nothing more than a repetition code, thus a termination block is used on the odd output of the second encoder.

The three streams: the systematic bits and the two parity bits are concatenated using vertical concatenation. A matrix interleaver is used to perform block interleaving by filling a matrix with the input symbols row by row and then sending the matrix contents to the output port column by column so as to avoid burst errors. The output then forwarded to the puncture block which will periodically remove bits from the encoded bit stream, thereby increasing the code rate. The puncture pattern is

specified by the vector parameters, which is  $= [1 \ 1 \ 0 \ 1 \ 0 \ 1]$  for the proposed Turbo code and thus, the code rate of the Turbo code increased from  $1/3$ - $1/2$ .

**Turbo decoding:** The truly unique aspect of Turbo codes is their iterative decoding process. The iterative decoding structure consists of two Soft-Input, Soft-Output (SISO) decoding modules that are separated by a pseudo-random interleaver/deinterleaver.

The performance analysis of Turbo codes always assumes the usage of a Maximum Likelihood (ML) decoder at the receiver for efficient data recovery. The output of each encoder depends on the last input bit and the generator matrix, which enables the encoding process of the Turbo code to be represented by two joint Markov processes. It is possible to decode Turbo codes, first by independently estimating each process and then refining the estimates by iteratively sharing information between two decoders (Avril, 2007). Since the two processes run on the same input data, it means that the output of one decoder can be used as a priori information by the other decoder. It is necessary for each decoder to produce soft-bit decisions in order to take advantage of this iterative decoding scheme. The soft-bit decisions are usually in the form of Log Likelihood Ratios (LLRs). The LLR data serves as the a priori information and is defined as the likelihood of the received bit being a one rather than a zero, where the decision 1 is made for a positive LLR and the decision 0 is made for the negative LLR.

A decoder that accepts input in the form of a priori information and produces output in the form of posteriori information is called a Soft Input Soft Output (SISO) decoder. The inputs to the decoder are systematic data, parity data and the a priori data from the previous decoder and the output of the decoder is the LLR data. The generic block diagram of a SISO decoder is shown in Fig. 14.

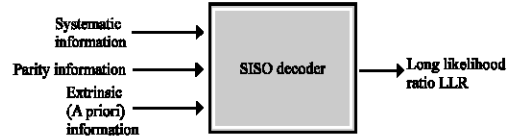


Fig. 14: Block diagram of SISO decoder

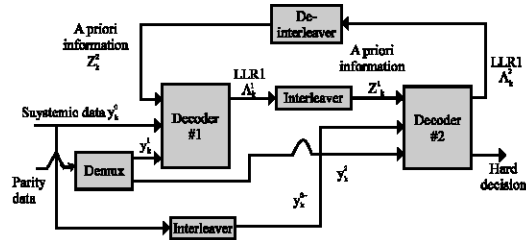


Fig. 15: Turbo decoder

Where:

- $a$  = Fading amplitude
- $E_s$  = Average symbol energy
- $N_o$  = Noise Power Spectral Density (PSD)

The channel reliability places more emphasis on the channel observation, when the SNR is high and there is no fading. Likewise, more emphasis is placed on the a priori information  $Z_k$ , when the SNR is poor or when there is a deep fade.

The output of the SISO decoders is expressed as a Log-Likelihood Ratio (LLR)  $\Lambda_k$ , where the decoder's output at time  $k$  can be broken down into three distinct parts: the scaled systematic channel estimate  $4aE_s/N_o \cdot y_k^o$ , the a priori information  $Z_k$  and the extrinsic information  $l_k$ . The  $k^{th}$  LLR is expressed as:

$$\Lambda_k = \frac{4aE_s}{N_o} y_k^o + z_k + l_k \quad (24)$$

**Operation of turbo decoding:** A block diagram of a Turbo decoder is shown in Fig. 15, which consists of two component decoders decoder #1 to decode data sequences from encoder 1 and decoder #2 to decode sequences from encoder 2. The first decoder operates on the systematic channel observation  $y_k^o$ , the parity channel observation from the first RSC encoder  $y_k^p$  and the a priori information  $Z_k^1$ . The a priori information for SISO (decoder #1) is initially set to all zeros, since the second decoder has not produced any information. This implies that each information bit is equally likely to be a 0 or a 1 initially. Both channel observations are multiplied by the channel reliability:

$$L_c = \frac{4aE_s}{N_o}$$

The extrinsic information is the new information generated by the current stage of decoding. In Turbo decoder, the extrinsic information for the first decoder is determined by subtracting the systematic channel observation and the current stage's a priori information from the LLR  $\Lambda_k^1$ , thereby preventing positive feedback. The extrinsic information is then permuted by pseudo-random interleaver and used as the weighted a priori information for the second decoder. The second decoding module operates on the weighted a priori information from first decoder  $y_k^2$ , the permuted channel observation  $\pi y_k^o$  and the parity channel observation from the second RSC encoder  $y_k^p$  that generate a new LLR  $\Lambda_k^2$ . The first decoder is presented with the result from the second decoder one can imagine that it might improve its performance,

compared to its first decoding attempt. The two decoders iteratively exchange this extrinsic information and improve their estimates about the decoded bits. If all the decoding iterations have been completed, the final output  $\Lambda_k^2$  is deinterleaved and hard-limited to produce the final decision. The iterative decoding process improves the BER performance of Turbo codes in a superior way (Neubauer *et al.*, 2007).

As stated before, when Berrou and Glavieux achieved  $BER = 10^{-5}$  at  $E_s/N_0$  within 0.7 dB of the Shannon limit using a rate 1/2 turbo code, they used 18 decoding iterations.

**Parallel concatenated code:** The receiver error handling mainly consists of two parts, which are the Receiver Front End and the Turbo Decoder block as shown in Fig. 16.

As shown in Fig. 17, the data is divided by the noise variance through the gain block, then sampled and held for a specified sample period by the zero-order hold block. After that, a matrix deinterleaver block is used, which fills the input symbols into a matrix column by column and then sending the matrix contents to the output port row by row.

Two interleaver blocks are used to reconstruct the data as produced by the two de-interleavers in the transmitter side, then the output of the both interleaver is forwarded to the Turbo decoder block.

Figure 18 shows the PCCC decoding process that consists of two APP decoder blocks, a random interleaver

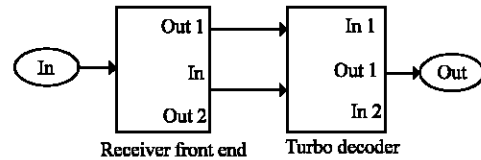


Fig. 16: Turbo decoder main blocks

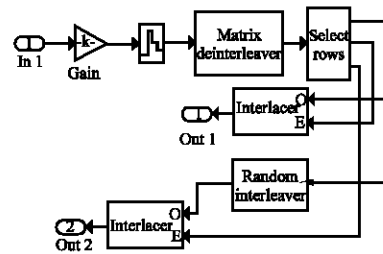


Fig. 17: Receiver front end

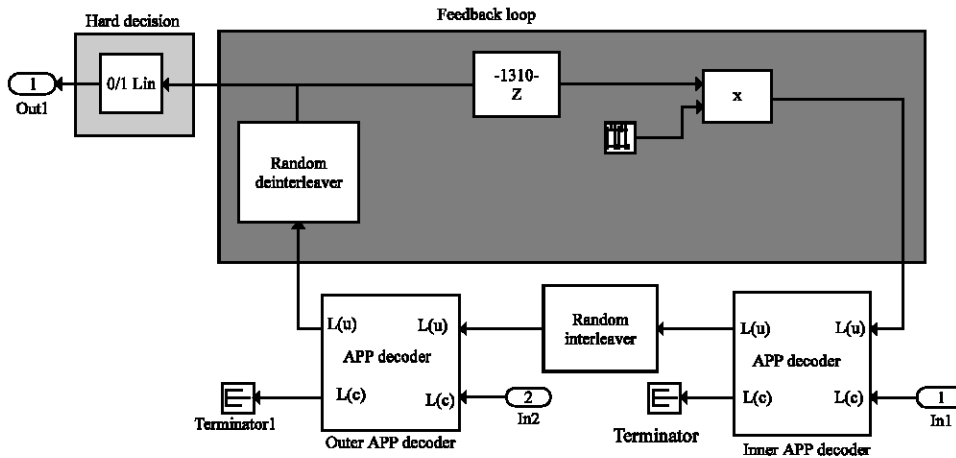


Fig. 18: Turbo code decoder

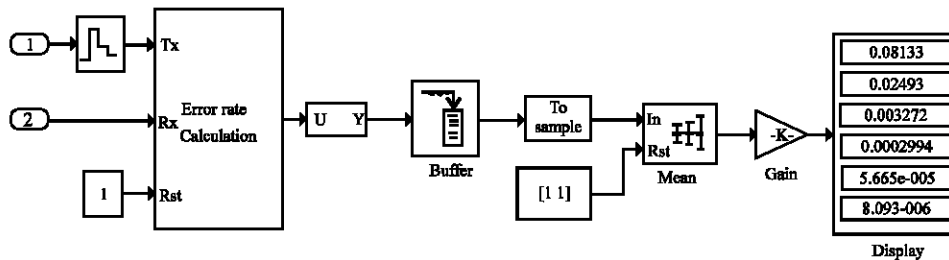


Fig. 19: Multiple iterations error rate calculation



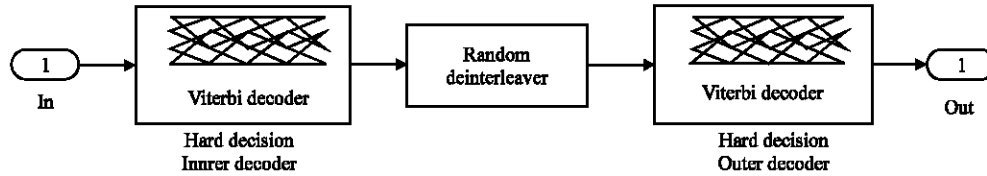


Fig. 20: Hard decision decoder

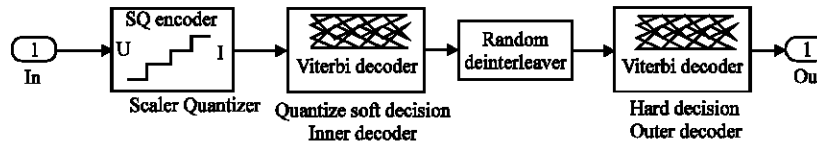


Fig. 21: Quantize soft decision decoder

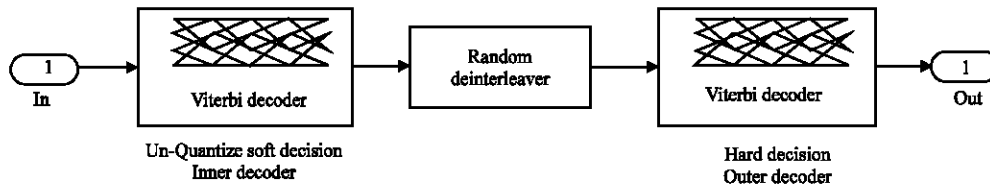


Fig. 22: Un-quantize soft decision decoder

and a feedback loop. As in SCCC, these blocks form a loop and operate at a rate of six times faster than the encoding portion.

The error rate block is the same as that used in SCCC system. As shown in Fig. 19, the data are sampled and held for the specified sample period by a zero-order hold block. The error rate is calculated for all iterations by comparing the received data with the transmitted data and the output is converted to six independent channel samples. Then, a mean block is used to return the mean of the input elements over the time. Finally, the display block shows error rates of the six iterations, where the final BER is obtained from the last iteration.

**System evaluation and results**

**System evaluation:** To evaluate the proposed FEC system, deep analysis of the decoding part were achieved, where a four scenarios were simulated and tested over the satellite link beside the PCCC, these are the hard decision, the quantized soft decision, the un-quantized soft decision and the serial concatenation convolution decoders. There different decodes will be briefly described in the following sub-sections.

Figure 20 shows the decoding operation of the proposed Hard Decision system, where the data of the inner encoder is decoded by the inner hard decision viterbi decoder then de-interleaved before sending it to

the outer decoder, which will recover the original data packet. Viterbi decoder is used to decode the convolution encoded signal by finding an optimal path through all the possible states of the encoder.

Figure 21 shows the decoding operation of Quantized Soft Decision system. The LLR values produces by the soft decisions QAM demodulator are mapped to the right quantizer index for use with Viterbi decoder.

Figure 22 shows the decoding operation of Un-Quantized soft decision system. In this system, the inner soft viterbi decoder accepts un-quantized soft data, which is an infinitely fine quantization.

Therefore, it is used to calculate theoretical bounds for the bit error rate of the soft convolutional code. However, this system couldn't implement practically, it is just used in this simulation for comparison with other proposed system.

Figure 23 shows the SCCC decoding, which consists of two A Posteriori Probability (APP) decoder blocks, a random deinterleaver and several other blocks. Together, these blocks form a loop and operate at a rate six times faster than that of the SCCC encoding portion, where in this model six iterations are used. The loop is structured to make the decoding portion an iterative process, where the APP decoder block is used to decode the convolutional code. Unlike viterbi decoder, the APP decoder accepts soft input and produces soft output.

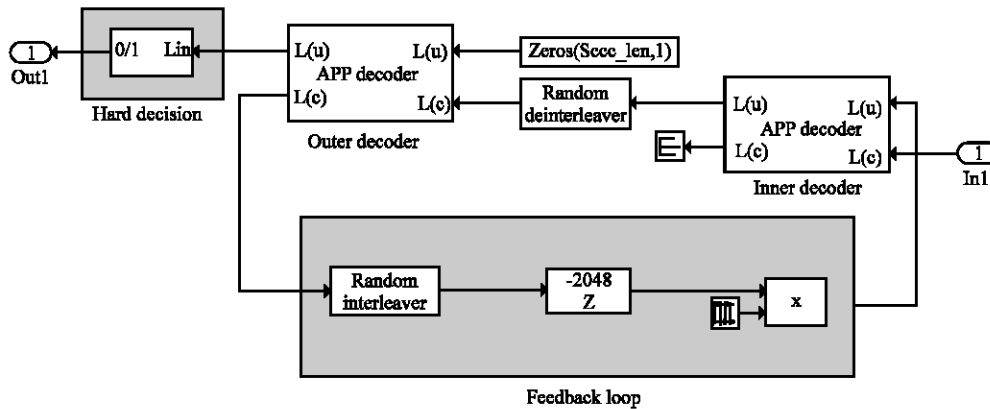


Fig. 23: Serial Concatenation Convolution Code (SCCC) decoder

**RESULTS AND DISCUSSION**

In Fig. 24, the BER is compared between un-coded and coded system, where it achieves better performance of 3 dB gain over the un-coded one. The Fig. 3 also shows the power of concatenation code, where a gain of 1.7 dB is achieved over the non concatenation code by using the same parameter for encoder and decoder. The decoder that accepts LLR from the soft decision demodulator achieved better BER, compared with the ordinary hard decision demodulator for the same SNR values as show in Fig. 25, it also shows that the Un-Quantized soft decision scheme achieves better performance of 1 dB compared with the Quantized soft decision scheme.

The introducing of iterative decoding is presented in Fig. 26 and 27, which show the BER versus SNR for six iterations of PCCC and SCCC error correction systems, respectively.

The first error rate reflects the performance of the decoding process that uses one iteration; the second error rate reflects the performance of the decoding process that uses two iterations and so on. The series of error rates calculation shows that the error rate generally decreases as the number of iterations increases. Then, a comparison between PCCC and SCCC is made as shown in Fig. 28, where the target  $10^{-6}$  BER is achieved for PCCC at 26.5 SNR, while for SCCC it needs 28.2 SNR.

Figure 29 elucidates the BER performance for all the proposed error correction systems compared with the Un-coded system.

The results illustrate the gain of using Hard decision error correction system over the Un-coded one, also show the improvement in BER performance, when using Log-Likelihood Ratio (LLR) in the Quantized soft decision instead of the Hard decision scheme. Finally the iterative process which composed of SCCC and PCCC gives the highest gain compared with the non iterative systems

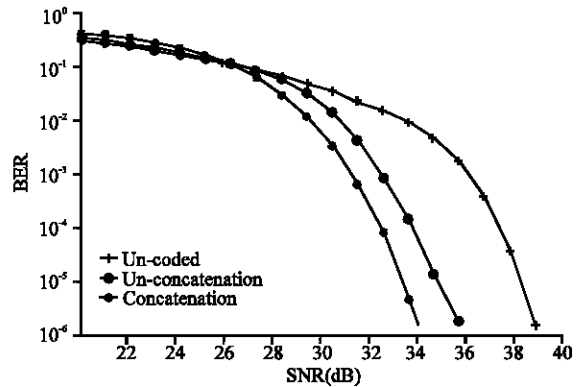


Fig. 24: BER comparison of un-coded, un-concatenation with concatenation convolution code

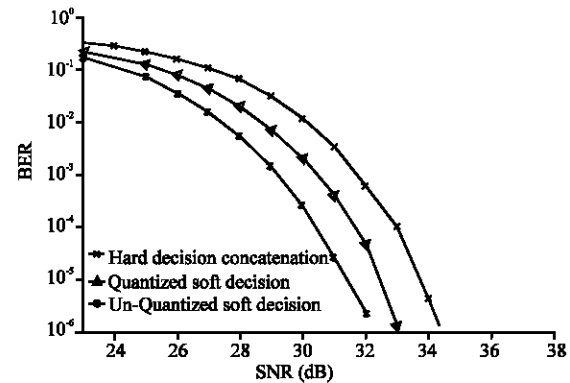


Fig. 25: BER comparison for Hard and Soft decision error correction system

above, where the Turbo code (PCCC) provides significant improvement versus the SCCC system in Fig. 30.

For Turbo code, a rate of  $r = 1/2$  can be achieved by puncturing the parity bits of the constituent encoders as shown in Fig. 30.

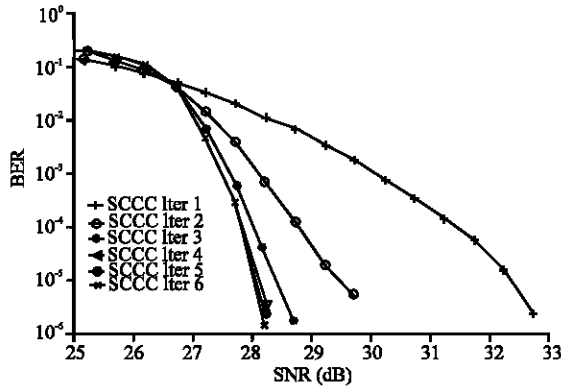


Fig. 26: The BER of serial concatenation convolution code (SCCC)

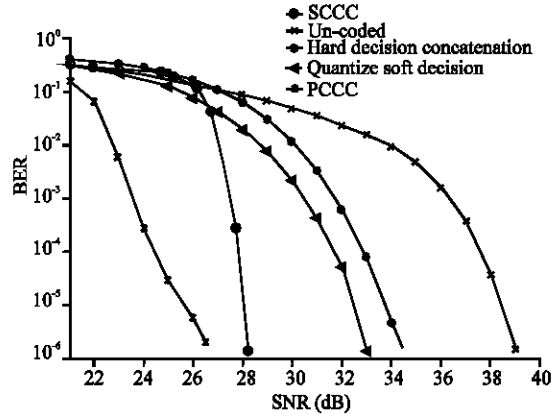


Fig. 29: BER comparison of all proposed error correction systems

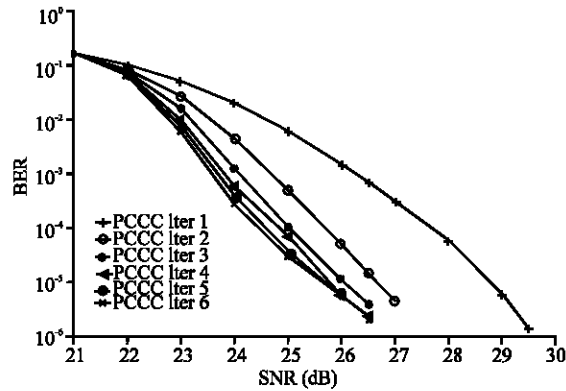


Fig. 27: Turbo code BER calculation

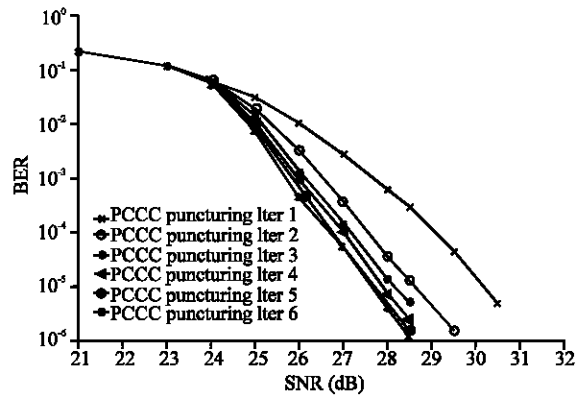


Fig. 30: Puncturing Turbo code (code rate 1/2)

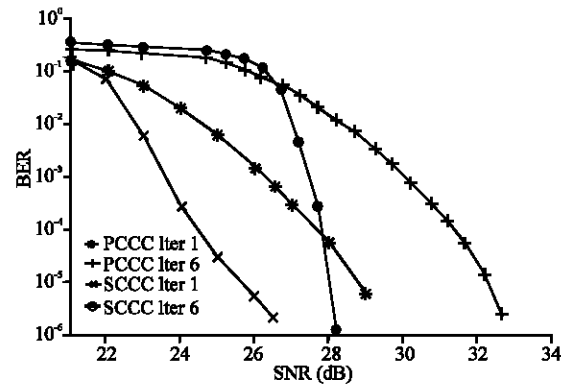


Fig. 28: BER comparison between PCCC and SCCC error correction systems

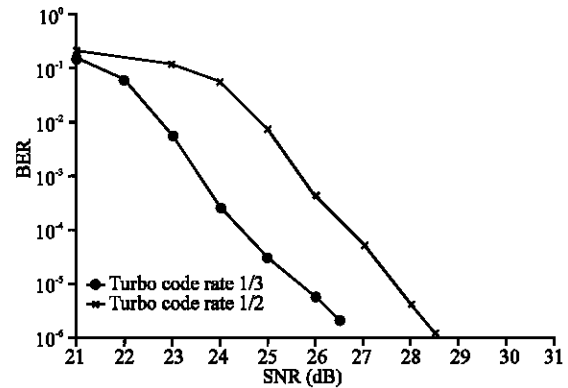


Fig. 31: BER comparison of puncturing Turbo code (code rate 1/2) with the Un puncturing Turbo code (code rate 1/3)

It can be seen that the increasing in code rate, decreased the system performance. Figure 31 displays a comparison between puncturing ( $r = 1/2$ ) and unpuncturing ( $r = 1/3$ ) Turbo code system, where the performance is decreased by 2 dB.

### CONCLUSION

Through the applying of concatenation code over satellite link, the following conclusions have been reached:

- The concatenation code scheme achieved better performance by 1.7 dB gain over the non concatenation scheme as shown in Fig. 24
- The Quantized soft decision system that used Log Likelihood Ratio (LLR) as soft decision demodulator algorithm achieved 1.3 dB gain over the ordinary hard decision demodulator for the same SNR values, as shown in Fig. 25
- The introduction of iterative decoding technique achieved an improved performance comparing with the general concatenation code. Figure 26 and 27 show that when the number of iterations increased, the performance gain is also increased
- The comparison between PCCC and SCCC systems is shown in Fig. 28, where the PCCC achieved 1.7 dB gain over the SCCC system
- For Turbo code system, it can be seen that increasing the code rate from 1/3-1/2, decreased the system performance by 2 dB as shown in Fig. 31
- Return back to the error model, it is clear that the proposed ECL can face the problem of burst error
- To get a better target of BER, the satellite modem link required higher SNR value, when using the designed modem, due to the satellite channel nature that suffers from different kinds of noise, attenuation and the uncontrolled properties
- The goal of getting fiber-like performance, it is actually the way of achieving high scalability of integrated network for high bit rate applications
- The proposed architecture reduces the RTT

## REFERENCES

Avril, G., F. Gauthier, F. Moulin, A. Zeddani and F. Nouvel, 2007. Characterization of time variation of the powerline channel frequency response simultaneously with impulsive noise. Proceedings of the IEEE International Symposium on Power Line Communications and Its Applications INSPEC, Mar. 26-28, Pisa, pp: 330-335.

- Conte, R., 2000. Satellite rural telephone network design: A methodology for performance optimization. Ph.D. Thesis, Virginia Polytechnic Institute and State University, September 2000.
- Mertzanis, L., G. Sfikas, R. Tafazolli and B.G. Evans, 1999. Protocol architectures for satellite ATM broadband networks. *IEEE Commun. Mag.*, 37: 46-54.
- Moreira, J.C. and P.G. Farrell, 2006. *Essentials of Error-Control Coding*. John Wiley and Sons Ltd., England, ISBN-10: 047002920X, pp: 209-271.
- Neubauer, A., J. Freudenberger and V. Kuhn, 2007. *Coding Theory: Algorithms, Architectures and Applications*. John Wiley and Sons Ltd., England, ISBN: 978-0-470-02861-2, pp: 163-212.
- Proakis, G.J., 2000. *Digital Communications*. 4th Edn., MacGraw-Hill, New York, ISBN: 0072321113.
- Toh, C.K. and O.K. Victor, 1998. Satellite ATM network architectures: An overview. *IEEE Network*, 12: 61-71.
- Walaa, H., Mclane and Peter, 2005. An integrated FEC coding scheme for ATM transmission over regenerative satellite networks. *Int. J. Satellite Commun. Network.*, 23: 33-46.
- Walid, K. and N. Khamiss, 2008. Performance mathematical model of developed ATM satellite link. Proceedings of the 1st Regional Conference of Engineering Sciences, Nahrain University, Iraq.
- Wang, K., B. Tian and K. Yi, 2003. A practical scheme of improving the link performance for ATM satellite communication networks. Proceeding of the 17th International Conference of Advanced Information Networking and Applications, March 27-29, Xi'an, China, pp: 472-475.