Performance Evaluation of Convolution Encoder and Viterbi Decoder in Terms of Gap and Coding Gain Capicity for 1/N Code Rate Using Matlab

Parveen Sharma[1], Ms. Kajal[2]

[1]M. Tech Student, ECE Department, RN College of Engineering & Technology, Rohtak
[2]Assistant Professor, ECE Department, RN College of Engineering & Technology, Rohtak

ABSTRACT

In today‘s era, efficient transmission is very important. In case of long distance transmission wired approach is adopted. But in wireless approach, efficient transmission as well as efficient reception is required. Moreover, the channel through which the communication is taking place has to be considered efficiently. The transmitted information will get modulated according to the various modulation techniques. Convolutional encoder is one of the techniques, which is used to correct erroneous bits at the receiver end. This technique is also called as forward error correction technique. At the receiver end to decode the convolutional codes, Viterbi decoding technique is used. The Viterbi decoder Algorithm is widely used for estimating and detecting problems in signal processing and digital communications. This algorithm is used to detect signals in communications channels with memory, and to decode the sequential error control codes which results in the enhancement of the performance of digital communication systems. The applications of the Viterbi decoding algorithm are: digital TV (QAM, ATSC, and DVB-T), satellite communications and radio relay.

I. INTRODUCTION

With the advancement in technology, people are looking for the simpler methods and cost effective technique for communication. SDR (software defined radio) is a technique which not only brings the flexibility but also it is cost effective and have power to drive communications forward with wide-reaching benefits realized by service providers and product developers through to end users[1]. Software defined radio (SDR) forum in collaboration with IEEE P1900.1 group, has given a clear overview of SDR technology and its benefits. Software Defined Radio is defined as: "Radio in which some or all of the physical layer functions are software defined"[1]. In SDR all of the radio‘s operating functions are implemented with the help of modifiable firmware operating on programmable processing technologies. These devices cover FPGA (field programmable gate arrays), DSP (digital signal processing), GPP (general purpose processors), SoC (programmable System on Chip) or many other programmable processors [1].

This paper describes the design and implementation of Convolutional encoded communication system based on a FPGA based platform. Detailed implementation results (performance, code size, and FPGA resources utilization) are presented. The main goal of the design case presented is to provide insight into the design aspects of a complex system based on FPGAs [2]. The results prove that an implementation based FPGAs are adequate for the communication system where the expected volumes are rather small. Convolutional encoding is a forward error correction technique that is used for correction of errors at the receiver end. Viterbi decoding is the technique for decoding the convolutional codes [3]. The Viterbi Algorithm, an application of dynamic programming, is widely used for estimation and detection problems in digital communications and signal processing. The paper aimed at the real time implementation of convolutional encoder and Viterbi decoder over SDR using QPSK digital modulation techniques.

Purpose

As we move onto the third generation, the reality is that first generation 3G systems did not satisfy the customer requirement of high-speed transmission, and the rates supported in practice were much lower than that claimed in the standards [4]. Enhanced 3G systems were subsequently deployed to remove the deficiencies. But, the data rate capabilities and network architecture of these systems are insufficient to address the unsatisfied consumer. With these considerations in hand, the move to 4G technologies like 3G LTE (long term evolution) and WiMAX is proceeding at an extremely rapid rate. The goal of next generation systems is to provide high-data rate, low-latency, high reliability (minimize outages and connection drops) employing packet-optimized radio access technology supporting flexible bandwidth allocation. Additional key objectives are to drive down the cost of infrastructure equipment and consumer terminals and to employ a more efficient modulation scheme than the CDMA technology used in 3G systems in order to make more optimal use of precious communication bandwidth [5].

To meet all of these requirements a significant re-structuring of both the physical layer (PHY) and network architecture is required. The initial intent of this paper is to generate the random bits for finite time period. Secondly, to design convolutional encoder and to generate an AWGN noise channel, so that the communication takes place through this medium. Thirdly, to get the RRC filter, used for the modulation purpose. Fourthly, combine the modulator, channel and Viterbi decoder and demodulator part together to get the original message at the receiver end. Convolutional encoding is an FEC technique that is particularly suited to a channel in which the transmitted signal is corrupted mainly by additive white Gaussian noise (AWGN). Viterbi algorithm is a well-known Maximum-likelihood algorithm for decoding of Convolutional codes. They have rather good correcting capability and perform well even on very noisy channels. It has been widely deployed in many wireless communication systems to improve the limited capacity of the communication channels [6]. Field programmable gate arrays (FPGAs) with their inherently parallel structure, are increasingly the technology of choice for addressing the compute and flexibility requirements of next generation systems. FPGA advantage for its friendly VHDL language interface and was chosen for RTL implementation of encoder and decoder.

CONVOLUTIONAL ENCODER AND VITERBI DECODER

Communication means transfer of information from the transmitter end to the receiver end through any medium. The medium may be air, wired. Due to the presence of noise, at the receiver end information will get noisy and actual information will not be received. So, at the receiver end their will be some errors. These errors can be detected and sometimes it‘s also possible that these errors can be corrected using the coding techniques.

Coding

The error control methods are generally of two types:-

- Error detection with retransmission
- Forward error correction

In error detection with retransmission method, when the receiver end detects the error acknowledgement is sent to the transmitter to send the data again as the previous was corrupted. Whereas in forward error correction method, proper coding techniques are used to detect and correct the errors. Coding technology uses a proper techniques of adding some extra bits to the transmitted message bits and these redundant bits are used to the detect errors at the receiver side. But addition of redundant bits results in decrease in bit rate at the transmitter and its power. But also have advantage that probability of error is less. Extra bits are added with the original message signal by the channel encoder. Whereas channel decoder finds these added extra bits and with the help of these bits detect errors and correct the corrupted signal.

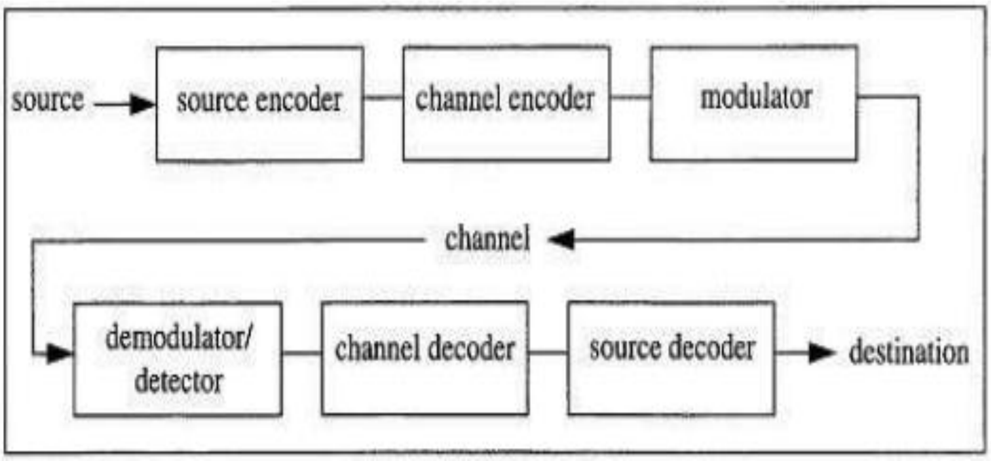

Figure 1: Communication System with channel encoding / decoding operations

Encoder structure

The encoder structure is usually consists of generator polynomial (g). The number of output bits in the encoder depends on the Modulo-2-adders used. But the what will be the result for individuals output bits depend on the selection of the shift register to be modulo-2 operated and that‘s known as the generator polynomial for individual output bit. Now how to select the shift registers, that found by the hit and trial method and side-by side checking the error through simulation on computer. In this paper, the polynomial used is just finding out by hit and trial method and checking the performance characteristics.

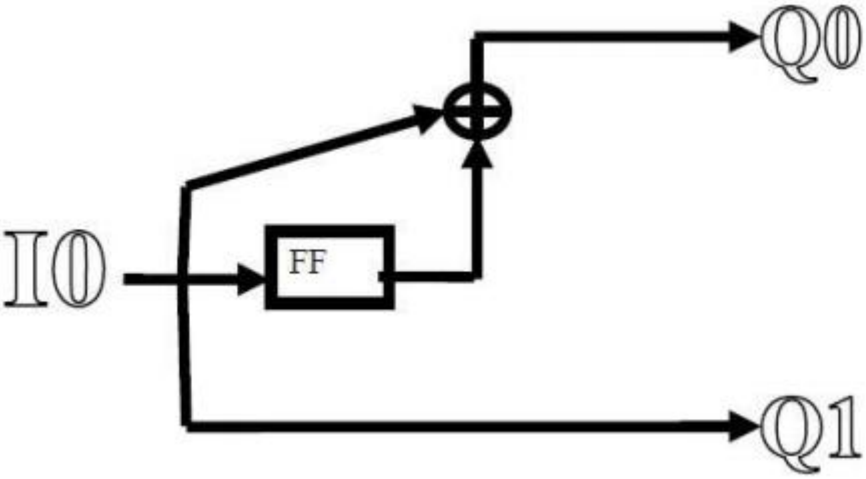

Figure 2: Convolutional coder k = 2, r = ½

SEQUENTIAL LINEAR MMSE ESTIMATION

In much real-time application, observational data is not available in a single batch. Instead the observations are made in a sequence. A naive application of previous formulas would have us discard an old estimate and recomputed a new estimate as fresh data is made available. But then we lose all information provided by the old observation. When the observations are scalar quantities, one possible way of avoiding such re-computation is to first concatenate the entire sequence of observations and then apply the standard estimation formula as done in Example 2. But this can be very tedious because as the number of observation increases so does the size of the matrices that need to be inverted and multiplied grow. Also, this method is difficult to extend to the case of vector observations. Another approach to estimation from sequential observations is to simply update an old estimate as additional data becomes available, leading to finer estimates. Thus a recursive method is desired where the new measurements can modify the old estimates. Implicit in these discussions is the assumption that the statistical properties of $x$does not change with time. In other words, $x$is stationary.

For sequential estimation, if we have an estimate $\hat{x}_1$based on measurements generating space Y_1, then after receiving another set of measurements, we should subtract out from these measurements that part that could be anticipated from the result of the first measurements. In other words, the updating must be based on that part of the new data which is orthogonal to the old data.

Suppose an optimal estimate $\hat{x}_1$has been formed on the basis of past measurements and that error covariance matrix is C_{e_1}. For linear observation processes the best estimate of $y$based on past observation, and hence old estimate $\hat{x}_1$, is $\hat{y} = A\hat{x}_1$. Subtracting $\hat{y}$from y, we obtain $\tilde{y} = y - \hat{y} = A(x - \hat{x}_1) + z = Ae_1 + z$. The new estimate based on additional data is now

$$\hat{x}_2 = \hat{x}_1 + C_{X\tilde{Y}} C_{\tilde{Y}}^{-1} \tilde{y},$$

Where $C_{X\tilde{Y}}$is the cross-covariance between $x$and $\tilde{y}$and $C_{\tilde{Y}}$is the auto-covariance of $\tilde{y}$.

Using the fact that $\mathrm{E}\{\tilde{y}\} = 0$and $x = \hat{x}_1 + e_1$, we can obtain the covariance matrices in terms of error covariance as

$$C_{\tilde{Y}} = AC_{e_1}A^T + C_Z,$$
$$C_{X\tilde{Y}} = \mathrm{E}\{(\hat{x}_1 + e_1 - \bar{x})(Ae_1 + z)^T\} = C_{e_1}A^T.$$

Putting everything together, we have the new estimate as

$$\hat{x}_2 = \hat{x}_1 + C_{e_1}A^T(AC_{e_1}A^T + C_Z)^{-1}(y - A\hat{x}_1),$$

and the new error covariance as

$$C_{e_2} = C_{e_1} - C_{e_1}A^T(AC_{e_1}A^T + C_Z)^{-1}AC_{e_1}.$$

The repeated use of the above two equations as more observations become available lead to recursive estimation techniques. The expressions can be more compactly written as

1. $K_2 = C_{e_1}A^T(AC_{e_1}A^T + C_Z)^{-1},$
2. $\hat{x}_2 = \hat{x}_1 + K_2(y - A\hat{x}_1),$
3. $C_{e_2} = (I - K_2A)C_{e_1}.$

The matrix $K$is often referred to as the gain factor. The repetition of these three steps as more data becomes available leads to an iterative estimation algorithm.

For example, an easy to use recursive expression can be derived when at each m-th time instant the underlying linear observation process yields a scalar such that $y_m = a_m^T x_m + z_m$, where $a_m^T$is 1-by-n known row vector whose values can change with time, $x_m$is n-by-1 random column vector to be estimated, and $z_m$is scalar noise term with variance σ_m^2. After (m+1)-th observation, the direct use of above recursive equations give the expression for the estimate $\hat{x}_{m+1}$as:

$$\hat{x}_{m+1} = \hat{x}_m + k_{m+1}(y_{m+1} - a_{m+1}^T\hat{x}_m)$$

where $y_{m+1}$is the new scalar observation and the gain factor $k_{m+1}$is n-by-1 column vector given by

$$k_{m+1} = \frac{(C_e)_m a_{m+1}}{\sigma_{m+1}^2 + a_{m+1}^T(C_e)_m a_{m+1}}.$$

The $(C_e)_{m+1}$is n-by-n error covariance matrix given by

$$(C_e)_{m+1} = (I - k_{m+1}a_{m+1}^T)(C_e)_m.$$

Here no matrix inversion is required. Also the gain factor k_{m+1}depends on our confidence in the new data sample, as measured by the noise variance, versus that in the previous data. The initial values of $\hat{x}$and $C_e$are taken to be the mean and covariance of the prior probability density function of x.

Maximum Likelihood

In statistics, maximum-likelihood estimation (MLE) is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for the model's parameters.

The method of maximum likelihood corresponds to many well-known estimation methods in statistics. For example, one may be interested in the heights of adult female penguins, but be unable to measure the height of every single penguin in a population due to cost or time constraints. Assuming that the heights are normally (Gaussian) distributed with some unknown mean and variance, the mean and variance can be estimated with MLE while only knowing the heights of some sample of the overall population. MLE would accomplish this by taking the mean and variance as parameters and finding particular parametric values that make the observed results the most probable (given the model). In general, for a fixed set of data and underlying statistical model, the method of maximum likelihood selects the set of values of the model parameters that maximizes the likelihood function. Intuitively, this maximizes the "agreement" of the selected model with the observed data, and for discrete random variables it indeed maximizes the probability of the observed data under the resulting distribution. Maximum-likelihood estimation gives a unified approach to estimation, which is well-defined in the case of the normal distribution and many other problems. However, in some complicated problems, difficulties do occur: in such problems, maximum-likelihood estimators are unsuitable or do not exist.

Suppose there is a sample $x_1, x_2, \ldots, x_n$ of n independent and identically distributed observations, coming from a distribution with an unknown probability density function $f_0(\cdot)$. It is however surmised that the function f_0 belongs to a certain family of distributions $\{ f(\cdot|\theta), \theta \in \Theta \}$ (where θ is a vector of parameters for this family), called the parametric model, so that $f_0 = f(\cdot|\theta_0)$. The value θ_0 is unknown and is referred to as the true value of the parameter vector. It is desirable to find an estimator $\hat{\theta}$which would be as close to the true value θ_0 as possible. Either or both the observed variables x_i and the parameter θ can be vectors.

To use the method of maximum likelihood, one first specifies the joint density function for all observations. For an independent and identically distributed sample, this joint density function is

$$f(x_1, x_2, \ldots, x_n \mid \theta) = f(x_1|\theta) \times f(x_2|\theta) \times \cdots \times f(x_n|\theta).$$

Now we look at this function from a different perspective by considering the observed values $x_1, x_2, \ldots, x_n$ to be fixed "parameters" of this function, whereas θ will be the function's variable and allowed to vary freely; this function will be called the likelihood:

$$\mathcal{L}(\theta\,;x_1, \ldots, x_n) = f(x_1, x_2, \ldots, x_n \mid \theta) = \prod_{i=1}^{n} f(x_i \mid \theta).$$

The method of maximum likelihood estimates θ_0 by finding a value of θ that maximizes $\hat{\ell}(\theta;x)$. This method of estimation defines a **maximum-likelihood estimator** (**MLE**) of θ_0 …

$$\{\hat{\theta}_{\text{mle}}\} \subseteq \{\underset{\theta\in\Theta}{\arg\max}\ \hat{\ell}(\theta\,;x_1, \ldots, x_n)\}.$$

… if any maximum exists. An MLE estimate is the same regardless of whether we maximize the likelihood or the log-likelihood function, since log is a strictly monotonically increasing function.

RESULT & CONCLUSION

This section discusses the performance of the convolutional encoding/Viterbi decoding forward error correction technique. These performance factors help to study how the forward error correction technique can be applied to daily applications.

The following performance factors are considered:

(i) Encoder Memory Size (ii) Signal to Noise Ratio (SNR)

Encoder Memory Size

The forward error correction (FEC) technique makes use of a convolutional encoder. This encoder has memory to store the previous bits/state information. The memory size of an encoder is the number of bits/states that can be stored in the encoder. The FEC technique implemented in this paper uses memory sizes of 2 and 6.

For the convolutional encoder, for larger memory size, the FEC technique has better performance as the coding algorithm becomes more sophisticated. However, the complexity of the decoder is exponentially increased. This is due to the fact that the number of states *n* is exponentially related to the encoder memory size *m*. Thus, the size of the trellis formed is also exponentially related to the encoder memory size. This results in the decoding time to increase significantly with the memory size. Thus, the encoder memory size is kept in smaller values typically between 2 and 6.

Signals to Noise Ratio (SNR)

The most direct factor to affect the performance of the FEC technique is the signal to noise ratio. For a fixed received signal power, noise level can be represented by the signal energy per bit to the noise power spectral density (E_b/N_o). As E_b/N_o increases, the noise level decreases. Thus, the performance of the FEC technique improves or the bit error rate (BER) decreases as E_b/N_o increases.

To analyze how the noise level affects the performance of the FEC technique, a number of simulation tests were conducted using the Matlab codes. The number of input bits considered was 10000. These input bits were encoded using a rate 1/2 and a rate 1/3, ¼,1/5, 1/6, 1/7 and 1/8 convolutional encoder. The encoder was also tested for values of *m* = 2 and *m* = 6 for both the code rates.

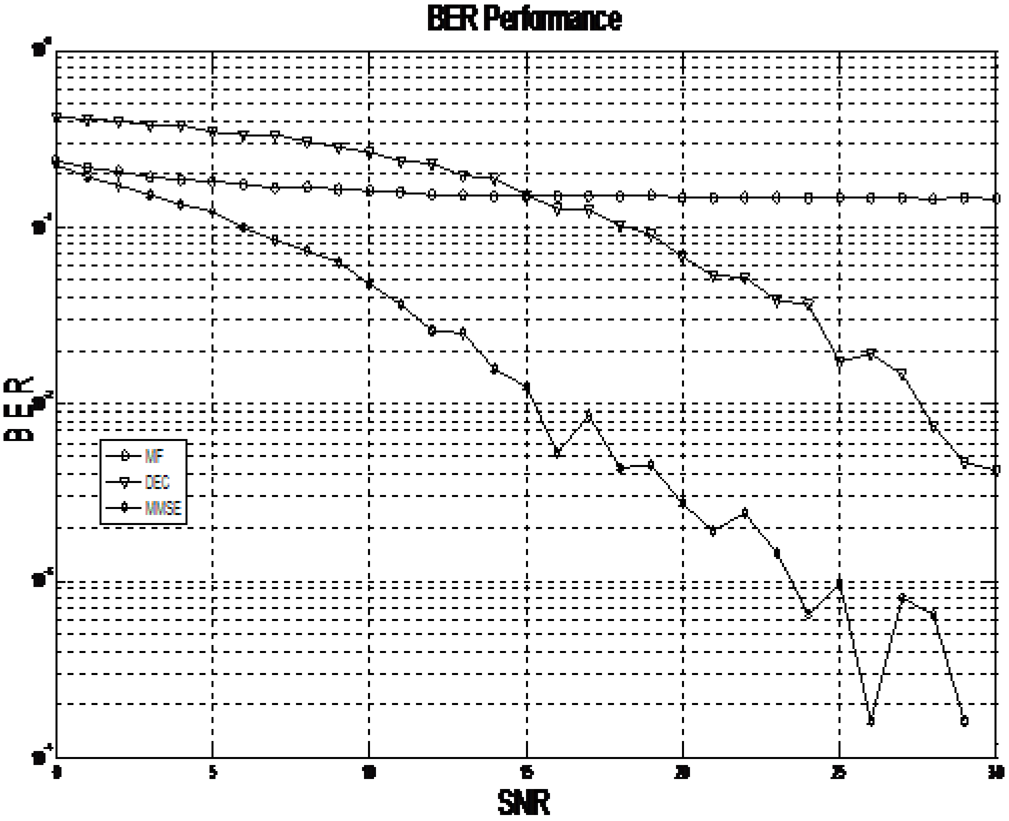

Figure 3: Bit Error Rate vs SNR per bit with Code rate =1/8

CONCLUSION

In this paper, firstly random bit sequence has been generated with the help of LFSR. Then this random bit sequence is encoded with the help of convolutional encoder. The encoded bits are then modulated over three roll-off factors value of 0, 1 and 2 with the help of RRC filter in QPSK modulation technique. These modulated bits are then decoded with the help of Viterbi decoder. The channel through which receiver is receiving the transmitted bits is AWGN channel. This paper also presents the design of AWGN channel. The proposed design has described the SDR implementation of Convolutional Encoder and Viterbi Decoder over AWGN channel. Using QPSK modulation RRC filters got used and depending upon the various roll-off factors BER over every 100 bits got analyzed. The timing summary after analysis got is minimum period is 27.362ns (i.e. maximum frequency is 36.547 MHz). Maximum input arrival time before clock is 4.213ns and maximum output required time after clock is 4.182ns.

REFERENCES

[1] Nejah Nasri, Laurent Andrieux, Abdennaceur Kachouri, Mounir Samet, ―Behavioral Modeling and Simulation of Underwater Channel", WSEAS Transactions on Communications, Issue 2, Volume. 8, Page No. 259-268, Feb 2009.

[2] J.Tulasi, T.Venkata Lakshmi, M.Kamaraju, ―FPGA Implementation of Convolutional Encoder and Hard Decision Viterbi Decoder‖, International Journal of Computer & Communication Technology, Issue 4, Volume 3, Page No. 43-47, Dec 2012.

[3] Yan Sun, Zhizhong Ding, ―FPGA Design and Implementation of a Convolutional Encoder and a Viterbi Decoder Based on 802.11a for OFDM‖, Wireless Engineering and Technology, Volume.3, Page No. 125-131, July 2012.

[4] Jinjin He, Huaping Ling, Zhongfeng Wang, Xinming Huang and Kai Zhang, ―High-Speed Low-Power Viterbi Decoder Design for TCM Decoders‖, IEEE Trans. VLSI, Volume. 20, Page No. 755-759, April 2012.

[5] Shraddha Shukla1, Nagendra Sah 2 "An Experimental Implementation of Convolution Encoder and Viterbi Decoder by FPGA Emulation" International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Issue 5, May 2014.

[6] Mr. Sandesh Y.M*, Mr. Kasetty Rambabu** " Implementation of Convolution Encoder and Viterbi Decoder for Constraint Length 7 and Bit Rate ½" Int. Journal of Engineering Research and Applications www.ijera.com ISSN : 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp.42-46.

[7] Yonghui Li, Senior Member, IEEE, Md. Shahriar Rahman, Branka Vucetic, Fellow, IEEE "Duality of Channel Encoding and Decoding - Part I: Rate-1 Convolution Codes", IEEE Transactions on Information Theory 33 pages, 19 figures, Submitted on 12 Jan 2012.

[8] A. Dubey, Sophomore Undergraduate, Electrical Engineering, IIT Delhi "Convolutional Encoding and Decoding with the Viterbi Algorithm" 2011EE20061, Digital Electronic Circuits, Fall-Winter 2012, IIT Delhi.

[9] K. S. Arunlal1 and Dr. S. A. Hariprasad2 "AN EFFICIENT VITERBI DECODER" International Journal of Advanced Information Technology (IJAIT) Vol. 2, No.1, February 2012.

[10] L. Dong, "Cross-layer design for cooperative wireless networks," Drexel Paper and Dissertations, 4 Dec. 2008.

[11] P. Subhashini, D. R. Mahesh Varma and Y. David Solomon Raju, ―Implementation Analysis of adaptive Viterbi Decoder for High Speed Applications‖, International Journal of Computer Applications, Volume 31, Issue.2, Page No. 0975—8887, October 2011.

[12] S.V.Viraktamath and G.V. Attimarad, ―Impact of constraint length on performance of Convolutional Codec in AWGN channel for image application", International Journal of Engineering Science and Technology, Volume 2, Issue 9, Page No. 4696-4700, Dec 2010.

[13] Ramtin, I. OudinoL S . Seatti. A. Le-Clery, and I. Lebm., ―Full transceiver Circuit simulation using VHDL-AMS‖ International Conference of Microwave Engineering, Page No. 29-33. May 2002.

[14] A. Fakhfakh, N. Milet - his , I. Tomas, and H. Levi, ―Behavioral modeling of phase noise and jitter in voltage-controlled oscillators with VHDL-AMS.‖ Proceedings of IEEE International Conference on Circuits and Systems for Communication, Page No. 370-373, 2002.

[15] U. Knoehel, Hang . and R. Kakerow, ―Verification of the RF subsystem within wireless LAN system level simulation‖ Design, Automation and Design in Euorpe Conference. Page No. 286-291, 2003.