ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

Design of Chaos Based Turbo Systems in Fading Channel

Ramya B M[1], Dr. Sudha K L[2]

M.Tech Student, Dept. of ECE, DSCE, Bangalore, India[1]

Professor, Dept. of ECE, DSCE, Bangalore, India[2]

Abstract: The increasing growth for widespread wireless communication system demands stronger coding scheme to reduce the signal deterioration during its transmission, generally without facilitating to the expensive hardware materials. Turbo code is a forward error correcting code. It mainly consists of RSC encoder, an interleaver in the encoder part and SISO module used in decoder part. The code performance is mainly based on design of an interleaver. The encoder uses multi tent map and inverse multi tent map structure for generating chaotic like samples. The chaotic sequence corresponds to the dynamics of tent map and they are used in engineering applications. For chaos based Turbo decoding, maximum a posteriori algorithm also known as BCJR algorithm, which is based on trellis coded estimation algorithm. It uses SISO module, is a soft decision decoder it performs the decoding operation in an iterative scheme.

Keywords: Turbo code, Interleaver, Chaos, Tent map, Soft decision.

I. Introduction

When an information signal is routed through the communication channel, error or noise gets added to it. In order to find the errors in the received information signal, Error Detection Coding is used and Error Correction Coding is used to rectify these errors in the received signal. Error Control Coding involves both Error Detection and Correction Coding. There are several types of coding schemes available in error control coding such as Linear Block Codes, Convolution Codes and Turbo Codes [1] etc. Linear Block Codes are effective in case of one to one mapping among the information signal and its code word. Convolution Codes have additional configuration such as block codes in generator matrix. Hence the encoding function is similar to the convolution operation. Convolutional codes serve a greater performance than block codes and also as the block length increases, decoding complexity is also increases.

A new kind of convolutional codes, Turbo Codes was introduced in 1993, as block length increases it gives inexpensive decoding complexity. These are also known as Parallel Concatenated convolutional codes (PCCC) and decoding complexity is small compared to traditional convolutional codes. Since decoding algorithm complexity becomes small, hence Shannon's Limit boundaries are also reachable for all practical purposes. With higher block length or constraint length theoretically it is feasible to achieve Shannon limit using block code and convolutional code. To decode such long codes, the required processing power would lead to an impractical. Turbo codes are used to come out of this disadvantage by using RSC coders and iterative soft decoders.

Chaotic signal is defined as outputs of nonlinear dynamical systems. It can be used in different engineering and communication field. According to signal analysis these signals provide greater useful models in natural aspects. As the consequence from signal synthesis, important features of these signals are greatly used in various broadband communication and radar applications. Therefore, to utilize chaotic signals in these applications, it requires strong and suitable algorithms for prediction and calculation of these signals in the proximity of noise or distortion.

The chaotic signals [5] are used to transfer the signal information. It was first invented in early 1990s. In digital communication system chaotic communications has a significant effect. In chaotic signal the height of responding to the changes in initial conditions is more, for example, theoretically in trellis based codes, it makes the separation of merging paths, and hence these systems also taken as better channel encoding components. In chaos based encoders, mainly two forms are available they are based on transmitted code word size. The first type of chaos based channel encoder contains non-linear functions it transmits digital signals and takes advantage of coding gain from the similarities between consecutive bits in the transmission. Because of the less spectral efficiency, a non-null free distance and better performance is easily obtained by optimizing this kind of codes. The second kind of CCM channel encoders contains an encoder which transmits a complex quasi-continuous code, which means that their characteristics are naturally chaotic.

The non-periodic, uncorrelation and unpredictability properties of chaotic signals are to be suitable for secure communications or multiple access systems, thus it promotes a great deal of researches and developments. Moreover, this benefit has been reduced since some systems were not as competitive as required. As a contrast, during the past decade the publication of some proposals competitive to the state-of-the-art standard systems. In multipath fading channels, the great use of Chaos coded system is obtained at waveform level and in coding level point it performs better in multiuser channels. Also chaos

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

coded modulator[2] (CCM) based systems operates well at both waveform and coding level and it could be robust under additive white Gaussian noise (AWGN) channel [3]. These CCM systems have greater uses in some classes of dispersive channels such as under flat fading or inter symbol interference.

II. LITERATURE SURVEY

The text book "Error Control Coding" by Shu Lin and Daniel J.Costello, [1] is referred for the basics of Turbo codes. It explains the features of turbo codes, is that they consist of two or more convolutional codes, arranged in concatenation manner along with the interleaver.

The research paper on "Chaos-Based Turbo Systems in Fading Channels" by F. J. Escribano et.al, [2] has explained about how to use chaos structure in turbo codes. Parallel concatenation chaos coded modulator (PCCCM) has been used in the encoder portion. Each CCM block consists of Q memory positions (binary), weighted sum of the register contents, and a feedback loop.

The research paper on "Turbo-like structures for chaos coding and decoding" by Francisco J. Escribano et.al, [3] he has discussed about how to create turbo structure for binary input and chaotic outputs with effective coding and decoding in AWGN. Here the concatenated encoder contains two chaotic encoders of code rate 1 and they fed with binary input and interleaver output.

The technical paper on "Improving the performance of chaos-based modulations via serial concatenation" by F. J. Escribano et.al, [4] explained the use of an inner chaos-based coded modulator and an outer convolutional channel encoder for serially concatenated system and how iterative decoding scheme used in concatenated structure with known extrinsic information and how the BER bounded with the channel encoder transfer function.

The research paper on "Chaos coded modulations over Rayleigh and Rician flat fading channels" by F. J. Escribano et.al, [5] explains chaos-based modulations in both Rayleigh and Rician frequency non-selective fading channel with additive white Gaussian noise (AWGN). For all kind of CCMs, it provides boundary to the bit error rate in these channels. The CCM system working at waveform level as well as in coding level has been potentially used in multipath fading channels.

III. TURBO ENCODER

Turbo code is the parallel concatenation of a number of Recursive Systematic Convolutional (RSC) code. Normally the number of RSC's will be two. The input to first encoder is systematic input whereas for the second encoder is an interleaved version of the systematic input, so that the outputs of encoder 1 and encoder 2 are time displaced codes generated from the same input sequence.

The input sequence is only presented once at the output. The outputs of the two coders are multiplexed into the stream giving a rate R=1/3 code.

A. Parallel Concatenated Encoder

The concatenated encoder structure is as shown in the fig 1. It consists of two Chaotic Convolutional Module [4] which accepts successive blocks of N bits, b={b1,……. bn} as input for CCM 1 and C={c1…….cn} for CCM 2, where C is the interleaved version of the input b.

The first CCM block produces a chaos encoded samples represented as Y2n-1, n=1………N at a rate of 1 sample and the second CCM block produces a second block of chaos encoded samples, Y2n , n=1………N per input bit, at the same rate. Thus the PCCCM output blocks of size 2N (total rate R=1/3 bits per chaotic symbol).

The interleaver which is considered here is the well-known S-random interleaver.

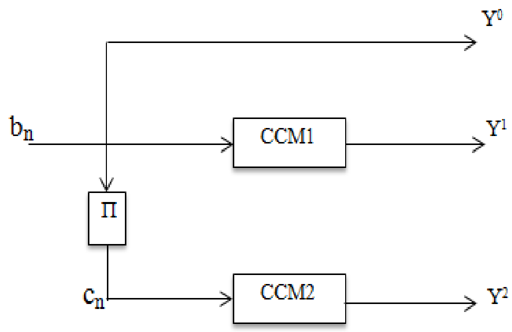

Fig 1: Chaos based Turbo encoder

CCM block outputs a chaos-like samples Zn, as shown in Fig 2 and 3, obtained by an encoding process which can be described by the finite-state/equivalent trellis encoder with z_Q states, so that z_n samples are quantized over z_Q values (i.e.,$z_n \in s_Q=\{z_0 \ldots\ldots\ldots z_{2Q-1}\}$) uniformly distributed over the interval [0,1].

Since there is a one-to-one correspondence between the output value and the finite state machine [5] at time, the properties of the chaos based sequence are kept regardless of the initial state of the finite-state machine process and the corresponding initial z_0 value.

This initial state is considered as the all-zero value, so that it can be used at the decoding stage for proper initialization. Interleaver coding gain depends on Q binary memory positions (ri) in each CCM block, a weighted sum of the memory position contents, and a feedback loop.

The chaos encoded sample Xn, is obtained by mapping the value from [0, 1] interval to a suitable set of values, by using amplitude/phase mapping.

We have chosen these two possibilities:

•Amplitude mapping:

Here the interval [0,1] is symmetrically mapped to the [-1,1] interval

$$Yn=2Z_n - 1 \qquad \ldots\ldots\ldots(1)$$

•Phase mapping:

Here the values in the interval [0, 1]. These values drive the phase in the interval [0, 2Π] of a set of baseband-equivalent complex-valued samples with normalized amplitude

$$Yn = e^{2j \Pi Zn} \qquad \ldots\ldots\ldots(2)$$

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

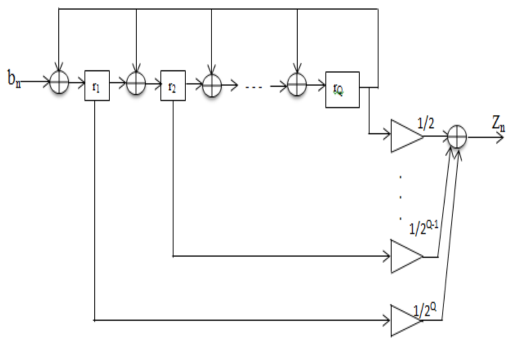

Fig 2: Inverse multi Tent map Finite state machine equivalent

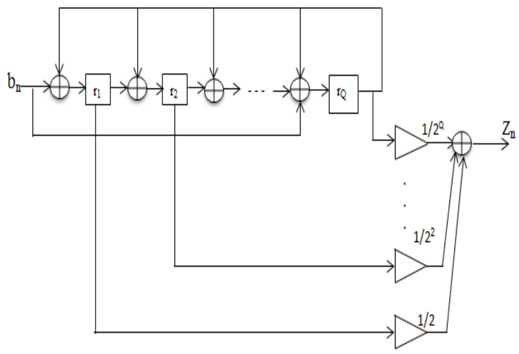

Fig 3: Multi Tent map Finite state machine equivalent

IV. TURBO DECODER

Turbo decoding algorithm is a high dimensional system characterized by more number of parameters. The algorithm is considered as a dynamical system characterized by a single parameter that nearly closer to the signal to noise ratio (SNR). The purpose of the turbo decoder or receiver is to recover the original data sequence from received systematic sequence and two parity sequences even though they are corrupted by noise.

For Turbo decoding two well-known algorithms are used they are: Soft output Viterbi algorithm (SOVA) and BCJR algorithm. SOVA was first proposed by Hagenauer and Hoher using Viterbi algorithm invented by Andrew J. Viterbi and BCJR algorithm was developed by Bahl, Cocke, Jelinek and Raviv.

The performance of BCJR algorithm is better than Viterbi algorithm. BCJR algorithm is also known as MAP algorithm [7] and it uses backward recursion method hence it is slightly complex than Viterbi algorithm. In received signal MAP algorithm looks for frequently occurring symbol and SOVA looks for frequently occurring sequence.

A. Decoding

The signal is transmitted through AWGN channel with SNR = 0.25. In MAP decoding algorithm, Es/N0 directly affects the system performance and it performs this by exaggerating the systematic bits. Hence, if the systematic bit contain an error that is also exaggerated. Here, LC denotes the SNR of the received signal.

$$L_C = 4E_b/N_0 = 4 \times 0.25 = 1.0 \qquad \ldots\ldots\ldots\ldots(3)$$

Multiplying Lc with received signal is described as in Eq (3) and its value will be 1.0. Decoding is done in each time instant k, by calculating log-likelihood values of transmitted bit. If the value is positive, then the decision is also positive. Computation of L(uk) or log-likelihood values is a little bit complex process. In order to calculate the L-values the main equation is given by

$$L(U_k) = [L^e(U_k) + L_c Y_k^{1,s}] + \log\left(\frac{\sum_{u+} \alpha_{k-1}(s')\,\beta_k(s)\,\gamma_e(s',s)}{\sum_{u-} \alpha_{k-1}(s')\,\beta_k(s)\,\gamma_e(s',s)}\right) \qquad \ldots\ldots\ldots(4)$$

The foremost term in the above Eq (4) is the intrinsic or a priori probability estimation obtained from Decoder 2 and it indicates the LLR estimation of the corresponding bit. Initially, the decoder doesn't know about this value, hence assumed it as 0. The second term, LC Yk1,s is obtained by multiplying the systematic bit with Lc. This denotes the L-values and SNR of the channel. The third big term is the a-posteriori probability. For each trellis branches, this value is computed. In trellis diagram, for each segment have only one value either +1 or -1. Finally, based on L(uk) computation decision will be made.

The entire equation can be simply written as addition of the three terms.

L-apriori: In the first iteration, it is an initially assumed value.

L-Channel: It is related to systematic bit Yk1,s and channel SNR LC.

Le – This is a-posteriori L-value computed in each iteration.

Now the Eq (4) can be rewritten as

$$L\_apriori + L\_Channel + Le(uk) \qquad \ldots\ldots\ldots(5)$$

In each iteration, an L-channel value does not changes since it depends on channel SNR and systematic bit. Neither Lc nor systematic bit changes, hence it is called K. A-priori and a-posteriori L-values are the only two terms changes here.

$$L(UK) = L\_apriori + K + L\_posteriori \qquad \ldots\ldots(6)$$

A priori estimation value of the first decoder is given next decoder to calculate the extrinsic probability value and this value is again passed to the first decoder or it is used to calculate L(uk). In each time instant K, only a posteriori probability value is calculated and decoder repeat the same operation for a fixed number of time. The log-likelihood posteriori value is also called extrinsic information. The iteration cycle for MAP decoding is as shown in fig 4.

The main intension is to calculate the extrinsic L-values, when this value will be high, then L(uk) is computed and make a decision. This L-value computation uses a Forward-Backward Recursion algorithm.

In the Eq (3), there are three terms in the ratio. They are

$$\alpha_k(s').\beta_k(s).\gamma^e_k(s',s) \qquad \ldots\ldots\ldots(7)$$

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

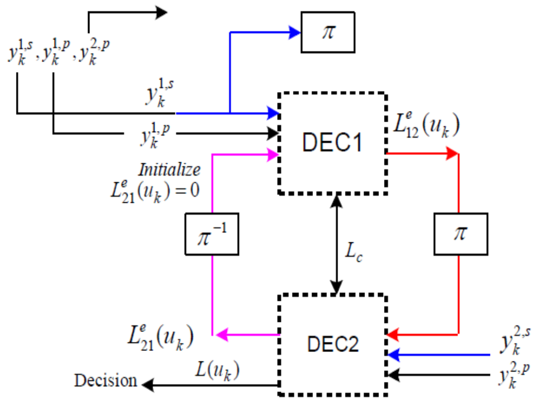

Fig 4: Iterative cycle of MAP decoding

The term, αk(s') is the forward metric. This value is calculated from forward recursion using branch metric denoted by βk(s). It is given by

$$\alpha_k = \sum_{s} \alpha_{k-1}(s').\gamma_k(s',s) \qquad ....(8)$$

The forward branch metric is computed by multiplying the previous forward metric with full branch metric. This is the recursive process. First full branch metric is computed, from which forward metric will be calculated. In trellis tree diagram, each branch metric is related to the trellis values and received signal. The correlation will be high, if the received value sign is as same as the coded bits. In each decoder, full branch metric uses both systematic and parity bits and partial branch metric uses only parity bits.

V. RESULTS AND DISCUSSION

A. MATLAB Results

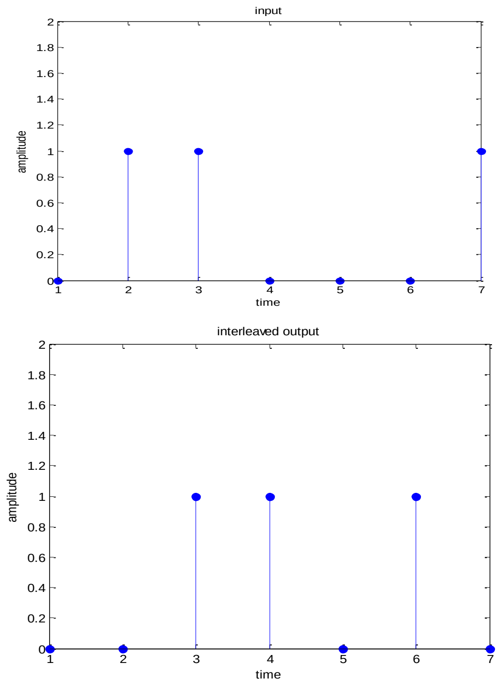

Fig 5: Input and interleaver output

The above figure shows that the systematic input (0,1,1,0,0,0,1) for CCM1 and for CCM2 the input will be the permuted/reordered version of the input signal i.e. (0,0,1,1,0,1,0). Here the order of the bits is changed and this can be done by the interleaver, which has been described in the previous section. Interleaver output acts as input for the second CCM. Each CCM block produces chaos like samples Zn. These samples are normalized by amplitude or phase mapping.

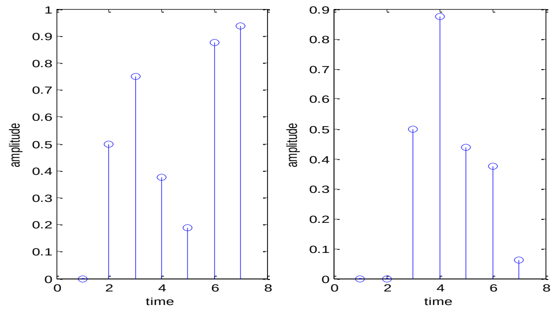

Fig 6: Output of CCM

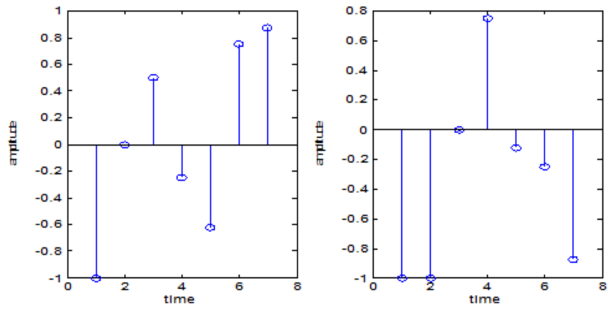

Fig 7: Output of Amplitude mapping

The finite-state-machine of a CCM produces quantized samples Zn, of a chaotic sequence as shown in fig 5.2, which is in the interval [0, 1]. The PCCCM system simulated has Q=4. The Zn samples are obtained by summing the weighted portion of each shift register contents. In the amplitude/phase mapping scheme, the samples are rescaled within the interval [-1,1]. The amplitude and phase mapping is performed by Eq (1) and Eq (2) respectively.

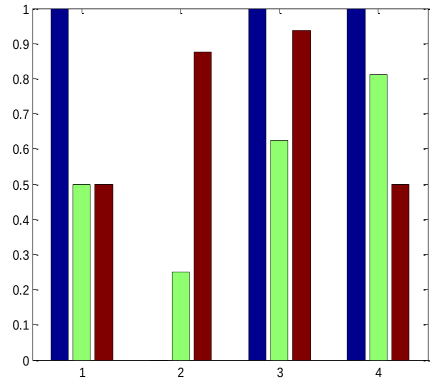

Fig 8: Multiplexing of input and parity bits

Input bit Parity bit1 Parity bit2

The above figure shows the multiplexing of the systematic bit and parity bit output from the CCM1 and CCM2 before transmitting through the communication channel.

 DOI 10.17148/IJARCCE.2015.47126

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

Extrinsic value: This is the output of decoder in every iteration. This value is computed by multiplication of the forward state metric, backward state metric and partial branch metric for each branch. After interleaving, these values are passed to Decoder 2 then it will become a-priori probabilities.

Table I: Extrinsic values of Decoder1

DEC1 Extrinsic values	K=1	K=2	K=3	K=4	K=5	K=6	K=7
Iteration 1	0.7879	0.3294	0.2498	0.19324	-1.5699	1.7461	2.4611
Iteration 2	0.9425	0.3812	-1.0209	1.8813	-1.9407	1.2232	0.4411
Iteration 3	1.5490	0.9743	-0.3125	-0.7248	-2.3669	-0.4298	1.7586
Iteration 4	1.8279	1.9005	1.8823	1.9281	2.7289	-1.1848	1.2757
Iteration 5	2.7451	2.9857	3.1836	3.2926	4.1313	-2.1854	4.9253

Table II: L(Uk) values of Decoder1

DEC1 L(Uk)	K=1	K=2	K=3	K=4	K=5	K=6	K=7
Iteration 1	0.1005	0.2906	0.2817	0.0757	-0.2139	-0.3306	0.5622
Iteration 2	0.3660	1.0587	1.0261	0.2759	-0.7790	-1.2044	2.0477
Iteration 3	0.4089	1.1826	1.1463	-0.5212	-0.8702	-1.3454	2.2875
Iteration 4	1.6649	4.8153	4.6672	-1.3659	-3.5432	-5.4781	9.3138
Iteration 5	-4.8400	18.8362	18.2569	-5.3430	-13.8601	-21.4288	36.4332

Table III: Extrinsic values of Decoder2

DEC2 Extrinsic values	K=1	K=2	K=3	K=4	K=5	K=6	K=7
Iteration 1	0.2844	0.4944	1.0148	0.0364	-0.1947	-0.1840	1.9055
Iteration 2	-1.8150	1.5795	1.1718	0.2544	-0.2590	-0.2484	1.9356
Iteration 3	2.1982	-1.2955	2.0784	0.9734	1.9677	-2.3635	4.9936
Iteration 4	-2.2256	-2.3214	3.0950	3.9646	2.9856	-4.3152	5.9983
Iteration 5	4.2281	8.3332	12.1370	5.9693	7.9966	-10.3068	21.0175

Table IV: L(Uk) values of Decoder2

DEC2 L(Uk)	K=1	K=2	K=3	K=4	K=5	K=6	K=7
Iteration 1	0.4369	1.2637	1.2248	0.3293	-0.9298	-1.4376	2.4442
Iteration 2	-0.5088	2.3892	2.3157	0.6227	-1.7580	-2.7180	4.6212
Iteration 3	-0.7488	4.7019	4.5573	1.2254	-3.4598	-5.3491	9.0945
Iteration 4	-1.3888	5.6299	5.4567	-1.3455	-4.1426	-6.4048	10.8893
Iteration 5	-8.9152	36.1388	35.0275	-8.6367	-26.5917	-41.1130	69.9001

Table V: Final decision output from decoder2

DEC2 decision	K=1	K=2	K=3	K=4	K=5	K=6	K=7
Iteration 1	1	1	1	1	0	0	1
Iteration 2	0	1	1	1	0	0	1
Iteration 3	0	1	1	1	0	0	1
Iteration 4	0	1	1	0	0	0	1
Iteration 5	0	1	1	0	0	0	1

The above table tabulates the extrinsic and L(Uk) values computed from the decoder1 and decoder 2 for each iteration. Based on the L(Uk) values of dec2 final decision will be taken. If the value is positive then decision will be 1, if it is negative then decision will be 0.

The figure shows the bit error rate plot for different number of iterations. Bit error rate is the ratio of number of bit errors divided by total number of transmitted bits. The BER is considered as an approximate estimate of the bit error probability. This estimate is accurate for a long time interval and a high number of bit errors. From the above figure as the number of iteration increases probability bit error will be reduced.

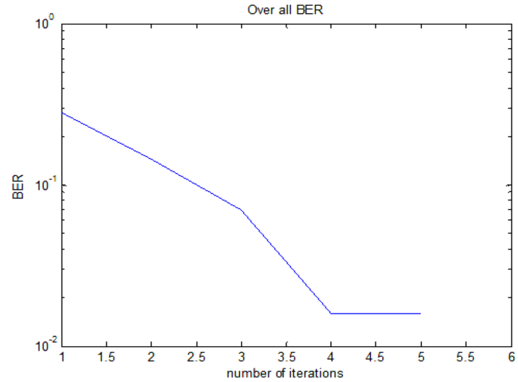

Fig 9: Plot of BER v/s number of iterations

B. VHDL Results

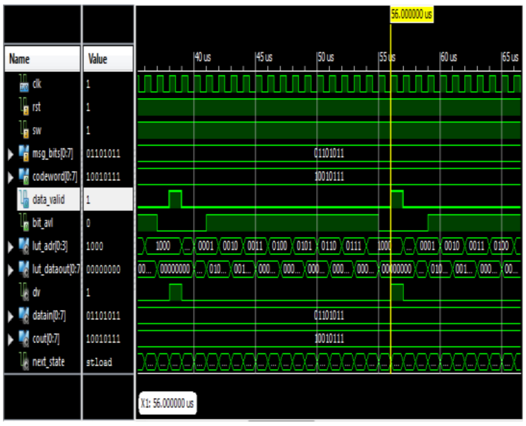

Fig 10: Output of chaos Turbo encoder

VHDL simulation result for chaos based Turbo encoder is as shown in fig 10. Input signal to the encoder is message bits of 0 to 7. Code word is the output of the encoder block. When reset signal is zero the output signal is undefined. When reset signal goes high, for every raising edge of the clock it performs encoding operation and code word appears at the output. Lut_adr and lut_dataout signal indicates the address of each bit and its corresponding data stored in the memory. Bit_avl signal indicates the availability of the signal at the output. Next_state shows the state of the signal in each clock cycle.

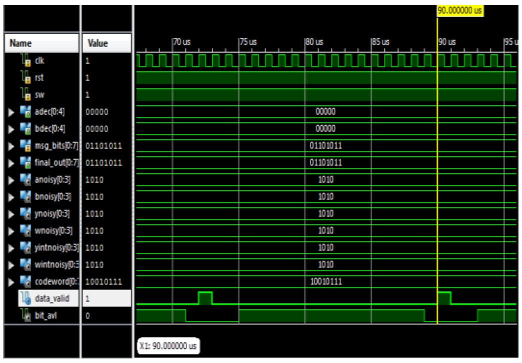

Fig 11: Output of chaos Turbo decoder

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

The above figure shows the simulation result of turbo decoder. The input signals are msg_bits of 8 bit, clock, reset and switch. For each positive edge of the clock with reset and switch signal high it performs decoding operation. The signals anoisy, bnoisy, ynoisy, wnoisy are the received noisy input signals for SISO decoder. The code word is the output of the turbo encoder and which is received at the decoder. After performing successive number of iterations final decoder output will be same as input message signal.

VI. CONCLUSION

In this paper parallel concatenated chaos-based systems are designed by the addition of amplitude mapping to the chaotic-like output samples. The performance of parallel concatenated chaos-based turbo coder is evaluated. Matlab and VHDL simulation of turbo encoder has been done. Normalized chaos values are obtained by amplitude mapping. The simulation results shows the possibilities and trade-offs in the PCCCM systems, and they can be used in chaos-based communications and also used in next generation advanced communication systems. Using log-MAP algorithm Turbo decoder has been designed using SISO module. Extrinsic L-values are computed in each iteration and final decision will be taken based on the decoder 2 output values. Bit error rate analysis is done by taking the number of errors in the transmitted bits. As the iteration increases error rate will be decreases.

REFERENCES

[1] Shu Lin and Daniel J.Costello, “Error Control Coding,” published by Pearson Education, Second edition, 2011
[2] Francisco J. Escribano Alexandre Wagemakers, and Miguel A. F. Sanjuán, “Chaos-Based Turbo Systems in Fading Channels,” IEEE transactions on circuits and systems, vol. 61, no. 2, Feb 2014
[3] F. J. Escribano, S. Kozic, L. López, M. A. F. Sanjuán, and M. Hasler, “Turbo-like structures for chaos coding and decoding,” IEEE Trans. Commun., vol. 57, no. 3, pp. 597–601, Mar. 2009
[4] F. J. Escribano, L. López, and M. A. F. Sanjuán, “Improving the performance of chaos-based modulations via serial concatenation,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 2, pp. 448–459, Feb. 2009
[5] F. J. Escribano, L. López, and M. A. F. Sanjuán, “Chaos coded modulations over rayleigh and rician flat fading channels,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 6, pp. 581–585, Jun. 2008
[6] S. ten Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” IEEE Trans. Commun., vol. 49, no. 10, pp. 1727–1737, Oct. 2001
[7] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “A soft-input soft-output APP module for iterative decoding of concatenated codes,” IEEE Commun. Lett, vol. 1, no. 1, pp. 22–24, Jan. 1997