International Journal of Electronics Engineering (ISSN: 0973-7383
Volume 11 • Issue 1 pp. 314-321 Jan 2019-June 2019 www.csjournals.com

Simulation of Adaptive Third Component Turbo Codes using Genetic Algorithm

Dr. K. Suman
Dept. of ECE, CBIT, Hyderabad, INDIA
suman_me192@cbit.ac.in

Abstract: Detection and correction of errors has become an essential part of any communication system. Over the years, error correction codes have been proposed, that work towards reaching the Shannon Limit. Among all the ones proposed, Turbo codes are the only codes that approach the Shannon Limit. The bit error rate is taken as the criteria and always an attempt was made to minimize it. The bit error rate while using 3D Turbo Codes[2] was improved compared to conventional turbo codes. To achieve better results, Adaptive 3D-TC have been proposed [1] and compared to 3D TC. It is observed that for majority of cases, BER was less. In this paper, the signal to noise ratio is further increased and the corresponding BER values are calculated.

Key words: Shannon limit, Adaptive 3D-TC, Turbo codes, 3D-TC, Bit error rates.

Introduction

Wireless Mobile communication occupies very important role in telecommunication industry and is fastest growing field. It provides access to the global network at any time irrespective of the location or mobility of the user. Due to the progress of Internet and Cellular communication, mobile communication has become indispensable. The future mobile communication requires transmission of data to be done at higher bit rates which are used for many services like image and signal processing and also many network related applications. Ideally, the information is to be transferred to the destination from the source without compromising on quality of the message which should be preserved. In this regard, communication system plays a vital role. Digital information is altered by modulator into analog waveforms. They are transmitted through a noisy medium. Subsequently, they are changed back into symbols in a sequence at receiver by the demodulator. The information is transferred with the aim of achieving reliable communication at transmission rates approaching channel capacity given by the Shannon limit. The information sequence that is to be transmitted consists of several parts that have different degrees of significance. Hence, there is a need for different levels of protection against noise. As the information is transmitted over the communication channel, presence of noise in the channel result in errors in the sequence. The information has to be protected from occurrence of errors. Protection is done by a method called coding or channel coding, where information is disguised using different codes.

Codes are the set of symbols to which meanings or values are attached and are designed to provide different levels of data protection. Hence, the role of error correction codes becomes more prominent. Introduction of error control codes improve the efficiency and accuracy of the transmitted information. Therefore, the coding used for controlling errors has become crucial part in the design of modern communication and digital storage systems. The elemental concept of error control coding is the addition of redundancy at the transmitter, which converts the transmitted bits to a longer sequence of bits (codeword) to combat errors introduced by the noise in the channels. The exploitation of this redundancy is done at the receiver to detect and / or correct errors.

A modified turbo code called Three-Dimensional Turbo Codes (3D-TC) has been chosen in the research for further improvement. Third component is introduced in traditional turbo codes which improved the code performance. In 3D-TC, the parameters such as permeability and permutation rates are constant. Under different noisy environments the permeability and permutation rates remain static and affect the performance. To address the problem, 'Adaptive third component Turbo Code (A3D-TC)' [1] is proposed in which the parameters are made to vary with different noisy environments.

The Adaptive Third Dimensional Turbo Codes

In A3D-TC, as shown in Fig: 1. the third component parameters are made adaptive. This is accomplished by generating a Genetic Algorithm (GA) based knowledge source and feeding it to feed forward neural network. The network outputs third component parameters according to the noise and signal strengths so that bit error rate at decoding section can be minimized in an effective way. The permeability and permutation rates are found with respect to the different noise strength. The data so obtained is utilized to train Artificial Intelligence (AI).

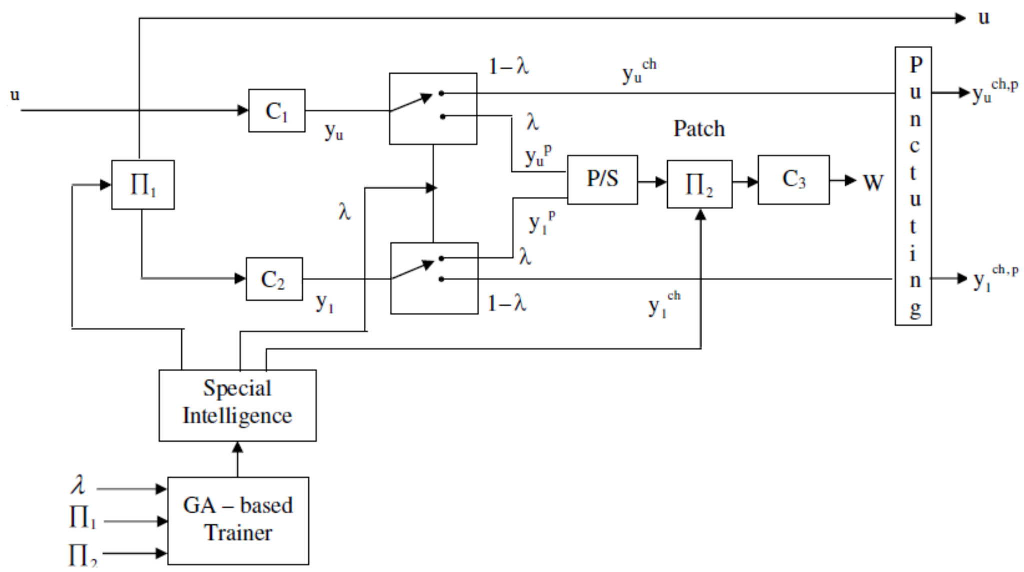


Figure:1 A3D-TC Encoder

The AI technique uses a classifier which will feed forward neural network. The classifier is trained in such a way that the permeability rate and permutation possibilities are decided as per the strength of the noise. As per the decision level, different states of connections are established with post encoder and hence the third component is determined. Such third component will be dynamically varying when the strength of the channel noise varies.

By introducing Special Intelligence (SI), A3D-TC improves the error correction capability and decides the permeability and permutation rates of the third component encoder. SI is tuned by generating Genetic Algorithm (GA) based knowledge source and knowledge feeding. Once tuning is completed, the encoder generates third component parameters dynamically according to the noise variance.

However, the addition of special intelligence in the third component of the encoder never disturbs the conventional third component decoder. The decoder is as shown in Fig:2. The feed forward neural network is used as the special intelligence.

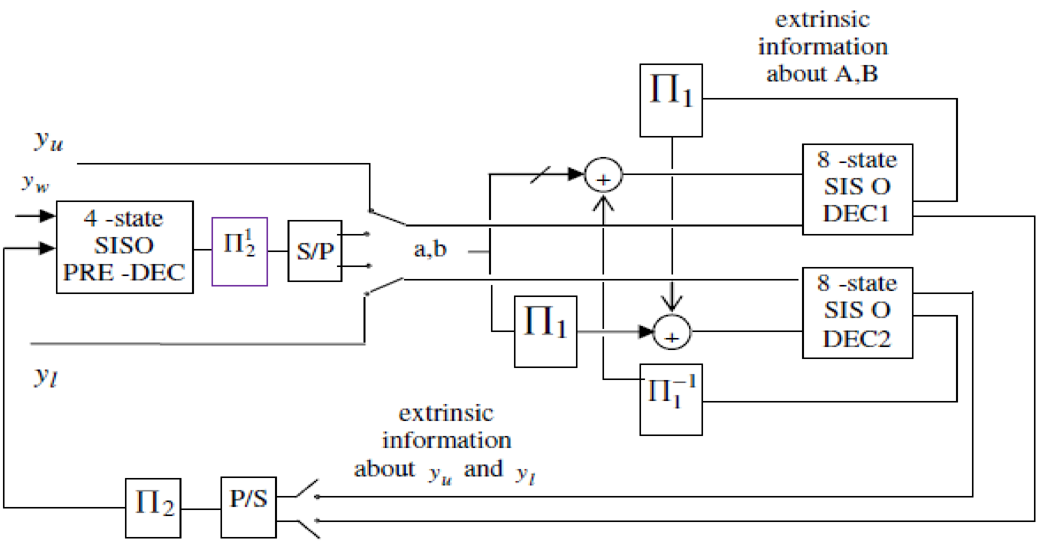


Figure: 2. A3D-TC Decoder

International Journal of Electronics Engineering (ISSN: 0973-7383
Volume 11 • Issue 1 pp. 314-321 Jan 2019-June 2019 www.csjournals.com

GA-Based Knowledge Source

The GA-based knowledge source is mainly used in the coding system to aid in training the special intelligence. This can be accomplished by generating a precise training dataset, in which the noise variance Nσ is considered as input and the suitable A3D-TC parameters such as permeability rate lambda λ, permutation rate Π1 and permutation rate Π2 as output. In order to obtain the suitable A3D-TC parameters provided, the classical GA procedure is described below.

Generate a population pool of N_c chromosomes, in which each chromosome can be represented as

$$X_i = [x_0 \;\; x_1 \;\; x_2]_i ;\; 0 \le i \le N_c - 1$$

where, x_0 , x_1 and x_2 are the genes of the chromosomes that are generated arbitrary within the corresponding limits i.e. $x_0 \in [\lambda_{\min}, \lambda_{\max}]$, $x_1 \in [\prod_1^{\min}, \prod_1^{\max}]$ and $x_2 \in [\prod_2^{\min}, \prod_2^{\max}]$ such that $\lambda_{\min}$, $\prod_1^{\min}$ and $\prod_2^{\min}$ are the minimum limits of λ, $\prod_1$ and $\prod_2$, whereas $\lambda_{\max}$, $\prod_1^{\max}$ and $\prod_2^{\max}$ are the maximum limits of λ, $\prod_1$ and $\prod_2$, respectively. In the most probable cases, $\prod_1^{\min} = \prod_2^{\min}$ and $\prod_1^{\max} = \prod_2^{\max}$.

To determine the fitness of every chromosome, the following steps are followed

(i) Design A3D-TC as per the chromosome parameters

(ii) Encode with random input bits

(iii) Add AWGN noise variance N_σ in the channel

(iv) Decode the data

(v) Determine the mean bit error rate (BER)

$$\xi_{BER} = \frac{1}{|N_\sigma|} \sum_{l=0}^{|N_\sigma|-1} BER_l^2$$

Select *Nc/2* chromosomes, which have minimum error, among all the chromosomes those are present in the population pool.

Perform crossover and mutation at a rate of *Cr* and *Mr* respectively so that *3Cr* genes are exchanged between two chromosomes in crossover operation to obtain child chromosomes and *3Mr* genes are replaced by new genes in every child chromosomes. In our work single point crossover operation and random mutation is used. After crossover and mutation, new chromosomes are obtained which combine with the parent *Nc/2* chromosomes (selected based on fitness) and form a new population pool. The new population is submitted for fitness evaluation and the process gets repeated till a maximum number of iterations get reached.

At the end of the iterations, a best set of A3D-TC parameters *Xbest* are obtained which are used for knowledge feeding the corresponding *Nσ*

As shown in the Fig:1. Special Intelligence (SI) is added to the third component of the encoder. However SI thus added never disturbs the conventional third component decoder [1].

The SI used here has a single input node, three output nodes and N_H hidden nodes. The hidden nodes and output nodes use sigmoid function and purelin function as activation/ transfer functions respectively. Noise variance N_σ is given as input to SI and the A3D-TC parameters such as permeability rate λ, permutation rate $\prod_1$ and permutation rate $\prod_2$ i.e. $X_1^{out} = \lambda$, $X_2^{out} = \prod_1$ and $X_3^{out} = \prod_2$ are obtained as outputs.

The complete model of the SI can be given as

International Journal of Electronics Engineering (ISSN: 0973-7383
Volume 11 • Issue 1 pp. 314-321 Jan 2019-June 2019 www.csjournals.com

$$X_g^{out} = \sum_{h=1}^{N_H} \frac{w_{hg}^{(2)}}{1+\exp(-N_\sigma w_{1h}^{(1)})} : g = 0, 1, 2$$

where, $w^{(1)}$ and $w^{(2)}$ are the weights between input-hidden layers and hidden-output layers respectively.

Results

For simulating Adaptive 3D Turbo Codes using Genetic Algorithm, the following steps are implemented

Part 1

1. Knowledge source is developed with Genetic Algorithm parameters
2. Input knowledge source is given to Special Intelligence (FFNN)
3. Encode with random input bits
4. Add AWGN noise variance
5. Decode data
6. Determine mean Bit Error Rate (BER)

Part 2

1. Select the chromosomes (bits) which have least amount of error
2. Perform Crossover and Mutation and form new population pool
3. Fitness valuation is done in a loop until a maximum number of iterations are reached
4. At the end of the iterations a best set of A3D-TC parameters are obtained

Simulation Evaluation

For every structure, twenty experiments are carried out and the results are presented in the Table I. Average performance is determined for every network structure and directly compared with conventional 3D-TC for different noise variance values (0.15, 0.3, 0.45, 0.6, 0.75).

Table: 1. BER performance of 3D-TC and A3D-TC with network structure having (i) 20 hidden neurons, (ii) 30 hidden neurons and (iii) 40 hidden neurons for different noise variances from different rounds of experiments.

20 Hidden Neurons

Experiment No.	0.15		0.3		0.45		0.6		0.75	
	3DTC	A3DTC_20	3DTC	A3DTC_20	3DTC	A3DTC_20	3DTC	A3DTC_20	3DTC	A3DTC_20
1	0.0908	0.13281	0.041	0.10449	0.0093	0.13623	0.0027	0.048828	0.0094	0.14551
2	0.1558	0.13281	0.1416	0.10449	0.1128	0.053223	0.0449	0.048828	0.0762	0.11426
3	0.1182	0.11328	0.1494	0.10449	0.0491	0.053223	0.0083	0.01123	0.0347	0.038086
4	0.1348	0.11328	0.0786	0.10107	0.1143	0.053223	0.0098	0.01123	0.0107	0.038086
5	0.126	0.11328	0.1328	0.080566	0.0459	0.053223	0.019	0.01123	0.0112	0.038086
6	0.1611	0.11328	0.0195	0.080566	0.1191	0.053223	0.0962	0.01123	0.0107	0.021973
7	0.021	0.11328	0.0444	0.080566	0.0254	0.053223	0.0728	0.007813	0.0009	0.021973
8	0.1328	0.11328	0.0781	0.062012	0.0387	0.053223	0.0202	0.007813	0.0073	0.021973
9	0.0923	0.091309	0.0845	0.062012	0.1113	0.053223	0.0586	0.007813	0.0016	0.021973
10	0.022	0.091309	0.0986	0.062012	0.1001	0.053223	0.0256	0.007813	0.0068	0.004113
11	0.0713	0.091309	0.0205	0.062012	0.0269	0.012695	0.0145	0.007813	0.0037	0.004113
12	0.1313	0.091309	0.0415	0.055664	0.1021	0.012695	0.0337	0.007813	0.0028	0.004113
13	0.0718	0.091309	0.0757	0.055664	0.0908	0.012695	0.127	0.007813	0.0073	0.004113
14	0.0898	0.091309	0.0285	0.055664	0.0186	0.012695	0.0225	0.007813	0.01	0.004113
15	0.0791	0.091309	0.0771	0.055664	0.0747	0.012695	0.0041	0.007813	0.0125	0.004113
16	0.1133	0.091309	0.0283	0.02832	0.0605	0.012695	0.0669	0.007813	0.0041	0.001113
17	0.1162	0.075195	0.0952	0.02832	0.1147	0.012695	0.0894	0.007813	0.0023	0.001113
18	0.1768	0.075195	0.1528	0.02832	0.0806	0.012695	0.0093	0.007813	0.0459	0.001113
19	0.1138	0.075195	0.0518	0.02832	0.0156	0.012695	0.0093	0.007813	0.0249	0.001113
20	0.1279	0.075195	0.04	0.02832	0.0317	0.012695	0.0254	0.007813	0.0303	0.001113
Averages:	0.107305	0.098828	0.073995	0.063427	0.06711	0.037109	0.03801	0.012598	0.015665	0.024608

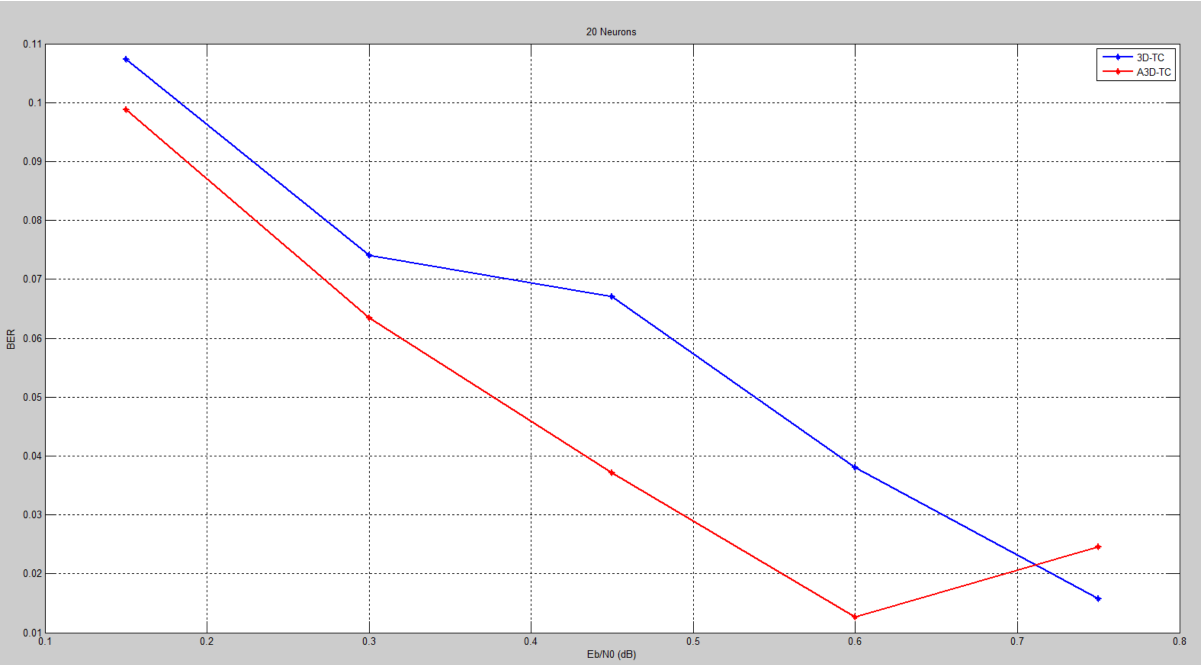
30 Hidden Neurons

Experiment No.	0.15		0.3		0.45		0.6		0.75	
	3DTC	A3DTC_30	3DTC	A3DTC_30	3DTC	A3DTC_30	3DTC	A3DTC_30	3DTC	A3DTC_30
1	0.0908	0.15918	0.041	0.17627	0.0093	0.072266	0.0027	0.083008	0.0094	0.010254
2	0.1558	0.15918	0.1416	0.11572	0.1128	0.039063	0.0449	0.012695	0.0762	0.010254
3	0.1182	0.15918	0.1494	0.11572	0.0491	0.039063	0.0083	0.012695	0.0347	0.010254
4	0.1348	0.15918	0.0786	0.11572	0.1143	0.039063	0.0098	0.012695	0.0107	0.010254
5	0.126	0.15918	0.1328	0.11035	0.0459	0.039063	0.019	0.012695	0.0112	0.010254
6	0.1611	0.12012	0.0195	0.11035	0.1191	0.039063	0.0962	0.012695	0.0107	0.010254
7	0.021	0.099609	0.0444	0.10596	0.0254	0.039063	0.0728	0.012695	0.0009	0.010254
8	0.1328	0.099609	0.0781	0.10596	0.0387	0.039063	0.0202	0.012695	0.0073	0.010254
9	0.0923	0.099609	0.0845	0.10596	0.1113	0.039063	0.0586	0.012695	0.0016	0.010254
10	0.022	0.099609	0.0986	0.10596	0.1001	0.039063	0.0256	0.012695	0.0068	0.010254
11	0.0713	0.086426	0.0205	0.097168	0.0269	0.039063	0.0145	0.012695	0.0037	0.010254
12	0.1313	0.086426	0.0415	0.097168	0.1021	0.039063	0.0337	0.012695	0.0028	0.010254
13	0.0718	0.086426	0.0757	0.097168	0.0908	0.039063	0.127	0.012695	0.0073	0.010254
14	0.0898	0.086426	0.0285	0.014648	0.0186	0.039063	0.0225	0.012695	0.01	0.010254
15	0.0791	0.086426	0.0771	0.014648	0.0747	0.039063	0.0041	0.012695	0.0125	0.010254
16	0.1133	0.086426	0.0283	0.014648	0.0605	0.039063	0.0669	0.007813	0.0041	0.010254
17	0.1162	0.086426	0.0952	0.014648	0.1147	0.039063	0.0894	0.007813	0.0023	0.010254
18	0.1768	0.081055	0.1528	0.014648	0.0806	0.039063	0.0093	0.007813	0.0459	0.010254
19	0.1138	0.081055	0.0518	0.014648	0.0156	0.039063	0.0093	0.007813	0.0249	0.010254
20	0.1279	0.081055	0.04	0.014648	0.0317	0.039063	0.0254	0.007813	0.0303	0.010254
Averages:	0.107305	0.10813	0.073995	0.078101	0.06711	0.040723	0.03801	0.01499	0.01566	0.010254

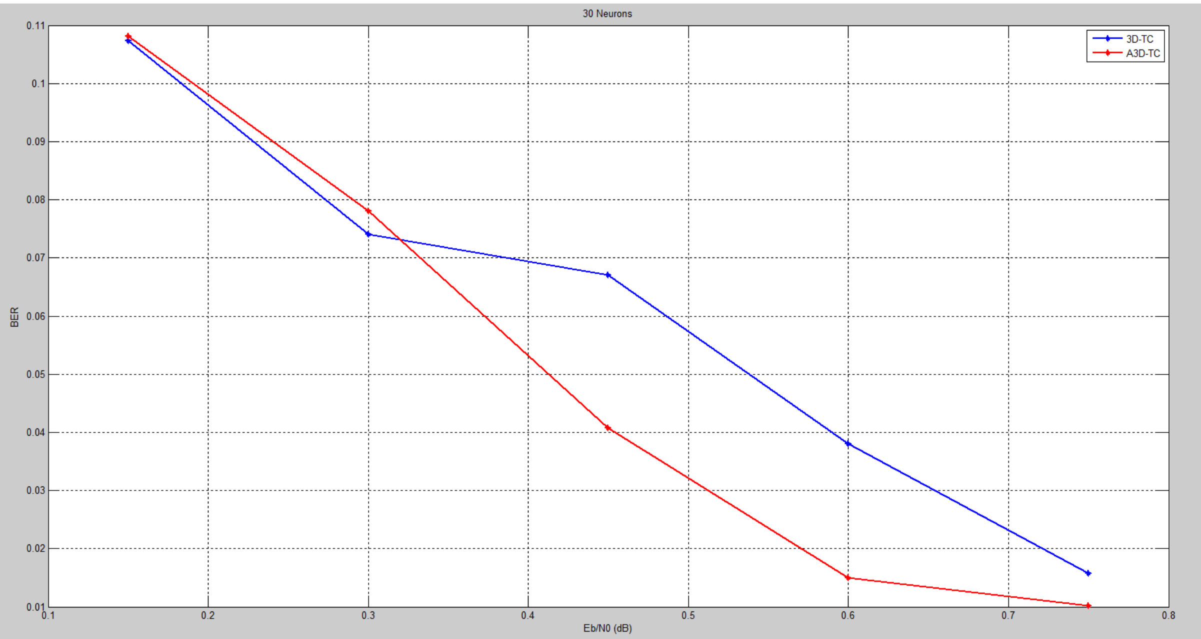
40 Hidden Neurons

Experiment No.	0.15		0.3		0.45		0.6		0.75	
	3DTC	A3DTC_40	3DTC	A3DTC_40	3DTC	A3DTC_40	3DTC	A3DTC_40	3DTC	A3DTC_40
1	0.0908	0.14258	0.041	0.049805	0.0093	0.028809	0.0027	0.12695	0.0094	0.10254
2	0.1558	0.14258	0.1416	0.049805	0.1128	0.028809	0.0449	0.098145	0.0762	0.090576
3	0.1182	0.14258	0.1494	0.049805	0.0491	0.028809	0.0083	0.079102	0.0347	0.073486
4	0.1348	0.14258	0.0786	0.008789	0.1143	0.028809	0.0098	0.079102	0.0107	0.037598
5	0.126	0.12988	0.1328	0.008789	0.0459	0.028809	0.019	0.050781	0.0112	0.037109
6	0.1611	0.12988	0.0195	0.008789	0.1191	0.028809	0.0962	0.050781	0.0107	0.006836
7	0.021	0.12988	0.0444	0.008789	0.0254	0.028809	0.0728	0.050781	0.0009	0.006836
8	0.1328	0.12988	0.0781	0.008789	0.0387	0.028809	0.0202	0.01709	0.0073	0.006836
9	0.0923	0.068848	0.0845	0.008789	0.1113	0.009277	0.0586	0.01709	0.0016	0.006836
10	0.022	0.068848	0.0986	0.008789	0.1001	0.009277	0.0256	0.01709	0.0068	0.006836
11	0.0713	0.068848	0.0205	0.008789	0.0269	0.009277	0.0145	0.01709	0.0037	0.006836
12	0.1313	0.068848	0.0415	0.008789	0.1021	0.009277	0.0337	0.01709	0.0028	0.006836
13	0.0718	0.068848	0.0757	0.008789	0.0908	0.009277	0.127	0.01709	0.0073	0.006836
14	0.0898	0.068848	0.0285	0.008789	0.0186	0.009277	0.0225	0.01709	0.01	0.006836
15	0.0791	0.068848	0.0771	0.008789	0.0747	0.009277	0.0041	0.01709	0.0125	0.006836
16	0.1133	0.068848	0.0283	0.008789	0.0605	0.009277	0.0669	0.01709	0.0041	0.006836
17	0.1162	0.068848	0.0952	0.008789	0.1147	0.009277	0.0894	0.01709	0.0023	0.006836
18	0.1768	0.068848	0.1528	0.008789	0.0806	0.009277	0.0093	0.01709	0.0459	0.006836
19	0.1138	0.068848	0.0518	0.008789	0.0156	0.009277	0.0093	0.014648	0.0249	0.003906
20	0.1279	0.068848	0.04	0.008789	0.0317	0.009277	0.0254	0.014648	0.0303	0.003906
Averages:	0.107305	0.095801	0.073995	0.014941	0.06711	0.01709	0.03801	0.037646	0.015665	0.021899

20 Hidden Neurons



30 Hidden Neurons



40 Hidden Neurons

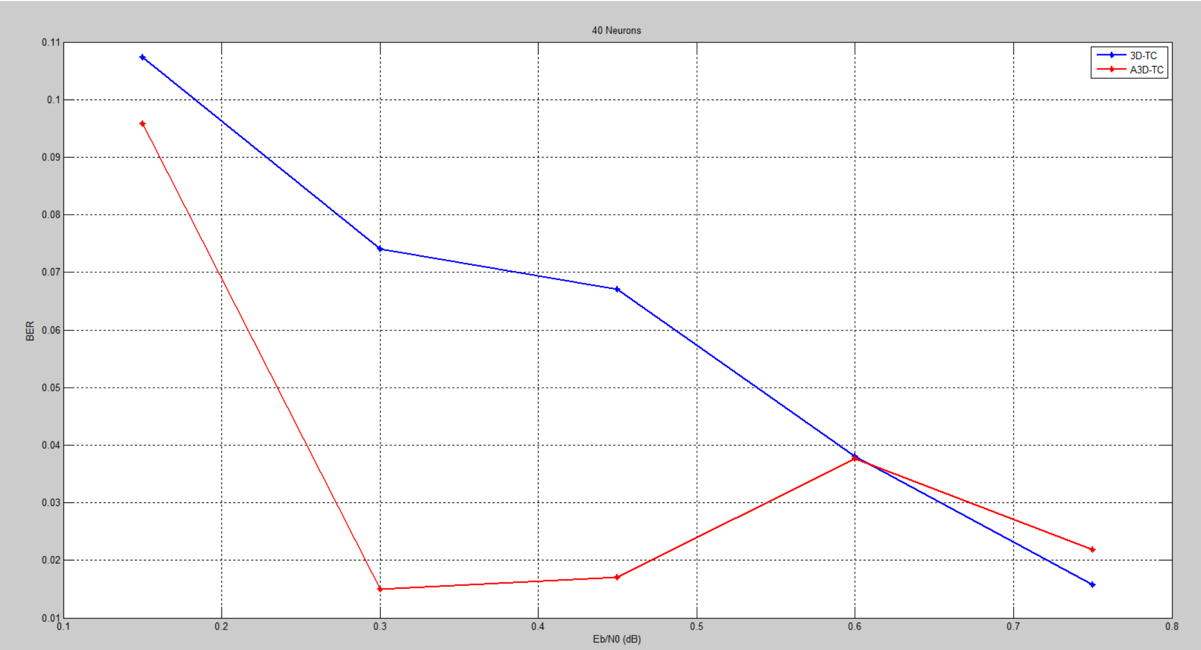


Figure: 4. Comparative Chart for BER vs E_b/N_o performance between A3D-TC (proposed) and 3D-TC (conventional) for network structure with (i) 20, (ii) 30 and (iii) 40 hidden neurons.

Conclusion

It is noticed that A3D-TC using Genetic Algorithm exhibits minimum BER in majority of the experiments except in few noisy environments. However, the failure deviation of A3D-TC is very less and the success deviation is high compared to 3D-TC.

For instance, when network complexity is of 20 hidden neurons, A3D-TC achieves average BER of 0.098828, whereas 3D-TC achieves 0.107305 for 0.15 noise variance. This means A3D-TC has performance deviation of 0.008477 but for other noise variances such as 0.3, 0.45, 0.6 and 0.75, the performance deviation is +0.010568, +0.030001, +0.025412 and -0.008943 respectively. On an average, A3D-TC achieves 0.021993 success deviations whereas the failure deviation is 0.008943 when compared to 3D-TC.

Similarly, for 30 neurons the success deviation is 0.026387 for 0.45 noise variance and for 40 neurons the success deviation is 0.011504 for 0.15 noise variance for A3D-TC. Hence it is found that increasing the network complexity i.e., increasing number of hidden neurons in the selected network, will minimize the BER positively. In A3D-TC, the parameters of third component are made adaptive by using Genetic Algorithm (GA) based knowledge source and feeding it to feed forward neural network by using special intelligence.

A3D-TC using GA has achieved minimum BER in majority of experiments. However, at a few instances of noisy environments A3D-TC has shown contrary result. The observations show that on an average, A3D-TC achieves 0.021993 success deviations whereas the failure deviation is 0.008943 when compared to 3D-TC. For instance, A3D-TC has performance deviation of 0.008477 but for other noise variances such as 0.3, 0.45, 0.6 and 0.75, the performance deviation is +0.010568, +0.030001, +0.025412 and -0.008943 respectively.

When compared to the performance time, the encoder in 3D Turbo Codes completes encoding in 0.0047 seconds whereas for A3D-Turbo Codes, the average encoding time is 0.0036 seconds. Hence using A3D-TC 0.0011 seconds of time is saved when compared to 3D-TC. The proposed solution offers better results due to the fact that the parameters in A3D-TC varying dynamically with channel conditions.

In this paper, A3D-TC is designed to improve the performance and overcome the shortcomings of static nature of permeability and permittivity and make 3D-TC adaptable to various noise environments so that the bit error

International Journal of Electronics Engineering (ISSN: 0973-7383
Volume 11 • Issue 1 pp. 314-321 Jan 2019-June 2019 www.csjournals.com

rate is improved. The implemented solutions are found to show good results compared to the existing method. Hence it is concluded that the proposed A3D-TC is efficient, scalable and robust.

References

[1]. Suman Kshirsagar and E. Nagabhooshanam, "The A3D-TC: An Adaptive Approach for Selecting Third Component Parameters to Generate Robust Turbo Codes", International Journal of Computer Applications, vol. 52, no. 21, pp. 37-42, 2012.

[2].C. Berrou, A. GraelliAmat, Y. Ould-Cheikh-Mouhamedou and Y. Saouter, "Improving the distance properties of turbo codes using a third component code: 3D turbo codes - [transactions letters", IEEE Transactions on Communications, vol. 57, no. 9, pp. 2505-2509, 2009.

[3].R. Lippmann, "Book Review: "Neural Networks, A Comprehensive Foundation", by Simon Haykin", International Journal of Neural Systems, vol. 05, no. 04, pp. 363-364, 1994.

[4]. N. Sellami, A. Roumy and I. Fijalkow, "A Proof of Convergence of the MAP Turbo-Detector to the AWGN Case", IEEE Transactions on Signal Processing, vol. 56, no. 4, pp. 1548-1561, 2008.

[5].D. Le Ruyet and H. Vu Thien, "Asymptotic Performances of Multiple Turbo Codes using the Gaussian Approximation", Citeseerx.ist.psu.edu, 2019. [Online]

[6] P. Kromer, V. Snasel, J. Platos and A. Abraham, "Optimization of Turbo Codes by Differential Evolution and Genetic Algorithms", Softcomputing.net, 2019.

[7].I. Chatzigeorgiou, M. Rodrigues and I. Wassell, "Punctured Binary Turbo-Codes with Optimized Performance", Cl.cam.ac.uk, 2019.