Paper

Support Vector Machine based Decoding Algorithm for BCH Codes

V. Sudharsan and B. Yamuna

Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, India

Abstract—Modern communication systems require robust, adaptable and high performance decoders for efficient data transmission. Support Vector Machine (SVM) is a margin based classification and regression technique. In this paper, decoding of Bose Chaudhuri Hocquenghem codes has been approached as a multi-class classification problem using SVM. In conventional decoding algorithms, the procedure for decoding is usually fixed irrespective of the SNR environment in which the transmission takes place, but SVM being a machine-learning algorithm is adaptable to the communication environment. Since the construction of SVM decoder model uses the training data set, application specific decoders can be designed by choosing the training size efficiently. With the soft margin width in SVM being controlled by an equation, which has been formulated as a quadratic programming problem, there are no local minima issues in SVM and is robust to outliers.

Keywords—BCH codes, Chase-2 algorithm, coding gain, kernel, multi-class classification, Soft Decision Decoding, Support Vector Machine.

1. Introduction

In communication systems, there is an increasing demand for reliable and efficient transmission of data. When data is transmitted over a noisy communication channel, errors are bound to occur. Error control coding techniques are used to detect and correct these errors. The two main types of error correcting codes are block and convolutional codes. Bose Chaudhuri Hocquenghem (BCH) codes are a type of cyclic error correcting block codes with applications in digital, space and satellite communications. Conventional Hard Decision Decoding (HDD) algorithms like Peterson-Gorenstein-Zierler algorithm and Berlekamp-Massey (BM) algorithm have a standard error correcting capability of $t = \left\lfloor \frac{d_{\min}}{2} \right\rfloor$ errors. Though these algorithms have a decent error correcting performance for BCH codes, much research has been on the Soft Decision Decoding (SDD) algorithms to increase the error correction capability. SDD algorithms make use of the channel statistic information, which associates a reliability value to each of the received bit and helps in estimating a more accurate codeword at the receiver.

In the past decade, there has been consistent work on the application of heuristic, evolutionary and artificial intelligence techniques to the decoding problem. These techniques were more robust and had a faster convergence rate. On similar lines, Kao and Berber used SVM, a maximum margin classification technique, for decoding convolutional codes [1]. The same has been extended to discuss the effect of channel and modulation techniques for basic error control coding schemes in wireless applications [2]. In this paper, a SVM based decoding algorithm has been proposed for BCH codes. The proposed algorithm can be used for any (n,k,d) BCH code. The decoding problem has been approached as a multi-class classification problem. The SVM decoder has been programmed and performance comparison has been established against conventional Chase-2 algorithm.

The paper is organized as follows: In Section 2 the existing decoding algorithms for BCH codes are reviewed. Section 3 gives an overview of Support Vector Machines. The proposed decoding algorithm for BCH codes has been explained systematically in Section 4. Section 5 discusses the simulation results of the proposed algorithm followed by conclusion in Section 6.

2. Decoding Algorithms for BCH Codes

BCH codes form a class of powerful error correcting cyclic codes constructed using finite fields. They are known for their multiple error-correcting capabilities and the ease of encoding and decoding [3]. Peterson, gave a decoding algorithm for binary BCH codes based on syndrome decoding. Based on his observation on the linear recurrence in BCH codes, he came up with a set of linear equations, solving which the error locations can be obtained [4]. This algorithm was further generalized to GF(p^m) by Gorenstein and Zierler [5]. Later, Chein devised a fast decoding algorithm for determining the roots of error locating polynomials over finite fields [6].

The Berlekamp-Massey-Forney algorithm is the most commonly used HDD algorithm for BCH codes. The Berlekamp-Massey algorithm is an alternative method to solve Peterson's linear equations to obtain the error location polynomial in a simplified manner [7], [8]. Forney proposed an algorithm for determining the roots of error correcting polynomial [9]. Chase, put forth a class of decoding algorithms that make use of the channel measurement information and claimed a two fold increase in error correcting capability over traditional HDD algorithms [10]. The Least Reliable Positions (LRPs) were identified based on the magnitude of each element in the received vector and the decoded codeword was estimated from a candidate set of probable codewords generated using LRPs. This additional improvement in performance comes with an additional complexity. Reeve and Amarasinghe proposed a parallel Viterbi decoder for cyclic BCH codes since the usual algebraic decoding methods are not readily adaptable for soft decoding [11]. Yingquan formulated a list decoding algorithm for BCH codes to correct upto $1-\sqrt{1-D}$ errors based on Guruswami-Sudan algorithm [12]. A Reliability Level List based decoding algorithm for binary BCH codes – which uses the exact reliability values to arrive at the most probable codeword - has been proposed by Yamuna and Padmanabhan [13]. In the past decade, much effort has been put on the application of heuristic evolutionary algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Neural Networks (NNs) to the decoding problem [14], [15]. They were more robust and had a faster decoding convergence. Azouaoui and Belkasmi applied the heuristic GA to BCH decoding to increase the robustness and efficiency [16]. These algorithms facilitate easier implementation of decoders for Software Defined Radio (SDR) applications, where adaptability is an important factor. To decrease the hardware complexity, an interpolation based one pass Chase decoder was proposed [17] and it was 2.2 times higher in hardware efficiency than Berlekamp in terms of throughput over area ratio. Torres *et al.* attempted a radial basis NN as error correction technique to decode BCH codes [18].

The different hard decision and soft decision schemes proposed in literature have different degrees of performance enhancement and complexity. Attempts on performance enhancement or complexity reduction, trading-off one for the other has been an open problem for researchers.

The procedure of traditional decoding algorithms has the same computational complexity even at a lower noise level. However, modern communication systems need adaptive decoders that cater to changes in channel characteristics. Given the dynamic requirements of emerging trends in channel decoding, in this paper – SVM – a multi-class classification technique has been applied to the decoding problem. The SVM model which is constructed according to the training data is channel adaptive and hence results in a much better performance than conventional methods.

3. SVMs for Data Classification

Support Vector Machines are a class of supervised learning algorithms based on Statistical Risk Minimization (SRM) principle. SVMs analyze the training data, recognize pattern and construct a model. The model is then used for the classification of unknown data. SVMs are generally used for classification and regression [19], [20]. Though SVM was traditionally used for binary classification problems, gradually it was used for multi-class classification problems as well. Cortes and Vapnik formulated a one against all SVM where a multi-class classification problem was converted into N binary classification problems, where N denotes the number of classes [21]. Krebel later came up with a pairwise one versus one approach, involving NC_2 binary classifiers and reduced the unclassifiable regions that occur in one versus all SVM [22]. Studies by Abe, Kao, Hsu and Lin [23]–[25] show that one versus one algorithm is best suited for multi-class classification problem. So this approach has been used in the proposed decoding algorithm.

In a binary classification problem, given a set of labeled, linearly separable training data that belong to two different classes, SVM finds an Optimal Separation Canonical Hyperplane (OSCH), i.e. to achieve the largest minimum distance that separates the data points of one class from the other class and constructs a decision function that defines the margin. Each classifier has a subset of training data – decision variables, x_i – called the support vectors, which are the data points that lie closer to the margin and they characterize the margin. Now, any unknown data can be classified to one of the two classes by evaluating the decision function for that unknown data. Each support vector (SV) has an associated coefficient vector w that defines its role in the classifier. In order to obtain an optimal classifier with minimum number of misclassified data, there is necessity to have maximum margin. Here $\frac{2}{||w||}$ is taken to be the classifier margin.

When the training data are linearly inseparable, we go for a soft margin SVM. To allow inseparability and compensate for the misclassifications, i.e. to accommodate the data that do not have the maximum margin, the non-negative slack variable (ξ) is introduced. Thus for a maximum margin classifier, the SV parameters should be minimized. This has been formulated as a quadratic programming function in SVM. To determine the optimal SV parameters namely coefficient vector w and bias term b, we need to minimize Eq. (1) given below:

$$\frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i, \tag{1}$$

where: w – coefficient vector, C – margin parameter, ξ – slack variable, with respect to the constraint in Eq. (2):

$$y_i(w^T\phi(x_i)+b) \geq 1-\xi_i \quad \text{and} \quad \xi_i \geq 0, \quad i=1,2,\ldots,N, \tag{2}$$

where: $\phi(x)$ – non-linear kernel function, b – bias term, y_i – class label.

In order to maximize the generalization ability and to enhance the classification of non-linear data, the input training data is mapped into a higher dimensional space called feature space using a kernel function. This is called kernel trick. Since the application of SVM to decoding problem comes under the non-linear category, Radial Bias Function (RBF) kernel as given in Eq. (3) has been incorporated to map the input training data into higher dimensional space. Further RBF kernel, which uses Euclidean distance prevents the effect of outliers in performance.

$$K(x_i,x_j) = \mathrm{e}^{-\gamma\|x_i-x_j\|^2}, \quad \gamma \geq 0. \tag{3}$$

4. SVM Based Decoding Algorithm for BCH Codes

Consider a BCH (n,k,d) code, consisting of 2^k message words where k denotes the number of bits in each message word, n denotes the codeword length and d denotes the minimum distance between codewords. With each message word considered as a class, there are $N = 2^k$ classes. These message words are encoded into a codeword of length n. Each bit in the codeword is a feature that defines the class to which the received codeword belongs. Each codeword in the $N = 2^k$ codeword set has a one to one correspondence to a unique message word and the codeword is associated with a class label y_i, where $1 \leq i \leq N$.
SVM based decoding involves two major phases: the training phase and the decoding phase. The decoder model trained and constructed in the training phase is used to classify the received sequence in the decoding phase.

4.1. Training Phase

In training phase, an appropriate model has to be constructed by generating suitable training data. Each codeword of class i is modulated, repeated M number of times and then corrupted by an Additive White Gaussian Noise (AWGN) of SNR, 0 dB.
Now, we have $N \times M$ number of codewords belonging to N different classes, which form the training data for the model. The training is done at a high level of noise at 0 dB, to represent the worst-case scenario and to maximize the generalization characteristic of the decoder. These training data are sent to the pairwise one versus one SVM decoder, where each class of data is compared with another class and NC_2 classifiers (decision functions) are constructed and support vectors (decision variables) which is usually a subset of the training data are obtained.
To develop an optimal model, optimal training parameters should be selected namely, the margin parameter C and kernel parameter γ. This is done using a v-fold cross-validation method. In a v-fold cross validation, the training data is divided into v equal sized subsets. The model is constructed using v-1 subsets as training data and tested with the one remaining set. For each combination of (C, γ), this process is repeated v times. The contour plots of a 10-fold cross validation for BCH $(15,7,5)$ code are shown in Fig. 1. The value of (C, γ) with highest cross validation accuracy is taken as the optimal training parameter. Thus at the end of training phase, we have an optimal decoder model with m (where $m < N \times M$) support vectors.

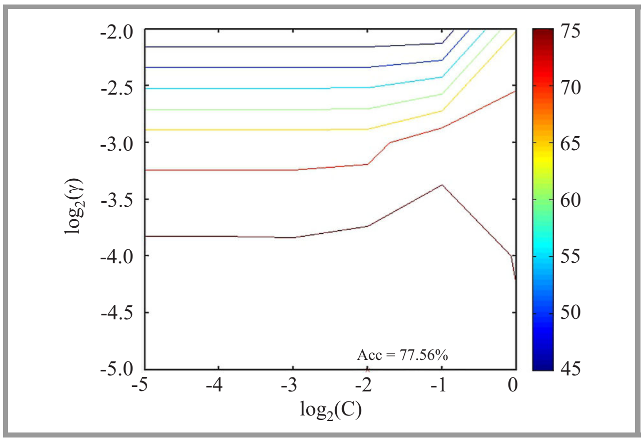

Fig. 1. A 10-fold cross validation done using LIBSVM.
(See color pictures online at www.nit.eu/publications/journal-jtit)

4.2. Decoding Phase

In decoding phase, each of the n bit received soft decision sequence is the unknown data that has to be classified into one of the N different classes, i.e. valid codewords. The decoding phase is thus a simple multi-class classification problem and each classifier is a binary classifier. The noisy received codeword is passed through NC_2 classifiers, where each classifier has a set of support vectors generated during the training phase. The received codeword is transferred into the higher dimensional space using the same RBF kernel function and evaluated using the decision function constructed during the training phase. The output of the decision function determines the class to which the received codeword belongs. This is repeated for all the NC_2 classifiers.
Now, each classifier would have given a vote to one of the N different classes. The received codeword gets decoded to the class which gets maximum number of votes. The output here refers to the maximum value of the decision function, which is directly related to the soft value associated with each received bit. This is known as winner – takes – all (WTA) principle [22]. Since there is a one to one correspondence between the codeword and the message word, the message word can be directly estimated by observing the class value. The proposed SVM based decoding algorithm is given in six steps in Algorithm 1.

Algorithm 1: SVM decoding proposal

1: For a (n, k, d) binary BCH code, each message word in the 2^k set is associated with a class label $y_i(N = 2^k)$.
2: Each message word is encoded into a n-bit codeword to obtain N unique codewords.
3: Each codeword is then transmitted M times through an AWGN channel with SNR= 0 dB.
4: These $N \times M$ codewords along with their associated class label form the training data. The training data set of one class is compared against training data of another class and hence NC_2 classifiers are constructed.
5: Each classifier has an associated set of Support Vectors (decision variables). Thus, the SVM model is constructed.
6: The unknown codeword is now passed through the decoder model and based on the WTA principle, it gets classified to one of the N classes and the corresponding message is obtained.

5. Simulation Results and Discussions

LIBSVM, a software for multi-class SVM classification and regression, has been used for the construction of SVM model and testing of received codeword [26]. The AWGN channel has been considered and Binary Phase Shift Keying (BPSK) is used for modulation. All simulations have been performed using Matlab.

The proposed SVM decoding algorithm has been applied to BCH (15,7,5) code and the performance of SVM based decoding algorithm has been compared against Chase-2 and HDD algorithm as shown in Fig. 2. At a BER of 10^{-3}, the SVM decoder is found to have a coding gain of 0.8 dB over Chase-2 algorithm and a coding gain of 2 dB over HDD algorithm.

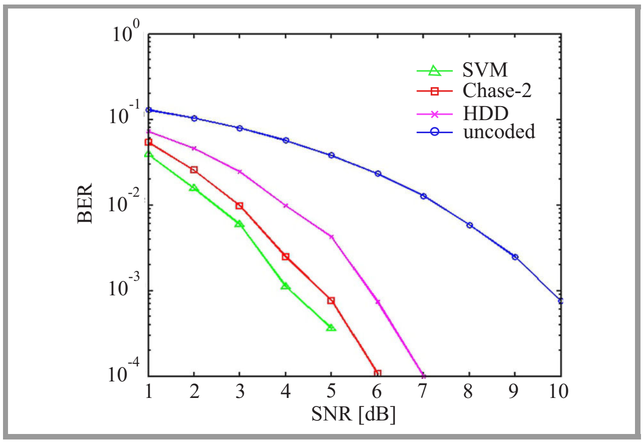

Fig. 2. BER versus SNR plot of BCH (15,7,5) code.

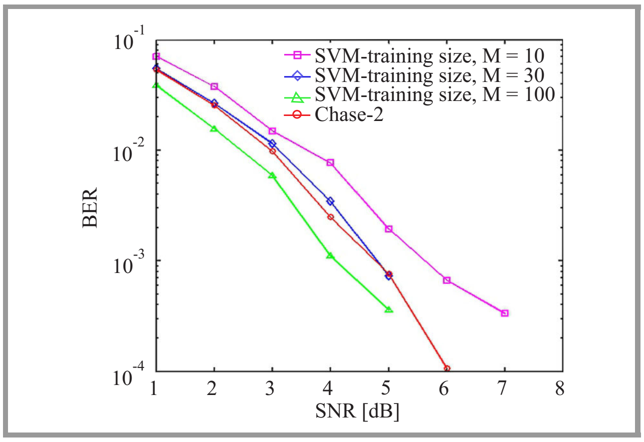

Fig. 3. Performance of SVM decoder for BCH (15,7,5) code at different training size.

Figure 3 shows that the performance of the SVM decoder improves when the training data size is increased. However, the increase in training size in turn increases the number of SVs. This complexity due to increase in SVs can be compensated by puncturing the classifiers, which consistently misclassifies the test data set during cross-validation. Though the increase in training size improves the performance, due to over fitting of data, improvement saturates as shown in Fig. 4.

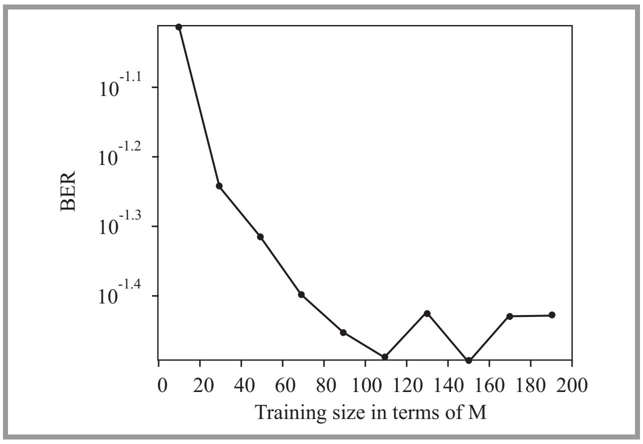

Fig. 4. BER of SVM decoder under different training size with SNR fixed at 1 dB.

Unlike in soft decision decoding algorithms like Chase-2 decoder, SVM based decoder does not involve hard decision error correction, thus eliminating HDD complexity completely. However, when the value of N increases, more classifiers have to be constructed and this results in additional complexity at the testing phase. This can be overcome by cascading SVM decoders. For a N-class problem, initially one decoder can be modeled to classify them into two classes and then two more decoders can be modeled to further classify them into $\frac{N}{2}$ sub-classes, thus reducing the complexity at each decoder. Thus, the proposed SVM algorithm can be combined with the cascading technique and applied to higher block length codes. The additional complexity due to this process is negligible because training is done only once during the initial setup of the communication system. The complexity at the decoding stage depends directly on the number of support vectors generated, which can be controlled according to the application thus striking a trade-off between complexity and performance.

6. Conclusion

This paper presents a SVM based decoding technique for BCH codes, where the decoding problem has been approached as a multi-class classification problem. This algorithm makes maximum use of the channel measurement information combined with the margin based classification feature of the SVM to give an optimal decoder estimate. From the simulation results, it can be seen that the proposed decoding algorithm has a better performance than the conventional Chase-2 algorithm at higher training size. The technique can be applied to higher block length codes by using cascaded SVM. A more generalized decision model, convergence to global optimal solution and prevention of outliers are major leads in this algorithm and thus proves to be efficient for the decoding of BCH codes. The proposed SVM based decoding algorithm can be extended to decoding of high performance robust turbo codes as well.

References

[1] J. Kao and S. Berber, "Error control coding based on support vector machine", in *Proc. 1st IAPR Worksh. Cogn. Inform. Process.*, Santorini, Greece, 2008, pp. 182–187.

[2] R. Ramanathan, N. Valliappan, S. Pon Mathavan, M. Gayathri, R. Priya, and K. Soman, "Generalised and channel independent SVM based robust decoders for wireless applications", in *Proc. IEEE Int. Conf. Adv. Recent Technol. in Commun. Comput. ARTCom'09*, Kottayam, Kerala, Indie, 2009, pp. 756–760.

[3] R. Bose and D. Ray-Chaudhuri, "On a class of error correcting binary group codes", *Inf. Control.*, vol. 3, no. 1, pp. 68–79, 1960.

[4] W. Peterson, "Encoding and error-correction procedures for the Bose-Chaudhuri codes", *IEEE. Trans. Inform. Theory*, vol. 6, no. 5, pp. 459–470, 1960.

[5] D. Gorenstein and N. Zierler, "A class of error-correcting codes in p^m symbols", *J. Soc. Ind. Appl. Math.*, vol. 9, no. 2, pp. 207–214, 1961.

[6] R. Chien, "Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes", *IEEE Trans. Inform. Theory*, vol. 10, no. 4, pp. 357–363, 1964.

[7] E. Berlekamp, "On decoding binary Bose-Chadhuri-Hocquenghem codes", *IEEE Trans. Inform. Theory*, vol. 11, no. 4, pp. 577–579, 1965.

[8] J. Massey, "Step-by-step decoding of the Bose-Chaudhuri-Hocquenghem codes", *IEEE Trans. Inform. Theory*, vol. 11, no. 4, pp. 580–585, 1965.

[9] G. Forney, "On decoding BCH codes", *IEEE Trans. Inform. Theory*, vol. 11, no. 4, pp. 549-557, 1965.

[10] D. Chase, "A Class of algorithms for decoding block codes with channel measurement information", *IEEE Trans. Inform. Theory*, vol. 18, no. 1, pp. 170–182, 1972.

[11] J. Reeve and K. Amarasinghe, "A parallel Viterbi decoder for block cyclic and convolution codes", *Signal Process.*, vol. 86, no. 2, pp. 273–278, 2006.

[12] Y. Wu, "New List Decoding Algorithms for Reed-Solomon and BCH Codes", *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3611–3630, 2008.

[13] B. Yamuna and T. R. Padmanabhan, "A reliability level list based SDD algorithm for binary cyclic block codes", *Int. J. Comput. Commun. Control*, vol. 7, no. 2, pp. 388–395, 2012.

[14] J. Yuan, L. Wang, Q. He, H. Li, and Y. Wang, "A novel genetic probability decoding (GPD) algorithm for the FEC code in optical communications", *Int. J. Light Elec. Opt.*, vol. 124, no. 15, pp. 1986–1989, 2013.

[15] J. Yuan, C. He, W. Gao, J. Lin, and Y. Pang, "A novel hard decision decoding scheme based on genetic algorithm and neural network", *Int. J. Light Electron Opt.*, vol. 125, no. 14, pp. 3457–3461, 2014.

[16] A. Azouaoui and M. Belkasmi, "A soft decoding of linear block codes by genetic algorithms", in *Proc. Int. Conf. Intell. Comput. Syst. ICICS'2012*, Dubai, United Arab Emirates, 2012.

[17] X. Zhang, "An efficient interpolation-based chase BCH decoder", *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 60, no. 4, pp. 212–216, 2013.

[18] H. Torres, M. Jamett, C. Urrea, and J. Kern, "Design of a fault tolerant digital communication system, by means of RBF networks. Comparison simulations with the encoding and decoding algorithms BCH (7,4,1)", *IEEE Latin America Trans.*, vol. 12, no. 8, pp. 1365–1374, 2014.

[19] J. Gokulachandran and K. Mohandas, "Comparitive study of two soft computing techniques for the prediction of remaining useful life of cutting tools", *Int. J. Intell. Manuf.*, vol. 26, no. 2, pp. 255–268, 2013.

[20] J. Gokulachandran and K. Mohandas, "Prediction of cutting tool life based on Taguchi approach with fuzzy logic and support vector regression techniques", *Int. J. Qual. Reliab. Manage.*, vol. 32, no. 3, pp. 270–290, 2015.

[21] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[22] U. Krebel, "Pairwise classification and Support Vector Machines", in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, USA: MIT Press, 1999, pp. 255–270.

[23] S. Abe, *Support Vector Machines for Pattern Classification*. Springer, 2005.

[24] J. Kao, "Methods of artificial intelligence for error control coding and multi-user detection", Ph.D. Thesis, The University of Auckland, New Zealand, 2010.

[25] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines", *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, 2002.

[26] C.-C. Chang and C.J. Lin, "LIBSVM: A library for support vector machines", *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 2, pp. 27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm

V. Sudharsan is pursuing his senior year of undergraduate B.Tech. program at the Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham. His research interests include error control coding, networking, soft computation and artificial intelligence.

E-mail: sud6har2san@gmail.com
Dept. of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham
Amrita University
India

B. Yamuna received her M.E. in VLSI design and Ph.D. in Error Control coding from Amrita School of Engineering, Amrita Vishwa Vidyapeetham, is working as an Associate Professor in the Dept. of ECE at Amrita School of Engineering. She has published and presented papers in international journals and conferences. She has reviewed papers for international and national conferences, and chaired sessions. Her current research interests include coding for reliable communication, spread spectrum communication, VLSI architectures for communication systems, genomic coding and cryptosystems based on error control codes. She is a member of ISTE and IETE.

E-mail: b_yamuna@cb.amrita.edu
Dept. of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham
Amrita University
India