

Performance Enhanced Iterative Soft-Input Soft-Output Decoding Algorithms for Block Turbo Codes

A. Sudharsan V, B. Vijay Karthik V, C. Vaishnavi J S , D. Abirami S, E. B Yamuna
*Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham,
Amrita University, India.
b_yamuna@cb.amrita.edu*

Abstract—Recently, there has been an extensive research on the decoding of Block Turbo Codes (BTCs) achieving near optimum performance at higher noise levels. In this paper, two performances for enhancing novel BTC decoders based on Particle Swarm Optimisation (PSO) and Support Vector Machine (SVM) have been proposed. The decoding latency for the PSO based BTC decoder was much lesser for higher block length BTCs. SVM was adaptable to the channel characteristics and this made it easy to design application specific decoder for BTCs based on SVM.

Index Terms—Block Turbo Codes (BTCs); Chase-2 Algorithm; Genetic Algorithm (GA); Iterative Soft-Input Soft-Output Decoding; Particle Swarm Optimization (PSO) Algorithm; Support Vector Machine (SVM).

I. INTRODUCTION

Block Turbo Codes (BTC) are a group of Forward Error Correction (FEC) codes formed by the serial concatenation of two linear block codes. P Elias introduced a recursive approach and constructed concatenated codes by combining an inner block code with an outer block code and achieved exponentially decreasing error performance [1]. Tanner proposed a hard-in hard-out (HIHO) iterative decoding method for concatenated product block codes [2] and these ‘high code rate’ concatenated codes found widespread applications in deep space communication systems. Berrou developed a class of FEC codes called Turbo codes constructed by either serial or parallel concatenation of convolutional codes as constituent codes [3]. Pyndiah proposed an iterative SISO decoding of Block Turbo Codes, where the decoding of the constituent block codes was carried out using Chase-2 [4] algorithm followed by iterative turbo decoding process [5]. Performance of the iterative decoding process proposed by Pyndiah was improved using Kaneko’s algorithm [6]. A less complex hybrid SISO decoder for BTCs has been proposed [7], reduction in complexity of constituent block code decoding has also been attempted using adaptive Chase algorithm [8] and the decoding latency of BTCs was reduced using a parallel decoder [9]. There has been a consistent research to reduce the complexity [10,11] and improve the efficiency of iterative SISO decoding for modern applications [12].

In the past decade, stochastic techniques have been applied to the decoding of error correcting codes. Belkasmi applied

Genetic Algorithm (GA) to decode Block Turbo Codes (BTCs) using the Most Reliable Basis (MRB) method [13]. Yuan et. al. implemented Genetic Algorithm based BTC decoding on an n dimensional space using the distorted sequences obtained as in Chase-2 algorithm as its initial population [14]. The Genetic Algorithm (GA) based decoding of BTCs has been found to have improved performance over the conventional Chase-2 based iterative SISO decoding algorithm. Particle Swarm Optimization (PSO), another stochastic search algorithm, is found to achieve the same performance as GA and has a faster convergence to global optima. In this paper, a novel decoding scheme for iterative SISO decoding of BTCs based on PSO has been proposed. Support Vector Machine (SVM) – a multi-class classification technique, based iterative SISO decoding algorithm has also been proposed. This algorithm has the advantage of being adaptable to the channel characteristics.

The paper is organized as follows. Section II discusses the BTCs and the traditional SISO iterative decoding algorithm. Section III is dedicated to Particle Swarm Optimization (PSO) and its application in the decoding of BTCs. Section IV explains about Support Vector Machine (SVM) and decoding of BTCs based on SVM. Section V discusses the results obtained using the proposed algorithm followed by the conclusion in section VI.

II. CONVENTIONAL ITERATIVE DECODING OF BLOCK TURBO CODES

Block Turbo Codes or Product Codes are serial concatenated linear block codes arranged in the form of a 2-D matrix. The principle of turbo codes is to interleave the output of first encoder (outer encoder) and feed it to the second encoder (inner encoder). Consider two codes C_1 and C_2 of information bits k_1 and k_2 and block lengths n_1 and n_2 respectively. The Block Turbo Codes consist of $k_1 \times k_2$ sized information bits, where k_1 is the number of columns and k_2 is the number of rows. The k_2 rows are encoded into n_1 columns using G_2 - the generator matrix of code C_2 and k_1 number of columns are encoded into n_2 number of rows using G_1 - the generator matrix of code C_1 . The construction of a Block Turbo Code is pictorially described in Figure 1 [5].

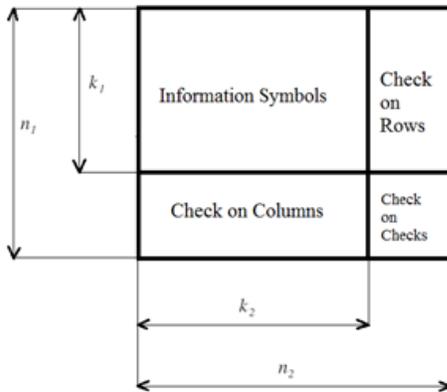


Figure 1: Structure of Block Turbo Code

Although the resulting turbo code is structurally complex, it can be easily decoded by individual constituent decoder iteratively. In the iterative SISO turbo decoding process of Block Turbo Codes (BTCs), Log Likelihood Ratio (LLR) values of received soft valued matrix R are fed as input to the constituent Chase-2 decoder. The LLR values are calculated using the formula in Equation (1).

$$\Lambda(y_i) = \ln \left(\frac{P_r \left\{ c_j = +\frac{1}{r_j} \right\}}{P_r \left\{ c_j = -\frac{1}{r_j} \right\}} \right) = \left(\frac{2}{\sigma^2} \right) r_j \quad (1)$$

Each row or column in the 2-D received matrix is decoded using the corresponding elementary decoder. The traditional SISO decoding proposed by Pyndiah uses Chase-2 algorithm [4]. The Chase-2 algorithm involves the following steps:

- a. Identify the $p = \lfloor d_{min}/2 \rfloor$ Least Reliable Positions (LRPs) using the received soft decision sequence.
- b. Generate 2^p error patterns ' e_i ' at the LRPs and obtain 2^p distorted sequences ' Z_i ' using the hard decision sequence ' y '.

$$Z_i = e_i \oplus y \quad (2)$$

- c. Decode each of the 2^p distorted sequences using an algebraic decoder or a hard decision decoder and add the decoded codeword to the candidate set Ω .
- d. Euclidean distance of each candidate codeword from the original received soft decision sequence is calculated and the codeword with closest Euclidean distance is taken as the decision codeword D . The soft output of the current half iteration is calculated using Equation (3).

$$r'_j = \left(\frac{|R - C|^2 - |R - D|^2}{4} \right) \times d_j \quad (3)$$

where C is the competing codeword in the set Ω which has the second least Euclidean distance to R .

In certain cases, when the complexity of finding the competing codeword increases exponentially with respect to p , the soft output is calculated using Equation (4).

$$r'_j = \beta \times d_j \quad (4)$$

where β is a reliability factor. The extrinsic information for the next half iteration is obtained using Equation (5).

$$w(m+1) = r'_j(m) - r_j(m) \quad (5)$$

The extrinsic information is added to the soft input of the next half iteration as in Equation (6).

$$R(m) = R + \alpha(m) \times w(m) \quad (6)$$

where α is the scaling factor to control the effect of the extrinsic information in $R(m)$ at early iterations. The above delineated steps are then repeated for all the columns in the soft input matrix. This completes one full iteration of the turbo decoding process. The whole process is repeated till the maximum number of iterations is reached.

III. PARTICLE SWARM OPTIMIZATION (PSO) BASED DECODING OF BTCs

The decoding performance of BTCs has been further improved using stochastic search techniques in the recent years. Genetic Algorithm, a heuristic search technique, was incorporated for decoding BTCs by Belkasmı et al. [13] and Yuan et al. [14]. Belkasmı attempted a Most Reliable Basis (MRB) based GA for the decoding of BTC, which involves the search for k most reliable bits, using which the n -bit transmitted codeword can be estimated without the need for a Hard Decision Decoder (HDD). Yuan et al. proposed a Least Reliable Position (LRP) based GA using the distorted sequences obtained in Chase-2 algorithm as its initial population. The decoding scheme involves a HDD at the end to obtain the decoded codeword. In GA, each member in the population is considered as a chromosome and is represented in binary form. The members in the population with the best fitness (Minimum Squared Euclidean Distance - MSED) at the end of each generation are selected. Crossover is done with the selected members from the population to produce offspring with best characteristics of the parents and few bits in the offspring are mutated to induce randomness to the search. This process is repeated for a fixed number of generations to find the optimal codeword.

Though GA based decoding has a larger error correction capability when compared to traditional algorithms, it has higher decoding latency. To overcome this issue, Particle Swarm Optimization (PSO), an evolutionary computational technique is applied to the decoding process of Block Turbo Codes. PSO is a heuristic search algorithm which searches for the global optima by mimicking the flocking behaviour of

birds and migratory pattern of fishes. Each member in the population is compared to a particle in a swarm. Similar to the genetic operators like selection, crossover and mutation, PSO involves velocity calculation, updating the position of each particle and pbest (personal best), gbest (global best) are updated at the end of each iteration (pbest – best fitness attained by a particle in all swarms; gbest – best fitness attained among all particles). At the end of fixed number of iterations, the particle with the best fitness is taken as the optimal codeword [15,16,17].

The biggest advantage of PSO algorithm over GA is its ability to converge to global optima at a greater speed. Belkamsi included 'Elitist' strategy in GA as a part of the selection process to retain the best individual at the end of each generation and improved the performance. PSO algorithm has the Elitism property inherently built i.e. the global best (gbest) in a generation is saved and preserved for every next generation. An LRP based PSO decoding algorithm has been proposed for block codes which employs a hard decision decoder at the end to arrive at a valid decoded codeword [18].

The novel PSO based decoding algorithm for BTCs proposed in this paper uses binary codewords as initial population and follows a MRB based decoding scheme without the use of a hard decision decoder. The proposed PSO based BTC decoding algorithm is elucidated below:

Step 1: Find the hard decision sequence Y from R .

Step 2: Find out the most reliable basis using the following steps.

- a. Sort r_i in decreasing order of reliability. The sorting order defines the column permutation π_1 .
- b. Permute the columns of G such that $G' = \pi_1[G]$.
- c. Form matrix G'' so that its first k columns are the first k linearly independent columns of G' . This defines the column permutation π_2 .
- d. The systematic form of G'' gives the most reliable basis G_s .

Step 3: Fix the objective function. Here it is the Minimum Squared Euclidean Distance (MSED) as given in Equation (7).

$$f_i = \sum_{j=1}^k (r_j - c_j^i)^2 \quad (7)$$

where r_j is the k most reliable bits (MRB) of the received sequence and c_j^i is the i^{th} PSO algorithm population member.

Step 4: Find $R'' = \pi_2[\pi_1[R]]$ and $Y'' = \pi_2[\pi_1[Y]]$.

Step 5: Now assign the first k bits of Y'' and R'' to C_k and R_k .

Step 6: Let N be the number of individuals in the initial population pool. Assign C_k as the first member of the pool. The other $N - 1$ members are randomly generated k bit sequences.

For (present iteration number < total number of iterations)

- a. Calculate the value of f in Equation (6) for all the N initial population vectors.
- b. Update pbest and gbest.
- c. Calculate the velocity v using the formula in Equation (8).

$$v = v + c_1 \times rand \times (pbest - particle) + c_2 \times rand \times (gbest - particle) \quad (8)$$

(c_1 and c_2 are constants whose values are often set to 2. $rand$ is a uniformly distributed random number between 0 and 1).

- d. Calculate the probability for all the bits in the velocity strings of all the particles being equal to 1 using Equation (9).

$$p_{ij}(x_{ij} = 1) = \frac{1}{1 + e^{-v_{ij}}} \quad (9)$$

- e. Update the position based on:

$$x_{ij} = \begin{cases} 1, & p_{ij} \geq U(0,1) \\ 0, & p_{ij} < U(0,1) \end{cases} \quad (10)$$

where p_{ij} is the probability calculated using Equation (9) and $U(0,1)$ is an uniformly distributed random number lying between 0 and 1.

- f. Now the present position vectors become the input for the next iteration.

g. Increment present iteration number by 1.
End For.

Step 7: Among the final population members of the PSO, find the fittest individual. This gives the global k -bit optima D_k .

Step 8: Encode D_k using G_s and apply double inverse permutation to get the decision D .

$$D = \pi_2^{-1}[\pi_1^{-1}[D_k \times G_s]] \quad (11)$$

Step 9: Steps 1-8 is repeated for each row/column decoding and decision matrix D is obtained at the end of each half iteration. The soft output of the current half iteration is calculated using Equation (3).

Step 10: The extrinsic information for the next half iteration is obtained using Equation (5).

Step 11: The extrinsic information is added to the soft input of the next half iteration using the formula given in Equation (6).

Step 12: Steps 1-11 are repeated for each half iteration until the predetermined number of iterations.

The turbo iterative process improves the performance over the specified number of iterations. In the conventional Chase-2 decoding of BTCs, the competing codeword computation increases with respect to p . However, in PSO based BTC

decoding, the competing codeword is the second best candidate in the final population. So the additional complexity of finding the competing codeword is avoided.

IV. SUPPORT VECTOR MACHINE (SVM) BASED DECODING OF BTCs

Traditional decoding algorithms have the same complexity and performance for all applications. To alter the complexity according to the performance required for a particular application, Support Vector Machine (SVM), a margin based classification and regression technique has been used for the decoding of constituent block code in BTC. SVM is based on the Statistical Risk Minimization (SRM) principle. The decoding of constituent block code has been approached as a multi-class classification problem. Based on the training data, SVM recognizes patterns and a model is constructed. Any unknown data can now be classified into one of the classes using the SVM model [19].

Chase-2 algorithm used for the decoding of constituent block code in the traditional SISO decoding has been replaced with a SVM based decoder. In the SVM based decoding algorithm, each row/column in the 2-D received matrix is passed to the SVM decoder and decision D is obtained. The training phase which is performed, deals with the construction of SVM. In the decoding phase, the received soft decision sequence is passed to the SVM decoder model and the class to which it belongs is predicted. Based on the identified class value, the original transmitted codeword is estimated.

A. Training Phase

Each constituent block code (n, k, d_{min}) consists of $N = 2^k$ valid codewords. Each valid codeword in this set is considered as a class. To construct an optimum training model we have to generate training data for each class. This is done by transmitting each modulated valid codeword C_i , ' M ' number of times at a worst case scenario of SNR= 0 dB. Now, we have $N \times M$ number of codewords in training set. Using a kernel function, the training data is now mapped into a higher dimensional space called feature space. Since the decoding of BTC falls under the non-linear category of SVM classification, radial bias function (RBF) kernel has been incorporated to perform the kernel trick [20]. RBF kernel is given by the Equation (12).

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma \geq 0 \quad (12)$$

SVM classifier is now constructed using their training data. Each class of data is compared with data of another class and NC_2 binary classifiers are constructed and decision variables i.e. Support Vectors are obtained. To obtain optimum SVM training parameters namely ' C '- margin parameter and ' γ '- kernel parameter, a ' v -fold' cross-validation (CV) is used [21]. The value of (C, γ) that gives the highest cross validation accuracy is taken as optimal training parameter set.

B. Decoding Phase

In decoding phase, each received soft decision sequence is considered as an unknown sequence for classification. The unknown sequence is now passed to the SVM model and evaluated using all NC_2 classifier. Each classifier will give a vote to one of the N different classes and the final decision is taken based on the winner-takes-all (WTA) principle. The decision D is estimated based on observing the class value [22]. The proposed algorithm for SVM based decoding of BTC is given below:

- Step 1: For every half iteration, each row in the received soft decision matrix is passed to the constituent SVM decoder and the class value c_i is obtained.
- Step 2: Based on the class value c_i , the corresponding codeword can be identified since there is a unique correspondence between the classes and the valid codewords.
- Step 3: The decoded codeword is mapped from $\{0,1\}$ to $\{-1,+1\}$ to obtain the decision codeword ' D '. The process is repeated for all n rows.
- Step 4: The soft output of the current half iteration is calculated using the formula in Equation (3).
- Step 5: The extrinsic information for the next half iteration is obtained by the formula in Equation (4) and is added to the soft input of the next half iteration using the formula as given in Equation (5).

Steps 1-5 are repeated for the subsequent iterations until the predefined number of iterations is reached. The output from the final iteration is taken as the optimal estimate of transmitted codeblock.

V. RESULTS AND DISCUSSIONS

The performance of the proposed algorithm is evaluated in an AWGN channel under BPSK modulation. The results obtained using this algorithm for the BTC $(15,7,5)^2$ are compared against the traditional iterative SISO algorithm. All simulations have been carried out using MATLAB (2015a). The scaling factor α is taken as $\alpha = [0.2 \ 0.2 \ 0.4 \ 0.5 \ 0.7 \ 0.9 \ 1.0 \ 1.0]$ and reliability factor is taken to be $\beta = [0.2 \ 0.4 \ 0.6 \ 0.8 \ 1.0 \ 1.0 \ 1.0 \ 1.0]$. The simulation parameters of PSO based decoding are given in Table 1.

Table 1
Simulation parameters for PSO based BTC decoding

Modulation	BPSK
No. of generations	20
No. of initial population members	20
Transmissions	100
No. of turbo iterations	4

LIBSVM, a software for multi-class classification has been used for the construction of SVM model [23]. The contour plots of 10-fold cross validation for the constituent $(15,7,5)$ code is shown in Figure 2. The simulation parameters of PSO based decoding are given in Table 2.

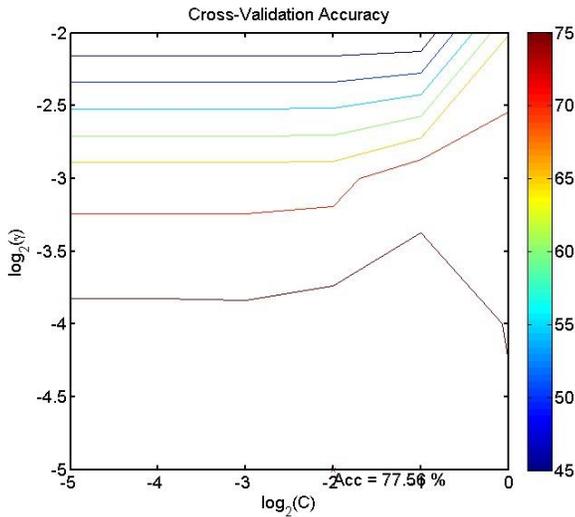


Figure 2: Contour plots of 10-fold cross validation [22]

Table 2

Simulation parameters for SVM based BTC decoding

Modulation	BPSK
Training Size	12800
ν – number of folds in CV	10
C – margin parameter	0.0313
γ – kernel parameter	0.0313

The performance of the proposed PSO and SVM based algorithms for the decoding of BTCs is compared against the conventional Chase-2 and GA based BTC decoding algorithms as shown in Figure 3. When compared to the Chase-2 based BTC algorithm, the PSO based BTC decoding algorithm has a coding gain of about 0.6 dB at a BER of 10^{-4} and the SVM based BTC decoding algorithm gives a coding gain of about 1.5 dB at BER of about 9×10^{-4} .

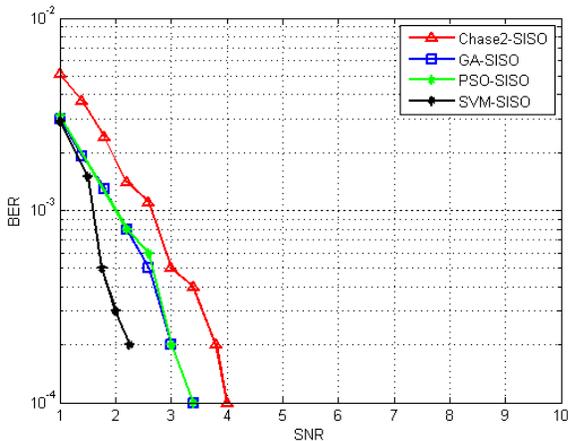


Figure 3: The performance comparison of Chase-2, GA, PSO and SVM based decoding of BTC (15,7,5)² at the 4th iteration

The results obtained from the PSO based decoding are comparable to the performance of GA based decoding algorithm. Though GA and PSO based BTC decoding algorithms have a similar performance, PSO based decoding scheme has a lower decoding latency. The decoding latency for BTC(15,7,5)² and BTC (31,16,7)² is compared against conventional Chase-2 based BTC decoding algorithm, GA

based BTC decoding and PSO based BTC decoding algorithms in Table 3.

Table 3

The time taken for decoding a single BTC(1 frame) block at SNR = 1 Db

BTC	Time Taken for Conventional Chase-2 based BTC decoding	Time Taken for GA based BTC decoding	Time Taken for PSO based BTC decoding
(15,7,5) ²	1.10 s	1.28 s	0.9617 s
(31,16,7) ²	5.34 s	3.39 s	2.47 s

The SVM decoder has a coding gain of about 0.8 dB over GA based BTC decoder. The performance of the SVM decoder depends directly on the training data size used. Higher the training data size, better the performance of the SVM decoder [22]. However, the increase in training size in turn increases the number of SVs. This increase in complexity at the training phase is negligible as it is performed only once for applications involving stationary channel. Thus application specific decoders can be constructed by choosing the training size efficiently based on the performance required.

VI. CONCLUSION

In this paper, a novel PSO and SVM based decoding algorithms of BTCs have been proposed. The proposed decoders consist of the standard SISO decoder with the constituent decoders replaced by PSO or SVM based decoders. The search space of PSO based BTC decoder includes all valid codewords. This increases the error correction capability of the proposed decoder than the conventional Chase-2 based BTC decoding algorithm. In addition, PSO is less complex and has a faster convergence than GA in arriving at the global optima. Based on the application, training size of SVM based BTC decoder is decided accordingly to have optimum performance and complexity, trading off one for the other. SVM can give improved performance with increase in the training data. These decoders can also be extended to BTCs with higher block length codes and non-binary cyclic codes as the constituent codes.

REFERENCES

- [1] P. Elias, "Error-free Coding", Transactions of the IRE Professional Group on Information Theory, vol. 4, no. 4, pp. 29-37, 1954.
- [2] R. Tanner, "A recursive approach to low complexity codes", IEEE Trans. Inform. Theory, vol. 27, no. 5, pp. 533-547, 1981.
- [3] Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes", IEEE Transactions on Communications, vol. 44, no. 10, pp. 1261-1271, 1996.
- [4] D. Chase, "Class of algorithms for decoding block codes with channel measurement information", IEEE Trans. Inform. Theory, vol. 18, no. 1, pp. 170-182, 1972.
- [5] R. Pyndiah, "Near-optimum decoding of product codes: block turbo codes", IEEE Transactions on Communications, vol. 46, no. 8, pp. 1003-1010, 1998.
- [6] S. Dave, Junghwan Kim and S. Kwatra, "An efficient decoding algorithm for block turbo codes", IEEE Transactions on Communications, vol. 49, no. 1, pp. 41-46, 2001.

- [7] Al-Dweik, S. Goff and B. Sharif, "A Hybrid Decoder for Block Turbo Codes", *IEEE Transactions on Communications*, vol. 57, no. 5, pp. 1229-1232, 2009.
- [8] Mahran and M. Benaissa, "Adaptive Chase algorithm for block turbo codes", *Electron. Lett.*, vol. 39, no. 7, pp. 617-619, 2003.
- [9] Argon and S. McLaughlin, "A parallel decoder for low latency decoding of turbo product codes", *IEEE Communications Letters*, vol. 6, no. 2, pp. 70-72, 2002.
- [10] A.J. Al-Dweik and B.S. Sharif, "Closed-Chains Error Correction Technique for Turbo Product Codes", *IEEE Transactions on Communications*, vol. 59, no. 3, pp. 632 - 638, 2011.
- [11] P. Lu, E. Lu and T. Chen, "An Efficient Hybrid Decoder for Block Turbo Codes", *IEEE Communications Letters*, vol. 18, no. 12, pp. 2077 - 2080, 2014.
- [12] L. Wang, M. Wang, L. Wang and F. Yuan, "A joint decoding algorithm for turbo product codes applied in PCM/FM telemetry system", 7th International Congress on Image and Signal Processing (CISP), pp. 826 – 830, 2014.
- [13] M. Belkasmi, H. Berbia and F. El Bouanani, "Iterative decoding of product block codes based on the genetic algorithms", *Source and Channel Coding (SCC)*, pp. 1 - 6, Morocco, 2008.
- [14] J. Yuan, Y. Tian, X. Hu, S. Huang, J. Lin and Y. Pang, "A novel BTC decoding algorithm based on the genetic algorithm in optical communication systems", *Optoelectronics Letters*, vol. 10, no. 2, pp. 123-125, 2014.
- [15] Kennedy, J. ,Eberhart, R.C. , "A discrete binary version of the particle swarm algorithm", *IEEE Conference on Systems, Man, and Cybernetics*, 1997.
- [16] S. Talukder, "Mathematical Modelling and Applications of Particle Swarm Optimization", Master of Science, Blekinge Institute of Technology, 2011.
- [17] P.A. Martin, "Adaptive Iterative Decoding: Block turbo Codes and Multilevel codes", Doctor of Philosophy, University of Canterbury, New Zealand, 2001.
- [18] S. Vadlamudi, H. Kolisetty, V. Chennam, R.R. Reddy, R.S. Bhargav Anand and B. Yamuna, "A fast converging decoding scheme based on particle swarm Optimisation for block codes", *International Journal of Applied Engineering Research (IJAER)*, vol. 10, no. 13, pp. 33161-33164, 2015.
- [19] S. Abe, "Support vector machines for pattern classification." London: Springer, 2005.
- [20] J. Kao, "Methods of Artificial Intelligence for Error Control Coding and Multi-User Detection", Ph.D. Thesis, The University of Auckland, 2010.
- [21] J. Kao and S. Berber, "Error control coding based on support vector machine", *IAPR Workshop on Cognitive Information Processing*, pp. 182-187, 2008.
- [22] Sudharsan. V and B. Yamuna, "Support Vector Machine based Decoding algorithm for BCH codes", *Journal of Telecommunications and Information Technology*, no. 2, pp. 108-112, 2016.
- [23] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1--27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.