International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 13 (2015) pp 33161-33164

A fast converging decoding scheme based on particle swarm optimization for block codes

***Sai Vamsi Krishna Vadlamudi, Harish Kolisetty, Vamsi Krishna Chennam, Reddy Raju R, Bhargav Anand Reddy S, and Yamuna B**
Dept. of Electronics and Communication Engineering
Amrita School of Engineering,
Amrita Vishwa Vidyapeetham,
Coimbatore, Tamil Nadu, India – 641 112, *Phone: +91 9492937280
*Email id: iamvsvk@gmail.com

Abstract—Genetic Algorithm (GA) is a stochastic process that is widely used for obtaining optimized solution. The adaptive heuristic search strategy of GA facilitates its introduction into the decoding process to ensure performance improvement. Like GA, Particle Swarm Optimization (PSO) is yet another optimization algorithm that is widely used. GA and PSO are similar in aspects like random population generation and fitness evaluation. However individuals/particles in PSO does not have genetic operators like crossover and mutation as in GA but updates are done with internal velocities. PSO has the added advantage of updating from the memory which leads to PSO requiring lesser number of iterations compared to GA. This advantage of faster convergence has been exploited in this work to solve the decoding problem. A PSO based decoding scheme for block codes is proposed which has reduced run-time complexity when compared to GA and better performance compared to Chase-II.

Keywords—Particle Swarm Optimization, Chase-II Algorithm, Genetic Algorithm, Linear Block codes.

I. INTRODUCTION

Reliability in communication implies faithful transmission and reception of data. Errors present in digital data can be detected and corrected using error control coding techniques. In today's scenario error control coding has been revolutionized by codes which are capable of approaching Channel Capacity. Conventional Soft Decision Decoding (SDD) involves high decoding complexity compared to Hard Decision Decoding (HDD). When compared to HDD, SDD technique achieves a coding gain of about 2 dB. Sub-optimal SDD algorithms for linear block codes like Chase-II, Generalized Minimum Distance (GMD) and Ordered Statistics Decoding (OSD) have been proposed to approach the theoretical limits of performance [1].

Intelligent algorithms introduced into the decoding technique have resulted in significant achievement in retrieving reliable information. GA is an optimization technique based on evolutionary mechanism. GA takes random search space and tries to exploit the best optimal solution based on ideas of natural selection and genetics like population, crossover and mutation [2]. In GA based SDD algorithms, GA is used for optimizing the search for the most likely codeword. GA based soft decoding has good error correction performance compared to Chase-II algorithm [3]. PSO is another optimization technique motivated by social behavior of bird flocking [4]. Population in PSO is called swarm and each member of population is called particle. An initial velocity and position are assigned to each particle. These particles move in multi-dimensional search space keeping track of the previous best position and finally converge to the best optimal position. PSO is similar to GA with the advantage of having fast convergence. Exploiting this in the decoding process gives the advantage of having fast convergence when compared to GA based SDD algorithm. In this paper, a fast converging decoding scheme based on PSO is proposed for block codes.

This paper is organized as follows: A brief overview of Chase-II algorithm and GA based decoding algorithm are given in section-II. In section-III, the proposed algorithm is presented with simulation results and discussions. Section IV presents the conclusion.

II. CHASE-II AND GENETIC ALGORITHM BASED SOFT DECISION DECODING ALGORITHMS

In GA based SDD approaches the decoding problem develops into an optimal search around the received word for the most likely transmitted codeword. It has been proved that such approaches have better performance when compared to Chase-II algorithm-the most widely used reliability based algorithm.

A. *Chase-II Algorithm*:

Chase-II algorithm uses a set of most likely error patterns. The selection of error patterns is based on the reliability values of the received symbols. The step by step procedure of Chase-II [3] is as follows:

Step 1: Generate test pattern $T_i = [t_{i1}, t_{i2}, t_{i3}, ..., t_{in}]$ where $1 \le i \le 2^{\lfloor d_{\min}/2 \rfloor}$. Of the $2^{\lfloor d_{\min}/2 \rfloor}$ number of test patterns one is zero error pattern and the remaining $2^{\lfloor d_{\min}/2 \rfloor} - 1$ test patterns have all the possible combinations of 1's in lowest reliable $\lfloor d_{\min} / 2 \rfloor$ positions. $d_{\min}$ is the minimum Hamming distance.

Step 2: Add test pattern to the hard decision received sequence to form a distorted sequence S_i as in equation (1).

$$S_i = \mathrm{mod}(M + T_i, 2) \qquad 1 \le i \le 2^{\lfloor d_{\min}/2 \rfloor} \tag{1}$$

Where M is obtained from $U(u_1, u_2, ...u_n)$ -the unquantized output of the matched filter- by using equation (2).

$$M(m_1, m_2, ...m_n), m_i = \begin{cases} 1, u_i > 0 \\ 0, u_i \le 0 \end{cases} \tag{2}$$

Each S_i is now decoded using a hard-decision decoder to obtain E_i -the candidate code sequence.

Step 3: Analog weight metric w is computed using the soft values received from the demodulator for each sequence as in equation (3).

$$w_i = \sum U \cdot (E_i + M) \tag{3}$$

The candidate code sequence with the best metric is selected as most likely transmitted codeword.

B. *Genetic Algorithm based decoding*:
GA is a class of Evolutionary Algorithms (EA) used for optimization process. GA takes a larger search space of possible solutions called population. It repeatedly modifies the population by continuous processes namely Natural selection, Crossover and Mutation. The implementation process of GA based decoding involves the following steps:

Step 1: Population generation: 2^t number of binary vectors are taken as initial population where t is the error correcting capability.
$M(m_1, m_2, ...m_n)$ obtained from $U(u_1, u_2, ...u_n)$ -the unquantized output of the matched filter- by using (4) is taken as one of the initial population [5].

$$M(m_1, m_2, ...m_n), m_i = \begin{cases} 1, u_i > 0 \\ 0, u_i \le 0 \end{cases} \tag{4}$$

The remaining $(2^t - 1)$ individuals of the initial population are generated by equation (5).

$$P_i = \mathrm{mod}(M + T_i, 2) \qquad 1 \le i \le 2^t \tag{5}$$

$T_i = [t_{i1}, t_{i2}, t_{i3}, ..., t_{in}]$ represents the test patterns where $1 \le i \le 2^{\lfloor d_{\min}/2 \rfloor}$. Of the $2^{\lfloor d_{\min}/2 \rfloor}$ number of test patterns one is zero error pattern and the remaining $2^{\lfloor d_{\min}/2 \rfloor} - 1$ test patterns have all the possible combinations of 1's in lowest reliable $\lfloor d_{\min}/2 \rfloor$ positions. $d_{\min}$ is the minimum Hamming distance.

Step 2: Fitness evaluation: The fitness value [6] of each individual in population is calculated by using correlation function as in equation (6).

$$\lambda(P, U) = \sum_{i=1}^{n} (p_i \cdot u_i) \tag{6}$$

Step 3: Natural selection: Selection of individuals from the initial population is done through roulette wheel selection i.e., based on the fitness value of the individuals. Here, higher the fitness of the individual implies greater the probability of an individual to get selected [6].

Step 4: Crossover: Crossover is a process of combining two individuals to produce new individuals [2]. The resultant individuals will have characters from both the previous individuals. One point crossover with a crossover rate of 0.9 has been used in this work.

Step 5: Mutation: Mutation is performed to prevent the individual falling into local optimum. It involves flipping of bits in an individual i.e., from 0 to 1 or 1 to 0.

Once the population generation as in step1 is completed, GA process from step 2 to step 5 are repeated till the termination criteria is met. The termination criteria is decided by fixing a pre-defined value of the number of generations. The output of GA now represents the individual with best fitness value which is then decoded using a hard-decision decoder.

III. PROPOSED FAST CONVERGING DECODING SCHEME BASED ON PARTICLE SWARM OPTIMIZATION

PSO is a stochastic algorithm inspired by social behavior of bird flocking. It iteratively tries to improve the candidate solution by moving the swarm towards the best optimal solution.

PSO based decoding scheme:
Each individual in search space represents a particle and the combination of these particles form a swarm [7]. The position of the particle $[P_i = (p_{i1}, p_{i2}, ..., p_{in})]$ in the swarm is called pbest and the particle with best fitness value among the swarm is called as gbest $[G_i = (g_{i1}, g_{i2}, ..., g_{in})]$. For an optimal search, each particle of the swarm in the entire search space is assigned with a velocity $[V_i = (v_{i1}, v_{i2}, ..., v_{in})]$ [8]. For a linear block code, this particle is of n-dimension type. In PSO, swarm is initialized using the received word along with particles generated using T test patterns with all combinations of 1's in the lowest reliable $\lfloor d_{\min}/2 \rfloor$ positions. Initial velocity of the swarm is fixed as zero. For the next iteration, new velocity and position of the particle is obtained as in equation (10) and equation (11). Fitness value is calculated for the updated particles using correlation function and the fitness value of the new pbests are compared with fitness value of the previous pbests and the best ones are retained. The particle with highest fitness value in the updated swarm is taken as new gbest. This process of iteration continues to move the swarm towards the optimum word.

A step by step procedure of PSO based decoding is as follows:

Step 1: 2^t number of binary vectors are taken as initial population where t is the error correcting capability. $M(m_1, m_2, ...m_n)$ obtained from $U(u_1, u_2, ...u_n)$ -the unquantized output of the matched filter- by using equation (7) is taken as one of the initial population.

$$M(m_1, m_2, ...m_n),\ m_i = \begin{cases} 1, u_i > 0 \\ 0, u_i \le 0 \end{cases} \tag{7}$$

The remaining $(2^t - 1)$ individuals of the initial population are generated by using equation (8).

$$P_i = \mathrm{mod}(M + T_i, 2) \qquad 1 \le i \le 2^t \tag{8}$$

$T_i = [t_{i1}, t_{i2}, t_{i3}, ..., t_{in}]$ represents the test patterns where $1 \le i \le 2^{\lfloor d_{\min}/2 \rfloor}$. Of the $2^{\lfloor d_{\min}/2 \rfloor}$ number of test patterns one is zero error pattern and the remaining $2^{\lfloor d_{\min}/2 \rfloor} - 1$ test patterns have all the possible combinations of 1's in lowest reliable $\lfloor d_{\min}/2 \rfloor$ positions. $d_{\min}$ is the minimum Hamming distance.

Step 2: The fitness value of each individual in the population is calculated by using correlation function as in equation (9).

$$\lambda(P, U) = \sum_{i=1}^{n} (p_i \cdot u_i) \tag{9}$$

Step 3: Velocity is initialized to zero for initial swarm [4]. For the next iteration, new velocity and position of the particle are obtained by using equation (10) and equation (11).

$$V_{i+1} = V_i + r_1 * (pbest - P) + r_2 * (gbest - P) \tag{10}$$

$$P_{i+1} = P_i + V_{i+1} \tag{11}$$

Where r_1 and r_2 are n-dimensional random vectors. P_{i+1} and V_{i+1} are the updated position and velocity of a particle in the swarm.

After few iterations all the particles moves towards the direction of the best optimal solution called gbest thus converging the entire swarm. When the current iteration value matches the pre-defined value of the number of generations, the individual with the best fitness value is taken as output of PSO. This is then decoded using hard-decision decoder.

Results and Discussions:

Decoding using Chase-II algorithm, GA based decoding algorithm and PSO based decoding algorithm are performed with an AWGN model for (15, 7, 5) and (31, 16, 7) BCH codes. The default parameters for GA and PSO based decoding algorithm are given in Table I and Table II respectively.The SNR vs BER plot for (15, 7, 5) BCH code and (31, 16, 7) BCH code are given in Fig. 1 and Fig. 2 respectively.

Table I. Default parameters for Genetic Algorithm

Parameter	**Value**
Generations	20
Cross type	One-point cross
Cross rate	0.9
Mutation rate	0.025
Selection type	Roulette
Channel	AWGN
Modulation	BPSK
Population size	2^t

Table II. Default parameters for Particle Swarm Optimization

Parameter	**Value**
Generations	10
Channel	AWGN
Modulation	BPSK
Population size	2^t

As can be seen from Fig.1 for BCH (15, 7, 5), among the three algorithms when the Bit Error Rate (BER) is 10^{-4}, PSO gains about 1.2 dB over HDD and 0.8 dB over Chase-II algorithm and for BCH (31, 16, 7) in Fig.2 when the BER is 10^{-4}, PSO gains about 1.2 dB over HDD and 1 dB over Chase-II algorithm. The error correction performance of proposed PSO based decoding scheme is better than that of Chase-II and is achieved with less number of iterations compared to GA. In GA, the new generation replaces the previous ones with the recombination complexity of $O(n)$ where n is the size of population. As the replacement strategy depends on the fitness of the individual, a sorting process is required to determine which individual in the generation is to be replaced. This sorting decides the quality of the solution. Another sorting process updates the rank of the individual at the end of each generation. Therefore, the quick sort complexity ranges from $O(n^2)$ to $O(n \log_2 n)$ which is the recombination complexity [9]. In PSO, the complexity of the velocity and position updating processes is $O(n)$ as it does not require any sorting which implies faster convergence in decoding.

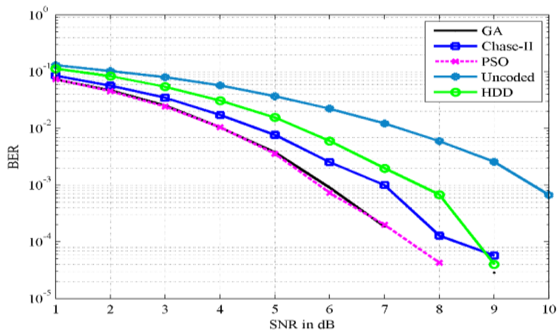

Fig. 1 Performance plot for BCH (15, 7, 5) code

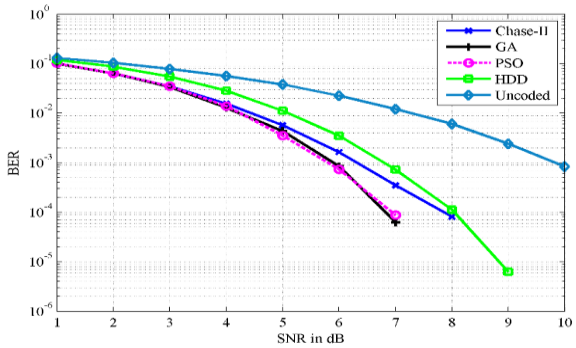

Fig. 2 Performance plot for BCH (31, 16, 7) code

IV. CONCLUSIONS

In the proposed algorithm, the search for the most likely transmitted codeword using the soft information from the demodulator is optimized by particle swarm approach. GA based soft decoding approaches uses more number of iterations to reach the optimal solution. In comparison to this, in the proposed PSO based decoding scheme, the gbest of the swarm moves towards the global optimum using lesser number of iterations since it involves memory based updation. The simulation results show that the proposed PSO based decoding scheme has faster convergence compared to GA and better performance compared to Chase-II.

REFERENCES

[1] Shu Lin, Daniel J. Costello, 2005, "Error Control Coding". Noida: Pearson.

[2] A. Azouaoui and Dr. M. Belkasmi, 2012, "A Soft Decoding of Linear Block Codes by Genetic Algorithms", *International Conference on Intelligent Computational Systems (ICICS'2012)*, Jan. 7-8, Dubai.

[3] D. Chase, 1972, "A Class of Algorithms for Decoding Block Codes with Channel Measurement, Information", *IEEE Trans. Inform. Theory*, vol. 18, pp. 170--181.

[4] Satyobroto Talukder, 2011, "Mathematical Modelling and Applications of Particle Swarm Optimization", *Master's Thesis*, School of Engineering at Blekinge Institute of Technology, Sweden.

[5] Jianguo Yuan, Lin Wang, Qingping He, Hao Li, Yong Wang, 2013, "A novel genetic probability decoding(GPD) algorithm for the FEC code in optical communications", *Optik - International Journal for Light and Electron Optics*, Elsevier, vol. 124, pp. 1986-1989.

[6] Jianguo Yuan, Changwei He, Wenchun Gao, Jinzhao Lin, Yu Pang, 2014, "A novel hard decision decoding scheme based on genetic algorithm and neural network" , *Optik - International Journal for Light and Electron Optics, Elsevier*, vol. 125, pp. 3457-3461.

[7] Jacob Robinson and Yahya Rahmat-Samii, 2004, "Particle Swarm Optimization in Electromagnetics", *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 2, pp. 397-407.

[8] Asanga Ratnaweera, Saman K. Halgamuge and Harry C. Watson, 2004, "Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients", *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp.240-254.

[9] Narmada Nayak and Durga Prasad Mohapatra, 2010, "Automatic Test Data Generation for Data Flow Testing Using Particle Swarm Optimization", *Proceedings of Third International Conference(IC-3'10)*, Part II, pp.8-9, August 9-11, Noida.