International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 5, Number 2 (2012), pp. 151-157

http://www.irphouse.com

Convolutional Code Optimization for Various Constraint Lengths using PSO

Ripple Dhingra[1] and Danvir Mandal[2]

[1]Research Scholar, IET Bhaddal, Punjab, India
E-mail: dhingra_ripple@rediffmail.com
[2]Astt. Professor, Dept. of Electronics Engineering,
IET Bhaddal, Punjab, India

Abstract

For transmitting and retrieving the digital data efficiently, channel coding techniques are used. Convolutional code is the most reliable method for transmitting or retrieving the error free data. Convolutional code encoder consists of shift registers and mod-2 adders. The performance of convolutional code depends upon the connections between shift registers and mod-2 adders. In this paper we are proposing a method for good convolutional code encoder structure for various constraint lengths using particle swarm optimization (PSO).The simulation results show that the proposed algorithm reduces the bit error rate (BER) with increase in constraint length.

Keywords: Convolutional code, PSO (particle swarm optimization), BER (bit error rate)

Introduction

The current development and deployment of wireless and digital communication has a great effect on the research activities in the domain of error correcting codes. Codes are used to improve the reliability of data transmitted over communication channels susceptible to noise. Coding techniques create code words by adding redundant information to the user information vectors. The convolutional codes takes advantage of the relativity between code blocks, so they have better error correction performance and are used widely. Unlike the block code, convolutional code is not memory less devices.

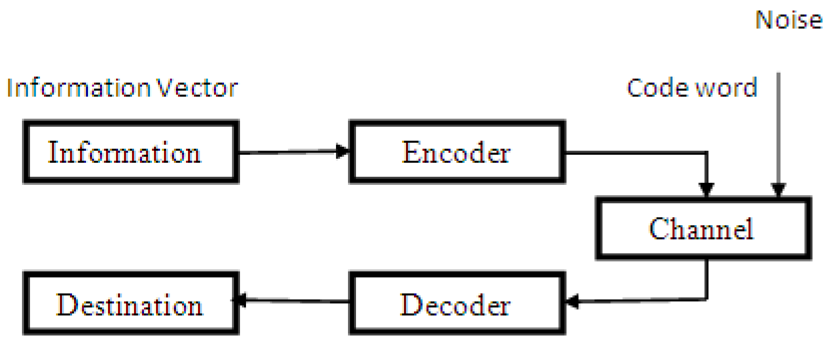

Figure 1: A simplified model of a communication system.

Because of its ability of error control, convolutional codes with longer constraint lengths are widely applied in domains such as satellite communications and digital video. Encoding algorithms generates the code word, which transmitted over the channel(Figure1). Convolutional code accepts a fixed number of message symbols and produces a fixed number of code symbols. Its computation depends not only on the current set of input symbols but also on some of previous input symbols. Convolutional code has many encoder structures (outputs connection with shift registers). The complexity of convolutional code encoder structure increased with the number of states. We have investigated that the PSO algorithm finds to be the best connections for convolutional code encoder.

PSO algorithm [4] has some good features such as good diversity, wide searching area and strong global optimize capability. In this paper we are presenting a method for good convolutional code encoder structure for various constraint lengths using particle swarm optimization (PSO).

Convolutional Code

Convolutional code was introduced by Elias. A convolutional code is a type of code in which each m-bit information to be encoded is transformed into an n-bit symbol. A convolutional code introduces redundant bits into the data stream through the use of linear shift registers as shown in (Figure2). The inputs to the shift registers are information bits and the output encoded bits are obtained by modulo-2 addition of the input information bits and the contents of the shift registers. The connections to the modulo-2 adders were developed heuristically with no algebraic or combinatorial foundation.

A convolutional code is described by three integers, n, k, and K. The code rate R for a convolutional code is defined as

$$R = k/n$$

where k is the number of parallel input information bits and n is the number of parallel output encoded bits at one time interval.

The constraint length K for a convolutional code is defined as

$$K = m + 1$$

where m is the maximum number of stages (memory size) in any shift register.

The number of encoder structure depends upon constraint length(K). For a particular value of K, the number of structure (N) is defined as:

$$N=(2^K-1)*(2^K-1)$$

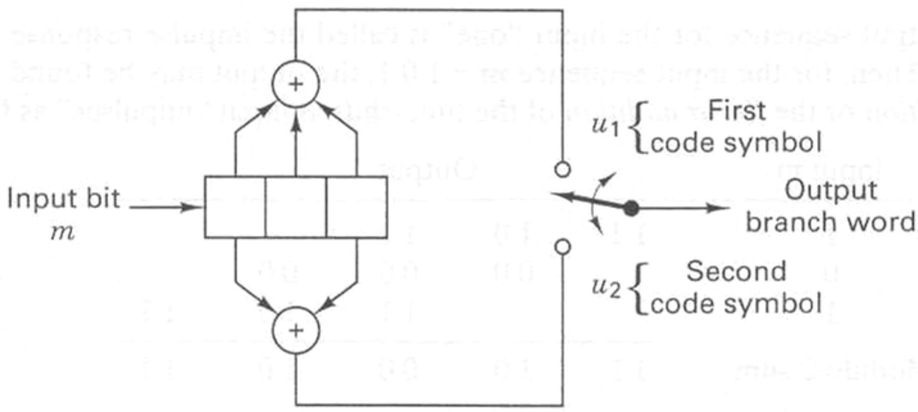

Figure 2: Convolutional encoder (Rate=1/2, K=3)

Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a technique used to explore the search space of a given problem to find the settings or parameters required to maximize a particular objective. This technique, first described by James Kennedy and Russell C. Eberhart in 1995 [6] originates from two separate concepts: the idea of swarm intelligence based off the observation of swarming habits by certain kinds of animals (such as birds and fish); and the field of evolutionary computation. The algorithm maintains a population potential where each particle represents a potential solution to an optimization problem. The PSO algorithm works by simultaneously maintaining several candidate solutions in the search space. During each iteration of the algorithm, each candidate solution is evaluated by the objective function being optimized, determining the fitness of that solution. Each candidate solution can be thought of as a particle "flying" through the fitness landscape finding the maximum or minimum of the objective function.

PSO Algorithm

The PSO algorithm consists of following steps, which are repeated until some stopping condition is met:

1. Initialize the population, location and velocity.
2. Evaluate the fitness of the individual particle (Pbest).
3. Keep track of the individual highest fitness (Gbest).

4. Modify velocity based on Pbest and Gbest location.
5. Update the particle position.

The first three steps are fairly trivial. Fitness evaluation is conducted by supplying the candidate solution to the objective function. Individual and global best fitnesses and positions are updated by comparing the newly evaluated fitnesses against the previous individual and global best fitnesses, and replacing the best fitnesses and positions as necessary.

The velocity and position update step is responsible for the optimization ability of the PSO algorithm. The velocity of each particle in the swarm is updated using the following equation:

$$v_i(t+1) = w.v_i(t) + c_1 r_1 [\, l_i(t) - x_i(t) \,] + c_2 r_2 [\, g(t) - x_i(t) \,] \qquad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (2)$$

where $v_i(t)$ & $x_i(t)$ is the velocity and position of the particle at time t and parameter w, c_1& c_2($0 \le w \le 1.2$,$0 \le c_1 \le 2$ and $0 \le c_2 \le 2$) are user supplied co-efficient. The values r_1 and r_2($0 \le r_1 \le 1$ and $0 \le r_2 \le 1$) are random value regenerated for each velocity update.

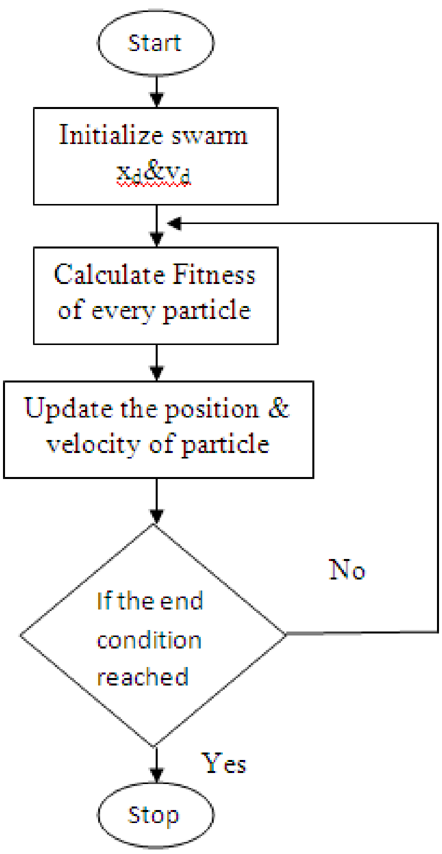

Figure 3: PSO Algorithm

Convolutional Code Optimization Using PSO

Optimization is the mechanism by which one finds the maximum or minimum value of a function or process. Optimization can refer to either minimization or maximization.

Step1 : Generate polynomial

A Polynomial description of convolution encoder describes the connection among shift registers and modulo -2 adders. Build a binary number representation by placing a 1 in each connection line from shift feed into the adder and 0 elsewhere. Convert this binary representation into an octal representation.

Step2 : Draw the trellis

A trellis description of a convolutional encoder shows how each possible input of encoder influences both the output and state transition of encoder. Start with a polynomial description of the encoder and use poly2trellis function to convert it to valid structure.

Step3 : Calculate BER

Calculate bit error rate using octal code and trellis structure. To decode convolutional code use the vitdec function with the flag hard and with binary input data. Because the output of convenc is binary, hard decision decoding can use the output of convenc directly. After convec adds white Gaussian noise to the code with AWGN.

Step4 : Update particle's position and velocity

At each time, all particles have an update. At iteration t, the t^{th} element in the vector is updated. Particle's position is decided by velocity as equation(2). At the decoding process, the update of v_i (t +1) and x_i (t) update must act up to transfer rule of encoder state. Select lowest value of bit error rate as fitness function.

Step 5 : Update personal best position and the global best position.

Update personal best position and the global best position after all particles position have been updated.

Step 6 : Ending condition

When iteration t=L, all particle's position have been updated for L times and reached the grids ending.

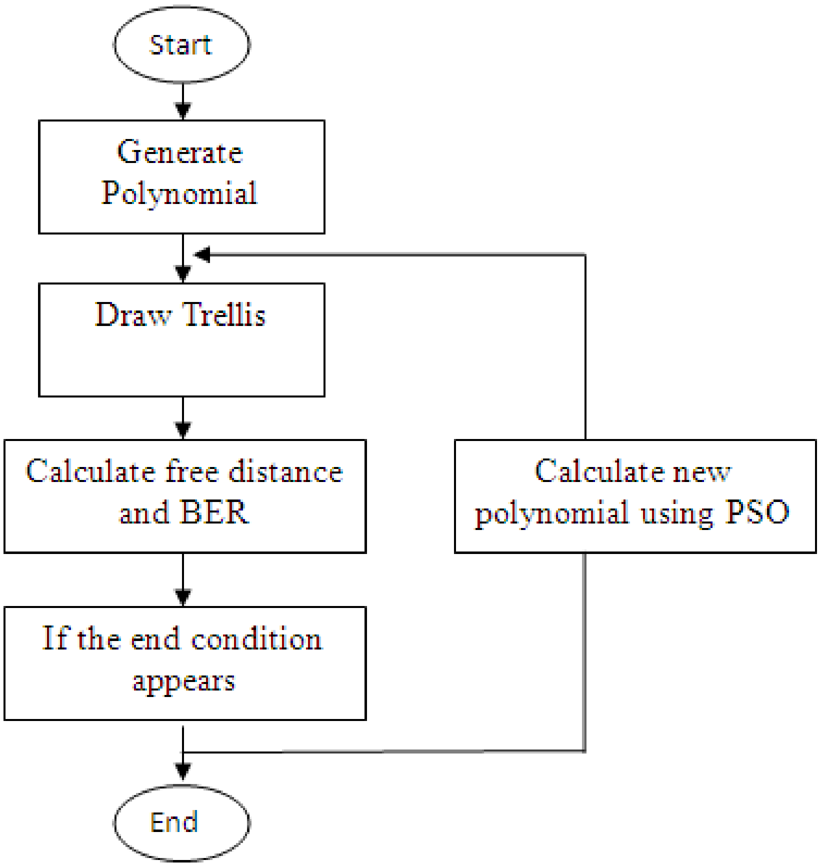

Figure 4: Convolutional Encoder using PSO

Result and Discussion

In the presented paper work a 1/2 rate encoder is design using PSO. Encoder is design using a constraint length from 2 to 10.

Memory Size	Generators		Free Distance	Bit Error Rate
M	G_1	G_2	d_f	BER
2	5	7	5	0.32
3	15	17	6	0.28
4	23	35	7	0.24
5	53	75	8	0.20
6	133	171	10	0.16
7	247	371	10	0.16
8	561	753	12	0.12
9	1131	1537	12	0.08
10	2473	3217	14	0.04

References

[1] Begin, G., Hacooun, D. and Paquin, C., Nov 1990, "Further Results on High-Rate Punctured Convolutional Codes for Viterbi and sequential decoding," IEEE Transaction on information theory, Vol. 38, No. 11, pp. 1922-1928.

[2] Chang, J., Hawang, D. and lin,M., Sept 1997 , " Some extended results on the search for good convolution codes," IEEE Transaction on communication, Vol. 43, No. 5, pp. 1682-1697.

[3] Frenger, P., Orten, P.and Ottosson, T., Nov 1999, "Convolutional codes with optimum distance spectrum," IEEE communication letters, Vol. 3, No. 11, pp. 317-319.

[4] Huang, X., Zhang,Y.and Xu, J., 200 , "Fast decoding of convolutional codes based on PS," IEEE Fourth International Conference on Natural Computation, pp. 619-623.

[5] Berhia,H., Belkasmi, H. and Elbouanani, F., 2008, " On decoding of convolutional codes using Genetic Algorithm," IEEE International Conference on Computer and Communication Engineering, pp. 667-670.

[6] Kennedy, J. and Eberhart, R., 1995, "Particle Swarm Optimization," IEEE International Conference on Neural Networks, Vol 4, pp. 1942–1948.

[7] Zhang, Q., Li, X. and Tran, Q., 2005, "A Modified PARTICLE SWARM OPTIMIZATION Algorithm". IEEE International Conference on Machine Learning and Cybernetics, pp. 2993-2995.

[8] Schutze, O., Tabli, E. and Coello, C., 2007, "A Memetic PSO Algorithm for scalar optimization problems," IEEE Swarm Intelligence symposium.

[9] Virabtamath, S.V. and Attimard, G.V., 2010, "Impact of constraint length on performance of Convolutional CODEC in AWGN channel for image application," International Journal of Engineering Science and Technology, Vol. 2, pp. 4696-4700.

[10] Wesel, R.D., 2002, "Convolutional Code," University of California at Los Angeles, California, pp. 1-8.

[11] Cedervall, M. and Johannesson, R., Nov.1989, "A fast algorithm for computing distance spectrum of convolution codes," IEEE Transaction on Information Theory, Vol. 35, pp.1146-1159.

[12] Fran, Chan, C. and Haccoun, D., Nov 1997, " Adaptive viterbi decoding of convolutional codes over memoryless channels," I*EEE Trans*action on Communications, Vol. 45, No.11..

[13] Shi, Y. and Eberhart, R., 1998, " A modified particle swarm optimizer," IEEE International Conference on Evolutionary Computation, pp. 69–73.

[14] Larsen, K.J. , May 1973," Short convolutional codes with maximal free distance for rates 1/2, 1/3 and 1/4,". IEEE Transaction on Information Theory, Vol. IT-19, pp. 371–372.