International Journal of Electronics Engineering, 4 (1), 2012, pp. 65–67

Convolution Code Encoder Design Using Particle Swarm Optimization

Poonam Beniwal[1] & BS Dhaliwal[2]
[1] ECE Deptt, Om Institute of Engineering & Management, Hisar, India,
[2]GNE, Ludhiyana, Punjab, India
(E-mail: poonammittan249@gmail.com)

Abstract: The error correcting codes are very use useful for transmitting information through noisy channels. Convolution code was introduced in 1995 and then onwards, these found many applications in communication. Convolution code encoder consists of shift registers and mod-2 adders. The performance of convolution code depends upon the connections between shift registers and mod-2 adders. So the particular connection combination affects the error correcting capability of convolution code. The literature many best codes has been purposed for different free distance. In purposed work, we have used particle swarm optimization (PSO) technique for determining the best connection combination. It has been found that our approach leads to an encoder design which has good performance.

Keywords: Convolution code, PSO, Shift Register.

1. INTRODUCTION

In modern digital [1] communication system, error control technology is usually used for reliable communication. Convolution code was first introduced by Peter Elias in 1995. Convolution code is used as error correcting [2] code in a wide variety of communication and recording systems. Convolution code is frequently used to correct error in noisy channel. It has good correcting capability and performs well on bad channels (with error probability of about 10^{-3}). Unlike the block code, convolution code is not a memory less device. Even though convolution code accepts a fixed number of message symbols and produce 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. Convolution code has many encoder structures (outputs connection with shift registers). The complexity of convolution code encoder structure increased with the number of states. We have investigated that the PSO algorithm finds to be the best connections for convolution code encoder.

PSO algorithm [3] has some good features such as good diversity, wide searching area and strong global optimize capability. So the best Convolution code encoder structure based upon particles swarm optimization algorithm is presented in this paper.

2. CONVOLUTION CODE

The Convolution coding is done by combining the fixed number of input bits. The input bits are stored in the fixed length shift register and they are [4] combining with the help of mod-2 adder. This operation is equivalent to convolution and it is called Convolution coding.

Convolution code protects information by adding redundant [5] bits. A rate-k/n Convolution encoder processes the input sequence of k-bit information symbols through one or more binary shift registers (possibly employing feedback). The Convolution encoder computes each n-bit symbol (n > k) of the output sequence from linear operations on the current input symbol and the contents of the shift register(s). Encoder-input information sequences enter [6] the encoder register sequentially at the left. After an information digit shifts out to the channel, each one of the bottom mod-2 adder outputs is tapped and passed to the channel.

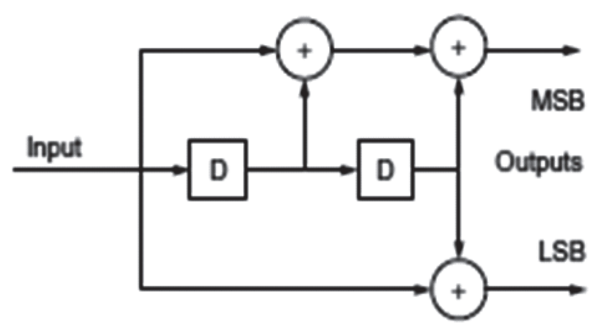

Figure 1: Convolution Code Encoder

Convolution code decoding algorithms infer the values of the input information sequence from the stream of received distorted output symbols. There are three major families of decoding algorithms for Convolution codes: sequential, Viterbi, and maximum a posterior (MAP). Viterbi algorithm is used to find the transmitted sequence (or codeword) that is closest to the received sequence.

The Viterbi algorithm uses the trellis diagram to compute the accumulated distances from the received sequence [7] to the possible transmitted sequences. The free distance d_{free} of a Convolution code is a good indicator of the error correcting performance of the code. As Convolution codes have an exponential growth in the number of states with increasing length, computing free distance is an horrendous job for long memory length codes.. The free distance of a Convolution code determines to a large extend the error rate. The bit error probability for a decoder is calculated by

$$P_b < dT(D,N)/dN \qquad \text{for } N = 1,\ D = 2\sqrt{p}(1\text{-}p)$$

$T(D,N)$ is [6] the generating function of encoder and P the transition probability of the binary symmetric channel Following table represents the output connections with shift register and free distance. For example $R = 1/2$, having number of input 1 and number of outputs 2. If the number of message bits are 50.

Table1
Connection with Shift Registers and Free Distance

k	C_1	C_2	D_{free}	B
8	00000001	11111111	4	17
8	00100000	11111111	6	12
8	01000111	11111111	8	13
8	01000111	11111111	6	14
8	01111011	11111111	6	13
8	01000011	11111111	6	14
8	01011001	11111111	6	15
8	11000111	11111111	10	16
8	01000110	11111111	9	17
8	00000100	11111111	8	14
8	11110000	11111111	6	13

The columns Cl and C2 give the connection polynomials in binary notation. The column d_{free} lists the free distance of the codes. Column B gives the total number of bit errors r. The column K gives the number of shift registers.

3. PSO ALGORITHM

Particle swarm optimization technique provides an evolutionary based search. PSO search [3] algorithm was introduced by Dr. Russ Eberart and Dr. James Kennely in 1995. PSO algorithm is especially useful for parameter optimization in continuous, multi-dimensional search space. The connection to search and optimization problem is made by assigning direction vector and velocity to each point in multi-dimensional search space. These particle evaluate the search space by moving with a particle speed toward the best particle found so far by particular heuristic including their experience from past generation.

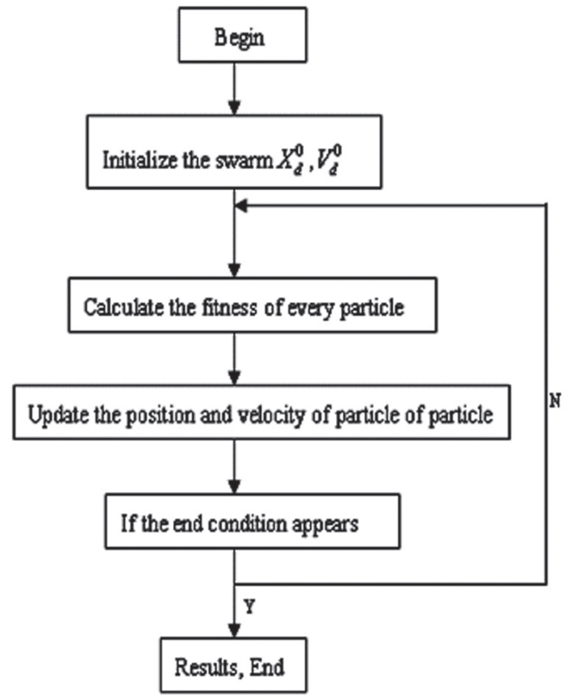

Figure 2: PSO Algorithm

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.
6. Terminate if condition is met.
7. Go to step 3.

After finding the two best values, the particle updates is velocity [3]and position with following equation:

$$v(t+1) = w.v(t) + c1.rp.(plb\text{-}x(t)) + c2.rg.(pgb\text{-}x(t)) \quad (1)$$

$$x(t+1) = x(t) + v(t+1) \quad (2)$$

The $v(t)$ & $x(t)$ is the velocity and position of the particle at time t.The value w, $c1$ and $c2$ ($0 \le w \le 1.2, 0 \le c1 \le 2$ and $0 \le c2 \le 2$) are user supplied co-efficient. The values of rp and rg ($0 \le rp \le 1$ and, $\le rgd \le 1$) and random value regenerated for each velocity update.

4. CONVOLUTION CODE ENCODER DESIGN USING PSO

PSO is limited in optimization solution, especially in solving discrete optimization solutions. So there are some improved algorithms.

Step1: Generate Polynomial

A Polynomial description of convolution encoder describes the connection among shift registers and

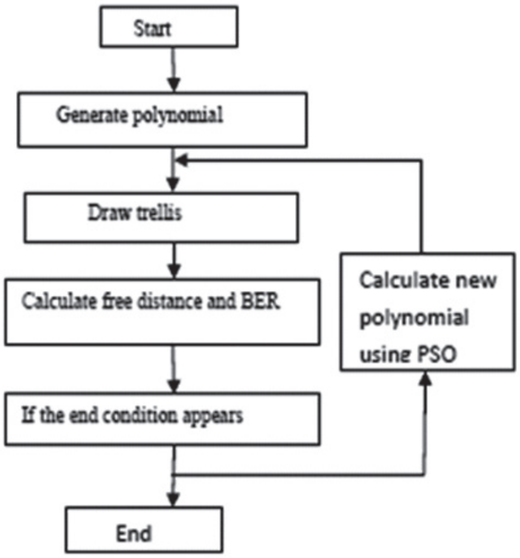

Figure 3: Convolutional Encoder Using PSO

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.

Step 2: Draw the Trellis

A trellis description of a Convolution 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.

Step 3: Calculate BER

Calculate bit error rate and free distance using octal code and trellis structure. To decode Convolution code use the vitdec function with the flag hard and with binary input data. Because the output of convence is binary, hard-decision decoding can use the output of convence directly. After convence adds white Gaussian noise to the code with awgn. Calculate free distance for Convolution code.

Step 4: Update Particle's Position and Velocity

At each time, all particles have an update. At iteration t, the tth element in the vector is updated.

Particle's position is decided by velocity as equation (2). At the decoding process, the update of vi (t +1) and xi (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.

5. RESULT AND DISCUSSION

In the presented paper work a 1/2 rate encoder is design using PSO. Encoder is design using a constraint length of 8.

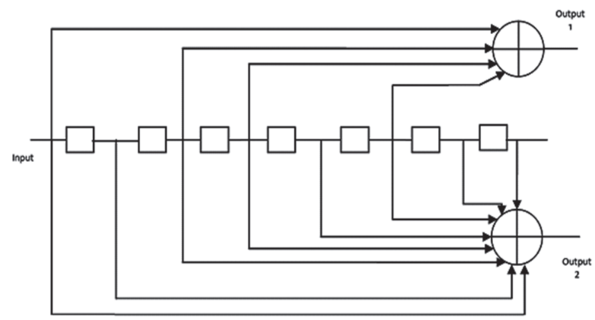

Figure 4: Best Convolution Code Encoder

REFERENCE

[1] A. J. Viterbi, "Error Bounds for Convolution Codes and an Asymptotically Optimal Decoding Algorithm", *IEEE Trans. Inform. Theory,* 13, 260-269, April 1967.

[2] G. D. Forney, Jr., "Convolution Codes I: Algebraic Structure", *IEEE Trans. Inform. Theory,* 16(6), 720-738, Nov. 1970.

[3] Xiaoling Huang, Yujia Zhang, Jinxue Xu and Yongfu Wang, "Fast Decoding of Convolution Codes Based on Particle Swarm Optimization", IEEE Trans. Fourth International Conference on Natural Computation, Nov. 2008.

[4] T. T. Kadota, "Constructive Encoder for Multiple Burst Correction of Binary Convolution Code", *IEEE Transaction on Information Theory,* November 1970.

[5] Richard D. Wesel, "Convolution Code University of California at Los Angeles, California 2002.

[6] A. J. Vinck, J. G. Van and A. J. P., "A Note on the Free Distance for Convolution Code", Z.w.o, ISBN 9061440750, June 1977.

[7] G. D. Fomery, "Structural Analysis of Convolution Code Via Dual Code", *IEEE Transaction of Information Theory,* Vol. 19, pp. 512-518, July 1973.