A fast algorithm for computing BER of convolutional codes encoder with code rate 1/N using PSO

Poonam ¹, Dr. D. P. Singh²,
¹ Research Scholar (Ph.D), ECE, Singhania University, Rajasthan, INDIA,
² Professor, Physics, SNVP, Hisar, Haryana, INDIA
epoonammitan249@gmail.com

ABSTRACT
During the transmission process, the transmitted signals pass through some noisy channel. Due to noise interference, some errors are introduced in the received data. These errors can be detected and correcting using coding technique. The error correcting codes are very useful for transmitting information through noisy channels. Convolutional coding is a Forward acting error-correcting code in digital communications. Convolutional codes are non-blocking codes that can be designed to either error detecting or correcting. Convolution coding has been used in communication systems including deep space communication and wireless communication. A convolutional 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 combined with the help of mod-2 adders. The performance of convolution code depends upon the connections between shift registers and mod-2 adder. Particle swarm optimization (PSO) technique is used for determining the best connection combination. In this paper, we present the best convolutional encoder structure (connection of shift registers with output with the help of mod-2 adder) for different code rates by using PSO.

Key word: Convolutional code, PSO, Mod-2 Adder, Shift register

1. Introduction
Today the use of digital cell phones, the internet, and CD and DVD players is ubiquitous. In all of these cases, digitally represented data is either transmitted from one place to another or retrieved from a storage device when needed. For the proper functioning of these systems, the transmitted or retrieved data must be sufficiently error-free. This can be accomplished efficiently by using channel efficiently coding techniques[8]. One such channel coding technique for achieving reliable data transmission or retrieval is to use convolutional codes. Like a block code, convolutional code is not memory less device, it accepts a fixed number of message symbols and produces a fixed number of output symbols. Convolutional code encoder structure depends upon the connection of output with the shift registers. We have found to be the best connections for convolution code encoder with the help of particle swarm optimization.

Figure1: Digital communication System
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 with different code rate 1/n based upon particles swarm optimization algorithm is presented in this paper[7]. In this paper present the best convolutional code encoder structure (connection of output with the shift registers) with different code rate.

2. Convolutional code
Convolutional codes are used extensively in numerous applications in order to achieve reliable data transfer, including digital video, radio, mobile communication, and communication. Channel coding issued to improve the capacity of a channel by adding some redundant information (extra bits) to the data being transmitted over the channel. Convolutional coding and block coding[6] are the two forms of channel coding. Convolutional codes operate on serial data, one or a few bits at a time. Block codes operate on relatively large message blocks. The major difference of block coding and the convolutional coding is that block coding is memory less. Given a string of k bits, a block coder outputs a unique n-bit data block. Convolutional codes accept a continuous stream of bits and map them into an output stream introducing redundancies in the process. The efficiency of a convolutional code is measured by the ratio of the number of bits in the input, k, and the number of bits in the output, n. In a convolutional code, there is some 'memory' that remembers the stream of bits that flow by. In convolutional code, the block of n code bits generated by the encoder in a particular time instant depends not only on the block of k message bits within that time instant but also on the block of data bits within a previous span of N-1 time instants (N>1). A convolutional code with constraint length N consists of an N-stage shift register (SR) and with modulo-2 adders[5].

Convolutional codes are usually described using two parameters: code rate and constraint length. The code rate = k/n, is expressed as a ratio of the number of bits into the convolutional encoder (k) to the number of channel symbols output by the convolutional encoder (n) in a given encoder cycle[4]. The constraint length parameter, K, denotes the "length" of the convolutional encoder, i.e. how many k-bit stages are available to feed the combinatorial logic that produces the output symbols.

2.1 CONVOLUTIONAL CODE ENCODER
The message bits are fed into a shift register. The set of delay elements in the shift register can be thought of as a state machine which has 2^d possible states where d is the number of delay elements in the register. Codeword bits are generated as functions of the current state and input. These functions are sums[3] of fixed patterns of taps into the shift register. Each codeword is dependent on the current message word and the state of the register which stores information about the past values of the message bits. Therefore successive codeword are not independent of one another. Output of the convolutional code depend upon the shift register connection with the help mod-2 adder. The message bit is shifted to position 'm0', the value of output depend upon the m0,m1,m2, m4,m5,m6

$$OUTPUT_1 = m_0 \oplus m_2 \oplus m_4 \oplus m_5$$  \hspace{1cm} (1).

$$OUTPUT_2 = m_0 \oplus m_1 \oplus m_2 \oplus m_3 \oplus m_5$$  \hspace{1cm} (2).

Figure 2: Convolutional encoder (CR=1/2,CL= 6)
Convolutional code for code rate $\frac{1}{4}$ encoder structure having number of input bit is 1 and number of encoded output bits for one message are 4. The encoder diagram for convolutional code encoder (code rate $1/4$) is shown in figure 3.

\[ \text{OUTPUT 1} = m_0 \oplus m_2 \oplus m_4 \]  
\[ \text{OUTPUT 2} = m_0 \oplus m_2 \oplus m_3 \oplus m_6 \]  
\[ \text{OUTPUT 3} = m_0 \oplus m_3 \oplus m_2 \oplus m_3 \oplus m_4 \]  
\[ \text{OUTPUT 4} = m_0 \oplus m_4 \oplus m_3 \oplus m_5 \]  

The performance of a convolutional code depend on the decoding algorithm employed. Decoding consists of finding the most appropriate path corresponding to the received message to decode. Decoding the codeword with minimal number of error,” the most appropriate path” means the path with the minimal hamming distance with the message to be decoded. The optimal decoding of convolutional code can employ[2] the viterbi decoding and maximum likelihood. Maximum likelihood decoding involves searching the entire code space and generally is impractical because of the large associate computational problem. The viterbi decoding algorithm decodes the convolutional code by choosing a path [6]in the trellis diagram such that the code sequence corresponding to the chosen path is at minimum distance from received sequence. The viterbi decoding algorithm essentially performs minimum hamming distance decoding.

3. PSO algorithm

PSO was developed by James Kennedy and Russell Eberhart in 1995 after being inspired by the study of bird flocking behaviour by biologist Frank Heppner. It is related to evolution-inspired problem solving techniques such as genetic algorithms.

Particle Swarm Optimization is an approach to problems [11]whose solutions can be represented as a point in an n-dimensional solution space. A number of particles are randomly set into motion through this space. At each iteration, they observe the “fitness” of themselves and their neighbours and “emulate” successful neighbours (those whose current position represents a better solution to the problem than theirs) by moving towards them. Various schemes for grouping particles into competing, semi-independent flocks can be used, or all the particles can belong to a single global flock. This extremely simple approach has been surprisingly effective across a variety of problem domains.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is[10] the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest.

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).
5. Update the particle position.
6. Terminate if condition is met
7. Go to step 2

After find the best values, update the velocity of the particle by equation
\[ v(t + 1) = w \cdot v(t) + c1 \cdot r_p \cdot (Pbest - x(t)) + c2 \cdot r_g \cdot (Gbest - x(t)) \] (7)
The value \( w, c1 \) and \( c2 \) (0 ≤ \( w \leq 1.2 \), 0 ≤ \( c1 \leq 2 \) and \( 0 \leq c2 \leq 2 \)) are user supplied co-efficient. The values of \( r_p \) and \( r_g \) (0 ≤ \( r_p \leq 1 \) and 0 ≤ \( r_g \leq 1 \)).
And update the position of particle by equation
\[ x(t + 1) = x(t) + v(t + 1) \] (8)

Figure 4: Flow diagram of particle swarm optimization algorithm

4. PSO based best convolutional code encoder
Particle Swarm Optimization Algorithm is used to find the optimal or best connection of Convolutional code encoder for different constraint length with different code rate

Step1: Generate Polynomial
Polynomial represents the Connection of shift register with output with modulo-2 adder. Connection represent by binary 1(output connect shift register is represent by binary number1, output not connect with the shift register is represent by binary 0). After creating binary polynomial that depend upon the connection convert into in octal form .in this paper we generate the connection polynomial for different code rate convolutional code encoder structure.

Step 2: Draw the Trellis
Trellis represent the convolutional code encoder input influence on output and state transition of encoder. Polynomial is used to draw the trellis diagram of convolutional code. . 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 uses the vitdec function with the flag hard and with binary input data. Because the output of convencode is binary, hard-decision decoding can use the output of convencode directly. After convencode 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 \( n \)th 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.

5. Result
The free distance \( dfree \) of a convolutional code is a good indicator of the error correcting performance of the code. Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12 represent the best connection of output with shift registers for different code rate.
Figure 7: Graph between BER and constraint length for code rate=1/3

Figure 8: Graph between BER and constraint length for code rate=1/4

Figure 9: Graph between BER and constraint length for code rate=1/5
Figure 10: Graph between BER and constraint length for code rate=1/6

Figure 11: Graph between BER and constraint length for code rate=1/7

Figure 12: Graph between BER and constraint length for code rate=1/8
References: