An Efficient Decoding Technique for Catastrophic Error Propagation in Convolutional Codes

Kanchana Katta[1], Dr. Ramesh Ch. Mishra[2]
Department of Electronics and Communication Engineering[1, 2]
Indian Institute of Information Technology Senapati, Manipur[1, 2]
[1]kanch414@gmail.com

Abstract— In a digital communication system the primary goal is to reduce the errors. These errors occur during the data transmission, for reliable communication these errors must be detected and corrected. Error coding is a technique for detecting and correcting errors to guarantee that data is accurately transferred from source to destination. If errors are found during a two-way communication channel between the source and the destination, the receiver can request for the retransmission of information, this scheme for detecting errors is known as Automatic repeat request (ARQ). Moreover, an error control code, also known as forward error correction (FEC), is helpful when retransmission is not possible. Convolutional codes are suited for the FEC technique particularly when the channel mainly exhibits errors in the transmitted signal. This paper mainly focuses on catastrophic error propagation (CEP), this phenomenon is explained clearly using a convolutional code with the help of an efficient maximum-likelihood decoding technique.

Keywords— convolutional codes; catastrophic error propagation; state diagram; state table; viterbi decoder.

I. INTRODUCTION

In every digital communication system, reliable transmission of data or message is not possible. The Data or messages corrupted during transmission can be detected and corrected. Error coding is a technique for detecting and correcting errors to guarantee that data is accurately transferred from source to destination [1]. One such technique, called the automatic repeat request (ARQ), in this method if there is any error has occurred, and sender request to retransmit the message. Another technique is known as forward error correction (FEC), this technique allows for the automatic correction of errors. In this method, the number of errors and the size of the message is important factors. Channel coding provides improved error performance by adding redundant information to the input data being transmitted through a channel. There are two major methods of channel coding techniques: block coding and convolutional coding. noisy channels. In this paper mainly focus on convolutional coding, is a method of adding redundancy to a data stream in a controlled manner to give the destination the ability to correct bit errors without asking the source to retransmit [2].

When the data transmit from the encoder to the decoder or source to destination there are some unpredictable changes in the data transmission because of interference. This interference can change the data to be transmitted when it reaches the decoder. Convolutional codes are superior to block codes for the same implementation complexity of encoder-decoder [3], these convolutional codes are decoded using the Viterbi algorithm. Using this algorithm decoding can be done in two ways; hard decision decoding using Hamming distance as a metric and soft decision decoding that employs Euclidean distance. There are two different kinds errors that can occur in transmission of data. Single bit error: This type of error shows only 1 bit of data is change from 1 to 0 or 0 to 1. Burst error: In this error two or more data bits changes from 1 to 0 or 0 to 1.

These errors can occur in any coding technique. Apart from these errors convolutional codes can exhibit catastrophic error propagation. A catastrophic error is defined as an event whereby a finite number of code symbol errors cause an infinite number of decoded data bit errors [4]. In this paper we mainly focus on CEP and explained clearly with an example. To the best of our knowledge, this work is not there in the existing literature.

The rest of the paper is organized as follows. In section 2 convolutional encoder discussed. Section 3 catastrophic error propagation discussed. In section 4 discussed Viterbi decoder algorithm. The results are discussed in section 5 and finally the paper is concluded in 6.

II. CONVOLUTIONAL ENCODER

Convolutional codes are generally referred to as continuous codes which are infinite length codes. These codes are described by three parameters (n, k, L). Where n is the number of output bits generated from the convolutional encoder, k is the number of input bits applied to the convolutional encoder, and the parameter L is the depth of code memory (i.e number of register stages) is called constraint length. The bit rate or code rate of a convolutional encoder is the ratio of the number the input bits (k) and the number of output bits (n), therefore the data rate defined as $r = k/n$ [5].

The Fig.1 shows the convolutional encoder, it consists of a three fixed shift registers. Each input bit is applied into a shift register, and the output of the encoder is generated by adding the bits together using the mod-2 operation. The number of output bits must be greater than the input bits ($n > k$). A bit rate higher than the message bit rate must be used by the convolutional encoder to produce output bits. A commutator switch is used to interleave the encoded bits which are obtained from the mod-2 adders from the shift registers.

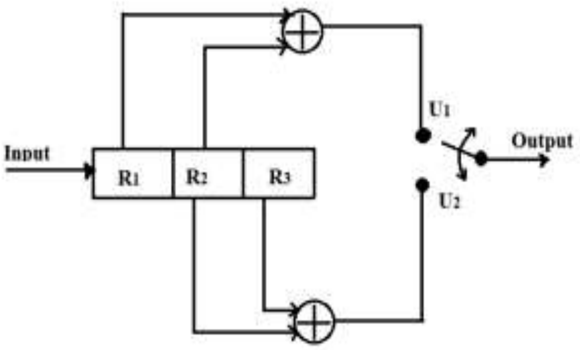

Fig. 1 Convolutional Encoder

III. CATASTROPHIC ERROR PROPAGATION

The convolutional encoder exhibits catastrophic error propagation. A CEP occurs when a finite number of bit errors can produce an infinite number of decoded bit errors, even though the succeeding bits are correct [4]. The concept of CEP can be understood from the following theorems:

Theorem 1: Catastrophic error propagation can be recognised if the generator polynomials have a common factor of degree at least one. The convolutional encoder of bit rate 1/n ensures to avoid catastrophic error propagation having polynomial generator matrix is

$$G(D) = [g_1(D), g_2(D), \text{--------------}, g_n(D)]$$

A necessary and sufficient condition to avoid CEP on the polynomial generator matrix is

$$\gcd[g_1(D)\ g_2(D)\ \text{------------------}g_n(D)] = D^l \quad l \geq 0$$

Theorem 2: Catastrophic error propagation can also be recognised with the state diagram, a state which a nonzero input bit causes a transition back to the same state and produces a all zero output.

In general the generator polynomial is defined by [6]

$$G_{(i)}(D)=g_0^{(i)+}g_1^{(i)}(D)+g_2^{(i)}(D^2)+\ldots\ldots..+g_M^{(i)}(D^M)$$

Where D is unit delay of the bit and M is number of stages of shift registers. The optimum code memory of order M = 3 had $G_1(D)$ =6 and $G_2(D)$ = 3 and generator polynomial is given by $G(D) = [1+D, D+D^2]$. The Figure 1 shows the encoder of (2, 1, 3), the first register R1 holds the incoming input bit and other two R2, R3 holds the states of the code. These two registers have four states represented by 00, 01, 10, and 11.

A. State diagram

A state diagram is one method of describing encoder, consists of different possible states, relationship between input and output bits transistion from present state to next state. The Fig.2 shows state diagram and each state is represented by a circle (a, b, c, and d), it demonstrates all of the encoder's possible state transitions, as well as their output state for each state transition. The transition takes place when the input bit is "0" which is represented by the solid line, and the transition when the input bit is "1" is represented by the dotted line.

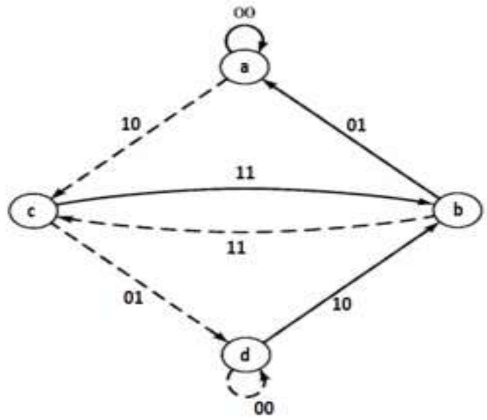

Fig. 2 State Diagram

The convolutional encoder that exhibits catastrophic error propagation shown in Figure 2 at state "d" i.e. 11 results in a transition back to the same state, with a all zero output.

B. State table

The state table is obtained from the state diagram. It is extremely useful to convert the information in the state diagram into tabular form which gives the relationship between input, output, and registers.

Table.1 State table for (2, 1, 3) convolution encoder

Input	State	U_1	U_2
0	00	0	0
1	00	1	0
0	10	1	1
1	10	0	1
0	01	0	1
1	01	1	1
0	11	1	0
1	11	0	0

IV. VITERBI DECODER ALGORITHM

Viterbi decoding was developed by Andrew J. Viterbi and this is the most frequently used algorithms in a variety of disciplines and engineering systems [7]. The Viterbi decoder algorithm performs maximum-likelihood (ML) decoding from the trellis diagram and find the best optimum performance. It has the following blocks

a. Branch Metric Unit (BMU)
b. Path metric Unit (PMU)
c. Add Compare and Select Unit (ACSU)
d. Trace Back Unit (TBU)

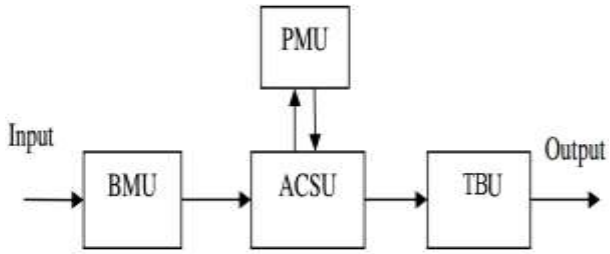

Fig. 3 Viterbi Decoder Blocks

Fig.3 shows the Viterbi decoder blocks. A Viterbi decoder consists of two metrics mainly, those are branch metric and path metric. The branch metric calculates the hamming distance for each state, is the distance between expected and received parity bits. The path metric is a value that corresponds to a trellis state. To calculate the entire trellis state, viterbi decoders employ numerous ACS data paths. Finally, the decoded path is called the surviving path or traceback path which has the smallest metric among the other paths [2].

A trellis diagram can be used to illustrate how the Viterbi algorithm calculates the metric and selects surviving paths. All the branches with all the possible states of outputs are shows for each instant of time. The Figure 4 shows trellis diagram, which is obtained from the state table shown in Table 1.

In trellis diagram the input bit 0 represents with the solid line and the dotted line indicates that the input bit is 1.The initial state at node $t_{0,}$ if input is 0 it will be in the same state 00, if input is 1 it will change to next state 10. At node t_1, if the input bit is 0 it will change from 10 to 01, if the input bit is 1 it will change from 10 to 11 and so on. So for each state transition the outputs are written based on the generator polynomial.

If we transmit the input through the encoder as 10111000, the two mod-2 adder outputs U_1 =11100100 and U_2=01110010. In order to generate the output stream, the commutator switch interleaved by the convolution encoder or codeword is Y=10 11 11 01 00 10 01 00. If we describe message sequence 10111000 in a polynomial form I (D) $=1+D^2+D^3+D^4$.

The output of the encoder at node U_1 is

$$U_1(D) = I(D).G_1(D) = (1+D^2+D^3+D^4).(1+D)$$

$$U_1(D) = 1+D+D^2+D^5 = (11100100)$$

The output of the encoder at node U_2 is

$$U_2(D) = I(D).G_2(D) = (1+D^2+D^3+D^4).(D+D^2)$$

$$U_2(D) = D+D^2+D^3+D^6 = (01110010)$$

Note: $D^i+D^i=0$

With interleaving, the output becomes Y=10 11 11 01 00 10 01 00, which is the same as we obtained using convolution.

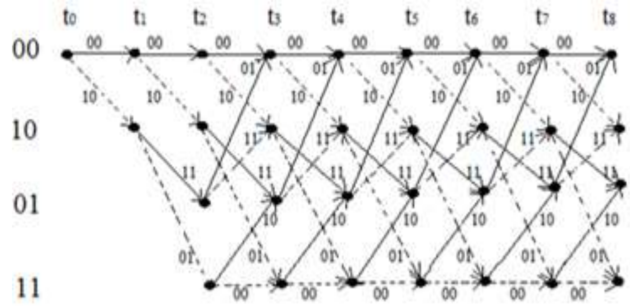

Fig. 4 Trellis diagram for the convolutional encoder

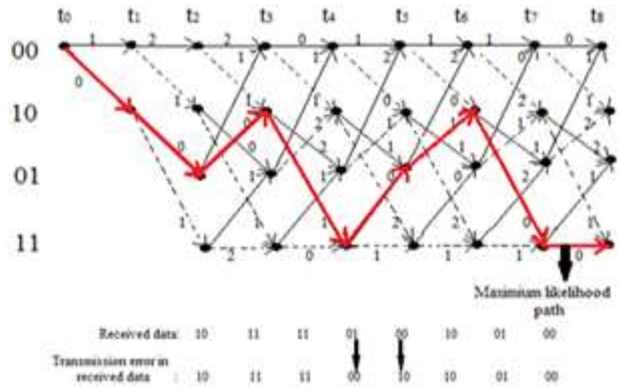

Fig. 5 Trellis diagram for error free decoding

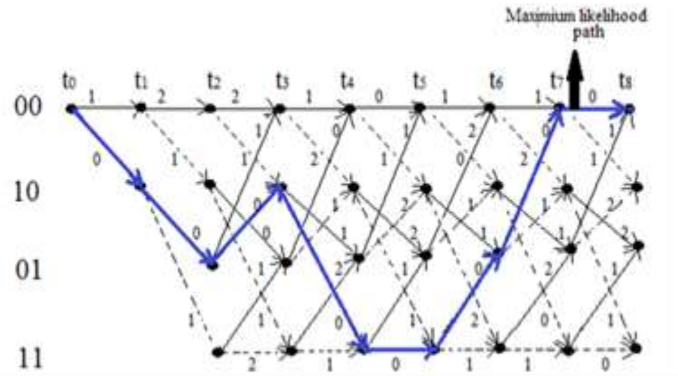

Fig. 6 Trellis diagram with transmission error

Fig. 5 shows trellis diagram with error free decoding, and in each branch of each surviving path the hamming weights are written. The received codeword Y with the minimum hamming distance from node t_0 to t_8 can be the maximum likelihood path shown in the blue color line. Without any errors in the received codeword, we can easily decode the original message sequence.

Now consider the decoder receiving two transmission errors in the received data which is shown in Fig. 6 trellis diagram. The received data with error, the minimum hamming distance from node t_0 to t_8 can be the maximum likelihood path shown in the red color line. Here when we trace this path for the input data we will get 10110111, which is not the same as the input send through the encoder. The encoder with generator polynomial (6, 3) produces the catastrophic error.

V. RESULTS

To check whether the generator polynomial of the convolution encoder that causes catastrophic error propagation or not the following MATLAB functions to use.

Trellis=poly2trellis (constraintlength,codegenerator)
iscatastrophic (Trellis)

poly2trellis: Convert convolutional code polynomial to trellis description.

iscatastrophic: If the trellis corresponds to a convolutional code that causes catastrophic error propagation returns logical 1. Otherwise, it returns logical 0.

For the generator polynomial (6, 3) and the convolution encoder (2, 1, 3), the MATLAB functions are

```
Trellis=poly2trellis (3, [6, 3])
CEP=iscatastrophic (Trellis)
Trellis = numInputSymbols: 2
numOutputSymbols: 4
numStates: 4
nextStates: [4x2 double]
outputs: [4x2 double]
CEP=1
```

Here in the above execution, CEP=1 that means the convolution code exhibits the catastrophic error propagation.
In the similar way we can check for any convolutional code polynomial.

VI. CONCLUSION

In this paper, the catastrophic error in the convolution encoder is shown with an example. The convolution encoder with generator polynomials which has a common factor cannot be decoded. So when we choose an encoder we should be careful that the generator polynomial should not have a common factor.

Here in the above execution, CEP=1 that means the convolution code exhibits the catastrophic error propagation. In the similar way we can check for any convolutional code polynomial.

VII. REFERENCES

[1] Behrouz A. Forouzan, Data communication and networking 5E, India, Edition 2013.

[2] A.Bruce Carlson,Paul B. crilly, communication systems,India ,Edition 2011.

[3] Y.Z. Boutros, G.P. Fiani, E.S. Looka,An efficient technique for detecting catastrophic convolutional codes,Engineering Fracture Mechanics,Volume 46, Issue 6,1993,Pages 1089-1095.

[4] B. Sklar, Digital Communications Fundamentals and Applications. Prentice-Hall, Englewood Cliffs, NJ (1988)

[5] Kanchana Katta,"Design of convolutional encoder and viterbi decoder using MATLAB", International Journal for Research in Emerging Science and Technology, Volume-1, Issue-7, December-2014.

[6] J.Massey and R.Liu ,"Application of Lyapunov's Direct Method to the Error Propagation Effect in Convolutional Codes," IEEE Trans.on Info. Th. IT-10, july, 1964, pp.248-250.

[7] Andrew J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, IEEE Transactions on Information Theory", Volume IT-13, in April, 1967 pages260-269.