Concept of Information Inversion in Forward Error Correction-A Supervised Learning Algorithm for Decoding

*Dr. Piratla Sri Hari

Professor, Department of Electronics and Communication Engineering,

Geetanjali college of Engineering & Technology, Hyderabad, India.

Email: **posttopshari@gmail.com**

Article Info
Page Number: 5575-5584
Publication Issue:
Vol. 71 No. 4 (2022)

Article History
Article Received: 25 March 2022
Revised: 30 April 2022
Accepted: 15 June 2022
Publication: 19 August 2022

Abstract:

The present work aims at proposing a binary forward error correction scheme, where, encoding of an information word depends on its weight (Number of non-zero elements it possesses). The structure of the

overheads of the code word is based whether the weight of the corresponding information word is Even or Odd.

The decoding is implemented using the principle of Supervised Learning, where, the code words are labelled (Label-1 and Label-2) depending on the structure of the individual's overheads. The concept of information inversion proposed to constitute the parity word(overheads) of the codeword made the entire process of encoding simple, unlike the conventional Block coding schemes, where the encoding process is associated with complex mathematics. The concepts of Machine learning, specifically, Supervised Learning used simplified the process of error detection and correction also. Generation of codewords using a simple inverter results a great reduction in the complexity of the hardware realization of the

encoder and the proposed decoding process facilitates a simple detection scheme, relative to conventional process.

Keywords: Information Inversion, Coding efficiency, Overheads, Supervised Learning, Label.

1. Introduction:

Source Coding aims at redundancy reduction in the encoded data and the source encoded data is ready for transmission. The transmitted signal power, bit transmission rate and the channel noise spectral properties are responsible for not having the received data with reasonably low error rate. The remedy in such cases is to introduce redundancy in the data being transmitted, which counteracts the errors introduced by the noisy channel. Such process of introducing redundancy to the information word being transmitted is referred to as Channel Coding or Error Control Coding.

An (n, k) binary Block code is one of such Error control coding schemes, where each k bit information word $M = [m_1 \ m_2 \ - \ - \ m_k]$ is transformed independently into an n bit code word $C = [c_1 \ c_2 - - c_n]$. [11]

In the present paper, a systematic single error correcting binary Block code with n=2k and with a feature of information inversion is proposed.

Each proposed code consists of 2^k code words, with a unique mapping between each M and C. Supervised learning-based decoding principle is proposed for extracting the information word from the received code word at the receiver. [1]

2. Code words with information inversion:

The structure of the redundant(overheads) word of each C of the proposed code may not be the same as other code words of the same code and depends on the weight (number of non-zero elements) of each associated M.

2.1. Structure of the overheads:

The code word of the proposed code is expressed as $C = [m_1 \ m_2 \ - \ - \ m_k \ r_1 \ r_2 \ - \ - \ r_k]$ where $M = [m_1 \ m_2 \ - \ - \ m_k]$ is the ‘k′ bit information word and $r = [r_1 \ r_2 - - r_k]$ is the k bit redundant word, which constitutes the overheads in each code word.

- In the C of an M with zero/even weight, $r = M$.
- In the C of an M with odd weight, $r = \overline{M}$ i.e., inverted information bits.

Thus, the concept of information inversion is implemented in the generation of the code words.

The following Table 1 illustrates the unique mapping between a M and the corresponding C of an (8,4) code.

Table.1. Some code words of an (8,4) code

Information word M				Weight of M	Corresponding Code word C							
m_1	m_2	m_3	m_4		c_1 $(=m_1)$	c_2 $(=m_2)$	c_3 $(=m_3)$	c_4 $(=m_4)$	c_5 $(=r_1)$	c_6 $(=r_2)$	c_7 $(=r_3)$	c_8 $(=r_4)$
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	2	0	0	1	1	0	0	1	1
0	1	1	1	3	0	1	1	1	1	0	0	0

2.2. Decoding (Supervised Learning): [8]

All the 2^k code words of the code are grouped under two labels: [4],[7]

- Label 1: Code words with overheads=Information bits
- Label 2: Code words with overheads= inverted information bits.

All these code words will be constituting the training data for the decoder (decoding algorithm). [6]

Decoding aims at classifying the received codeword to a group under any one of the above labelled.

C under each label satisfies the following:

'+' being the Modulo-2 sum, for each of the code words, there will be k summations, such that,

❖ Under the label "overheads=Information bits"

○ $\sum_j (m_j + r_i) = 0$, j= 1,2, ... k (1)

❖ Under the label "overheads= Inverted Information bits"

○ $\sum_j (m_j + r_i) = 1$, j= 1,2, ... k (2)

2.2.1. Classification: [5],[6]

Let the code word at the receiver be $C'= [m_1 \quad m_2 \quad - \quad - \quad m_k \quad r_1 \quad r_2 \quad - \quad - \quad r_k]$, which is the test data for the decoder and may not be the same as the sent i.e., C.

The decoder has to classify C′ to any of the two labeled groups. [5],[10]

For C′

Under Label-1

✓ M + r = $[0\ 0\ ..0]_{1Xk}$, under error free reception

✓ $M + r$ consists of $(k - 1)$ number of 0s, under reception with error

Under Label-2

✓ $M + r = [1\ 1\ ..1]_{1Xk}$ under error free reception

✓ $M + r$ consists of $(k - 1)$ number of 1s, under reception with error

2.2.2. Estimating the Error Location:

Deviation in any of the $'k'$ summations is an indication that either m_j or r_j of that summation is in error.

Each individual errors results in results in a corrected code word, among which, only one belongs to group of code words under Label-1 or Label-2 and the other will not be a valid code word. Thus, the correct codeword can be estimated and classified into any one of the two groups. [5]

3. Error Detection and Correction-Analysis:

- Let the received code word of an (8, 4) proposed code be $C' = [m_1m_2m_3m_4r_1r_2r_3r_4] =$ [1 0 0 1 0 1 0 0]
- The 4 (=k) Modulo 2 summations of the information bits and redundant bits of C' are
  $m_1 + r_1 = 1 + 0 = 1; m_2 + r_2 = 0 + 1 = 1; ; m_3 + r_3 = 0 + 0 = 0; m_4 + r_4 = 1 + 0 = 1.$
- Since $M + r = [1\ 1\ 0\ 1]$ is consisting of 3 $(= k - 1)$ 1s, it is the case of Label-2 i.e., overheads= Inverted Information bits
- Either m_3 or r_3 of C' can be in error.
- Considering m_3 is in error, the corrected code word is [1 0 1 1 0 1 0 0]
- Considering r_3 is in error, the corrected code word is [1 0 0 1 0 1 1 0]
- The later i.e. [1 0 0 1 0 1 1 0] is not a valid code word of the code, since the information word [1 0 0 1] being of even weight, the overheads should be same as the information bits, and the corresponding code word should be

[1 0 0 1 1 0 0 1].

- Thus, for the received code word $C' = [m_1m_2m_3m_4r_1r_2r_3r_4] =$ [1 0 0 1 0 1 0 0], the transmitted code word is [1 0 1 1 0 1 0 0] and thus is classified under the group with Lable-2.
- $[m_1m_2m_3m_4]$ of the corrected code word constitutes the information word.

4. Software implementation:

<u>4.1.</u> <u>Code word generation (Encoding)</u>: [3]

Various stages of encoding process are indicated in Figure 1.

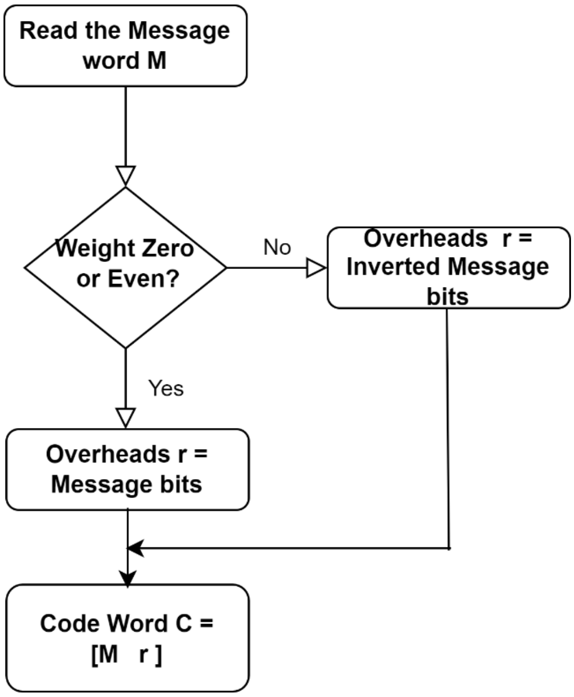

Fig.1. Encoding-Flowchart

4.2. Decoding (Error Correction): [2], [3],[9]

The steps involved in the process of Error-Detection and Correction can be explained using Figure 2.

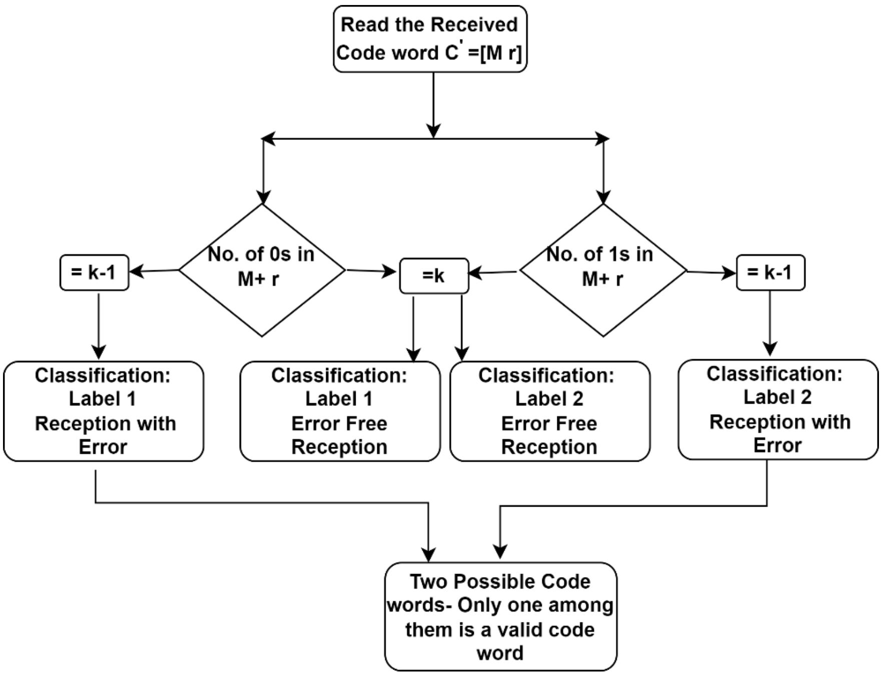

Fig.2. Error Detection and Correction-Flowchart

4.3. Scilab Code for Encoding and Decoding:

```
clear all
clc
//Encoding
k=input ('Enter the information word length')
n=2*k;
MI= 0:(2^k)-1;
M=bitget(MI,k:-1:1);
C=zeros (2^k, n);
//C gives all the possible code words of the code.
for i=1:2^k
   C(i,1:k)=M(i,:);
end
for i=1:2^k
        if modulo(length(find(M(i,:))),2)==0
           C(i,k+1:n)=M(i,:);
       else
            C(i,k+1:n)= bitcmp(M(i,:),1);
end
end
//Decoding
C1=input ('Enter the received code word')
S=bitxor(C1(1:k),C1(k+1:2*k));
S1=find(S);
S11=length(S1);
S2=find(S==0);
S22=length(S2);
X=zeros(1,2*k);
```

```
Y=X;
select S11
  case 0 then
   disp('The reception is error free and the code word is classified under Label-1')
  case k then
   disp('The reception is error free and the code word is classified under Label-2')
  case 1 then
   X(S1)=1;
   Y(S1+k)=1;
   disp("The code word is classified under Label-1")
end
if S22==1
   X(S2)=1;
   Y(S2+k)=1;
   disp("The code word is classified under Label-2")
   end
   codeword1=bitxor(X,C1)
   codeword2=bitxor(Y,C1)
for i=1:2^k
   if codeword1==C(i,:)|codeword2==C(i,:)
      disp('The code word sent is:')
      disp(C(i,:))
      disp('The information word sent is:')
   disp(C(i,:)(1:k))
end
end
```

5. Case Study:

Consider an (8,4) code.

❖ **Output of the executed Scilab code for Encoding and Decoding:**

- Enter the information word length 4
- The possible codewords of the code are

C:

0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	1.	1.	1.	1.	0.
0.	0.	1.	0.	1.	1.	0.	1.
0.	0.	1.	1.	0.	0.	1.	1.
0.	1.	0.	0.	1.	0.	1.	1.
0.	1.	0.	1.	0.	1.	0.	1.
0.	1.	1.	0.	0.	1.	1.	0.
0.	1.	1.	1.	1.	0.	0.	0.
1.	0.	0.	0.	0.	1.	1.	1.
1.	0.	0.	1.	1.	0.	0.	1.
1.	0.	1.	0.	1.	0.	1.	0.
1.	0.	1.	1.	0.	1.	0.	0.
1.	1.	0.	0.	1.	1.	0.	0.
1.	1.	0.	1.	0.	0.	1.	0.
1.	1.	1.	0.	0.	0.	0.	1.
1.	1.	1.	1.	1.	1.	1.	1.

- Enter the received code word [1 0 0 1 0 1 0 0]

"The code word is classified under Label-2"

"The code word sent is": 1. 0. 1. 1. 0. 1. 0. 0.

"The information word sent is:" 1. 0. 1. 1.

The unique mapping between all the possible information words and the codewords of the proposed (8,4) code is given in Table 2.

Table.2. Information words and Codewords of an (8, 4) code

Information word M				Corresponding Code word C							
m_1	m_2	m_3	m_4	c_1 $(= m_1)$	c_2 $(= m_2)$	c_3 $(= m_3)$	c_4 $(= m_4)$	c_5 $(= r_1)$	c_6 $(= r_2)$	c_7 $(= r_3)$	c_8 $(= r_4)$
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1	0	1	0	0
1	1	0	0	1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	1	0	0	1	0
1	1	1	0	1	1	1	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1

6. Discussion:

The proposed Forward Error correction schemes under the concept of Information Inversion are single error correcting codes and are of 50% coding efficiency. The probability of undetected error [2], [11] i.e., the probability for the receiver to miss the error undetected is also found to be considerably low, which can be concluded form the weight enumerator of the code, which is $1 + 14\alpha^4 + \alpha^8$ for the (8, 4) code i.e. One code word of the code is of zero weight; Fourteen codewords of the code are of weight 4; and one codeword is of weight 8. When the code is only used for error detection, under a transition probability of 0.1 with a Binary Symmetric Channel, the probability of undetected error for an (8, 4) code using Mc William's identity is found to be 0.0009185, which can be interpreted as out of if there are 10^4 code digits present at the detector, on average 9 incorrect digits will be undetected. This can be further improved with increase in the code word length. (10)

7. Conclusion:

The channel coding schemes under the concept of Information Inversion are found to possess considerably better probability of undetected error and under the principle of Machine Learning-Supervised learning used for decoding, are able to detect and correct the single random error present in the received code word and the information word is properly identified, which is to be conveyed to the final recipient of information. To enhance the error correcting capability of the codes, the concept of Product codes can be used, where the proposed codes can be the constituent codes.

8.References:

1. P. SriHari & Dr.B.C.Jinaga (2007)-"DATA INVERTING CODES"-Ph.D Thesis, Jawaharlal Nehru Technological University, Hyderabad, India.
2. Kevin P.Murphy-Machine learning-A Probabilistic Perspective-The MIT Press Cambridge, Massachusetts London, England,2012
3. Michael Paluszek, Stephanie Thomas-MATLAB Machine learning-Apress,2017
4. Pratap Dangeti -Statistics for Machine Learning- - Packt Publishing, UK,2017
5. Tobias Gruber, Sebastian Cammerer, Jakob Hoydis, Stephan ten Brink - On deep learning-based channel decoding- in Proceedings of Conference: 2017 51st Annual Conference on Information Sciences and Systems (CISS),2017. DOI:10.1109/CISS.2017.7926071
6. Dr. M. Gopal-Applied Machine Learning-McGraw Hill Education (India) Private Limited, 2018
7. Ruise He and Zhiguo Ding (Edited by)-Applications of Machine learning in Wireless Communications-IET The institution of Engineering and Technology, IET Telecommunication series 81,2019.
8. Fabrizio Carpi , Christian Häger , Marco Martalò , Riccardo Raheli , and Henry D. Pfister- Reinforcement Learning for Channel Coding: Learned Bit-Flipping Decoding- arXiv:1906.04448v2 [cs.IT] 21 Aug 2019 ; https://doi.org/10.48550/arXiv.1906.04448
9. The MathWorks Inc.- Matlab Statistics and Machine Learning Toolbox User's Guide, R2020a, March,2020, 18-2 to 18-6 p.
10. Yihan Jiyang -Deep Learning for Channel Coding- Ph.D Dissertation, University of Washington, 2021, 1-6 p
11. Venkatesan Guruswami, Atri Rudra, Madhu Sudan- Essential Coding Theory-Department of Computer Science and Engineering, University at Buffalo, SUNY. Work supported by NSF CAREER grant CCF-0844796, 2022, 43-62p.