On Trellis Structure of Error Correction Codes

S. Adishesh and M.A. Ambroze

Centre for Security, Communications and Network Research,
Plymouth University, Plymouth, UK
e-mail: info@cscan.org

Abstract

Error Correction Coding, a technique that is used extensively for efficient transmission of data bits in almost all fields of communication. A graphical representation of the code by constructing trellis diagrams has evolved as a valuable tool for reducing decoding complexities and improving the transmission efficiencies of the code. This project involves the study of Trellis diagrams, their construction, the encoding and decoding patterns and their implementation. The paper focuses on the trellis construction of single parity check codes and their coding/ decoding mechanisms using trellis diagrams. This implementation is carried out using C+ programming languages.

Keywords

Error Correction Codes, Single Parity Check Codes, Trellis Diagrams.

1 Introduction

Communication can be regarded as a simple process of transmission of information bits. The main aim of a communication channel is to transmit the information from the source to the sink over a channel which can be in the form of a physical cable, a wireless link or a storage device. The background of communication thus revolves around the concepts of transmission and reception. (Bhatacharya. A, 2006)

Any communication process involves the following three basic steps:

- Coding a message at its source.
- Transmitting the message through a communication channel.
- Decoding the message at its destination.

Error Correction Codes:

The communication systems strive to achieve error free transmission and reception of messages without errors by focussing on signal processing techniques like error correction codes. Error correction coding is a mechanism in which the errors introduced in the digital data during the transmission process are detected and corrected upon the reception of the data. It can be regarded as a signal processing technique which is used to improve the reliability and efficiency of communication on digital channels. The detection and correction of errors is carried out by the process of adding redundant bits into the string of messages that needs to be delivered. This technique of adding extra bits to the digital messages is termed as redundancy. Redundant bits have the ability to make each of the messages unique and maintain their unique structure even if some of the bits in the message are infected by various sources (Clark. G, 1981).These codes follow the concepts of Shannon's theorems that states that efficient transmission of bits can be attained if the transmission takes place as quick as possible with no or few errors. This fastness of the channel is guarded by channel capacity or the Shannon's limit which can be defined as the rate at which information is transmitted. This rate is nothing but the ultimate speed limit set for any communication system and is given by the notion: 'C'. (Ambroze. M. A, 2011)

Equation: **C = B log2 (1+SNR)**
Where: B = Bandwidth of the communication channel &
SNR = Signal to noise ratio.

Thus, error correcting codes can be of several variations like the hamming code, parity check code, goolay code, perfect code etc. In this paper we aim at the specific TRELLIS construction of single parity check codes

2 Single Parity Check Codes

Single Parity check codes form a part of the linear block codes for which each codeword consists of (N-1) information bits and a single parity check bit. These codes have the ability to detect any error patterns containing an odd number of errors. They are characterized as (N, N-1) codes, having $2^{(N-1)}$codewords and are used in various error detection operations. For a (3, 2) SPC code, the codewords can be written as: {0 0 0, 0 1 1, 1 01, 1 1 0}. The format of this codeword can be represented as | m1 m2 p|. Where: m1, m2 are the message bits and p is the parity check bit. The parity bits are inserted in the final column such that there exists an even number of 1's when one codeword in the row is considered. The generator matrix for this message block is formulated by selecting the identity matrix from the codewords.

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

3 Trellises and Viterbi Algorithm

Viterbi algorithm can be marked as a maximum likelihood decision device with efficient computational technique that can determine the probable path taken by the code to and output the correct transmitted codeword. It works on the principle of add- compare- select (ACS) to process a code trellis and to eliminate the less probable paths for further consideration. (Viterbi. A, 2006). A trellis code is a graphical representation of a code, conventional or block, in which every path represents a codeword. It can be defined as a directed graph which consists of levels of nodes (or vertices) and directed branches (or edges) that connect two consecutive nodes together. (Khchischang. F, 1995)

3.1 Construction of Trellises from the obtained Generator matrix:

- Each row in the generator matrix corresponds to the information bits of the code.
- The active span of each row is marked. The active span extends from the first 1 in the row to the last 1 in the row.
- The active spans in each row of the G matrix contribute to the number of states in each section of the trellis and the number of sections in the trellis is denoted by each column of the G matrix
- The information bits are matrix multiplied with the generator matrix.
- By considering the product and the active spans in each column of the G matrix, we determine the branch weights and the states respectively and thus construct the trellis diagram.

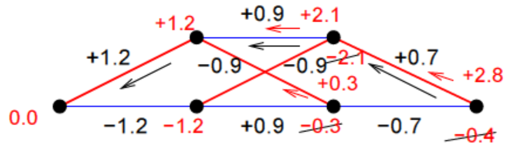
4 Illustration of the Viterbi decoding (soft decision) on (3, 2) SPC code

We consider the trellis diagram obtained for the (3, 2) SPC code and assign the codeword values to the three sections of the trellis.

i. In the first section, as a 1 is encoded by the vertical line, the received value remains the same. The received value undergoes a sign change on the horizontal line encoding a zero. Similarly, the weighting for each transitions are calculated in sections two and three of the trellis.
ii. The start node is always characterized by a node value of 0. The node values are calculated by adding the previous node value with the transition weight.
iii. The value of +1.2 in the top node of 1st section is obtained by: 0.0 + (+1.2)= +1.2.
   The value of -1.2 in the bottom node of 1st section is obtained by: 0.0 + (-1.2) = -1.2.

Similar calculations are made for each of the nodes. For the node having two values, the largest value is considered and the smallest value is eliminated. The route through which the largest value is obtained is marked by an arrow mark as shown.

iv. Now if we trace back, we get:



The final decoded codeword by utilizing trace back technique is 101.

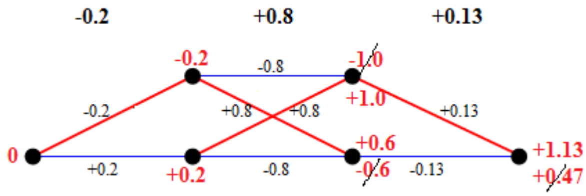
5 Results

SPC codes from (3, 2) to (6, 5) is considered and the results are produced for soft decision decoding. The addition of noise components on the received codeword is randomly added.

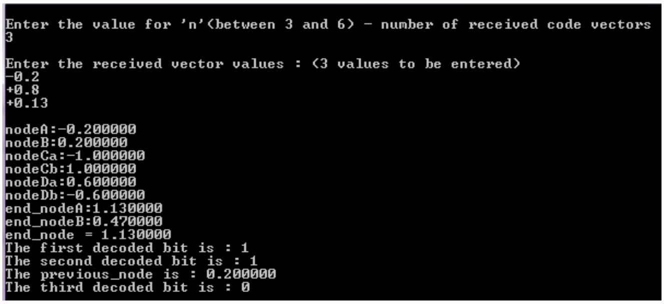
Transmitted codeword	Received codeword	Decoded codeword
0 0 0	-0.3 -0.5 -0.4	0 0 0
0 1 1	-0.2 +0.8 +0.13	0 1 1

Table 1: The decoded codewords of a (3, 2) SPC code:

Illustration 1: The received codeword is (-0.2, +0.8, +0.13)



The decoded codeword is [0 1 1].

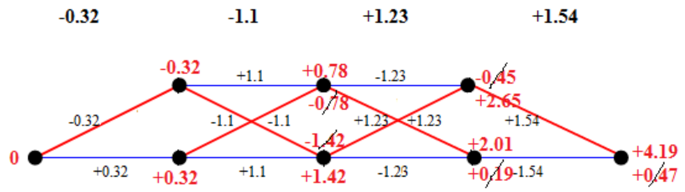


5.1 The decoded codewords of (4, 3) SPC code:

Transmitted codeword	Received codeword	Decoded codeword
0 0 0 0	-0.9 -0.34 -0.76 -0.1	0 0 0 0
0 0 1 1	-0.32 -1.1 +1.23 +1.54	0 0 1 1
0 1 0 1	-0.9 +0.5 -0.1 +0.67	0 1 0 1
0 1 1 0	-0.23 +0.9 +1.0 -0.12	0 1 1 0
1 0 0 1	+0.76 -0.2 -1.3 +1.3	1 0 0 1

Illustration 2:

The received codeword is (-0.32, -1.1, +1.23, +1.54)



The decoded codeword is [0 0 1]

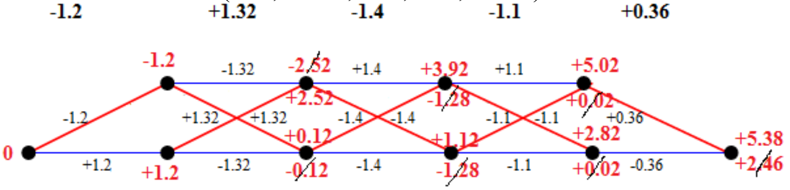
```
Enter the value for 'n'(between 3 and 6) - number of received code vectors
4

Enter the received vector values : (4 values to be entered)
-0.32
-1.1
+1.22
+1.3

nodeA:-0.320000
nodeB:0.320000
nodeCa:0.780000
nodeCb:-0.780000
nodeDa:-1.420000
nodeDb:1.420000
nodeC:0.780000
nodeD:1.420000
nodeEa:-0.440000
nodeEb:2.640000
nodeFa:2.000000
nodeFb:0.200000
nodeE:2.640000
nodeF:2.000000
end_nodeA:3.940000
end_nodeB:0.700000
end_node = 3.940000
The first decoded bit is : 1
The second decoded bit is : 1
The third decoded bit is : 0
The previous_node is : 0.320000
The fourth decoded bit is : 0_
```

5.2 The decoded codewords of (5, 4) SPC code:

Transmitted codeword	Received codeword	Decoded codeword
0 0 0 0 0	-0.12 -0.7 -1.1 -0.98 -1.2	0 0 0 0 0
0 0 0 1 1	-0.23 -0.76 -0.4 +0.7 +0.2	0 0 0 1 1
0 0 1 0 1	-0.65 -0.9 +0.9 -0.123 +1.3	0 0 1 0 1

Illustration 3:

The received codeword is (-1.2, +1.32, -1.4, -1.1, +0.36)



The decoded codeword is [0 1 0 0 1]

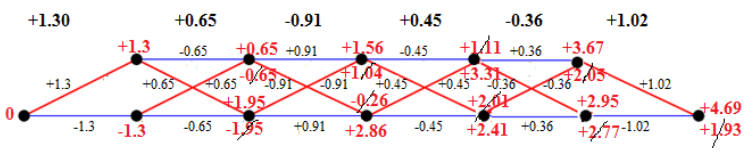
```
Enter the value for 'n'(between 3 and 6) - number of received code vectors
5

Enter the received vector values : (5 values to be entered)
-1.2
+1.32
-1.4
-1.1
+0.36

nodeA:-1.200000
nodeB:1.200000
nodeCa:-2.520000
nodeCb:2.520000
nodeDa:0.120000
nodeDb:-0.120000
nodeC:2.520000
nodeD:0.120000
nodeEa:3.920000
nodeEb:-1.280000
nodeFa:1.120000
nodeFb:1.520000
nodeE:3.920000
nodeF:1.520000
nodeGa:5.020000
nodeGb:0.420000
nodeHa:2.820000
nodeHb:2.620000
nodeG:5.020000
nodeH:2.820000
end_nodeA:5.380000
end_nodeB:2.460000
end_node = 5.380000
The first decoded bit is : 1
The second decoded bit is : 0
The third decoded bit is : 0
The Fourth decoded bit is : 1
The previous_node is : 1.200000
The fifth decoded bit is : 0
```

5.3 The decoded codewords of (6, 5) SPC code:

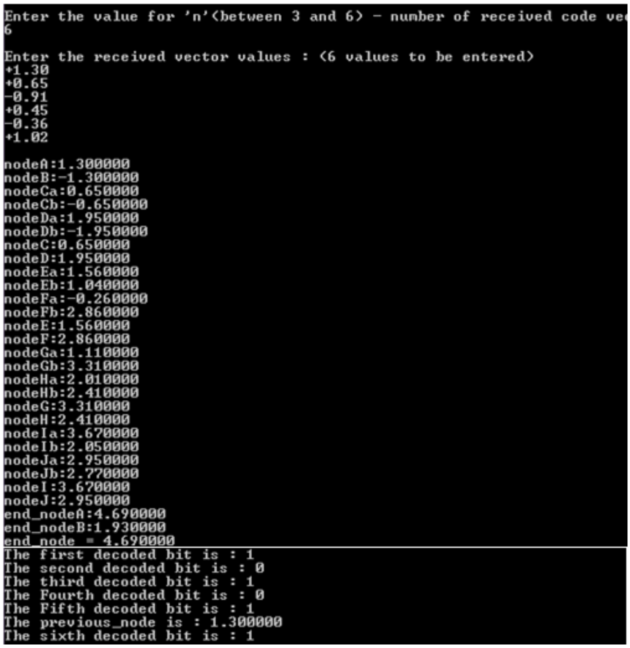
Transmitted Codeword	Received Codeword	Decoded Codeword
0 0 0 0 0 0	-0.1 -0.2 -0.1 -0.3 -0.2 -0.6	0 0 0 0 0 0
0 0 0 0 1 1	-1.2 -0.91 -0.13 -0.72 +0.12 +1.9	0 0 0 0 1 1
0 0 0 1 0 1	-0.23 -0.98 -0.43 +1.2 -1.8 +1.67	0 0 0 1 0 1
0 0 0 1 1 0	-0.12 -0.76 -1.2 +0.91 +1.1 -0.53	0 0 0 1 1 0
0 0 1 0 0 1	-0.43 -1.43 +0.63 -0.66 -0.75 +1.54	0 0 1 0 0 1

Illustration 4:

The received codeword is (+1.30 +0.65 -0.91 +0.45 -0.36 +1.02)



The decoded codeword is [1 1 0 1 0 1]



5.5. When the received codeword undergoes an error:

Received codeword with error/s for (3, 2) SPC Code:

Transmitted codeword	Received codeword	Decoded codeword
0 0 0	-0.3 -0.61 0.2 +0.3 +0.61 +0.2 +0.3 -0.61 -0.2	0 0 0 0 1 1 1 0 1
0 1 0 1	+1.32 +0.3 -0.21 +1.1 -1.32 -0.3 -0.21 +1.1 -1.32 +0.3 +0.21 +1.1	1 1 1 1 1 1 0 0 1 0 1 0
0 0 1 0 1	-0.65 -0.9 +0.9 +0.123 +1.3 -0.65 -0.9 +0.9 +0.123 -1.3 +0.65 +0.9 -0.9 -0.123 +1.3	1 1 0 1 1 0 1 1 0 0 1 0 1 1 1
0 0 0 0 1 1	-1.2 -0.91 -0.13 -0.72 +0.12 -1.9 +1.2 +0.91 -0.13 -0.72 +0.12 +1.9 +1.2 -0.91 +0.13 -0.72 -0.12 +1.9	0 0 0 0 0 0 1 1 0 0 1 1 1 1 1 0 0 1

6 Conclusion

This paper on Trellis structure on error correction codes involves a detailed working of the Single Parity Check codes, Viterbi Algorithm and the trellis construction of SPC codes. The soft decision coding is illustrated with various input parameters and the trellis is constructed for the same. Minimal trellis for (3, 2), (4, 3), (5, 4) and (6, 5) single parity check codes are constructed and Viterbi decoding algorithm is employed to decode the codes to obtain an output from the decoder. The software implementation for soft decision coding is carried out using C+ programming language, in which a decoded codeword is obtained on feeding the input bit values.

7 Limitations and Future work

The software implementation for encoding and the decoding is done for (N, N-1) Single Parity Check Codes where N = 3, 4, 5, 6. Viterbi decoding algorithm is implemented for the decoding purposes. Future scope is to upgrade the program to have a more concise program that would take care of (N, N-1) SPC codes.

8 References

Ambroze. M. A, 2010, *Digital and Wireless Communications notes*, School of Computing, and Communications Engineering (SoCCE), University of Plymouth. Available online: https://tulip.plymouth.ac.uk/Module/ELEC508/LectureNotes/notes.pdf Accessed on: 3.6.2011.

Bhatacharya. A (2006). *Digital Communication*. India: Tata Mc-Graw Hill. P3-6.

Clark. G, 1981, *Error Correction Coding for Digital Communication.* United States of America: Plenum Press. P7-15.

Khchischang. F, 1995, *On the Trellis Structure of Block Codes*, IEEE Transactions on Information Theory, Vol: 41. Available online: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=476317. Accessed on: 4.7.2011.

Viterbi. A, 2006, *A Personal History of the Viterbi Algorithm*, IEEE Signal Processing Magazine, July. Available online: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1657823. Accessed on: 18.10.2011.