CONSTRUCTION AND MINIMAL TRELLIS FOR (32, 16, 8) DECOMPOSABLE CODE

Gökmen Altay[1]
e-mail: galtay@bahcesehir.edu.tr

Osman Nuri Uçan[2]
e-mail: oucan@istanbul.edu.tr

[1]Bahçeşehir University, Faculty of Engineering, Department of Electrical & Electronics Engineering, 34538, Bahçeşehir, Istanbul, Turkey.

[2]Istanbul University, Faculty of Engineering, Department of Electrical & Electronics Engineering, 34850, Avcılar, Istanbul, Turkey.

Key words: Decomposable codes, minimal trellis.

ABSTRACT

Decomposable codes are important for channel coding as they have simpler trellis structure and therefore allow using maximum likelihood soft decision Viterbi algorithm for decoding a code with reduced complexity. In this study, (32, 16, 8) optimal decomposable code is constructed and also its minimal trellis presented.

I. INTRODUCTION

Block codes can be employed with difficulty in trellis decoders since they have complex trellis structures. Decomposable codes can be constructed to overcome this difficulty [1]. A decomposable code can be decomposed into component codes of smaller dimensions or shorter lengths, which leads to simplified sub-trellises with less state complexity in it [2]. Hence, a decomposable code can be constructed by combining two or more component codes with simple trellis structure. Some of the decomposable codes are product codes with their good minimal trellis structure [3], [4], [5], and codes obtained by squaring, |a + x|b + x|a + b + x| , and *Turyn* construction [1],[6]. The trellis diagram of a block code, known as a coset trellis, can be constructed as a set of parallel sub-trellises with similar structure [1], [7]. Minimal trellis representation promises a good trade-off between the number of states in the trellis and the complexity of it [8]. In this paper, (32, 16, 8) optimal code constructed based on product code construction approach but with a little modification on it. Then, minimal trellis structure for this construction is presented to show that maximum likelihood decoding of the constructed code using Viterbi algorithm can be performed with reduced complexity.

II. CODE CONSTRUCTION

A decomposable general product code, C, can be constructed by combining two component codes $C_1 = (n_1, k_1, d_1)$ and $C_2 = (n_2, k_2, d_2)$, whose generator matrices are G_1 and G_2, respectively. The generator matrix, G, of C is formed by the *Kronecker product* of generator matrices G_1 and G_2, as $G = G_2 \otimes G_1$. When G_2 is a single parity check matrix as,

$$G_2 = \begin{bmatrix} 1 & & & 1 \\ & 1 & & 1 \\ & & \ddots & \vdots \\ & & 1 & 1 \end{bmatrix} \qquad (1).$$

then G is obtained as a result of the operation $G = G_2 \otimes G_1$ as,

$$G = \begin{bmatrix} G_1 & & & G_1 \\ & G_1 & & G_1 \\ & & \ddots & \vdots \\ & & G_1 & G_1 \end{bmatrix} \qquad (2).$$

Utilising this generic form of product code construction we could only construct a (32, 7, 8) code, which is not an optimal code. In order to obtain an optimal code we augment two other component generator matrices G_3 and G_4 onto G in a way that was proposed in [9] but with a slight modification of it. To construct the final optimal code C′ = (32, 16, 8) we employ the following component codes: $C_1 = (4, 1, 4)$ with generator matrix $G_1 = [\,1\ 1\ 1\ 1\,]$, $C_2 = (8, 7, d_2)$ with generator matrix as in (1). $C_3 = (4,2,2)$ and $C_4 = (4, 1, 1)$ with generator matrices G_3 and G_4 as below,

$$G_3 = \begin{bmatrix} 1\ 1\ 0\ 0 \\ 0\ 1\ 1\ 0 \end{bmatrix} \quad (3), \qquad G_4 = [\,1\ 0\ 0\ 0\,] \quad (4).$$

All the specified generator matrices is composed as in (5),

$$G' = \begin{bmatrix} & & G_{(7\,x\,32)} & & & & & \\ G_3 & G_3 & G_3 & G_3 & & & & \\ & & G_3 & G_3 & G_3 & G_3 & & \\ & & & & G_3 & G_3 & G_3 & G_3 \\ G_3 & & G_3 & & G_3 & & G_3 & \\ G_4 & G_4 & G_4 & G_4 & G_4 & G_4 & G_4 & G_4 \end{bmatrix} \qquad (5).$$

When we write (5) explicitly we obtain the eventual construction of G' as in (6),

$$
G' = \begin{pmatrix}
\mathbf{1111}000000000000000000000000\mathbf{1111} \\
0000\mathbf{1111}00000000000000000000\mathbf{1111} \\
00000000\mathbf{1111}0000000000000000\mathbf{1111} \\
000000000000\mathbf{1111}000000000000\mathbf{1111} \\
0000000000000000\mathbf{1111}00000000\mathbf{1111} \\
00000000000000000000\mathbf{1111}0000\mathbf{1111} \\
000000000000000000000000\mathbf{11111111} \\
\mathbf{1100110011001100}0000000000000000 \\
\mathbf{0110011001100110}0000000000000000 \\
00000000\mathbf{1100110011001100}00000000 \\
00000000\mathbf{0110011001100110}00000000 \\
0000000000000000\mathbf{1100110011001100} \\
0000000000000000\mathbf{0110011001100110} \\
\mathbf{1100}0000\mathbf{1100}0000\mathbf{1100}0000\mathbf{1100}0000 \\
\mathbf{0110}0000\mathbf{0110}0000\mathbf{0110}0000\mathbf{0110}0000 \\
\mathbf{10001000100010001000100010001000}
\end{pmatrix} \quad (6).
$$

It is important to emphasize that we have augmented nine other row in (6) onto the base generator matrix G. This $C' = (n', k', d') = (32, 16, 8)$ code is not systematic since it does not hold the information bits in the first 16 positions, whereas it is considered to be quasi-systematic as the information bits can be obtained by elementary column operations of G'. Using the same component codes and with similar construction approach we also obtained (16,5,8) and (24,9,8) codes, whereas we focused on (32,16,8) in this study as we consider it better than others.

III. MINIMAL TRELLIS FOR (32, 16, 8) CODE

Trellis structure is a good way of describing a block code and vital to perform maximum likelihood decoding of codewords utilising Viterbi algorithm. For practical considerations it not easy for trellis oriented encoders and decoders to perform encoding and decoding of a block code, which has complex trellis structure. Therefore, reducing the number of trellis complexity parameters such as states and branches is main concern for designing trellis of a code. For Viterbi decoding, the total number of trellis branches per unit time is usually regarded as a more accurate measure of decoding complexity than the size of the state space. Thus the branch complexity profile may be of more practical importance than the state complexity profile [3]. A trellis structure that reduces the number of states and branches simultaneously is regarded to be minimal trellis structure.

The minimal trellis structure of single-parity-check product codes of (2) is shown in Fig. 1., where $b \in C_1$ is an n_1-tuple and 0 is an all-zero n_1-tuple [6], [7], [8].

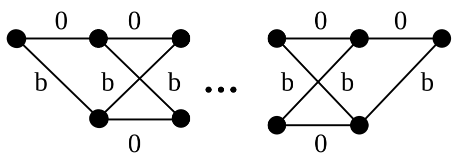

Fig. 1. Generalized minimal trellis of product codes.

By taking the minimal trellis structure of Fig. 1. as a base, we formed the minimal trellis structure, T, for the constructed (32, 16, 8) code as shown in Fig. 2.

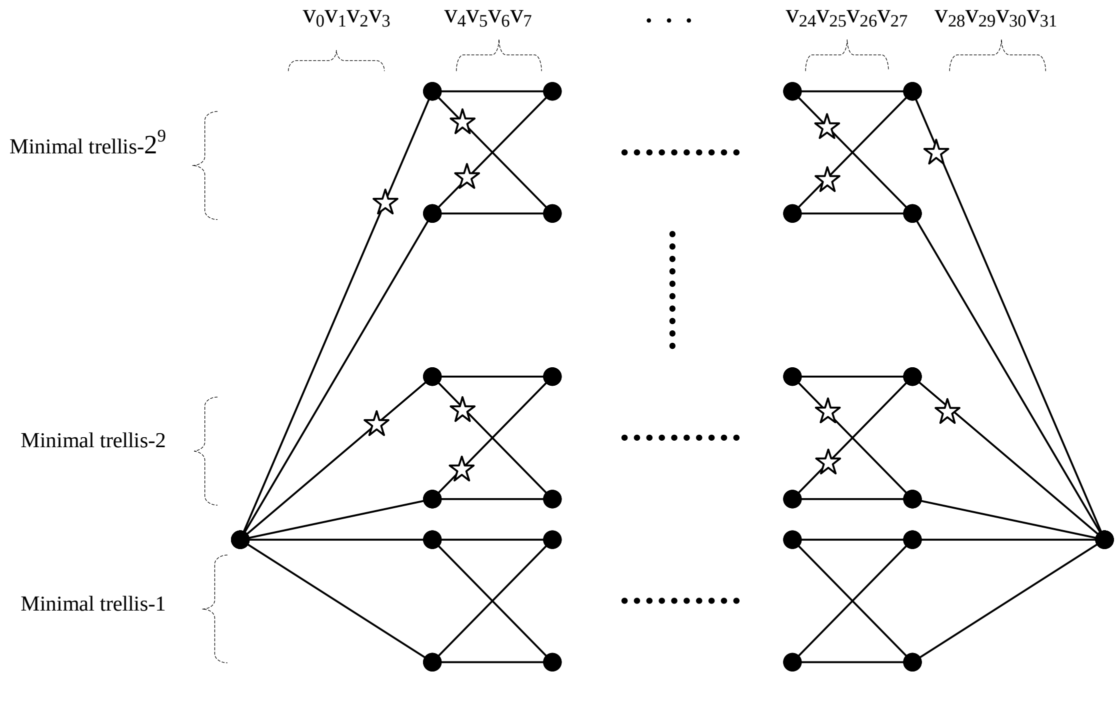

Fig. 2. Minimal trellis structure, T, for (32, 16, 8) code

As it can be seen from Fig. 2., the T has 2^9 parallel and structurally identical minimal trellises of Fig. 1. where each branch is 4-tuple. Lets call the rows of G_3 and G_4 as augmenting rows, h_i, i=1,2,…,9. Minimal trellis-1 of T is the same as in Fig. 1. but here b is all-one 4-tuple, b=[1111], and 0 is all-zero 4-tuple, 0=[0000], which represents the G in G′. The rest of the minimal trellises of T is composed of linear combinations of h_i by the rows of G. In other words, minimal trellis-2 is due to h_1 , the 8^{th} row of G′, minimal trellis-3 is due to h_2 and so on. There are two types of major branches considered while assigning 4-tuple code bits in the minimal trellises of T, which may be called cross-branch and direct-branch. Minimal trellis-1 of T is explicitly shown in Fig. 1. where b is assigned to cross-branches and 0 is assigned to direct-branches. Similarly, in the rest of minimal trellises of T, branches can be discriminated such that, where a branch that has a star on it is a cross-branch and otherwise is a direct-branch. Now, we can define the way of assigning 4-tuples to a branch in a minimal trellis:

a) An augmenting row h_i or linear combination of h_is, is separated to four where each part is 4-tuple that corresponds $v_j v_{j+1} v_{j+2} v_{j+3}$ where j = 0,1,...,28. If first 4-tuple, a, is not all-zero, then it is assigned to first cross-branch of a minimal trellis of T, which means $v_0v_1v_2v_3$ has been assigned at this point. Also first direct-branch is assigned with a′, where $a' = a \oplus [1111]$. On the other hand, If first 4-tuple, a, was all-zero, then, instead of a we would assign all-one 4-tuple, [1111], to first cross-branch and all-zero 4-tuple, [0000], to first direct-branch.
b) After first branches is complete we can continue to place 4-tuples from left to right, branch by branch, along the minimal trellis of T. We can now process second part of the separated row. The same processes are performed but this time notice that there are two cross-branches and two direct-branches to be assigned. Therefore, while assigning, both of cross-branches are assigned with the same 4-tuple, that is also the case for direct-branches. This process is performed until all the branches of the minimal trellis are assigned. Afterwards, we begin to assign the other minimal trellises of T by changing the row of h_i or linear combination of h_is until all the minimal trellises of T is complete.

After assigning codewords over T, one should consider the place of information bits, $u = (u_0u_1u_2u_3 \ldots u_{12}u_{13}u_{14}u_{15})$ among a codeword, $v = (v_0v_1v_2v_3 \ldots v_{28}v_{29}v_{30}v_{31})$, so that Viterbi decoding can be performed. We specify the information bits with their corresponding bits as follows: $u_0 = v_3$, $u_1 = v_7$, $u_2 = v_{11}$, $u_3 = v_{15}$, $u_4 = v_{19}$, $u_5 = v_{23}$, $u_6 = v_{27}$, $u_7 = v_5 + v_6$, $u_8 = v_6 + v_7$, $u_9 = v_{13} + v_{14} + v_5 + v_6$, $u_{10} = v_{14} + v_{15} + v_6 + v_7$, $u_{11} = v_{29} + v_{30}$, $u_{12} = v_{30} + v_{31}$, $u_{13} = v_1 + v_2 + v_5 + v_6$, $u_{14} = v_2 + v_3 + v_6 + v_7$, $u_{15} = v_{28} + v_{29} + v_{30} + v_{31}$.

IV. CONCLUSION

In this paper, we presented the construction of decomposable (32, 16, 8) optimal code and also designed a minimal trellis for the code in order to enable Viterbi decoding of the code with reduced complexity. Although, this study focused on a unique case, we consider it as a first step to attain a generic construction of decomposable distance-8 optimal codes.

REFERENCES

1. G.D. Forney, Jr., Coset codes II: Binary lattices and related codes IEEE Trans. Inform. Theory, vol. IT-34, 1152-1187, 1988.
2. Marc P.C. Fossorier, Shu Lin, Some Decomposable Codes: The |a + x|b + x|a + b + x| Construction, IEEE Trans. Inform. Theory**,** vol. 43, 1663-1667, 1997.
3. G.D. Forney, Jr., Dimension/length profiles and trellis complexity of linear block codes, IEEE Trans. Inform. Theory, vol. 40, 1741-1752, 1994.
4. J. K. Wolf, Efficient maximum likelihood decoding of linear block codes using a trellis, IEEE Trans. Inform. Theory, vol. 24, 76-80, 1978.
5. R.J. McEliece, On the BCJR trellis for linear block codes, IEEE Trans. Inform. Theory, vol. 42, 1072-1092, 1996.
6. F.J. Mac Williams and N.J. A. Sloane, The Theory of Error-Correcting Codes, Amsterdam: North-Holland Mathematical Library, 1977.
7. G.D. Forney, Jr., M.D. Trott, The dynamics of group codes: State spaces, trellis diagram, and canonical encoders, IEEE Trans. Inform. Theory, vol. 39, 1491-1513, 1993.
8. D.J. Muder, Minimal trellises for block codes, IEEE Trans. Inform. Theory, vol. 34, 1049-1053, 1988.
9. Gökmen ALTAY, Investigation of Product Codes & Enhancing Augmented Product Codes, M.Sc. Final Project, South Bank University, London, November 2001.