Minimal Trellises for Linear Block Codes and Their Duals

A. B. Kiely, S. Dolinar, and L. Ekroot
Communications Systems Research Section

R. J. McEliece
California Institute of Technology
and
Communications Systems Research Section

W. Lin
California Institute of Technology

We consider the problem of finding a trellis for a linear block code that minimizes one or more measures of trellis complexity for a fixed permutation of the code. We examine constraints on trellises, including relationships between the minimal trellis of a code and that of the dual code. We identify the primitive structures that can appear in a minimal trellis and relate this to those for the minimal trellis of the dual code.

I. Introduction

Every linear block code can be represented by a minimal trellis, originally introduced by Bahl et al. [1], which is a labeled graph that can be used as a template for encoding or decoding. As shown by McEliece,[1] the minimal trellis simultaneously minimizes the maximum number of states, the total numbers of vertices and edges in the trellis, and the total numbers of additions and path comparisons required for decoding with the Viterbi algorithm.

In this article, we examine properties of the minimal trellis representation of a code and its dual for a fixed permutation. A companion article [2] uses these results to examine the problem of finding a permutation that minimizes one or more trellis complexity measures.

Section II reviews the subject of minimal trellises for a fixed permutation of a code. We examine the building blocks of such trellises and identify several different measures of trellis size or complexity. In Section III, we illustrate the connection between the minimal trellis of a code and that of the dual code. The section includes results that describe the structure and complexity of trellises for self-dual and other special codes.

[1] R. J. McEliece, "On The BCJR Trellis for Linear Block Codes," submitted to *IEEE Trans. Inform. Theory.*

It can be shown in [7] that for $0 \leq x < 1$,

$$\sum_{k=0}^{\infty} \binom{k+L-4}{k} x^k = \left(\frac{1}{1-x}\right)^{L-3} \tag{B-6}$$

Using Eq. (B-6) in Eqs. (B-4) and (B-5), we then obtain, for $0 \leq x < 1$,

$$1 \leq f_L(x) \leq \frac{L-2}{L-3} \tag{B-7}$$

The upper and lower bounds given in Eq. (B-7) are both asymptotically tight in the limit as $L \to \infty$. So we can conclude that for $0 \leq x < 1$, $f_L(x) \to 1$ as $L \to \infty$, where the convergence is uniform in x.

II. Minimal Trellis Representation of a Code

A. The Minimal Span Generator Matrix

For any linear (n, k) block code $\mathcal{C}$ over $GF(q)$ there exists a minimal span generator matrix (MSGM) representing $\mathcal{C}$. A minimal trellis $\mathcal{T}$ for the code can be constructed from the MSGM. The trellis has $n + 1$ levels of vertices and n levels of edges. The vertex levels, called depths, are numbered from 0 to n; the edge levels, called stages, are numbered from 1 to n. Each stage of edges corresponds to one stage of encoding or decoding using the trellis. Each vertex at depth i represents a possible encoder state after the ith stage of encoding. The ith stage corresponds to the ith column of the generator matrix, whereas the ith depth corresponds to the "space between" columns i and $i + 1$.

The edge span of any row of the generator matrix is the smallest set of consecutive integers (stages) containing its nonzero positions. The vertex span of the row is the set of depths i such that at least one nonzero symbol occurs before and after depth i. Using the generator matrix to encode k information symbols in n stages of encoding, the edge span of the jth row represents the interval of stages during which the jth information symbol can affect the encoder output. The vertex span of the jth row is the set of depths at which the jth information symbol can affect the encoder state.

For example, the (6,3) shortened Hamming code has the minimal span generator matrix

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

The edge spans are $\{1, 2, 3\}$, $\{2, 3, 4, 5, 6\}$, and $\{3, 4, 5\}$. The vertex spans are $\{1, 2\}$, $\{2, 3, 4, 5\}$, and $\{3, 4\}$. We use the term span length to refer to the cardinality of a span.

A remarkable result is that the MSGM simultaneously makes all of the spans as short as possible: The edge spans (vertex spans) for any other generator matrix representing $\mathcal{C}$ always contain the corresponding spans of some row-permuted MSGM.[2] Any generator matrix can be put into minimal span form using the following greedy algorithm: At each step, perform any row operation that reduces the edge span of any row of the matrix. The rows of the MSGM are then "atomic codewords," according to the terminology of Kschischang and Sorokine [5].

Each vertex or state at a given depth can be uniquely labeled using k or fewer symbols from $GF(q)$. But any given state-label symbol can be reused to represent several information symbols, as long as the vertex spans of the corresponding rows of the generator matrix do not overlap. This reassignment of state-label symbols to multiple rows of the generator matrix is the key to efficient trellis representations of the code.

For example, the minimal trellis $\mathcal{T}$ produced for the (6,3) shortened Hamming code with MSGM given in Eq. (1) is shown in Fig. 1. For this trellis, we can define the binary state label to be $s_2 s_1$, where $s_2 = 1$ at depth i if the second information bit is 1 and i is within the vertex span of the second row, and $s_1 = 1$ if either (1) the first information bit is 1 and i is within the vertex span of the first row or (2) the third information bit is 1 and i is within the vertex span of the third row. This time-sharing arrangement for state bit s_1 is possible because the vertex spans of the first and third rows do not overlap.

In the sequel, we will be interested primarily in nondegenerate codes, which we define as codes whose minimum distance d and dual code minimum distance $d^\perp$ are both at least 2. Degenerate codes have a simple interpretation: If $d < 2$, the vertex span of some row of the MSGM must be empty; if $d^\perp < 2$,

[2] Ibid.

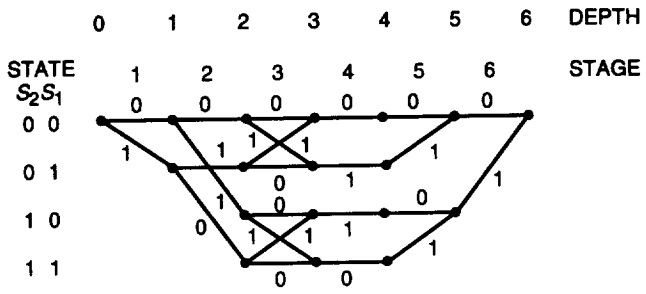

Fig. 1. A minimal trellis for the (6,3) shortened Hamming code.

some column of the generator matrix must be identically zero. For a degenerate code, we can simply ignore the extraneous symbol positions (if $d^{\perp} < 2$) and/or separately decode the unprotected information symbols (if $d < 2$). The code consisting of the remaining code symbols is then nondegenerate.

B. Past and Future Subcodes

Following Forney [3], let us define the ith past and future subcodes, denoted $\mathcal{P}_i$ and $\mathcal{F}_i$, to be the sets of all codewords whose vertex spans are contained in $[0, i-1]$ and $[i+1, n]$, respectively. The dimensions of these codes can be easily determined from the MSGM: $f_i \triangleq \dim(\mathcal{F}_i)$ is the number of rows for which the leftmost nonzero entry lies in column $i+1$ or later, and $p_i \triangleq \dim(\mathcal{P}_i)$ is the number of rows for which the rightmost nonzero entry lies in column i or earlier.[3] This implies that p_i and f_i are monotonic,

$$0 = p_0 \leq p_1 \leq \cdots \leq p_n = k = f_0 \geq f_1 \geq \cdots \geq f_n = 0 \tag{2}$$

and never change by more than 1 from one index to the next.

For each $1 \leq i \leq n$, we define the left- and right-basis indicators, $l_i, r_i \in \{0, 1\}$, to identify the positions where the future and past dimensions change:

$$l_i \triangleq f_{i-1} - f_i \qquad r_i \triangleq p_i - p_{i-1}$$

For any i, $l_i = 1$ if and only if the edge span of some row of the MSGM G *begins* in column i, or equivalently, the ith column of G is linearly independent of the $i-1$ columns to the *left*. Similarly, $r_i = 1$ if and only if the edge span of some row of G *ends* in column i, i.e., the ith column of G is linearly independent of the $n-i$ columns to the *right*. The columns where $l_i = 1$ and the columns where $r_i = 1$ each forms a basis for the column space of G, and these sets are called the left basis and the right basis, respectively. The positions of the left and right basis columns can be regarded as information positions when the generator matrix is used to encode the information left to right or right to left, respectively.

C. Primitive Structures of a Minimal Trellis

There are four basic building blocks that can be used to construct the minimal trellis for any nondegenerate code. At any given stage i, all primitive structures are of the same type, which is determined by the values of l_i and r_i. The primitive structures are

[3] Ibid.

(1) Simple extension ($-$): This primitive structure appears at stage i when $l_i = 0, r_i = 0$, e.g., stage 4 in Fig. 1. Simple extensions at stage i imply a single edge out of each vertex at depth $i-1$ and a single edge into each vertex at depth i; hence, the number of vertices remains constant.

(2) Simple expansion ($<$): This corresponds to $l_i = 1, r_i = 0$, e.g., stages 1 and 2 in Fig. 1. There are q edges out of each vertex at depth $i-1$, and a single edge into each vertex at depth i, hence, multiplying by q the number of states from one vertex depth to the next.

(3) Simple merger ($>$): This corresponds to $l_i = 0, r_i = 1$, e.g., stages 5 and 6 in Fig. 1. A simple merger is a time-reversed simple expansion, reducing the number of states by a factor of q.

(4) Butterfly ($\bowtie$): This corresponds to $l_i = 1, r_i = 1$, e.g., stage 3 in Fig. 1. There are q edges out of each vertex at depth $i-1$ and q edges into each vertex at depth i; hence, the number of states is constant.

The total numbers of such primitive structures in the trellis are denoted by N_-, $N_<$, $N_>$, and $N_{\bowtie}$, respectively. For example, the trellis in Fig. 1 has $N_< = 3 = N_>$, $N_{\bowtie} = 2$, $N_- = 4$. Because the graph has exactly one initial node and one terminal node, the total number of simple expansions must equal the total number of simple mergers:

$$N_< = N_>$$

The total number of edges in the trellis, E, can be found by counting the number of edges associated with each primitive trellis structure:

$$E = N_- + qN_< + qN_> + q^2 N_{\bowtie} \tag{3}$$

Similarly, the total number of mergers M is the sum of the number of simple mergers and the mergers included in butterflies:

$$M = N_> + qN_{\bowtie} \tag{4}$$

If we count the total number of vertices associated with each primitive structure, then each vertex in the trellis (excluding initial and terminal nodes) will be counted twice, so the total number of vertices V satisfies

$$2V - 2 = 2N_- + (q+1)N_< + (q+1)N_> + 2qN_{\bowtie}$$

which gives

$$V = 1 + N_- + (q+1)N_< + qN_{\bowtie} \tag{5}$$

Combining Eqs. (3), (4), and (5), we find

$$M = \frac{E - V + 1}{q - 1}$$

This is the generalization of the binary version of this result found by McEliece.[4]

[4] Ibid.

D. Measures of Trellis Complexity for Viterbi Decoding

The vertex space dimension at depth i is

$$v_i = k - f_i - p_i, \qquad i = 0, \cdots, n \tag{6}$$

and the edge space dimension at stage i is

$$e_i = k - f_i - p_{i-1}, \qquad i = 1, \cdots, n \tag{7}$$

The total number of vertices at depth i is q^{v_i}, and the total number of edges at stage i is q^{e_i}. Of course, $v_i \geq 0$ for all i since at least one vertex must exist at each depth. Also, for nondegenerate codes, $e_i \geq 1$ for all i, i.e., no stage consists of a single edge.

The most commonly used measure of Viterbi decoding complexity for a minimal trellis is the maximum dimension of its state space,

$$s_{\max} \triangleq \max_i v_i \tag{8}$$

This complexity metric has been cited as one of the essential characteristics of any code [6]. Similarly, the maximum dimension of the edge space is

$$e_{\max} \triangleq \max_i e_i \tag{9}$$

Forney argues that this is a more relevant complexity measure because, unlike $s_{\max}$, this quantity cannot be reduced by combining adjacent stages of a trellis [4].

A different metric, used in McEliece's derivation of the MSGM,[5] is the total length of all the edge spans of the rows of the MSGM:

$$\varepsilon \triangleq \sum_{j=1}^{k} \varepsilon_j \tag{10}$$

where ε_j denotes the length of the edge span of the jth row of the MSGM. A similar span length metric is the total length of all the vertex spans:

$$\nu \triangleq \sum_{j=1}^{k} \nu_j$$

where $\nu_j = \varepsilon_j - 1$ is the length of the vertex span of the jth row of the MSGM. These two metrics are equivalent to the sums of all the edge dimensions or vertex dimensions (summed over stages or depths, respectively):

[5] Ibid.

$$\nu + k = \varepsilon = \sum_{i=1}^{n} e_i = k + \sum_{i=0}^{n} v_i$$

McEliece argues that more meaningful measures of Viterbi decoding complexity are the total numbers of edges E, vertices V, and mergers M, rather than simply the vertex or edge dimensionality:[6]

$$E = \sum_{i=1}^{n} q^{e_i} \tag{11}$$

$$V = \sum_{i=0}^{n} q^{v_i} \tag{12}$$

$$M = \sum_{i=1}^{n} r_i q^{v_i} = \sum_{i=1}^{n} l_i q^{v_i - 1} = \frac{1}{q} \sum_{i=1}^{n} l_i q^{e_i} = \frac{1}{q} \sum_{i=1}^{n} r_i q^{e_i} \tag{13}$$

E is equal to the number of binary additions required to compute path metrics, and M is the number of q-ary comparisons required to merge trellis paths. The computational complexity of Viterbi decoding is proportional to E.[7]

III. Minimal Trellis Representation of the Dual Code

In this section, we explore the relationship between the minimal trellises for a code $\mathcal{C}$ and its dual $\mathcal{C}^{\perp}$.

A. Past and Future Subcode Relationships

As discussed in Section II.B, $l_i = 0$ if and only if the ith column of the MSGM can be written as some linear combination of the $i-1$ columns to its left. In other words, there exists a dual codeword y of the form

$$y = \underbrace{XXX\cdots X}_{i-1} 1 \underbrace{000\cdots 0}_{n-i}$$

Where $XXX\cdots X$ denotes some sequence of symbols from $GF(q)$. Defining $y_1, y_2, \cdots, y_{n-k}$ in this manner for each of the left-dependent columns in the MSGM produces $n-k$ dual codewords of the form

$$\begin{array}{lcll} y_1 & = & XXX\cdots X1000\cdots & 0 \\ y_2 & = & XXX\cdots & X1000\cdots 0 \\ \vdots & & & \\ y_{n-k} & = & XXX\cdots & X1 \end{array}$$

These dual codewords are clearly linearly independent and, thus, can be used as the rows of the generator matrix for $\mathcal{C}^{\perp}$. We see that the positions where $r_i^{\perp} = 1$ are precisely the positions where $l_i = 0$; the same argument applied to the right-dependent columns shows that the positions where $l_i^{\perp} = 1$ are precisely the

[6] Ibid.

[7] Ibid.

positions where $r_i = 0$. Here $l_i^\perp$ and $r_i^\perp$ are the left- and right-basis indicators for $\mathcal{C}^\perp$. These observations lead to the following theorem.

Theorem 1. For each $0 \leq i \leq n$, the left- and right-basis indicators for a code and its dual are related by

$$l_i + r_i^\perp = l_i^\perp + r_i = 1$$

and the dimensions p_i, f_i of the past and future subcodes of a code are given in terms of those of the dual code $p_i^\perp$, $f_i^\perp$ as follows:

$$p_i = k - n + i + f_i^\perp$$

$$f_i = k - i + p_i^\perp$$

We believe that this result, which relates minimal trellises of a code and dual for any *fixed* permutation, is more fundamental than similar dual relationships for permutations of codes. This result is also contained in [4], but is derived by first considering permutations of codes.

B. Primitive Trellis Structures for the Dual Code

Much information about the trellis for the dual code can be inferred from the trellis structure of the code. For example, if the code has a simple expansion at the ith stage, then $l_i = 1, r_i = 0$, which implies, using Theorem 1, that the dual code has $l_i^\perp = 1$, $r_i^\perp = 0$; hence, the trellis of the dual code also has a simple expansion structure at this stage. Repeating this procedure, we find the "dual" of each primitive structure, shown in Table 1.

Given an unlabeled trellis, Table 1 can be used to determine the number and type of primitive structures present at every depth of the trellis for the dual code. However, we cannot in general determine the interconnections without additional information about the code.

The dual relationship for primitive structures shown in Table 1 implies that

$$N_<^\perp = N_< = N_> = N_>^\perp$$

and

$$N_- = qN_\times^\perp$$

C. Dual-Code Complexity Measures

The following well-known result, first noted by Forney [3], is a consequence of Eq. (6) and Theorem 1.

Lemma. A code and its dual code have equivalent vertex spaces, namely, for each i,

$$v_i = v_i^\perp$$

Table 1. Dual primitive structures.

Code structure	Dual structure
Simple extension (−)	Butterfly (⧖)
$l_i = 0, r_i = 0$	$l_i^\perp = 1, r_i^\perp = 1$
Simple expansion (<)	Simple expansion (<)
$l_i = 1, r_i = 0$	$l_i^\perp = 1, r_i^\perp = 0$
Simple merger (>)	Simple merger (>)
$l_i = 0, r_i = 1$	$l_i^\perp = 0, r_i^\perp = 1$
Butterfly (⧖)	Simple extension (−)
$l_i = 1, r_i = 1$	$l_i^\perp = 0, r_i^\perp = 0$

Consequently, many of the trellis complexity measures for a code can be determined by evaluating the same measure on the dual code:

$$V = V^\perp$$

$$s_{\max} = s_{\max}^\perp$$

$$\varepsilon - k = \nu = \nu^\perp = \varepsilon^\perp - (n - k)$$

Note that this implies $\varepsilon = \varepsilon^\perp$ for any rate 1/2 code.

The number of edges in the minimal trellis of a code and its dual is not as conveniently related. From Eq. (7) and Theorem 1,

$$e_i = e_i^\perp + (1 - r_i^\perp - l_i^\perp)$$

for each $1 \le i \le n$. Consequently, since $|1 - r_i^\perp - l_i^\perp| \le 1$, and from the definition of E,

$$\frac{1}{q}E \le E^\perp \le qE$$

Equality is possible only for the degenerate $(n, n, 1)$ code or its dual.

D. Minimal Trellises for Self-Dual and Other Special Codes

For self-dual codes, the theory of the previous two sections collapses neatly to yield stronger results because, for any such code, $l_i = l_i^\perp$ and $r_i = r_i^\perp$ for all i. Consequently, from Theorem 1,

Theorem 2. For any self-dual code C, for each $i = 1, 2, \cdots n$, either

(1) $l_i = 1$ and $r_i = 0$, or

(2) $l_i = 0$ and $r_i = 1$

i.e., every stage corresponds to an information symbol when encoding from one direction and a parity symbol when encoding from the other direction. The only primitive trellis structures in $\mathcal{T}(\mathcal{C})$ are simple expansions and simple mergers.

The converse of Theorem 2 does not hold: A code whose minimal trellis contains only simple expansions and mergers need not be self-dual. However, such a trellis can always be relabeled to represent a self-dual code.

The following theorem, which is a consequence of Theorem 2 and Eqs. (3), (4), and (5), shows that, for self-dual codes, the complexity measures E, V, and M are linearly related, and the maximum edge and vertex dimensions are equal.

Theorem 3. For any self-dual code,

$$V = \frac{q+1}{2q}E + 1$$

$$M = \frac{1}{2q}E$$

$$s_{\max} = e_{\max}$$

There is another case where we can restrict the type of structures that can appear in the trellis for a code:

Theorem 4. If $\mathcal{C}$ is a code with all codeword weights divisible by some integer $m > 2$, then,

(1) There does not exist a position i such that $l_i = r_i = 1$, i.e., $\mathcal{T}(\mathcal{C})$ contains no butterfly structures.

(2) $\mathcal{C}$ cannot have rate greater than $\frac{1}{2}$.

(3) $e_{\max} = s_{\max}$

(4) $V \geq \left(\frac{q+1}{q}\right)E + 1$

(5) $M \leq \frac{1}{2q}E$

(6) $V^{\perp} \leq \left(\frac{q+1}{q}\right)E^{\perp} + 1$

(7) $M^{\perp} \geq \frac{1}{2q}E^{\perp}$

Proof: If $l_i = r_i = 1$, then the ith column begins and ends spans in the MSGM. This implies the existence of codewords of the form $x = XXX\cdots X10^{n-i}$ and $y = 0^{i-1}(-1)XXX\cdots X$, where (-1) denotes the additive inverse of 1 in $GF(q)$ and $XXX\cdots X$ denotes some string of symbols in $GF(q)$. Then $x + y$ is a codeword of weight $|x| + |y| - 2$, which cannot be divisible by m. This proves (1). From (1), we have $l_i + r_i \leq 1$ for all i, so $2k = \sum_{i=1}^{n}(l_i + r_i) \leq \sum_{i=1}^{n} 1 = n$, which proves (2). The fact that $\mathcal{T}(\mathcal{C})$ can have no butterfly structures proves (3). From Eq. (13), $2qM = \sum_{i=1}^{n}(l_i + r_i)q^{e_i} \leq \sum_{i=1}^{n} q^{e_i} = E$, proving (5), and (4) follows directly. Since $l_i + r_i \leq 1$, Theorem 1 implies $l_i^{\perp} + r_i^{\perp} \geq 1$, which gives (6) and (7). ❐

Codes for which all codeword weights are divisible by some integer other than one are called divisible codes [7]. Examples of divisible codes include the (31,10,12) binary cyclic codes and doubly even self-dual codes such as the extended Golay code.

The converse of Theorem 4 does not hold—a code is not necessarily divisible when $l_i + r_i \leq 1$ for all i. If a code and its dual satisfy the conditions of Theorem 4, then the code strongly resembles a self-dual code: The code must have rate 1/2 and its trellis contain only simple expansions and simple mergers.

IV. Conclusion

In this article, we have examined the trellis complexity problem by first considering the minimal span generator matrix for a fixed permutation of a code. McEliece showed that the so-called minimal trellis indeed minimizes not only the maximum state dimension of the trellis but also a whole gamut of complexity measures.[8] Here we have augmented the list of reasonable complexity measures and interrelated them. We have also illustrated the connection between the complexity measures and the four primitive structures of a minimal trellis for a nondegenerate code.

We developed some useful relationships between the minimal trellis of a code and that of its dual. The duality relationships lead to interesting connections among several of the complexity measures for the special case of self-dual codes.

Acknowledgment

A portion of W. Lin's contribution was sponsored by AFOSR grant no. F4960-94-1-005. Part of this work was presented at the 32nd Annual Allerton Conference on Communication, Control, and Computing in Allerton, Illinois, in October 1994.

References

[1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory*, vol. IT-20, no. 2, pp. 284–287, March 1974.

[2] A. B. Kiely, S. Dolinar, R. J. McEliece, L. Ekroot, and W. Lin, "Trellis Complexity Bounds for Decoding Linear Block Codes," *The Telecommunications and Data Acquisition Progress Report 42-121, January–March 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 159–172, May 15, 1995.

[3] G. D. Forney, "Coset Codes—Part II: Binary Lattices and Related Codes (Appendix A)," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, 1988.

[4] G. D. Forney, "Dimension/Length Profiles and Trellis Complexity of Linear Block Codes," *IEEE Trans. Inform. Theory*, vol. 40, no. 6., pp. 1741–1752, November 1994.

[8] Ibid.

[5] F. R. Kschischang and V. Sorokine, "On the Trellis Structure of Block Codes," *Proceedings of the 1994 IEEE International Symposium on Information Theory*, Trondheim, Norway, June 27–July 1, 1994, p. 337, 1994.

[6] D. J. Muder, "Minimal Trellises for Block Codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, 1988.

[7] H. N. Ward, "Divisible Codes," *Arch. Math.*, vol. 36, pp. 485–494, 1991.