The Trellis Complexity of Convolutional Codes[1]

Robert J. McEliece and Wei Lin
California Institute of Technology

Abstract — **We develop a theory of minimal trellises for convolutional codes, and find that the "standard" trellis need not be the minimal trellis.**

I. Introduction

From a minimal generator matrix $G(D)$ for an (n,k,m) convolutional code, it is possible to construct a "standard" trellis representation for $\mathcal{C}$. This trellis is in principle infinite, but it has a very regular structure, consisting (after a short initial transient) of repeated copies of what we shall call the *trellis module* associated with $G(D)$. The trellis module consists of 2^m "initial states" and 2^m "final states," with each initial state being connected by a directed edge to exactly 2^k final states. Each directed edge is labelled with an n-bit binary vector, namely, the output produced by the encoder in response to the given state transition.

Since the trellis module has 2^{k+m} edges, and each edge has "length" (measured in bits) n, then total edge length of the trellis module is $n \cdot 2^{k+m}$. Since each trellis module represents the encoder's response to k input bits, we are led to define the "standard trellis complexity" of the code as

$$\frac{n}{k} \cdot 2^{m+k} \quad \text{edges per bit.} \tag{1}$$

The standard trellis complexity as defined in (1) is a measure of the effort *per decoded bit* required by Viterbi's algorithm. However, we will see in the next section that this complexity can sometimes be reduced, by the construction of a simplified trellis for the code.

II. Example

Consider the $(8,4,3)$, $d_{\text{free}} = 8$, "partial unit memory" convolutional code with minimal generator matrix

$$G(D) = \begin{pmatrix} 11111111 \\ 11101000 \\ 10110100 \\ 10011010 \end{pmatrix} + \begin{pmatrix} 00000000 \\ 11011000 \\ 10101100 \\ 10010110 \end{pmatrix} D \tag{2}$$

(see [3]). According to (1), the "standard" trellis complexity of this code is 256 edges per bit. However, it is quite easy to reduce this number, as follows.

We view the code in (2) as an (infinite-length) block code, with "scalar" generator matrix

$$G_{\text{scalar}} = \begin{bmatrix} G_0 & G_1 & & & & \\ & G_0 & G_1 & & & \\ & & G_0 & G_1 & & \\ & & & G_0 & G_1 & \\ & & & & & \ddots \end{bmatrix} \tag{3}$$

where $G(D) = G_0 + D \cdot G_1(D)$. From this representation, and using a modification of the now "standard" theory of trellises for block codes [4], one can see that the code has a minimal trellis, built from trellis modules, each of which has 480 edges. Since each module represents four encoded bits, the trellis complexity, as measured in trellis edges per encoded bit, is thereby reduced to 120.

In this example, the trellis complexity can be reduced still further, if we allow column permutations of the original generator matrix $G(D)$ in 2. Indeed, by computer search we have found that one "minimal complexity" column permutation for this particular code is the permutation (01243567), which results in the generator matrix (cf. (2))

$$G(D) = \begin{pmatrix} 11111111 \\ 11110000 \\ 10101100 \\ 10011010 \end{pmatrix} + \begin{pmatrix} 00000000 \\ 11011000 \\ 10110100 \\ 10001110 \end{pmatrix} D. \tag{4}$$

Then after putting the minimal generator matrix of (4) into "minimal span" form, it becomes

$$G(D) = \begin{pmatrix} 11111111 \\ 00001111 \\ 01111111 \\ 00111111 \end{pmatrix} + \begin{pmatrix} 00000000 \\ 11111000 \\ 11111100 \\ 11111110 \end{pmatrix} D. \tag{5}$$

The trellis complexity of the generator matrix in (5) turns out to be 104 edges per encoded bit.

III. General Results

We have found a simple algorithm algorithm for finding a generator matrix $G(D)$ for a convolutional code, for which the corresponding "scalar" generator matrix (cf. (3)) is in "minimal span" form [4]. This generator matrix can then be used to produce the minimal trellis for the convolutional code. In principle, the theory of minimal trellises for convolutional codes can be deduced from the general "Forney-Trott" theory [2], but we believe the observation that the Viterbi decoding complexity of convolutional codes can be thereby systematically reduced is new, as are the details of the algorithms for producing the minimal trellises.

One nice by-product of our theory is that when we apply our techniques to a convolutional code obtained by puncturing [1], we always find a trellis for that code which is as least as simple as the "punctured" trellis. Thus in the new theory, punctured convolutional codes no longer appear as a special class, but simply as high-rate convolutional codes whose trellis complexity turns out to be unexpectedly small.

References

[1] J. B. Cain, G. C. Clark, and J. M. Geist, "P:unctured Convolutional Codes of rate $(n-1)/n$ and simplified maximum likelihood decoding," *IEEE Trans. Infrom. Theory*, vol. IT-25 (January 1979), pp. 97–100.

[2] G. D. Forney, Jr., and M. D. Trott, "The dynamics of group codes: state spaces, trellis diagrams, and canonical encoders." *IEEE Trans. Inform. Theory, vol. IT-39 (1993), pp. 1491–1513.*

[3] G. S. Lauer, "Some optimal partial-unit-memory codes," *IEEE Trans. Inform. Theory*, vol. IT-25 (March 1979), pp. 540–547.

[4] R. J. McEliece, "On the BCJR Trellis," submitted to *IEEE Trans. Inform. Theory.*

[1]This work was partially supported by a grant from Pacific Bell.