A Multivariate Weight Enumerator for Tail-biting Trellis Pseudocodewords

Nigel Boston*

Departments of Mathematics, Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706, USA

Abstract

Tail-biting-trellis representations of codes allow for iterative decoding algorithms, which are limited in effectiveness by the presence of pseudocodewords. We introduce a multivariate weight enumerator that keeps track of these pseudocodewords. This enumerator is invariant under many linear transformations, often enabling us to compute it exactly. The extended binary Golay code has a particularly nice tail-biting-trellis and a famous unsolved question is to determine its minimal AWGN pseudodistance. The new enumerator provides an inroad to this problem.

Keywords: Tail-biting trellis; Binary Golay code; Pseudocodewords; Weight enumerators; Invariant theory

Introduction

This paper makes headway into a longstanding problem, namely that of whether the Golay tail-biting trellis found by Calderbank, Forney, and Vardy [1] has any pseudocodewords of AWGN pseudoweight less than 8. This problem was addressed by many experts in the field before being abandoned as being computationally too hard. We circumvent this issue by introducing a new kind of multivariate weight enumerator that is invariant under the action of a large group of matrices. This enumerator is not covered by the encyclopedic work of Nebe, Rains, and Sloane [2], but the method, dating back to Gleason [3], of using invariant theory so as to constrain drastically the form of the enumerator polynomial, works well. In particular, we show that there are no Golay pseudocodewords of period less than 5 and AWGN pseudoweight less than 8. Deeper explorations of these techniques should permit us to extend this result to periods of 5 and higher.

Tail-biting Trellises

Definition: A tail-biting trellis $T = (V, E, A)$ of depth $n$ is a directed graph with vertex set $V$ and edge set $E$ such that each edge has a label taken from alphabet $A$ and with the following property: the set $V$ can be partitioned into $n$ subsets $V = V_0 \cup V_1 \cup \ldots \cup V_{n-1}$ such that every edge in $T$ either begins at a vertex of $V_0$ and ends at a vertex of $V_i$, for some $i$, with $0 \leq i \leq n - 2$ or begins at a vertex of $V_{n-1}$ and ends at a vertex of $V_{n'}$.

The set of edge labels along a cycle in $T$ starting at a vertex in $V_0$ is an $n$-tuple in $A^*$. We say that $T$ represents a linear block code $C$ over $A$ if $C$ is precisely the set of all edge-label sequences in $T$.

Conventional trellises, corresponding to the case where $|V_0| = 1$, have an older history dating back to the 60’s. Trellis representations of linear block codes provide efficient decoding algorithms such as the two-way, or BCJR, algorithm [4,5] and the Viterbi algorithm [6]. Their complexity depends on the state complexity $\sigma = \max|V_i|$ and associated to each code is a minimal conventional trellis, unique up to isomorphism.

Tail-biting trellises date back to 1979 [7]. Their advantage is that they can have much lower state complexity (in fact $\sqrt{n}$) than that, $\sigma$, of the minimal conventional trellis associated to the code. Also, they are the simplest kind of factor graph with cycles. On the other hand, there are different notions of minimality [8] and so no uniqueness for minimal tail-biting trellises associated to a given code. Additionally, their construction can be hard, as in one famous case given next.

The Extended Binary Golay Code

One of the most extraordinary binary linear codes is the [24, 12] extended binary Golay code. Muder [9] showed that the state complexity of its minimal conventional trellis is at least 256 and Forney [10] that this bound is met. It follows that any associated tail-biting trellis has state complexity at least 16 and in 1996 Forney issued the challenge of finding a tail-biting trellis meeting this bound. In 1999, Calderbank, Forney, and Vardy successfully answered this challenge [1].

The corresponding trellis-oriented generator matrix is

```
1 1 0 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Pseudocodewords

Because of the local nature of iterative decoding algorithms they can decode to vectors that are not codewords of the original code but come from some 'covering' code instead. This is most commonly studied in the case of belief propagation algorithms on Tanner graphs of low density parity check codes, where since finite covers of Tanner graphs.
graphs are locally identical to the original graph, the algorithm can be misled by codewords associated to the cover. This leads to a theory of pseudocodewords for which advanced algebraic tools have already been developed by, for instance, Koetter, Li, Vontobel, and Walker [11]. Much work has also been carried out to define suitable pseudoweights for various channels and the minimum pseudoweight of a nonzero pseudocodeword becomes a better performance measure for the code than its minimum distance.

Tail-biting trellises also yield pseudocodewords. Namely, given \( T = (V, E, A) \) and any positive integer \( m \), define tail-biting trellis \( T^m = (V^m, E^m, A) \) of depth \( mn \) by letting \( V^m \) be a copy of \( V \) where \( j = i (\text{mod} \ n) \) and letting the edges from \( V^m \) to \( V^{m+1} \) be those from \( V_{(i+1) \text{mod} \ n} \) to \( V_{(j+1) \text{mod} \ n} \). The edge-labels on cycles in \( T^m \) starting at \( V^m \) yield a code \( C_m \). Let \( C^0 \) be the old code \( C \) represented by \( T \) and \( C_m \) be the extended \([8, 4, 4]\) Hamming code and \( C_m \) be those from \( C^0 \) and \( \text{Dr} \).

Let \( \overline{r} \) be a pseudocodeword of period \( m \). Attach \( r_i \) to \( \overline{r} \) for \( i = 0, \ldots, m \). Note that \( \overline{r} \) is of period \( m \) is defined to be \( r_i \) for \( i = 0, \ldots, m \). The edge-labels on cycles in \( T^m \) starting at \( V^m \) yield a code \( C_m \).

Assume that \( C \) is binary. If \( (c_{0,i}, \ldots, c_{m,i}) \) is in \( C_m \), its associated pseudocodeword is defined to be \( \mathbf{p} := (\mathbf{p}_1, \ldots, \mathbf{p}_m) \) where \( \mathbf{p} \) counts the number of nonzero \( c_i \) for \( i \) (mod \( n \)) = \( j \). Note that sometimes this is normalized by dividing each entry by \( m \). We say that \( \mathbf{p} \) is of period \( m \).

In the case where \( C \) is the extended Golay code, a trellis-oriented generator matrix for \( C_m \) is given as follows. Let \( M \) be the generator matrix given in section 2. Let \( A \) be the matrix obtained by zeroing out the ones in the bottom left hand corner of \( M \) and \( B = M - A \), i.e. the matrix obtained by zeroing out everything but the bottom left hand corner. Let \( Z \) be the zero matrix of the same dimensions as \( M \). Define \( M_n \) to be the \( 12m \)-by-24m block matrix

\[
\begin{pmatrix}
A & B & Z & Z & Z & Z & Z \\
B & Z & A & B & Z & Z & Z \\
Z & Z & Z & A & B \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
Z & Z & Z & Z & A & B \\
B & Z & Z & Z & Z & A \\
\end{pmatrix}
\]

Then \( C_m \) is the \( [24m, 12m, 8] \) binary linear code with generator matrix \( M_m \). Note that in the limit as \( m \rightarrow \infty \), \( M_m \) yields the infinite recurring generator matrix for the Golay convolutional code.

Much work has been done on pseudoweights of pseudocodewords in this case, notably in Aji et al. [13] and Forney et al. [14]. The question of whether there exist nonzero pseudocodewords of AWGN pseudoweight less than 8 apparently became a major challenge but remains open to this day. There are many nontrivial near-misses - for example, there are pseudocodewords of period 2 with 2 in 2 positions, 1 in 8 positions, and 0 elsewhere, which therefore have AWGN pseudoweight \( 12/16 \) of 9, and pseudocodewords of period 3 with 3 in 2 positions, 2 in 2 positions, 1 in 6 positions, and 0 elsewhere, which therefore have AWGN pseudoweight 16/32 of 8.

This question motivated the current work. It is feasible to find all 224 codewords of \( C_m \) and hence all corresponding pseudocodewords but to do the same with \( C_m \) is already computationally intense. Resolving the question by brute force would require doing the same for all \( C_m \) for \( 1 \leq m \leq 16 \) since cycles can go around up to 16 times before necessarily returning to the same vertex of \( V \).

**Pseudocodeword Weight Enumerators**

Let \( \mathbf{p} := (p_0, \ldots, p_{m-1}) \) be a pseudocodeword of period \( m \). Attach \( p_r \) to \( \mathbf{p} \) the monomial \( x_0^{p_0} x_1^{p_1} \ldots x_n^{p_n} \) where \( r \) is the number of occurrences of \( i \) in \( \mathbf{p} \). Note that \( r_0 + r_1 + \ldots + r_m = n \). For example, the two pseudocodewords referred to in the penultimate paragraph of section 3 yield monomials \( x_0^{14} x_1^{8} x_2^{1} \) and \( x_0^{4} x_1^{6} x_2^{2} x_3^{2} \) respectively. Note that the AWGN pseudoweight can be calculated from the corresponding monomial.

Next, define the pseudocodeword weight enumerator \( W_m \) associated to pseudocodewords of period \( m \) to be the sum of all these monomials. Note that in the limit as \( m \rightarrow \infty \), \( W_m \) is a polynomial in \( m + 1 \) variables \( x_0, \ldots, x_n \) with non-negative integer coefficients. So, for example, \( W_2 \) is the usual weight enumerator. For the extended Golay code,

\[
W_2 = x_0^{24} + 759 x_0^{16} x_1 + 2576 x_0^{12} x_1^2 + 759 x_0^{8} x_1^3 + x_1^4
\]

As noted above, a brute force calculation of \( W_m \) for the extended Golay code is feasible. We thereby obtain:

\[
W_m = x_0^{24} + 294 x_0^{16} x_1 + 759 x_0^{12} x_1^2 + 9792 x_0^{8} x_1^3 + 5152 x_0^4 x_1^4 + 178248 x_0^2 x_1^5 + x_1^6
\]

Knowing \( W_m \) is enough to establish that there are no nonzero pseudocodewords of period 2 with pseudoweight less than 8. Our strategy then will be to try to compute \( W_m \) for all \( m \) by using the fact that \( W_m \) has some very nice transformation properties.

**Invariant Theory**

In her 1962 Harvard PhD thesis [15], MacWilliams showed that the weight enumerator \( W \) of the dual of a binary linear code \( C \) is closely related to that of the code. In particular, \( W \) is invariant under the transformation

\[
\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \mapsto \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}
\]

If the weights of all code words are divisible by 4 (\( C \) is then called doubly even), then \( W \) is also invariant under the transformation

\[
\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}
\]

Thus, \( W \) is invariant under all possible compositions of these two transformations, of which there are 192, forming what is called the Clifford-Weil group \( G_r \).

In 1970, Gleason [3] observed that this imposes a strong restriction on the structure of \( W \), since every homogeneous polynomial in \( x_0, x_1 \) invariant under \( G_r \) is a polynomial in the weight enumerators \( W_2 = x_0^{24} + 14 x_0^{22} x_1^2 + x_1^4 \) of the extended \([8, 4, 4]\) Hamming code and \( W_2 = W_2 \) given in the previous section) of the extended Golay code. This permits quick computation of weight enumerators of large self-dual doubly even codes.

In the last few decades, hundreds of papers have appeared generalizing and applying Gleason’s results, culminating in the book [2] by Nebe, Rains, and Sloane unifying these theories. They define the
Type of self-dual code such that the weight enumerator of any code of that Type lies in the ring of a certain Clifford-Weil group associated with that Type, and furthermore such that this invariant ring is spanned by weight enumerators of that Type. There are also fascinating algebra isomorphisms due to Broué and Enguehard between various rings of modular forms and rings of weight enumerators [16].

We are guided by a similar philosophy below, in seeking to compute the multivariate weight enumerators \(W_f\) for the Golay case. This theory is new, not covered by the above book.

**Symmetric Power Invariance**

Let \(A\) be a 2-by-2 matrix. Then \(A\) acts on \(x_0, x_1\) by linear transformations. Substituting these into \(x_0^m, x_0^{m-1} x_1, x_0^{m-2} x_1^2, ..., x_1^m\), yields a linear transformation of those \(m + 1\) terms and hence produces an \(m + 1\)-by-\(m + 1\) matrix, denoted \(\text{Sym}^m(A)\). For example, if \(A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}\) then \(\text{Sym}^m(A) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\). Similarly one computes that \(\text{Sym}^m(x_0 x_1) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\) and \(\text{Sym}^m(x_0^2) = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}\). It follows that \(A\) sends

\[
\begin{pmatrix} x_0^2 \\ x_0 x_1 \\ x_1^2 \end{pmatrix} \mapsto \frac{1}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} x_0^2 \\ x_0 x_1 \\ x_1^2 \end{pmatrix}
\]

The above 3-by-3 matrix is then \(\text{Sym}^3(A)\). Applying this to all 192 matrices in the Clifford-Weil group \(G_3\), above yields 96 3-by-3 matrices (we say that the homomorphism \(\text{Sym}^m\) has a kernel of order 1). Each such matrix defines a linear transformation in \(x_0, x_1\). One can check using a computer algebra system such as Magma that \(W_f\) is invariant under all 96 of these transformations.

In fact \(W_f\) is also invariant under the transformation \(x_0 \mapsto x_0, x_1 \mapsto x_1, x_2 \mapsto -x_2\). Compositions formed from this and the 96 transformations above yield a group \(G_3\) of 384 3-by-3 matrices, all leaving \(W_f\) invariant.

If we had known a priori that \(W_f\) is invariant under \(G_3\), then we could have used Magma to compute all homogeneous degree 24 polynomials invariant under \(G_3\) (it turns out that they are spanned by 6 such polynomials \(I_1, ..., I_6\)). The correct linear combination of those 6 polynomials (in fact \(I_6 + 294I_4\)) could then be found by exploiting a computation of low weight codewords of \(C_4\) and \(C_3\), which would otherwise have been out of reach. The main (nontrivial) point, proven in David Conti’s upcoming University College Dublin PhD thesis, is that, for every \(m\), \(W_f\) is invariant under every matrix \(\text{Sym}^m(A)\) where \(A\) is in \(G_3\) and under certain diagonal matrices. This produces a typically large group \(G_3\), of \((m + 1)\)-by-\((m + 1)\) matrices leaving \(W_f\) invariant.

The group \(G_3\) is the group of 3-by-3 quasipermutation matrices which have exactly one nonzero entry, a 4th root of unity, in every row and every column. This makes it isomorphic to the wreath product \(S_3 \wr S_3\). It is also a complex reflection group, which makes its invariant theory particularly nice, leading to the following pretty formula. Let

\[
f(x_0, x_1, x_2) = x_0^4 + x_0^2 x_1 + x_1^2 + x_2^4 + x_2^6 + x_3^8 + 7560x_4^8 + 288x_5^8 + 4608x_6^8 + ...
\]

Then \(W_f(x_0, x_1, x_2) = x_0^4 f(4, x_0 + x_2) / 2, x_1, (x_0 - x_2) / 2\).

**Golay Pseudocodeword Enumerators**

We move first to computing \(W_f\). The above theory shows that \(W_f\) is invariant under \(\text{Sym}^m(A)\) for \(A\) in \(G_3\). This yields 192 transformations. In addition, \(W_f\) is invariant under the transformation

\[
x_0 \mapsto x_1, x_1 \mapsto x_2, x_2 \mapsto x_3, x_3 \mapsto x_1\cdot
\]

All possible compositions of the above transformations yield a group \(G_5\) of 1152 4-by-4 matrices leaving \(W_f\) invariant.

Next, using Magma, we compute all homogeneous degree 24 polynomials in \(x_0 x_1, x_1 x_2, x_0 x_1\) invariant under \(G_5\). Magma produces 26 polynomials \(I_1, ..., I_{26}\) that span this space. Using low weight codewords in \(C_5\) yields a simple linear combination of them that must equal \(W_f\).

Namely, \(I_1 + I_2 + I_3 + I_4 + I_5 + I_6\). This is the strategy followed that \(W_f\) sends \(A\) to \(A\) for every \(A\), and the analysis of low weight code words of \(C_5\) becomes computationally too expensive. It is clear that there are 147 \(m\)-by-\((m + 1)\) matrices leaving \(W_f\) invariant under \(G_3\). Unfortunately, both \(C_3\) and \(C_4\) are pseudocodewords that have pseudoweight at least 8.

As for \(W_f\), we similarly obtain a group \(G_3\) of 384 5-by-5 matrices that leave \(W_f\) invariant. The space of homogeneous degree 24 polynomials in \(x_0 x_1 x_2\) invariant under \(G_3\) is spanned by 153 very lengthy polynomials which Magma gives explicitly. Of these 153 polynomials, 87 have the property that every monomial occurring in them corresponds to pseudocodewords of pseudoweight at least 8. We show that \(W_f\) is a span of these 87 polynomials by excluding the other 66 polynomials as follows. Every such polynomial contains a monomial that occurs in it and none of the remaining 152 polynomials. Examining these special monomials, we see that, for those 66 polynomials, if the special monomial were present, it would come from a codeword of \(C_5\) of weight at most 24. By analyzing low weight codewords of \(C_5\) we can show that this does not happen. Thus, there are no nonzero pseudocodewords of period \(\leq 4\) of pseudoweight less than 8.

Likewise, for \(m \geq 5\), there is an explicitly given group \(G_3\) of \(m + 1\)-by-\((m + 1)\) matrices leaving \(W_f\) invariant. Unfortunately, both the computation of homogeneous degree 24 polynomials invariant under \(G_5\) and the analysis of low weight code words of \(C_5\) become computationally too expensive. It is clear that there are 147 \(m\)-by-\(m\) code words of \(C_5\) of weight 8, but beyond that patterns are hard to spot. For example, the codewords of \(C_5\) of weight 12 correspond to pseudocodewords either consisting of 12 ones and 12 zeros or 2 twos, 8 ones, and 14 zeros, but there is no clear, even conjectural, formula for the number of either kind.

**Conclusions and Further Work**

We have introduced new and useful multivariate polynomials attached to a tail-biting trellis. These keep track of what kinds of pseudocodewords exist and indeed how many there are of each kind. This can in turn be used to measure how good the code is as regards iterative decoding, with various formulae for pseudoweight being used, depending on the channel. It has been a longstanding question to determine whether the AWGN pseudo weight of a nonzero pseudocodeword for the tail-biting trellis of the extended binary Golay code is ever less than 8. Our new invariant theory methods allow us to answer this question in the negative for all pseudocodewords of period \(\leq 4\).
Pseudocodeword weight enumerators $W_m$ are defined for any tail-biting trellis. It is not true, however, that they are invariant under the same group $G_m$ as for the Golay code above. For example, for tail-biting trellises attached to the extended [8, 4, 4] Hamming code, the polynomials $W_m$ are invariant under slightly smaller groups than $G_m$.

The author’s PhD student, David Conti, is developing a theory that should hopefully clarify the notion of Type for a tail-biting trellis and allow one to define an analogue of the Clifford-Weil group for each Type.

References
1. Calderbank AR, Forney GD, Vardy A (1999) Minimal tail-biting trellises: The Golay code and more. IEEE Transactions on Information Theory 45: 1435-1455.
2. Nebe G, Rains EM, Sloane NJA (2006) Self-Dual Codes and Invariant Theory. Springer-Verlag.
3. Gleason AM (1971) Weight polynomials of self-dual codes and the MacWilliams identities. Gauthiers-Villars, Paris 3: 211-215.
4. Bahl LR, Cocke J, Jelinek F, Raviv J (1974) Optimal decoding of linear codes for minimizing symbol error rate. IEEE Transactions on Information Theory 20: 284-287.
5. Gallager RG (1963) Low-density parity-check codes. MA, MIT Press, Cambridge.
6. Viterbi AJ (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13: 260-269.
7. Solomon G, van Tilborg HCA (1979) A connection between block and convolutional codes. SIAM Journal of Applied Mathematics 37: 358-369.
8. Koetter R, Vardy A (2003) The structure of tail-biting trellises: minimality and basic principles. IEEE Transactions on Information Theory 49: 1877-1901.
9. Muder DJ (1988) Minimal trellises for block codes. IEEE Transactions on Information Theory 34: 1049-1053.
10. Forney GD (1994) Dimension/length profiles and trellis complexity of linear block codes. IEEE Transactions on Information Theory 40: 1741-1752.
11. Koetter R, Li WCW, Vontobel PO, Walker JL (2007) Characterizations of pseudocode words of (low-density) parity-check codes. Advances in Mathematics 213: 205-229.
12. Wilber N (1996) Codes depending on general graphs, Doctor of Philosophy Dissertation. Department of Electrical Engineering, University of Linkoping, Sweden.
13. Aji S, Horn G, McEliece R, Xu M (1998) Iterative min-sum decoding of tail-biting codes. Proc ITW workshop, Killamey.
14. Forney GD, Koetter R, Kschischang FR, Reznik A (1999) On the effective weights of pseudocodewords for codes defined on graphs with cycles, in Codes. The IMA Volumes in Mathematics and its Applications 123: 101-112.
15. MacWilliams FJ (1963) A theorem on the distribution of weights in a systematic code. Bell Syst Tech J 42: 79-94.
16. Solé P (2008) Codes, invariants, and modular forms, talk at Bonn MPIM.