A Labeling Procedure for Linear Finite-State Codes

K.-M. Cheung
Communications Systems Research Section

In this article a method to define the labels of the state diagram of a linear finite-state code [1] is presented and investigated. This method is particularly suitable for simple hardware implementation since it simplifies the encoder structure. The method can also be applied to the labeling of a state diagram that is not completely connected to obtain a linear finite state code with larger free distance.

I. Introduction

It was shown in [1] that a finite-state code (*FS* code) on a completely connected state diagram with 2^m states requires at least 2^{m+1} labels. Also, a simple method to define such labels has been suggested in [1]. However, the codes constructed using the method in [1] are not linear. In this article, another method using shift registers to define the labels of the state diagram of the *FS* codes is presented. This method is particularly suitable for simple hardware implementations since it simplifies the encoder structure. The method can also be applied to the labeling of a state diagram that is not completely connected to obtain an *FS* code with larger free distance. Lastly, a mapping scheme to assign the cosets to the labels generated by the shift registers is described. It can be shown that by using the above method, a linear *FS* code can be constructed.

In order to facilitate the discussion on *FS* codes with non-completely connected state diagrams as well as those with completely connected state diagrams, the following definition of *FS* codes is adopted:

Definition 1: An (n, k, m) finite state code (*FS* code) on a c-connected state diagram is a code with the following properties:

(1) The code has rate k/n.

(2) Its operation can be represented by a state diagram with 2^m states.

(3) There are 2^c $(c \leqslant m)$ branches going into each state and 2^c branches going out of each state.

(4) Each branch of the state diagram is associated with a code (code word length $= n$ and code size $= 2^{k-c}$), and any two different codes associated with different branches are disjoint.

II. Preliminaries

Some important results in the theory of convolutional codes will now be reviewed. These results will be referred to in the proofs in later sections.

A typical encoder of an (n_1, c, m) convolutional code consists of a linear sequential circuit (with c shift registers) that accepts c input bits and outputs n_1 bits. It is well known that the operation of the encoder can be represented by (1) a state diagram with 2^m states, 2^c branches going into each state, and 2^c branches going out of each state; or (2) a $c \times n_1$ transfer function matrix (denoted by $G[D]$) such that the entries of the matrix are polynomials in D, representing the generator sequences of the code.

In order to avoid catastrophic error propagation, the transfer function matrix must satisfy Massey and Sain's condition [2] (a necessary and sufficient condition) on non-catastrophic codes:

$$GCD\left[\Delta_i(D), \qquad i = 1, 2, \ldots, \binom{n_1}{c}\right] = D^l$$

for some $l \geqslant 0$, where $\Delta_i(D)$, $i = 1, 2, \ldots, \binom{n_1}{c}$ are the determinants of the $\binom{n_1}{c}$ distinct $c \times c$ submatrices of the transfer function matrix $G(D)$.

III. Generation of Labels by Shift Register

FS code encoders have structural properties very similar to those of convolutional encoders, and their operation can be described by a state diagram. In the case of a convolutional code, each branch of the state diagram is labeled by an n_1-bit output sequence, whereas in the case of a finite-state code according to Definition 1, each branch is labeled by a code that is not necessarily linear. Because of the similarities between convolutional codes and finite state codes, it should be expected that much of the theory on structural properties of convolutional codes will be applicable to finite state codes.

In order to guarantee a noncatastrophic finite state code with good distance properties, the labeling of the branches of the state diagram must satisfy the following conditions [1]: (1) different labels out of each state; (2) different labels into each state; and (3) no disjoint paths with identical labels that remain unmerged indefinitely.

A method to assign the labels of the state diagram of a finite state code by using the linear sequential circuit (with shift registers) of a noncatastrophic (n_1, c, m) convolutional code is now described. Let the c shift registers have lengths $l_1, l_2, \ldots, l_c$ where $l_1 + l_2 \ldots + l_c = m$. The pth row of the corresponding $c \times n_1$ transfer function matrix thus consists of polynomials in D of degree no greater than l_p for $1 \leqslant p \leqslant c$. The state diagram of the convolutional code consists of 2^m states (each state is defined by the shift register content); also, there are 2^c branches going into each state and 2^c branches going out of each state. Each branch in the state diagram is assigned an n_1-bit sequence $b_0, b_1, \ldots, b_{n_1-1}$, which consists of the n_1 output bits of the shift registers. Let us assign to the branches of the state diagram, which are associated with the n_1-bit sequence $b_0, b_1, \ldots, b_{n_1-1}$, the label i such that $i = b_0 + 2b_1 + \ldots + 2^{n-1} b_{n_1-1}$. Each of these labels represents one of the disjoint codes. There are 2^{n_1} of them. This modified state diagram of the convolutional code is used as the state diagram of an (n, k, m) finite state code on a c-connected state diagram.

The construction of a shift register circuit that generates the state diagram of a finite state code that satisfies conditions 1, 2, and 3 is given as follows. It is not hard to see that condition 1 is satisfied if, for a fixed shift register content, different inputs to the shift registers produce different outputs. This can be achieved if there exists at least one $c \times c$ submatrix $\Omega_i(D)$ of the transfer function matrix $G(D)$, $i = 1, 2, \ldots, \binom{n_1}{c}$, such that the term "1" appears exactly once in each row and in each column of $\Omega_i(D)$. Similarly, condition 2 is satisfied if, for a fixed input, different shift register contents produce different outputs. This can be achieved if there exists at least one $c \times c$ submatrix $\Omega_j(D)$, $j = 1, 2, \ldots, \binom{n_1}{c}$, such that the term D^{l_p} representing the last shift register stage of the pth shift register appears exactly once in row p for $1 \leqslant p \leqslant c$, and each of these $D^{l_1}, D^{l_2}, \ldots, D^{l_c}$ terms appears in different columns of $\Omega_j(D)$.

It was shown in [3] that if the (n_1, c, m) convolutional code that generates the state diagram of the finite state code is noncatastrophic, then the labeling also satisfies condition 3. Thus, the $c \times n_1$ transfer function matrix $G(D)$ of the convolutional code must satisfy Massey and Sain's condition [2]. It will be shown in later sections that the minimum value n_1 could have is $c + 1$. Two algorithms to construct a $c \times c + 1$ transfer function matrix $G(D)$ of the convolutional code are given as follows:

Algorithm 1: Completely connected state diagram, d_f = 2 branches.

(1) Construct a $c \times c$ matrix $G'(D)$ such that

$$\begin{aligned} G'_{ij}(D) &= 1, & i &= 1, \ldots, c, & j &= i \\ &= D, & i &= 1, \ldots, c, & j &= (i+1) \bmod c \\ &= 0 & & \text{otherwise} \end{aligned}$$

(2) Append the column $[1, \ldots, 0]^T$ to $G'(D)$ to obtain a $c \times c + 1$ matrix $G(D)$.

An example of a 3×4 transfer function matrix $G(D)$ constructed using the above algorithm is given in Table 1. It is obvious that $G(D)$ satisfies conditions 1 and 2. Also, it is not hard to see that one of the determinants, $\Delta_j(D)$, $j = 1, \ldots, \binom{c+1}{c}$, equals 1 and the rest are nonzero. Thus

$$GCD\left[\Delta_j(D), \quad j = 1, \ldots, \binom{c+1}{c}\right] = 1$$

Massey and Sain's condition is satisfied and the state diagram generated by this transfer function matrix satisfies conditions 1, 2, and 3.

Algorithm 2: Non-completely connected state diagram. d_f = 3 branches.

(1) Construct a $c \times c$ matrix $G'(D)$ such that

$$\begin{aligned} G'_{ij}(D) &= 1 + D, && i = 1, \ldots, c, \quad j = i \\ &= D^2, && i = 1, \ldots, c, \quad j = (i+1) \bmod c \\ &= 0 && \text{otherwise} \end{aligned}$$

(2) Append the column $[1, \ldots, 0]^T$ to $G'(D)$ to obtain a $c \times c + 1$ matrix $G(D)$.

An example of a 3×4 transfer function matrix $G(D)$ constructed using the above algorithm is given in Table 2. Again it is obvious that $G(D)$ satisfies conditions 1 and 2. Also, it can be shown that one of the determinants, $\Delta_j(D)$, equals $D^{2(c-1)}$ and the rest are nonzero. Thus

$$GCD\left[\Delta_j(D), \quad j = 1, \ldots, \binom{c+1}{c}\right] = D^l$$

where l is some integer. Massey and Sain's condition is satisfied and the state diagram generated by $G(D)$ satisfies conditions 1, 2, and 3.

IV. Properties

On the basis of the labeling procedure by shift register above, which is based on a linear sequential circuit, the finite state code possesses a mathematical structure that facilitates encoding/decoding and simplifies hardware implementation. Also, this labeling procedure is applicable to the construction of finite state codes with incompletely connected state diagrams to obtain larger free distance.

Definition 2: Let N be the number of states of a finite state code. A labeling matrix L of the state diagram is defined to be an $N \times N$ matrix, where $L(i, j)$ denotes the label from state i to state j.

Let $\underline{u} = (u_1, u_2, \ldots, u_c)$ represent the c input bits to the convolutional encoder. Let $\underline{D} = (D_1, D_2, \ldots, D_c)$ represent the last c shift register stages of the convolutional encoder. That is, D_p represents the term D^{l_p} in row p for $1 \leqslant p \leqslant c$. In the following theorems, some properties of *FS* codes which use the new labeling procedure are revealed.

Theorem 1: For a state diagram with 2^m states generated by $G(D)$ which satisfies conditions 1, 2, and 3, if the graph has 2^c branches going into each state and 2^c branches going out of each state, $c \leqslant m$, at least 2^{c+1} labels are required.

Proof: Suppose that 2^c labels suffice. The transfer function matrix $G(D)$ of the convolution code that generates the state diagram of the *FS* code is then a $c \times c$ matrix. By condition 1, since different labels are coming out of each state, the c output bits can be written as

$$\underline{u}\mathbf{A} + \underline{d}$$

where $\mathbf{A}$ is a $c \times c$ nonsingular matrix and $\underline{d}$ is a constant binary c-tuple which depends upon the shift register contents of the encoder. Thus, $|\mathbf{A}| = \alpha$, where α is a nonzero integer. Thus, the term α is contained in the expression of $|\mathbf{M}|$. Similarly, by condition 2, since different labels are going into each state, the c output bits can be written as

$$\underline{D}\mathbf{B} + \underline{e}$$

where $\mathbf{B}$ is a $c \times c$ nonsingular matrix and $\underline{e}$ is a constant binary c-tuple, depending upon the input bits and the shift register contents other than $D_1, \ldots, D_c$. Again, $|\mathbf{B}| = \beta$ for some nonzero integer β. Therefore the term $\beta D^m = \beta D^{l_1 + \ldots + l_c}$ is contained in the expression of $|\mathbf{M}|$. Thus, $|\mathbf{M}| = \beta D^m + \ldots + \alpha$ and $|\mathbf{M}|$ is not of the form D^l for some $l \geqslant 0$. This violates Massey and Sain's condition and the convolutional code is catastrophic. This in turn implies that the state diagram generated by this convolutional encoder is catastrophic and thus at least $c + 1$ output bits for the convolutional encoder are needed. This implies that at least 2^{c+1} labels are needed in the state diagram. ■

In fact, Algorithm 1 and Algorithm 2 in Section II show that $c + 1$ output bits are sufficient to guarantee that conditions 1, 2, and 3 are satisfied.

Theorem 2: Let L be the labeling matrix of a state diagram generated by $G(D)$ which satisfies conditions 1, 2, and 3. Row i and row j (column i and column j), $i \neq j$, of L have

either the same set of labels or a completely different set of labels.

Proof: The state of the convolutional encoder that generates the required state diagram of the finite state code is defined as the shift register contents of the encoder. For an (n_1, c, m) convolutional code, let the binary m-tuple $[D_1, \ldots, D_m]$ denote the state that corresponds to the shift register stages $D_1, \ldots, D_m$ of the encoder. Note that the encoder is constructed in such a way that for a fixed state $[D_1, \ldots, D_m]$, different inputs to the shift registers produce different outputs (condition 1). If $[D_1, \ldots, D_m] = [0, \ldots, 0]$, the set of all possible binary n_1-tuples (labels) that represent the output bits of the encoder forms a c-dimension subspace K of an n_1-dimension vector space over $GF(2)$ (because the encoder is a linear sequential circuit). This set K is isomorphic to the row of the labeling matrix L that corresponds to the state $[0, \ldots, 0]$. Now, if $[D_1, \ldots, D_m] \neq [0, \ldots, 0]$, then it is not hard to see that the set of all possible output binary n_1-tuples (output bits of the encoder) is of the form $K + \underline{e}$, where $\underline{e}$ is a binary n_1-tuple (constant) determined by $[D_1, \ldots, D_m]$. If $\underline{e} \notin K$, then K and $K + \underline{e}$ are disjoint (since K is a c-dimensional subspace in an n_1-dimensional vector space). If $\underline{e} \in K$, then $K = K + \underline{e}$. A similar argument holds for the case of $K + \underline{e}_1$ and $K + \underline{e}_2$, where $\underline{e}_1$ and $\underline{e}_2$ are binary n_1-tuples determined by different $[D_1, \ldots, D_m]$'s. That is, if $\underline{e}_1 \notin K + \underline{e}_2$, then $K + \underline{e}_1$ and $K + \underline{e}_2$ are disjoint. If $\underline{e}_1 \in K + \underline{e}_2$ then $K + \underline{e}_1 = K + \underline{e}_2$. This proves that any two rows of a labeling matrix L have either the same set of labels or a completely different set of labels. The proof for the case of the columns is similar to the one above. ■

V. Assignment of Cosets to Labels

A code C over $GF(q)$ is said to be linear if and only if the following condition is satisfied:

$$\forall \underline{a}, \quad \underline{b} \in C \text{ and } \forall \gamma, \quad \delta \in GF(q), \quad \gamma \underline{a} + \delta \underline{b} \in C$$

In an FS code, even though we have a linear convolutional structure (labels are generated by outputs of shift registers), the overall code may not be linear if the cosets are not properly assigned to the outputs of shift registers. There may exist two code word sequences such that their sum is not a legal code word sequence. In order to generate a linear FS code the following well-known theorem in linear algebra can be used:

Theorem 3 (without proof): If C is a vector space and S is a proper subspace of C, then there exists a subspace W of C such that

$$S + W = C$$

$$S \cap W = \{\underline{0}\}$$

$$\dim S + \dim W = \dim C$$

The following discussion describes a way to generate a linear FS code. The labeling of an FS code can be divided into two parts: (1) generation of labels to the branches in the state diagram; and (2) assignment of cosets to the labels. Part 1 was taken care of by using a convolutional encoder to generate labels to the state diagram of the FS code. For part 2, the method proceeds as follows. Let C be the parent (n, k) code. Let S be an (n, k_1) subcode of C. By Theorem 3, there exists a subcode W of C (W is an $[n, k - k_1]$ code) such that

$$S + W = C$$

$$S \cap W = \{\underline{0}\}$$

$$\dim S + \dim W = \dim C$$

The 2^{k-k_1} cosets are constructed by adding each word in W to S. That is,

$$\underline{w} + S \qquad \forall \underline{w} \in W$$

Note that the set of all binary $k - k_1$-tuples is isomorphic to W. Let $\{\underline{w}_0, \underline{w}_1, \ldots, \underline{w}_{k-k_1-1}\}$ be a basis of W. Let $b_0, b_1, \ldots, b_{k-k_1-1}$ be the $k - k_1$ output bits of the convolutional encoder. Let the coset assigned to the branches labeled by the binary $(k - k_1)$-tuples $b_0, b_1, \ldots, b_{k-k_1-1}$ be denoted by $L(b_0, b_1, \ldots, b_{k-k_1-1})$. Let us assign

$$L(b_0, b_1, \ldots, b_{k-k_1-1}) = S + \{b_0 \underline{w}_0 + b_1 \underline{w}_1 + \ldots + b_{k-k_1-1} \underline{w}_{k-k_1-1}\}$$

This assignment of cosets to the branches in the state diagram guarantees the linearity of the FS code.

References

[1] F. Pollara. R. McEliece, and K. Abdel-Ghaffar, "Constructions for Finite State Codes," *TDA Progress Report 42-90*, vol. April–June 1987, Jet Propulsion Laboratory, Pasadena, California, pp. 42–49, August 15, 1987.

[2] J. Massey and M. Sain, "Inverse of Linear Sequential Circuit," *IEEE Trans. Comput.*, vol. C-17, 1968.

[3] K.-M. Cheung, "Error-Correction Coding for Data Storage Systems," PhD thesis, California Institute of Technology, 1987.

Table 1. Transfer function matrix to generate a completely connected state diagram with 8 states and 16 labels

$$G(D) = \begin{matrix} 1 & 0 & D & 1 \\ D & 1 & 0 & 0 \\ 0 & D & 1 & 0 \end{matrix} \qquad d_f = 2 \text{ branches}$$

Table 2. Transfer function matrix to generate a non-completely connected state diagram with 64 states and 16 labels

$$G(D) = \begin{matrix} 1+D & 0 & D^2 & 1 \\ D^2 & 1+D & 0 & 0 \\ 0 & D^2 & 1+D & 0 \end{matrix} \qquad d_f = 3 \text{ branches}$$