An efficient method to calculate the free distance of convolutional codes

RANJAN BOSE
Department of Electrical Engineering,
Indian Institute of Technology, Hauz Khas, New Delhi-110016,
INDIA.
Phone: +91-11-26591048

Abstract – We present a novel method to calculate the free distance (d_{free}) of a linear Trellis code (Convolutional code). The proposed method may be used to obtain the d_{free} of a given Convolutional code without having to compare all paths in the Trellis with the all-zero path. Using the proposed method, the d_{free} of a Convolutional code may be obtained merely by looking at the Trellis diagram of the code. The proposed method is based on the modified state diagram approach of finding d_{free}. Through several examples, we demonstrate that the proposed method may be used to calculate the d_{free} of large Convolutional codes with almost zero *online* effort.

Key-words - Trellis codes; Convolutional codes; free distance

1. Introduction

Trellis Codes [1] are popular for the error-correction capabilities they offer. Many practical communication systems use Trellis codes to safeguard data against channel imperfections. Unlike block encoders, which are memoryless and where the output codeword in a cycle has a one-to-one correspondence to the input information frame, Trellis encoders use memory so that the output codeword in a cycle depends not only on the current information frame but on previous information frames as well.

The linear, time-invariant subclass of Trellis codes is called Convolutional codes. Convolutional encoders typically consist of a shift register (of length s, called the constraint length) and a linear combinational circuit (involving XOR gates) [1], [2]. A typical Convolutional encoder is shown in fig. 1. The contents of the shift register at any time define the state of the encoder at that time.

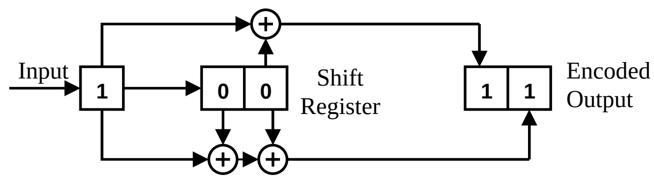

Figure 1: An example Convolutional encoder

At each encoder cycle, k bits of input information are pushed into the shift register to yield the next state.

Trellis codes are usually represented using a Trellis diagram. A Trellis diagram is a semi-infinite structure, which shows all possible state transitions that may occur in a Trellis code, going from one cycle to the other. The output codeword corresponding to each state transition is marked on the respective branch of the Trellis. Fig. 2 shows the Trellis diagram representation of the Trellis code defined by the encoder shown in fig. 1.

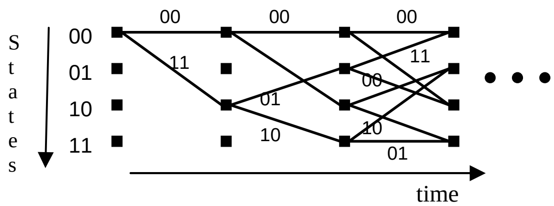

Figure 2: Trellis diagram corresponding to encoder of fig. 1

Note that if *s* is the constraint length of the system, 2^s gives the number of states in the Trellis. We call this Trellis a (*s*, *k*) Trellis.

Trellis codes are characterized by a parameter called the free distance, d_{free}. d_{free} is defined as the minimum Hamming distance between any two distinct code sequences. In other words, it is the minimum sum of Hamming distances between the codewords mapped on the branches of two paths in the Trellis, which first diverge and then merge back. d_{free} is a measure of the error-correcting capability of a Trellis code. A large d_{free} implies that the paths in the Trellis are far apart and hence, are easily distinguishable.

The calculation of d_{free} of a Trellis code involves a comparison between every possible pair of paths in the Trellis. However, for Convolutional codes, utilizing the linearity property, the d_{free} may be calculated by comparing every path with the all-zero path. (Note that the Trellis diagram of a Convolutional code always contains an all-zero path).

In this paper, we propose a technique using which, the d_{free} of a Convolutional code may be obtained just by looking at the Trellis diagram representation of the code. The proposed method is based on the modified state diagram approach [2] to calculate the d_{free} of a Convolutional code, without having to compare every path in the Trellis with the all-zero path. The proposed technique may be used to obtain the d_{free} of a Convolutional code with very little computational effort as compared to other methods. This technique is of high practical significance because Convolutional codes form a part of many practical communications systems and calculating d_{free} of these codes is a very commonly encountered problem.

The paper is structured as follows. Section I presents a brief introduction to the basic theory required to understand the rest of the paper. Section II explains the proposed idea and demonstrates it with a few examples. Section III summarizes the main ideas and concludes the paper.

2. The Proposed Method

The proposed method to calculate the d_{free} of a Convolutional code is based on an important observation that comes from the modified state diagram method of calculating the d_{free} of a Convolutional code. We shall first briefly discuss about the modified state diagram method and then present our method. Let us consider the Convolutional code, the Trellis diagram for which has been given in fig.3.

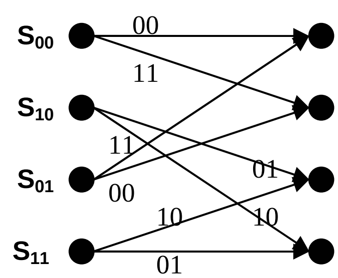

Figure 3: A unit of the Trellis diagram of an example Convolutional code

This Convolutional code may be alternatively represented by a modified state diagram, as shown in fig.4. In this diagram, the exponents of *D* correspond to the weights of the codewords which are given out on the respective state transitions. This representation is similar to the signal-flow graph of a simple single-input-single-output system. The transfer function for this system may be established as given by equation (1).

$$T(D) = \frac{X_{out}}{X_{in}} = \frac{D^5}{1 - D^1 - D^2} \tag{1}$$

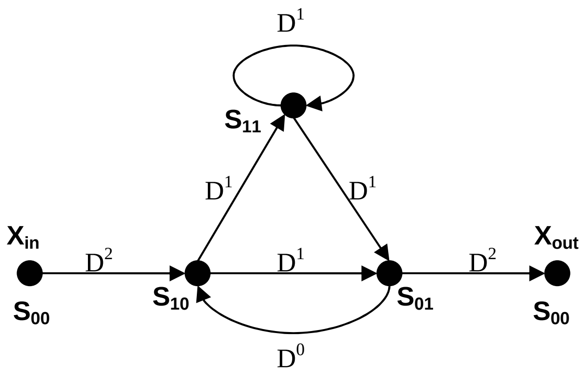

Figure 4: The modified state diagram corresponding to the code of fig. 3

Using the binomial expansion, the expression in equation (1) may be written as

$$T(D) = D^5\left[1 + \left(D^1 + D^2\right) + 2\left(D^1 + D^2\right)^2 + ...\right] \tag{2}$$

Clearly, by definition of d_{free}, the d_{free} of this Convolutional code corresponds to the smallest exponent of D in the expression given in equation (2). Hence, the d_{free} of this Convolutional code is 5.

Now consider the generalized version of the Trellis unit of fig.3. This is shown in fig. 5, where the codewords have been replaced by the variables w_1-w_8 that correspond to the weights of the codewords mapped on these branches.

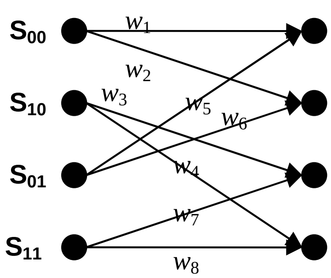

Figure 5: Generalized version of the Trellis unit of fig. 3

Deriving an expression for the transfer function of the single-input-single-output system thus formed, we get

$$T(D) = \frac{X_{out}}{X_{in}} = \frac{D^{(w_2+w_3+w_5)} - D^{(w_2+w_3+w_5+w_8)} + D^{(w_2+w_4+w_5+w_7)}}{1 - D^{w_8} - D^{(w_4+w_6+w_7)} - D^{(w_3+w_6)} + D^{(w_3+w_6+w_8)}} \tag{3}$$

Simplifying equation (3) using the binomial expansion as above and eliminating terms with large exponents, we get

$$\begin{aligned} d_{free} &= \min\{(w_2+w_3+w_5),\ (w_2+w_3+w_5+w_8),\ (w_2+w_4+w_5+w_7)\} \\ &= \min\{(w_2+w_3+w_5),\ (w_2+w_4+w_5+w_7)\} \\ &= w_2+w_5+\min\{w_3,\ (w_4+w_7)\} \end{aligned} \tag{4}$$

Equation (4) may be used to obtain the d_{free} of any (s=2, k=1) Convolutional code which has a Trellis connectivity as the one shown in fig. 5. It is important to note that only some specific branches of the Trellis are significant with respect to the d_{free} of the Convolutional code.

Similar expressions may be derived for Convolutional codes with other values of s and k. With these expressions in hand, the d_{free} of a given Convolutional code may be obtained just by replacing the weight terms with their values for the given case.

3. Examples

This section illustrates the strength of the proposed scheme with the help of examples. Let us consider the example Convolutional code represented by the Trellis diagram of fig. 3. Here, w_2=2, w_3=1 and w_5=2. Hence, d_{free} $=w_2+w_3+w_5 = 5$.

As another example, consider the Convolutional code given by the encoder shown in fig. 6.

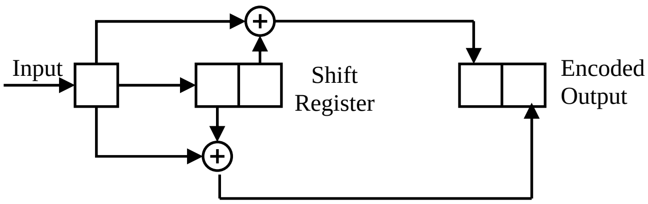

Figure 6: Another example of a Convolutional

The corresponding Trellis diagram is given in fig. 7.

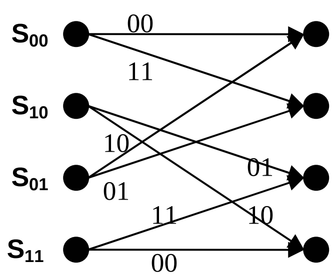

Figure 7: A unit of the Trellis diagram corresponding to encoder of fig. 6

For this example, w_2=2, w_3=1 and w_5=1. Evaluating equation (4) for this case gives d_{free} =4.

The examples given in this section illustrate that the proposed method may be used to obtain the d_{free} of a Convolutional code, just by looking at the Trellis diagram of the code.

4. Conclusions

We have proposed a method to obtain the d_{free} of a Convolutional code, with very little computational effort. Using the proposed approach, the d_{free} of a Convolutional code may be calculated just by looking at its Trellis diagram representation. Through the example of a (s=2, k=1) Trellis, we have demonstrated that equations for d_{free} may be derived in terms of the weights of the codewords mapped on the branches of the Trellis, for different values of s and k. The d_{free} of any given (s, k) Convolutional code may then be obtained simply by inserting the values of the weights in the corresponding equation for d_{free}. Clearly, using this approach, a comparison of all paths in the Trellis with the all-zero path is not required. Also, we need not solve the single-input-single-output system obtained using the modified state diagram approach each time we need to find the d_{free} of a Convolutional code.

References

[1] J. G. Proakis, *Digital Communications*, 3rd Edition, McGraw-Hill International Editions, Singapore, 1995.

[2] R. Bose, *Information Theory, Coding and Cryptography*, McGraw-Hill, Singapore, 2003.