On Equivalent Convolutional Encoders

Alexander E. Geyer and A. J. Han Vinck, *Senior Member IEEE*

***Abstract* – The $\tilde{D}$-description of a rate *R=k/n* convolutional encoder as a linear sequential circuit with *k* inputs and one output is proposed. We discuss the classical definition of equivalent codes (type 1 equivalence) and propose the definition of encoders that generate different codes but with the same code spectrum (type 2 equivalence). We conclude with some examples of encoder implementations that do not give the same code but the same code spectrum at lower encoder complexity (less memory elements).**

***Index terms* – Convolutional codes, algebraic structure of convolutional codes, equivalent encoder, minimal encoder, decoding complexity**

I. Introduction

Structural properties of convolutional codes and their generator matrices have been investigated in a series of papers and books [1-6]. One of the main problems considered is that of finding convolutional encoders that provide minimal complexity of trellis decoding for the same code properties. The "classical" D-polynomial generator matrix of a R = *k/n* convolutional code is given by

$$G(D)=\begin{pmatrix} G_1^1(D) & G_1^2(D) & \dots & G_1^n(D) \\ G_2^1(D) & G_2^2(D) & \dots & G_2^n(D) \\ \dots & \dots & \dots & \dots \\ G_k^1(D) & G_k^1(D) & \dots & G_k^n(D) \end{pmatrix}, \qquad (1)$$

where $G_j^i(D)$ are D-generator polynomials, that relate the information sequences $I^j(D)$ and code sequences $T_i(D)$, where $j=\overline{1,k}$ and $i=\overline{1,n}$, as

$$T_i(D)=\sum_{j=1}^{k} I^j(D)\cdot G_j^i(D). \qquad (2)$$

***Definition* 1**. A convolutional code generated by a R = *k/n* convolutional encoder with generator matrix G(D) over $F_2(D)$ is the set of output sequences $T_i(D)$.

The "base"-encoder can be realized with k shiftregisters and n modulo-2 adders. Note, that the output as given in (2) consists of n parallel output sequences. Before transmission, one may use a parallel to serial converter to have a serial transmission of the encoder output. In this contribution we investigate the properties of a *R=k/n* convolutional encoder as a linear sequential circuit with *k* inputs and one output. Note that we have to realize, that the timing of input symbols and output symbols must be consistent. To achieve a consistent desciption of the input/output relation we define the following sequences:

$$R(\tilde{D}) = 1 + \tilde{D} + \dots + \tilde{D}^{n-1} \quad ;$$

$$I^j(\tilde{D}^n) = i_0^j + i_1^j \tilde{D}^n + i_2^j \tilde{D}^{2n} + \cdots ;$$

$$G_j(\tilde{D}^n) = G_j^1(\tilde{D}^n) + \tilde{D} G_j^2(\tilde{D}^n) + \cdots \tilde{D}^{n-1} G_j^n(\tilde{D}^n) \ ;$$

$$I^j(\tilde{D}) = R(\tilde{D})\, I^j(\tilde{D}^n); \qquad G_j(\tilde{D}) = G_j(\tilde{D}^n) / R(\tilde{D}).$$

The k-input/1-output relation can now be formalized as

$$T(\tilde{D}) = \sum_{j=1}^{k} I^j(\tilde{D})\cdot G_j(\tilde{D}) = \sum_{j=1}^{k} I^j(\tilde{D}^n)\cdot G_j(\tilde{D}^n) ;$$

$$G(\tilde{D}) = \begin{pmatrix} G_1(\tilde{D}) \\ G_2(\tilde{D}) \\ \dots \\ G_k(\tilde{D}) \end{pmatrix}.$$

***Definition* 2.** A convolutional code generated by a R = *k/n* convolutional encoder with generator matrix $G(\tilde{D})$ is the set of output sequences $T(\tilde{D}) = \sum_{j=1}^{k} I^j(\tilde{D})\cdot G_j(\tilde{D})$.

The obvious realization of the encoding procedure is given in Figure 1. The adjoint obvious realization has only one shift register and one output, see also [2,6].

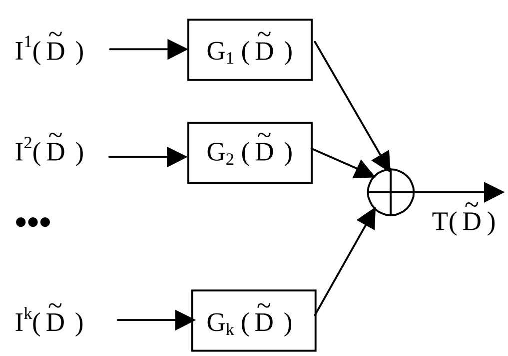

Fig. 1. The k-input, 1-output encoding circuit

This work was supported in part by German Science Foundation DFG.
Geyer A. E. is with the Ukrainian State Academy of Telecommunication, Odessa, 270021, Ukraine (e-mail: geyer@usat.ukrtel.net)
A. J. Han Vinck is with the Institute of Experimental Mathematics, Essen University, 45326 Essen, Germany (e-mail: vinck@exp-math.uni-essen.de).

For mathematicians, the D description is not different from the $\tilde{D}$-description. However, the $\tilde{D}$-presentation is useful if one tries to realize the encoding circuit. This can be seen in the timing diagram of Fig. 2 for n = 3.

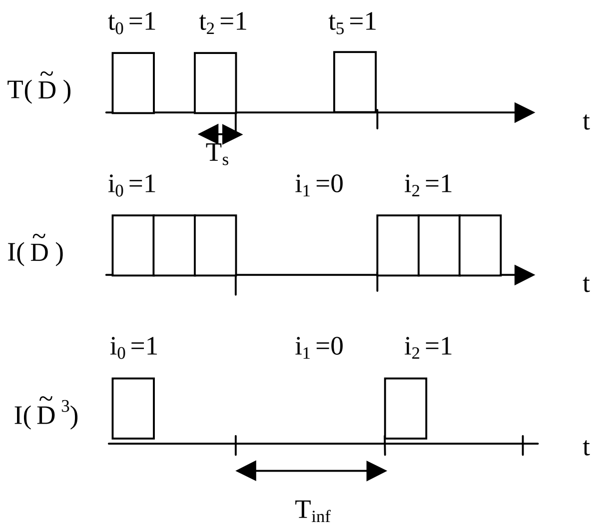

Fig. 2. Input symbols in the $\tilde{D}$ - description.

The duration of a code symbol $T_s = T_{inf} / n$.

***Example* 1**. Consider a one-input/one-output R = 1/2 encoder with D-generator matrix of the base encoder

$$G(D) = (\ G^1(D), G^2(D)\ ) = (\ 1 + D + D^2,\ 1 + D^2\ ).$$

According to the above definition

$$G(\tilde{D}) = (1 + \tilde{D} + \tilde{D}^2 + \tilde{D}^4 + \tilde{D}^5)/(1 + \tilde{D}),$$

$$T(\tilde{D}) = I(\tilde{D})\, G(\tilde{D}).$$

A particular implementation of the encoder is given in Fig. 3.

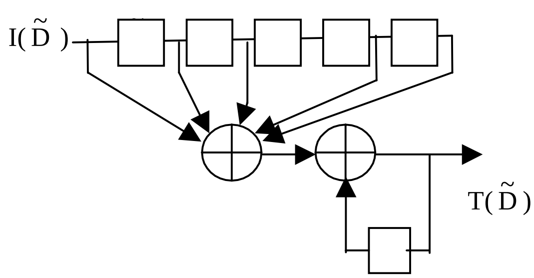

Fig.3. Encoder for the example 1 code.

***Example* 2.** Consider a R = 2/4 two-input/one-output encoder with a D-generator matrix of the base encoder

$$G(D) = \begin{pmatrix} D^2 & D^2 & 1 & 0 \\ 0 & D & 0 & 1 \end{pmatrix}.$$

From this we obtain

$$G_1(\tilde{D}) = \frac{\tilde{D}^2 + \tilde{D}^8 + \tilde{D}^9}{1 + \tilde{D} + \tilde{D}^2 + \tilde{D}^3}; \quad G_2(\tilde{D}) = \frac{\tilde{D}^3 + \tilde{D}^5}{1 + \tilde{D} + \tilde{D}^2 + \tilde{D}^3}.$$

Hence the $\tilde{D}$ - generator matrix of this encoder is

$$G(\tilde{D}) = \frac{1}{1 + \tilde{D} + \tilde{D}^2 + \tilde{D}^3} \begin{pmatrix} \tilde{D}^2 + \tilde{D}^8 + \tilde{D}^9 \\ \tilde{D}^3 + \tilde{D}^5 \end{pmatrix}. \qquad (3)$$

One can see that the matrix (3) gives rise to a delay of 2 time instants in its output sequence. We may take out the delay and have a simpler implementation of the $\tilde{D}$-encoder. The question is what kind of effect this then has on the description of a matrix in the D-notation. This is topic of the next chapter, where we consider the equivalence between encoder matrices. The goal is to obtain an encoder that leads to minimum decoding complexity, i.e. the encoder with the minimum number of memory elements in its base realization.

II. Code and Code spectrum equivalence

The goal of our investigation is to find the D-generator matrices of convolutional encoders with the smallest number of memory elements.

***Definition* 3.** Two convolutional encoders are called D-equivalent if they encode the same code (type 1), [2].

For D-equivalent encoders, the encoding matrices

$$G'(D) = A(D)G(D);\ A^{-1}(D)\,A(D) = D^s I_k;\ s \geq 0,$$

where A(D) and $A^{-1}(D)$ are matrices over $F_2(D)$. Obviously, G(D) and G'(D) generate the same set of output sequences.

***Definition* 4.** If the generator matrix G(D) isn't D-delay-free, it can be written as $G(D)=D^i G_d(D)$, where $i \geq 1$ and the generator matrix $G_d(D)$ is D-delay-free, [4].

Remark. We change the terms an "equivalent" encoder and a "delay-free" matrix used in [1-6] by a "D-equivalent" encoder and a D-delay-free matrix only for showing the difference with the introduced $\tilde{D}$-equivalent encoder and $\tilde{D}$-delay-free matrix.

***Definition* 5.** Convolutional encoders with generator polynomial matrices $G^1(\tilde{D})$ and $G^2(\tilde{D})$ are called $\tilde{D}$-equivalent if they encode the same code.

***Theorem* 1.** Two rate R= *k/n* convolutional generator matrices $G(\tilde{D})$ and $G'(\tilde{D})$ are $\tilde{D}$-equivalent if and only if there is k×k non-singular matrix $A(\tilde{D}^n)$ over $F_2(\tilde{D})$ such that

$$G(\tilde{D}) = \tilde{D}^s \cdot A(\tilde{D}^n) \cdot G'(\tilde{D})\,;\ \ s \geq 0.$$

***Definition* 6.** A base convolutional encoder with generator matrix $G'(D)$ is said to be minimal if it has a realization of the D-generator matrix with the smallest number of memory elements among all possible $\tilde{D}$-equivalent encoders of the same code.

***Definition* 7.** If the $\tilde{D}$-generator matrix $G(\tilde{D})$ is not $\tilde{D}$-delay-free, it can be written as $G(\tilde{D}) = \tilde{D}^i G_d(\tilde{D})$, where $i \geq 1$ and the $\tilde{D}$-generator matrix $G_d(\tilde{D})$ is $\tilde{D}$-delay-free.

Example 3. In accordance with definition 3, two encoders with the *D*-generator matrices

$$G_1(D) = (\ 1,\ 1+D\ ),\ G_2(D) = (\ D+D^2,\ 1\ ) \qquad (4)$$

are not equivalent. The $\tilde{D}$-generator polynomials of these codes are

$$G_1(\tilde{D}) = \frac{1}{1+\tilde{D}}(\ 1+\tilde{D}+\tilde{D}^3\ )\ , G_2(\tilde{D}) = \frac{\tilde{D}}{1+\tilde{D}}(\ 1+\tilde{D}+\tilde{D}^3\ )\ . \quad (5)$$

It is obvious, that the information sequences $I_1(\tilde{D}) = \tilde{D}\, I_2(\tilde{D})$ encode the same code, because $T^2(\tilde{D}) = \tilde{D}\, T^1(\tilde{D})$, i.e. the codes (4) are $\tilde{D}$-equivalent in accordance to our definition 5, as the matrix $G_2(\tilde{D})$ is not $\tilde{D}$- delay-free.

In Fig. 4 we illustrate the difference between the delay in the D- and the $\tilde{D}$-description. For the D-decription the delay is defined over code blocks of length n, whereas for the $\tilde{D}$-decription the delay is defined over output digits.

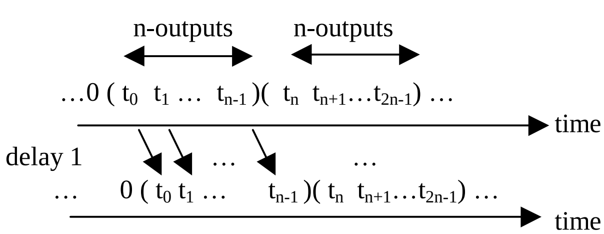

Fig. 4 Delay in the $\tilde{D}$-description

To see the relation between the $\tilde{D}$- and the D-description we further define the nxn matrix

$$S_1 = \begin{bmatrix} 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & & 1 \\ D & 0 & 0 & & 0 \end{bmatrix};\ S_2 = \begin{bmatrix} 0 & 0 & 1 & & 0 \\ 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ D & 0 & 0 & & 1 \\ 0 & D & 0 & & 0 \end{bmatrix}.$$

If we look at the product $G(D)S_1$, we obtain an encoder matrix that generates code sequences that cannot be generated by a D-equivalent matrix, but the generated code has exactly the same code spectrum as the starting matrix G(D). The product

$$G(D)S_1 = \begin{pmatrix} DG_1^n(D) & G_1^1(D) & G_1^2(D) & \dots & G_1^{n-1}(D) \\ DG_2^n(D) & G_2^1(D) & G_2^2(D) & \dots & G_2^{n-1}(D) \\ \dots & & \dots & \dots & \dots \\ DG_k^n(D) & G_k^1(D) & G_k^1(D) & \dots & G_k^{n-1}(D) \end{pmatrix},$$

which is equivalent to a shift of the output sequence over one position. The same is true for the product $G(D)[S_1]^s$. Hence, if we multiply G(D) s times with S_1, we shift the output sequence over s positions. This is in agreement with theorem 1 and definition 7. Other operations on the output that cannot be achieved with definition 3 of code equivalence is that of column permutation and column delay. For n = 3 examples of the operations are given below:

$$S_1 = \begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ D & 0 & 0 \end{matrix};\qquad S_2 = \begin{matrix} 0 & 0 & 1 \\ D & 0 & 0 \\ 0 & D & 0 \end{matrix};$$

$$\text{delay}\ \begin{matrix} 1 & 0 & 0 \\ 0 & D^i & 0 \\ 0 & 0 & 1 \end{matrix};\qquad \text{permute}\ \begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{matrix}.$$

We therefore introduce the following natural definition.

Definition 8. Two convolutional encoders are called spectrum equivalent if their generated codes have the same spectrum (type 2).

Example 4. The code

$$G'(D) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \text{ and } G(D) = \begin{pmatrix} D^2 & D^2 & 1 & 0 \\ 0 & D & 0 & 1 \end{pmatrix}$$

are spectrum equivalent, since

$$\begin{pmatrix}1 & 0\\ 0 & D\end{pmatrix}\begin{pmatrix}D^2 & D^2 & 1 & 0\\ 0 & D & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & D^2 & 0\\ 0 & 0 & 0 & D\end{pmatrix}=$$

$$= D^2\begin{pmatrix}1 & 1 & 1 & 0\\ 0 & 1 & 0 & 1\end{pmatrix}.$$

Note that G(D) is basic and minimal according to the classical definition. The encoder realization takes 3 memory elements. However, by row and column operations that do not change the code spectrum we obtain G'(D) which is memoryless. Note that

$$G(\tilde{D}) = \tilde{D}^2\left(\frac{1}{1+\tilde{D}+\tilde{D}^2+\tilde{D}^3}\begin{pmatrix}1+\tilde{D}^6+\tilde{D}^7\\ \tilde{D}+\tilde{D}^3\end{pmatrix}\right). \qquad (6)$$

The $\tilde{D}$-delay-free $\tilde{D}$-generator matrix in round brackets is the matrix of the encoder, which is $\tilde{D}$-equivalent to the encoder with the matrix (6). The D-generator matrix of the $\tilde{D}$-equivalent base encoder is

$$G(D) = \begin{pmatrix}1 & 0 & D & D\\ 0 & 1 & 0 & 1\end{pmatrix}. \qquad (7)$$

For G'(D), the $\tilde{D}$-equivalent is given by

$$G(\tilde{D}) = \left(\frac{1}{1+\tilde{D}+\tilde{D}^2+\tilde{D}^3}\begin{pmatrix}1+\tilde{D}+\tilde{D}^2\\ \tilde{D}+\tilde{D}^3\end{pmatrix}\right).$$

Example 5. Consider a two input and one output R = 2/4 encoder with the D-generator matrix of base encoder

$$G(D) = \begin{pmatrix}1 & 1 & 1 & 0\\ D^2 & D & 0 & 1\end{pmatrix}. \qquad (8)$$

For this matrix

$$\begin{pmatrix}D & 0\\ 0 & 1\end{pmatrix}G(D)S_1=\begin{pmatrix}0 & D & D & D\\ D & D^2 & D & 0\end{pmatrix}=D\begin{pmatrix}0 & 1 & 1 & 1\\ 1 & D & 1 & 0\end{pmatrix}.$$

Hence we have the $\tilde{D}$-delay-free, $\tilde{D}$-equivalent matrix

$$G(\tilde{D}) = \frac{1}{1+\tilde{D}+\tilde{D}^2+\tilde{D}^3}\begin{pmatrix}\tilde{D}+\tilde{D}^2+\tilde{D}^3\\ 1+\tilde{D}^2+\tilde{D}^5\end{pmatrix}.$$

The decoding complexity of such an encoder is two times smaller than that of the minimal encoder with matrix (8).

***Example* 6.** Consider a R = 2/4 two inputs and one output encoder with the D-generator matrix of base encoder

$$G(D) = \begin{bmatrix}D & D & 0 & 1\\ 1+D & 1 & 1 & 0\end{bmatrix};\ G'(D) = \begin{bmatrix}1 & 1 & 0 & 1\\ 1+D & 1 & 1 & 0\end{bmatrix}.$$

The encoder G'(D) has the same code spectrum since

$$\begin{pmatrix}D & 0\\ 0 & 1\end{pmatrix}G'(D)\begin{pmatrix}D & 0 & 0 & 0\\ 0 & D & 0 & 0\\ 0 & 0 & D & 0\\ 0 & 0 & 0 & 1\end{pmatrix}=D\begin{bmatrix}D & D & 0 & 1\\ 1+D & 1 & 1 & 0\end{bmatrix}.$$

The trellis complexity connected with an encoder G'(D) is two times smaller than that connected with the minimal encoder with matrix G(D).

III. Conclusion

On the basis of the proposed $\tilde{D}$-description of a rate $R=k/n$ convolutional encoder as a linear sequential circuit with k inputs and one output the $\tilde{D}$-equivalent convolutional encoders with k inputs and one output are considered. The introduced description leads to less complex encoders. Furthermore, we introduce spectrum equivalent encoders. Both concepts cannot be defined in the framework of classical algebraic theory of convolutional codes.

REFERENCES

[1] J. L. Massey and M. K. Sain, "Inverses of linear sequential circuits," *IEEE Trans. Comput. Theory*, pp. 330-337, vol. 17, November 1968.

[2] G. D. Forney, "Convolutional Codes I: algebraic structure," *IEEE Trans. Inform. Theory*, pp. 720-738, vol. 16, November 1970.

[3] P. Piret, Convolutional Codes: An Algebraic Approach, MIT Press, Cambridge, MA, 1988.

[4] R. Johannesson and Z. Wan, "A linear algebra approach to minimal convolutional encoders," *IEEE Trans. Inform. Theory*, pp. 1219–1233, vol. 39, July 1993.

[5] G. D. Forney, R. Johannesson and Z. Wan, "Minimal and canonical rational generator matrices for convolutional codes," *IEEE Trans. Inform. Theory*, pp. 1865-1880, vol. 42, November 1996.

[6] R. Johannesson and K. Sh. Zigangirov, Fundamentals of Convolutional Coding, New York IEEE Press, 1999.