The Extended Invariant Factor Algorithm with Application to the Forney Analysis of Convolutional Codes

Robert J. McEliece and Ivan Onyszchuk
California Institute of Technology
Pasadena, California 91125, USA

Summary.

In his celebrated paper on the algebraic structure of convolutional codes, Forney [1] showed that by using the invariant-factor theorem, one can transform an arbitrary polynomial generator matrix for an (n,k) convolutional code C into a basic (and ultimately a minimal) generator matrix for C. He also showed how to find a polynomial inverse for a basic generator matrix for C, and a basic generator matrix for the dual code $C^\perp$. In this paper, we will discuss efficient ways to do all these things. Our main tool is the "entended invariant factor algorithm," which we introduce here.

1. The Extended Invariant Factor Algorithm.

The goal of the *invariant factor algorithm* (see e.g. [2, Sec. 6.2.4], [3, Sec. 6.3.3], or [4, Sec. 12.2]) is to take an arbitrary $k \times n$ matrix G (with $k \le n$) over a Euclidean domain R, and by a sequence of elementary row and column operations, to reduce G to a $k \times n$ diagonal matrix $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_k)$, whose diagonal entries are the invariant factors of G, i.e., $g_i = \Delta_i/\Delta_{i-1}$, where Δ_i is the gcd of the $i \times i$ minors of G. The goal of the *extended invariant factor algorithm*, which we introduce in this paper, is to take the same input, and not only find Γ, but also to find a $k \times k$ unimodular matrix X, and an $n \times n$ unimodular matrix Y, such that $XGY = \Gamma$.

To describe the extended invariant factor algorithm, we need to take a closer look at the original invariant factor algorithm. Formally, it can be described as follows. Beginning with the matrix $G_0 = G$, it produces a sequence of $k \times n$ matrices G_i, where G_{i+1} is derived from G_i by either an elementary row operation or an elementary column operation. We can represent this algebraically as

$$G_{i+1} = E_{i+1} G_i F_{i+1}, \tag{1.1}$$

where E_{i+1} and F_{i+1} are $k \times k$ and $n \times n$ elementary matrices, respectively. If G_{i+1} is obtained from G_i via a row operation, then $F_{i+1} = I_n$, but if G_{i+1} is obtained from G_i via a column operation, then $E_{i+1} = I_k$. After a finite number N of steps, we obtain $G_N = \Gamma$. (The details of which elementary row and column operations to perform, and in which order, are of central importance, of course, but for reasons of space, we refer the reader to [2, Sec. 6.2.4], or [3, Section 6.3.3] for them)

The extended invariant factor algorithm builds on the invariant factor algorithm. In addition to the sequence $G_0, G_1, \ldots, G_N$, the extended invariant factor algorithm also works with a sequence of unimodular $k \times k$ matrices $X_0, \ldots, X_N$, and a sequence of unimodular $n \times n$ matrices $Y_0, \ldots, Y_N$. The sequences (X_i) and (Y_i) are initialized as $X_0 = I_k$, $Y_0 = I_n$, and updated via the rule (cf. Eq.(1.1))

$$X_{i+1} = E_{i+1} X_i \tag{1.2}$$
$$Y_{i+1} = Y_i F_{i+1}. \tag{1.3}$$

It is a simple matter to prove by induction that

$$X_i G Y_i = G_i \qquad \text{for } i = 0, 1, \ldots, N, \tag{1.4}$$

so that specializing (1.4) with $i = N$, we have

$$X_N G Y_N = \Gamma, \tag{1.5}$$

which is the desired "invariant-factor" diagonalization of G. A rough analysis of this algorithm shows that it requires $O(dnk^2)$ polynomial divisions, or $O(d^3nk^2)$ field operations (addition, subtraction, multiplication, or division in F), where d denotes the maximum degree of any polynomial in G.

2. Application to the Analysis of Convolutional Codes.

We define an (n,k) convolutional code C over a field F to be a k-dimensional subspace of $F(D)^n$, where $F(D)$ is the field of rational functions in the indeterminate D over F. A *generator matrix* for C is a $k \times n$ matrix with entries in $F(D)$ whose rows form a basis for C. Given an arbitrary generator matrix G for C, we can easily transform G to a generator matrix with polynomial entries by multiplying the ith row of G by the lcm of the denominators of its components. In this section, we will see how the extended invariant factor algorithm introduced in Section 1 can be used to transform an arbitrary *polynomial* generator matrix for C into a *basic* generator matrix for C. (The transition from a basic to a minimal generator can, if desired, then be done by the simple algorithm originally described in [1], or perhaps more lucidly in Kailath [3, Sec. 6.3.2], where the process is described as "row-reducing" a polynomial matrix). We will see that the extended invariant factor algorithm also produces, more or less for free, a polynomial inverse for the basic generator matrix, and a basic generator matrix for the dual code $C^\perp$.

Assume then that G is a $k \times n$ polynomial generator matrix for a convolutional code C over a field F. Since the ring of polynomials over F is a Euclidean domain, we may apply the extended invariant factor algorithm described in Section 1, thereby obtaining a decomposition of the form (1.5). In what follows, the matrices X_N and Y_N produced by the extended invariant factor algorithm will be denoted simply by X and Y.

The matrices, X, Y, and Γ, contain much valuable information about the code C and the generator matrix G. To extract this information, however, we need to define several useful "pieces" of these matrices, which we call Γ_k, Γ'_k, K, and H:

$$\Gamma_k = \text{leftmost } k \text{ columns of } \Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_k). \tag{2.1}$$
$$\Gamma'_k = \gamma_k \cdot \Gamma_k^{-1} = \mathrm{diag}(\gamma_k/\gamma_1, \ldots, \gamma_k/\gamma_k). \tag{2.2}$$
$$K^T = \text{leftmost } k \text{ columns of } Y. \tag{2.3}$$
$$H^T = \text{rightmost } n-k \text{ columns of } Y. \tag{2.4}$$

Here then are useful "outputs" of the extended invariant factor algorithm, when applied to G.

- *A basic generator matrix for C:* $G_{\text{basic}} = \Gamma_k^{-1} XG$. (That is, G_{basic} is obtained by dividing the ith row of XG by the invariant factor γ_i, for $i = 1, \ldots, k$.)
- *A polynomial inverse for G_{basic} :* K^T
- *A polynomial pseudo-inverse for G, with factor γ_k:* $K^T \Gamma'_k X$.
- *A basic generator matrix for $C^\perp$, i.e., parity-check matrix for C:* H.

References.

[1] Forney, G. D., "Convolutional Codes I: Algebraic Structure." *IEEE Trans. Inform. Theory vol. IT-16 (November 1970), pp. 720–738.*

[2] Gantmacher, F. R., *The Theory of Matrices, vol. I.* New York: Chelsea Publishing Co., 1977.

[3] Kailath, T. *Linear Systems.* Englewood Cliffs, N. J.: Prentice-Hall, 1980.

[4] van der Waerden, B. L., *Algebra, vol. 2.* New York: Frederick Ungar, 1970.

Acknowledgements.

The contibution of Ivan Onyszchuk, and a portion of the contribution of Robert J. McEliece, to this paper, was carried out at Caltech's Jet Populsion Laboratory, under contract with the National Aeronautics and Space Administration. A portion of McEliece's contribution was also carried out at Caltech's Electrical Engineering department, and supported by AFOSR grant no. 91-0037