EE 229B ERROR CONTROL CODING **Spring 2005**

Lecture notes on the structure of convolutional codes

Venkat Anantharam

(based on scribe notes by Lawrence Ip and Xiaoyi Tang)

Warning : Use at your own risk ! These notes have not been sufficiently carefully screened.

1 Convolutional Codes

1.1 Introduction

Suggested reference for convolutional codes : *"Fundamentals of Convolutional Coding"* by Rolf Johannesson and Kamil Zigangirov, IEEE Press, 1999.

An encoder for a binary block code takes a block of information bits and converts it into a block of transmitted bits (a codeword). A binary convolutional encoder takes a stream of information bits and converts it into a stream of transmitted bits, using a shift register bank. Redundancy for recovery from channel errors is provided by transmitting more bits per unit time than the number of information bits per unit time. Maximum likelihood decoding can be done using the *Viterbi algorithm*; other decoding algorithms such as SOVA (soft output Viterbi algorithm) and the BCJR algorithm are also commonly used. In practice the information stream is of finite duration and one typically appends a few termination bits to the input stream to bring the shift register bank back to the all zeros state, so that the convolutional code is in effect used as a very long block code.

Often convolutional codes are used as inner codes with burst error correcting block codes as outer codes to form *concatenated codes*. Errors in Viterbi-like decoding algorithms for convolutional codes tend to occur in bursts because they result from taking a wrong path in a trellis. The burst error correcting capability of the outer code is used to recover from such burst error patterns in the decoding of the inner code. See Section 15.6 on pp. 760 -761 of the text of Lin and Costello (2nd edition) for a short discussion of this idea.

1.2 Formal definitions

Convolutional codes are linear codes over the field of one-sided infinite sequences. The symbols can be from any field but we will just consider symbols from $GF(2)$. For notational simplicity, and to give an indication of how the more general case works, in the following we will write F for the field $GF(2)$. We begin with a few definitions.

Definition 1 *Let*

$$F((D)) = \left\{ \sum_{i=r}^{\infty} x_i D^i : x_i \in F, r \in Z \right\}$$

be the set of binary Laurent series*.*

The D is a formal place holder to represent one step of delay. With addition and multiplication defined in the obvious way $F((D))$ becomes a field. Because the Laurent series are finite to the left, multiplication is well defined (as then we don't have to sum an infinite number of terms). All the axioms of a field are easy to check except the existence of a multiplicative inverse. A multiplicative inverse can be constructed explicitly by "long division".

Definition 2 *Let*

$$F[[D]] = \left\{ \sum_{i=r}^{\infty} x_i D^i : x_i \in F, r \geq 0 \right\}$$

be the set of binary formal power series*.*

With addition and multiplication defined in the obvious way $F[[D]]$ is a ring (in a ring, as opposed to a field, we do not require the existence of a multiplicative identity and we do not requrie the existence of multiplicative inverses for nonzero elements).

Definition 3 *Let*

$$F[D] = \left\{ \sum_{i=r}^{d} x_i D^i : x_i \in F, 0 \leq r \leq d < \infty \right\}$$

be the set of binary formal polynomials*.*

With addition and multiplication defined in the obvious way $F[D]$ is a ring.

Definition 4 *Let*

$$F(D) = \left\{ \frac{p(D)}{q(D)} : p(D), q(D) \in F[D] \right\}$$

be the set of binary rational functions*.*

With addition and multiplication defined in the obvious way $F(D)$ is a field.

Note that we have

$$F[D] \subset F(D) \subset F((D))$$

and

$$F[D] \subset F[[D]] \subset F((D)).$$

Definition 5 *Let*

$$F^k((D)) = \left\{ (x_1(D), x_2(D), \cdots, x_k(D)) : x_j(D) \in F((D)), 1 \leq j \leq k \right\}.$$

Definition 6 *A rate $R = k/n$ ($k \leq n$)* convolutional mapping *is a linear transformation*

$$\tau : F^k((D)) \rightarrow F^n((D))$$

defined by

$$\tau(x(D)) = x(D)G(D)$$

where $G(D) \in F(D)^{k \times n}$ ($k \times n$ matrices of rational functions) and $G(D)$ has rank k. $G(D)$ is called a transfer function*.*

Definition 7 *A rate $R = k/n$* convolutional code *is the image of a rate R convolution mapping.*

Given a $k \times n$ transfer function $G(D)$ and $x(D) \in F^k((D))$, then $y(D) = x(D)G(D) \in F^n((D))$ is the *code sequence* corresponding to the *input sequence* $x(D)$.

Definition 8 *Given $x(D) \in F((D))$, its* delay *$del(x(D))$ is the smallest r for which $x_r = 1$.*

Definition 9 *$x(D)$ is called* delay free *if $del(x(D)) = 0$.*

Definition 10 *A rational function $p(D)/q(D)$ is called* realizable *if $q(D)$ is delay free.*

Definition 11 *$G(D) \in F(D)^{k \times n}$ is called* realizable *if each of its elements is realizable. Such a $G(D)$ is called a* generator matrix*.*

Definition 12 *A generator matrix (a transfer function matrix that is realizable) is called* delay free *if at least one of its elements is delay free.*

Theorem 1 *Any convolutional code $\mathcal{C} \subseteq F^n((D))$ is the image of a convolutional mapping with a transfer function matrix that is a delay free generator matrix.*

Proof: Suppose $\mathcal{C}$ is the image of $F^k((D))$ under the mapping corresponding to $G(D) \in F(D)^{k \times n}$. Write

$$G(D) = [g_{ij}(D)]^{k \times n}$$

where $g_{ij}(D) = p_{ij}(D)/q_{ij}(D)$. Now write

$$g_{ij}(D) = D^{s_{ij}} \tilde{p}_{ij}(D)/\tilde{q}_{ij}(D)$$

where $\tilde{p}_{ij}(0) = 1, \tilde{q}_{ij}(0) = 1$. Define $s = \min_{i,j} s_{ij}$ and

$$\hat{G}(D) = D^{-s} G(D).$$

Then $\hat{G}(D)$ is a delay free generator and the image of $F^k((D))$ under the transformation corresponding to $\hat{G}(D)$ is $\mathcal{C}$. □

In general a transfer function matrix may not be realizable, so any shift register bank that implements it would need to be noncausal. The preceding result shows that any convolutional code has a generator matrix (a realizable transfer function matrix), which can now be built with a causal shift register bank. Further we can ensure that there is no "unnecessary delay" (this the content of the delay free condition). The following result shows that we can actually even use any convolutional code with feedforward shift register banks (i.e. no need for feedback, or, in other words, a polynomial transfer function matrix).

Theorem 2 *Any rate $R = k/n$ convolutional code $\mathcal{C}$ is the image of $F^n((D))$ under a transfer function matrix that is*

- *a generator*
- *delay free*
- *polynomial*

Proof: We already know from the previous theorem that $\mathcal{C}$ is the image of $F^k((D))$ under a $G(D) \in F(D)^{k \times n}$ that is realizable and delay free. Write

$$G(D) = [g_{ij}(D)]^{k \times n}$$

where $g_{ij}(D) = p_{ij}(D)/q_{ij}(D)$, $q_{ij}(0) = 1, 1 \leq i \leq k, 1 \leq j \leq n$, and there is at least one (i, j) with $p_{ij}(0) = 1$. Let $q(D) = \mathrm{lcm}(\{q_{ij}(D) : 1 \leq i \leq k, 1 \leq j \leq n\})$. Let

$$\hat{G}(D) = q(D)G(D).$$

Then $\mathcal{C}$ is the image of $F^k((D))$ under $\hat{G}(D)$. $\hat{G}(D)$ is a polynomial matrix and thus realizable. Since all the $q_{ij}(0) = 1$, $q(0) = 1$. So for the same (i, j) for which $p_{ij}(0) = 1$ will have $q(0)p_{ij}(0)/q_{ij}(0) = 1$ making $\hat{G}(D)$ delay free.

Definition 13 *A rate $R = k/n$ convolutional mapping is said to be* systematic *if some k of the code sequences are exactly the k input sequences. Equivalently, after reordering the output coordinates the corresponding transfer function matrix $G(D)$ has the form*

$$G(D) = [I_{k \times k} R(D)],$$

where $R(D) \in F(D)^{k \times n-k}$.

Every convolutional code has both systematic and non-systematic convolutional mappings that result in the same code.

In block codes errors cannot propagate very far because of finite size blocks. This is not necessarily the case for convolutional codes as it is possible for a “finite” error in the code sequence to have an “infinite” error in the corresponding input sequence.

Definition 14 *A convolutional mapping is said to be* catastrophic *if there is some code sequence $y(D)$ with finitely many 1s that results from an input sequence $x(D)$ with infinitely many 1s.*

Every code has catastrophic and non-catastrophic mappings that result in that code.

Several examples relating to the definitions were discussed in class. See also the text of Lin and Costello and the book of Johannesson and Zigangirov.

2 Smith form of a polynomial matrix

$G(D)$, a $k \times n$ polynomial matrix, can be written as

$$G(D) = A(D)\Gamma(D)B(D)$$

where $A(D)$ is an unimodular $k \times k$ polynomial matrix (an unimodular matrix is a polynomial matrix with a polynomial inverse), $B(D)$ is an unimodular $n \times n$ matrix, and

$$\Gamma(D) = \begin{bmatrix} \gamma_1(D) & & & \\ & \ddots & & \mathbf{0} \\ & & \gamma_r(D) & \\ & \mathbf{0} & & \mathbf{0} \end{bmatrix}, \qquad r = \text{rank } G(D)$$

with $\gamma_i(D) \,|\, \gamma_{i+1}(D), \;\; 0 \le i \le r-1$. In fact

$$\gamma_i(D) = \frac{\Delta_i(D)}{\Delta_{i-1}(D)}$$

where $\Delta_0(D) = 1$ and $\Delta_i(D) =$ gcd of the $i \times i$ minors of $G(D)$.

This was proved in class. See also the book of Johannesson and Zigangirov.

3 Smith form of a rational matrix $G(D) \in F(D)^{k \times n}$

Suppose $q(D)$ is the lcm of the denominator polynomials in the entries of $G(D)$. Then $q(D)G(D)$ is a polynomial matrix. So

$$q(D)G(D) = A(D)\hat{\Gamma}(D)B(D) \quad \text{(Smith Form)}$$

Therefore,

$$G(D) = A(D)\Gamma(D)B(D)$$

where $A(D)$ and $B(D)$ are unimodular with size $k \times k$ and $n \times n$, respectively, and

$$\Gamma(D) = \begin{bmatrix} \frac{\alpha_1(D)}{\beta_1(D)} & & & \\ & \ddots & & \mathbf{0} \\ & & \frac{\alpha_r(D)}{\beta_r(D)} & \\ & \mathbf{0} & & \mathbf{0} \end{bmatrix}$$

where $\alpha_i(D) \mid \alpha_{i+1}(D), \;\; 0 \le i \le r-1$ and $\beta_{i+1}(D) \mid \beta_i(D), \;\; 0 \le i \le r-1$.

4 Massey-Sain characterization of non-catastrophic generator matrices

Theorem 1 *A rational generator matrix $G(D)$ for a convolutional code C is non-catastrophic (Recall $G(D)$ $k \times n$ has rank k and $k \le n$) iff $\alpha_k(D) = D^s$ for some $s \ge 0$.*

Proof: Let $G(D) = A(D)\Gamma(D)B(D)$ (Smith form) and

$$\Gamma(D) = \left[\;\text{diag}\left(\tfrac{\alpha_i(D)}{\beta_i(D)}\right), \; 1 \le i \le k, \;\; \mathbf{0} \;\right]$$

Assume $\alpha_k(D) = D^s$. A right inverse for $G(D)$ is given by

$$G(D)^{-1} = B^{-1}(D) \begin{bmatrix} \text{diag}\left(\tfrac{\beta_i(D)}{\alpha_i(D)}\right), \;\; 1 \le i \le k \\ \mathbf{0} \end{bmatrix} A^{-1}(D)$$

Since $\alpha_1(D) \mid \alpha_2(D) \mid \ldots \mid \alpha_k(D) = D^s$, each $\alpha_i(D) = D^{s_i}$ for for some $s_1 \le s_2 \le \ldots \le s_k = s$. So

$$D^s G(D)^{-1} = B^{-1}(D) \begin{bmatrix} \text{diag}\left(D^{s-s_i}\beta_i(D), \; 1 \le i \le k\right) \\ \mathbf{0} \end{bmatrix} A^{-1}(D)$$

This is a polynomial matrix.

So if $y(D) = x(D)G(D)$, then $y(D)D^sG(D)^{-1} = D^sx(D)$. Because $D^sG(D)^{-1}$ is a polynomial matrix we see that if $y(D)$ is polynomial, then $D^sx(D)$ is polynomial. Therefore, $x(D)$ corresponds to a sequence that has only finitely many 1's whenever $y(D)$ corresponds to a sequence that has only finitely many 1's. This means that $G(D)$ is non-catastrophic

Conversely, if $\alpha_k(D)$ has a factor which is not just D, then $\frac{\beta_k(D)}{\alpha_k(D)}$ has infinite weight.

Now

$$G(D) = A(D)\left[\ \text{diag}\left(\tfrac{\alpha_i(D)}{\beta_i(D)}\right),\ 1 \le i \le k,\quad \mathbf{0}\ \right] B(D)$$

so taking

$$x(D) = \left[0 \ldots 0\ \frac{\beta_k(D)}{\alpha_k(D)}\right] A^{-1}(D)$$

first note that it corresponds to a sequence with infinitely many $1's$. Also

$$x(D)G(D) = \left[0 \ldots 1 \underbrace{0 \ldots 0}_{n-k}\right] B(D)$$

is a polynomial, so it corresponds to a sequence with finitely many 1's. Thus we have found an input sequence with infinitely many 1's that is encoded as an output sequence with finitely many 1's. This means that $G(D)$ is catastrophic.

□

Definition 15 *A generator matrix $G(D)$ is called* basic *if it is polynomial and has a polynomial right inverse.*

Note: Any basic generator matrix is an (basic) encoding matrix. (A generator matrix $G(D)$ is called an *encoding* matrix if $G(0)$ is invertible.)

Theorem 2 *Every convolutional code C can be described by a basic generator matrix.*

One consequence is that decoding can be conceptually viewed as in the figure, when a basic generator matrix is used for encoding. Namely, one can view the decoding problem in two stages as that of first finding the most likely output sequence that explains the (noisy) observations and then using the (feedforward) inverse of the basic encoding matrix to recover the corresponding information sequence.

Proof: We already know that C can be described by a polynomial delay free matrix. Call it $G(D)$. Let $G(D) = A(D)\Gamma(D)B(D)$ be the Smith form of $G(D)$, where

$$\Gamma(D) = [\text{diag}(\gamma_i(D)),\ 1 \le i \le k,\quad \mathbf{0}]\,.$$

Note that

$$G(D) = A(D)\,[\text{diag}(\gamma_i(D)),\ 1 \le i \le k]\,\hat{G}(D)$$

where $\hat{G}(D)$ is the $k \times n$ matrix consisting of the k rows of $B(D)$. Also $A(D)\,[\text{diag}(\gamma_i(D))]$ is invertible as a rational matrix. So $\hat{G}(D)$ also describes C. Further $\hat{G}(D)$ is a polynomial matrix

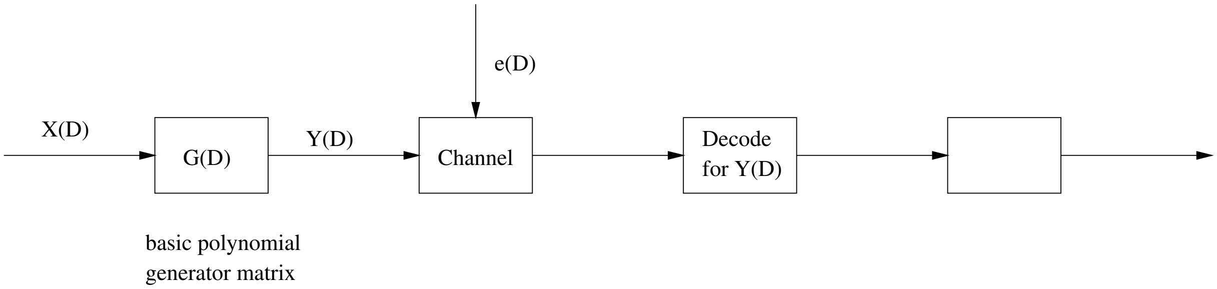

Figure 1:

and the first k columns of $B^{-1}(D)$ give a right inverse for $\hat{G}(D)$, and this is a polynomial right inverse because $B^{-1}(D)$ is polynomial.

$\square$

Theorem 3 *A polynomial generator matrix $G(D)$ (a $k \times n$ matrix of rank k, $k \leq n$) has a polynomial right inverse iff $\gamma_k(D) = 1$ where*

$$G(D) = A(D)\,[\mathrm{diag}(\gamma_i(D)),\ 1 \leq i \leq k,\ \mathbf{0}]\,B(D)$$

is the Smith form of $G(D)$.

Proof: See the book of Johannesson and Zigangirov.

$\square$

Corollary: For any convolutional code C, any basic (i.e. polynomial with polynomial inverse) generator matrix is non-catastrophic.

Proof : $\gamma_k(D) = 1 = D^0$ and use Massey-Sain, i.e. Theorem 1.

$\square$

5 Some deeper results w/o proof

Consider any polynomial generator matrix for C. Call it $G(D)$.

Def: Let

$$\gamma_i = \max_{1 \leq j \leq n} \deg g_{ij}(D).$$

Call it the i^{th} constraint length.

$m = \max_{1 \leq i \leq k} \gamma_i$ is called the memory.

$\gamma = \sum_{i=1}^{k} \gamma_i$ is called the overall constraint length.

The terminology comes from associating to the polynomial generator matrix a feedforward shift register implementation (controller form) that encodes information sequences into code sequences.

The i-th coordinate of the information sequence (viewed as a block of k bits at any symbol interval) will need to go into a shift register with γ_i blocks, the longest such shift register will have length m, and the overall number of blocks in all the k shift registers involved will be γ.

Definition 16 *A minimal basic encoding matrix is one whose overall constraint length is the smallest among all basic encoding matrices for the same code.*

Note: A convolutional code can have more than one basic encoding matrix describing it. In fact, if a basic encoding matrix is multiplied on the left or the right by any unimodular matrix, this gives another basic encoding matrix for the same code.

Now consider $G(D)$, an arbitrary generator matrix for C.

$$G(D) = \left[\frac{p_{ij}(D)}{q_{ij}(D)} \right]$$

Define $q_i(D) = \text{lcm}(q_{ij}(D),\ 1 \leq j \leq n)$. Write

$$\frac{p_{ij}(D)}{q_{ij}(D)} = \frac{\widetilde{p}_{ij}(D)}{q_i(D)}$$

Define: $\gamma_i = \max_{1 \leq j \leq n}(\deg\ q_i(D), \deg\ \widetilde{p}_{ij}(D))$ and $m = \max_{1 \leq i \leq k} \gamma_i$ and $\gamma = \sum_{i=1}^{k} \gamma_i$

γ_i is called the i-th constraint length of the generator matrix, m is called the memory, and γ is called the overall constraint length.

The terminology comes from associating to the generator matrix a feedback shift register implementation (controller form) that encodes information sequences into code sequences. The i-th coordinate of the information sequence (viewed as a block of k bits at any symbol interval) will need to go into a shift register with γ_i blocks, the longest such shift register will have length m, and the overall number of blocks in all the k shift registers involved will be γ.

Definition 17 *A minimal generator matrix for C is one whose overall constraint length γ is smallest among all generator matrices describing C.*

Theorem :

(I) A basic encoding matrix $G(D)$ is minimal iff the 0-1 matrix of the highest degree terms in each row is full rank. For example

$$G(D) = \begin{bmatrix} 1+D & D^2 & D^2+D & 1 \\ D & D^3 & D^2 & 1+D^3 \end{bmatrix} \Longrightarrow \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

and this is a full rank matrix, so $G(D)$ is a minimal basic encoding matrix for the code that it describes.

(II) Every minimal basic encoding matrix is a minimal generator matrix.

(III) Any two minimal generator matrices have exactly the same constraint lengths up to reordering.

(IV) Any systematic generator matrix is a minimal generator matrix.

(V) Any minimal generator matrix is non-catastrophic.

For proofs of these results, see the book of Johannesson and Zigangirov.

Note : Every convolutional code has a systematic generator matrix (this is just using the assumption that any generator matrix for the code has full rank), but there are convolutional codes that do not have systematic polynomial generator matrices. For instance, the rate 1/2 code with generator matrix

$$G(D) = [1 + D \;\; 1 + D + D^2] \; .$$

6 Duality for Convolutional Codes

Let $G(D)$ be a (rational) generator matrix for $\mathcal{C}$ with Smith form $G(D) = A(D)\Gamma(D)B(D)$ where

$$B(D) = \begin{bmatrix} \hat{G}(D) \\ (H^T(D))^{-1} \end{bmatrix} \text{ and } B^{-1}(D) = \begin{bmatrix} \hat{G}(D)^{-1} & H^T(D) \end{bmatrix}$$

where $\hat{G}(D)^{-1}$ is a right inverse of $G(D)$ and $(H^T(D))^{-1}$ is a right inverse for $H^T(D)$. The convolutional code generated by $H(D)$ is called the dual code of $\mathcal{C}$.

The connection with the block code notion of duality was described in class. See also the book of Johannesson and Zigangirov.