Convolutional Codes

Qi Zhang

Aarhus University School of Engineering

06/03/2014

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

Introduction

- Convolutional coding is a second technique in ECC.
- Convolutional codes differ from block codes in that encoder contains memory
  - The encoder outputs at any given time instant depends not only on the inputs at that time instant but also on some previous inputs.
- To simplify decoding, linear convolutional codes are preferred.
- A rate $R = k/n$ convolutional encoder with memory order K can be realized as a k-input and n-output linear sequential circuit with input memory K.
  - It is denoted as $C_{conv}(n, k, K)$.
- In CC, large minimum distances and high error correction capability are achieved by increasing the memory order K, which is different from block codes.
  - i.e. the higher the level of memory, the higher the complexity of the convolutional decoder and the stronger the error correction capability.

Linear Sequential Circuits

- Linear sequential circuits are constructed by using basic memory units (or delays), combined with adder or scalar multiplies that operate over $GF(q)$.
- Linear sequential circuits are known as finite state sequential machines(FSSMs).
- FSSM analysis is usually performed by means of a rational transfer function $G(D)$ of polynomial in the D domain, called the delay domain.
    - where message sequence and code sequence use the polynomial form $M(D)$ and $C(D)$, respectively.
- A convolutional encoder is a structure created using FSSMs which generates an output sequence based on a given input sequence.

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

Convolutional Encoder-1

- A convolutional encoder takes a k-tuple $\mathbf{m}_i$ of message elements as input, and generates the n-tuple $\mathbf{c}_i$ of coded elements as the output at a given time i.
- $\mathbf{c}_i$ depends not only on the input k-tuple $\mathbf{m}_i$ at time instant i but also on the previous k-tuples, $\mathbf{m}_j$ $(i - K \leq j < i)$.

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Convolutional Encoder-2

- This CC encoder takes two input elements and generates at the same instant three output elements. Code Rate $R_c = 2/3$.

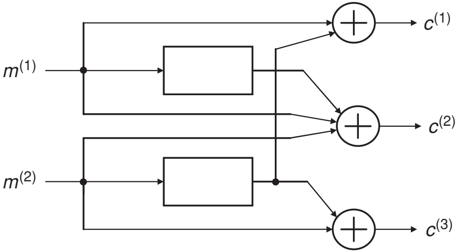


Figure: An example of a convolutional encoder

- Note: convolutional encoders of different structures can generate the same convolutional code.

Convolutional Encoder-3

Another example of CC encoder:

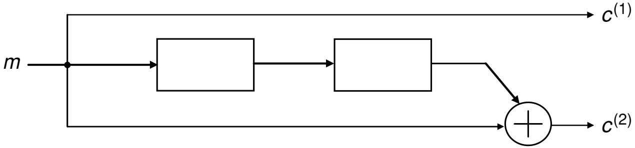


Figure: The structure of a systematic convolutional encoder of rate $R_c = 1/2$

- Note:the message elements appear explicitly in the output sequence together with the redundant elements, so it is a systematic code.

Convolutional Encoder-3

- In the given convolutional encoder below operating in GF(2), the input message k-tuple is simply one bit.
- At time instant i, the encoder takes one bit m_i and generates an output of two bits $c_i^{(1)}$ and $c_i^{(2)}$.
- The input sequence denoted by $\mathbf{m} = (m_0, m_1, m_2, \ldots)$.
- The output sequences denoted by $\mathbf{c}^{(1)} = (c_0^{(1)}, c_1^{(1)}, c_2^{(1)}, \ldots)$ and $\mathbf{c}^{(2)} = (c_0^{(2)}, c_1^{(2)}, c_2^{(2)}, \ldots)$

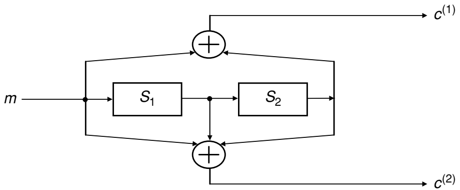


Figure: A CC encoder of rate $R_c = 1/2$

Convolutional Encoder-4

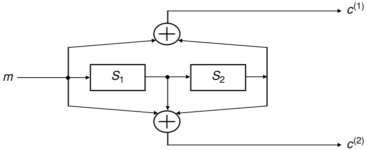


Figure: A CC encoder of rate $R_c = 1/2$

If message sequence $\mathbf{m} = (1, 0, 0, 0)$, then the states of the memory and the output sequences are

Table 6.1 Generator sequences for the FSSM of Figure 6.3

i	m	S_1	S_2	$c^{(1)}$	$c^{(2)}$
0	1	0	0	1	1
1	0	1	0	0	1
2	0	0	1	1	1
3	0	0	0	0	0

Convolutional Encoder-5

- The output sequences can be obtained as the convolution between the input sequence and the two impulse responses of the encoder.
- The impulse responses of the encoder $\mathbf{g}^{(1)}$ and $\mathbf{g}^{(2)}$ can be obtained by applying the unit impulse input sequence $\mathbf{m} = (1, 0, 0, \ldots)$ and observing the resulted outputs $c_i^{(1)}$ and $c_i^{(2)}$.

$$
\begin{aligned}
\mathbf{g}^{(1)} &= \left(g_0^{(1)}, g_1^{(1)}, g_2^{(1)}, \ldots, g_K^{(1)}\right) \\
\mathbf{g}^{(2)} &= \left(g_0^{(2)}, g_1^{(2)}, g_2^{(2)}, \ldots, g_K^{(2)}\right)
\end{aligned}
$$

- For this example, if the input is the unit impulse, then $\mathbf{c}^{(1)} = \mathbf{g}^{(1)}$, $\mathbf{c}^{(2)} = \mathbf{g}^{(2)}$,

$$
\begin{aligned}
\mathbf{g}^{(1)} &= (101) \\
\mathbf{g}^{(2)} &= (111)
\end{aligned}
$$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Convolutional Encoder-6

- Impulse responses are known as the generator sequences of the CC.
- The encoded sequences can be expressed as

$$\begin{aligned}\mathbf{c}^{(1)} &= \mathbf{m} * \mathbf{g}^{(1)} \\ \mathbf{c}^{(2)} &= \mathbf{m} * \mathbf{g}^{(2)}\end{aligned}$$

where $*$ denotes discrete convolution modulo 2. For an integer $l \geq 0$

$$c_l^i = \sum_{i=0}^{K} m_{l-i} g_i^{(j)} = m_l g_0^{(j)} + m_{l-1} g_1^{(j)} + \ldots + m_{l-K} g_K^{(j)}$$

For the given example of Figure 6.3, there are

$$\begin{aligned}c_l^{(1)} &= \sum_{i=0}^{2} m_{l-i} g_i^{(1)} = m_l + m_{l-2} \\ c_l^{(2)} &= \sum_{i=0}^{2} m_{l-i} g_i^{(2)} = m_l + m_{l-1} + m_{l-2}\end{aligned}$$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

- **Example:** let message sequence $\mathbf{m} = (1, 0, 1, 1)$, then calculate the output sequences.
- **Solution**:

$$\begin{aligned} \mathbf{c}^{(1)} &= (1011) * (101) = (100111) \\ \mathbf{c}^{(2)} &= (1011) * (111) = (110001) \end{aligned}$$

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

Description in the D-Transform Domain -1

- As we know, convolution in the time domain becomes multiplication in the spectral domain.
- This indicates a better convolutional codes based on expression given in D-transform domain or the delay domain.
- In D-transform domain, Convolution $*$ operation becomes multiplication.
- The sequences can use a polynomial form expressed in the variable D and the exponent of this variable determines the position of the element in the sequence.
- For example, the message sequence $\mathbf{m}^{(i)} = (m_0^{(i)}, m_1^{(i)}, m_2^{(i)}, \ldots)$ can be represented in polynomial form as

$$M^{(i)}(D) = m_0^{(i)} + m_1^{(i)} D + m_2^{(i)} D^2 + \ldots$$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Description in the D-Transform Domain -2

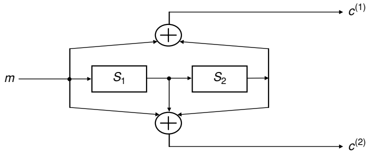
- Similarly, impulse responses or the generate sequences can also adopt a polynomial form:

$$\begin{aligned} \mathbf{g}_i^{(j)} &= \left(g_{i0}^{(j)}, g_{i1}^{(j)}, g_{i2}^{(j)}, \ldots\right) \\ G_i^{(j)}(D) &= g_{i0}^{(j)} + g_{i1}^{(j)} D + g_{i2}^{(j)} D^2 + \ldots \end{aligned}$$

- $G_i^{(j)}(D)$ is used as a generic expression for a convolution encoder with multiple inputs and outputs. $G_i^{(j)}(D)$ shows the relation between input i and output j.
- This polynomial can be regarded as generator polynomial for each output sequence of encoder.
- In this input-to-output path, the number of delays or memory units D is called the length of the register which is equal to the degree of the generator polynomial for this path.

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Description in the D-Transform Domain -3



- Use the example of encoder shown in Fig. 6.3, polynomial expression of the output sequences can be obtained as:

$$
\begin{aligned}
C^{(1)}(D) &= M(D)G^{(1)}(D) = M(D)(1+D^2) = c_0^{(1)} + c_1^{(1)}D + c_2^{(1)}D^2 + \dots \\
C^{(2)}(D) &= M(D)G^{(2)}(D) = M(D)(1+D+D^2) = c_0^{(2)} + c_1^{(2)}D + c_2^{(2)}D^2 + \dots
\end{aligned}
$$

- By multiplexing output polynomial $C^{(1)}(D)$ and $C^{(2)}(D)$, the code sequence in polynomial can be written as

$$C(D) = C^{(1)}(D^2) + DC^{(2)}(D^2)$$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

- **Example**: let message sequence $\mathbf{m} = (1, 0, 1, 1)$, determine the output sequence using D-transform domain.
- **Solution**:

$$\begin{aligned}
C^{(1)}(D) &= (1 + D^2 + D^3)(1 + D^2) = 1 + D^3 + D^4 + D^5 \\
C^{(2)}(D) &= (1 + D^2 + D^3)(1 + D + D^2) = 1 + D + D^5 \\
\mathbf{c}^{(1)} &= (100111) \\
\mathbf{c}^{(2)} &= (110001) \\
\mathbf{c} &= (11, 01, 00, 10, 10, 11)
\end{aligned}$$

- Note: each output sequence of the encoder has K bits more than the corresponding input sequence. Thus the code sequence becomes $2K$ bits longer for a $C_{conv}(2, 1, 2)$.
- Work on the Example 6.1 at home.
  - with $\mathbf{m} = (100011)$, to calculate the output sequences.

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

Description in the D-Transform Domain -4

- If $M^{(i)}(D)$ is the polynomial expression of the input sequence at input i and $C^{(j)}(D)$ is the polynomial expression of the output j generated by this input, then the polynomial $G_i^{(j)}(D) = C^{(j)}(D)/M^{(i)}(D)$. It is the transfer function that relates input i and output j.
- For a general encoder or FSSM structure which has k inputs and n outputs, there will be kn transfer functions that can be arranged in matrix form as

$$\mathbf{G}(D) = \begin{bmatrix} G_1^{(1)}(D) & G_1^{(2)}(D) & \dots & G_1^{(n)}(D) \\ G_2^{(1)}(D) & G_2^{(2)}(D) & \dots & G_2^{(n)}(D) \\ \vdots & \vdots & & \vdots \\ G_k^{(1)}(D) & G_k^{(2)}(D) & \dots & G_k^{(n)}(D) \end{bmatrix}$$

Description in the D-Transform Domain -5

- A CC $C_{conv}(n, k, K)$ produces the output sequences expressed in polynomial form as

$$\mathbf{C}(D) = \mathbf{M}(D)\mathbf{G}(D)$$

where $\mathbf{M}(D) = \left(M^{(1)}(D), M^{(2)}(D), \ldots, M^{(k)}(D)\right)$
and $\mathbf{C}(D) = \left(C^{(1)}(D), C^{(2)}(D), \ldots, C^{(n)}(D)\right)$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Description in the *D*-Transform Domain -5

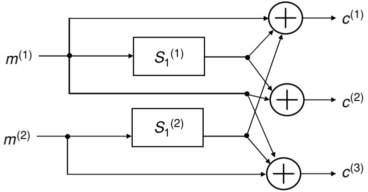


Figure: Encoder of $C_{conv}(3, 2, 1)$ of code rate $R_c = 2/3$.

- The input vector is $\mathbf{m} = \left(m_0^{(1)} m_0^{(2)}, m_1^{(1)} m_1^{(2)}, m_2^{(1)} m_2^{(2)}, \ldots\right)$.
- It can be written into separate input sequences: $\mathbf{m}^{(1)} = \left(m_0^{(1)}, m_1^{(1)}, m_2^{(1)}, \ldots\right)$ and $\mathbf{m}^{(2)} = \left(m_0^{(2)}, m_1^{(2)}, m_2^{(2)}, \ldots\right)$.
- Impulse responses or the generator sequence which relates input i and j: $\mathbf{g}_i^{(j)} = \left(g_{i,0}^{(j)}, g_{i,1}^{(j)}, \ldots, g_{i,K}^{(j)}\right)$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Description in the D-Transform Domain -6

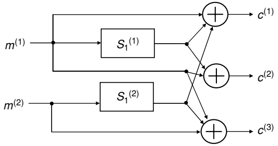


Figure: Encoder of $C_{conv}(3, 2, 1)$ of code rate $R_c = 2/3$.

- The generator sequences:

$$\begin{array}{lll} \mathbf{g}_1^{(1)} = (11) & \mathbf{g}_1^{(2)} = (11) & \mathbf{g}_1^{(3)} = (10) \\ \mathbf{g}_2^{(1)} = (01) & \mathbf{g}_2^{(2)} = (00) & \mathbf{g}_2^{(3)} = (11) \end{array}$$

- Expression for the generator polynomials in the D domain are

$$\begin{array}{ccc} G_1^{(1)}(D) = 1 + D & G_1^{(2)}(D) = 1 + D & G_1^{(3)}(D) = 1 \\ G_2^{(1)}(D) = D & G_2^{(2)}(D) = 0 & G_2^{(3)}(D) = 1 + D \end{array}$$

- If the message vector is $\mathbf{m}^{(1)} = (101)$ and $\mathbf{m}^{(2)} = (011)$, calculate the code sequences.

Description in the D-Transform Domain -7

- As the message vector is $\mathbf{m}^{(1)} = (101)$ and $\mathbf{m}^{(2)} = (011)$, the message polynomials are

$$M^{(1)}(D) = 1 + D^2 \quad M^{(2)}(D) = D + D^2$$

- The code polynomials are

$$\begin{aligned} C^{(1)}(D) &= M^{(1)}(D)G_1^{(1)}(D) + M^{(2)}(D)G_2^{(1)}(D) = 1 + D \\ C^{(2)}(D) &= M^{(1)}(D)G_1^{(2)}(D) + M^{(2)}(D)G_2^{(2)}(D) = 1 + D + D^2 + D^3 \\ C^{(3)}(D) &= M^{(1)}(D)G_1^{(3)}(D) + M^{(2)}(D)G_2^{(3)}(D) = 1 + D + D^2 + D^3 \end{aligned}$$

- Hence, the output sequence at each branch is

$$\mathbf{c}^{(1)} = (1100) \quad \mathbf{c}^{(2)} = (1111) \quad \mathbf{c}^{(3)} = (1111)$$

- The code sequence is $\mathbf{c} = (111, 111, 011, 011)$

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

A short summary on convolutional code encoder

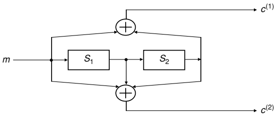
- The general structure of the encoder can be designed to have different memory level K_i in each of its branches.
- The memory order of the encoder is defined as the maximum register length, i.e.,

$$K = \max_{1 \leq i \leq k} (K_i)$$

  where i represent the input index.
- For a given $C_{conv}(n, k, K)$, if the input vector is a sequence of kL information bits, then the code sequence contains $n(L+K)$ bits.
- The amount $n_A = n(K+1)$ is the maximum number of output bits that one given input bit can influence, which is defined as constraint length of the code.
- Generally speaking the code rate of $C_{conv}(n, k, K)$ is k/n if $L \gg K$. However, for a given finite input sequence of length L, the code rate would be

$$R_c = \frac{kL}{n(L+K)}$$

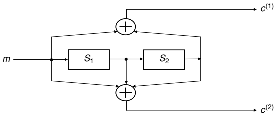
Representation of Connections



- When the input sequence is $\mathbf{m} = (100011)$, the corresponding output sequences can be given as follows:

Input m_i	State at t_i	State at t_{i+1}	$c^{(1)}$	$c^{(2)}$
–	0 0	0 0	–	–
1	0 0	1 0	1	1
0	1 0	0 1	0	1
0	0 1	0 0	1	1
0	0 0	0 0	0	0
1	0 0	1 0	1	1
1	1 0	1 1	1	0
0	1 1	0 1	1	0
0	0 1	0 0	1	1
0	0 0	0 0	0	0

State Diagram Representation -1



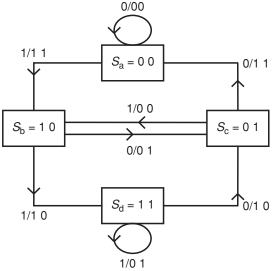
- For this encoder, we can also make a table which list all the possible transitions for constructing a state diagram of a convolutional encoder.
- Note: Table 6.5 there is NO relationship between the adjacent rows.

Table 6.5 Table of all the possible transitions for constructing a state diagram of a convolutional encoder

Input m_i	State at t_i	State at t_{i+1}	$c^{(1)}$	$c^{(2)}$
–	0 0	0 0	–	–
0	0 0	0 0	0	0
1	0 0	1 0	1	1
0	0 1	0 0	1	1
1	0 1	1 0	0	0
0	1 0	0 1	0	1
1	1 0	1 1	1	0
0	1 1	0 1	1	0
1	1 1	1 1	0	1

State Diagram Representation -2

- The state diagram is a pictorial representation of the evolution of the state sequences for the codes.
- The state of the FSSM that forms the encoder of a $1/n$ rate convolutional code is defined as the content of its K register stages.
- For this case there are four states, labelled $S_a = 00$, $S_b = 10$, $S_c = 01$, and $S_d = 11$.
- There are two possible transitions emerging from and arriving at any of these states, as there are only two possible input value, "1" or "0".



State Diagram Representation -3

- This state diagram has four states. However, each state only has two possible exits (i.e., transitions to the other states) only to the specific states.
- This kind of memory will be useful in determining if some transitions are not allowed in the decoded sequence to assist the decision for error correction.

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

Trellis Representation -1

- Trellis diagram can clearly describes the start of the state sequence but also the repetitive structure of the state evolution.
- Trellis diagram is a state versus time instant representation.

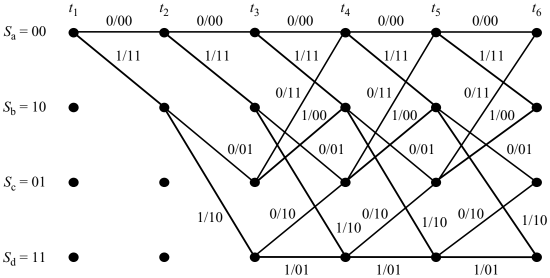


Figure: Trellis representation of the convolutional encoder of Figure 6.3.

- There are $2^K = 4$ possible states and the state structure becomes repetitive after time instant t_4.