Chapter 7: Channel coding: Convolutional codes

Vahid Meghdadi

University of Limoges
meghdadi@ensil.unilim.fr

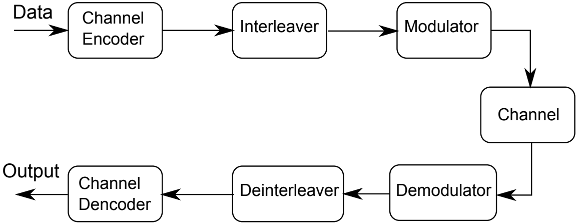
Reference : Digital communications by John Proakis; Wireless communication by Andreas Goldsmith

Channel coding

Convolutional encoder
- Encoder representation

Decoding
- Maximum Likelihood (ML) decoding
- Viterbi algorithm
- Error probability of CC

Communication system



The data is first coded then interleaved. The interleaving is used to avoid the channel presenting burst errors. Then it is transmitted through a kind of modulation over a noisy channel. At the receiver, all the operations must be inversed to estimate the transmitted data.

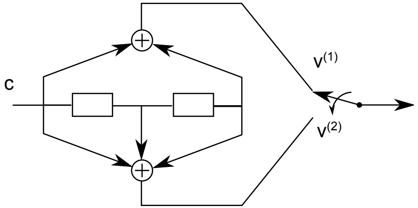
Channel coding

Channel coding can be classified in two categories:

- block coding
  - Cylclic codes
  - BCH
  - Reed-Solomon
  - Product Turbo code
  - LDPC
- trellis coding
  - Convolutional coding
  - TCM (Trellis code modulation)
  - Turbo codes (SCCC or PCCC)
  - Turbo TCM

Here, we are concentrating on convolutional coding.

Convolutional encoder



This is the simplest convolutional encoder. The input sequence is

$$\mathbf{c} = (\ldots, c_{-1}, c_0, c_1, \ldots, c_l, \ldots)$$

where l is the time index. The output sequences are:

$$\mathbf{v}^{(1)} = (\ldots, v_{-1}^{(1)}, v_0^{(1)}, v_1^{(1)}, \ldots, v_l^{(1)}, \ldots)$$

$$\mathbf{v}^{(2)} = (\ldots, v_{-1}^{(2)}, v_0^{(2)}, v_1^{(2)}, \ldots, v_l^{(2)}, \ldots)$$

Convolutional code characteristics

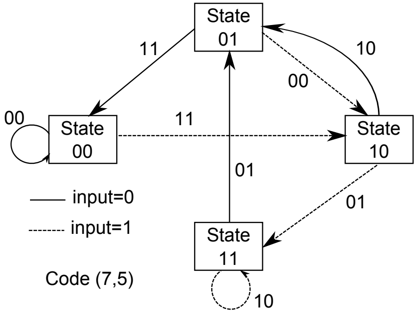
- The constraint length of a CC is the number of input bit involved to generate each output bit. It is the number of the delay elements plus one. For the previous example, the constraint length is 3.
- Once all the output are serialized and get out of the coder, k right shift occurs.
- The code rate is defined as $R_c = k/n$ where n is the number of output.
- For the previous example $n = 2$, $k = 1$, and $R_c = 0.5$.

Describing a CC by its generator

- In the previous example, assuming all-zero state, the sequence $\mathbf{v}_1^{(1)}$ will be [101] for a 1 at the input (impulse response).
- At the same time the sequence $\mathbf{v}_1^{(2)}$ will be [111] for a 1 at the input.
- Therefore, there are two generators $\mathbf{g}_1 = [101]$ and $\mathbf{g}_2 = [111]$ and the encoder is completely known.
- It is convenient to represent the encoder in octal form of as $(\mathbf{g}_1, \mathbf{g}_2) = (5, 7)$.

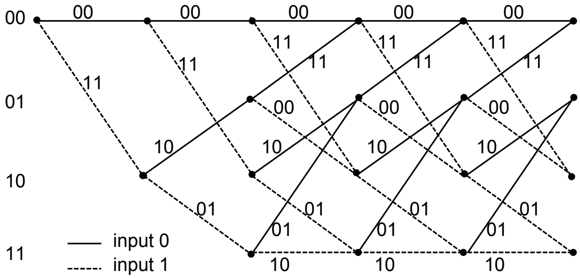
Exercise: Draw the encoder circuit of the code (23,35).

State diagram representation



Using this diagram one can easily calculate the output of the coder.
Exercise: What is the coded sequence of [10011]?

Trellis representation



We assume that the trellis begins at the state zero. The bits on the diagram show the output of the encoder $(v_t^{(1)}, v_t^{(2)})$.

Exercise: Using above diagram what is the coded sequence of the [1 0 1 1 0 1]?

Modulation and the channel

The coded sequence is denoted by **C**. The output of the encoder is modulated, then sent to the channel.
We assume a BPSK modulation with $0 \rightarrow -1$ volt and $1 \rightarrow +1$ volt. The channel adds a white Gaussian to the signal.
The RX receives a real sequence **R**. The decoder should estimate the transmitted sequence by observing the channel output.
For the i-th information bit, corresponding to the i-th branch in the trellis, the code sequence of size n is denoted by C_{ij} where $0 < i < n-1$. Assuming a memoryless channel, the received signal for the i-th information bit is:

$$r_{ij} = \sqrt{E_c}(2C_{ij} - 1) + n_{ij}$$

Maximum Likelihood decoding

Given the received sequence $\mathbf{R}$, the decoder decides that the coded sequence $\mathbf{C}^*$ was transmitted if

$$p(\mathbf{R}|\mathbf{C}^*) \geq p(\mathbf{R}|\mathbf{C}) \qquad \forall \mathbf{C}$$

It means that the decoder finds the maximum likelihood path in the trellis given $\mathbf{R}$. Thus, over an AWGN channel and for a code of rate $1/n$ and for an information sequence of size L, the likelihood can be written as:

$$p(\mathbf{R}|\mathbf{C}) = \prod_{i=0}^{L-1} p(R_i|C_i) = \prod_{i=0}^{L-1} \prod_{j=0}^{n-1} p(R_{ij}|C_{ij})$$

Log Likelihood function

Taking logarithm:

$$\log p(\mathbf{R}|\mathbf{C}) = \sum_{i=0}^{L-1} \log p(R_i|C_i) = \sum_{i=0}^{L-1}\sum_{j=0}^{n-1} \log p(R_{ij}|C_{ij})$$

The expression $B_i = \sum_{j=0}^{n-1} \log p(R_{ij}|C_{ij})$ is called the *branch metric*. Maximizing the likelihood function is equivalent to maximizing the log likelihood function.
The log likelihood function corresponding to a given path in the trellis is called the *path metric*, which is the sum of the branch metrics in the path.

ML and hard decoding

In hard decoding, the sequence **R** is demodulated and contains only 0 and 1. This is called hard decision. The probability of error in hard decision depends to the channel state and is represented by p. If **R** and **C** are N symbol long and differ in d places (i.e. Hamming distance), then

$$p(\mathbf{R}|\mathbf{C}) = p^d(1-p)^{N-d}$$

$$\log p(\mathbf{R}|\mathbf{C}) = -d \log \frac{1-p}{p} + N \log(1-p)$$

The second term is independent of **C**. Therefore only the first term should be maximized. Conclusion: the sequence **C** with minimum Hamming distance to the received sequence **R** corresponds to the ML sequence.

Soft decoding

In an AWGN with a zero mean Gaussian noise of variance $\sigma^2 = N_0/2$,

$$p(R_{ij}|C_{ij}) = \frac{1}{\sqrt{2\pi}\sigma} exp\left[-\frac{(R_{ij} - \sqrt{E_c}(2C_{ij} - 1))^2}{2\sigma^2}\right]$$

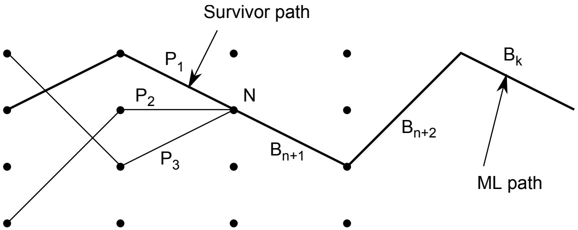
ML is equivalent to choose the C_{ij} that is closest in Euclidean distance to R_{ij}.
Taking logarithm and neglecting all scaling factor and common terms to all $\mathbf{C}_i$, the branch metric is:

$$\mu_i = \sum_{j=1}^{n} R_{ij}(2C_{ij} - 1)$$

Viterbi algorithm (1967)

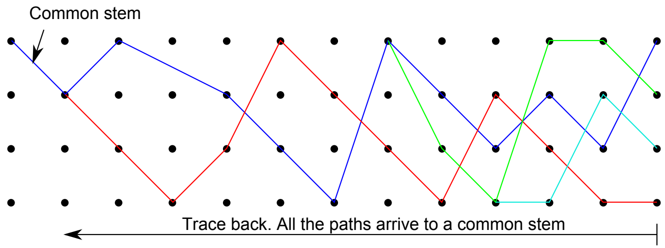
- There are a huge number of paths to test for even moderate values of L. The Viterbi algorithm reduces considerably this amount.
- Viterbi proposed an ML decoding by systematically removing paths that cannot achieve the highest path metric.
- When two paths enter into a given node, the path with lowest likelihood cannot be the survivor path: it can be removed.

Viterbi algorithm



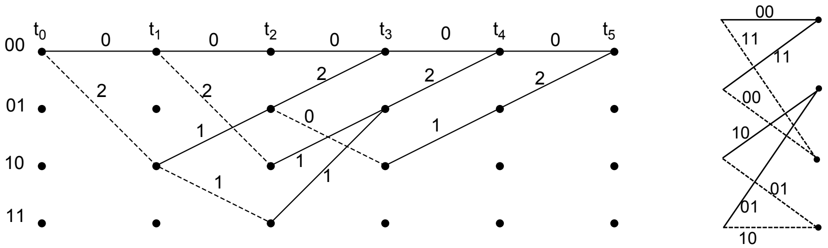
- P_1 is the survivor path because the partial path metric P_1 is greater than P_2 and P_3 where $P_i = \sum_{k=0}^{n-1} B_k^i$.
- The ML path starting from node N to infinity has the path metric of $P_i + \sum_{k=n}^{\infty} B_k$ $(i = 1, 2, 3)$.
- Thus, the paths P_2 and P_3 cannot complete the path from n to ∞ and give a ML path. They will be discarded.

Trace back



- Theoretically, the depth of the trace back operation is ∞. However, it is observed that all the surviving paths merge finally to one common node, giving the decoding delay.
- Statistically speaking, for sufficiently large trellis' depth (typically five times the constraint length), all the paths merge to a common node. Therefore a decoding delay of $5K$ is considered.

Distance properties (example CC(7,5))



- The error occurs when the survivor path is to be selected.
- If the Hamming distance between two sequences entering to a node is large, the probability of making error becomes small.
- The *free distance* of CC(7,5) is 5 corresponding to the path 00-10-01-00.

Good codes

Testing all the possible combinations of encoders, the best codes for a given constraint length are proposed in tables.

Constraint Length	Generaor in octal	d_{free}
3	(5,7)	5
4	(15,17)	6
5	(13,35)	7
6	(53,75)	8
7	(133,171)	10

Rate 1/2 maximum free distance CC

Good codes

Constraint Length	Generaor in octal	d_{free}
3	(5,7,7)	8
4	(13,15,17)	10
5	(25,33,37)	12
6	(47,53,75)	13
7	(133,145,175)	15

Rate 1/3 maximum free distance CC

Error probability

- Convolutional code is linear so the P_e is calculated by supposing all zero sequence is sent.
- Coherent BPSK over AWGN is assumed.
- The energy per coded bit is related to the energy per information bit by $E_c = R_c E_b$.
- The probability of mistaking all zero sequence by another sequence with Hamming distance of d is:

$$P_e(d) = Q\left(\sqrt{\frac{2E_c}{N_0}d}\right) = Q\left(2\gamma_b R_c d\right)$$

- This probability is called *pairwise error probability*.
- There is no analytic expression for bit error rate but just some upper bounds.

Error probability