Capacity-Approaching Coded-Modulation Schemes

5.9 级联 TCM

TCM 码可以按 Turbo 方式并行级联，也可以与其它码构成串行级联系统（例如，在 RS+TCM 中做内码，在 TCM+STBC 中做外码）。

5.9.1 TCM 码的(Symbol-Based Nonbinary) MAP 译码算法

Consider a general trellis code. Let $\mathbf{u}=(\mathbf{u}_1,\mathbf{u}_2,...,\mathbf{u}_N)$ be an information symbol sequence of length N. Each information symbol is assumed to be a binary m-tuple, i.e., $\mathbf{u}_k=(u_{k,0},u_{k,1},...,u_{k,m-1})\in\mathbb{F}_2^m$, for k=1,2,…N. For convenience, we will denote $\mathbf{u}_k$ by its decimal representation; i.e., assume $u_k\in\mathcal{U}=\{0,1,...,2^m-1\}$. The m-bit symbols are fed into a TCM encoder, producing a sequence of N modulated symbols, $\mathbf{x}=(x_1,x_2,...,x_N)$, where $x_k\in\mathcal{X}$ with $\mathcal{X}$ standing for the signal constellation of size $|\mathcal{X}|=2^n$. We assume that the symbols x_k are transmitted over an AWGN channel. Let $\mathbf{y}=\mathbf{y}_1^N=(y_1,y_2,...,y_N)$ be the received sequence, where y_k (k=1,2,…N) is the noisy observation of x_k if x_k is not punctured; otherwise, set $y_k=*$ (implying no observation obtained).

Figure 5.9.1 A trellis coding system

The received sequence is fed to the TCM decoder to produce the estimate of transmitted 2^m-ary information symbols. To minimize the probability of symbol errors, the symbol-based MAP decoder may be used. This decoder computes the APP for every 2^m-ary information symbol; i.e., $P(u_k=i\,|\,\mathbf{y}_1^N)$, for $i=0,1,...,2^m-1, k=1,2,...,N$. Then it makes decisions according to the MAP criterion:

$$\hat{u}_k=\arg\max_{i\in\mathcal{U}}P(u_k=i\,|\,\mathbf{y}_1^N)$$

where $\mathcal{U} = \{0,1,...,2^m - 1\}$.

We now describe the MAP algorithm in detail by using the trellis diagram. For no loss of generality of the result, we assume that the trellis diagram contains parallel transitions. Refer to Fig. 5.9.2. Denote by $\mathcal{S}$ the set of encoder states and by $M = |\mathcal{S}|$ the number of states. In Fig.5.9.2 $|\mathcal{S}|$=4 is assumed. Let S_k be the encoder state at time k. A branch at the kth section in the corresponding trellis diagram can be specified by $b \equiv (S_{k-1}, u_k, x_k, S_k)$, where u_k and x_k are the information and modulated symbols associated with the state transition $S_{k-1} \to S_k$. Denote by $B_k(s', s)$ the set of all the parallel branches connecting $S_{k-1} = s' \in \mathcal{S}$ and $S_k = s \in \mathcal{S}$. The non-binary MAP algorithm can be derived as follows.

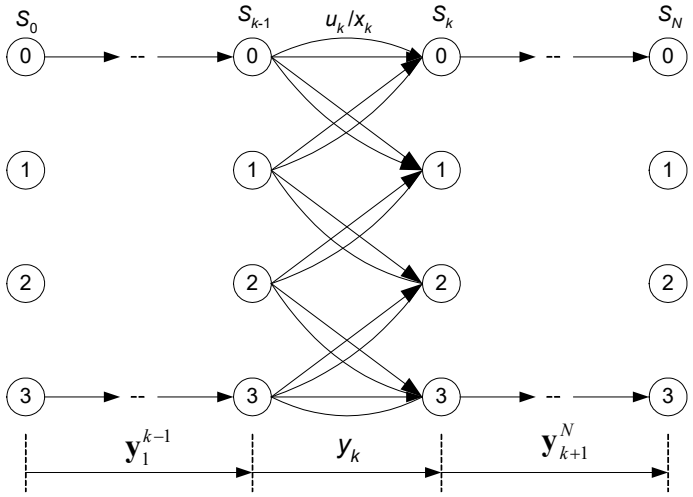


Figure 5.9.2

$$P(u_k = i \mid \mathbf{y}_1^N) = \sum_{\forall (s',s) \in B_k^i} P(u_k = i, S_{k-1} = s', S_k = s \mid \mathbf{y}_1^N)$$

$$= \sum_{\forall (s',s) \in B_k^i} \frac{p(u_k = i, S_{k-1} = s', S_k = s, \mathbf{y}_1^N)}{p(\mathbf{y}_1^N)} \tag{5.9.1}$$

where B_k^i is the set of transitions $S_{k-1} \to S_k$ that are caused by the input u_k=i. According to the BCJR algorithm, the probabilities $p(u_k = i, S_{k-1} = s', S_k = s, \mathbf{y}_1^N)$ can be calculated as

$$p(u_k = i, S_{k-1} = s', S_k = s, \mathbf{y}_1^N) = \alpha_{k-1}(s') \cdot \gamma_k^i(s', s) \cdot \beta_k(s)$$

where

$\alpha_k(s) \equiv p(S_k = s, \mathbf{y}_1^k)$ is the forward state metric;

$\beta_k(s) \equiv p(\mathbf{y}_{k+1}^N | S_k = s)$ is the backward state metric; and

$\gamma_k^i(s',s) \equiv p\left(u_k = i, S_k = s, \mathbf{y}_k \mid S_{k-1} = s'\right)$ is the metric of the trellis branch b connecting $S_{k-1} = s'$ to $S_k = s$ with $u_k = i$.

Note that we do not necessarily need the exact values of $P\left(u_k = i \mid \mathbf{y}_1^N\right)$, but only their ratios. Since for a fixed k, the vector $P\left(u_k = i \mid \mathbf{y}_1^N\right)$, being a probability vector, must sum to unity, thus by normalizing the sum to unity, we can omit the constant factor $\frac{1}{p(\mathbf{y}_1^N)}$ in (5.9.1) and use the following simplified expression:

$$P\left(u_k = i \mid \mathbf{y}_1^N\right) \propto \sum_{\forall (s',s) \in B_k^i} \alpha_{k-1}(s') \cdot \gamma_k^i(s',s) \cdot \beta_k(s)$$

Now we consider the calculation of the branch metric $\gamma_k^i(s',s)$, which is the key point of the symbol-based MAP algorithm, and is also the primary difference between the MAP algorithm for binary turbo codes and the algorithm under consideration.

Using Bayes' rule, the term $\gamma_k^i(s',s)$ can be written as

$$\gamma_k^i(s',s) = p(\mathbf{y}_k \mid S_k = s, S_{k-1} = s', u_k = i) \cdot P(S_k = s \mid S_{k-1} = s', u_k = i) \cdot P(u_k = i \mid S_{k-1} = s')$$

$$= p(y_k \mid x_k) \cdot P(u_k = i) \cdot P(S_k = s \mid S_{k-1} = s', u_k = i) \qquad (5.9.2)$$

where $p(y_k \mid x_k)$ is the probability of receiving y_k given the transmitted symbol x_k, $P(u_k)$ is the *a priori* probability of $u_k = i$, and $P(S_k = s \mid S_{k-1} = s', u_k = i)$ is 1 or 0, indicating whether or not a state transition $S_{k-1} \to S_k$ with input $u_k = i$ exists. Thus, for a trellis branch b, we have

$$\gamma_k^i(s',s) = \begin{cases} P(u_k), & \text{if } y_k = * \\ p(y_k \mid x_k) \cdot P(u_k), & \text{otherwise} \end{cases} \qquad (5.9.3)$$

Define

$$\gamma_k(s',s) \equiv p\left(S_k = s, \mathbf{y}_k \mid S_{k-1} = s'\right)$$

$$= \sum_{\forall b_k \in B_k(s',s)} \gamma_k^i(s',s)$$

which represent the combined branch metrics. Then the non-binary BCJR algorithm can be summarized as follows.

Forward recursion: $$\alpha_k(s) = \sum_{s' \in \mathcal{S}} \alpha_{k-1}(s')\gamma_k(s',s) \tag{5.9.4}$$

Backward recursion: $$\beta_{k-1}(s') = \sum_{s \in \mathcal{S}} \beta_k(s)\gamma_k(s',s) \tag{5.9.5}$$

Output: $$P\left(u_k = i \,|\, \mathbf{y}_1^N\right) \propto \sum_{\forall (s',s) \in B_k^i} \alpha_{k-1}(s') \cdot \gamma_k^i(s',s) \cdot \beta_k(s) \tag{5.9.6}$$

The boundary conditions are the same as for the binary turbo codes. For numerical stability, it is necessary to control the dynamic range of the likelihood terms computed in (5.9.4) to (5.9.6). This can be performed by normalizing the sum of the $\alpha_k(s)$ and the $\beta_k(s)$ values to unity at every particular *k* symbol. Of course, the algorithm can also be implemented in the log-domain with reduced computational complexity, resulting in the Log-MAP or Max-Log-MAP version.

5.9.2 Turbo Trellis-Coded Modulation (TTCM)

TTCM 是由两个（或多个）TCM 码按照 Turbo 码的方式级联起来构成的并行级联编码调制系统，它是标准二元 turbo 码的直接推广。This technique was originally proposed by Robertson and Worz in 1996 [?].

5.9.2.1 TTCM Encoder

A TTCM encoder consists of the parallel concatenation of two (or multiple) trellis-coded modulation (TCM) schemes in the same fashion as binary turbo codes. Each component TCM encoder consists of a convolutional encoder of rate $m/(m+1)$ and a signal mapper. The mapping of coded bits to signal points follows the Ungerboeck's rule of mapping by set partitioning. The first TCM encoder operates on the original input bit sequence, while the second TCM encoder manipulates the interleaved version of the input bit sequence. Here, the interleaving works on groups of bits instead of individual bits. See Fig. 5.9.3, where a symbol-based odd-even interleaver is assumed. To avoid excessive rate loss, a special puncturing technique is used: Symbols from a 2^{m+1}-ary signal constellation are transmitted alternately from the first and second encoders; i.e., the puncturing matrix is given by

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Thus each information bit pair is transmitted in exactly one transmitted symbol with the parity bit alternately chosen from the first and second encoders.

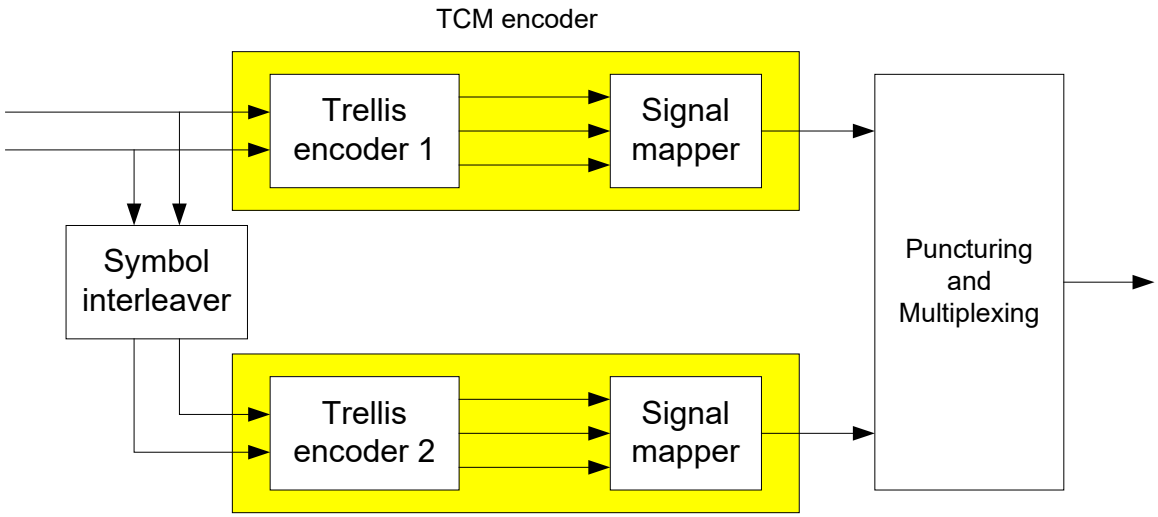


Figure 5.9.3 TTCM encoder. The puncturer is used to avoid rate loss by enabling the transmission of signals carrying the same information bits only once.

Assume that a D-dim constellation is used. At each step, m information bits are input to the TTCM encoder and then a modulated symbol selected from a D-dim signal constellation of size 2^{m+1} is transmitted. This yields a throughput of $2m/D$ bits per 2-dim symbol.

We now consider the above odd-even interleaving-puncturing rule more 更深入、更一般地. Let L denote the number of component codes used in a parallel concatenated system. Then the above odd-even interleaving-puncturing rule may be expressed mathematically via introducing the following definitions.

Definition: A (pseudo-random) interleaver satisfying the following constraint is called a modulo-L interleaver:

$$\pi(k) \bmod L = k \bmod L ,$$

where k stands for the symbol position before interleaving.

Definition: A modulo-L puncturer is one that punctures all the modulated symbols in the lth component code, except those at position $\{k \mid k \bmod L = l\}$.

For a TTCM scheme, the modulo-L interleaving together with the modulo-L puncturing rule ensure that one and only one modulated symbol carrying the same $\boldsymbol{u}_k$ is transmitted, and that the punctured symbols are uniformly distributed in each component code. It is clear that when L=2, the above modulo-L interleaving-puncturing rule is equivalent to that used in [3].

As an example, Fig.5.9.4 shows an 8-state TTCM encoder for 8-PSK.

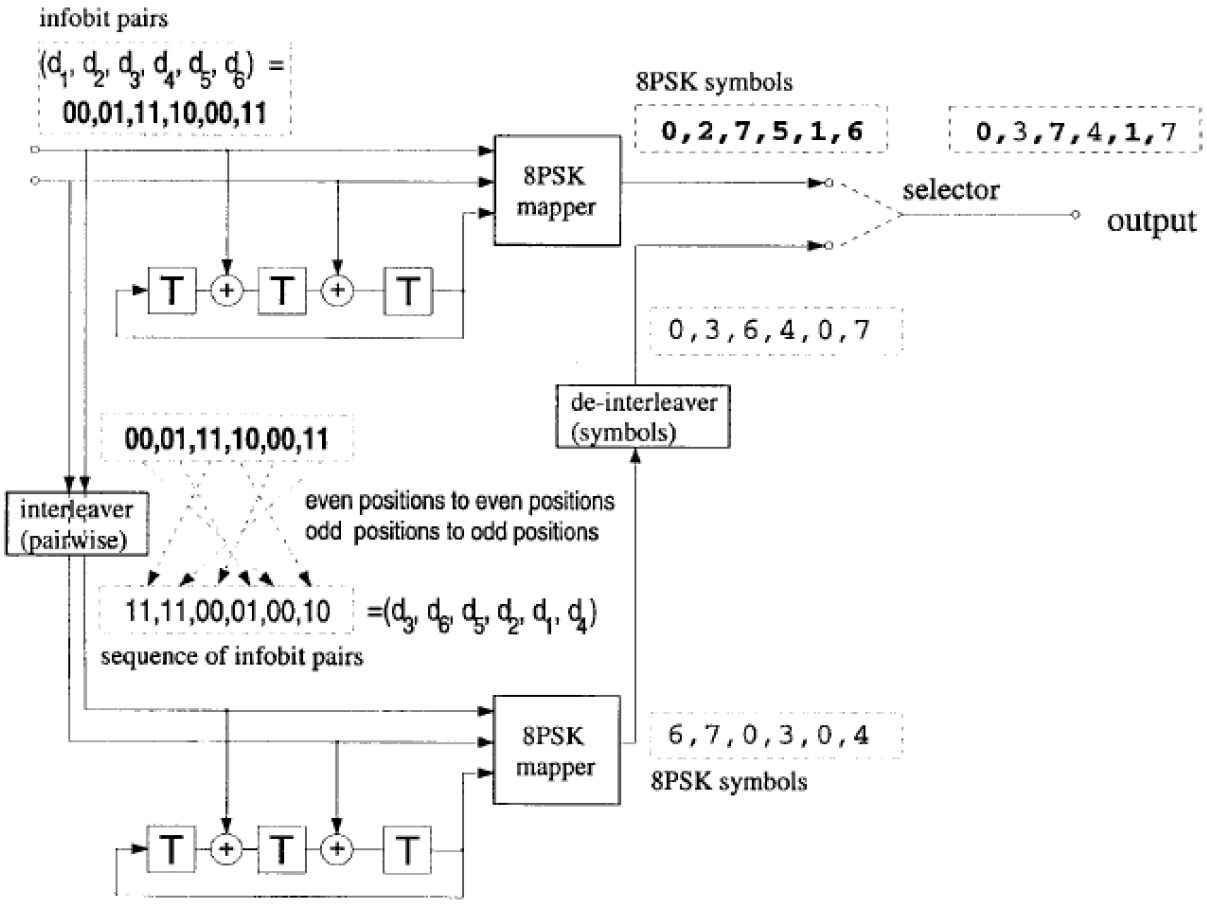


Figure 5.9.4 From [Robertson98]

Table 5.9.1 lists the generator polynomials of some component TCM codes that can be employed in the TTCM scheme. These polynomials were obtained by Robertson and Worz using an exhaustive computer search method. In Table 5.9.1, $\tilde{m}$ denotes the number of information bits to be encoded from the m information bits contained in an input symbol.

"PUNCTURED" TCM CODES WITH BEST MINIMAL DISTANCE FOR 8-PSK AND QAM (IN OCTAL NOTATION)

Code	$\tilde{m}$	$H^0(D)$	$H^1(D)$	$H^2(D)$	$H^3(D)$	$d^2_{\text{free}}/\Delta^2_0$
2-dim. 8-PSK, 8 states	2	11	02	04		3
4-dim. 8-PSK, 8 states	2	11	06	04		3
2-dim. 8-PSK, 16 states	2	23	02	10		3
4-dim. 8-PSK, 16 states	2	23	14	06		3
2-dim. Z^2, 8 states	3	11	02	04	10	2
2-dim. Z^2, 16 states	3	21	02	04	10	3
2-dim. Z^2, 8 states	2	11	04	02		3
2-dim. Z^2, 16 states	2	21	04	10		4

5.9.2.2 TTCM Decoder

The iterative decoder for TTCM is similar to that used to decode binary turbo codes, except that there is a difference in the nature of the information passed from one decoder to the other. Fig. 5.9.5 illustrates the concept of *a priori*, *a posteriori* and extrinsic information

used in TTCM decoder.

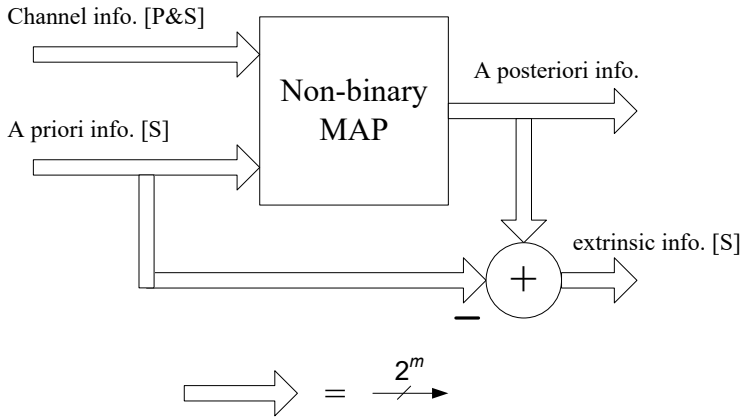


Figure 5.9.5. Schematic of the component decoders for TTCM codes

Note that in a symbol-based nonbinary TTCM scheme, the *m* systematic information bits and the parity bit are transmitted together in the same modulated symbol. Hence, the systematic component (corresponding received systematic value in binary turbo decoder) of the non-binary symbol cannot be separated from the extrinsic component. In this scenario the symbol-based information can be split into only two components: La and $L_{e\&s}$. Each decoder passes only the latter information to the next component decoder while the *a priori* information is removed at each component decoder's output.

The complete TTCM decoder operating in the log domain is depicted in Fig. 5.9.6. At the beginning of decoding, the received symbols are input to the "metric" block so as to generate a set of $M=2^{m+1}$ symbol likelihoods. The selector switches select the current symbol's reliability metric produced by the "metric block" if the current symbol was not punctured by the corresponding encoder. Otherwise puncturing will be applied where the likelihoods of the various legitimate symbols at index *k* are set to 0 in the log-domain. That is 初始化

$$L_{p\&s,k}=\begin{cases}\log p(y_k \mid x_k), & \text{for un-punctured symbols}\\ 0, & \text{for punctured symbols}\end{cases}$$

On this basis, the TTCM decoder operates according to the "turbo principle".

Specifically, for the MAP decoder 1, if the modulated symbol from the component TCM encoder 1 is punctured at time *k*, then y_k = * for decoder 1, and set $L_{p\&s}=0$. The only input is *a priori* information $L_a^{(1)}=\tilde{L}_{e\&s}^{(2)}$ from the other component decoder, and this includes the systematic information for u_k. At the output of the MAP decoder 1, the *a priori* information is subtracted from the *a posteriori* information, so that the same information is not used more than once in the other component decoder. Thus, the information passed from the decoder 1 to the decoder 2, $L_e=L-L_a$, contains only the extrinsic information on u_k.

If the modulated symbol from the component TCM encoder 1 is not punctured at time *k*, then $L_{p\&s}\triangleq\log p(y_k \mid x_k)$ represents the channel information, and the information passed

from the decoder 1 to the decoder 2, $L_e = L_{e\&s}$, contains the inseparable *extrinsic* as well as *systematic* information component on u_k.

The decoder 2 operates according to the same message-passing principle as above.

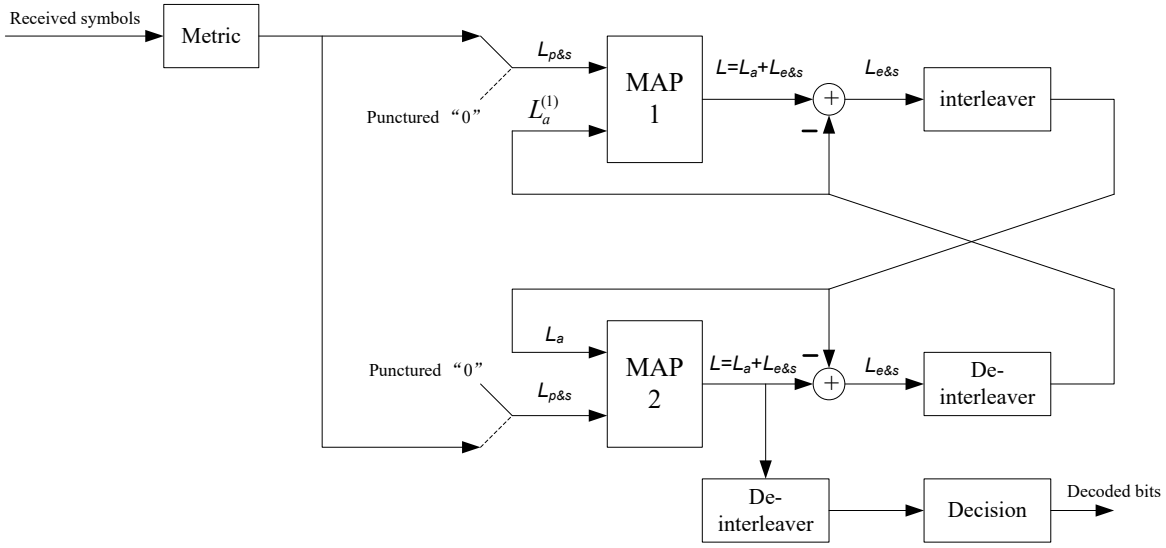


Figure 5.9.6 Structure of the TTCM decoder.

5.9.2.3 Numerical results

See Fig. 5.9.10b for BER performance.

5.9.3 Concatenated Two-State TCM (CT-TCM) Schemes

5.9.3.1 Encoder

The component encoder of a CT-TCM code consists of a binary two-state trellis encoder followed by a multi-ary signal mapper, see Fig. 5.9.7. Let a binary n-tuple $\boldsymbol{d}_k = (d_{k,0}, \cdots, d_{k,n-1})$ be an information symbol. Let $\boldsymbol{d} = \{\boldsymbol{d}_k, k \geq 0\}$ be an input sequence to the binary encoder, producing a coded symbol sequence $\boldsymbol{c}=\{\boldsymbol{c}_k\}$. Each $\boldsymbol{c}_k$ contains a parity check bit q_k, i.e., $\boldsymbol{c}_k = (\boldsymbol{d}_k, q_k)$, and is mapped to an appropriate 2^{n+1}-ary signal constellation, producing a modulated symbol x_k. In the following, $\boldsymbol{c}$ and $\boldsymbol{x}=\{x_k\}$ are also referred to as unmodulated and modulated codewords, respectively.

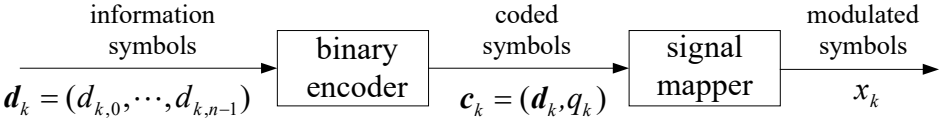


Fig. 5.9.7. The component encoder structure for the CT-TCM scheme.

We assume that the binary encoder in Fig. 5.9.7 is characterized by the two-state trellis in Fig. 5.9.8 (similar to the tree encoder in [Liping2001]). The parity check bit q_k is generated by

$$q_k = q_{k-1} + \boldsymbol{d}_k \cdot \boldsymbol{g}_k = \sum_{i=0}^{k} \boldsymbol{d}_i \cdot \boldsymbol{g}_i \qquad \text{mod } 2, \quad k > 0 \tag{5.9.7}$$

with $q_0 = \boldsymbol{d}_0 \cdot \boldsymbol{g}_0$. Here $\boldsymbol{g}_k = (g_{k,0}, g_{k,1}, \cdots, g_{k,n-1})^T$ is an indication vector defined by

$$g_{k,j} = \begin{cases} 1 & \text{if } d_{k,j} \text{ participates in parity check} \\ 0 & \text{otherwise} \end{cases} \tag{5.9.8}$$

The code in Fig. 5.9.8 is completely specified by $\{\boldsymbol{g}_k\}$.

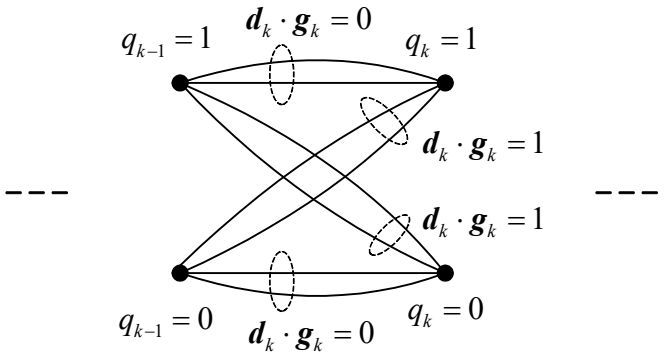


Fig. 5.9.8. A two-state trellis diagram with 2^{n+1} branches in a trellis section. (In this figure, n=2.)

■ The Global Encoder of CT-TCM Scheme

Fig. 5.9.9 depicts a global CT-TCM scheme, where M component encoders are concatenated in parallel by M symbol-interleavers. Modulo-M interleavers satisfying the following constraints are assumed,

$$\pi^{(m)}(k) \bmod M = k \bmod M, \qquad \text{for } m = 0,1, \ldots M\text{-}1$$

In order to increase spectral efficiency, we puncture all the modulated symbols in the mth component code, except those at position $\{k \mid k \bmod M = m\}$. This, together with the constraint in (3), ensures that one and only one modulated symbol carrying the same $\boldsymbol{d}_k$ is transmitted, and that the punctured symbols are uniformly distributed in each component code. We will always assume that a signal constellation of size 2^{n+1} is used. This yields a spectral efficiency of n bits per symbol.

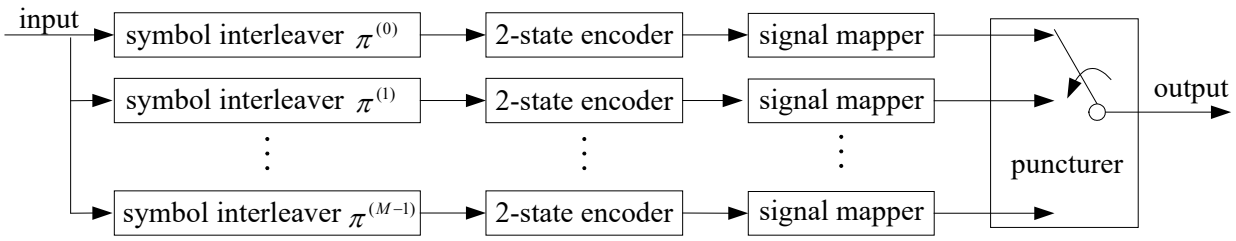


Fig. 5.9.9. A global CT-TCM encoder structure.

5.9.3.2 Decoder

The CT-TCM decoder structure is based on the multi-dimensional turbo-decoder incorporating the BCJR algorithm. The global decoder operating in the log domain for a CT-TCM code is shown in Fig. 5.9.10a. It consists of M local APP (*a posteriori* probability) decoders, each for one component code. The variables involved in Fig. 5.9.10a are the log-likelihood (LL) values, as detailed below.

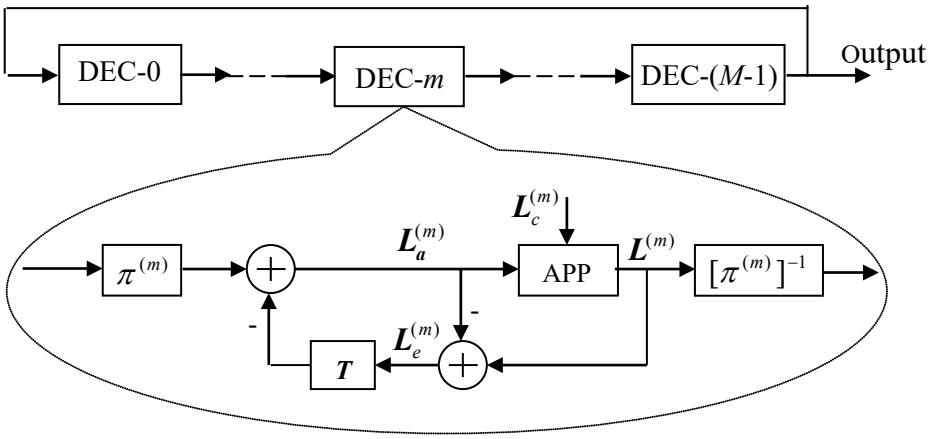


Fig. 5.9.10a. The global decoder: "T" for delay of one iteration and "π" for interleaving.

$\boldsymbol{L}^{(m)}$ the *a posteriori* LL values for all information symbols after decoding the mth component code.

$\boldsymbol{L}_c^{(m)}$ the LL values $\{\log p(y_k \mid x_k)\}$ based on individual channel observations of the mth component code. Its elements $\{L_{c,k}^{(m)}\}$ are calculated as

$$L_{c,k}^{(m)} = \begin{cases} \log p(y_k \mid x_k), & \text{for un - punctured symbols} \\ 0, & \text{for punctured symbols} \end{cases}$$

$\boldsymbol{L}_a^{(m)}$ the *a priori* LL values for all information symbols for the mth component code. It is initialized to zeroes, implying no *a priori* information.

$\boldsymbol{L}_e^{(m)}$ the extrinsic information produced by the mth component code, defined by

$$\boldsymbol{L}_e^{(m)} = \boldsymbol{L}^{(m)} - \boldsymbol{L}_a^{(m)}.$$

The M local decoders operate successively. $\boldsymbol{L}_a^{(m)}$ contains the accumulated extrinsic information generated by all the local decoders except DEC-m,

$$\boldsymbol{L}_a^{(m)} = \underbrace{\boldsymbol{L}_e^{(m+1)} + \boldsymbol{L}_e^{(m+2)} \cdots + \boldsymbol{L}_e^{(M-1)}}_{\text{from the previous iteration}} + \underbrace{\boldsymbol{L}_e^{(0)} + \boldsymbol{L}_e^{(1)} + \cdots + \boldsymbol{L}_e^{(m-1)}}_{\text{from the current iteration}}$$

It is used together with channel observations $\boldsymbol{L}_c^{(m)}$ in the next APP decoding of the mth component code. A discussion on this global decoder can also be found in [Li-Bai, TIT2003].

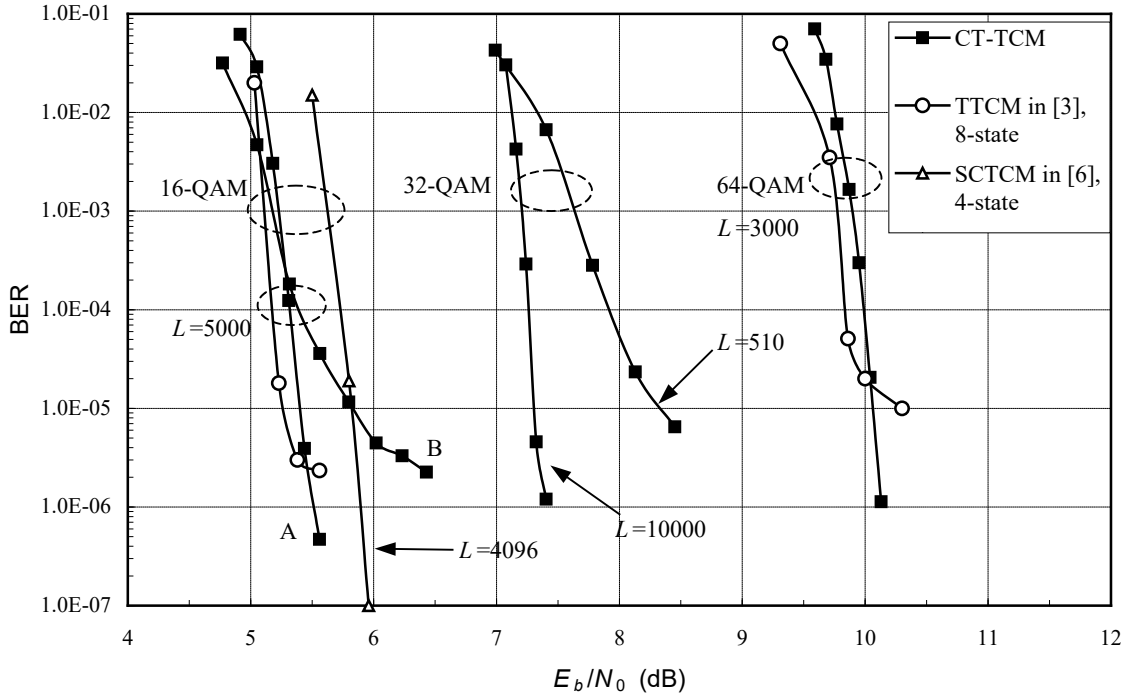


Fig. 5.9.10b BER Performance of CT-TCM schemes for different modulations. "L" represents interleaver size (in symbols). The spectral efficiencies for 16-QAM, 32-QAM and 64-QAM are 3, 4 and 5 bits/symbol, respectively. The corresponding constrained-capacity limits are 4.5dB, 6.8dB and 9.2dB, respectively. All the results are produced using 18 iterations except those cited from [6] (8 iterations).

5.9.4 PC-TCM

In this scheme, two parallel concatenated rate $k/(k+1)$ convolutional codes use all their information bits to produce the parity bit which is sent to their respective symbol mappers, but the information bits are punctured in such a fashion that one half of them are used in the first trellis code and the other half in the second trellis code [Benedetto-Divsalar96]. This is done to limit the number of points in the signal constellation to $2^{(k/2)+1}$. Also, the information bits transmitted by the two trellis encoders are interleaved by different bit interleavers (see Figure 5.9.11). In [Fragouli-Wesel01], a variation of this scheme is presented where a symbol interleaver is used instead of bit interleavers, resulting in improved BER performance in the waterfall region.

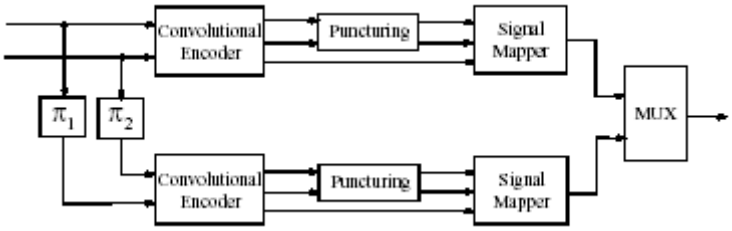
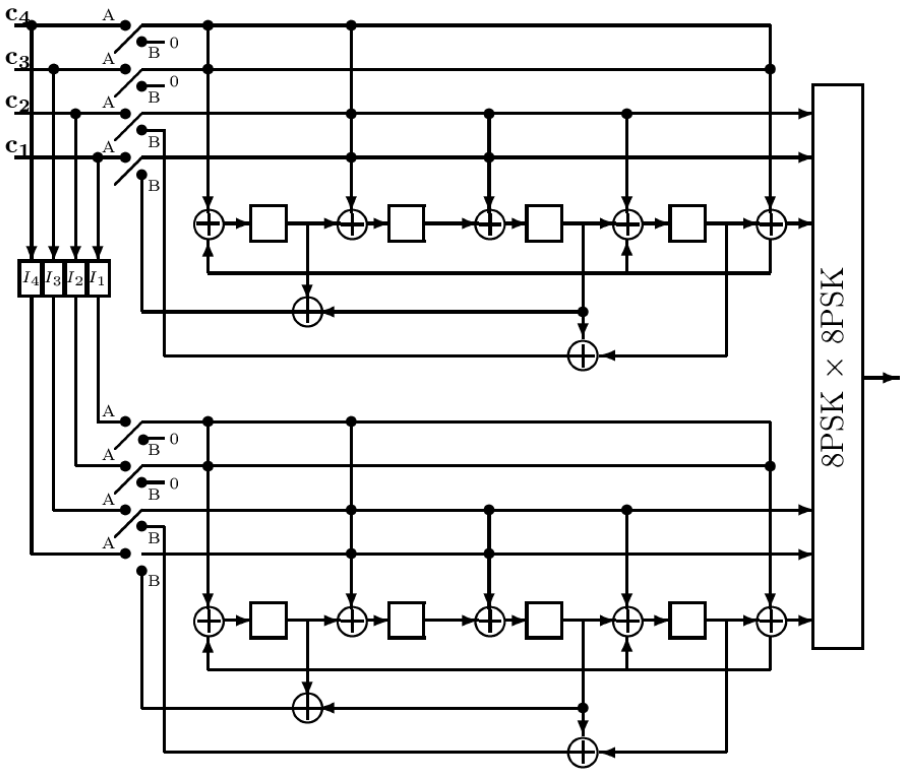


Figure 5.9.11



5.10 Multilevel Codes and Multistage Decoding

MLC was first proposed by Imai and Hirakawa in 1977, with the idea of protecting each address bit $c^{(i)}$ of the signal point by an individual binary code $\mathcal{C}_i$ at level i. The individual codes were chosen in such a way that the minimum diatance of the Euclidean space code was maximized. This is now known as balanced distance rule. At the receiver side, each code $\mathcal{C}_j$ is decoded individually starting from the lowest level and taking into account decisions of prior decoding stages. This is known as multistage decoding.

Compared with TCM, the MLC approach provides flexible transmission rates; any code (e.g., block codes, convolutional codes, or concatenated codes) can be used as component code. TCM can be viewed as a special case of MLC using a single convolutional code with a nonbinary output alphabet while higher levels remain uncoded.

5.10.1 Block coded modulation (BCM)

BCM is an example of multilevel coded modulation. 从 TCM 的 set-partitioning 可以看出，信号 label 中的不同 bit 要求不同的保护能力。在 BCM 中，we provide this by 用不

同纠错能力的码编不同 bit.具体地说，encode the LSB with a powerful code, and encode the MSB with the least powerful code (甚至不编码)。

将 label 排成一个阵列，forming columns with LSB at the top.

$$\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\\ \mathbf{c}\end{bmatrix}=\begin{bmatrix}a_0 & a_1 & \cdots & a_{n-1}\\ b_0 & b_1 & \cdots & b_{n-1}\\ c_0 & c_1 & \cdots & c_{n-1}\end{bmatrix}$$

$$\Rightarrow \mathbf{x}=\begin{bmatrix}x_0 & x_1 & \cdots & x_{n-1}\end{bmatrix}$$

Vectors **a**, **b**, and **c** should form 三个不同的长为 n 的码。

As an example, Fig. 5.10.1 shows a 3-level BCM code over the 8-PSK signal set.

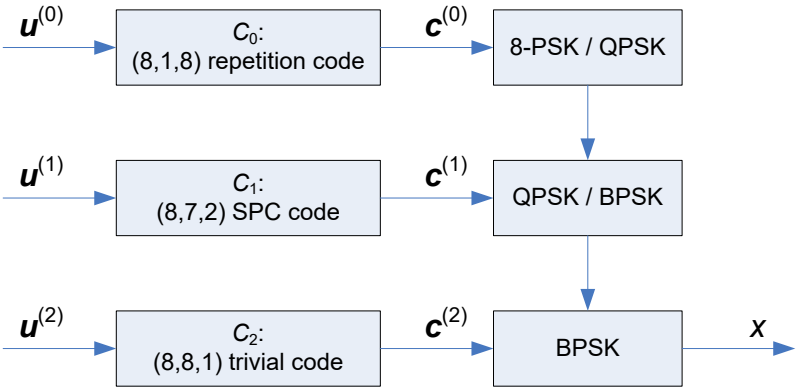


Fig. 5.10.1

An l-level BCM code over a signal set (M-PSK or M-QAM) with 2^l signal points is constructed in the same manner as a 3-level 8-PSK BCM code. For $0\le i<l$, let $\mathcal{C}_i$ be a binary (n, k_i, d_i) linear block code of length n, dimension k_i, and minimum Hamming distance d_i. Let

$$\begin{aligned}\mathbf{c}^{(0)}&=\left(c_0^{(0)},c_1^{(0)},...,c_{n-1}^{(0)}\right)\\ \mathbf{c}^{(1)}&=\left(c_0^{(1)},c_1^{(1)},...,c_{n-1}^{(1)}\right)\\ &\vdots\\ \mathbf{c}^{(l-1)}&=\left(c_0^{(l-1)},c_1^{(l-1)},...,c_{n-1}^{(l-1)}\right)\end{aligned}$$

be l codewords in $\mathcal{C}_0$, $\mathcal{C}_1$,…, $\mathcal{C}_{l-1}$, respectively. Forming the following sequence

$$\mathbf{c}^{(0)}*\mathbf{c}^{(1)}*\cdots*\mathbf{c}^{(l-1)}=\left(c_0^{(0)}c_0^{(1)}\cdots c_0^{(l-1)},...,c_j^{(0)}c_j^{(1)}\cdots c_j^{(l-1)},...,c_{n-1}^{(0)}c_{n-1}^{(1)}\cdots c_{n-1}^{(l-1)}\right)$$

For $0\le j<n$, we take $c_j^{(0)}c_j^{(1)}\cdots c_j^{(l-1)}$ as the label of a signal point $a\in\mathcal{X}$. Then

$$\begin{aligned}f\left(\mathbf{c}^{(0)}*\mathbf{c}^{(1)}*\cdots*\mathbf{c}^{(l-1)}\right)&=\left(f\left(c_0^{(0)}c_0^{(1)}\cdots c_0^{(l-1)}\right),...,f\left(c_j^{(0)}c_j^{(1)}\cdots c_j^{(l-1)}\right),...,f\left(c_{n-1}^{(0)}c_{n-1}^{(1)}\cdots c_{n-1}^{(l-1)}\right)\right)\\ &=\left(x_0,x_1,...,x_{n-1}\right)\end{aligned}$$

is a sequence of n signals, with $x_j = f\left(c_j^{(0)} c_j^{(1)} \cdots c_j^{(l-1)}\right) \in \mathcal{X}$. Then

$$\mathcal{C} \triangleq f\left(\mathcal{C}^{(0)} * \mathcal{C}^{(1)} * \cdots * \mathcal{C}^{(l-1)}\right) = \left\{ f\left(\mathbf{c}^{(0)} * \mathbf{c}^{(1)} * \cdots * \mathbf{c}^{(l-1)}\right) \middle| \mathbf{c}^{(i)} \in \mathcal{C}_i \text{ for } 0 \le i < l \right\}$$

is an l-level BCM code over $\mathcal{X}$ with dimension $k = k_0 + k_1 + \cdots + k_{l-1}$.

5.10.2 MLC

MLC is BCM 的推广。分量码不必是等长的二进制线性分组码，也可以是卷积码、级联码甚至非二进制码。它的编码器的一般结构如图 5.10.2 所示。A generic MLC encoder is shown in Fig.5.10.2. A block of K binary source data symbols, $\mathbf{u} = (u_1, u_2, ..., u_K), u_k \in \{0,1\}$, is partitioned into l blocks:

$$\mathbf{u}^{(i)} = \left(u_1^{(i)}, u_2^{(i)}, ..., u_{K_i}^{(i)}\right), i = 0, ...l-1$$

of length K_i with $\sum_{i=0}^{l-1} K_i = K$. Each data block $\mathbf{u}^{(i)}$ is fed into an individual binary encoder ENC$_i$, producing codewords $\mathbf{c}^{(i)} = \left(c_0^{(i)}, ..., c_{n-1}^{(i)}\right)$ of the component code $\mathcal{C}_i$. For simplicity, we have assumed equal code lengths n at all levels, but in principle the choice of component codes is arbitrary. The code symbols $\mathbf{c}_j = \left\{c_j^{(i)}, 0 \le i < l\right\}$ at time j is mapped to the signal constellation, producing the transmitted signal $x_j = f(\mathbf{c}_j) = f\left(c_j^{(0)} c_j^{(1)} \cdots c_j^{(l-1)}\right) \in \mathcal{X}$.通常，调制器使用 Ungerboeck's mapping by set-partitioning 完成信号的映射。

The code rate of $\mathcal{C}$ is given by

$$R = \sum_{i=0}^{l-1} R_i = \sum_{i=0}^{l-1} \frac{K_i}{n} = \frac{K}{n} \tag{5.10.1}$$

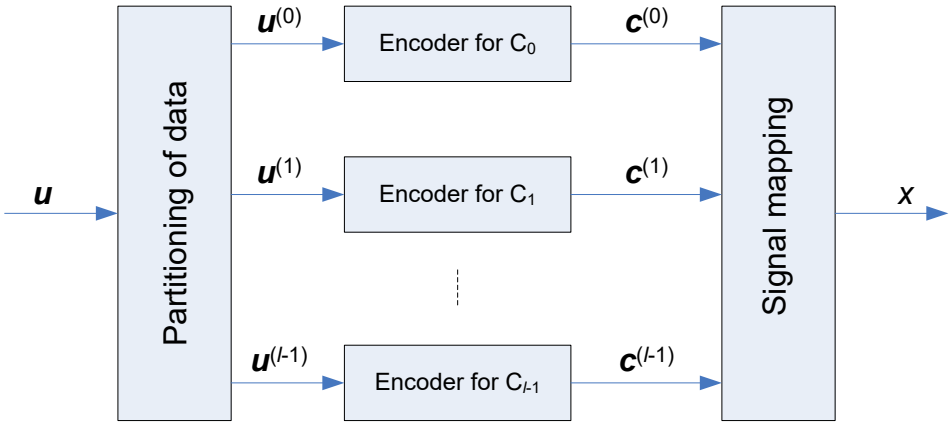


Figure 5.10.2. MLC encoder

■ Information-theoretic considerations: MLC approach（together with its MSD procedure）is a straightforward consequence of the chain rule of mutual information.

Consider a MLC scheme with l levels. Assume that a binary code is used as a component code at each level, and 调制器使用 Ungerboeck's 子集划分方法进行二进制 label 向量 $\mathbf{c}=\left(c^{(0)},c^{(1)},\cdots,c^{(l-1)}\right)$ 到信号点 x 的映射。

At partition level i, the subset $\mathcal{X}\left(c^{(0)}...c^{(i-1)}\right)$ are iteratively divided into two further subsets $\mathcal{X}\left(c^{(0)}...c^{(i-1)}0\right)$ and $\mathcal{X}\left(c^{(0)}...c^{(i-1)}1\right)$ at level i+1:

$$\mathcal{X}\left(c^{(0)}...c^{(i-1)}c^{(i)}\right)=\left\{x=f(\mathbf{c})\middle|\mathbf{c}=\left(c^{(0)}...c^{(i)}b^{(i+1)}...b^{(l-1)}\right),b^{(j)}\in \mathrm{GF}(2),i+1\le j<l\right\}$$

Since the mapping $f(.)$ is bijective independently of the actual partitioning strategy, the mutual information $\mathcal{I}(\mathcal{X};Y)$ between the transmitted signal point $x\in\mathcal{X}$ and the received signal point $y\in\mathcal{Y}$ is equal to the mutual information $I\left(c^{(0)}...c^{(l-1)};Y\right)$ between the label vector $\mathbf{c}\in\{0,1\}^l$ and the received signal point $y\in\mathcal{Y}$.

From the chain rule for mutual information [Gallager68], we have

$$\begin{aligned}\mathcal{I}\left(\mathcal{X};Y\right)&=\mathcal{I}\left(c^{(0)}...c^{(l-1)};Y\right)\\&=\mathcal{I}\left(c^{(0)};Y\right)+\mathcal{I}\left(c^{(1)};Y\middle|c^{(0)}\right)+\cdots+\mathcal{I}\left(c^{(l-1)};Y\middle|c^{(0)}...c^{(l-2)}\right)\end{aligned}\qquad(5.10.2)$$

This implies that transmission of vectors with binary digits $c^{(i)},0\le i<l$, over the physical channel can be separated into the parallel transmission of individual digits $c^{(i)}$ over l equivalent channels, provided that $c^{(0)}...c^{(i-1)}$ are known. This 基本原理 is illustrated in Fig.5.10.3 .

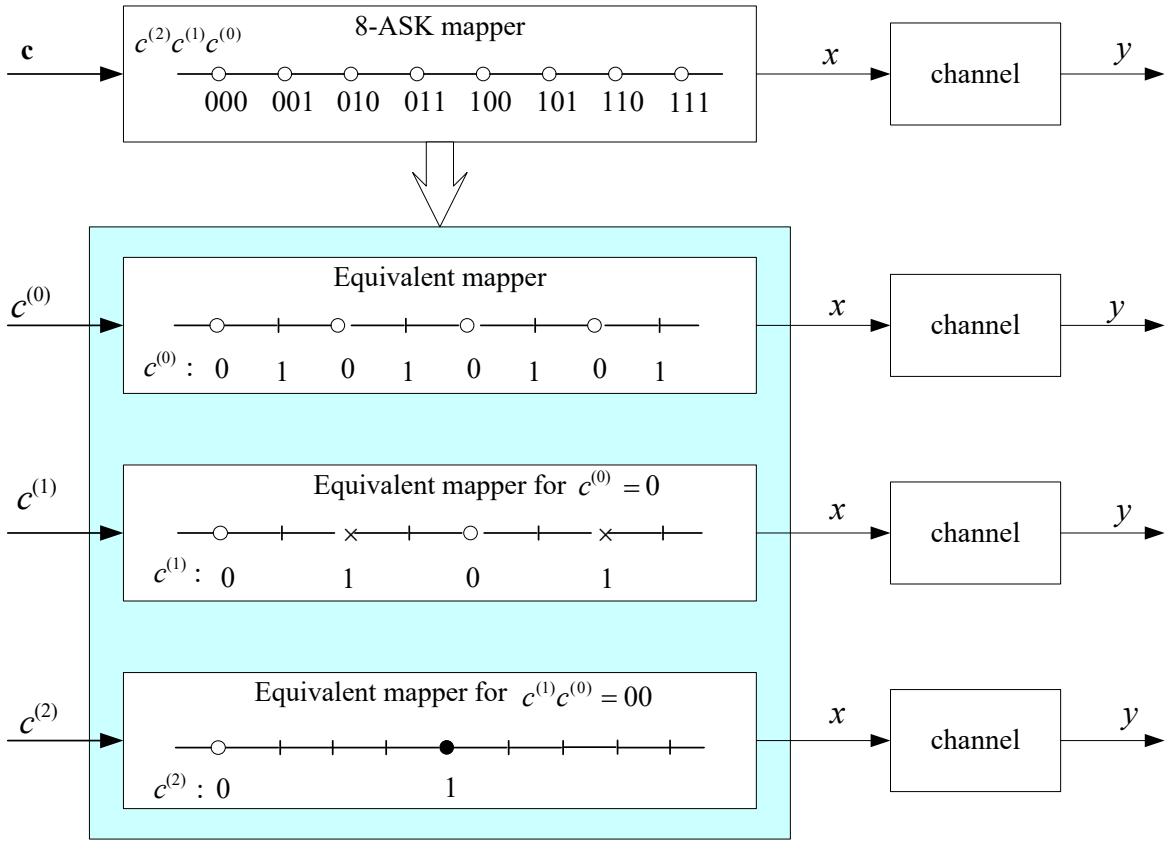


Figure 5.10.3

From the chain rule, the mutual information of the equivalent channel *i* can be calculated as

$$\mathcal{I}\left(c^{(i)};Y\middle|c^{(0)}...c^{(i-1)}\right)=\mathcal{I}\left(c^{(i)}...c^{(l-1)};Y\middle|c^{(0)}...c^{(i-1)}\right)-\mathcal{I}\left(c^{(i+1)}...c^{(l-1)};Y\middle|c^{(0)}...c^{(i)}\right)$$

where $\mathcal{I}\left(c^{(i)}...c^{(l-1)};Y\middle|c^{(0)}...c^{(i-1)}\right)$ is calculated by averaging over all possible combinations of $c^{(0)}...c^{(i-1)}$. (在第 *i* 层的所有信号子集上求平均)。即

$$\mathcal{I}\left(c^{(i)}...c^{(l-1)};Y\middle|\underbrace{c^{(0)}...c^{(i-1)}}_{\text{Random variables}}\right)=\mathsf{E}_{c^{(0)}...c^{(i-1)}}\left\{\mathcal{I}\left(c^{(i)}...c^{(l-1)};Y\middle|\underbrace{c^{(0)}...c^{(i-1)}}_{\text{Outcome}}\right)\right\}$$

等效信道的概念是分析与设计编码调制方案的基本工具。

Thus, the capacity of the equivalent channel is given by

$$C^{(i)}=\mathcal{I}\left(c^{(i)};Y\middle|c^{(0)}...c^{(i-1)}\right)=\mathsf{E}_{c^{(0)}...c^{(i-1)}}\left[C\left(\mathcal{X}\left(c^{(0)}...c^{(i-1)}\right)\right)\right]-\mathsf{E}_{c^{(0)}...c^{(i)}}\left[C\left(\mathcal{X}\left(c^{(0)}...c^{(i)}\right)\right)\right]$$

where $C\left(\mathcal{X}\left(c^{(0)}...c^{(i)}\right)\right)$ denotes the capacity when using only the subset $\mathcal{X}\left(c^{(0)}...c^{(i)}\right)$ with a priori probabilities $P(a)/P\left\{\mathcal{X}\left(c^{(0)}...c^{(i)}\right)\right\}$.

The capacity $C^{(i)}$ 将作为第 *i* 层分量码的码率设计依据。The total capacity C of a

multilevel coded modulation scheme is then

$$C = \sum_{i=0}^{l-1} C^{(i)}$$

■ Design of multilevel codes:

The essential point for the design of a MLC scheme is the assignment of code rates to the individual coding levels. In [Huber99], fives rules are introduced for rate design. For the capacity design rule, the individual rates are chosen to be equal to the capacities of the equivalent channels, i.e.,

$$R_c^{(i)} = C^{(i)}, \quad i = 0,1,...,l-1$$

For the choice of FEC codes, powerful probabilistic coding methods such as turbo codes and LDPC codes are really needed only at the lower levels. At the higher levels, the channels become quite clean and the capacity $C(\Lambda_{i-1}/\Lambda_i)$ approaches $\log_2(\Lambda_{i-1}/\Lambda_i)$, so that the desired redundancy approaches zero. For these levels, algebraic codes and decoding methods may be more appropriate. The state of the art in multilevel coding is reviewed in [Wachsmann99]. Figures 5.10.4-5.10.6 depict the code rate design based on the capacity rule for 8-ASK and 8-PSK MLC schemes.

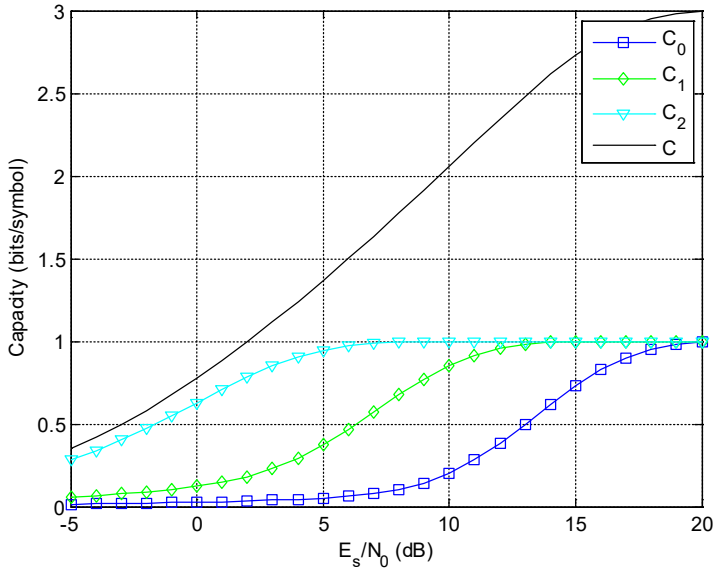
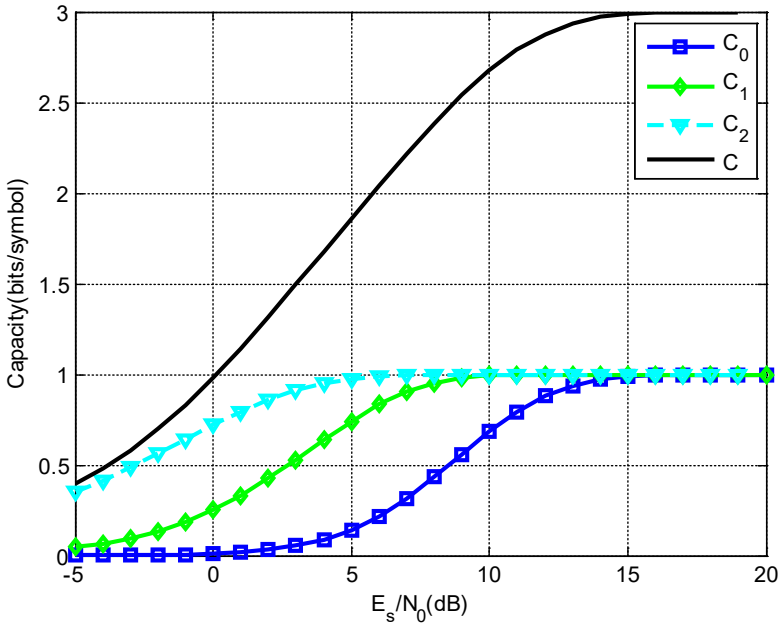
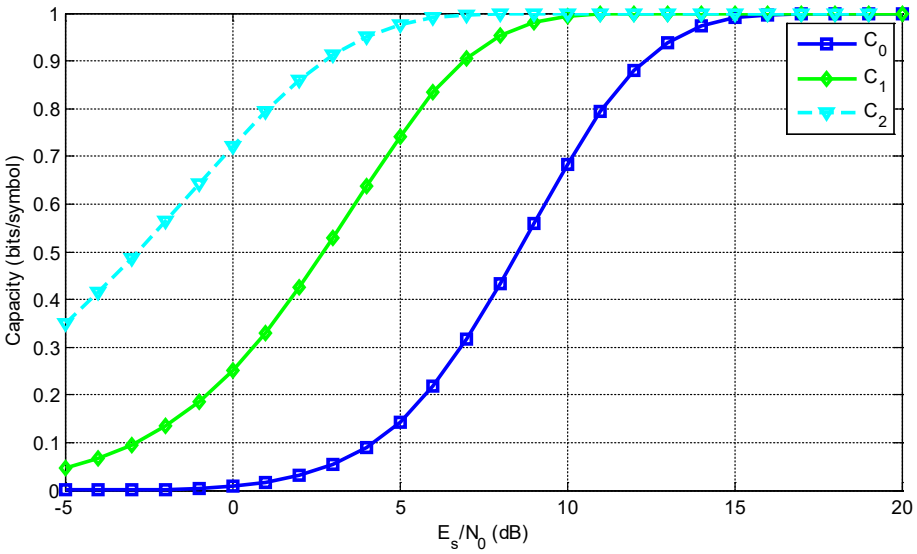


Fig. 5.10.4 采用 8-ASK 的 MLC 系统各层容量



(a)



(b)

Fig. 5.10.5 Capacity of partitioned subsets for 8-PSK modulation (with Ungerboeck partitioning)

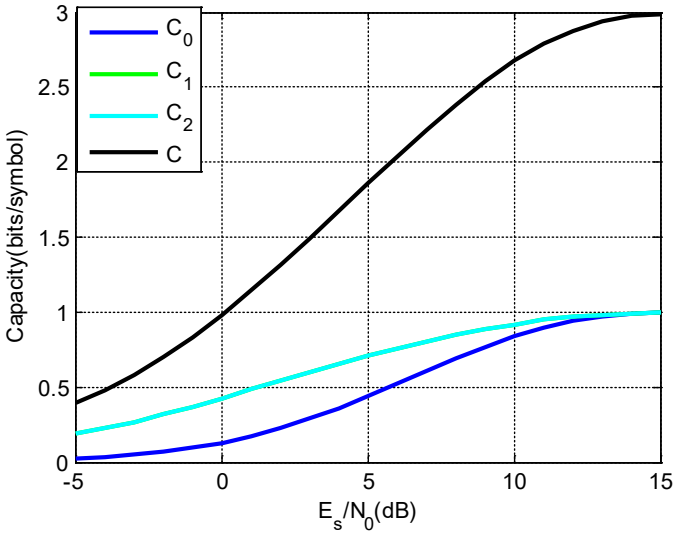


Fig. 5.10.6 Capacity of 8-PSK MLC with Gray mapping (in the figure, C_1=C_2)

5.10.3 多级译码 (MSD)

The component codes $\mathcal{C}_i$ are successively decoded by the corresponding decoder DEC-*i*. At stage *i*, decoder DEC-*i* processes not only the block $\mathbf{y} = (y_0, ..., y_{n-1}) \in \mathcal{Y}^N$ of received signal points, but also decision $\hat{c}^{(j)}, 0 \le j < i-1$ of previous decoding stages *j*. As long as error-free decisions $\hat{c}^{(j)} = c^{(j)}$ are generated by the decoder DEC-*j*, MSD can be interpreted as an implementation of the chain rule.

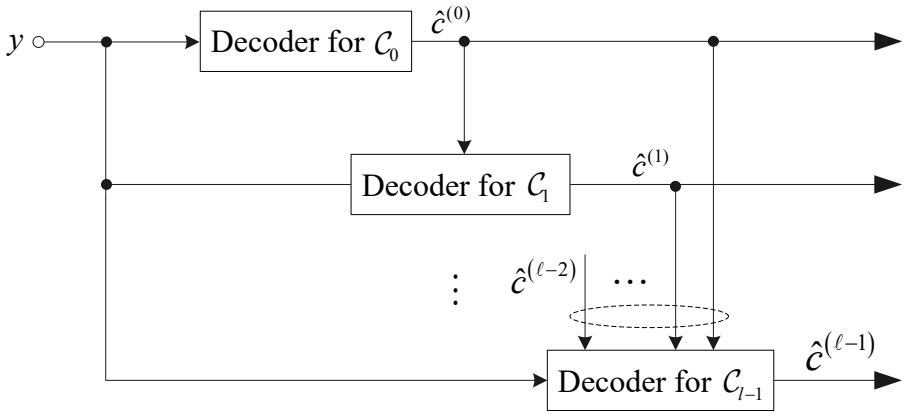


Figure 5.10.7

Note that 尽管 capacity 是相同的 for 不同 labeling strategies, 但是对于 MLC/MSD with finite code length, 不同 labeling strategies（Ungerboeck partitioning, mixed partitioning, Gray labeling）会影响其 BER 性能。In practice, we recommend MLC schemes based on Ungerboeck set partitioning, where powerful codes with soft-decision decoding are employed

only at lower levels and hard-decision decoding is applied at higher levels.

5.10.4 多层码的迭代译码

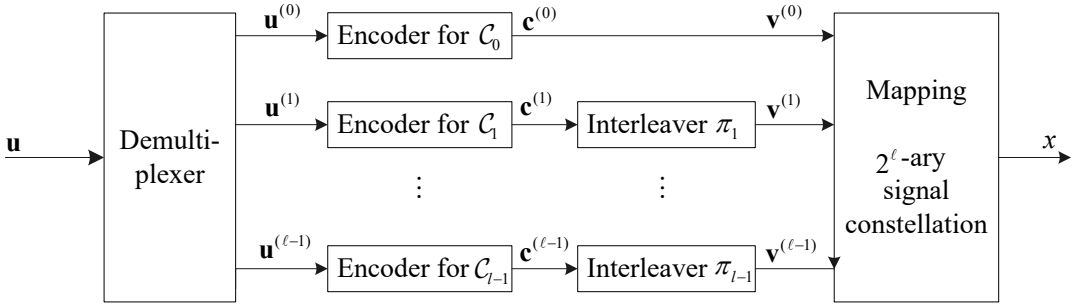
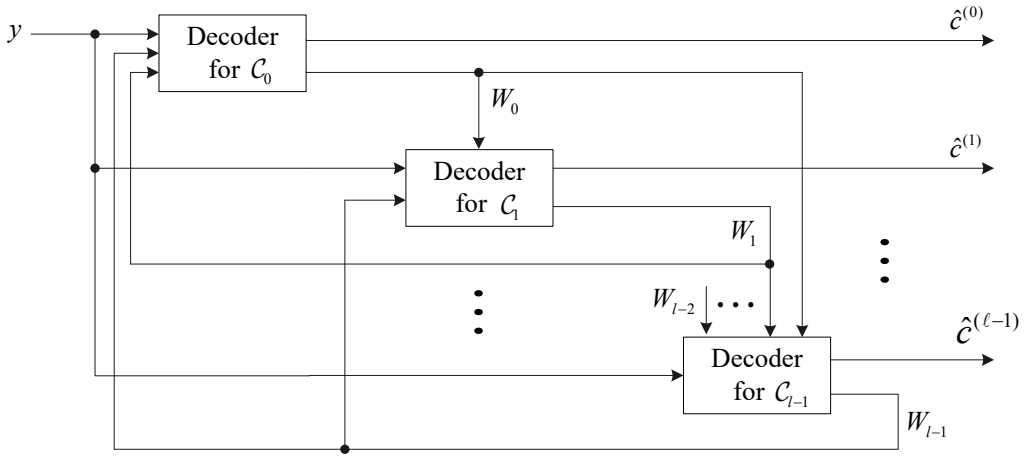
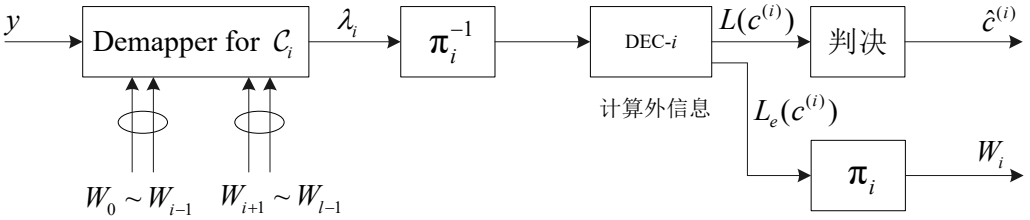


Figure 5.10.8 Encoder



（a）迭代多级译码器



（b） 第 *i* 级分量译码器结构

Figure 5.10.9 MLC 的迭代译码

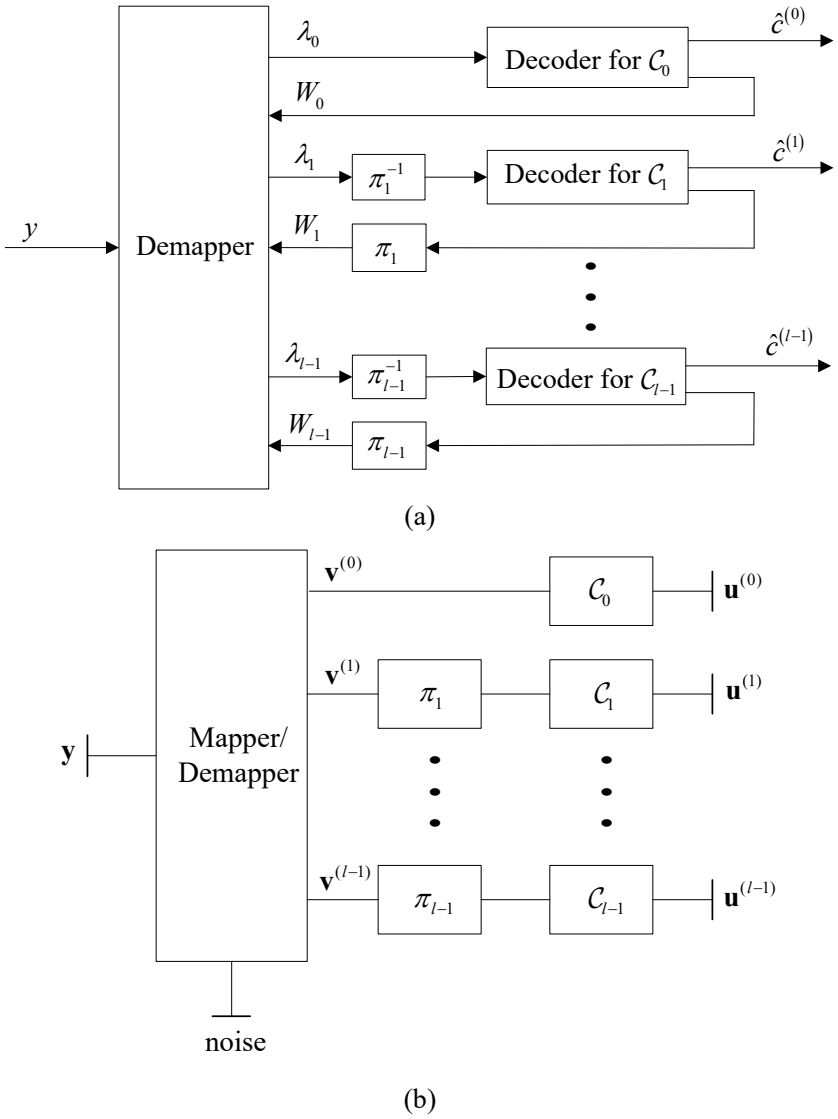


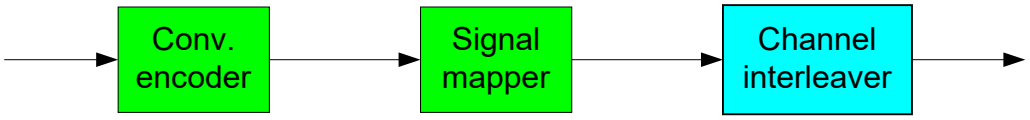
Figure 5.10.10 将各级的 demapper 综合在一起的 Decoder (a)译码器结构框图 （b）Normal graph

5.11 Bit-Interleaved Coded Modulation (BICM) with Iterative Decoding

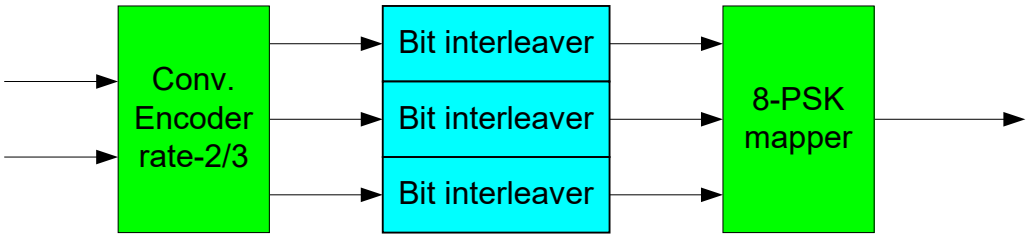
BICM was first developed by Zehavi primarily for fading channels with the aim of increasing the diversity order of Ungerboeck's TCM schemes, but has turned out to be capable of approaching the capacity of high-SNR AWGN channels as well. Fig. 5.11.1 depicts the coded modulation schemes for fading channels. In Fig. 5.11.1 (a), the channel interleaver is of symbol-based, which can also be placed between the convolutional encoder and signal mapper. The symbols are interleaved with a depth exceeding the coherent time of the fading process. Fig. (b) is the original BICM scheme proposed by Zehavi. Compared with the TCM encoder in (a), the BICM in (b) uses independent bit interleavers for all the bits of a symbol rather than a single symbol-based interleaver. The purpose of bit interleaving is:

- To disperse the bursty errors induced by the correlated fading and to maximize the diversity order of the system;
- To render the bits associated with a given transmitted symbol uncorrelated or independent of each other.

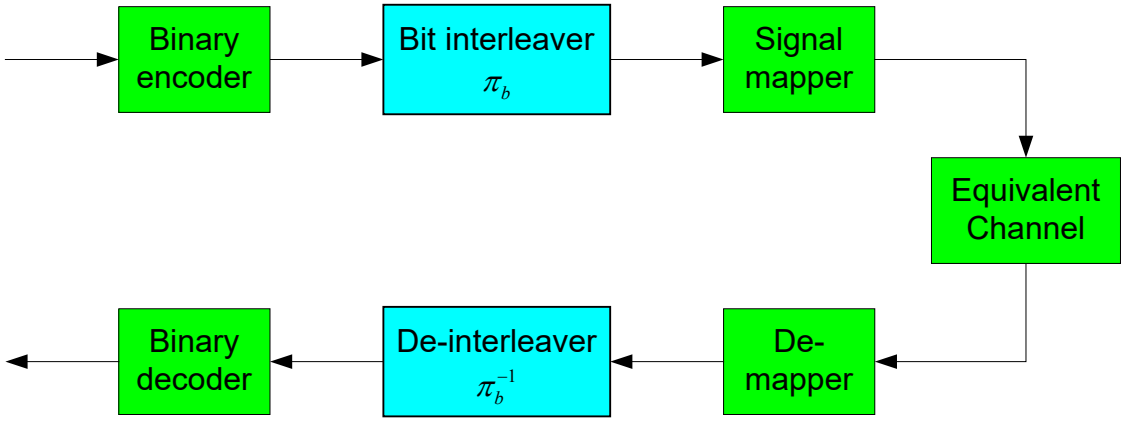
Since in BICM the bit-based minimum Hamming distance (the number of different bits between the erroneous path in the shortest error event and the correct path) is maximized, BICM will give a lower BER in Rayleigh fading channels than that of TCM that maximizes the FED. In other words, for the same complexity BICM achieves better code diversity (i.e., the minimum Hamming distance), while TCM achieves better minimum Euclidean distance.
In fact, the three separate bit interleavers of size K can be replaced by a single bit-interleaver with size of $3K$, resulting in a generic BICM scheme as shown in Fig. 5.11.1 (c). In general, a BICM transmitter comprises an encoder for a binary code $\mathcal{C}$, a "bit interleaver," and a signal mapper. The output sequence of the bit interleaver is segmented into m-bit blocks, which are mapped to a 2^m-point signal constellation using Gray coding. The BICM decoder implements the inverse process, as shown in Fig. 5.11.1 (c). Good performance close to capacity can be obtained if $\mathcal{C}$ is a long Turbo code or LDPC code.



(a) TCM scheme for fading channels



(b) Zehavi's BICM for 8-PSK



(c) A general BICM scheme

Figure 5.11.1

5.11.1 BICM-ID

BICM can be viewed as a special type of SCCC, and the "inner code" is the symbol mapper, which can be considered as a nonrecursive single-state encoder. So the iterative

decoding method for SCCC can be applied to BICM.

BICM with iterative decoding (BICM-ID) was proposed by Li and Ritcey [Li98] for further improving the FED of Zehavi's BICM scheme. This FED improvement can be achieved with the aid of combining SP(set partitioning)-based constellation labeling, as in TCM, and by invoking soft-decision feedback from the decoder's output to the demodulator's input, in order to exchange soft-decision based information between them.

The BICM-ID scheme using soft-decision feedback is shown in Fig. 5.11.2, where soft-in soft-out (SISO) detector/decoder is used in the receiver modulo. Let $\mathcal{C}$ be a binary code. Assume that it is concatenated with an N-dim memoryless modulator over a signal set $\mathcal{X} \subseteq \mathbb{C}^N$ of size $|\mathcal{X}| = M = 2^m$, through a bit interleaver π and a one-to-one binary labeling map $f : \{0,1\}^m \to \mathcal{X}$. Denote the code sequence by $\mathbf{c}$. The interleaved sequence $\pi(\mathbf{c})$ is segmented into m-bit blocks denoted by $\mathbf{c}_t = \left(c_t^{(0)}, c_t^{(1)}, ..., c_t^{(m-1)}\right)$, which are mapped into signal points $x_t = f(\mathbf{c}_t)$ with $\mathbf{c}_t$ as the label of x_t. The signal x_t are then transmitted over the channel at time t, resulting in the received signal y_t.

Let x denote a signal point in $\mathcal{X}$, and $b_i(x)$ the ith bit of the label of x. With the use of iterative demodulation/decoding at the receiver, the demapper can use the MAP algorithm to compute APPs for each label bit of x_t and deliver them to the decoder; i.e.,

$$P\left(c_t^{(i)} = b \mid y_t\right) = \sum_{x_t \in \mathcal{X}_b^{(i)}} P(x_t \mid y_t) \propto \sum_{x_t \in \mathcal{X}_b^{(i)}} p(y_t \mid x_t) P(x_t), \quad \text{for } i=1\ldots,m$$

where

$$\mathcal{X}_b^{(i)} = \left\{x \in \mathcal{X} \mid b_i(x) = b\right\} = \left\{x_t = f(\mathbf{c}_t) \in \mathcal{X} \mid c_t^{(i)} = b\right\}$$

is the subset of all the signals $x \in \mathcal{X}$ whose label has the value $b \in \{0,1\}$ in position i, and $P(x_t)$ is the *a priori* probability of transmitting x_t. At the first iteration, $P(x_t)$ is initialized to be $1/M$ for all values of x_t. 对于第二次及以后各次迭代，

$$P(x_t) = \prod_{j=0}^{m-1} P\left(c_t^{(j)} = b_j(x_t)\right) \qquad \text{(with 充分长的交织器)}$$

考虑到交织/解交织，在图 5.11.2 中即为

$$P(x_t) = \prod_{j=0}^{m-1} P\left(v_t^{(j)} = b_j(x_t); I\right)$$

Therefore,

$$P\left(c_t^{(i)} = b \mid y_t\right) = \sum_{x_t \in \mathcal{X}_b^{(i)}} P(y_t \mid x_t) \prod_{j=0}^{m-1} P\left(c_t^{(j)} = b_j(x_t)\right)$$

在迭代解调中，demapper 送给译码器的外信息为

$$P\left(c_t^{(i)}=b;O\right)=\frac{P\left(c_t^{(i)}=b\mid y_t\right)}{P\left(c_t^{(i)}=b;I\right)}=\sum_{x_t\in\chi_b^{(i)}}\left(P(y_t\mid x_t)\prod_{\substack{j=0\\ j\neq i}}^{m-1}P\left(c_t^{(j)}=b_j(x_t)\right)\right)$$

它被译码器看作输入的先验信息。在对数域中，它可以表示为

$$L_e^{demapper}\left(c_t^{(i)}=b\right)=\log P\left(c_t^{(i)}=b\mid y_t\right)-\log P\left(c_t^{(i)}=b;I\right)$$

The SISO decoder for FEC code is used for generating the APPs $P\left(u_t^{(i)};O\right)$ of the information bits, as well as the extrinsic APPs $P\left(c_t^{(i)};O\right)$ （应该是纯外信息，等于 $APP(c_t^{(i)}\mid y)/P(c_t^i;I)$，不包含信道值）of the coded bits, from the deinterleaved probabilities $P\left(v_t^{(i)}=b;O\right)$. It can be implemented based on the MAP, Max-Log- MAPA and other such algorithms.

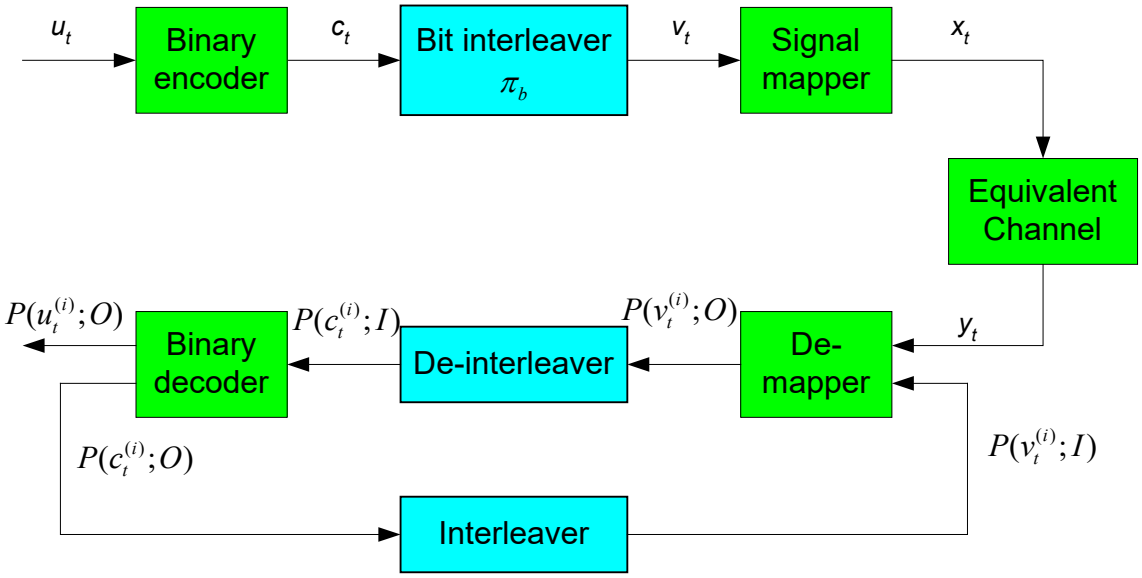


Figure 5.11.2. BICM-ID scheme using soft-decision feedback

■ Labeling method

It should be pointed that labeling plays a key role in BICM. The performance of BICM depends on the labeling used. Fig.x shows the process of subset partitioning for each of the three bit positions for both Gray labeling and SP labeling. It is seen that Gray labeling has a lower number of nearest neighbours compared to SP-based labeling. Hence, Gray labeling is a more appropriate mapping during the first decoding iteration, and so it is adopted by the non-iterative BICM scheme of Fig.x. With iterative decoding and SP-based labeling, BICM-ID can achieve a high diversity order as well as a high FED.

By using the density evolution approach for BICM, it is demonstrated that there is no one mapper having the best performance over all SNR regions [Tan05, IEEE Trans. Wireless,

vol.4, no.2, 2005].

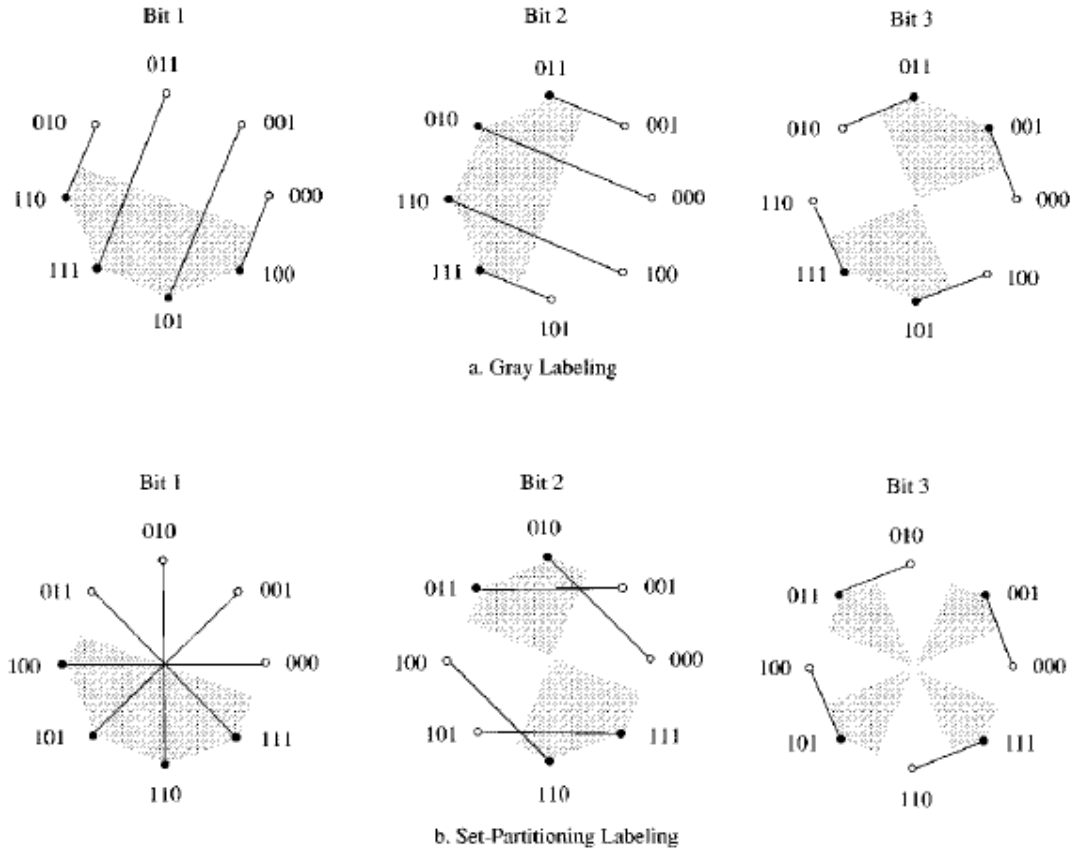
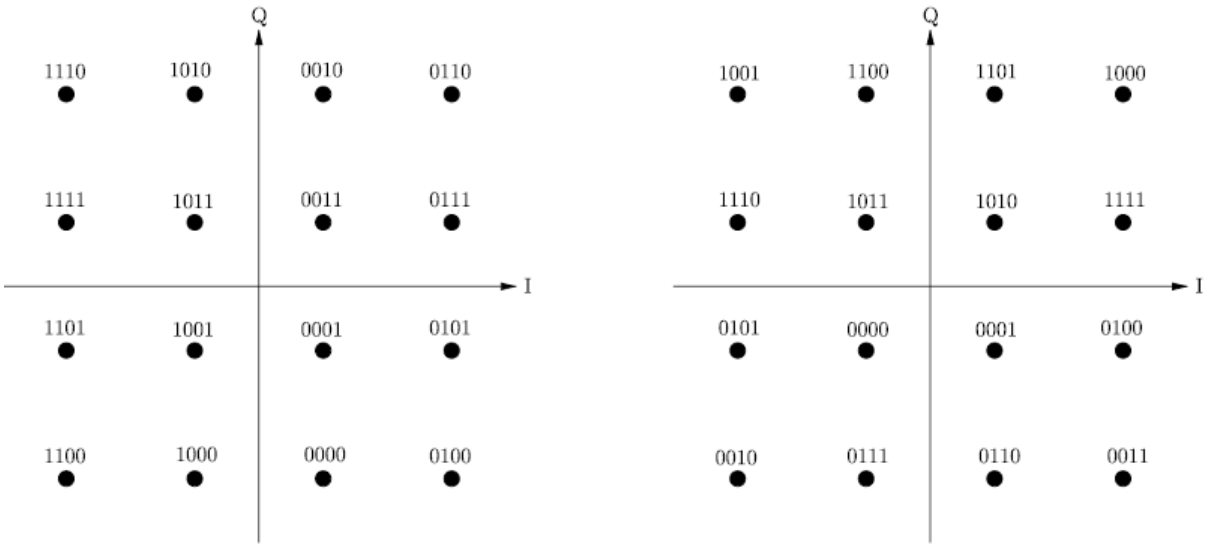


图 5.11.4 给出了 16-QAM 星座的 binary labeling rule.



(a) Gray (b) Set partitioning

Figure 5.11.4 Binary labeling rules for 16-QAM

■ Interleaver design: randomly generated.

5.11.2 More About BICM: Information-theoretic aspects [Gallager68, Caire98]

In Chapter 6 of [Gallager68], Gallager proved a coding theorem for parity-check codes,

which may be addressed as follows.

■ ***Theorem:*** *Binary linear codes can be used to achieve capacity on an arbitrary discrete memoryless channel.*

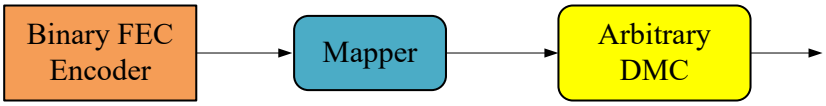


Figure 5.11.5

Gallager also demonstrated a simple algorithm to generate codewords with the probability distributions required to achieve the results of the coding theorem. However, the problem is that finding decoding algorithm is not simple.

Fig. 5.11.6 shows an example, where each channel codeword is a sequence of N ternary independent digits with the probabilities $Q(a)$=3/8, $Q(b)$=3/8 and $Q(c)$=2/8. We will call it Gallager mapping later.

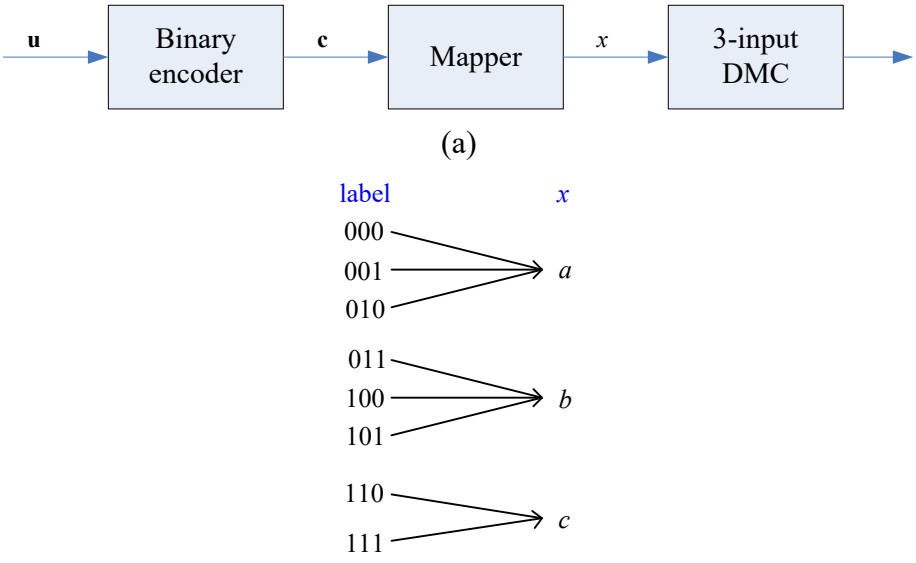


Figure 5.11.6 An example for use of binary codes on a DMC. (a) System model. (b) Mapping rule

Now, BICM-ID with the use of Turbo-like codes provides an effective solution. Joint demapping-decoding greatly reduces the decoding complexity.

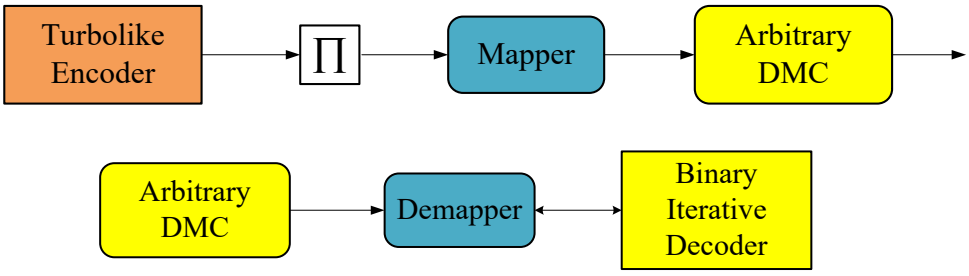


Figure 5.11.7

Caire *et al.* provided in [IT1998] a comprehensive analysis of BICM in terms of information rates and error probability, showing that in fact the loss incurred by the BICM

interface may be very small. Furthermore, this loss can essentially be recovered by using iterative decoding.

Coded-modulation capacity for various signal constellation in the AWGN channel has been discussed in Chapter 2. Here, we briefly review the results on the achievable rates for BICM by following Caire.

Define the sets

$$\mathcal{X}_{b_{j_1},\ldots,b_{j_l}}^{(i_{j_1}\ldots i_{j_l})} = \left\{ x \in \mathcal{X} \mid b_{i_{j_1}}(x) = b_{j_1}, \ldots, b_{i_{j_l}}(x) = b_{j_l} \right\}$$

as the subsets of signal points having the l binary labels $b_{j_1},\ldots,b_{j_l}$ in positions $i_{j_1}\ldots i_{j_l}$.

Under the assumption of an infinite-length interleaver, capacity and cutoff rate were studied in [Caire98]. This assumption yields a set of m independent parallel binary-input channels, for which the corresponding mutual information is the sum of the corresponding rates of each subchannel, and are given by

$$I^{\text{ind}}(X;Y) = \sum_{i=1}^{m} \mathsf{E}\left[\log \frac{\sum_{x' \in \mathcal{X}_B^i} P_{Y|X}(Y \mid x')}{\frac{1}{2}\sum_{x' \in \mathcal{X}} P_{Y|X}(Y \mid x')} \right]$$

$$= m - \sum_{i=1}^{m} \mathsf{E}_{B,Y}\left[\log \frac{\sum_{x' \in \mathcal{X}} P_{Y|X}(Y \mid x')}{\sum_{x' \in \mathcal{X}_B^i} P_{Y|X}(Y \mid x')} \right]$$

where B is a random variables taking values on $\{0,1\}$. The joint distribution $p(B, Y)$ of the channel-i is

$$p(B,Y \mid i) = 2^{-m} \sum_{x \in \mathcal{X}_B^i} p_{Y|X}(y \mid x)$$

The BICM capacity is given by

$$C^{\text{BICM}} \equiv I^{\text{ind}}(X;Y)$$

Since BICM can be regarded as MLC with (Gray mapping and) parallel independent decoding (i.e., decoder at level i makes no use of decisions of other levels $j \neq i$), it is shown that [Wachsmann-Huber99]

$$C^{\text{BICM}} = \sum_{i=1}^{m} \frac{1}{2} \sum_{b=0}^{1} \left(C_{\mathcal{X}}^{\text{CM}} - C_{\mathcal{X}_b^{(i)}}^{\text{CM}} \right)$$

where $C_{\mathcal{A}}^{\text{CM}}$ is the capacity (information rate) for coded modulation over a general constellation $\mathcal{A}$.

The BICM (或MLC/PDL) approach is simply a suboptimum approximation of an optimum coded modulation scheme, its capacity strongly depends on the particular labeling of signal points. Caire *et al*. showed that the gap to an optimum scheme is surprisingly small with Gray labeling of the signal points. However, it has been illustrated that the gap increases significantly if the finite length of the code is taken into account. 码长越短，性能差别越大。

Figure 5.11.8 shows the BICM mutual information for some signal constellations, different binary labeling rules and uniform inputs for the AWGN channel, as a function of SNR. For illustration simplicity, we have only plotted the information rate for the Gray and set partitioning binary labeling rules from Figure x. Observe that binary reflected Gray labeling pays a negligible penalty in information rate, being close to the coded modulation capacity.

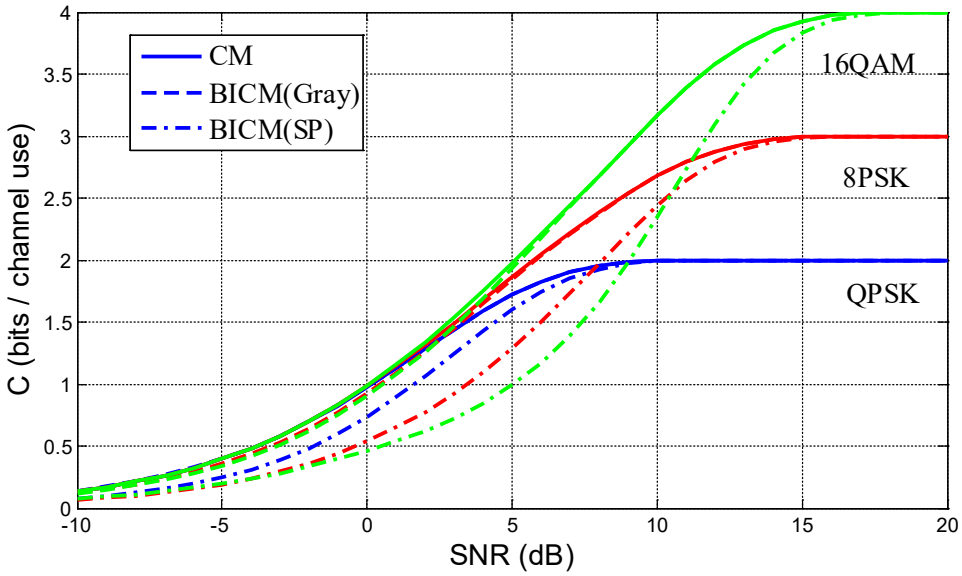


Fig. 5.11.8 Coded modulation and BICM capacities (in bits per channel use) for multiple signal constellations with uniform inputs in the AWGN channel. Gray and set partitioning labeling rules correspond to dashed and dash-dotted lines, respectively. The thick solid lines indicate CM channel capacity.

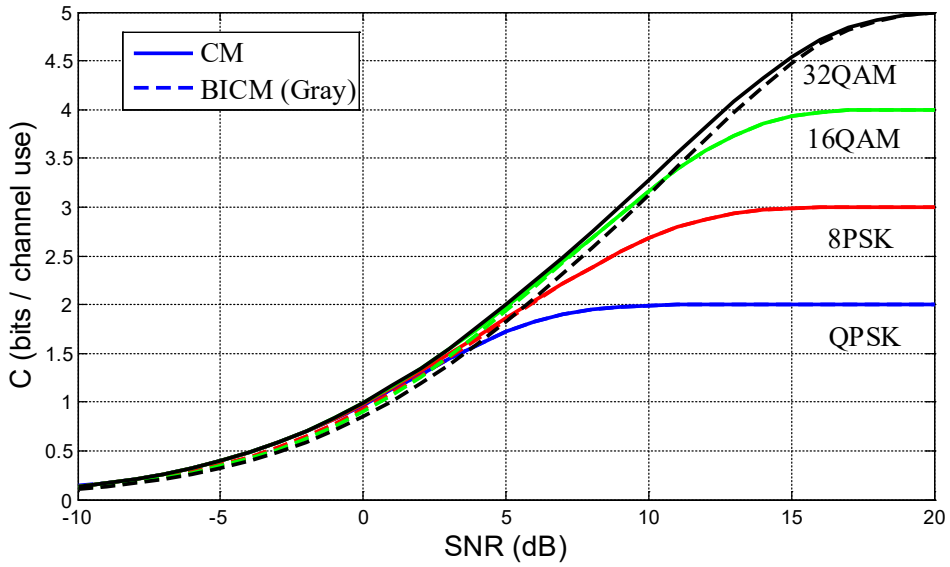


Fig. 5.11.9

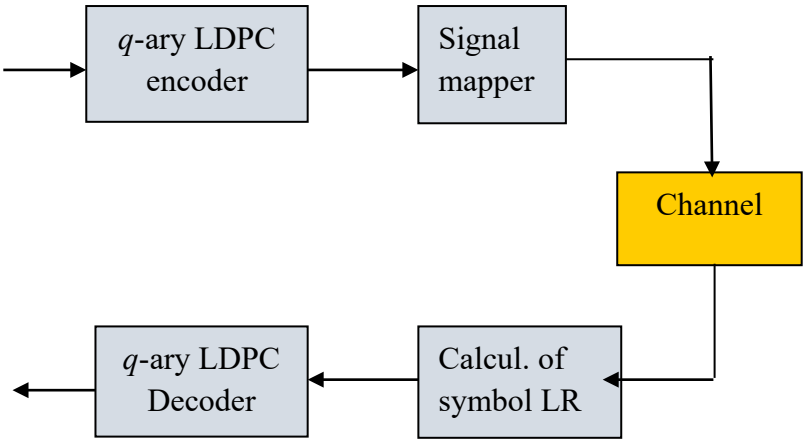
Comparison with Multilevel Coding:

While MLC with MSD achieves the coded modulation capacity, it does not achieve the

coded modulation error exponent. This is due to the MLC (with or without MSD) error exponent always being given by the minimum of the error exponents of the various levels, which results in an error exponent smaller than 1. While BICM suffers from a nonzero, yet small, capacity loss compared to CM and MLC/MSD, BICM attains a larger error exponent, whose loss with respect to CM is small. This loss may be large for an MLC construction. In general, the decoding complexity of BICM is larger than that of MLC/MSD, since the codes of MLC are shorter. One such example is a code where only a few bits out of the m are coded while the rest are left uncoded. In practice, however, if the decoding complexity grows linearly with the number of bits in a codeword, e.g., with LDPC or turbo codes, the overall complexity of BICM becomes comparable to that of MLC/MSD.

Summary: Since its introduction, BICM has been regarded as a pragmatic yet powerful scheme to achieve high data rates with general signal constellations. Nowadays, BICM is employed in a wide range of practical communication systems, such as DVB-S2, Wireless LANs, DSL, WiMax, and the 4G cellular systems.

5.12 Q-ary LDPC-coded Modulations



5.13 Combined Coding and Shaping

从图 5.1.7 可知，采用均匀的 2D 信号集，最好的系统性能最多只能达到 constrained-capacity，而此极限距离 Shannon 容量限还有 1dB 左右的差距。To achieve the Shannon-capacity-approaching performance (in the high SNR regime) for coded-modulation systems, it is necessary to use constellation shaping in conjunction with coding. Modern channel coding techniques are subjects of the previous chapters, and the related shaping methods are discussed below.

5.13.1 Overview of Constellation Shaping Methods

从前面关于成形增益的讨论中，我们知道信号星座成形基本上有两种技术：

- 第一种我们称为*几何成形*，使信号星座的几何形状类似于球体（在高维信号空间中）；（Signal points are selected equiprobably from an N-dim sphere）
- 第二种我们称为*概率成形*，使发送的信号具有与正态分布相近的分布。

(Directly generate an appropriate nonuniform low-dimensional distribution.)

It is noteworthy that 从一个没有冗余的数据序列产生一个非均匀分布是信源编码的对偶运算(dual operation)。在信源编码中，不等概的（有冗余）输入被转换成无冗余的输出；hence, the symbols are equiprobable. This duality allows the use of source decoders as encoders in shaping schemes. We will also see that shaping and channel coding are dual operations, too. Hence, it is possible to use channel decoders for signal shaping, as well.

In summary, source coding, channel coding and signal shaping are mutually dual. Fig. 5.13.0 shows the relations.

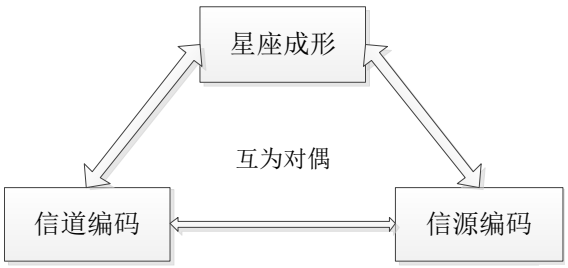


Fig. 5.13.0 Duality

另外，signal shaping is intended to decrease 平均发送功率，但是由于其低维分量信号星座扩展（constituent one-dimensional constellation），它将增加（一维）发送信号的峰值功率。星座扩展和 increased peak energy are the price to be paid for a gain in average energy.

在编码调制系统中具体实现时，这两种成形技术可以有多种实现方法，但大体上可以分为两大类：单层编码与两层编码方法。

在两层方案中[Forney84]，一层为信道编码，另一层为信源编码（成形编码）。图 5.13.1 所示是编码与成形结合的两层方案。首先把一个给定的 n-维信号星座 $\mathcal{A}$ 划分成若干个信号点子集 $\mathcal{A}_i$，$1 \le i \le m$。二进制数据流 $\underline{u}$ 按照一定的速率划分为两个子数据流 $\underline{u}^{(c)}$和 $\underline{u}^{(s)}$。子数据流 $\underline{u}^{(c)}$进入一个信道编码器，输出的序列 $\underline{c}=(c_1,\ldots,c_j,\ldots,c_L)$用来选择信号子集序列 $\underline{s}=(s_1,\ldots,s_t,\ldots,s_N)$，其中 $s_t \in \{\mathcal{A}_i, 1 \le i \le m\}$。这一层称为信道编码层。子数据流 $\underline{u}^{(s)}$进入一个成形编码器，输出的序列用来选择一个特定的信号点序列 $\underline{x}=(x_1,\ldots,x_t,\ldots,x_N)$，其中 x_t 是子集 s_t 里的信号点。这一层称为信源编码层。根据不同的信号选择方法，信号星座中每个点的使用概率或许不一样。

信道编码层的作用是在子集序列中引入冗余，从而使得接收端能够无误地（以尽可能小的错误概率）恢复子集序列。信源编码层的作用是对于给定的速率，选择信号点使得信号星座的平均能量尽可能小；等价地，对于给定的平均能量，选择尽可能多的信号点使得传输速率尽可能高。

两层方案的最佳性（从达到信道容量的角度）是基于编码和成形的可分性的。已经证明，对于高信噪比的系统，如果信道编码层和信源编码层分开设计并适当地构成一个两层系统，则信道容量的损失是可以忽略的[Forney2000]。

图 5.13.2 所示是编码和成形结合的单层方案。二进制数据序列 $\underline{u}$ 进入信道编码器，产生输出码字 $\underline{c}$。码字 $\underline{c}$ 进入成形映射器后输出信号序列 $\underline{x}$。在该方案中，信号映射器具有成形编码功能，它把均匀等概的二进制序列转化为统计特性与信道特性相匹配的信道符号序列，使得相应的互信息尽可能地大。在此基础上，再设计一个（二元或多元）信道码，以逼近这个互信息，从而逼近信道容量。单层结构的最佳性是由 Gallager 引理保证的[Gallager68]，其对所有的信噪比都成立。理论上，在接收端可以实现最小误比特率(BER)译码／解映射算法。在实际中，可以采用迭代的译码／解映射算法。这时，往往要在编码器和映射器之间引入一个随机交织器。

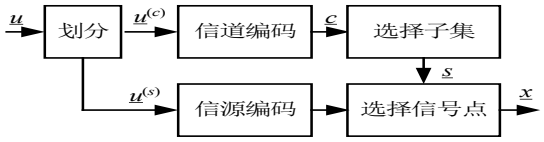


图 5.13.1 编码与成形结合的两层实现方案。

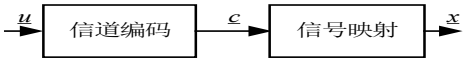


图 5.13.2 编码与成形结合的单层实现方案。

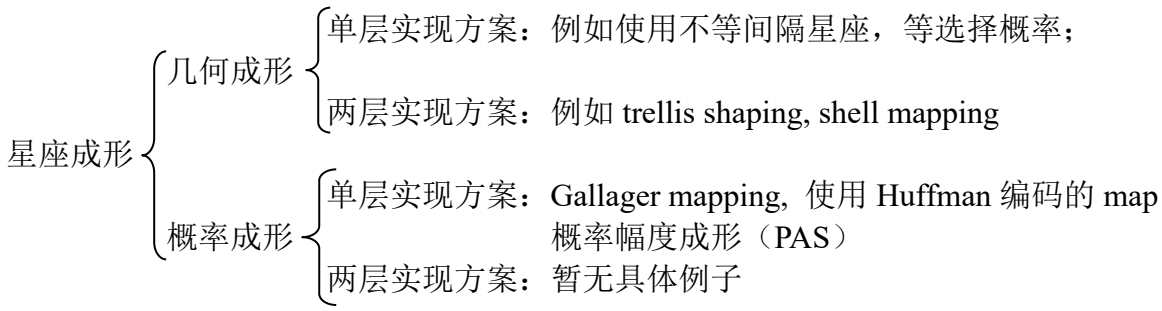
如果允许使用变速率的成形编码，则一个简单的方法是使用前缀码（如 Haffman 编码）。这种方法可以认为是**概率成形**方法，其出发点是计算最佳的信源分布，然后利用不等长的编码技术去逼近这个分布。可以证明，对于给定的平均能量，使得速率最大的分布是 Maxwell-Boltzmann 分布[4]。若以此为“信源分布”，则前缀码是最佳的。Livingston 把低维的信号星座划分成几个大小不等的子星座，然后利用不等长编码获得成形增益，可以认为是**几何成形**方法。变速率编码的优点是其考虑的信号点都来自低维信号星座，因而映射/解映射（无论用算法还是用查表）容易实现。但是，正如 Forney 指出的那样，变速率成形的最大问题是数据需要缓存和再同步，因而限制了其实用范围。

固定速率编码的目的是使得信号星座的形状接近于球体。但是，要逼近信道容量，信号星座的维数必须足够大。Forney（针对两层方案）提出了一种基于无限维空间的成形技术—“篱笆图成形”(Trellis Shaping)，可以很容易地获得 1dB 左右的成形增益。还有一种实用的技术是“壳映射”（Shell Mapping）[9]，这些可以看作是**几何成形**方法，在维数适中的情况下可以获得部分成形增益。

在单层方案中，使用不等星座点间隔的信号星座（如 Sun 和 van Tilborg 提出的不等间隔的 PAM 信号星座，一个例子见下图），可以看作是**几何成形**方法。Gallager 提出的多对一映射[Gallager68]、X. Ma 和 Li Ping 提出的线性叠加映射[Ma04]，均使得信号的信号点概率分布接近于正态分布，从而获得成形增益，它们属于**概率成形**方法。

图 5.13.3 使用不等间隔的 PAM 信号几何成形

小结：综上所述，我们可以将信号星座成形方法总结如下：



下面我们着重讨论两层方案中的几何成形与单层方案中的概率成形。二者的联系在于，高维球体（或类球体）在任一维上的投影近似服从正态分布。二者的区别在于，前者依靠优化高维信号星座的形状来获得成形增益，而后者依靠优化低维信号的概率分布来获得成形增益。前者在译码时一般不需要译码器和映射器之间进行迭代，而后者通常需要实现迭代的译码-解映射算法。

5.13.2 Gallager Mapping

令 X 表示信道的输入符号，Y 表示信道的输出符号。对于一个给定的信号星座 $\mathcal{A}_x$，假定 $P_X^*(x), x \in \mathcal{A}_x$ 是信道的最佳输入分布（能使互信息 $I(X;Y)$ 最大化）。Gallager 映射器的基本设计方法如下：

首先将信道编码器输出的编码比特序列 $\mathbf{c}$ 按一定长度进行分组 $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \ldots$，然后采用多对一的映射方法，将一个或多个比特序列 $\mathbf{v}_i$ 映射到 $\mathcal{A}_x$ 的每一个星座点上，使得由此导致的星座点选择概率（即 X 的实际分布）近可能接近于 $P_X^*(x)$。

图5.13.4是一个使用4-PAM星座时的Gallager映射例子。在接收端，我们需要采用BICM-ID这种软信息迭代联合解调/译码方式来进行信号检测。

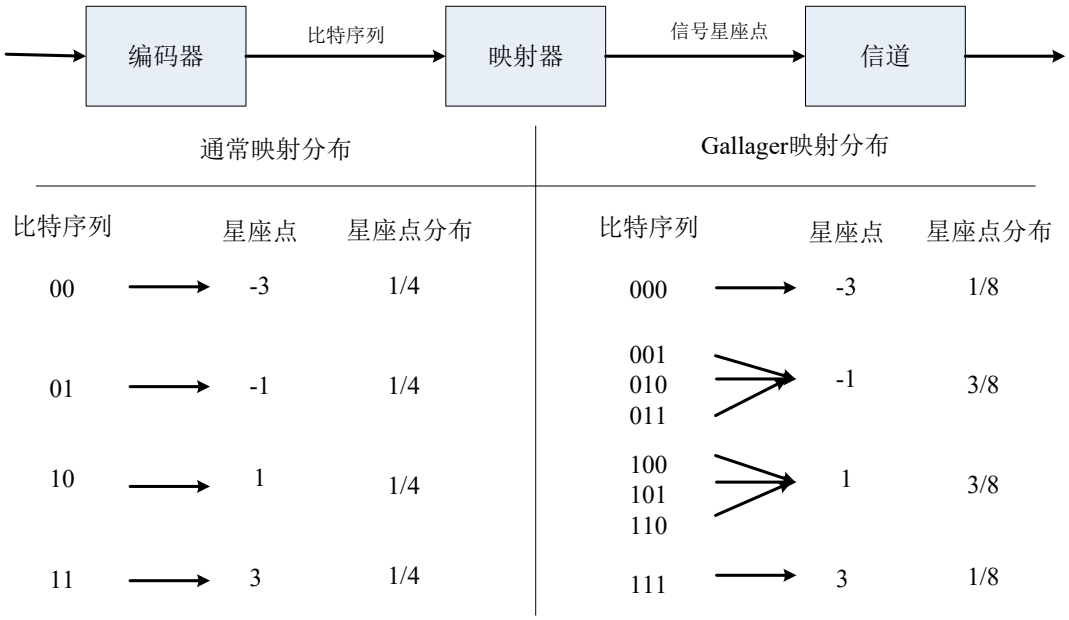


图 5.13.4

类似于BICM，对于有限码长，在Gallager映射器中采用不同的labeling策略，系统的BER性能也会不同。对于BICM系统中Gallager映射器的优化设计，我们提出采用下述的labeling设计准则：

1) 对于同一组内的不同标号(label)，设计的时候尽量最小化标号之间的最大汉明距离；
2) 对于不同组内的不同标号，设计的时候同样尽量最小化标号之间的最大汉明距离。

Example 5.13.1 将4比特长符号调制到8-PAM星座的Gallager映射

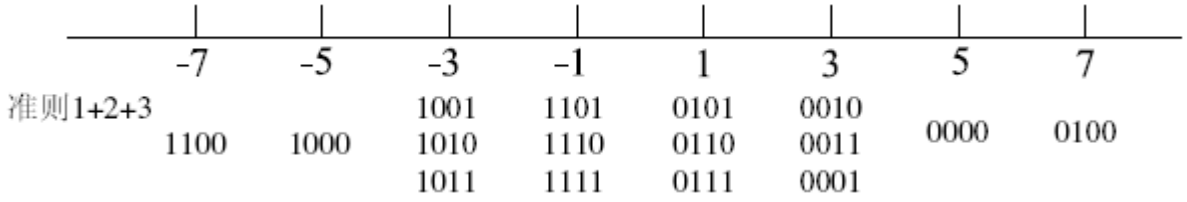


图 5.13.5

该映射不但考虑了星座点使用概率近似离散高斯分布，而且最小化了组内以及组外相邻标号之间的最大汉明距离。经过这样的优化设计，不但给系统带来了成形增益，而且在解调译码出错的时候，最大程度地降低了系统的误比特率。

对于PAM 信号的设计可以直接推广至常用的QAM 调制。下面给出一个将符号长度为5比特的符号映射到16-QAM星座图上的Gallager 映射器具体图样。

Example 5.13.2 将5比特长符号调制到16-QAM星座的Gallager映射

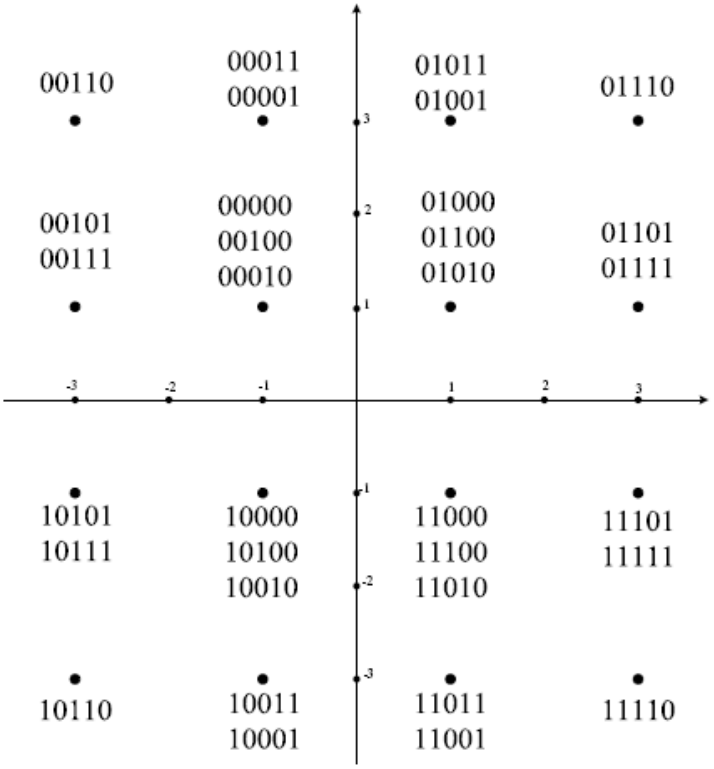
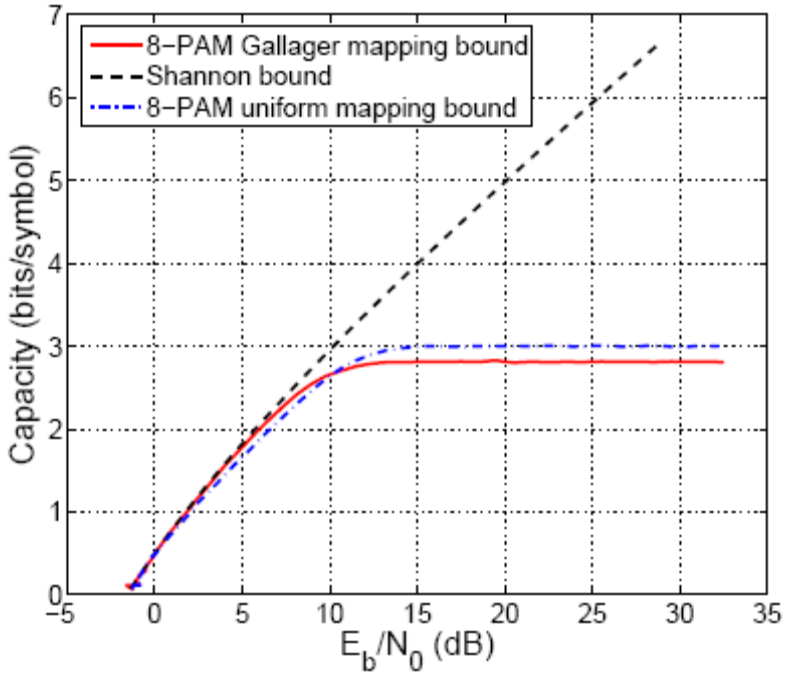
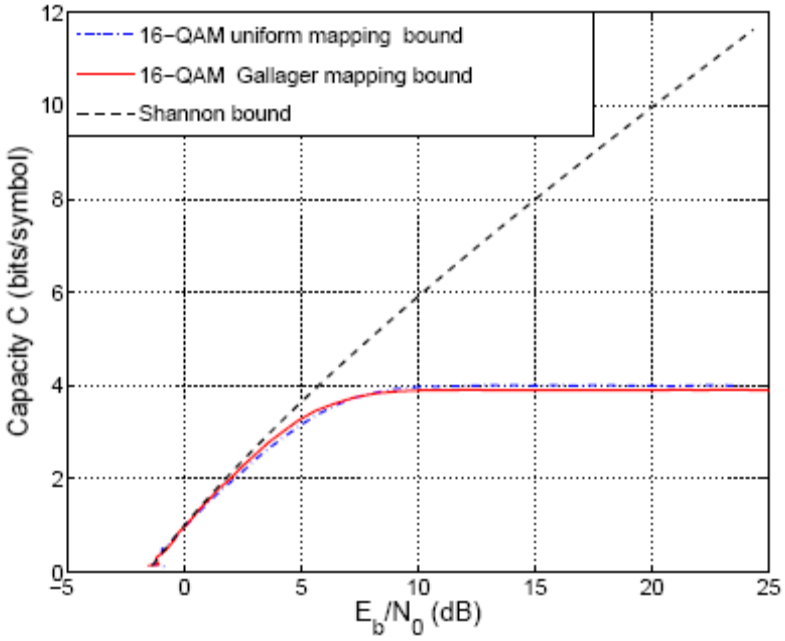


图 5.13.6

在上述 Gallager 映射器所给出的输入分布下，采用 8-PAM 与 16-QAM 星座的编码调制系统的信道容量分别如图 5.13.7(a)和(b)所示。



(a) 8-PAM



(b) 16-QAM

图 5.13.7 不等概输入分布下编码调制系统的容量

下面我们以基于LDPC码的BICM-ID系统为例，来讨论有限码长下采用Gallager映射器的具体性能。LDPC码选取了中国移动多媒体广播(China Mobile Multimedia Broadcasting, CMMB) 标准中的1/2码率规则二元LDPC码，该码校验矩阵的行重为6，列重为3，码长为9216比特。信号映射器采用图5.13.6所示的16-QAM Gallager映射器，假设信道模型为AWGN。LDPC码译码器采用和积算法(SPA)，最大译码迭代次数设定为10次；而解映射器与译码器之间的外部迭代次数设定为5次。图5.13.8显示了各种不同设计准则下谱效率为2.5 bits/s/Hz的Gallager映射编码调制系统仿真结果。可以观察到，当BER=10^{-5} 时，优化的映射设计方案比未优化的自然映射方案获得的增益超过1dB；与相同谱效率的常规均匀32-QAM调制对比，优化后的Gallager映射系统BER性能大约要好0.25 dB。

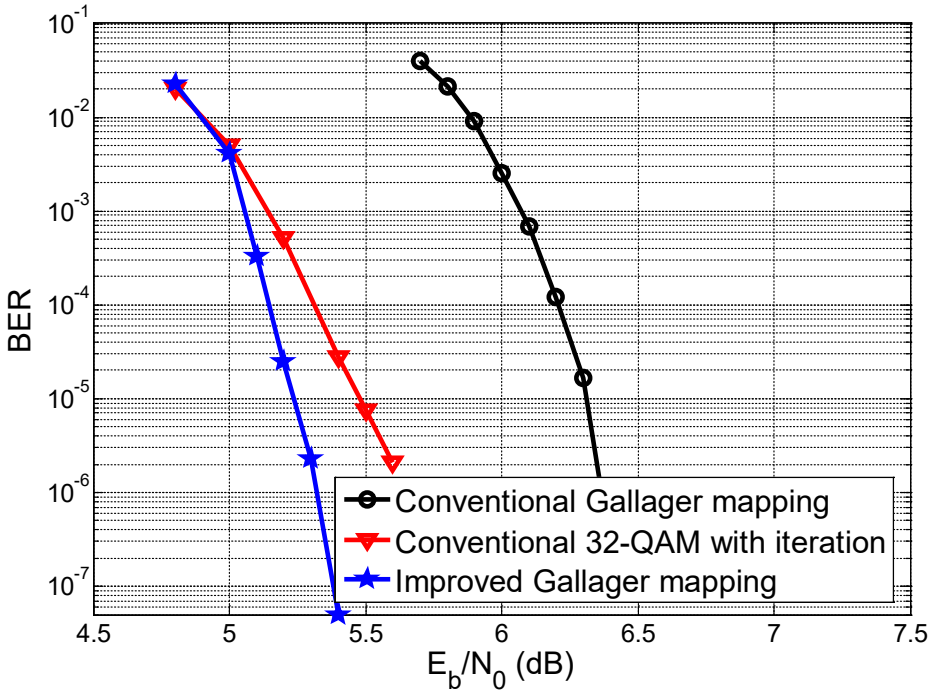


图 5.13.8

EXIT Chart分析：

下面我们以EXIT chart 作为分析工具，从理论的角度来分析前述的Gallager映射器设计准则的有效性。EXIT chart 性能分析原理框图如图5.13.9所示。

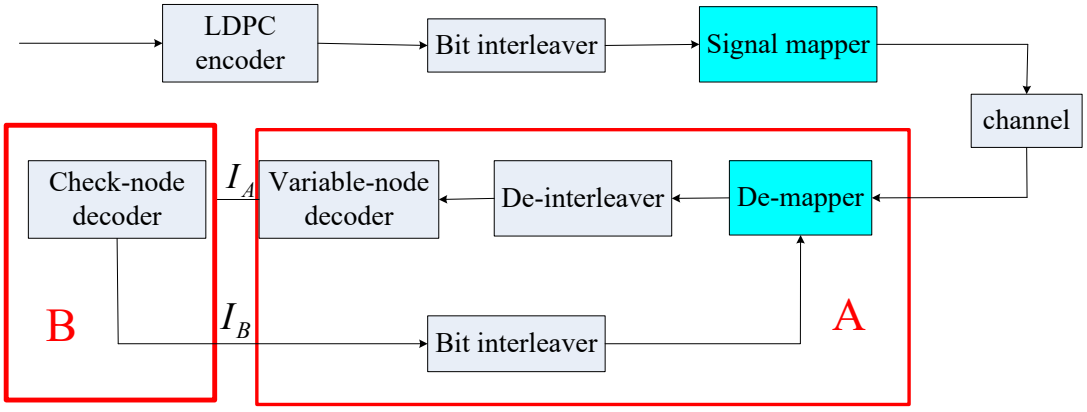


图5.13.9

在图5.13.9中，模块**A**包括DeM以及VND，而模块**B**只包括CND 一部分。模块**A**输出的互信息记为I_A，模块**B**输出的互信息记为I_B，DeM 输出的互信息记为I_D。模块A的因子图如图5.13.10所示。

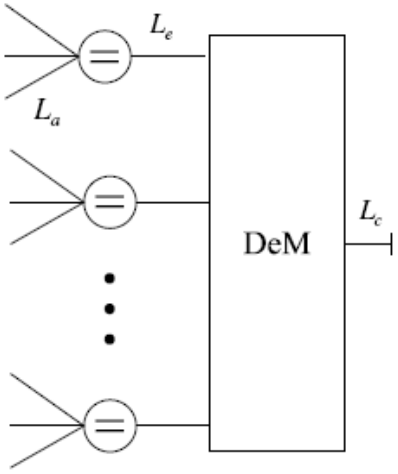


图5. 13. 10 模块**A**的Forney 型因子图

应用对LDPC码进行EXIT Chart分析的类似方法，我们可以画出满足不同映射准则的EXIT曲线，图5.13.11给出了8-PAM Gallager映射器的分析结果。其中LDPC码集合为规则LDPC码集合$\mathcal{C}(3, 6)$。

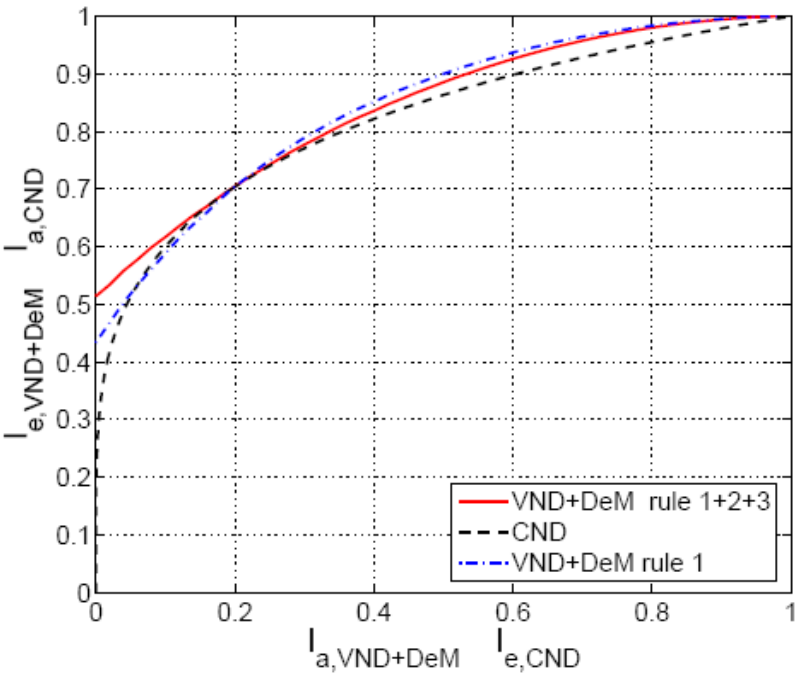


图5.13.11 8-PAM Gallager映射EXIT chart 分析@ $E_b/N_0 = 7.9$ dB

5.13.3 叠加 Mapping

将经单层或多层信道编码后的数据流划分为多层（如 K 层）比特流，记第 k 层中的二进制比特流为 $\underline{v}^k = (v_1^k, \cdots, v_N^k)$。在 t 时刻，所有 K 层的编码符号 v_t^k 组成一个矢量 $\underline{v}_t = (v_t^0, v_t^1, \cdots v_t^{K-1})$，经过信号映射器映射到信号集 A 中的一个信号点 $x_t \in A$。线形叠加

映射器的结构如图 5.13.12 所示。

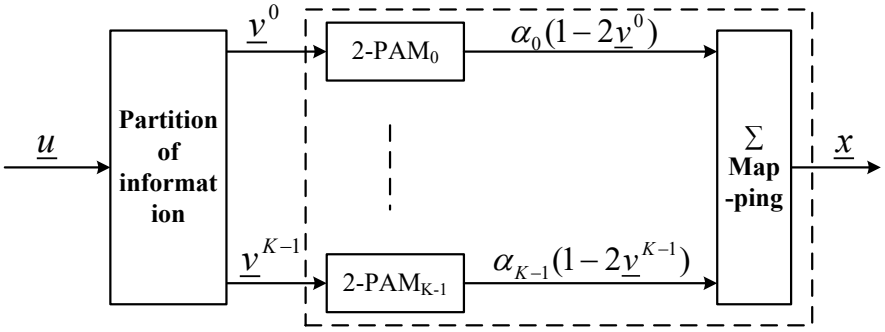


图 5.13.12 叠加映射器结构

在具体映射过程中，各层编码符号首先被 2-PAM 映射器映射成数据流 $\alpha_k \cdot (1-2\underline{v}^k)$，$k \in \{0,1,\cdots,K-1\}$，其中 α_k^2 是第 k 层信息所对应的能量。然后，Σ 映射器将所有 K 层信号进行求和，形成一种“和”信号 $\underline{x}$：

$$\underline{x} = \sum_{k=0}^{K-1} \alpha_k \cdot (1-2\underline{v}^k)$$

这里，$\{\alpha_k\}$ 满足 $\sum_{k=0}^{K-1} \alpha_k^2 = E_s$。如果所有 α_k 相同，则映射器称为 type-I sigma-mapper;否则，称为 type-II sigma-mapper。由 type-I sigma-mapper 产生的信号星座是等间隔、不等概率的，如图 5.13.13 所示。由 type-II sigma-mapper 产生的信号星座一般是不等间隔的。

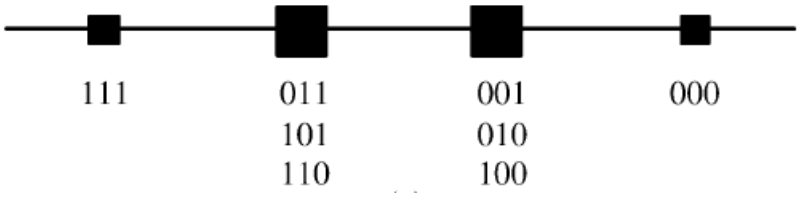


图 5.13.13 一个 sigma mapping with K=3 的例子。In this case, *P*x={1/8, 3/8, 3/8, 1/8}. 对应信号集为 $\mathcal{A}$ = {-3,-1,1,3} with $\alpha_0 = \alpha_1 = \alpha_2 = 1$。

在 type-I sigma-mapper 中，各层对应的信号能量相同，采用互信息设计准则，各层具有不同的码率，这样就类似于前述的 Multilevel coding 方案。在 type-II sigma-mapper 中,所有层可采用码率相同的信道码，但是每一层对应不同的能量；通过适当调节层间的能量关系，可达到在接收端各层能正确译码。功率分配方案有多种形式，可参考文献[?]。当各层的信道码相同（包括码率）时，一种最简单的功率分配方案如下：设 t 时刻发送信号总能量为 E_s，则各层信号能量 α_k^2 之间的约束关系为

$$\begin{cases} \sum_{k=0}^{K-1} \alpha_k^2 = E_s \\ \dfrac{\alpha_k^2}{1+\sum_{j>k} \alpha_j^2} = \alpha_{K-1}^2, \quad k = K-2, K-3, \cdots, 0 \end{cases} \tag{11}$$

这里我们对信道噪声方差进行了归一化处理，即 $\sigma_n^2 = 1$。

在传统的多层编码调制系统中，互信息的链式法则表现为系统中各层之间的速率分配，而这里表现为各层之间的功率分配。

In [Ma-Li, TIT2004], 一种基于叠加映射的多层编码调制方案 is proposed.

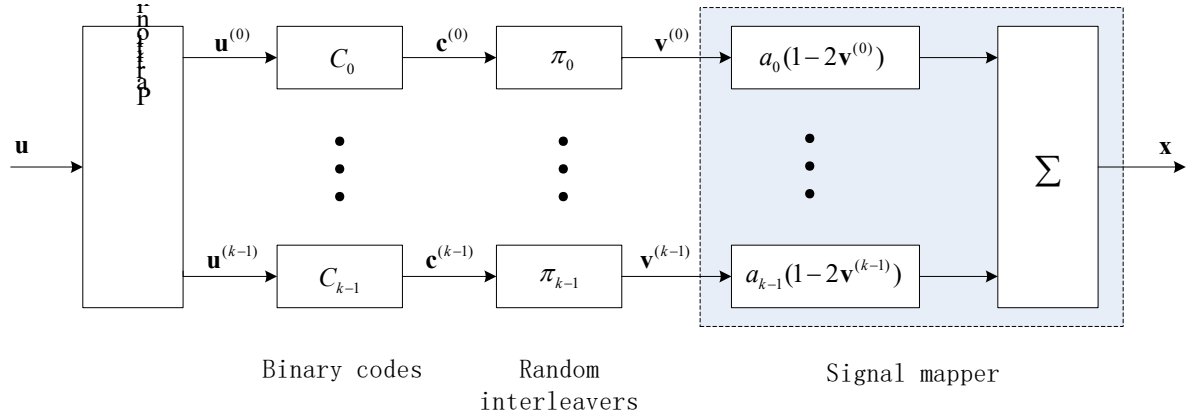
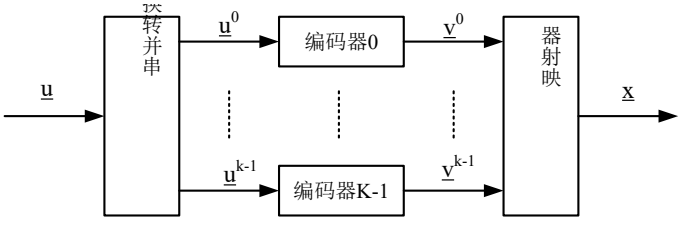


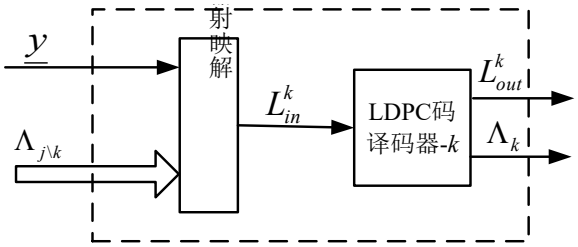
图 5.13.14

类似于对多址信道接收信号的处理，对于接收到的叠加编码信号，接收端可采用 successive interference cancellation (SIC)或高斯近似方法进行解映射与译码。

下面以第 k 层为例，说明高斯近似解映射算法。假定各层采用 LDPC 编码，其译码结构框图如图 5.13.15 所示。信道接收到的信息 **y** 和其它层的软输出外信息 $\Lambda_{j\backslash k}$ 一起输入解映射器，通过高斯近似解映射算法，得到分量码译码器的软输入信息 L_{in}^k 。



(a)



(b)

图 5.13.15 第 k 层译码器框图

高斯近似解映射算法如下：

设信道噪声、接收端的接收信号以及第 k 层的编码输出信号分别表示为随机变量 W、Y 和 V^k，则有：

$$Y=\sum_{k=0}^{K-1}\alpha_k(1-2V^k)+W \tag{12}$$

现在考虑对信号 V^k 进行解映射，则(12)式可以改写为

$$Y=\alpha_k(1-2V^k)+\widetilde{W} \tag{13}$$

其中 $\widetilde{W}=W+\sum_{i\neq k}\alpha_i(1-2V^i)$。不失一般性，我们总假设信道噪声的方差 $\sigma^2=1$；而且将其它各层信息以及信道噪声一起近似看成是一种高斯噪声，即 $\widetilde{W}$ 近似看作满足高斯分布 $\widetilde{W}\sim N(\mu_{\widetilde{W}},\sigma_{\widetilde{W}}^2)$，其中

$$\mu_{\widetilde{W}}=\sum_{i\neq k}\alpha_i(1-2\mu_{V^i})\ ,\quad \sigma_{\widetilde{W}}^2=1+\sum_{i\neq k}4\alpha_i^2\sigma_{V^i}^2$$

$$\mu_{V^i}=P_{V^i}^{(a)}(1)\ ,\quad \sigma_{V^i}^2=P_{V^i}^{(a)}(0)P_{V^i}^{(a)}(1)$$

则有

$$P_{V^k}^{(e)}(m)\propto\exp\left(-\frac{(y-\alpha_k(1-2m)-\mu_{\widetilde{W}})^2}{2\sigma_{\widetilde{W}}^2}\right) \tag{14}$$

其中 m 取值为 0 或 1。根据(14)式可知，在对数域中，第 k 层译码的软输入信息可表示为：

$$L_{in}^{k}=\frac{2\alpha_k(y-\mu_{\widetilde{W}})}{\sigma_{\widetilde{W}}^2} \tag{15}$$

5.13.4 Trellis Shaping

Trellis shaping, proposed by Forney [Forney92], is an efficient scheme for signal shaping, in which mapping of information to the transmit sequence is done via a search through the trellis of a shaping code. (Sign bit shaping 是 trellis shaping on regions 的特例：星座被分为

4 个 subregions.)

Shaping on Regions.

我们将星座点进行分组，form groups of signal points or regions. The groups are selected by an algorithm, whereas the actual point from the group is addressed memoryless. Assuming a suitable labeling of signal points, i.e., by Ungerboeck's set partitioning, channel coding operates on the least significant bits and shaping on the MSBs. Fig. 5.13.16 sketches the general concept.

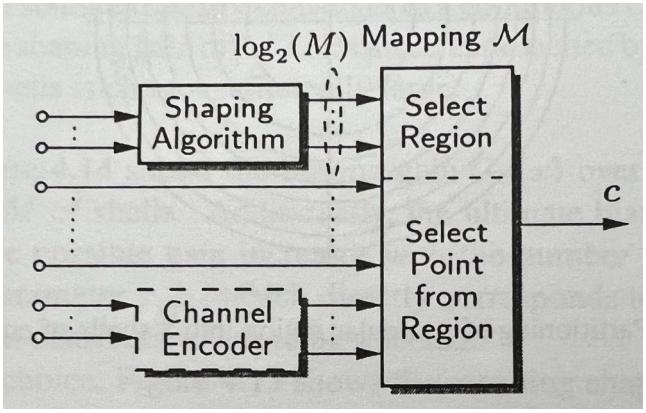
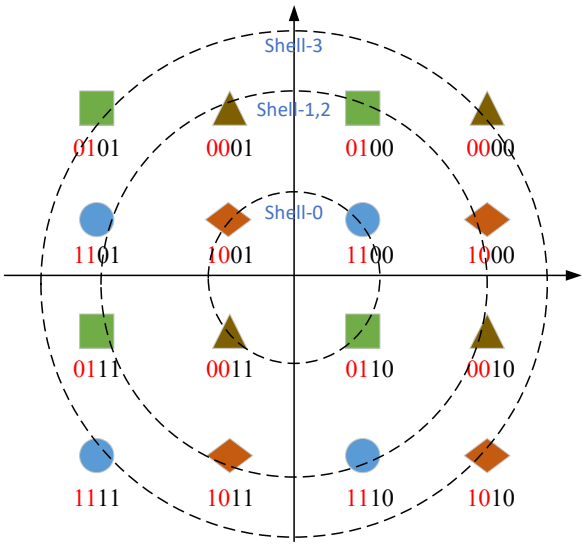


Figure 5.13.16 General concept of shaping on regions (combined with channel coding)



An obvious approach is to arrange points with equal, or at least similar, energy into the same region. Since energy is proportional to the squared Euclidean norm, the regions are then nested shells or rings, as is the case for onions. In order to operate shaping and coding separately, the partitioning into regions has to be done such that each region contains an equal number of signal points from each coset of the coding lattice. Then, regardless of actually selected region, coded-modulation works as usual; in particular the achievable coding gain is not affected. 因为对于中等星座大小，channel coding and signal shaping interact, 它们的增益不能简单相加，因此 if a moderate, fixed rate should be transmitted as reliably as possible, the effective gain is much lower than predicted by the ultimate shaping gain.

Shaping Code.

假定信号星座 is partitioned into M regions (shells), and taking a frame of N consecutive symbols into consideration. We view the set of permitted shell N-tuples as some kind of shaping code. The codewords are of length N and the code symbols stem from an M-ary alphabet; the code size is 2^K. The shaping 算法 maps the information to a codeword and the shaping decoder again extracts this information carried in the shell indices.

As a very simple example for a shaping code, we consider M=2, N=3, and K=2 information bits to be represented in the shell selection. The shaping code 是一个码长为 3 的二元码，由下面 4 个码字组成：

（000），（001），（010），（100）

其中“0”代表选 inner shell，“1”代表选 outer shell。上述 Shaping code 表示 only shell combinations with at most one “outer shell” are used.

为了说明成形与信道编码的对偶关系，我们将二元（3,1）重复码的标准阵列于表 X 中。

(000)	*(111)*
(001)	(110)
(010)	(101)
(100)	(011)

可以看出，陪集首就是上述成形码的码字。This principle can be generalized: **Starting from a channel code, we can derive a shaping code as the set of 最小重量陪集首**。That is, a shaping code can be defined as the set of coset members which lie in the Voronoi region of the all-zero codeword. Information is then carried in the number of the coset（陪集编号）.

Shaping by Channel Decoding.

We now consider the problem of addressing the minimum weight coset member. 令 **G** be the $(N-K)\times N$ generator matrix of the channel code, and **H** be the 相应的校验矩阵 of size $K\times N$, such that $\mathbf{G}\mathbf{H}^T=\mathbf{0}_{(N-K)\times N}$. 与一个长为 N 的向量 **z** 相对应的 K-digit 伴随式 syndrome **s** 定义为

$$\mathbf{s}=\mathbf{z}\mathbf{H}^T$$

Sometimes, $\mathbf{H}^T$ is called *syndrome former*.

A particular property of a coset is that all its members result in the same syndrome, and that the syndromes for 不同陪集 differ. Hence, it is the syndrome which carries the information in the context of signal shaping. We use a left inverse $(\mathbf{H}^{-1})^T$ for $\mathbf{H}^T$, i.e., $(\mathbf{H}^{-1})^T\mathbf{H}^T=\mathbf{I}_{K\times K}$, as a coset representative generator. Then, given the syndrome **s**, a coset representative **z** can be obtained as

$$\mathbf{z}=\mathbf{s}(\mathbf{H}^{-1})^T$$

Since $\mathbf{z}\mathbf{H}^T = \mathbf{s}(\mathbf{H}^{-1})^T\mathbf{H}^T = \mathbf{s}$. Hence, there is an easy way to address a particular, but not necessarily minimum weight, coset member.

With the coset representative **z**, any codeword **c** can be added without changing the coset or the syndrome, and thus preserve information, since

$$(\mathbf{z} \oplus \mathbf{c})\mathbf{H}^T = \mathbf{z}\mathbf{H}^T \oplus \mathbf{c}\mathbf{H}^T = \mathbf{s} \oplus \mathbf{0} = \mathbf{s}$$

因此，in signal shaping, the task of a "channel decoder" matched to **G** is to find a valid codeword which, added to coset representative, results in the desired minimum weight coset number. But this is exactly the codeword closest to **z**; i.e., the codeword that will be found by a minimum-distance decoder.

这样，mapping of information to a block of shell indices is thus transformed into the problem of decoding a (linear) channel code.

小结：Note that 所要求的最小重量陪集成员对应于（信道噪声引起的）错误图样。对 **z** 进行信道译码得到 **c**，然后 $\mathbf{z} \oplus \mathbf{c} = \mathbf{e}$ 得到陪集首。

如果将信道码用 trellis 来表示，并用 Viterbi 算法进行译码，则这里的分支度量就是 Hamming 重量；The best (最小 Hamming 重量)path is preferably found by applying VA.

图 5.13.17 sketches the basic concept of shaping by channel decoding. *K* information bits are contained in the syndromes. A channel decoder searches for a suitable codeword **c**, or equivalently for a vector **r**, such that **c** = **rG**. 这里，**r** 表示 *N*-*K* 比特的成形冗余。

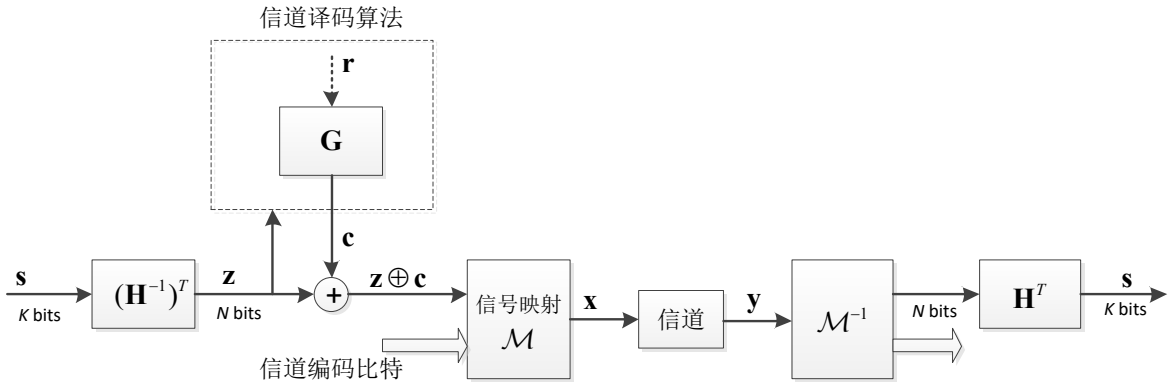


图 5.13.17 shaping by channel decoding 原理框图

- With the use of *M*-ary channel code (instead of 二元码)，该方法可直接推广到 *M* 个 shell 数。

Example: *Shaping using Hamming codes*

As an example, we study the binary (7, 4) Hamming code. Generator matrix **G** and the respective parity-check matrix may be given as

$$\mathbf{G} = \begin{bmatrix} 1101000 \\ 0110100 \\ 0011010 \\ 0001101 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1011100 \\ 0101110 \\ 0010111 \end{bmatrix}$$

The (7, 4) Hamming code has $2^{7-4} = 8$ cosets – the coset leaders are all binary 7-tuples with Hamming weight at most 1. Hence, using this code, a shaping code for frame size N=7, carrying K=3 information bits, can be defined. The corresponding trellis representation is shown in Fig. 5.13.18.

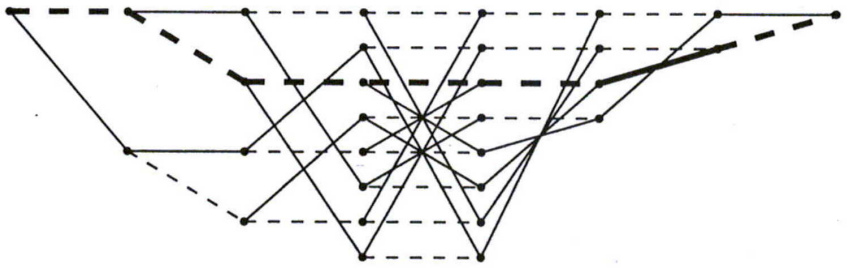


Fig. 5.13.18 Trellis diagram of the coset with s=(011) of the (7, 4) Hamming code. Solid branches represent code symbol '1', and dashed branches represent code symbol '0'. The minimum weight codeword (0000010) is marked.

It is easy to very that a possible choice for the left inverse of $\mathbf{H}^T$ is

$$(\mathbf{H}^{-1})^T = \begin{bmatrix} 1000000 \\ 0100000 \\ 0000001 \end{bmatrix}$$

If, for example, information **s** = (011) should be encoded in the shell selection, first the coset representative is generated, and reads

$$\mathbf{z} = \mathbf{s}(\mathbf{H}^{-1})^T = [011] \begin{bmatrix} 1000000 \\ 0100000 \\ 0000001 \end{bmatrix} = [0100001]$$

This vector has to be added to all possible codewords, and then the minimum weight word of this coset has to be determined. Using the code trellis, the trellis representation of the coset is simply obtained by adding the symbols of **z** to the corresponding code symbols, i.e., branches, of the trellis. Searching the above trellis for the minimum weight codeword, i.e., using the Hamming weight as cost (branch metric) by applying the Viterbi algorithm, results in the decoded word (0000010), which is the desired 7-tuple of shell indices, and can be expressed as

$$[0000010] = \underbrace{[0100001]}_{\mathbf{z}} \oplus \underbrace{[0100011]}_{\mathbf{c}}$$

At the receiver, the sequence of shells (0 or 1) is observed, and from this 7-tuple data are recovered to be

$$\mathbf{s} = [0000010]\mathbf{H}^T = [011]$$

Performing this procedure for all information triples **s**, we may achieve the mapping of information to shell 7-tuples.

Trellis Shaping on Regions

前面讨论的是用分组码作为 shaping code 的例子，成形是在有限维信号空间中进行的。1992 年，Forney 提出了采用卷积码作为成形码的方法，称为 trellis shaping, in which 前述的生成矩阵和 syndrome former are replaced by 转移函数矩阵。As the scheme is sequence-oriented, the resulting signal constellation is characterized in a 无限维信号空间。

For trellis shaping, we need the definition of cosets of convolutional codes and their associated syndromes. 为区分实（复）数序列，下面的讨论中我们用 D 变换的记号来表示二元符号序列：$\mathbf{s}(D) = \sum_k s_k D^k, s_k \in \mathrm{GF}(2)$。令 $\mathbf{G}(D)$表示一个卷积码$\mathcal{C}$的生成矩阵。

- A *syndrome former* associated with a rate-k/n 卷积码 is a linear system specified by an $n \times m$, with $m = n - k$, transfer matrix $\mathbf{H}^T(D)$ with maximum rank, such that

$$\mathbf{G}(D)\mathbf{H}^T(D) = \mathbf{0}_{k\times m}$$

The syndrome sequence $\mathbf{s}(D)$ corresponding to $\mathbf{z}(D)$ is then

$$\mathbf{s}(D) = \mathbf{z}(D)\mathbf{H}^T(D)$$

In [Forney70, 73a], it is proved that for each linear time-invariant binary convolutional code, there exists a feedback-free syndrome former which can be realized with the same number of states as the convolutional encoder.

- A *coset representative generator* can be used to specify a *coset representative sequence* $\mathbf{z}(D)$. Its $m \times n$ transfer matrix $\left(\mathbf{H}^{-1}(D)\right)^T$

$$\left(\mathbf{H}^{-1}(D)\right)^T \mathbf{H}^T(D) = \mathbf{0}_{m\times m}$$

The Shaping Algorithm

We assume that the constituent is partitioned into $M=2^n$ regions (shells), and $m < n$ bits should be transmitted in the shell selection. Clearly, a binary rate-k/n 卷积码$\mathcal{C}_S$ with generator matrix $\mathbf{G}_S(D)$ meets our requirements. In each time instant, $m = n - k$ bits can be carried by specifying the coset.

The resulting transmission scheme with trellis shaping is given in Fig. 5.13.19. Binary information to be transmitted is split into two parts: the less significant bits are processed as usual. 例如 in TCM, the data are encoded by a channel encoder. Shaping and channel coding combine seamlessly if care is taken that each region contains an equal number of cosets (subsets) of the coding lattice. The most significant bits, combined into the sequence $\mathbf{s}(D)$ of binary m-tuples, is transformed into the sequence of coset representative sequence $\mathbf{z}(D)$ by

filtering with the inverse of the syndrome former

$$\mathbf{z}(D) = \mathbf{s}(D)\left(\mathbf{H}_S^{-1}(D)\right)^T$$

Given this sequence, a trellis decoder for $\mathcal{C}_S$ – preferably a VA – searches through the code trellis and determines a valid code sequence **c**(D). The sum

$$\mathbf{w}(D) = \mathbf{z}(D) \oplus \mathbf{c}(D)$$

finally constitutes the sequence of shell indices or regions (binary *n*-tuples). Assume that shell 是嵌套环形的 and the shell index is expressed as an integer number. The Viterbi decoder determines a code sequence such that the sum over the shell indices is minimum. The branch metrics are the elements of **w**(D) expressed as integer numbers.

一种更直接、通用的方法是：Shaping is done using a decoder by considering the actual signal-point energy. The branch metric λ_k in the trellis decoder is then simply chosen equal to the Euclidean norm of corresponding constellation point a_k, i.e.,

$$\lambda_k = |a_k|^2$$

With this metric, there is no longer any need to arrange the regions as concentric rings. Any (regular) partitioning of the 信号星座 into shaping regions is possible.

At the receiver, channel decoding can be done as usual. The binary symbols are recovered via the inverse mapping $\mathcal{M}^{-1}$. The estimates of the bits addressing the region are contained in the sequence $\hat{\mathbf{w}}(D)$. From it, information is retrieved by the syndrome former. Assuming error-free transmission, i.e., $\hat{\mathbf{w}}(D) = \mathbf{w}(D)$, we have

$$\begin{aligned}\hat{\mathbf{w}}(D)\mathbf{H}_S^T(D) &= \left(\mathbf{s}(D)\left(\mathbf{H}_S^{-1}(D)\right)^T \oplus \mathbf{r}(D)\mathbf{G}_S(D)\right)\mathbf{H}_S^T(D) \\ &= \mathbf{s}(D)\left(\mathbf{H}_S^{-1}(D)\right)^T \mathbf{H}_S^T(D) \oplus \mathbf{r}(D)\underbrace{\mathbf{G}_S(D)\mathbf{H}_S^T(D)}_{=0} = \mathbf{s}(D)\end{aligned}$$

which recovers the initial syndrome sequence.

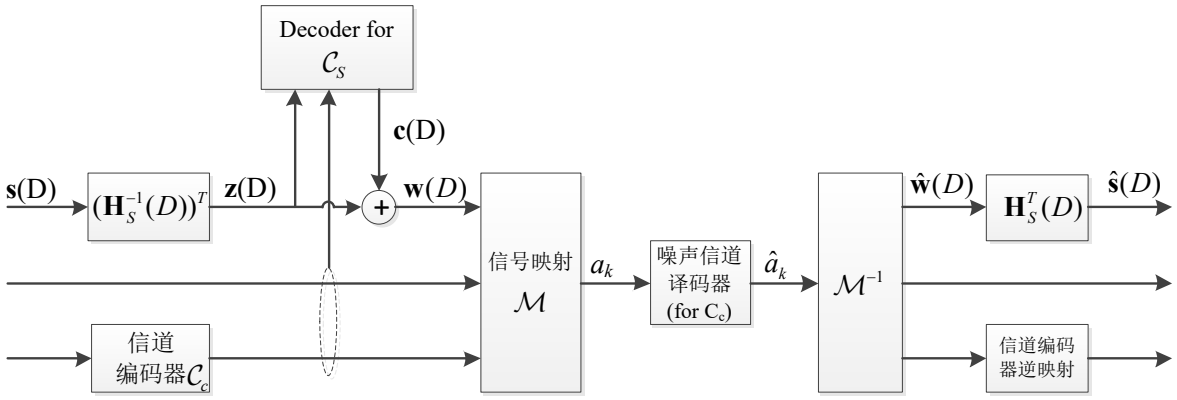


Fig. 5.13.19 收发机结构 for Trellis shaping

为避免成形译码错误的恶性传播，it is essential that (接收端) syndrome former 的实现 is feedback free. Fortunately, a feedback-free realization is always possible [Forney70, 73a]. Nevertheless, some error multiplication takes place when recovering data. Since error multiplication grows with the number of active taps in the syndrome former $\mathbf{H}_S^T(D)$, realizations with the least number of taps are preferable.

A Simple Example for Trellis Shaping: *Sign-Bit Shaping*

Following [Forney92], we now give a detailed example for trellis shaping. Since "large" signal sets are preferable for shaping, we apply a 16*16 two-dim (256-QAM) signal constellation. Since we are only interested in the performance of signal shaping, uncoded transmission is considered. Using one redundant bit for shaping, the transmission rate is 7 bits per 2D symbol.

The two most significant bits involved in shaping should simply give the sign of the in-phase and quadrature component, respectively. In other words, the constellation is divided into 4 shaping regions which are chosen equal to the 1/4 象限. Since only the sign of the signal is affected by shaping, this special case is called *sign-bit shaping*.

我们采用 Ungerboeck 的 4 状态，1/2 码率卷积码作为成形码，其生成矩阵 is given by

$$\mathbf{G}_S(D) = \begin{bmatrix} D & 1+D^2 \end{bmatrix}$$

Straightforward examination shows that a feedback-free syndrome former can be specified by the transfer matrix

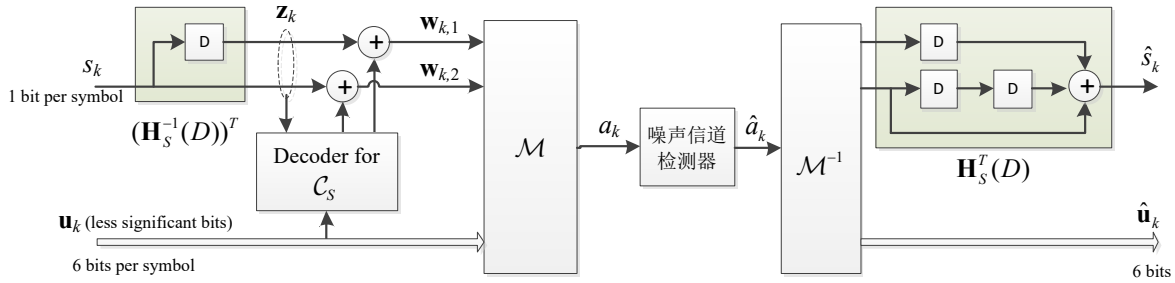
$$\mathbf{H}_S^T(D) = \begin{bmatrix} 1+D^2 \\ D \end{bmatrix}$$

And an adequate inverse of the syndrome former is

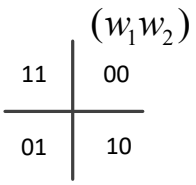
$$\left(\mathbf{H}_S^{-1}(D)\right)^T = \begin{bmatrix} 1 & D \end{bmatrix}$$

The block diagram of the sign-bit shaping scheme is shown in Fig. 5.13.20 (a). A standard Ungerboeck set partitioning is used for labeling the signal points. The least significant bits select one of 64 points in each 象限 and the most significant bits addressing the 象限 as

shown in Fig. 5.13.20 (b). The shaping gain G_s over the decoding delay (the path registers length of the VA) for the 4-state sign-bit shaping scheme is depicted in Fig. 5.13.21. We can see that for delays larger than approximately 20, the shaping gain saturates and approaches 0.94dB with a simple 4-state VA. For reference, the shaping gain of a hypersphere with dimension equal to twice the decoding delay is also shown.



(a)



(b) 象限的 Labeling

Fig. 5.13.20 Block diagram of sign-bit shaping using a 4-state rate-1/2 convolutional shaping code

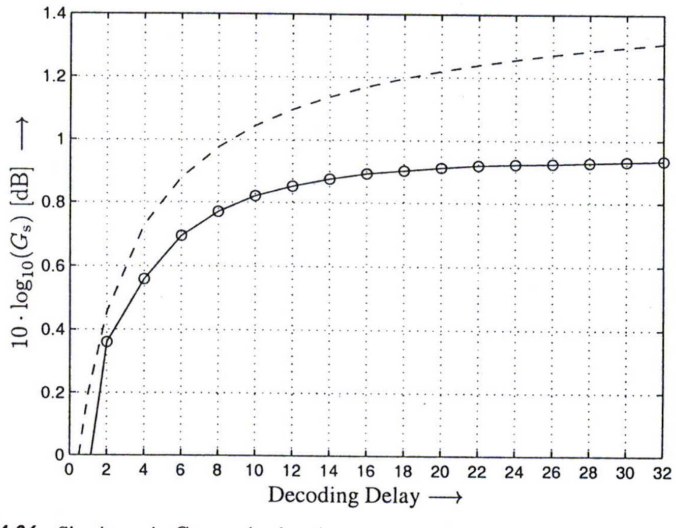


Fig. 5.13.21 Shaping gain for the 4-state sign-bit shaping scheme (solid).

Some Practical Considerations

- Shaping regions derived from lattice partitions: In principle, trellis shaping works with any partitioning of signal set into regions. A systematic approach is to base the selection of the regions on lattices and lattice partitions.

  If channel coding is applied, the coding lattice Λ_c has to be a sublattice of the signal

lattice Λ_a. Since signal shaping should not interfere with channel coding; i.e., the choice of the shaping regions may not change the coding coset, the shaping lattice Λ_s has to be a sublattice of the coding lattice. In summary,

$$\Lambda_a / \Lambda_c / \Lambda_s / \Lambda_b$$

should constitue a lattice partition chain, where Λ_b denotes boundary lattice.

- Choice of the Shaping code: In practice, 0.5 bit redundancy per dimension is sufficient in order to achieve almost the entire shaping gain. Hence, using two-dim signaling, one redundancy shaping bit suffices. In turn, shaping is preferably based on rate-1/n convolutional codes. Coset representative generator and syndrome former are then (n-1)-input n-output and n-input (n-1) output systems, respectively.
  Ungerboeck codes, originally designed for TCM, also work well for signal shaping. 根据对偶性，In [Forney92] it is conjectured that dual Ungerboeck codes and a dual lattice partition are appropriate.
  另外，4—8 个 regions seem to be a reasonable choice. 对于成形区域的选择，as shown above, a possible approach is to select the regions based on pure lattice theoretic considerations. 另外，用 concentric ring as shaping regions 也是可以的(spherical regions).
- 研究表明，trellis shaping based on rate-1/2、4-8 个状态的卷积码 ＋4 个 shaping regions 就能 achieve a significant portion of 最大成形增益 1.53dB.

Table 4.16 Comparison of trellis-coded modulation (TCM) [Ung82], trellis-coded quantization (TCQ) [MF90], and trellis shaping (TS) [For92].

	Trellis-Coded Modulation	Trellis-Coded Quantization	Trellis Shaping
Field	Channel Coding	Source Coding	Signal Shaping
Design Criterion	*Maximum Euclidean distance* between sequences of signal points	*Minimum average distortion* between original and quantized sequence	*Minimum average energy* of sequence of signal points
Lattice-Theoretic View	*Channel coding problem:* lattice with minimum probability of error *Packing problem:* consider only minimum Euclidean distance	*Quantization problem:* lattice whose Voronoi region has min. normalized second moment *Covering problem:* consider only peak distortion	*Quantization problem:* lattice whose Voronoi region has min. normalized second moment
Information Represented in	Sequence of *subsets* and point from subset	Sequence of *subsets* and point from subset	Sequence of *regions* and point from region
Encoder	Linear shift register (convolutional encoder)	Viterbi algorithm Metric: distortion	Viterbi algorithm Metric: signal-point energy
Decoder	Viterbi algorithm Metric: Euclidean distance	Linear shift register (convolutional encoder)	Linear shift register (syndrome former)
Main Complexity	Channel decoder	Source encoder	Shaping encoder
Gains	Typically 3–6 dB over uncoded transmission	Maximum 1.53 dB over scalar quantization of uniform distribution	Maximum 1.53 dB over uniform signaling

5.13.6 概率幅度成形（*Probabilistic Amplitude Shaping*）

概率幅度成形(PAS)是 Böcherer 等人于 2015 年提出的一种概率成形方案 [Böcherer-Tcom2015]。它针对一维 ASK 调制信号，采用一个分布匹配器级联一个高码率（Rc=(m-1)/m）系统型 FEC 码构成编码调制方案，分布匹配器用于产生概率成形所要求的信号幅度序列。考虑一个包含 $M=2^m$ 个信号点的 ASK 星座，其信号集合为 $\mathcal{X}=\{\pm1,\pm3,...,\pm(M-1)\}$。假定发送信号在 AWGN 信道上传输，其输入输出关系为：

$$Y=\Delta X+Z$$

其中 $Z\sim\mathcal{N}(0,1)$ 是均值为 0，方差为 1 的高斯噪声样本；Δ 为星座缩放因子。发送信号功率约束为：

$$\mathbb{E}[|\Delta X|^2]\le P$$

传统的均匀输入分布下，ASK 信号的可达信息速率我们已在第二章讨论过。这里主要讨论非均匀输入分布的情况，其信道容量可表示为：

$$C=\max_{\Delta,P_X:\mathbb{E}[(\Delta X)^2]\le P} I(X;\Delta X+Z)$$

对于离散输入 X，其最优分布 P_X 难以用解析法获得（可通过 Blahut-Arimoto 算法进行数值近似计算）。因此，下面考虑使用一个（次优的）熵最大化输入分布。令信号点 $x\in\mathcal{X}$ 出现的概率为

$$P_X(x)=A_v e^{-vx^2},\quad A_v=\frac{1}{\sum_{x'\in\mathcal{X}} e^{-vx'^2}} \tag{5.13-3}$$

其中 v 是一个非负参数，标量 A_v 是归一化系数。对于每个星座缩放因子 Δ，通过调整 v，使得 $\mathbb{E}[|\Delta X|^2]=P$。式（5.13-3）的概率分布称为Maxwell-Boltzmann（MB）分布[Cover2006, Ch.12]，它能够在平均功率约束条件下，最大化X的熵$H(X)$。通过优化 v 和 Δ，找到使得信道互信息最大的 P_{X^*} 和 Δ^*。通过这种方法得到的 P_{X^*} 是次优分布（数值结果表明，采用 P_{X^*} 与最优分布在传输速率相同时所需要的SNR相差很小，基本可以忽略）。

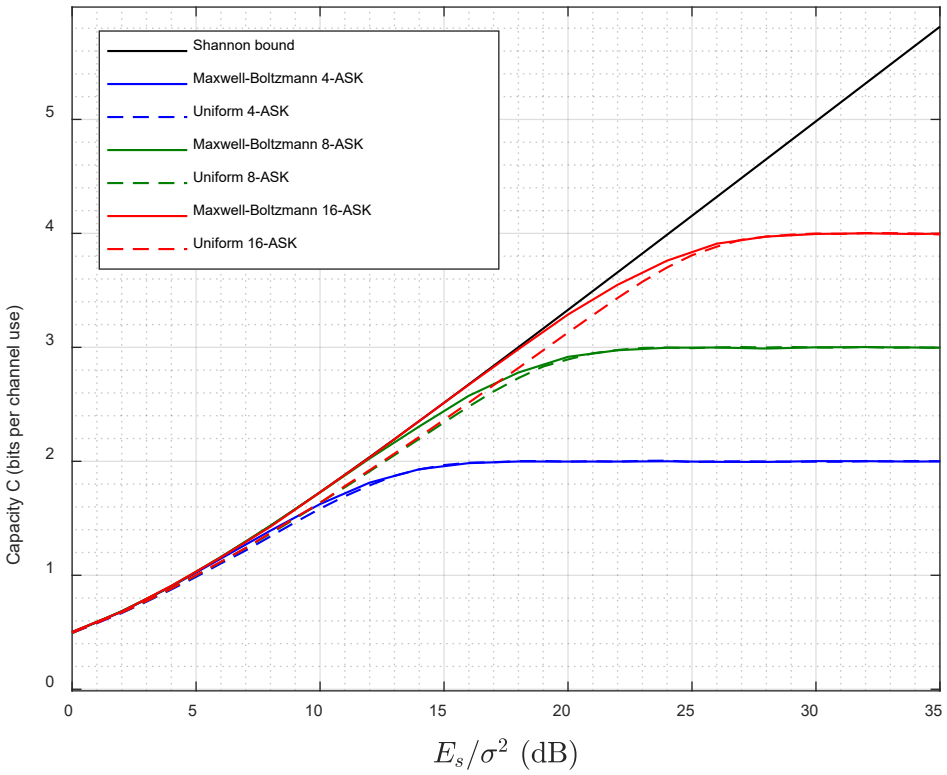


图5.13.22

星座符号 X 可以表示为 $X = A \cdot S$。其中，$A = |X|$ 表示幅度，$S = \text{sign}(X)$ 表示极性。A 的取值范围为 $\mathcal{A} = \{1, 3, ..., 2^m - 1\}$，用 $m-1$ 个比特表示幅值 A，用 1 个比特表示极性 S。对于对称于原点的 ASK 信号星座，其分布满足

$$P_X(x) = P_A(|x|) \cdot P_S(\text{sign}(x)), \quad x \in \mathcal{X}$$
$$P_S(+1) = P_S(-1) = 1/2$$

概率幅度成形系统框图如图5.13.23所示。其基本思想是使用分布匹配器产生非均匀的幅度序列，然后使用FEC码产生均匀分布的极性序列（注：FEC码的校验位 are the modulo-two sum of a large enough number of data bits, then their distribution is close to uniform）。用 N 表示成形后的信号长度。将信息比特分为两部分，第一部分的长度为 k，记为 U^k；第二部分的长度为 γN，$0 \le \gamma < 1$，记为 $U^{\gamma N}$。分布匹配器将 U^k 映射为非均匀的幅度序列 A^N，且该映射是可逆的。我们期望分布匹配器输出的符号分布接近最优分布。分布匹配器可以是恒定分量匹配器（Constant composition distribution matcher, CCDM）[Schulte-TIT2016]或者枚举球成形器（Enumerative sphere shaping，ESS）[Gültekin-TWC2020]。对于分布匹配器，定义成形速率为

$$R_s = k / N \qquad (2\text{-}1)$$

系统的传输速率（per channel use）为

$$R_{\mathrm{t}} = k / N + \gamma \qquad (2\text{-}2)$$

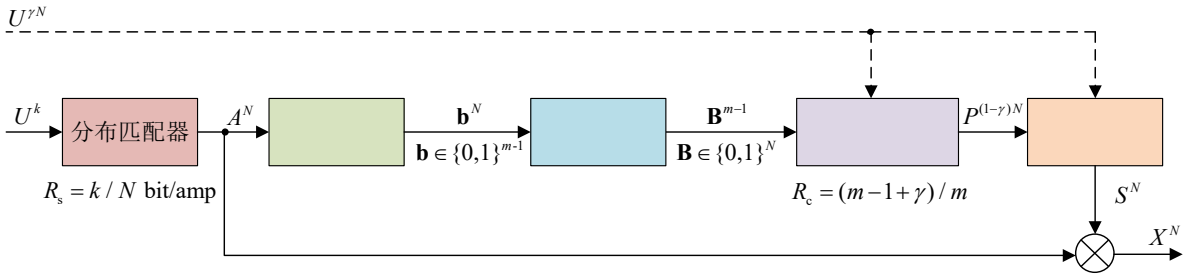


图 5.13.23 PAS 系统的一般传输结构

对于每个幅度 A，用一个长度为 $m-1$ 的二进制标号 $\mathbf{b} = \left(b^{(1)} b^{(2)} \cdots b^{(m-1)}\right) \in \{0,1\}^{m-1}$ 唯一表示。如果采用比特交织编码调制（BICM）方式，则使用一个比特交织器对序列 $\mathbf{b}^N$ 进行行列交织，如图 5.13.24 所示，交织器大小为 m-1 行，N 列。将 N 个二进制标号 $\mathbf{b}_n$ 按列写入，按行读出，记交织后 level-i 的 N 长比特序列为 $\mathbf{B}_i$，则有 $\left(\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_{N-1}\right) \Rightarrow \left(\mathbf{B}_1 \mathbf{B}_2 \cdots \mathbf{B}_{m-1}\right)$。

例如，对于 16-ASK，m=4，幅度序列 A^N 对应的二进制标号序列及交织后序列为

$$\left(\mathbf{b}(A_0), \mathbf{b}(A_1), \cdots, \mathbf{b}(A_{N-1})\right) = \left(b_0^{(1)} b_0^{(2)} b_0^{(3)}, b_1^{(1)} b_1^{(2)} b_1^{(3)}, \cdots, b_{N-1}^{(1)} b_{N-1}^{(2)} b_{N-1}^{(3)}\right)$$

$$\rightarrow \left(b_0^{(1)} b_1^{(1)} \cdots b_{N-1}^{(1)}, b_0^{(2)} b_1^{(2)} \cdots b_{N-1}^{(2)}, b_0^{(3)} b_1^{(3)} \cdots b_{N-1}^{(3)}\right) \triangleq \left(\mathbf{B}_1 \mathbf{B}_2 \mathbf{B}_3\right)$$

$$\begin{array}{c} \begin{matrix} \mathbf{b}_0 & \mathbf{b}_1 & & \mathbf{b}_{N-1} \end{matrix} \\ \left[\begin{matrix} b_0^{(1)} & b_1^{(1)} & \cdots & b_{N-1}^{(1)} \\ b_0^{(2)} & b_1^{(2)} & \cdots & b_{N-1}^{(2)} \\ \vdots & \vdots & & \vdots \\ b_0^{(m-1)} & b_1^{(m-1)} & \cdots & b_{N-1}^{(m-1)} \end{matrix}\right] \begin{matrix} \rightarrow \mathbf{B}_1 \\ \rightarrow \mathbf{B}_2 \\ \vdots \\ \rightarrow \mathbf{B}_{m-1} \end{matrix} \end{array} \xrightarrow{\mathbf{B}_1 \mathbf{B}_2 \cdots \mathbf{B}_{m-1}} \mathrm{FEC}$$

图 5.13.24

在文献[Böcherer- Tcom2015]中，还引入一个 bit-level 交织器对送入 FEC 编码器的 $\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_{m-1}$ 的顺序进行置换，以优化比特映射器。记置换后的序列为 $\mathbf{B}^{m-1} = \left(\mathbf{B}_{\pi(1)} \mathbf{B}_{\pi(2)} \cdots \mathbf{B}_{\pi(m-1)}\right)$。例如，对于 m=4，bit-level 交织器 $\pi = (2,1,3)$ 表示 $\left(\mathbf{B}_1 \mathbf{B}_2 \mathbf{B}_3\right) \xrightarrow{\pi} \left(\mathbf{B}_2 \mathbf{B}_1 \mathbf{B}_3\right)$。然后用系统形式的 FEC 编码器对 $\mathbf{B}^{m-1}$ 进行编码，产生 $N(1-\gamma)$ 个均匀分布的校验比特 $P^{(1-\gamma)N}$。FEC 码率为

$$R_{\mathrm{c}}=\frac{m-1+\gamma}{m} \qquad (2\text{-}3)$$

将 $P^{(1-\gamma)N}$ 和 $U^{\gamma N}$ 映射成极性序列 S^N。然后根据 $X_i=A_i\cdot S_i$， $i\in 1,\ldots,n$，就可得到信道的输入 X^N。

■ **CCDM 及其实现：**

为了生成非均匀分布的信号点，需要一个能将均匀分布的输入比特转化为服从所要求分布的输出符号序列的分布匹配器。CCDM 是一种典型的分布匹配器，下面做简要介绍。

定义一个幅度序列 a^{n_c}，用 n_α 表示幅度 α 出现的次数，即 $n_\alpha=\left|\{i:a_i=\alpha\}\right|$。用 P_A 表示 a^{n_c} 的经验分布（即 types），即

$$P_A(\alpha)=n_\alpha/n_c,\quad \alpha\in\mathcal{A} \qquad (2\text{-}11)$$

可以发现， a^{n_c} 的所有置换序列都满足经验分布 P_A。这个序列集合我们用 $\mathcal{T}_{P_A}^{n_c}$ 表示，称为 P_A 的type class。如果一个集合中的所有序列是同一类型的，则称该集合为constant composition code。CCDM的输出序列会形成一个恒定分量码。

现假设 a^{n_c} 的经验分布满足以下条件，

$$n_\alpha\approx n_c\cdot P_A(\alpha),\quad \alpha\in\mathcal{A} \qquad (2\text{-}12)$$

建立 $\mathcal{T}_{P_A}^{n_c}$ 和信息序列的一一映射关系，则信息序列的长度为

$$k_{\mathrm{c}}=\left\lfloor \log_2\left|\mathcal{T}_{P_A}^{n_c}\right|\right\rfloor=\left\lfloor \log_2\frac{n_{\mathrm{c}}!}{n_1!n_3!\cdots n_{2^m-1}!}\right\rfloor \qquad (2\text{-}13)$$

从 $\mathcal{T}_{P_A}^{n_c}$ 中选择一个子集 $\mathcal{C}_{\mathrm{ccdm}}$，其中 $\left|\mathcal{C}_{\mathrm{ccdm}}\right|=2^{k_c}$，建立 $\{0,1\}^{k_c}$ 和 $\mathcal{C}_{\mathrm{ccdm}}$ 的一一映射关系。当 n_c 足够长时，CCDM 的输出分布能够很好地近似 P_A。

举例：Consider the 4-ASK constellation $\mathcal{X}=\{\pm1,\pm3\}$ and the block length $n_{\mathrm{c}}=4$. The amplitudes are A = {1, 3}. Suppose the desired distribution is

$$P_A(1)=3/4,\quad P_A(3)=1/4$$

The probabilities are integer multiples of 1/4, so P_A is a 4-type. The corresponding type class of length $n_c=4$ sequences is

$$\mathcal{T}_{P_A}^4=\{(1,1,1,3),\ (1,1,3,1),\ (1,3,1,1),\ (3,1,1,1)\}$$

A CCDM maps length k_{c} binary strings to sequences in $\mathcal{T}_{P_A}^4$， so we have $k_c=\log_2|\mathcal{T}_{P_A}^4|=2$.

The following lookup table defines a CCDM:

$$00 \mapsto (1,1,1,3), \quad 01 \mapsto (1,1,3,1),$$
$$10 \mapsto (1,3,1,1), \quad 11 \mapsto (3,1,1,1).$$

需要说明的是，CCDM 的速率损失对于大块长可以忽略不计，但对于短块，CC（the constant composition）约束增加了大量冗余，导致一定的速率损失。为此，文献[Gültekin-TWC2020]提出了枚举球成形方法。

实现：给定P_A后，我们首先用一个n-type分布 $P_{\bar{A}}=\left\{\frac{n_1}{n},\frac{n_3}{n},\cdots,\frac{n_{2^m-1}}{n}\right\}$来近似$P_A$，其中 $n=n_1+n_3+\cdots+n_{2^m-1}$。因为$n_\alpha$取值的选择有多种可能组合，我们基于相对熵来优化$n_\alpha$的取值：

$$P_{\bar{A}}=\arg\min_{P_{\bar{A}'}} \mathbb{D}\left(P_{\bar{A}'}\,\|\,P_A\right) \quad \text{s.t. } P_{\bar{A}'} \text{ is } n\text{-type}$$

该优化问题的解可用下列算法求得[Böcherer2015, Algorithm 2]，其中

$$\Delta_i(k)=k\log k-(k-1)\log(k-1)-\log P_A(i)$$

Algorithm 2. Informational divergence optimal quantization.

Initialize $c_i \leftarrow 0, i=1,\ldots,n.$
for $m=1,2,\ldots,M$
  Choose $j=\min\{\operatorname{argmin}_i \Delta_i(c_i+1)\}$. *//choose the smallest index first.*
  Update $c_j \leftarrow c_j+1.$
end for
Return $\boldsymbol{c}$.

令$\mathcal{T}^n_{P_{\bar{A}}}$表示类型为$P_{\bar{A}}$的符号序列集合，我们将CCDM的输入比特序列长度置为 $k=\left\lfloor \log_2 |\mathcal{T}^n_{P_{\bar{A}}}| \right\rfloor$，这样就有CCDM编码函数$f$：$\{0,1\}^k \to \mathcal{T}^n_{P_{\bar{A}}}$。实际映射关系可用算术编码来实现，详细介绍见文献[Schulte-TIT2016]。

例1： $\mathcal{A}=\{1,3\}$，所希望的分布是$P_A(1)$=0.84，$P_A(3)$=0.16。设n=6，使用上述算法获得 $P_{\bar{A}}(1)=5/6, P_{\bar{A}}(3)=1/6$，$k$=2。基于算术编码的映射关系如图5.13.25所示。

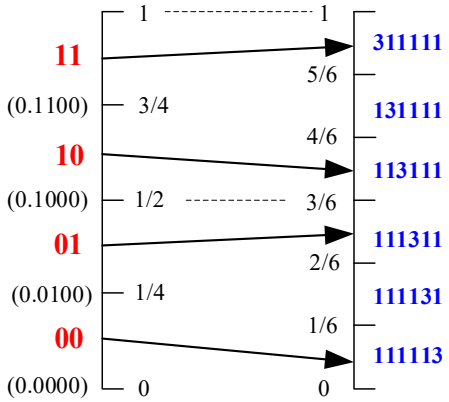


Fig. 5.13.25 Diagram of a constant composition arithmetic encoder with k=2, n=6

The intervals are ordered lexicographically. The arithmetic encoder can link an output sequence to an input sequence if the lower border of the output interval is inside the input interval.

■ **枚举球成形（ESS）：**

文献[Gültekin-TWC2020]使用球码来实现成形。定义序列的最大能量 $E_{\max}$。用 $S_{E_{\max}}^{N}$ 表示能量受限的 N 长幅度序列集合，即

$$S_{E_{\max}}^{N}=\left\{s_1,s_1,\ldots,s_N \left| \sum_{n=1}^{N} s_n^2 \le E_{\max} \right.\right\} \tag{2-4}$$

对球码中的幅度序列规定一种字典排序方式：对于序列 $a^N=a_1,a_2,\ldots,a_N$ 和 $b^N=b_1,b_2,\ldots,b_N$，如果存在一个整数 n，满足 $a_n>b_n$，并且当 $1\le j\le n-1$ 时，$a_j=b_j$，那么就称 $a^N>b^N$。基于这种大小关系，对每个幅度序列分配一个索引

$$i\left(a^N\right) \triangleq \left|\left\{b^N\in S_{E_{\max}}^{N} : a^N>b^N\right\}\right| \tag{2-5}$$

有界能量网格图如图 5.13.26 所示。网格图的列数为 $N+1$，行数为 $L=\left\lfloor \left(E_{\max}-N\right)/8 \right\rfloor+1$。最左边的节点为起始节点，最右边的节点为终止节点。一条路径从起始节点出发到达终止节点，共包含 $N+1$ 个节点和 N 个分支。每个分代表了一个幅度，因此一条路径定义了一个幅度序列。起始节点的能量为 0。路径在经过第 n 列节点时具有的累积能量为 $e\left(a^n\right)=\sum_{k=1}^{n} a_k^2$，$n\in\{1,\ldots,N\}$。用 T_n^e 表示从第 n 个节点出发到达终止节点的路径总数。对于 $n=N-1,N-2,\ldots,0$

$$T_n^e \triangleq \sum_{a\in\mathcal{A}} T_{n+1}^{e+a^2} \tag{2-6}$$

对于终止节点，$T_N^e=1$。通过枚举球成形得到的概率分布为

$$P_A(a)=\frac{T_1^{a^2}}{\sum_{b\in\mathcal{A}}T_1^{b^2}} \tag{2-7}$$

序列的平均能量为

$$E_{\text{av}}=N\sum_{a\in\mathcal{A}}P_A(a)\,a^2 \tag{2-8}$$

球码的成形速率为

$$R_{\text{s}}=\left\lfloor \log_2 T_0^0 \right\rfloor / N \tag{2-9}$$

图 5.13.26 中 $\mathcal{A}=\{1,3,5\}$，$E_{\max}=28$，共有 $T_0^0=19$ 个能量序列。球码所能支持的最大信息长度为 4。对于 $a\in\{1,3,5,7\}$，概率分布为 $P_A(a)=\{11/19,7/19,1/19,0\}$，序列的平均能量为 $E_{\text{av}}=20.84$。

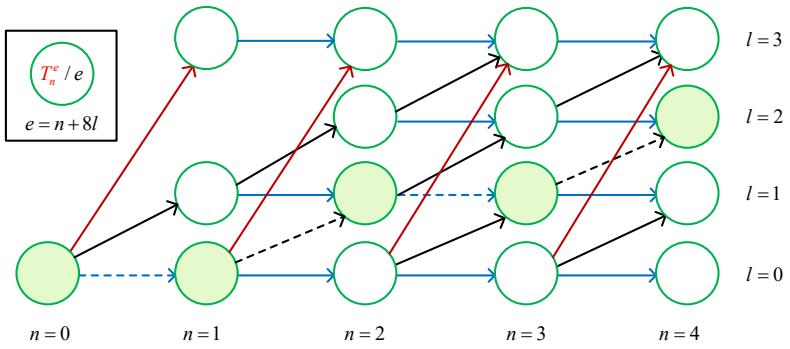


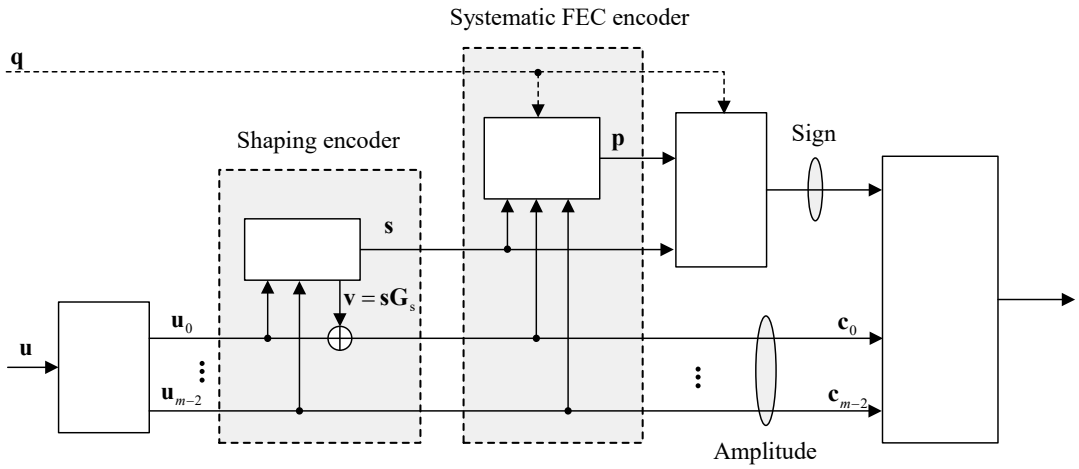
图 5.13.26 有界能量网格图，$\mathcal{A}=\{1,3,5,7\}$，$N=4$，$E_{\max}=28$

成形和解成形运算在网格图上进行，因此需要将 T_n^e 存储起来。

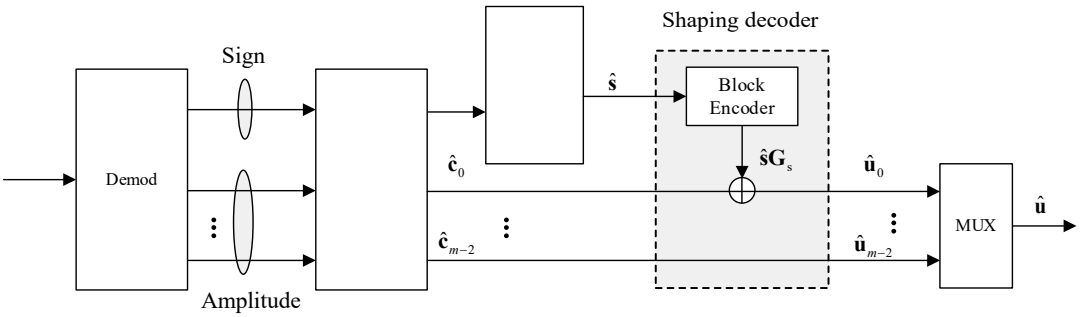
■ **基于线性分组码的 PAS 成形:**

文献[Matsumine-Tcom2021]提出了基于线性分组码的 PAS 成形方案，原理框图如图 5.13.27 所示。将信息序列 $\mathbf{u}$ 分成 $(m-1)$ 路比特流。成形对第 0 层比特流进行操作。选用线性分组码作为成形码。用 $\mathbf{G}_{\text{s}}$ 表示成形码的生成矩阵。$\mathbf{G}_{\text{s}}$ 的大小为 $K\times N$。成形码字 $\mathbf{v}=\mathbf{s}\mathbf{G}_{\text{s}}$。将成形码字叠加到第 0 层比特流，得到 $\mathbf{c}_0=\mathbf{u}_0+\mathbf{v}$。对于系统 FEC 编码器，输入包括附加信息、$\mathbf{s}$ 和 $(m-1)$ 个比特层。控制信号极性的比特包括附加信息、$\mathbf{s}$ 和校验比特。对于该系统，传输速率为 $(m-1+\gamma)$ 比特/一维符号，成形冗余为 K/N 比特/一维

符号。



(a) 发送端



(b) 接收端

图 5.13.27 基于线性分组码的 PAS 系统

性能比较：for 64-QAM

设定目标频谱效率为 4 比特/二维符号。对于每个实维，频谱效率为 2 比特/一维符号。对于 ESS，分布匹配器的成形速率为 $R_s=1.5$ 比特/一维符号，附加信息速率为 $\gamma=0.5$ 比特/一维符号。对于基于线性码的成形方案，分布匹配器的成形速率为 $R_s=2$ 比特/一维符号，附加信息速率为 $\gamma=0$ 比特/一维符号。FEC 码率为 5/6。FEC 码长为 648 比特。仿真结果如图 5.13.28 所示。在 $\mathrm{FER}=10^{-3}$ 时，相较于均匀分布的情形，使用 64 状态卷积码的成形方案约有 0.8 dB 的增益。使用 4 状态卷积码和扩展戈莱码具有相同的增益，约为 0.6 dB。

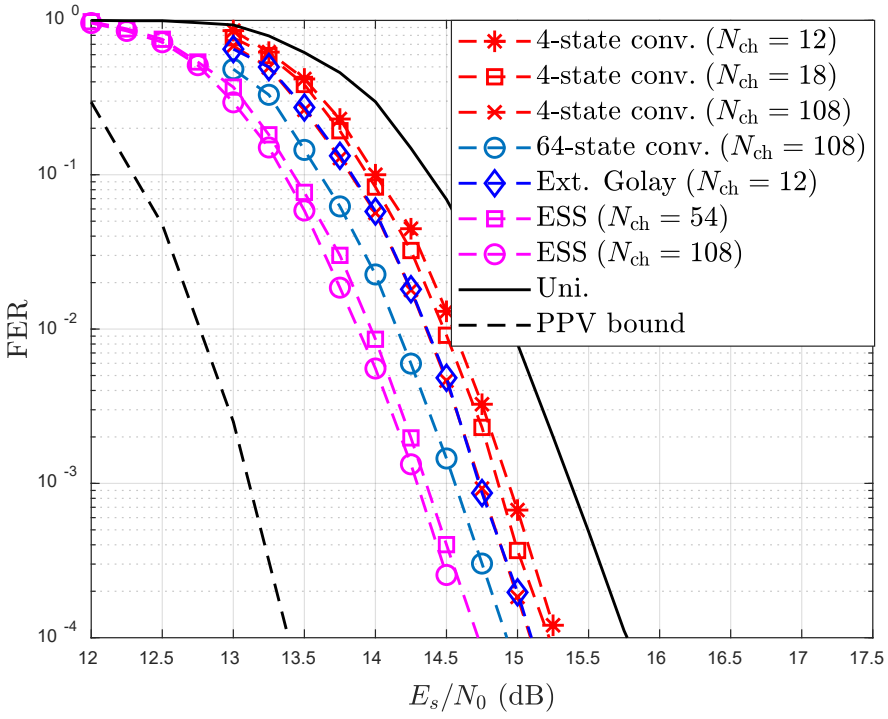


图 5.13.28 多种 PAS 方案下的 FER 性能，调制方式为 64-QAM，频谱效率为 4 比特/二维符号

5.14 Implementation Considerations and Application Examples

Coded-modulations have been widely used in various modern communication systems, including voice modems as typical examples for TCM applications. Other applications include mobile communications, DVB, digital TV, and etc.

5.14.1 Multidimensional TCM Example: The V.34 Modem Standard

（V.34, V.92, Pragmatic TCM）

In this subsection, we discuss the TCM scheme used for the V.34 voice-band modem standard. This modem is capable of transmitting up to 33.6 kbps over the standard telephone system. Some of the aspects related to the error correction coding are summarized as follows [Forney96].

- The modem is adaptive in the symbol rate it employs and the size of the constellation. It is capable of sending at symbol rates of 2400, 2743, 2800, 3000, 3200, or 3429 symbols/second. (These peculiar-looking choices are rational multiples of the basic rate of 2400 symbols/second.) The symbol rate is selected by a line probing sequence employed during initialization which determines the available bandwidth.
- The modem is capable of transmitting variable numbers of bits per symbol. At the highest rate, 8.4 bits/symbols are carried. Rate is established at link initialization, and rate adjustments can occur during data transmission.
- Adaptive precoding and decision feedback equalization is also employed to compensate for channel dispersion.
- Shaping via shell mapping is employed, which provides modest gains in addition to

the coding gains.

- Adaptive trellis coding is employed. The main code is a 16-state four-dimensional trellis code. This code provides 4.66 dB of gain and is rotationally invariant. However, two other trellis codes are also included: a 32-state four-dimensional code with 4.5 dB of gain and a 64-state four dimensional code providing 4.7 dB of gain.

V.34 modem的编码调制系统框图如图5.14.1所示. The trellis encoder for the modem is shown in Figure 5.14.2, with the corresponding convolutional encoder in Figure 5.14.3.

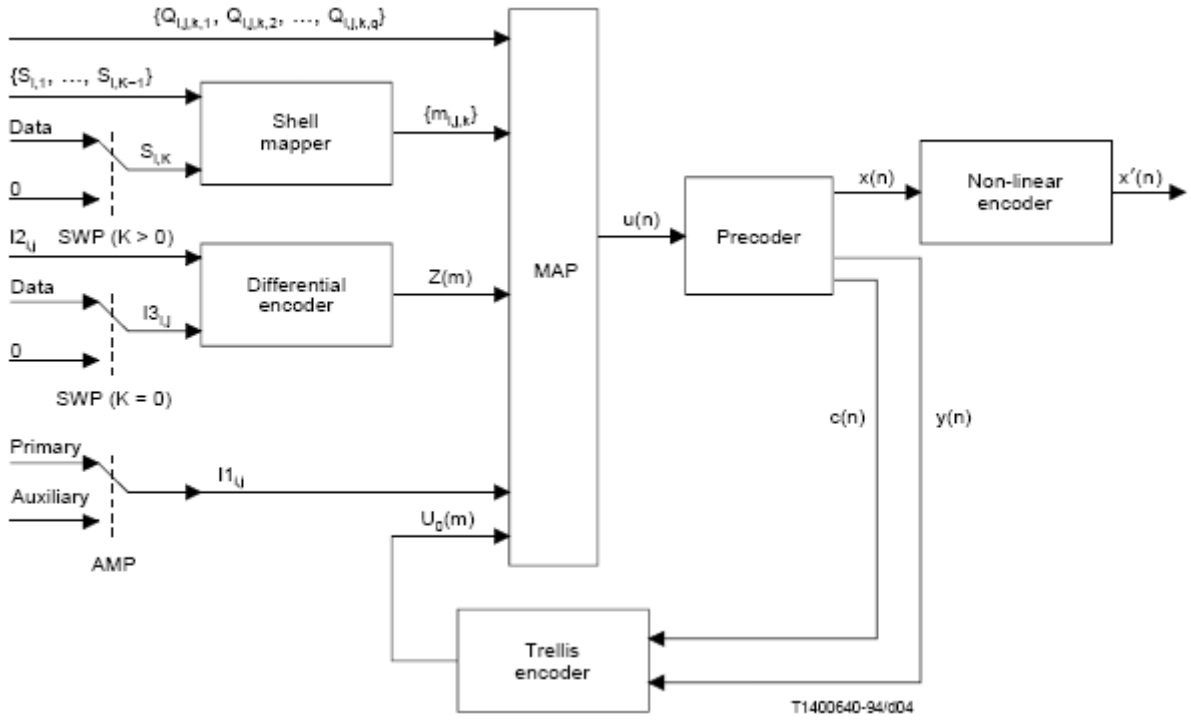


Figure 5.14.1 V.34 – Encoder block diagram

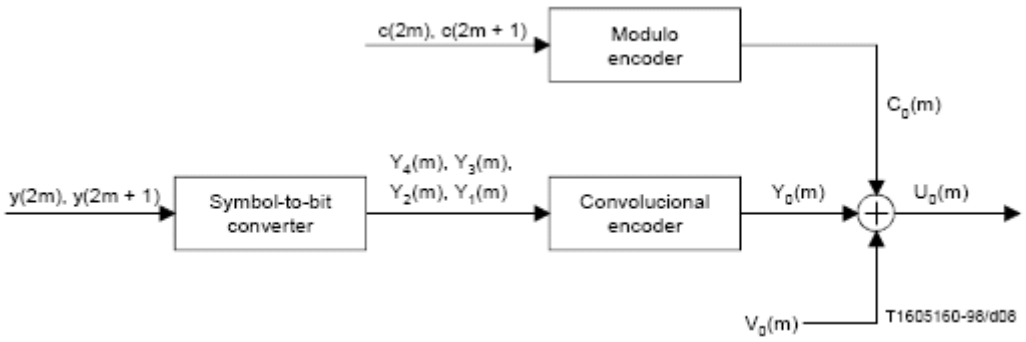


Figure 5.14.2 V.34 – Block diagram of trellis encoder

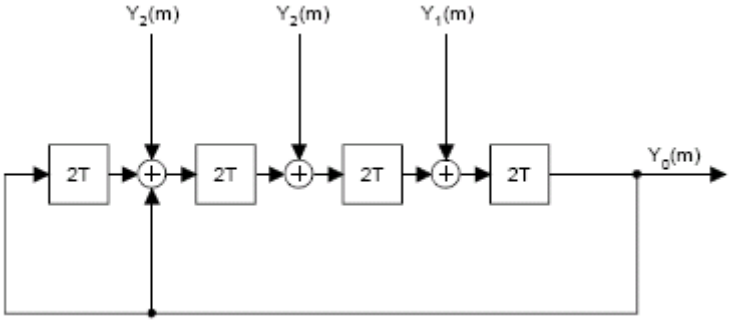


Figure 5.14.3 V.34 – 16-state convolutional encoder

5.14.2 TCM Schemes in Deep Space Applications

CCSDS 建议了 a 4D 8-PSK trellis-coded modulation for use in bandwidth-constrained communications between remote satellites and Earth stations. 它可以实现分数谱效率。

Example: Assuming the bit rate R_b of input data equal to 100 Mbps, the 4D-8PSK-TCM channel symbol rate is 50 Msps for 2 bits/channel-symbol or 40 Msps for 2.5 bits/channel-symbol.

The 4D-8PSK trellis-coded modulator consists of a serial to parallel converter, a differential coder, a 64 状态、3/4 码率的 convolutional coder, a constellation mapper and an 8PSK modulator. Spectral efficiencies of 2, 2.25, 2.5, and 2.75 bits/channel-symbol are achieved with four possible architectures of the constellation mapper. 谱效率为 2.5b/s/Hz 时的编码器结构如图 5.14.4 所示。Carrier phase ambiguity is resolved by the use of a differential coder located prior to the trellis encoder. The output switch addresses successively one of the four symbols ($Z^{(0)}$ - $Z^{(3)}$) from the constellation mapper to the 8PSK modulator.

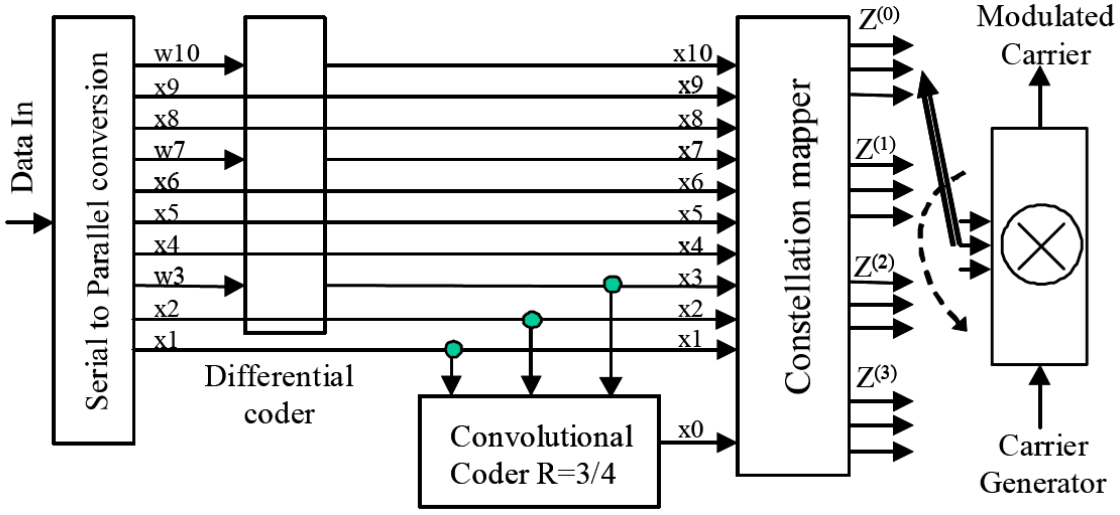
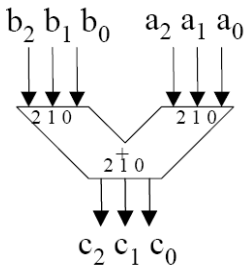


Figure 5.14.4. Coder and Mapper Implementation at 2.5 Bits/Channel-Symbol

The differential coder depicted in figure 5.14.4 is used to eliminate phase ambiguity on carrier synchronization. 在 2.5 bits/channel-symbol 时，the input bits to differential coder are w3, w7 and w10, and the output bits are x3, x7, and x10. The structure of the modulo-8 adder is shown in figure 5.14.5; it is applicable to both the coder mapper and differential coder.



$c_0 = a_0 \oplus b_0$ with carry r_0
$c_1 = a_1 \oplus b_1 \oplus r_0$ with carry r_1
$c_2 = a_2 \oplus b_2 \oplus r_1$
$r_0 = a_0.b_0$
$r_1 = (a_1.b_1) + (a_1.r_0) + (b_1.r_0)$

图 5.14.5 modulo-8 adder

卷积编码器的结构如图 5.14.6 所示，其连接多项式为（8 进制）：h3=050, h2=024, h1=006, h0=103.

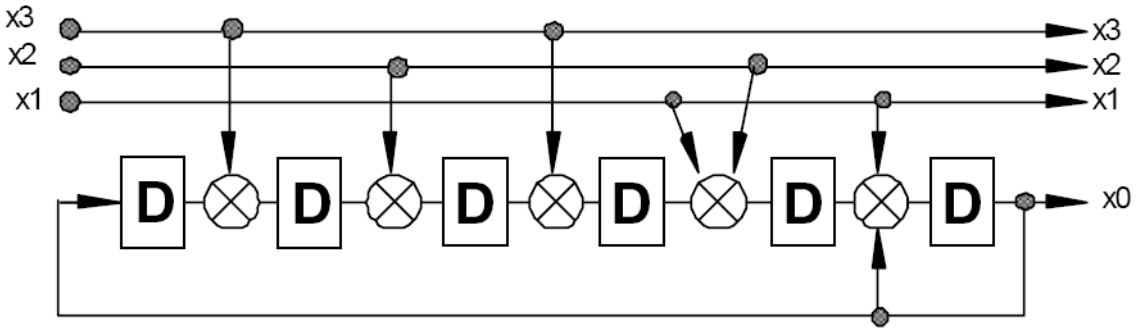


图 5.14.6. 64 状态 convolutional encoder

The constellation mapper principle is given in figures 5.14.7 for the efficiency of 2.5 bits/channel-symbol. The correspondence between the signals $Z^{(i)}$ at the input of the modular and the 8-PSK phase states of the constellations follows a natural mapping (i.e., 0, 1, 2, …, 7).

$$\begin{bmatrix} Z^{(0)} \\ Z^{(1)} \\ Z^{(2)} \\ Z^{(3)} \end{bmatrix} = \left[\left(4x^{(10)} + 2x^{(7)} + x^{(3)}\right) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + 4 \begin{pmatrix} 0 \\ x^{(9)} \\ x^{(8)} \\ x^{(9)} + x^{(8)} + x^{(6)} \end{pmatrix} + 2 \begin{pmatrix} 0 \\ x^{(5)} \\ x^{(4)} \\ x^{(5)} + x^{(4)} + x^{(2)} \end{pmatrix} + \begin{pmatrix} 0 \\ x^{(1)} \\ x^{(0)} \\ x^{(1)} + x^{(0)} \end{pmatrix} \right] \bmod 8$$

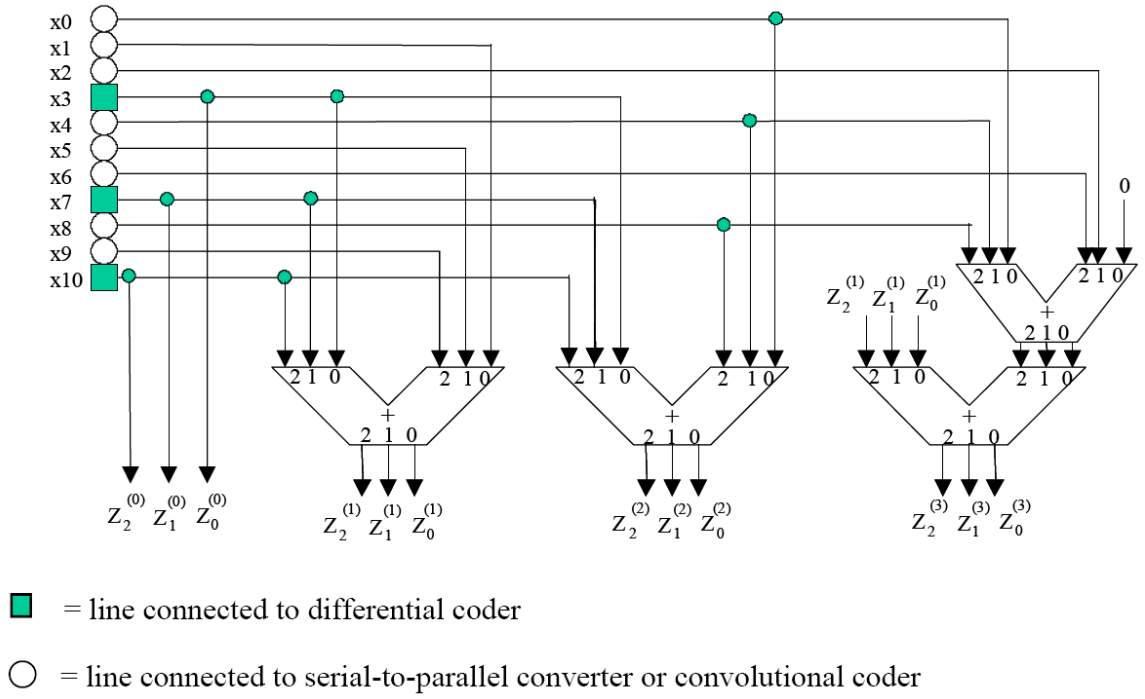


Figure 5.14.7: Constellation Mapper for 2.5 Bits/Channel-Symbol

5.14.3 BICM Scheme for 5G Data Channel

5G移动通信系统数据信道的LDPC编码链如图5.14.8 所示。物理层接收到MAC层发来的传输块（Transport Block, TB），根据信息比特长度和码率对其添加CRC并分割成能够进行编码的码块（Code Block, CB），经过LDPC编码后，根据冗余版本信息进行速率适配，然后对各CB进行比特交织和码块级联，最后映射到高阶调制信号星座。

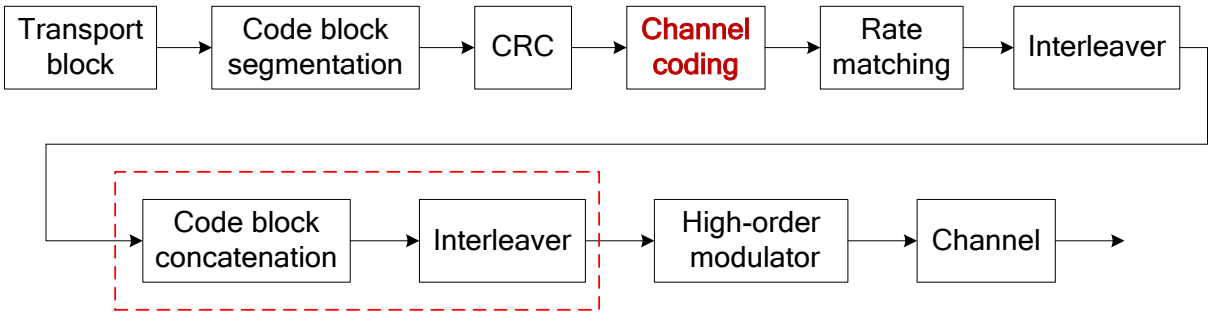


Figure 5.14.8 5G数据信道的LDPC编码链

References

[1] E. Zehavi, “8-PSK trellis codes for a Rayleigh channel,” *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 873–884, 1992.

[2] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, 1998.

[3] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, “Multilevel codes: Theoretical concepts and practical design rules,” *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1361–1391, July 1999.
[4] J. Tan and G. Stuber, “Analysis and design of symbol mappers for iteratively decoded BICM,” *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, pp. 662–672, 2005.
[5] J. L. Massey, “Coding and modulation in digital communications,” in *International Zurich Seminar Digital Communications, Zurich, Switzerland*, 1974.
[6] A. Martinez, A. Guill′en i F`abregas, G. Caire, and F. Willems, “Bit-interleaved coded modulation in the wideband regime,” *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5447–5455, December 2008.
[7] A. Martinez, A. Guill′en i F`abregas, G. Caire, and F. Willems, “Bit-interleaved coded modulation revisited: A mismatched decoding perspective,” *submitted to IEEE Transactions on Information Theory*, http://arxiv.org/abs/0805.1327, 2008.
[8] X. Li, A. Chindapol, and J. A. Ritcey, “Bit-interleaved coded modulation with iterative decoding and 8 PSK signaling,” *IEEE Transactions on Communications*, vol. 50, no. 8, pp. 1250–1257, 2002.
[9] X. Li and J. A. Ritcey, “Trellis-coded modulation with bit interleaving and iterative decoding,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 715–724, 1999.
[10] X. Li and J. A. Ritcey, “Bit-interleaved coded modulation with iterative decoding,” *IEEE Communication Letters*, vol. 1, no. 6, pp. 169–171, November 1997.
[11] R. F. H. Fischer, *Precoding and Signal Shaping for Digital Transmission*. USA, New York：John Wiley & Sons, Inc., 2002.
[12] Xiao Ma and Li Ping, “Coded modulation using superimposed binary codes,” *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3331–3343, December 2004.
[13] 马啸、白宝明，“信号星座的概率成形与几何成形”，pp.166-169，中国电子学会第13届信息论学术年会，2005年10月，中国长沙。
[14] G. D. Forney, Jr, “Trellis shaping,” *IEEE Transactions on Information Theory*, vol. 38, pp. 281–300, 1992.
[15] G. D. Forney, Jr, M.D. Trott, and S.-Y. Chung, “Sphere-bound-achieving coset codes and multilevel coset codes,” *IEEE Trans. Inform. Theory*, vol. 46, pp. 820–850, May 2000.
[16] G. Böcherer, F. Steiner, and P. Schulte, “Bandwidth efficient and rate-matched low-density parity-check coded modulation,” *IEEE Trans. Commun.*, vol. 63, no.12, Dec. 2015.
[17] P. Schulte and G. Böcherer, “Constant composition distribution matching,” *IEEE Transactions on Information Theory*, vol. 62, no. 1, pp. 430-434, Jan. 2016.
[18] Y. C. Gültekin, W. J. van Houtum, A. G. C. Koppelaar, and F. M. J. Willems, “Enumerative sphere shaping for wireless communications with short packets,” *IEEE Trans. Wireless Commun*., vol. 19, no. 2, pp. 1098-1112, Feb. 2020.
[19] T. Matsumine, T. Koike-Akino, and H. Ochiai, “A low-complexity probabilistic amplitude shaping with short linear block codes,” *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 7923 – 7933, Dec. 2021.