Lecture 6

Related Topics of TCM

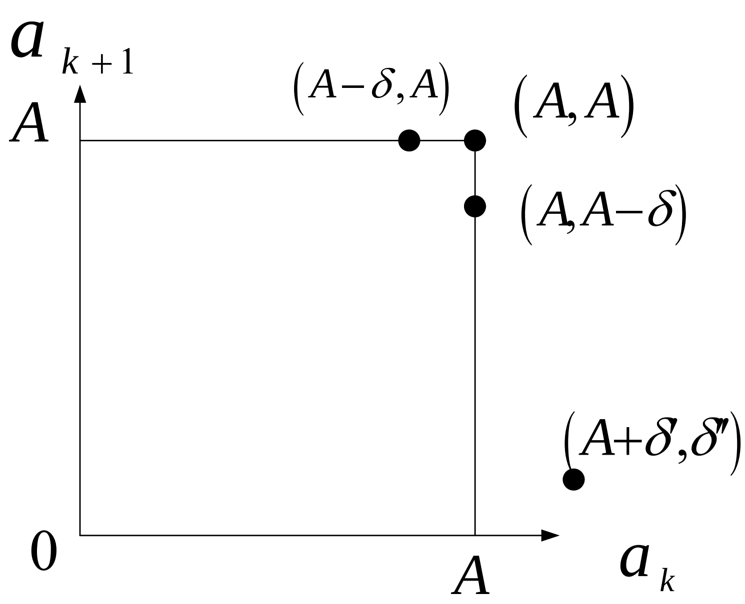
Reference:
R. Laroia, N. Farvardin and S.A. Tretter," On optimal shaping of multidimensional constellations, " *IEEE Tran. Inform. Theory,* pp.1044-1056, July 1994 (algorithm 2 in p.1048,1049)

Shell Mapping

- There is a modest but not insignificant gain in forming signal constellations in high-dimensional spaces to minimize average signal power, quite independent of coding gain.

- This so-called shaping gain can never be greater than a factor of $\frac{\pi e}{6}$ (1.53dB); however, it is not difficult to achieve shaping gain on the order of 1dB.

Consider a constellation whose maximum value of amplitude is *A*. Let *M* denote the constellation size. At time *k* and time *k*+1, two signal points x_k and x_{k+1} are chosen from this constellation respectively. Let a_k and a_{k+1} represent the amplitude of x_k and x_{k+1} respectively. There are M^2 possibilities in choosing x_k and x_{k+1}, including



$(a_k, a_{k+1}) = (A, A), (A-\delta, A), (A, A-\delta), \cdots$ etc. However, $(a_k = A, a_{k+1} = A)$ needs signal power $2A^2$, while $(a_k = A+\delta', a_{k+1} = \delta'')$ may need less signal energy $(< 2A^2)$ if an expanded constellation is used.

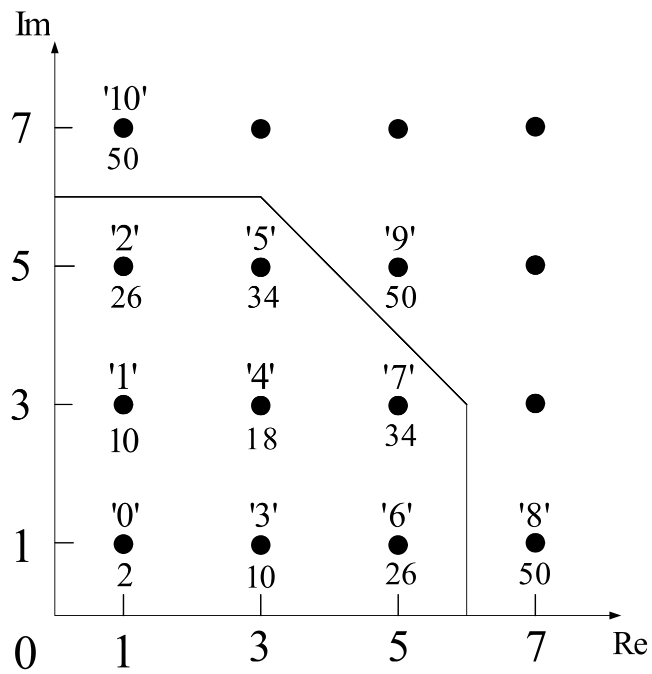
- That means, to form the set S of two signal points (x_k, x_{k+1}) with minimum average power, one should choose $S = \left\{(x_k, x_{k+1}) \mid a_k^2 + a_{k+1}^2 \leq C\right\}$, where C is a constant decided according to the size of S.

- The optimum shape in theory is a sphere in a high number of dimensions, but spherical constellations are difficult to implement and yield excessive constellation, which can lead to greater susceptibility to nonlinear impairments.

For example,

M=8

- (0,0) 4
- (0,1), (0,3), (1,0), (3,0) — 12
- (0,4), (4,0), (1,3), ⋮ — 20
- ⋮ ⋮
- ⋮, (7,2), (7,6) — 60
- (7,7), (7,5), (5,7), (5,5) — 68



- (0,8), (0,9), (0,10), (8,0), (9,0) — 52
- (10,0)
- (1,8), ⋮ — 60

- After consideration of several shaping methods, a technique called shell mapping was eventually included in V.34. Shell mapping is an algorithmic method of achieving near-spherical constellation shaping in a high number of dimensions with bounded QAM constellation expansion.

- V.34 specifies shell mapping in 16 dimensions with QAM constellation expansion limited to about 25 percent, which yields a shaping gain of about 0.8dB. It also includes a shaping option with essentially no constellation, which still achieves a shaping gain of about 0.2dB.

⇨ An Example

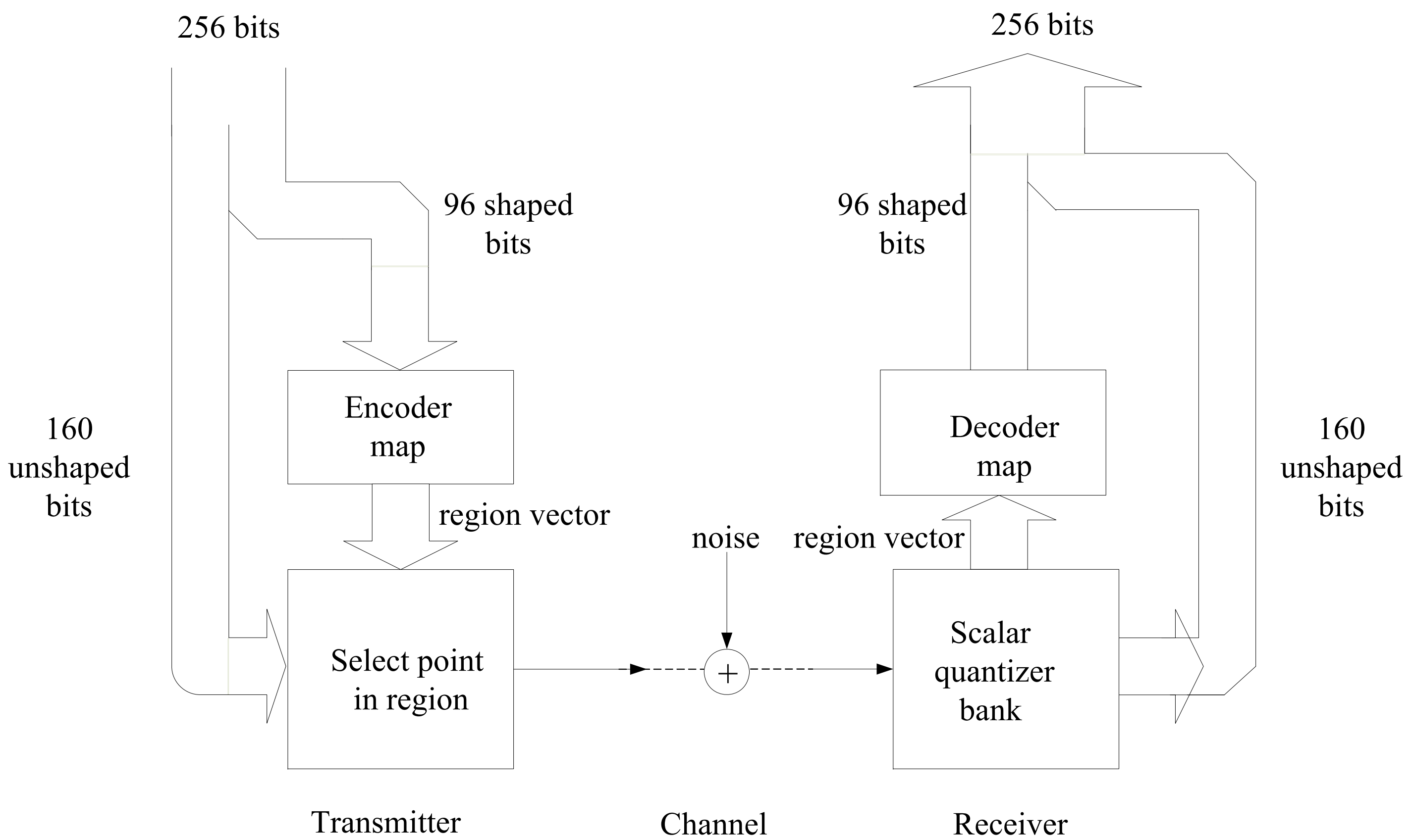
Assume that it is desired to transmit binary data using a 64-dimensional shaped uncoded constellation at the rate of 8 bits/2D. The constituent 2D constellation must consist of at least 256 points.

In this example, we assume that a shaping CER (constellation expansion ratio) of 1.5(corresponding to a 384-point 2D constellation) is acceptable. The 2D circular constellation A0 is partitioned into 12 regions R1,R2,…R12,each containing 32 points.

The region R1 consists of the 32 lowest energy(smallest squared distance from the origin) points in A0, R2 consists of the 32 next higher energy points in A0, and so on.

All 32 points in any given region Ri, $i \in J_{12} \equiv \{1, 2, \ldots, 12\}$ are used with the same probability.

We assign the same cost to all points in the same region. Every region Ri, is assigned a cost $i \in J_{12}$. The justification for this cost $l_i = i$ assignment is that for a large number of points in each region, the average cost of the region is approximately proportional to the region number.



⇨ Shell Mapper in V.34

input : K data bits ($S0,S1,\ldots,SK\text{-}1$)

output : the index of 8 2D signals($m0,m1,\ldots,m7$)

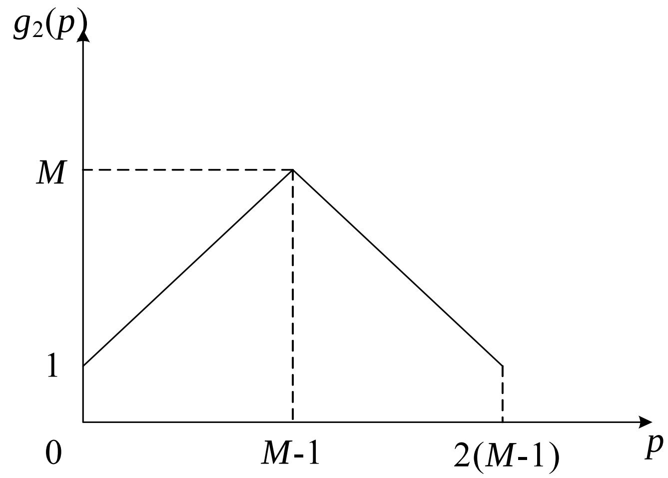
m1 is an integer satisfying $0 \le m_i < M \quad \forall i = 0 \sim 7$

☆ The 2D signal regions are M. Each region has its own cost $m \in \{0,1,2,\ldots,M-1\}$ which is roughly in proportion to the signal power.

☆ Let $g_i(p)$ represent the number of distinct i-vectors ($m0,\ldots,mi\text{-}1$) such that their total cost $\sum_{l=0}^{i-1} m_l$ is equal to $p, \forall i$ =2,4 and 8.

☆ $g_2(p) = M - |p - M + 1|, \ 0 \le p \le 2(M-1)$

$\qquad = 0 \qquad\qquad\qquad\qquad$, otherwise



$\because$ if $\ p=0 : (0,0) \rightarrow \ g_2(0) = 1$

$p=1 : (0,1),(1,0) \rightarrow \ g_2(1) = 2$

$\vdots$

$p \le (M-1) \ : (0,p),(1,p\text{-}1),\ldots,(p,0) \rightarrow \ g_2(p) = p+1$

$p > (M-1) \ : (p\text{-}(M\text{-}1),M\text{-}1),\ldots,(M\text{-}1,p\text{-}(M\text{-}1)) \rightarrow g_2(p) = 2M\text{-}p\text{-}1$

$\vdots$

$p=2M\text{-}3 : (M\text{-}2,M\text{-}1),(M\text{-}1,M\text{-}2) \rightarrow \ g_2(2M\text{-}3) = 2$

$p=2M\text{-}2 : (M\text{-}1,M\text{-}1) \rightarrow \ g_2(2M\text{-}2) = 1$

☆ $g_4(p) = g_2(0)\, g_2(p) + g_2(1)\, g_2(p\text{-}1) + \ldots\, g_2(p)\, g_2(0), \ \ 0 \le p \le 4(M-1)$

$\qquad = 0 \qquad\qquad\qquad\qquad\qquad\qquad$, otherwise

$g_8(p) = g_4(0)\, g_4(p) + g_4(1)\, g_4(p-1) + \ldots g_4(p)\, g_4(0), \quad 0 \le p \le 8(M-1)$

$= 0$, otherwise

- $Z_8(p) = g_8(0) + g_8(1) + \ldots + g_8(p-1)$ $, 0 \le p \le 8(M-1)$

- The rule of priority

$(m_0, m_1, \ldots, m_7)$ has priority over $(m'_0, m'_1, \ldots, m'_7)$ if

a. $\sum_{i=0}^{7} m_i < \sum_{i=0}^{7} m'_i$

b. $\sum_{i=0}^{7} m_i = \sum_{i=0}^{7} m'_i \;\&\; \sum_{i=0}^{3} m_i < \sum_{i=0}^{3} m'_i$

c. $\sum_{i=0}^{7} m_i = \sum_{i=0}^{7} m'_i \;\&\; \sum_{i=0}^{3} m_i = \sum_{i=0}^{3} m'_i \;\&\; \sum_{i=4}^{5} m_i < \sum_{i=4}^{5} m'_i$

d. $\sum_{i=0}^{7} m_i = \sum_{i=0}^{7} m'_i \;\&\; \sum_{i=0}^{3} m_i = \sum_{i=0}^{3} m'_i \;\&\; \sum_{i=4}^{5} m_i = \sum_{i=4}^{5} m'_i \;\&\; m_6 < m'_6$

e. $\sum_{i=0}^{7} m_i = \sum_{i=0}^{7} m'_i \;\&\; \sum_{i=0}^{3} m_i = \sum_{i=0}^{3} m'_i \;\&\; \sum_{i=4}^{5} m_i = \sum_{i=4}^{5} m'_i \;\&\; m_6 = m'_6 \;\&\; m_4 < m'_4$

f. and so on.

⇨ Algorithm: determine 8 integers A,B,C,D,E,F,G,H as follows:

1. R0=S0+2*S1+22*S2+…+2K-1*SK-1: the R0-th element

2. Find the largest integer A for which $Z_8(\mathrm{A}) \le \mathrm{R}_0$

$\rightarrow$ total cost $\sum_{i=0}^{n} m_i = \mathrm{A}$

3. Determine the largest integer B such that $\mathrm{R}_1 \ge 0$, where

$$\mathrm{R}_1 = \mathrm{R}_0 - Z_8(\mathrm{A}) \qquad \text{if B=0}$$

$$= \mathrm{R}_0 - Z_8(\mathrm{A}) - \sum_{p=0}^{\mathrm{B}-1} g_4(p)\, g_4(\mathrm{A}-p) \qquad \text{if B>0}$$

$\rightarrow$ the left half of cost $\sum_{i=0}^{3} m_i = \mathrm{B}$

4. Determine the integers:

$R_2 = R_1$ modulo $g_4(B)$, where $0 \le R_2 \le g_4(B)-1$

$R_3 = (R_1 - R_2) / g_4(B)$

→ R_1: the R_1-th element in the set of $(\sum_{i=0}^{3} m_i = B \ \& \ \sum_{i=4}^{7} m_i = A-B)$

R_2: the R_2-th element in the set of 4-vectors (m_0, m_1, m_2, m_3)

satisfying $\sum_{i=0}^{3} m_i = B$

R_3: the R_3-th element in the set of 4-vectors (m_4, m_5, m_6, m_7)

satisfying $\sum_{i=4}^{7} m_i = A-B$

5. a. Determine the largest integer C such that $R_4 \geq 0$, where

$$R_4 = R_2 \qquad \text{if } C=0$$

$$= R_2 - \sum_{p=0}^{C-1} g_2(p) g_2(B-p) \qquad \text{if } C>0$$

b. Determine the largest integer D such that $R_5 \geq 0$,where

$$R_5 = R_3 \qquad \text{if } D=0$$

$$= R_3 - \sum_{p=0}^{D-1} g_2(p) g_2(A-B-p) \qquad \text{if } D>0$$

$\rightarrow \quad m_0+m_1 = C \;\&\; m_4+m_5 = D$

R_4: the R_4-th element in the set of 4-vectors (m_0,m_1,m_2,m_3)

satisfying $m_0+m_1 = C \;\&\; m_2+m_3 = B-C$

R_5: the R_3-th element in the set of 4-vectors (m_4,m_5,m_6,m_7)
satisfying $m_4+m_5 = D \;\&\; m_6+m_7 = A-B-D$

cost

6. a. Determine the integers:

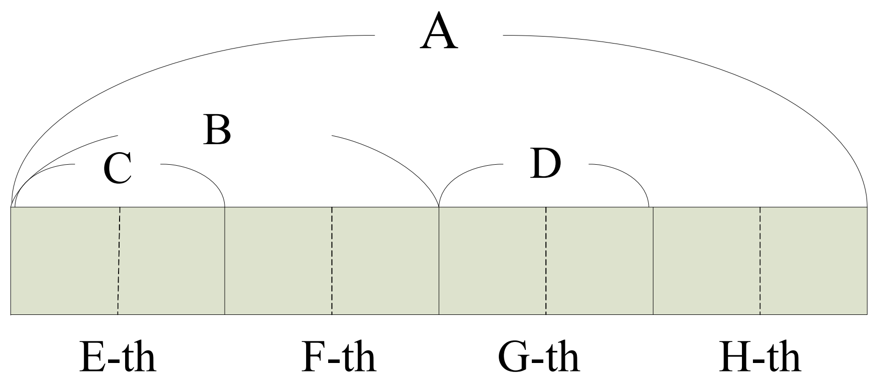
$E = R_4$ modulo $g_2(C)$, where

$F = (R_4 - E) / g_2(C)$

b. Determine the integers:

$G = R_5$ modulo $g_2(C)$, where

$H = (R_5 - G) / g_2(D)$



→ E: the E-th element in the set of 2-vectors (m_0,m_1) satisfying

$m_0+m_1=C$

F: the F-th element in the set of 2-vectors (m_2,m_3) satisfying

$m_2+m_3=B-C$

G: the G-th element in the set of 2-vectors (m_4,m_5) satisfying

$m_4+m_5=D$

H: the H-th element in the set of 2-vectors (m_6,m_7) satisfying

$m_6+m_7=A-B-D$

(m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7) are determined from the integers A,B,C,D,E,F,G,H as follow:

$$\begin{cases} \text{if C} < M\text{, then } m_0 = \text{E} \ \& \ m_1 = \text{C} - m_0 \\ \text{if C} \geq M\text{, then } m_0 = M - 1 - \text{E} \ \& \ m_1 = \text{C} - m_0 \end{cases}$$

$$\begin{cases} \text{if B-C} < M\text{, then } m_2 = \text{F} \ \& \ m_3 = \text{B} - \text{C} - m_2 \\ \text{if B-C} \geq M\text{, then } m_2 = M - 1 - \text{F} \ \& \ m_3 = \text{B} - \text{C} - m_2 \end{cases}$$

$$\begin{cases} \text{if D} < M\text{, then } m_4 = \text{G} \ \& \ m_5 = \text{D} - m_4 \\ \text{if D} \geq M\text{, then } m_4 = M - 1 - \text{G} \ \& \ m_5 = \text{D} - m_4 \end{cases}$$

$$\begin{cases} \text{if A-B-D} < M\text{, then } m_6 = \text{H} \ \& \ m_7 = \text{A} - \text{B} - \text{D} - m_6 \\ \text{if A-B-D} \geq M\text{, then } m_6 = M - 1 - \text{H} \ \& \ m_7 = \text{A} - \text{B} - \text{D} - m_6 \end{cases}$$

⇨ Two schemes used in shell mapper:

1. given R-th element in cost C
$\rightarrow$ get the cost of a left half C_1 & R'-th element in (C_1+C_2 = C)

2. given R'-th element in (C_1+C_2)
$\rightarrow$ get R_1-th in C_1 & R_2-th in C_2

- Shell Mapper:

1. given $(R_0-Z_8(A))$-th in A

   $\rightarrow$ get B & R_1-th in (B+(A-B))

2. given R_1-th in (B+(A-B))

   $\rightarrow$ get R_2-th in B & R_3-th in (A-B)

3. given R_2-th in B

   $\rightarrow$ get C & R_4-th in (C+(B-C))

   given R_3-th in (A-B)

   $\rightarrow$ get D & R_5-th in (D+(A-B-D))

4. given R_4-th in (C+(B-C))

   $\rightarrow$ get E-th in C & F-th in (B-C)

   given R_5-th in (D+(A-B-D))

   $\rightarrow$ get G-th in D & H-th in (A-B-D)

For example:

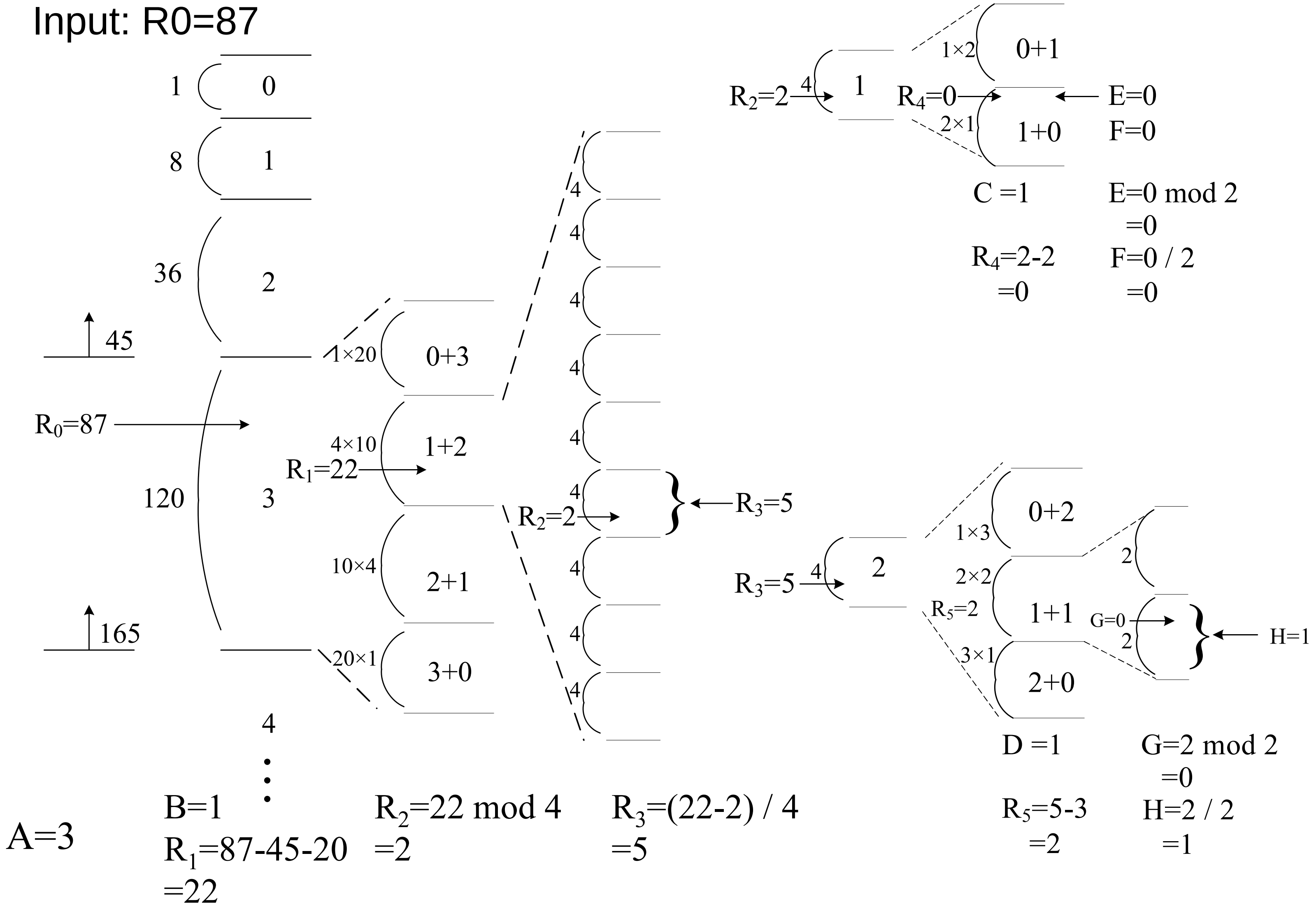
Consider M=4

then

$$
\begin{array}{llll}
g_2(0)=1 & g_4(0)=1 & g_8(0)=1 & Z_8(1)=1 \\
g_2(1)=2 & g_4(1)=4 & g_8(1)=8 & Z_8(2)=9 \\
g_2(2)=3 & g_4(2)=10 & g_8(2)=36 & Z_8(3)=45 \\
g_2(3)=4 & g_4(3)=20 & g_8(3)=120 & Z_8(4)=165 \\
g_2(4)=3 & \vdots \quad \vdots & \vdots \quad \vdots & \vdots \quad \vdots \\
g_2(5)=2 & \vdots \quad \vdots & \vdots \quad \vdots & \vdots \quad \vdots \\
g_2(6)=1 & \vdots \quad \vdots & \vdots \quad \vdots & \vdots \quad \vdots
\end{array}
$$

Input: $R_0=87$

Input: R0=87


1
0
8
1
36
2
45
R_0=87
120
3
165
4
1×20
0+3
4×10
1+2
R_1=22
10×4
2+1
20×1
3+0
4
R_2=2
R_3=5
A=3
B=1
R_1=87-45-20
=22
R_2=22 mod 4
=2
R_3=(22-2) / 4
=5
R_2=2
1
R_4=0
1×2
0+1
2×1
1+0
E=0
F=0
C =1
E=0 mod 2
=0
R_4=2-2
=0
F=0 / 2
=0
R_3=5
2
1×3
0+2
2×2
1+1
R_5=2
3×1
2+0
G=0
2
H=1
D =1
G=2 mod 2
=0
R_5=5-3
=2
H=2 / 2
=1

⇨ Decoder for shell mapper

input: $(m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7)$

output: R_0

1. According to $(m_0, m_1, ..., m_7)$, the values of A,B,C,D,E,F,G and H can be obtained.

2. $R_0 = Z_8(A) + \sum_{p=0}^{B-1} g_4(p) g_4(A-p) + R_3 g_4(B) + R_2$, where

“total cost $< A$”

“total cost $= A$ cost of left half $< B$”

“total cost $= A$ cost of left halt $= B$ index of right half $< R_3$”

“total cost $= A$ cost of left half $= B$ index of right half $= R_3$ index of left half $< R_2$”

$$R_2 = \sum_{p=0}^{C-1} g_2(p) g_2(B-p) + F \cdot g_2(C) + E$$

$$R_3 = \sum_{p=0}^{D-1} g_2(p) g_2(A-B-p) + H \cdot g_2(D) + G$$

⇨ Reference:

R. Laroia, N. Farvardin and S.A. Tretter," On optimal shaping of multidimensional constellations, " *IEEE Tran. Inform. Theory*, pp.1044-1056, July 1994 (algorithm 2 in p.1048,1049)

There are differences between V.34 and paper. When the cost of the left half is the same, paper compares the index of the left half while V.34 compares the index of the right half.

$$\left\{\begin{array}{ll} \text{V.34}: & R_1 = R_3 \cdot g_4(\text{B}) + R_2 \\ \text{paper}: & R_1 = R_2 \cdot g_4(\text{A-B}) + R_3 \end{array}\right\}$$

$$\left\{\begin{array}{ll} \text{V.34}: & R_4 = \text{F} \cdot g_2(\text{C}) + \text{E} \\ \text{paper}: & R_4 = \text{E} \cdot g_2(\text{B-C}) + \text{F} \end{array}\right\}$$

$$\left\{\begin{array}{ll} \text{V.34}: & R_1 = \text{H} \cdot g_2(\text{D}) + \text{G} \\ \text{paper}: & R_1 = \text{G} \cdot g_2(\text{A-B-D}) + \text{H} \end{array}\right\}$$

ISI Coder

- All previous voice modems, such as V.32 (14.4K bits/sec), use adaptive linear equalizers in the receiver to combat ISI. In these modems, the transmission band is confined to a sweet spot of 2400 Hz or less in which it is known a priori that channel attenuation will not be too severe.

- In contrast, in V.34 every effort is made to make use of all available bandwidth, including frequencies near the band edges where there can be attenuation of as much as 10-20 db. In such a situation, it is well known that linear equalizers cause significant noise enhancement.

- It is also well known that a decision-feedback equalizer (DFE) is well suited to such channels. However, it is not possible to combine coding with a DFE straightforwardly. The solution to this problem involves precoding: putting the feedback part of the DFE into the transmitter.

- The simplest form of precoding is to pre-emphasize the signal before transmission. This, however, boosts the signal power and results in suboptimal use of power for channels with a constraint on the average transmit power. Tomlinson-Harashima (TH) precoding is a nonlinear technique that also pre-equalizes the signal before transmission. For uniformly distributed inputs, the TH precoder does not boost the transmit power. While the TH precoding is simple to implement and can be used with coded modulation to realize coding gains, it does not allow the realization of any shaping gain, as the precoder output tends to be uniformly distributed in a cube.

TH precoding

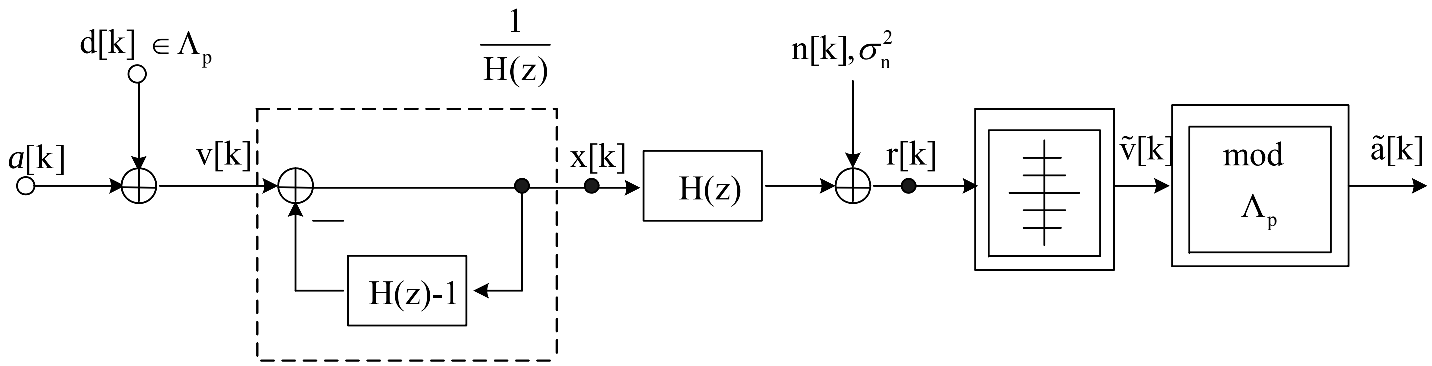


Fig.2. Transmission scheme using THP.

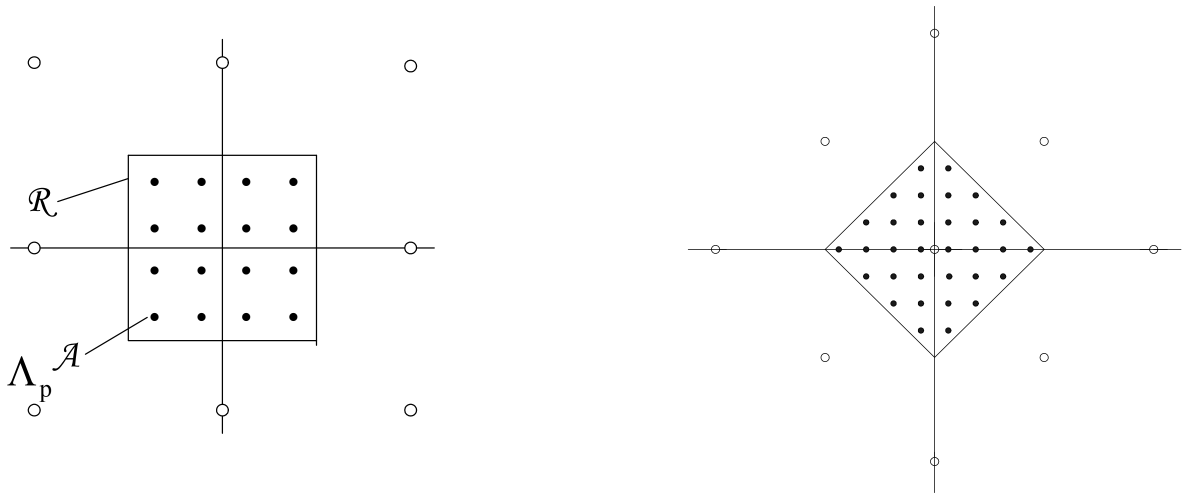


Fig.3. Two-dimensional boundary regions of signal constellation for THP

- M-point QAM signal set $\mathrm{A} = \{\pm 1, \ldots, \pm(\sqrt{M} - 1)\} \times \{\pm 1, \ldots, \pm(\sqrt{M} - 1)\}$,

  M is a even square number.

  K: number of tap.

  $\{\mathrm{a}[\mathrm{k}]\}$: data sequence. $\mathrm{a}[\mathrm{k}] \in \mathcal{A}$

  $\{\mathrm{d}[\mathrm{k}]\}$: predecoding sequence. $\mathrm{d}[\mathrm{k}] \in \Lambda_{\mathrm{p}} = 2\sqrt{\mathrm{M}}\,\mathbb{Z}^2$

  $\{\mathrm{x}[\mathrm{k}]\}: \mathrm{x}[\mathrm{k}] = \mathrm{a}[\mathrm{k}] + \mathrm{d}[\mathrm{k}] - \sum_{i=1}^{\mathrm{K}} \mathrm{h}[\mathrm{i}]\mathrm{x}[\mathrm{k}-\mathrm{i}] = \mathrm{v}[\mathrm{k}] - \mathrm{f}[\mathrm{k}]$

  $$\{\mathrm{r}[\mathrm{k}]\}: \mathrm{r}[k] = \sum_{i=0}^{\mathrm{K}} \mathrm{h}[i]\mathrm{x}[\mathrm{k}-i] + \mathrm{n}[\mathrm{k}] = x[k] + \sum_{i=1}^{\mathrm{K}} \mathrm{h}[i]\mathrm{x}[\mathrm{k}-i] + \mathrm{n}[\mathrm{k}]$$
  $$= \mathrm{x}[\mathrm{k}] + \mathrm{f}[\mathrm{k}] + \mathrm{n}[\mathrm{k}] = \mathrm{v}[\mathrm{k}] + \mathrm{n}[\mathrm{k}]$$

  The signal set $\mathcal{A}$ is the intersection of $2\mathbb{Z}^2 + (1,1)$ and the Voronoi region $\mathcal{R}$ of Λ_{p}.

- For $\log_2 M$ odd, $\mathrm{d}[\mathrm{k}] \in 2\sqrt{M}\,\mathrm{R}\mathbb{Z}^2$, where $\mathrm{R} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$ is the rotation operator .

LTF precoder

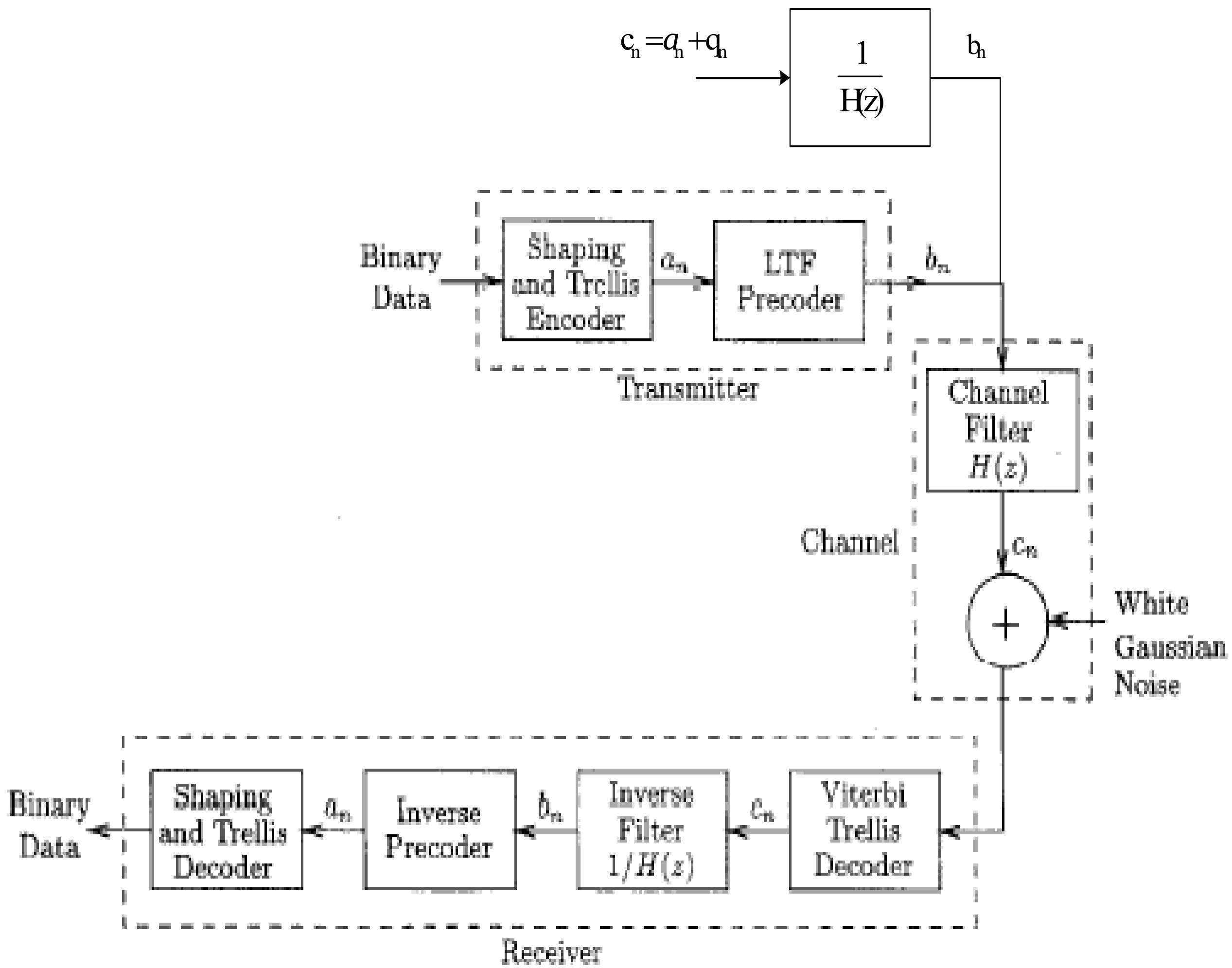


Fig.1. Block diagram of a communication system using LTF predecoder.

a_n: the output of shaping and trellis encoder $\{a_n\}$ is a code sequence.

b_n : $a_n - m_n$: the transmitted signal since $\{a_n\}$ has shaping gain, the value of $|m_n|$ should be as small as possible.

$c_n = a_n + q_n$

We should let $\{c_n\}$ be a code sequence.

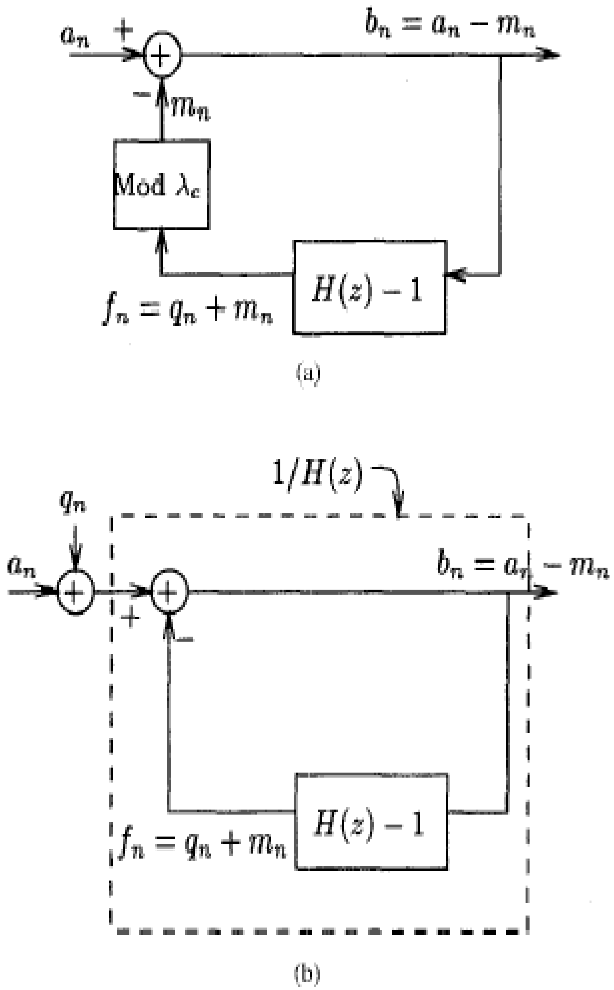
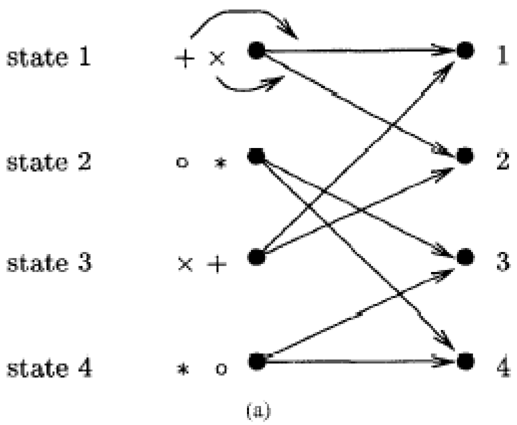


Fig.3. (a)The LTF predecoder and (b) equivalent linear circuit

Take Fig.2. as an example.



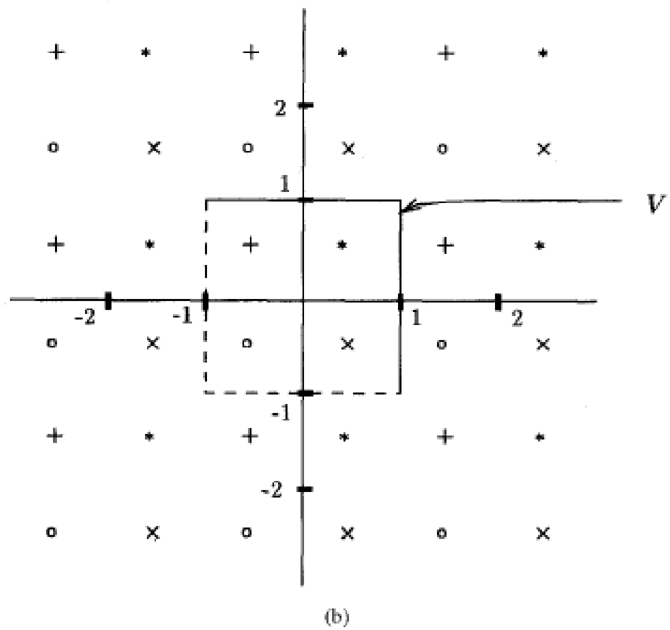


Fig. 2. (a) Finite-state machine used to generate the 4-state 2D trellis code. (b) Four-way coset partitioning used to generate the 4-state 2D trellis code.

Assume that $\{a_n\}$ is a sequence from an Ungerboeck 4-state 2D trellis code with redundant 1 bits/2D generated by the finite-state machine of Fig.2(a) and based on a four-way partition of the lattice translate $\lambda = \mathbb{Z}^2 + (\frac{1}{2}, \frac{1}{2})$ into cosets marked by "+", "*", "∘", "×" in Fig.2.(b)

$\lambda_c = 2\mathbb{Z}^2$: coset lattice

$f_n = q_n + m_n$

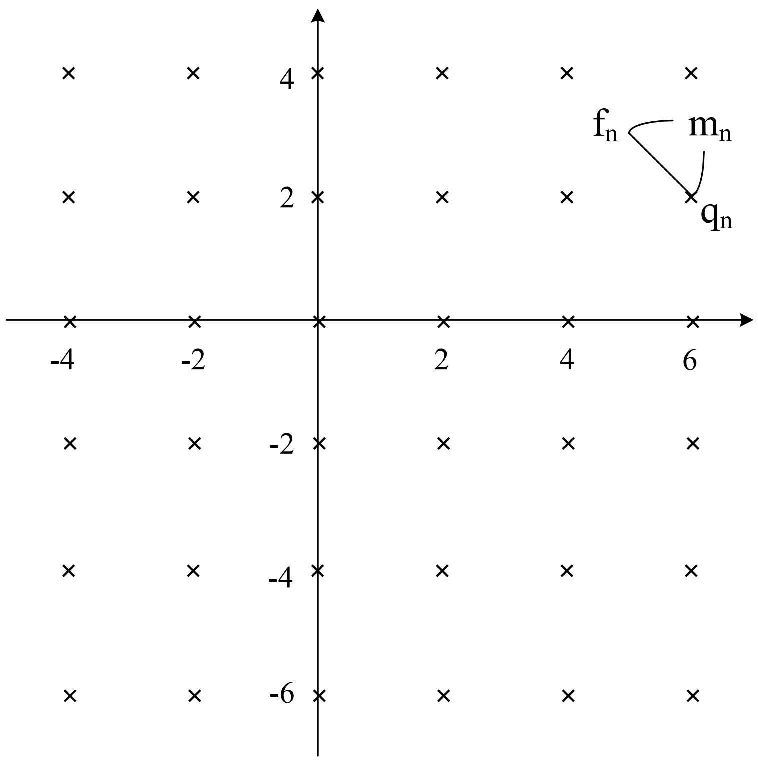
$q_n \in \lambda_c$

if $a_n \in$'x', $c_n \in$'x'

if $a_n \in$'$\circ$', $c_n \in$'$\circ$'

if $a_n \in$'$*$', $c_n \in$'$*$'

Hence, even $\{c_n\}$ is different from $\{a_n\}$, $\{c_n\}$ is a code sequence!



- The inverse precoder recovers a_n simply by quantizing $b_{n\text{-}}$ to the closest point in the same coset as c_n.

- Notably, m_n is always contained inside the Voronoi region V of the coset lattice . In this example, $|m_n| \leq \sqrt{2}$

- When the transmission rate is high, the precoding loss ($|m_n|$) could be a negligible fraction of the total transmit energy. At low rates, the loss could be significant.

- Notice that the precoding loss of the LTF precoder depends on the trellis code used to realize the coding gain. For the trellis code considered above, this loss is four times the loss due to TH precoding. When a more powerful trellis code is used to realize a higher coding gain, the corresponding coset lattices have bigger Voronoi regions resulting in higher losses. For example, using a trellis code based on an 8-way or a 16-way partition would has a precoding loss which is eight and sixteen times the TH precoding loss, respectively.

➤ISI coder

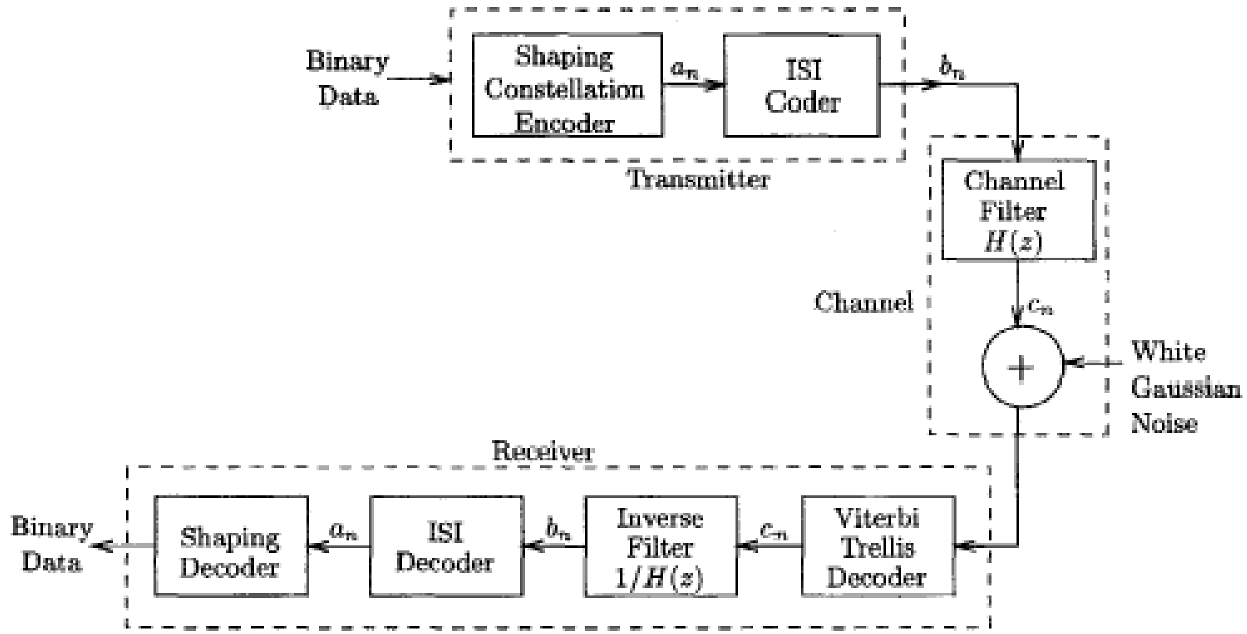


Fig. 5. Block diagram of a communication system with ISI coder/decoder.

ISI coder is the combination of trellis encoder and precoder. To understand the ISI coder, we first study an important property of all Ungerboeck-type trellis codes as follows. Consider any 2D trellis code with redundancy 1 bit/2D. In the first level of partitioning, the lattice translate $\mathbb{Z}^2+(\frac{1}{2},\frac{1}{2})$ is partitioned into two subsets A and B. If you consider the trellis diagram of such a code, then all trellis states in the diagram can be divided into two types: transmitting A or transmitting B.

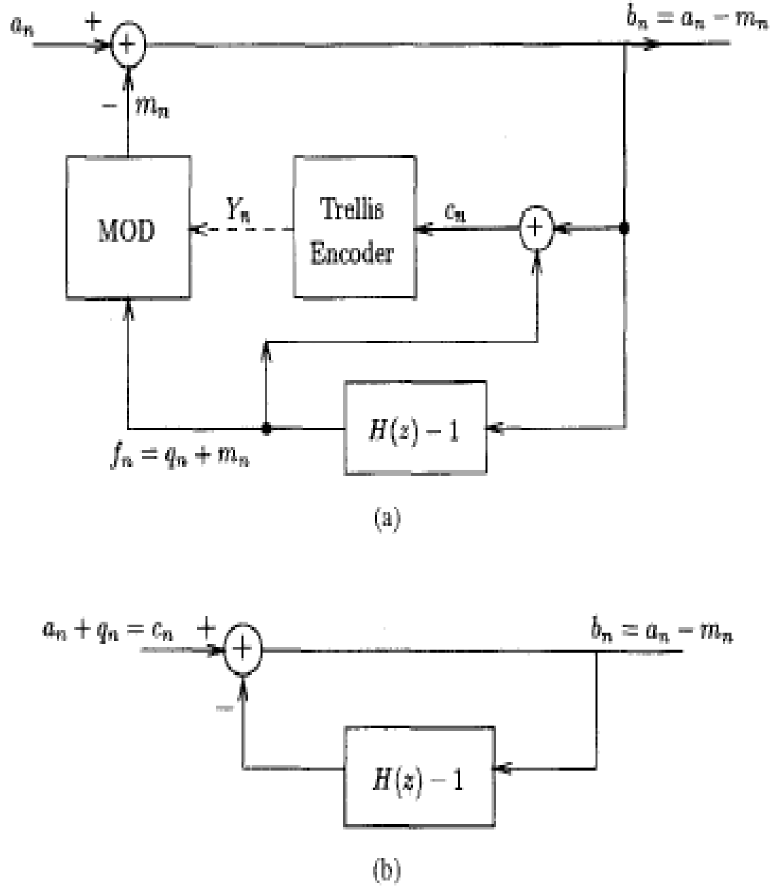
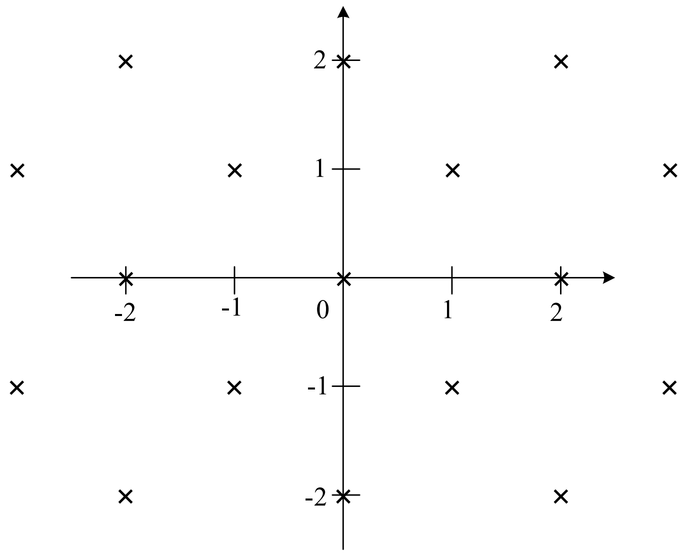


Fig. 6. (a) The ISI coder. (b) Equivalent linear circuit.

- $a_n \in \mathrm{A}$ At time n, according to c_{n-1} and previous state S_{n-1}, the new state S_n can be obtained. According to S_n, transmitting $\mathrm{A}(c_n \in \mathrm{A})$ or $\mathrm{B}(c_n \in \mathrm{B})$ can be judged.



"R$\mathbb{Z}^2$"

- If $c_n \in \mathrm{A}$, let $Y_n=0$; if $c_n \in \mathrm{B}$, let $Y_n=1$.

- If $Y_n=0$, the ISI coder performs the modulo $R\mathbb{Z}^2$ function on the input f_n.

$$c_n = a_n + q_n \in \mathrm{A}!$$

- If $Y_n=1$, the ISI coder performs the modulo $R\mathbb{Z}^2+(1,0)$ function.

$$c_n = a_n + q_n \in \mathrm{B}!$$

- Note that a_n can be recovered by the ISI decoder which simply quantizes b_n to the nearest point in the subset A.

- For the ISI coder, m_n is always contained inside the Voronoi region V' of the $R\mathbb{Z}^2$ lattice. ($|m_n| \leqq 1$)

⇨ Modify ISI coder

- A rotated by $90^\circ \rightarrow$ B

  B rotated by $90^\circ \rightarrow$ A

- MOD $\mathbb{Z}^2$

  if $q_n \in R\mathbb{Z}^2, W_n=0$

  if $q_n \in R\mathbb{Z}^2+(1,0), W_n=1$

$$\begin{cases} W_n = 0 \leftrightarrow c_n, a'_n & \text{the same subset} \\ W_n = 1 \leftrightarrow c_n, a'_n & \text{different subset} \end{cases}$$

- The output of trellis encoder Y_n is the same as the Y_n in the ISI encoder.

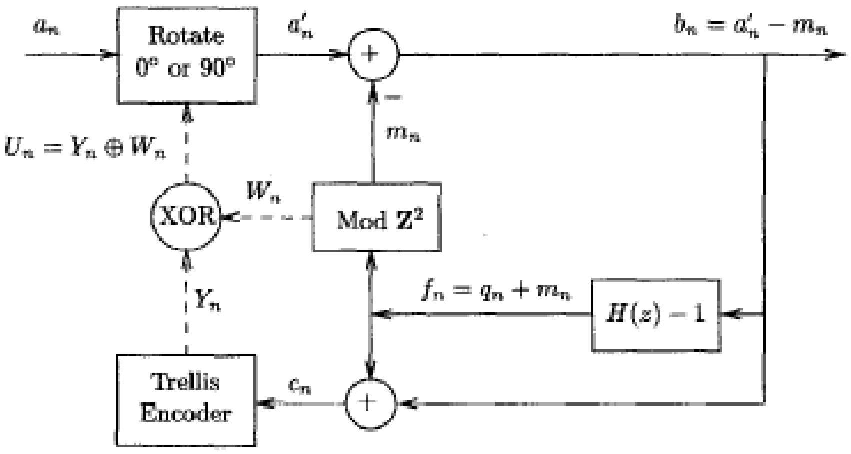
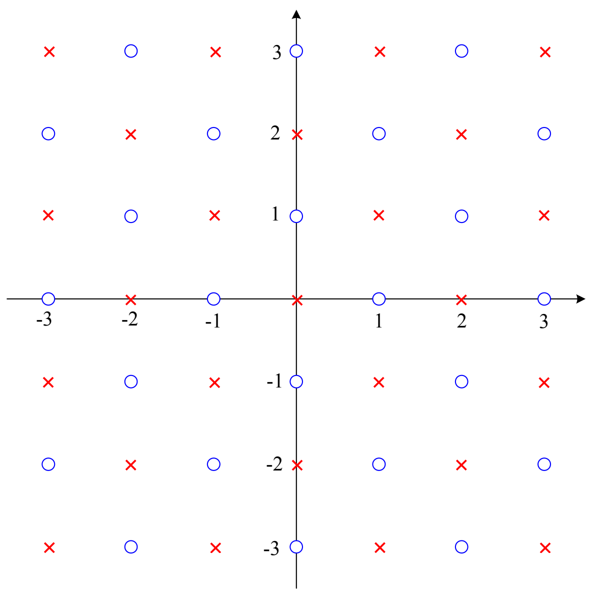


Fig. 7. The modified ISI coder.



⇨ Modified ISI coder

- if Y_n=0, we need $c_n \in$ A
  if Y_n=1, we need $c_n \in$ B

- if Y_n=0, W_n=0, let $a'_n \in$A
  if Y_n=0, W_n=1, let $a'_n \in$B
  if Y_n=1, W_n=0, let $a'_n \in$B
  if Y_n=1, W_n=0, let $a'_n \in$A
  $\therefore$ Let $U_n = Y_n \oplus W_n$

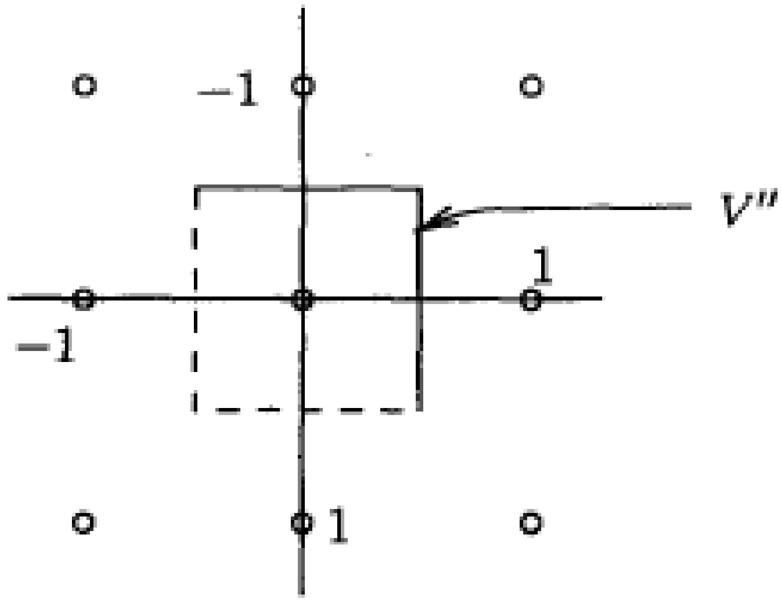


Fig. 8. Voronoi region V'' of the Z^2 lattice.

- if U_n=0, rotate 0° (do nothing) $\rightarrow a_n = a'_n \in$ A

- if U_n=1, rotate 90° (do nothing) $\rightarrow a'_n \in$B

- Since m_n is contained in the Voronoi region V'' of the $\mathbb{Z}^2$ lattice, in the "ISI decoder", a'_n can be obtained simply by quantizing b_n to the closest point in $\mathbb{Z}^2 + (\frac{1}{2}, \frac{1}{2})$. If $a'_n \in A$, then $a_n = a'_n$; else if $a'_n \in B$, then a_n is obtained by rotating a'_n by -90°.

- The precoding loss of the modified ISI coder is now equal to the loss due to TH precoding. The modified ISI coder is believed to result in the smallest possible precoding loss of any precoding scheme and hence completes the evolution of the precoder.

- Generalization to higher dimensional codes

  We consider here 2k-dimensional codes with redundancy $\frac{1}{k}$ bits per 2D. The first level of lattice partitioning divides the (translate of $\mathbb{Z}^{2k}$) lattice into two subsets A and B. The input to the (modified) ISI coder consists of a sequence of uncoded symbols in A. The first k-1 QAM symbols of a trellis-code symbol use a fixed modulo $\mathbb{Z}^2$ operation in the (modified) ISI coder. Depending on the k-1 channel output QAM symbols produced by these, for the ISI coder, the k-th QAM symbol uses one of mod $R\mathbb{Z}^2$ or mod $R\mathbb{Z}^2 + (1.0)$; for the modified ISI coder, the k-th symbol is rotated by 0° or 90°, to ensure that the channel output is in the desired subset (A or B).

- The V.34 standard provides for a simple three-tap precoding filter (representing the feedback filter in a DFE) whose coefficients are determined during initial training by the receiver and sent to the transmitter. The feedforward filter in the DFE is realized as an adaptive linear equalizer in the receiver and continues to adapt during data transmission.

References:

1. R.Laroia, ”Coding for intersymbol interference channels–combined coding and precoding,” *IEEE Tran. Inform. Theory*, pp.1053-1601, July 1996.

2. G.D. Forney, Jr. et.al. ”The V.34 high-speed modem standard,” *IEEE Communication Magazine*, pp.28-33, Dec 1996