Information theory and its application to optical communication

Citation for published version (APA):

DOI:
10.1364/SPPCOM.2017.SpTu3E.1

Document status and date:
Published: 01/01/2017

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 31. Oct. 2023
Information Theory and its Application to Optical Communication

Frans M.J. Willems

ICT Lab, SPS Group
Department of Electrical Engineering
Eindhoven University of Technology

SPPCom, Advanced Photonics 2017, July 25, New Orleans
Optical communication links carry most of the data that is transmitted around the world. Home connections are being replaced by optical links.

We like to achieve the highest possible data rates for the smallest cost. Replacing links should be delayed as long as possible.

Therefore advance transmission protocols (equalisation, modulation, coding) are required.

INFORMATION THEORY tells us what the ultimate performances are (e.g. capacity), and what the techniques are that achieve ultimate performance.

Wireless communication is characterized by major developments (coding, mimo, cooperative communications, etc.), often boosted by information theoretical methods.

Optical Communication is going through a similar innovation cycle now. Information theory can also be useful here.
Claude Shannon (1916-2001)

- **1948**: “A Mathematical Theory of Communication,” Bell Syst. Tech. J.: Shannon combined the noise power spectral density $N_0/2$, the channel bandwidth $W$, and the transmit power $P$, into a single parameter $C$, which he called the **channel capacity**. More precisely

$$C = W \log_2(1 + \frac{P}{N_0W})$$

represents the maximum number of bits that can be sent per second **reliably** from transmitter to receiver. **Codes** can be used to achieve capacity.

- **1938**: Shannon also applied **Boole’s algebra to switching circuits** (MSc thesis, MIT).

- **WW2**: Shannon developed **cryptographic equipment** for transoceanic conferences (Roosevelt-Churchill). His ideas can be found in **“Communication Theory of Secrecy Systems”**, a confidential report from 1945, published in 1949.

- **1949**: Shannon introduced the **sampling theorem** to the engineering community.
Let $X$ be a discrete random variable with alphabet $\mathcal{X}$ and probability mass function $p(x) = \Pr\{X = x\}$ for $x \in \mathcal{X}$.

**Definition**

The **entropy** $H(X)$ of discrete random variable $X$ is defined as

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} \text{ [bit]}.$$ 

**Example**

Binary random variable $X$ with alphabet $\mathcal{X} = \{0, 1\}$. Let

$$X = \begin{cases} 0 \text{ with probability } 1 - p, \\ 1 \text{ with probability } p. \end{cases}$$

Then the entropy of $X$ is

$$H(X) = h(p),$$

where

$$h(p) \triangleq (1 - p) \log_2 \frac{1}{1 - p} + p \log_2 \frac{1}{p} \text{ [bit]}.$$
Observe that $h(p) = h(1 - p)$ and that e.g. $h(0.1) = 0.4690$.

- Think of $H(X)$ as the **uncertainty** in $X$.
- It can be shown that $0 \leq H(X) \leq \log_2 |X|$.
Conditional Entropy

Let $X$ and $y$ be discrete random variables with alphabets $\mathcal{X}$ and $\mathcal{Y}$ respectively and joint probability mass function $p(x, y) = \Pr\{X = x, Y = y\}$ for $x \in \mathcal{X}$ and $x \in \mathcal{Y}$.

Note that $p(y) = \sum_{x \in \mathcal{X}} p(x, y)$ and $p(x|y) = p(x, y)/p(y)$.

**Definition**

The **conditional entropy** $H(X|Y)$ of discrete random variable $X$ given $Y$ is defined as

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log_2 \frac{1}{p(x|y)} \text{ [bit]}.$$ 

- Think of $H(X|Y)$ as the **uncertainty** in $X$ when $Y$ is given.
- It can be shown that $0 \leq H(X|Y) \leq H(X)$. Conditioning can only reduce entropy.
- Note also that

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p(y) H(X|Y = y),$$

where

$$H(X|Y = y) = \sum_{x \in \mathcal{X}} p(x|y) \log_2 \frac{1}{p(x|y)}.$$ 

Conditional entropy $H(X|Y)$ is the expected value of entropies $H(X|Y = y)$ w.r.t. $p(y)$. 


Example: Binary Symmetric Channel (BSC)

Example

- **Transition probabilities** \( p(Y = 1|X = 0) = p(Y = 0|X = 1) = p. \)

\[
\begin{align*}
X = 1 & \quad 1 - p & \quad Y = 1 \\
\quad & \quad p & \quad X = 0 \\
& \quad 1 - p & \quad Y = 0
\end{align*}
\]

- For **uniform** \( p(X = 0) = p(X = 1) = 1/2 \) we obtain that the **entropy**

\[ H(X) = h(1/2) = 1. \]

- Moreover:

\[
\begin{align*}
p(Y = 1) &= p(X = 0) \cdot p + p(X = 1) \cdot (1 - p) = 1/2, \\
p(X = 1|Y = 0) &= \frac{p(X = 1, Y = 0)}{p(Y = 0)} = \frac{p(X = 1) \cdot p}{p(Y = 0)} = p, \\
p(X = 1|Y = 1) &= \frac{p(X = 1, Y = 1)}{p(Y = 1)} = \frac{p(X = 1) \cdot (1 - p)}{p(Y = 1)} = 1 - p,
\end{align*}
\]

and the **conditional entropy**

\[ H(X|Y) = p(Y = 0)h(p) + p(Y = 1)h(1 - p) = h(p). \]
Mutual Information

**Definition**

The **mutual information** $I(X; Y)$ between the discrete random variable $X$ and $Y$ is defined as

$$I(X; Y) = H(X) - H(X|Y) \ [\text{bit}].$$

- Think of $I(X; Y)$ as the **decrease in uncertainty** about $X$ when $Y$ is released. Equivalently it is the information that $Y$ contains about $X$.
- It can be shown that always $0 \leq I(X; Y) \leq H(X)$.
- $I(X; Y)$ is also the **decrease in uncertainty** about $Y$ when $X$ is released.

**Example**

- Binary symmetric channel (BSC) with transition probability $p$:

  $$
  \begin{align*}
  X = 1 & \quad \xrightarrow{1-p} \quad Y = 1 \\
  X = 0 & \quad \xrightarrow{p} \quad Y = 1 \\
  X = 1 & \quad \xrightarrow{p} \quad Y = 0 \\
  X = 0 & \quad \xrightarrow{1-p} \quad Y = 0
  \end{align*}
  $$

- For **uniform** $p(X = 0) = p(X = 1) = 1/2$, we obtain that

  $$I(X; Y) = H(X) - H(X|Y) = 1 - h(p).$$

  For $p = 0.1$ we obtain $I(X; Y) = 1 - 0.4690 = 0.5310 \ \text{bit}.$
Discrete Memoryless Channel

Definition

Channel input alphabet $\mathcal{X}$, channel output alphabet $\mathcal{Y}$. For each $x \in \mathcal{X}$ the transition probabilities $\Pr\{Y = y | X = x\}$ for $y \in \mathcal{Y}$ are denoted by $p(y|x)$, where $\sum_{y \in \mathcal{Y}} p(y|x) = 1$.

Example

$|\mathcal{X}|$ $|\mathcal{Y}|$

Here $\mathcal{X} \triangleq \{1, 2, \ldots, |\mathcal{X}|\}$ and $\mathcal{Y} \triangleq \{1, 2, \ldots, |\mathcal{Y}|\}$.
Channel Capacity, Definition

Observe that:

- the **channel input distribution** \( \{p(x), x \in \mathcal{X}\} \) determines the joint distribution
  
  \[ p(x, y) = p(x)p(y|x), \text{ for all } x \in \mathcal{X}, y \in \mathcal{Y}, \]

- and therefore the **mutual information**
  
  \[ I(X; Y) = H(X) - H(X|Y). \]

- The maximum value of \( I(X; Y) \) is called the **channel capacity** \( C \). Hence

**Definition**

\[ C_{\text{DMC}} = \max_{p(x)} I(X; Y) \quad [\text{bit/channel use}]. \]

**Example**

For a **BSC with crossover probability** \( p \) a **uniform input** \( p(X = 0) = p(X = 1) = 1/2 \) achieves maximum mutual information \( 1 - h(p) \). The **channel capacity** of the BSC is therefore

\[ C_{\text{BSC}}(p) = 1 - h(p). \]
Capacity of the BSC

![Graph showing the capacity of the BSC as a function of error probability.](image-url)
Message index $m$ assumes values in $\{1, 2, \ldots, |M|\}$, uniformly. There is a codeword $x_1x_2 \cdots x_N$ of length $X$ for each message index $w$. The codeword is transmitted over the DMC with transition probabilities \( \{p(y|x), x \in \mathcal{X}, y \in \mathcal{Y}\} \). The receiver makes an estimate $\hat{m}$ of the transmitted index $m$ from the channel output $y_1y_2, \cdots y_N$.

Transmission rate

$$\frac{1}{N} \log_2 |M|.$$ 

Error probability

$$P_e = \Pr\{\hat{M} \neq M\}.$$ 

Theorem (Shannon, 1948)

Rate $R$ is said to be achievable if, for any $\varepsilon > 0$, for all large enough $N$, there exist codes with operational rate $\frac{1}{N} \log_2 |M| \geq R - \varepsilon$ and error probability $P_e \leq \varepsilon$.

Rates $R$ not exceeding $C$ are achievable. Rates $R$ larger than $C$ are not achievable.
Channel Capacity, Proof

1. **RANDOM CODING ARGUMENT**
   Shannon showed that if the $|M|$ codewords are generated at random according to the capacity-achieving input distribution $\{p(x), x \in \mathcal{X}\}$ the error probability averaged over the ensemble of codes
   
   $$
   \overline{P_e} = \sum_{\text{all codes}} P(\text{code})P_e(\text{code})
   $$

   can be made arbitrarily small for $\frac{1}{N} \log_2 |M| = C - \varepsilon$ and $N \to \infty$, for any $\varepsilon > 0$. There exist codes with arbitrarily small $P_e$ therefore.

2. **CONVERSE**
   Using Fano’s inequality\(^1\) it can be shown that for $\frac{1}{N} \log_2 |M| > C + \delta$ the error probability $P_e$ can not be made arbitrarily small for large $N$, for any $\delta > 0$.

3. **LINEAR CODES** Hamming introduced Hamming codes. Elias [1955] demonstrated that for the BSC also parity check codes achieve capacity.

---

\(^1\) $H(W|\hat{W}) \leq h(P_e) + P_e \log_2(|M| - 1)$
A **linear code** is defined by its **generator-matrix** $G$ or by the corresponding **parity-check matrix** $H$.

Codewords are **linear combinations of the rows** of generator matrix $G$. There are $2^K$ codewords. With the parity-check matrix $H$ it can be **checked** whether $x = (x_1, x_2, \ldots, x_N)$ is a codeword or not.

$N$ is the length of the codewords and $K$ the number of rows in $G$. Now $N - K$ is the number of rows in $H$, which is the number of parity-check equations.

### Example

Hamming code, $N = 7$, $K = 4$, can correct a **single error**.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

If $xH^T = 0$ then $x$ must be a codeword in our code. For non-codewords the so-called **syndrome** of $x$

$$s = xH^T \neq 0.$$

This syndrome $s = (s_1, s_2, s_3)$ can assume eight different values, $(0, 0, 0)$ when $x$ is a codeword, $(1, 1, 1)$, when there is an error at position 1, $(1, 1, 0)$, when there is an error at position 2, etc.
Additive Gaussian Noise (AGN) Channel, Definition, Capacity

**Definition (AGN channel)**

- Input variable $X$ satisfies $E[X^2] \leq E_x$. Here $E_x$ is input symbol energy.
- Noise variable $N$ is zero-mean Gaussian, variance $\sigma^2$, hence
  
  \[
  p(n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{n^2}{2\sigma^2}\right)
  \]

- Output $Y = X + N$.

**MODEL** for transmission links where thermal noise is dominant.

For the capacity of the AGN channel we find that

\[
C_{AGN} = \frac{1}{2} \log_2 \left(1 + \frac{E_x}{\sigma^2}\right) \quad \text{[bit/channel use]}
\]
AGN Channel, Capacity Derivation

- **Derivation**

\[
C_{AGN} \triangleq \max_{X : \mathbb{E}[X^2] \leq E_x} I(X; Y) = \max_{X : \mathbb{E}[X^2] \leq E_x} h(Y) - h(Y|X) = \max_{X : \mathbb{E}[X^2] \leq E_x} h(Y) - h(N) \leq \max_{X : \mathbb{E}[X^2] \leq E_x} \frac{1}{2} \log_2 \frac{2\pi e (E_x + \sigma^2)}{2\pi e \sigma^2} - \frac{1}{2} \log_2 2\pi e \sigma^2 = \frac{1}{2} \log_2 (1 + \frac{E_x}{\sigma^2}).
\]

Note that (a) is power constrained optimization, (b) splits \( I(X; Y) \) into differential entropies, (c) is based on upper bound on entropy given variance.

- Observe that **equality (capacity) is obtained only if \( X \) is Gaussian.**

- **Signal-to-noise ratio** definition:

\[
\text{SNR} \triangleq \frac{E_x}{\sigma^2}.
\]
AGN Channel, Capacity Plot

\[ C_{\text{AGN}} \text{ (bit/channel use)} \]

\[ E_{x}/\sigma^{2} \text{ (dB)} \]
Q1: What is the **minimal signal-to-noise ratio** $E_x/\sigma^2$ for rate $R$? From

$$R \leq C_{\text{AGN}} = \frac{1}{2} \log_2(1 + \frac{E_x}{\sigma^2})$$

we obtain that

$$\left(\frac{E_x}{\sigma^2}\right)_{\text{min}} = 2^{2R} - 1,$$

This is called **Shannon limit**.

Q2: What is the **minimal transmit energy** per transmitted bit?

$$\left(\frac{E_x}{R}\right)_{\text{min}} = \sigma^2 \cdot \frac{2^{2R} - 1}{R}.$$ 

Since

$$\lim_{R \downarrow 0} \frac{2^{2R} - 1}{R} = \lim_{R \downarrow 0} \frac{\exp(2R \ln 2) - 1}{R} = 2 \ln 2.$$

the **minimal energy per bit** is equal to $2\sigma^2 \ln 2$, achieved only if $R \downarrow 0$. 
Message index $m$ assumes values in $\{1, 2, \cdots, |M|\}$, uniformly.
- There is a waveform $s_m(t)$ for each message index $m$.
- The waveform is transmitted over the channel that adds white noise $n_w(t)$ to it. The channel output-waveform is

$$r(t) = s_m(t) + n_w(t).$$

The Gaussian stationary noise process $N_w(t)$ is zero-mean, hence $E[N_w(t)] = 0$ for all $t$, and its autocorrelation function

$$E[N_w(t)N_w(s)] = \frac{N_0}{2} \delta(t - s).$$

Transmission rate

$$\frac{1}{N} \log_2 |M|.$$

Error probability

$$P_e = \Pr\{\hat{W} \neq W\}.$$
If the (effective) time-duration of the signals is \( \Delta \) then the energy of signal \( s_m(t) \), \( m \in \{1, 2, \cdots, |M|\} \) should satisfy

\[
E_{sm} = \int_{-\infty}^{\infty} s_m^2(t) dt \leq P\Delta.
\]

This inequality is called the \textit{power constraint}, power is \( P \).

Moreover the signals \( s_m(t) \), \( m \in \{1, 2, \cdots, |M|\} \), should satisfy the \textit{bandwidth constraint}

\[
S_m(f) = \int_{-\infty}^{\infty} s_m(t) \exp(-j2\pi ft) dt = 0 \text{ for } |f| > W,
\]

where \( W \) is the bandwidth.
The Sinc-Pulse

The sinc-pulse

\[ p(t) = \frac{1}{\sqrt{T}} \frac{\sin(\pi t / T)}{\pi t / T} \]

has Fourier spectrum

\[ P(f) = \begin{cases} \sqrt{T} & \text{for } |f| < 1/(2T) \\ 0 & \text{for } |f| > 1/(2T). \end{cases} \]

Therefore this sinc-pulse satisfies the bandwidth constraint for \( T = \frac{1}{2W} \).

Example

The pulse \( p(t) \) and its spectrum \( P(f) \) for \( T = 1 \).
We can now transmit a sinc-pulse every $T$ seconds. If we give the pulse $p(t - kT)$ amplitude $x_k \in \mathcal{X}$, and add these scaled pulses, we get the waveform

$$s(t) = \sum_{k=0,K-1} x_k p(t - kT).$$

**Example**

Let $\mathcal{X} = \{-3, -1, +1, +3\}$ and take $K = 8$. Now let $x_1, \cdots, x_8 = (-3, +3, +3, +1, -3, -1, +3, -1)$ then the scaled pulses ($T = 1$) and their sum are:
Orthonormality of the Pulses

### Definition

The sinc-pulses $p(t - kT)$ for $k = 0, 1, \ldots, K - 1$ are **orthonormal**, i.e.

$$
\int_{-\infty}^{\infty} p(t - kT)p(t - k'T)dt = \begin{cases} 
1 & \text{for } k' = k, \\
0 & \text{for } k' \neq k.
\end{cases}
$$

For the **correlation** $y_k$ of the output waveform $r(t)$ with the pulse $p(t - kT)$ for $k = 0, 1, \ldots, K - 1$ we can write

$$
y_k = \int_{-\infty}^{\infty} r(t)p(t - kT)dt
= \int_{-\infty}^{\infty} \left( \sum_{k'} x_{k'}p(t - k'T) + n_w(t) \right) p(t - kT)dt = x_k + n_k
$$

with

$$
n_k = \int_{-\infty}^{\infty} n_w(t)p(t - kT)dt.
$$

Note that this correlation **yields the desired signal amplitude** $x_k$ to which noise term $n_k$ is added.
The noise variables $N_k$ for $k = 0, 1, \cdots, K - 1$ are jointly Gaussian.

For their expectation we get

$$E[N_k] = E \left[ \int_{-\infty}^{\infty} N_w(t)p(t - kT)dt \right] = \int_{-\infty}^{\infty} E[N_w(t)]p(t - kT)dt = 0.$$ 

Moreover their correlation

$$E[N_k N_{k'}] = E \left[ \int \int N_w(t)p(t - kT)N_w(t')p(t' - k'T)dtdt' \right]$$
$$= \int \int E[N_w(t)N_w(t')]p(t - kT)p(t' - k'T)dtdt'$$
$$= \int \int \frac{N_0}{2} \delta(t - t')p(t - kT)p(t' - k'T)dtdt'$$
$$= \int \frac{N_0}{2} p(t - kT)p(t - k'T)dt = \begin{cases} \frac{N_0}{2} & \text{for } k' = k, \\
0 & \text{for } k' \neq k. \end{cases}$$

Hence the noise variables (a) are Gaussian, (b) have zero mean, (c) are independent of each other, and (d) have variance $\frac{N_0}{2}$. 
The effective time-duration of the signals

\[ s_m(t) = \sum_{k=0}^{K-1} x_{mk} p(t - kT) \text{ for } m \in \{1, 2, \cdots, |M|\}, \]

is \( KT \).

The energy of the signal \( s_m(t), m \in \{1, 2, \cdots, |M|\} \), should therefore satisfy

\[
E_{s_m} = \int_{-\infty}^{\infty} s_m^2(t) dt = \int_{-\infty}^{\infty} \sum_k \sum_{k'} x_{mk} p(t - kT) x_{mk'} p(t - k'T) dt \\
= \sum_k \sum_{k'} x_{mk} x_{mk'} \int_{-\infty}^{\infty} p(t - kT)p(t - k'T) dt \\
= \sum_{k=0}^{K-1} x_{mk}^2 \leq PKT.
\]
We have transformed the $W$-bandlimited waveform channel into a vector channel, i.e.

$$
\begin{pmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{K-1}
\end{pmatrix}
= 
\begin{pmatrix}
x_{m,0} \\
x_{m,1} \\
\vdots \\
x_{m,K-1}
\end{pmatrix}
+ 
\begin{pmatrix}
n_0 \\
n_1 \\
\vdots \\
n_{K-1}
\end{pmatrix},
$$

where transmission of each component (dimension) requires $T = \frac{1}{2W}$ seconds.

The noise vector consists of $K$ independent Gaussian zero-mean components, each having variance $\frac{N_0}{2}$.

The code vectors $(x_{m,0}, \cdots, x_{m,K-1})$ should satisfy the power constraint

$$
\frac{1}{K} \sum_{k=0, K-1} x_{mk}^2 \leq TP = \frac{P}{2W}
$$

for all $m \in \{1, 2, \cdots, |M|\}$.

Moreover the actually transmitted waveforms $s_m(t)$ satisfy the bandwidth constraint

$$
S_m(f) = 0 \text{ for } |f| > W \text{ for all } m \in \{1, 2, \cdots, |M|\}.
$$
The $W$-bandwidth constraint imposes a restriction on the number of vector components (dimensions) that are available per second$^2$. It can be shown that this number is at most $2W$.

Our signaling method achieves the optimum since $1/T = 2W$. 

---

$^2$Wozencraft and Jacobs [1965], Dimensionality Theorem.
We have seen before that the capacity of the AGN channel is

\[ C_{\text{AGN}} = \frac{1}{2} \log_2 \left( 1 + \frac{E_x}{\sigma^2} \right) \text{ [bits/channel use]} \]

Note that there are \( 2W \) channel uses (dimensions) per second.
Moreover the energy per channel use \( E_x = PT = \frac{P}{2W} \).
Noise variance \( \sigma^2 = \frac{N_0}{2} \).
Therefore:

**Theorem (Shannon, 1948)**

The capacity (in bits per second) of the \( W \)-bandlimited waveform channel when the transmit power is \( P \), is

\[ C_{W\text{-bandlim.ch.}} = W \log_2 \left( 1 + \frac{P}{N_0 W} \right) \text{ [bits/second]} \]
Bandpass Channel, Model

- **Bandpass constraint.** For all signals

\[ S_m(f) = \int_{-\infty}^{\infty} s_m(t) \exp(-j2\pi ft) dt = 0 \text{ except for } |f \pm f_0| < W, \]

where \( W \) is the **bandwidth**, and \( f_0 \) the **center frequency**.

- **Power constraint.** If the (effective) time-duration of the signals is \( \Delta \) then the energy of all the signal \( s_m(t) \) should satisfy

\[ E_{s_m} = \int_{-\infty}^{\infty} s_m^2(t) dt \leq P\Delta. \]
Bandpass Channel, Quadrature Amplitude Modulation

- **Carrier transmission**, use frequency $f_0$.
- Take $T = \frac{1}{2W}$, let $a_k \in \mathcal{X}$, $b_k \in \mathcal{X}$ for $k = 0, 1, \ldots, K - 1$, then let
  \[
  s(t) = \sum_{k=0,1,\ldots,K-1} a_k p(t-kT) \sqrt{2} \cos(2\pi f_0 t) + b_k p(t-kT) \sqrt{2} \sin(2\pi f_0 t).
  \]
  
  This leads to $2 \cdot 2W = 4W$ orthonormal components per second. Observe that the total bandwidth is now $4W$ however.

- **Schematic:**

```
\[ \sum_k a_k p(t - kT) \times \sqrt{2} \cos(2\pi f_0 t) \]
\[ \sum_k b_k p(t - kT) \times \sqrt{2} \sin(2\pi f_0 t) \]
\[ + \]
\[ s(t) \]
```
Again note that the **capacity of the AGN channel** is

\[ C_{\text{AGN}} = \frac{1}{2} \log_2(1 + \frac{E_x}{\sigma^2}) \]  

[bits/channel use].

Now there are 4W channel uses (dimensions) per second.

Therefore the energy per channel use \( E_x = PT = \frac{P}{4W} \).

**Noise variance** \( \sigma^2 = \frac{N_0}{2} \).

Therefore:

**Theorem**

*The capacity (in bits per second) of the W-bandpass waveform channel when the transmit power is P, is*

\[ C_{W\text{-bandpass.ch.}} = 2W \log_2(1 + \frac{P}{2N_0W}) \]  

[bits/second].

**Spectral Efficiency (capacity per Hz):**

\[ \log_2(1 + \frac{E_c}{N_0}) \]  

[bit/second/Hz],

where \( E_c \) is the energy per \((a, b)\) pair.
For signal-to-noise ratio $\text{SNR} = P/\sigma^2$ horizontally in dB the AGN capacity $C_{\text{AGN}}$ in bits/chan use is depicted in **BLACK** in the figure below.
Instead of using a Gaussian inputs we can use equally likely binary inputs $-\sqrt{\text{SNR}}$ and $+\sqrt{\text{SNR}}$, assuming that $\sigma^2 = 1$.

The capacity of such a binary-in, soft-out channel is

$$C_{2,\text{soft}} = \frac{\text{SNR} - E[\ln(\cosh(\text{SNR} + \sqrt{\text{SNR}} N))]}{\ln(2)}.$$ 

This capacity is depicted in the plot in BLUE. Note that for $\text{SNR} \downarrow 0$ this capacity approaches $C_{\text{AGN}}$.

The receiver can make a hard-decision based on the channel output, with a threshold at 0. The resulting channel is a BSC with cross-over probability

$$p = Q(\sqrt{\text{SNR}}) = \int_{\sqrt{\text{SNR}}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\alpha^2}{2}\right) d\alpha.$$ 

The capacity

$$C_{2,\text{hard}} = 1 - h(p)$$

of this binary-in, hard-out channel is depicted in RED in the figure. For $\text{SNR} \downarrow 0$ hard decision results in an SNR-loss of roughly 2 dB.
For larger SNR’s we need more signal points. Assume that we use **equidistant** and **equiprobable** points. This leads to the following constellations:

- $M = 2$ (2-PAM):
  
  \[
  \begin{array}{c|c}
  1/2 & 1/2 \\
  \hline
  -1 & +1
  \end{array}
  \]

- $M = 4$ (4-PAM):
  
  \[
  \begin{array}{c|c|c|c}
  1/4 & 1/4 & 1/4 & 1/4 \\
  \hline
  -3 & -1 & +1 & +3
  \end{array}
  \]

- $M + 8$ (8-PAM):
  
  \[
  \begin{array}{c|c|c|c|c|c|c|c}
  1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 \\
  \hline
  -7 & -5 & -3 & -1 & +1 & +3 & +5 & +7
  \end{array}
  \]

Note that the **average energy** of a signal set is

\[
E_{\text{PAM}} = \frac{M^2 - 1}{3}.
\]
For signal-to-noise ratio $\text{SNR} = \frac{E_{\text{PAM}}}{\sigma^2}$ horizontally in dB the PAM “capacities” in blue and the AGN capacity $C_{\text{AGN}}$ in black in bits/channel use are depicted in the figure below.

Observe that for $2^K$-PAM can at most reach a capacity of $K$ bit.
A Convolutional Code (NASA Code)

- Elias [1955]
- Binary input digits $b_1, b_2, \cdots, b_K$ are independent and uniform.
- Digits are encoded using a 64-state convolutional encoder. Schematic:

![Convolutional Encoder Diagram]

- Description:
  \[
  c_1(k) = b(k) \oplus b(k - 2) \oplus b(k - 3) \oplus b(k - 4) \oplus b(k - 6),
  \]
  \[
  c_2(k) = b(k) \oplus b(k - 1) \oplus b(k - 2) \oplus b(k - 3) \oplus b(k - 6).
  \]
- One input digit produces two output digits, code rate is 1/2.
- **Free Hamming distance** $d_H = 10$ of the code is 10, hence up to 4 errors can be corrected, and 5 detected.
- When used e.g. for the AGN channel, **soft-decision decoding can be realized by the Viterbi algorithm [1967].**
- **NASA code (WiFi).**
Our $R = 1/2$ code has constraint length $\nu = 7$. Coding gain at bit-error probability $10^{-5}$ is roughly 6 dB. Gap to the Shannon bound is 3.8 dB. (Clark and Cain [1981])

Figure 6-13. Bit error probability for $R = 1/2$ codes with Viterbi decoding (infinite quantization) and PSK modulation.
**Turbo Codes** (Berrou, Glavieux, and Thitimajhshima [1993])

Based on systematic recursive convolutional codes connected by an interleaver.

- Near-Shannon-limit performance (with a dB).
**LDPC Codes**

- **Low Density Parity Check (LDPC)-codes** (Gallager [1963], rediscovered in 1993):

  Code is specified by its **parity-check matrix**. The symbol nodes on the left are checked by equation nodes on the right. Low density of the matrix makes **message-passing algorithms** possible.

- Near-Shannon-limit performance (within tenths of a dB).
Polar Codes

- **Polar codes** Arikan [2006], Arikan and Telatar [2007].

IDEA (Polarization): Two identical channels can be transformed into a channel that is better and a channel that is worse than the original ones. The sum of the capacities remains constant however.

**Shannon-limit performance.** Decoding complexity $|M| \log_2 |M|$. 
- **Modulation** maps binary digits onto signals for the AGN channel, e.g. three binary digits map onto an 8-PAM signal.

- **Coding** is used to map message (data) sequences onto a set of binary codewords. These codewords are input to the modulator. The codewords are chosen such that the corresponding signal sequences are e.g. far apart (large Euclidean distance), or e.g. maximize mutual information between AGN channel input and output.

Block diagram:
Trellis Coded Modulation

- **COMBINE CODING and MODULATION**, Ungerboeck [1982]
- **BASELINE**: Uncoded 4-PAM. Now average energy $E_{PAM}(unc) = 5$ and squared Euclidean distance $d_E^2(unc) = 4$.

\[
\begin{array}{cccc}
-3 & -1 & +1 & +3 \\
\end{array}
\]

- Coding starts by expanding the signal constellation to 8-PAM. Now $E_{PAM}(cod) = 21$.

\[
\begin{array}{cccccccc}
-7 & -5 & -3 & -1 & +1 & +3 & +5 & +7 \\
\end{array}
\]

- **Partition the 8-PAM signal set** into 4 subsets $A_{00}$, $A_{01}$, $A_{11}$, and $A_{10}$, each containing 2 signals for which the distance is 8.

\[
\begin{array}{cccccccc}
A_{00}(0) & A_{00}(1) & A_{01}(0) & A_{01}(1) & A_{11}(0) & A_{11}(1) & A_{10}(0) & A_{10}(1) \\
-7 & +1 & -5 & +3 & -3 & +5 & -1 & +7 \\
\end{array}
\]

- The distance between signals in different subsets can be as small as 2.
- Points in subsets with complementary labels have a distance 4 or more.
• **Use the NASA code.** For each channel use let the **coded** binary digits $c_1$ and $c_2$ determine the subset $A_{c_1c_2}$ and the uncoded bit $b_2$ determine the symbol within this subset. Hence the mapper realizes

$$x = A_{c_1c_2}(b_2).$$

• Decoding done with **Viterbi algorithm**. Trellis structure:
Trellis Coded Modulation

- **Distance analysis:** An error event starts and ends with $c_1 = c_2$. This leads to a starting and an ending subset with complementary labels and squared distance $4^2$.

  For the Euclidean distance between the 8-PAM sequences we obtain, using $d_H = 10$ of the NASA code that

  $$d_E^2(\text{cod}) = \min(8^2, (d_H - 4)^2 + 2 \cdot 4^2) = 56.$$  

- What we have gained is now

  $$G = \frac{d_E^2(\text{cod})/E_{av}(\text{cod})}{d_E^2(\text{unc})/E_{av}(\text{unc})} = \frac{56/21}{4/5} = 10/3 = 5.2 \text{ dB}.$$  

- The gain $G$ is **asymptotic coding gain**. This implies that at large SNR TCM achieves the same error probability as uncoded transmission with 5.2 dB less SNR.

- We followed here the **Pragmatic approach to trellis coded modulation** (Viterbi, Wolf, Zehavi, and Padovani [1989]).

- Ungerboeck received the **Shannon Award** recently from the IEEE Information Theory Society.
Consider a one-to-one mapping from three binary digits to an 8-PAM symbol. We consider here a set partition mapping.

<table>
<thead>
<tr>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$b_3$</th>
<th>$x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>−5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>−3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>−1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>+1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>+3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>+5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+7</td>
</tr>
</tbody>
</table>

Another representation of this mapping

<table>
<thead>
<tr>
<th>000</th>
<th>100</th>
<th>010</th>
<th>110</th>
<th>001</th>
<th>101</th>
<th>011</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>−7</td>
<td>−5</td>
<td>−3</td>
<td>−1</td>
<td>+1</td>
<td>+3</td>
<td>+5</td>
<td>+7</td>
</tr>
</tbody>
</table>

Note that the distance increases by a factor of 2 after $b_1$ is exposed. If in addition $b_2$ is exposed the distance is again increased by a factor of 2.

Ungerboeck.
Multi-Level Coding: Use Binary Code for Every Label

- Use a first (strong) code for bit labels $b_1$, a second code for bit labels $b_2$, and a third code for bit labels $b_3$. All codewords have length $N$.

```
  code 1  \[ b_1(1)b_1(2) \cdots b_1(N) \]

  code 2  \[ b_2(1)b_2(2) \cdots b_2(N) \]

  code 3  \[ b_3(1)b_3(2) \cdots b_3(N) \]

  mapper  \[ x(1)x(2) \cdots x(N) \]
```

- Bit labels $B_1$, $B_2$, and $B_3$ are uniform and independent of each other.
- The mapper combines the three codewords into a 8-PAM sequence of length $N$ that is transmitted.
Decoding the Binary Codes for All Labels Sequentially Using Already Decoded Results

- The first decoder decodes the first bit-label sequence $b_1(1)b_1(2) \cdots b_1(N)$.
- Then the second decoder decodes the second bit-label sequence $b_2(1)b_2(2) \cdots b_2(N)$, using the decoded first bit-label sequence.
- Finally the third decoder decodes the third bit-label sequence $b_3(1)b_3(2) \cdots b_3(N)$, using the decoded first bit-label sequence and the decoded second bit-label sequence.

![Block diagram]

- Decoding must be performed **sequentially**.
The first bit channel has as input $B_1$ and as output $Y$.
The second bit channel has as input $B_2$ and as output $(YB_1)$.
The third bit channel has as input $B_3$ and the output is $(YB_1B_2)$.
Therefore the **mutual information (MI)** is:

$$I(B_1; Y) + I(B_2; YB_1) + I(B_3; YB_1B_2)$$

$$= I(B_1; Y) + I(B_1; B_2) + I(B_2; Y|B_1) + I(B_3; B_1B_2) + I(B_3; Y|B_1B_2)$$

$$= I(B_1; Y) + I(B_2; Y|B_1) + I(B_3; Y|B_1B_2)$$

$$= I(B_1B_2B_3; Y)$$

$$= I(X; Y)$$

**This implies that there is no loss!**

- Imai and Hirakawa [1977].
Decoding Not Using Already Decoded Results

**Bit-Interleaved Coded Modulation (Zehavi [1991], also Caire, Taricco, and Biglieri [1998]):**

- The first decoder decodes the first bit-label sequence $b_1(1)b_1(2)\cdots b_1(N)$.
- The second decoder decodes the second bit-label sequence $b_2(1)b_2(2)\cdots b_2(N)$.
- The third decoder decodes the third bit-label sequence $b_3(1)b_3(2)\cdots b_3(N)$.

**Block diagram:**

- Decoding can be performed *in parallel.*
Bit-Interleaved Capacity Analysis

- The first bit channel has as input $B_1$ and as output $Y$.
- The second bit channel has as input $B_2$ and as output $Y$.
- The third bit channel has as input $B_3$ and the output is $Y$.
- Therefore now the **generalised mutual information (GMI)** is:

$$I(B_1; Y) + I(B_2; Y) + I(B_3; Y)$$

$$\leq I(B_1; Y) + I(B_2; Y, B_1) + I(B_3; Y, B_1, B_2)$$

$$= I(B_1; Y) + I(B_2; Y|B_1) + I(B_3; Y|B_1, B_2)$$

$$= I(B_1, B_2, B_3; Y)$$

$$= I(X; Y)$$

This implies that **now there could be loss!**
Consider a one-to-one mapping from three binary digits to an 8-PAM symbol. We consider now a Gray mapping\(^3\).

<table>
<thead>
<tr>
<th>(b_1)</th>
<th>(b_2)</th>
<th>(b_3)</th>
<th>(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−7</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>−5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>−3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>−1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>+5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>+7</td>
</tr>
</tbody>
</table>

Another representation of this mapping

<table>
<thead>
<tr>
<th>(b_1)</th>
<th>(b_2)</th>
<th>(b_3)</th>
<th>(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>001</td>
<td>011</td>
<td>010</td>
</tr>
<tr>
<td>−7</td>
<td>−5</td>
<td>−3</td>
<td>−1</td>
</tr>
</tbody>
</table>

Note that only one digit changes if we go from a signal point to its neighbour.

\(^3\)Binary Reflected Gray Mapping.
The figure contains the capacities of the sub-channels and the total bit-interleaved capacity for Gray coding.

Loss is acceptable for the Gray mapping!
Bit-Interleaved Coded Modulation, Use a SINGLE Binary Code

Transmitter:

- Single code (length $3N$)
- $b_1(1)b_1(2)\cdots b_1(N)$
- $b_2(1)b_2(2)\cdots b_2(N)$
- $b_3(1)b_3(2)\cdots b_3(N)$
- Mapper
- $x(1)x(2)\cdots x(N)$
Bit-Interleaved Coded Modulation: Use a SINGLE Binary Code

- **Receiver:**

  
  \[
  \begin{align*}
  y(1)y(2)\ldots y(N) & \rightarrow \text{single decoder} \\
  & \text{length } 3N \\
  b_1(1)b_1(2)\ldots b_1(N) & \rightarrow b_2(1)b_2(2)\ldots b_2(N) \\
  & \rightarrow b_3(1)b_3(2)\ldots b_3(N)
  \end{align*}
  \]

- **Log-Likelihood Ratio calculation:**

  \[
  LLR_i = \frac{\sum_{b_1,b_2,b_3:b_i=0} p(y|x(b_1,b_2,b_3))}{\sum_{b_1,b_2,b_3:b_i=1} p(y|x(b_1,b_2,b_3))} \\
  \approx \frac{\max_{b_1,b_2,b_3:b_i=0} p(y|x(b_1,b_2,b_3))}{\max_{b_1,b_2,b_3:b_i=1} p(y|x(b_1,b_2,b_3))}.
  \]

- **Used everywhere.**
Remarks

- **Trellis Coded Modulation (Ungerboeck)** focussed on obtaining *distance gain*.
- **Multi-Level Coding** and **Bit-Interleaved Coded Modulation** is based on *mutual information considerations*. 
For signal-to-noise ratio SNR horizontally in dB the AGN capacity $C_{\text{AGN}}$ in bits/channel use is depicted in black in the figure below.

The curves for uniform 2-PAM, 4-PAM, 8-PAM, 16-PAM and 32-PAM are depicted in blue. A gap to AGN-capacity appears since the PAM inputs are not Gaussian, but Uniform.
Assumptions: (a) $M$-PAM where $M \to \infty$ and (b) SNR $\to \infty$ or equivalently that $\sigma_n^2 \to 0$.

Consider difference of the capacity $I(X_g; Y_g)$, where $X_g$ is the Gaussian channel input and $Y_g$ the corresponding output, and the mutual information $I(X_u; Y_u)$, where $X_u$ is a uniform channel input and $Y_u$ the corresponding output:

$$I(X_g; Y_g) - I(X_u; Y_u) = h(Y_g) - h(Y_g | X_g) - h(Y_u) + h(Y_u | X_u)$$

$$= h(Y_g) - h(Y_u)$$

$$= \frac{1}{2} \log_2 (2\pi e \sigma_x^2) - \log_2 12 \sigma_x^2 = \frac{1}{2} \log_2 \frac{\pi e}{6} = 0.2546 \text{ bit,}$$

or equivalently $1.53$ dB in SNR.
Probabilistic Shaping: Equidistant Signal Points

Eight equidistant signals $x \in \{-7\gamma, -5\gamma, \cdots, +7\gamma\}$ for $\gamma = 1.089$.

Non-uniform probability distribution $P(x)$: $\{0.0521, 0.0989, 0.1562, 0.1927, 0.1927, 0.1562, 0.0989, 0.0521\}$.

In the plot:

\[
p(x, y) = P(x)p(y|x) \text{ for } x \in \{-7\gamma, -5\gamma, \cdots, +7\gamma\}
\]

\[
p(y) = \sum_x P(x)p(y|x) \text{ where } p(y|x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y-x)^2}{2}\right).
\]

Moreover $I(X; Y) = 2.001$ bit at SNR = 11.96 dB.
Geometric Shaping: Equi-probable Signals

Uniform probability distribution $P(x) : \{1/8, 1/8, \cdots, 1/8\}$ for all possible $x$.

Eight non-equidistant signals: $x \in \{-6.86, -3.98, -2.10, -0.56, +0.56, +2.10, +3.98, +6.86\}$.

In the plot
\[
p(x, y) = P(x)p(y|x) \quad \text{for} \quad x \in \{-6.86, -3.98, \cdots, +6.86\}
\]
\[
p(y) = \sum_x P(x)p(y|x) \quad \text{where} \quad p(y|x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y-x)^2}{2}\right)
\]

Moreover $I(X; Y) = 2.001$ bit at SNR $= 12.28$ dB.
In the plot the **output densities** \( p(y) \). **Blue:** Probabilistic Shaping (SNR = 11.96 dB). **Red:** Geometric Shaping (SNR = 12.28 dB). \( I(X; Y) = 2.001 \) bit in both cases.
Q: How can we generate sequences with a given composition?

We can use Distribution Matching (Boecherer [2014], Schulte and Boecherer [2016]).

Distribution matching is inspired by arithmetic data compression techniques (e.g. Langdon and Rissanen [1979], Witten, Neal, and Cleary [1987]). In their methods sequences are represented by intervals.
We want to generate binary sequences of length 5 containing 2 ones. There are \( \binom{5}{2} = 10 \) such sequences.

Each of these sequences corresponds to a subinterval of length 1/10 of the [0, 1) interval. This interval can be computed sequentially from the sequence. The first digit of the sequence splits the interval in fractions 3/5 and 2/5. After a first 0 the interval [0, 3/5) is split according to 2/4 and 2/4, etc.

Note that the sequences and their intervals are now in a lexicographical order.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sequences and Intervals, Example

We want to generate binary sequences of length 5 containing 2 ones. There are \( \binom{5}{2} = 10 \) such sequences.

Each of these sequences corresponds to a subinterval of length 1/10 of the [0, 1) interval.

This interval can be computed **sequentially** from the sequence. The first digit of the sequence splits the interval in fractions 3/5 and 2/5. After a first 0 the interval [0, 3/5) is split according to 2/4 and 2/4, etc.

Note that the sequences and their intervals are now in a **lexicographical order**.

```
0.0

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>0.0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.7</td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.7</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.6</td>
<td>0.3</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.6</td>
<td>1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.8</td>
<td>1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>
```

We want to generate binary sequences of length 5 containing 2 ones. There are \( \binom{5}{2} = 10 \) such sequences. Each of these sequences corresponds to a subinterval of length 1/10 of the [0, 1) interval. This interval can be computed **sequentially** from the sequence. The first digit of the sequence splits the interval in fractions 3/5 and 2/5. After a first 0 the interval [0, 3/5) is split according to 2/4 and 2/4, etc. Note that the sequences and their intervals are now in a **lexicographical order**.
We want to generate binary sequences of length 5 containing 2 ones. There are \(^5\binom{2}{2} = 10\) such sequences.

Each of these sequences corresponds to a subinterval of length 1/10 of the [0, 1) interval.

This interval can be computed sequentially from the sequence. The first digit of the sequence splits the interval in fractions 3/5 and 2/5. After a first 0 the interval [0, 3/5) is split according to 2/4 and 2/4, etc.

Note that the sequences and their intervals are now in a lexicographical order.
Sequences and Intervals, Example

We want to generate binary sequences of length 5 containing 2 ones. There are \( \binom{5}{2} = 10 \) such sequences.

Each of these sequences corresponds to a subinterval of length 1/10 of the [0, 1) interval.

This interval can be computed **sequentially** from the sequence. The first digit of the sequence splits the interval in fractions 3/5 and 2/5. After a first 0 the interval [0, 3/5) is split according to 2/4 and 2/4, etc.

Note that the sequences and their intervals are now in a **lexicographical order**.

<table>
<thead>
<tr>
<th></th>
<th>0.0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
</tr>
</tbody>
</table>
Sequences and Intervals, Example

We want to generate binary sequences of length 5 containing 2 ones. There are $\binom{5}{2} = 10$ such sequences.

Each of these sequences corresponds to a subinterval of length $1/10$ of the $[0, 1)$ interval.

This interval can be computed **sequentially** from the sequence. The first digit of the sequence splits the interval in fractions $3/5$ and $2/5$. After a first 0 the interval $[0, 3/5)$ is split according to $2/4$ and $2/4$, etc.

Note that the sequences and their intervals are now in a **lexicographical order**.
Consider 3-digit indices 000, 001, ⋯, 111. Index $b_1 b_2 b_3$ connects to a constant composition sequence if $b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3}$ is in the interval corresponding to the constant composition sequence.

### Table: Constant Composition Sequence Intervals, Indices

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0.9</th>
<th>0.8</th>
<th>0.7</th>
<th>0.6</th>
<th>0.5</th>
<th>0.4</th>
<th>0.3</th>
<th>0.2</th>
<th>0.1</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.875(111)</td>
<td>0.750(110)</td>
<td>0.625(101)</td>
<td>0.500(100)</td>
<td>0.375(011)</td>
<td>0.250(010)</td>
<td>0.125(001)</td>
<td>0.000(000)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td>0.750(110)</td>
<td>0.625(101)</td>
<td>0.500(100)</td>
<td>0.375(011)</td>
<td>0.250(010)</td>
<td>0.125(001)</td>
<td>0.000(000)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) Only one index can connect to a sequence since $2^{-3} \geq 1/10$.
(b) Observe that for not all constant composition sequences there is an index.
(c) For all indices there is a sequence however.
Consider 3-digit indices 000, 001, ⋯ , 111. Index \( b_1 b_2 b_3 \) connects to a constant composition sequence if \( b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} \) is in the interval corresponding to the constant composition sequence.

(a) Only one index can connect to a sequence since \( 2^{-3} \geq 1/10 \).
(b) Observe that for not all constant composition sequences there is an index.
(c) For all indices there is a sequence however.
From an index to a const. comp. sequence and back

- Using this method (distribution matching) we find a constant composition sequence $a$ for all indices $i$, and from such a constant composition sequence $a$ the original index $i$ can be recovered (inverse distribution matching).

$$i \rightarrow \text{distribution matcher} \rightarrow a \rightarrow \text{inverse distr. matcher} \rightarrow i$$

- **Use as index the message $m$ that is to be transmitted.** The resulting const. comp. sequence $a$ can be used as amplitude sequence.

- Now take a short block length $N = 96$. The amplitude level composition is

$$96 \ast (0.1927, 0.1562, 0.0989, 0.0521) \ast 2 \approx (37, 30, 19, 10).$$

This leads to

$$\frac{96!}{37!30!19!10!} = 2^{168.72} \text{ const. comp. sequences.}$$

**Rate** is $\frac{168}{96} = 1.75$ [bit/symbol], and the **sequence energy**

$$37 \ast 1 + 30 \ast 9 + 19 \ast 25 + 10 \ast 49 = 1272.$$
Combining Shaping with Coding

- Schematic:

The distribution matcher converts message \( m \) into amplitude sequence \( a \) of the desired composition.

The Gray demapper (sign bit missing!) converts the amplitude sequence \( a \) into the two amplitude bitstreams \( b_2 \) and \( b_3 \) both of length \( N \).

<table>
<thead>
<tr>
<th>( a )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Now parity is generated from \( b_2 \) and \( b_3 \), using a systematic code of rate \( 2/3 \). This parity is used as bitstream \( b_1 \). This bitstream represents the sign bits.
Combining Shaping with Coding

- The three bitstreams are combined into an 8-PAM symbol $x$ stream, using Gray mapping.

<table>
<thead>
<tr>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$b_3$</th>
<th>$x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−7</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>−5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>−3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>−1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>+5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>+7</td>
</tr>
</tbody>
</table>

- We have described a **Bit-Interleaved Coded Modulation** construction, where only sequences with **constant amplitude composition** are generated.

- Log-Likelihood Ratio calculation now includes **a priori symbol information**.

$$LLR_i = \frac{\sum_{b_1b_2b_3:b_i=0} p(b_1b_2b_3)p(y|x(b_1, b_2, b_3))}{\sum_{b_1b_2b_3:b_i=1} p(b_1b_2b_3)p(y|x(b_1, b_2, b_3))}$$
Boecherer Simulations

- FER (frame error rate) = $10^{-3}$, LDPC codes from DVB-S2. Boecherer, Schulte, and Steiner [2016].
Consider composition \((37, 30, 19, 10)\) that leads to \(2^{168.72}\) sequences and sequence energy 1272.

**QUESTION:** Can we obtain more sequences such that the average sequence energy does not exceed 1272?

- Note first that there are more compositions with energy equal to 1272. Add the corresponding sequences. This leads to \(2^{172.75}\) sequences.
- Add all the sequences with an energy smaller than 1272. Now we obtain \(2^{175.04}\) sequences. Moreover the average energy drops to 1242.4.

If we are interested in rate 1.75 we can decrease the radius to \(\sqrt{1120}\). Now we find \(2^{168.03}\) sequences with average sequence energy 1096.9. Gain = \(\frac{1272}{1096.9} = 0.6431\) dB.

Analysis by Y. Gultekin [2017, TU/e].
Consider composition \((37, 30, 19, 10)\) that leads to \(2^{168.72}\) sequences and sequence energy 1272.

**QUESTION:** Can we obtain more sequences such that the average sequence energy does not exceed 1272?

- Note first that there are **more compositions** with energy equal to 1272. Add the corresponding sequences. This leads to \(2^{172.75}\) sequences.

- Add all the sequences with an **energy smaller than 1272**. Now we obtain \(2^{175.04}\) sequences. Moreover the **average energy drops** to 1242.4.

If we are **interested in rate** 1.75 we can decrease the radius to \(\sqrt{1120}\). Now we find \(2^{168.03}\) sequences with average sequence energy 1096.9.

\[ \text{Gain} = \frac{1272}{1096.9} = 0.6431 \text{ dB}. \]

Analysis by Y. Gultekin [2017, TU/e].
Consider composition (37, 30, 19, 10) that leads to $2^{168.72}$ sequences and sequence energy 1272.

**QUESTION:** Can we obtain more sequences such that the average sequence energy does not exceed 1272?

- Note first that there are more compositions with energy equal to 1272. Add the corresponding sequences. This leads to $2^{172.75}$ sequences.
- Add all the sequences with an energy smaller than 1272. Now we obtain $2^{175.04}$ sequences. Moreover the average energy drops to 1242.4.

If we are interested in rate 1.75 we can decrease the radius to $\sqrt{1120}$. Now we find $2^{168.03}$ sequences with average sequence energy 1096.9.

Gain = $\frac{1272}{1096.9} = 0.6431$ dB.

Analysis by Y. Gultekin [2017, TU/e].
Enumerative Shaping: Bounded Energy Trellis

Wuijts [1991, TU/e], W. and Wuijts [1993]

$N = 4$, amplitude alphabet is $\{1, 3, 5, \cdots \}$, $E_{\text{max}} = 28$ i.e. sphere radius $\sqrt{28}$. 

References

REMARKS
Lexicographical Ordering. Index of sequence 3131 is 13.
Lexicographical ordering. Sequence with index 8 is 1331.
Enumerative Shaping, Analysis

- Maximum energy level $E_{max} = 28$.
- **Adding signs** leads to $16 \cdot 19 = 304$ sequences.
- Total rate
  $$R_{tot} = \frac{\log_2 304}{4} = 2.062 \text{ bits/symbol.}$$
- Average energy per symbol $E_{av}/N = 5.211$.
- Gain
  $$G = \frac{2^{2R-1} - 1}{3 E_{av}/N} = 0.218 \text{ dB.}$$
- More results for rate $R \approx 2$, where $N$ is sequence-length.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$E_{max}$</th>
<th>$E_{av}/N$</th>
<th>$R_{tot}$</th>
<th>$G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>48</td>
<td>5.169</td>
<td>2.102</td>
<td>0.509</td>
</tr>
<tr>
<td>16</td>
<td>80</td>
<td>4.638</td>
<td>2.064</td>
<td>0.734</td>
</tr>
<tr>
<td>32</td>
<td>136</td>
<td>4.100</td>
<td>2.006</td>
<td>0.901</td>
</tr>
<tr>
<td>64</td>
<td>264</td>
<td>4.051</td>
<td>2.019</td>
<td>1.039</td>
</tr>
</tbody>
</table>
Combining Shaping with Coding

- Schematic:

  - The enumerative shaper converts a message $m$ into a bounded energy amplitude sequence $a$.
  - No difference with Boecherer’s approach.
Consider a hypersphere in $N$ (even) dimensions with radius $\rho$, then

$$h(X) = \frac{h(X_1) + h(X_2) + \cdots + h(X_N)}{N} \geq \frac{h(X_1, X_2, \cdots, X_N)}{N}$$

$$= \frac{1}{N} \log \frac{\pi^{N/2}}{(N/2)!} \rho^N$$

$$\geq \frac{1}{N} \log \frac{\pi^{N/2}}{(N/2)^{N/2} \sqrt{\frac{e^{2N}}{2}}} \rho^N$$

$$= \frac{1}{N} \log 2\pi e \frac{\rho^2}{N} - \frac{1}{2N} \log \frac{e^{2N}}{2}$$.

$$h(X) = \frac{h(X_1) + h(X_2) + \cdots + h(X_N)}{N} \leq \frac{\log 2\pi e E[X_1^2] + \log 2\pi e E[X_2^2] + \cdots + \log 2\pi e E[X_N^2]}{2N}$$

$$\leq \frac{1}{2} \log 2\pi e \frac{E[X_1^2] + E[X_2^2] + \cdots + E[X_N^2]}{N} = \frac{1}{2} \log 2\pi e \frac{\rho^2}{N}.$$

**HENCE** enumerative shaping for large $N$ results in Gaussians components!
Both **probabilistic shaping** and **enumerative shaping** lead to Gaussian input distributions, which is required to **achieve capacity**.

In a **coding environment** the bounded energy constraint could be **more elementary** than the Gaussian input constraint.

For short blocklengths there is something to gain with enumerative shaping.

For large blocklengths the gain is negligible (sphere hardening argument).

Both probabilistic shaping and enumerative shaping are lexicographical indexing methods.

Complexity of enumerative shaping is larger than that of probabilistic shaping.
Geometric Shaping

Advantage:
- No probabilistic shaper required.

Disadvantages:
- Mapping from symbols to bits causes problems (dependent bits).
- Performance (Steiner and Boecherer [2016]):

Smart ideas needed!

Fig. 4. Comparison of the coded performance of geometrically shaped ATSC 3.0 modcodes with one single PAS scheme.
Shaping References

- **Forney, Gallager, Lang, Longstaff, and Qureshi [1984]:** constellation shaping, coding and shaping can be separated.
- **Kschischang and Pasupathy [1990]:** variable-rate shaping, geometric shaping.
- **Calderbank and Ozarow [1991]:** shaping on regions.
- **Forney [1992]:** trellis shaping (more codewords for same data, choose lowest energy codeword), sign-bit shaping, constellation expansion, peak-to-average power ratio expansion.
- **Sun and van Tilborg [1993]:** geometrical shaping.
- **Laroia, Farvardin, and Tretter [1994]:** enumerative shaping (two schemes).
- **V34 Modem standard [1994]:** shell mapping, enumerative shaping.
- **Fischer [2002]:** overview.
Information Theory and Coding Theory is useful in understanding, analysing, and improving communication systems, also optical communication systems.

Performance indicators as e.g. capacities and mutual informations are very powerful, and well studied (see e.g. work of Szczecinski and Alvarado [2015], Guillen i Fabregas, Martinez, and Caire [2008]).

Next step. Non-linearities ...