Information Theory and Coding

M. Sc. Marko Hennhöfer

Ilmenau University of Technology
Communications Research Laboratory

Winter Semester 2011

Contents

Literature

- Thomas M. Cover, Joy A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.
- J. Proakis, *Digital Communications*. John Wiley & Sons, 4th edition, 2001.
- Branka Vucetic, Jinhong Yuan, *Turbo Codes – Principles and applications*. Kluwer Academic Publishers, 2000.

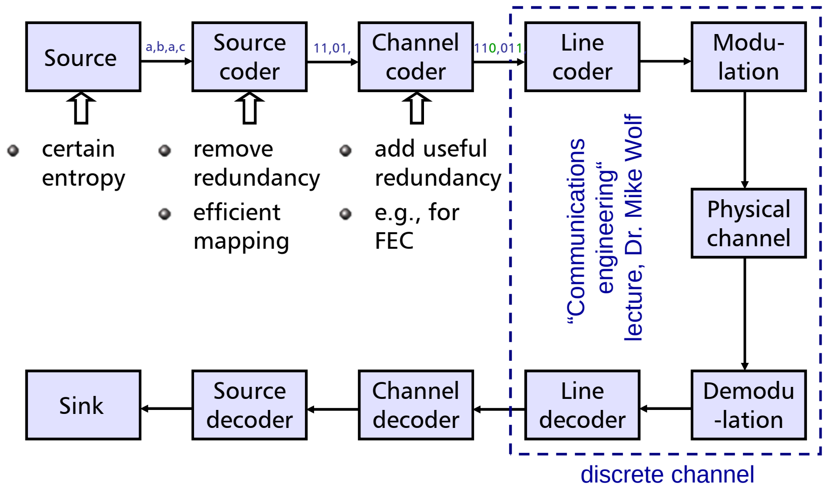
1 Review

Some references to refresh the basics:

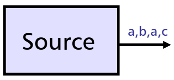
- S. Haykin and B. V. Veen, *Signals and Systems*. John Wiley & Sons, second edition, 2003.
- E. W. Kamen and B. S. Heck, *Fundamentals of Signals and Systems Using the Web and MATLAB*. Upper Saddle River, New Jersey 07458: Pearson Education, Inc. Pearson Prentice Hall, third ed., 2007.
- A. D. Poularikas, *Signals and Systems Primer with MATLAB*. CRC Press, 2007.
- S. Haykin, *Communication Systems*. John Wiley & Sons, 4th edition, 2001
- A. Papoulis, Probability, *Random Variables, and Stochastic Processes*. McGraw-Hill, 2nd edition, 1984.
- G. Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge Press, Wellesley, MA, 1993.

2 Information Theory

Overview: communication system



2.1 Information, entropy



e.g.: $\mathcal{S} = \{a, b, c\}$

- **Discrete source**, emits symbols from a given alphabet
  $\mathcal{S} = \{s_0, s_1, \ldots, s_{K-1}\}$
  - modelled via a random variable S with probabilities of occurence
    $P(S = s_k) = p_k;\ k = 0, 1, \ldots, K-1$
  - $\sum_{k=0}^{K-1} p_k = 1$
- **Discrete memoryless source**.
  - subsequent symbols are statistically independent

2.1 Information, entropy

What is the ammount of information being produced by this source?

- if: $\left.\begin{array}{l} p_k = 1 \\ p_i = 0;\ \forall i \neq k \end{array}\right\}$ no uncertainty, no surprise, i.e., no information

- for small p_k the surprise (information) is higher as compared to higher values of p_k

- Occurence of an event:
  - Information gain (removal of uncertainty $\sim \frac{1}{p_k}$
  - **Information** of the event $S = s_k$

$$\boxed{I(s_k) = \log\left(\frac{1}{p_k}\right) = -\log(p_k)}$$

2.1 Information, entropy

Properties of information:

- $I(s_k) = 0 \quad \text{if} \quad p_k = 1$

- $I(s_k) \geq 0 \quad \text{if} \quad 0 \leq p_k \leq 1$

  The event $S = s_k$ yields a gain of information (or no information) but never a loss of information.

- $I(s_k) > I(s_i) \quad \text{if} \quad p_k < p_i$

  The event with lower probability of occurence has the higher information

- $I(s_k s_l) = I(s_k) + I(s_l)$

  For statistically independend events s_k and s_l

2.1 Information, entropy

The basis of the logarithm can be chosen arbitrarily.

Usually: $\boxed{I(s_k) = \log_2(\frac{1}{p_k}) = -\log_2 p_k; \quad k = 0, 1, \dots, K-1}$

$$[I(s_k)] = \text{bit} \qquad (\textbf{bi}\text{nary digi}\textbf{t})$$

- Information if one of two equal probable events occurs
  $p_k = \frac{1}{2}: \quad I(s_k) = 1 \text{ bit}$
- $I(s_k)$ is a discrete random variable with probability of occurence p_k

2.1 Information, entropy

Entropy

- mean information of a source
  (here: discrete memoryless source with alphabet S)

$$\boxed{\begin{aligned} H(\mathcal{S}) &= E\{I(s_k)\} = \sum_{k=0}^{K-1} p_k I(s_k) \\ &= \sum_{k=0}^{K-1} p_k \log_2(\frac{1}{p_k}) \end{aligned}}$$

2.1 Information, entropy

Important properties of the entropy

- $0 \leq H(\mathcal{S}) \leq \log_2 K$

  where K is the number of Symbols in S

- $$H(\mathcal{S}) = 0 \Leftrightarrow \begin{cases} p_k = 1 \\ p_i = 0; \ \forall i \neq k \end{cases}$$

  no uncertainty

- $$H(\mathcal{S}) = \log_2 K \Leftrightarrow p_k = \frac{1}{K}; \ \forall k$$

  maximum uncertainty.
  All symbols occur with the same probabilities

2.1 Information, entropy

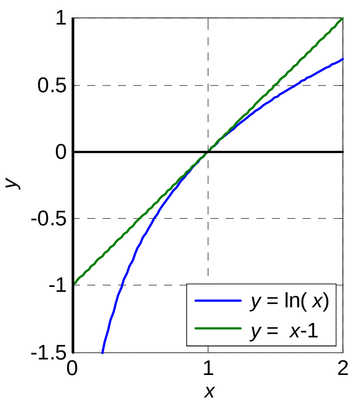
Bounds for the entropy

- Lower bound:
  $$p_k \leq 1; \ \forall k$$
  $$p_k \log_2\left(\frac{1}{p_k}\right) \geq 0; \ \forall k$$
  $$\Rightarrow \boxed{H(\mathcal{S}) \geq 0}$$

- Upper bound:
  Use $\ln x \leq x - 1; \quad x \geq 0$

  Given two distributions
  $\{p_0, p_1, \ldots, p_{K-1}\}$
  $\{q_0, q_1, \ldots, q_{K-1}\}$

  for the alphabet
  $\mathcal{S} = \{s_0, s_1, \ldots, s_{K-1}\}$



2.1 Information, entropy

Upper bound for the entropy continued:

$$\sum_{k=0}^{K-1} p_k \log_2\left(\frac{q_k}{p_k}\right) = \frac{1}{\ln 2}\sum_{k=0}^{K-1} p_k \ln\left(\frac{q_k}{p_k}\right) \leq \frac{1}{\ln 2}\sum_{k=0}^{K-1} p_k \left(\frac{q_k}{p_k} - 1\right)$$

$$= \frac{1}{\ln 2}\sum_{k=0}^{K-1} (q_k - p_k) = \frac{1}{\ln 2}\left(\sum_{k=0}^{K-1} q_k - \sum_{k=0}^{K-1} p_k\right) = 0$$

This yields Gibb's inequality:

$$\sum_{k=0}^{K-1} p_k \log_2\left(\frac{q_k}{p_k}\right) \leq 0 \qquad " = " \text{ if } q_k = p_k \quad \forall k$$

Now assume $q_k = \frac{1}{K}; \quad \forall k$ $\qquad \sum_{k=0}^{K-1} p_k \left[\log_2\left(\frac{1}{p_k}\right) - \log_2\left(\frac{1}{q_k}\right)\right] \leq 0$

$$\boxed{H(\mathcal{S}) = \sum_{k=0}^{K-1} p_k \log_2\left(\frac{1}{p_k}\right) \leq \sum_{k=0}^{K-1} p_k \log_2(K) = \log_2(K)}$$

2.1 Information, entropy

Summary:

$$\boxed{0 \leq \underbrace{H(\mathcal{S})}_{H_1} \leq \underbrace{\log_2(K)}_{H_0}}$$

- H_1 Entropy of the current source
- H_0 Entropy of the “best” source

- Redundancy and relative redundancy of the source

$$\boxed{R = H_0 - H_1 \qquad r_c = \frac{H_0 - H_1}{H_0}, \quad \text{in} \quad \%}$$

- High redundancy of a source is a hint that compression methods will be beneficial.
  E.g., Fax transmission:
  - ~90% white pixels
  - low entropy (as compared to the “best” source)
  - high redundancy of the source
  - redundancy is lowered by run length encoding

2.1 Information, entropy

Example: Entropy of a memoryless binary source

- Symbol 0 occurs with probability p_0
- Symbol 1 occurs with probability $p_1 = 1 - p_0$
- Entropy: $H(\mathcal{S}) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$

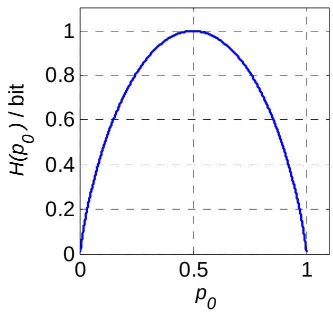
$$= -p_0 \log_2 p_0 - (1-p_0) \log_2 (1-p_0) \ \text{ bits}$$

Characteristic points:

$$p_0 = 0 : H(\mathcal{S}) = 0$$
$$p_0 = 1 : H(\mathcal{S}) = 0$$
$$H(\mathcal{S}) = 1 \text{ bit}, \text{ falls } p_1 = p_0 = \frac{1}{2}$$

$$\boxed{H(p_0) = -p_0 \log_2 p_0 - (1-p_0) \log_2 (1-p_0)}$$

Entropy function (Shannon's Function)



2.1 Information, entropy

Extended (memoryless) sources:
Combine *n* primary symbols from S
to a block of symbols (secondary symbols from S^n)

$$\boxed{H(\mathcal{S}^n) = n \cdot H(\mathcal{S})}$$

Example:

$$\mathcal{S} = \{s_0, s_1, s_2\}, \quad \text{with} \quad p_0 = \frac{1}{4}, \ p_1 = \frac{1}{4}, \ p_2 = \frac{1}{2}$$

$$H(\mathcal{S}) = \frac{1}{4} \cdot \log_2(4) + \frac{1}{4} \cdot \log_2(4) + \frac{1}{2} \cdot \log_2(2) = \underline{\frac{3}{2} \ \text{ bits}}$$

e.g., *n*=2, the extended source will have 3^n =9 symbols, $\mathcal{S}^2 = \{e_0, e_1, ..., e_8\}$

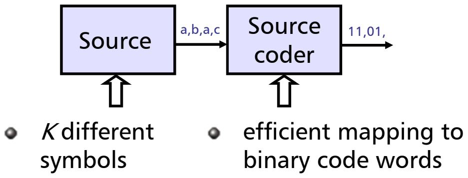
secondary symbol	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
primary symbols	s_0s_0	s_0s_1	s_0s_2	s_1s_0	s_1s_1	s_1s_2	s_2s_0	s_2s_1	s_2s_2
probability $p(e_i)$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$

$$H(\mathcal{S}^2) = \sum_{i=0}^{8} p(e_i) \log_2 \left(\frac{1}{p(e_i)} \right) = 2 \cdot \frac{3}{2} \text{ bits} = \underline{3 \text{ bits}}$$

2.2 Source Coding

Source coding theorem (Shannon)

- Efficient representation (Coding) of data from a discrete source
- Depends on the statistics of the source
  - short code words for frequent symbols
  - long code words for rare symbols
- Code words must uniquely decodable



s_k has the probabilities of occurence p_k and the code word length l_k

2.2 Source Coding

Source coding theorem (Shannon)

- Mean code word length (as small as possible)

$$\boxed{H_c = \sum_{k=0}^{K-1} p_k l_k}$$

- Given a discrete source with entropy $H(S) = H_1$.
  For uniquely decodable codes the entropy is the lower bound for the mean code word length:

$$\boxed{H_c \geq H_1}$$

- Efficiency of a code:

$$\boxed{\eta = \frac{H_1}{H_c}}$$

- Redundancy and relative redundancy of the coding:

$$\boxed{R_c = H_c - H_1 \quad r_c = \frac{H_c - H_1}{H_c}, \quad \text{in} \quad \%}$$

2.2 Source Coding

Fano Coding

- Important group of prefix codes
- Each symbol gets a code word assigned that approximately matches it's infomation
- Fano algorithm:
  1. Sort symbols with decreasing probabilities. Split symbols to groups with approximately half of the sum probabilities
  2. Assign "0" to one group and "1" to the other group.
  3. Continue splitting

Fano Coding, example:

Code the symbols S={a, b, c, d, e, f, g, h} efficiently. Probabilities of occurence p_k={0.15,0.14, 0.13, 0.1, 0.12, 0.08, 0.06, 0.05}

2.2 Source Coding

Fano Coding, example:

Symbol	prob.					CW	l_k / bit
c	0.3	0	0			00	2
a	0.15	0	1			01	2
b	0.14	1	0	0		100	3
e	0.12	1	0	1		101	3
d	0.1	1	1	0	0	1100	4
f	0.08	1	1	0	1	1101	4
g	0.06	1	1	1	0	1110	4
h	0.05	1	1	1	1	1111	4

Source Entropy

$$H_1 = 2.78 \frac{\text{bit}}{\text{symbol}}$$

Mean CW length

$$H_c = 2.84 \frac{\text{bit}}{\text{symbol}}$$

Redundancy

$$R_c = 0.06 \frac{\text{bit}}{\text{symbol}}$$

$$r_c = 2.14\%$$

Efficiency

$$\eta = 97.86\%$$

In average 0.06 bit/symbol more need to be transmitted as information is provided by the source. E.g., 1000 bit source information -> 1022 bits to be transmitted.

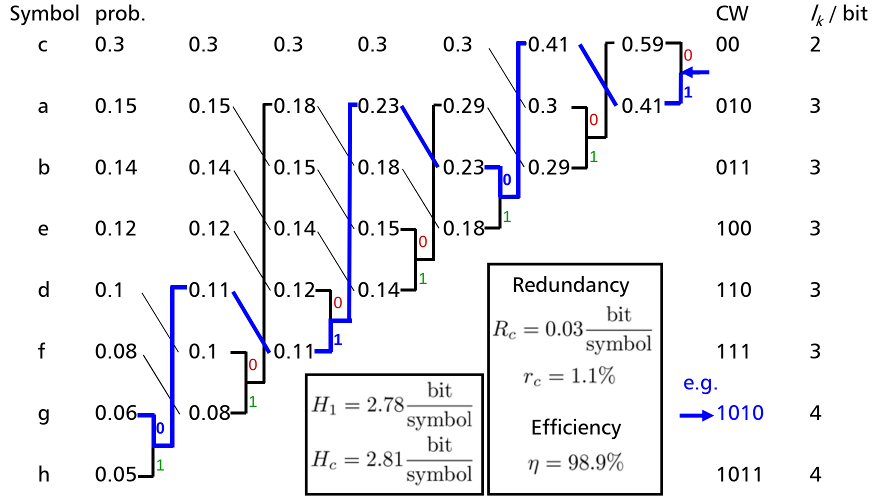
2.2 Source Coding

Huffman Coding

- Important group of prefix codes
- Each symbol gets a code word assigned that approximately matches it's infomation
- Huffman coding algorithm:
  1. Sort symbols with decreasing probabilities. Assign "0" and "1" to the symbols with the two lowest probabilities
  2. Both symbols are combined to a new symbol with the sum of the probabilities. Resort the symbols again with decreasing probabilities.
  3. Repeat until the code tree is complete
  4. Read out the code words from the back of the tree

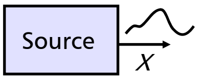
2.2 Source Coding

Huffman Coding, example:



In average 0.03 bit/symbol more need to be transmitted as information is provided by the source. E.g., 1000 bit source information -> 1011 bits to be transmitted.

2.3 Differential entropy



- **Continuous (analog) source**
  - modelled via a continuous random variable X with pdf $f_X(x)$.
- differential entropy

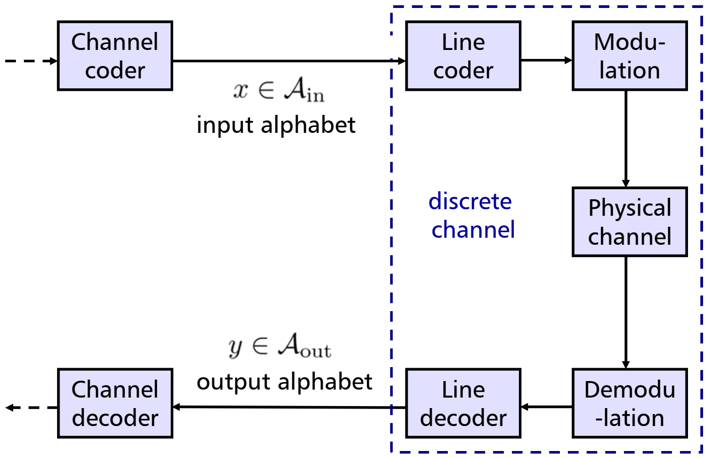
$$\boxed{h(X) = \int_{-\infty}^{\infty} f_X(x) \cdot \log_2 \left(\frac{1}{f_X(x)} \right) \mathrm{d}x = -\int_{-\infty}^{\infty} f_X(x) \cdot \log_2 \left(f_X(x) \right) \mathrm{d}x}$$

- Example: Gaussian RV with pdf $f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \mathrm{e}^{-\frac{x^2}{2\sigma^2}}$

$$\boxed{h(X) = \frac{1}{2} \cdot \log_2 \left(2\pi \mathrm{e} \sigma^2 \right)}$$

2.4 The discrete channel

The discrete channel



2.4 The discrete channel

Discrete channel:

- $\mathcal{A}_{\text{in}}$: Input alphabet with q values/symbols. Easiest case $q = 2$, i.e., binary codes. Commonly used $q = 2^m, m \in \mathcal{N}$, i.e., symbols are bit groups.
- $\mathcal{A}_{\text{out}}$: Output values
  - **Hard decision**: $\mathcal{A}_{\text{out}} = \mathcal{A}_{\text{in}}$
    Decoder estimates directly the transmitted values, e.g., in the binary case $\mathcal{A}_{\text{out}} = \mathcal{A}_{\text{in}} \in \{0, 1\}$.
  - **Soft decision**:
    $\mathcal{A}_{\text{out}}$ has more values as $\mathcal{A}_{\text{in}}$. Extreme case: $\mathcal{A}_{\text{out}} \in \mathcal{R}$, continuous-valued output. Allows measures for the reliability of the decision

2.4 The discrete channel

Conditional probabilities / transition probabilities:

- $P_{Y|X}(\eta, \xi)$
  conditional probability that $Y = \eta$ is received if $X = \xi$ has been transmitted.
- X, Y are assumed to be random variables with $\eta \in \mathcal{A}_{\text{out}}$ and $\xi \in \mathcal{A}_{\text{in}}$.

Discrete memoryless channel, DMC:

- Subsequent symbols are statistically independent.
  Example: Probability that a 00 is received if a 01 has been transmitted.

$$P(00|01) = P(0|0) \cdot P(0|1)$$

General:

$$\boxed{P(y_0, ..., y_{N-1}|x_0, ..., x_{N-1}) = \prod_{i=0}^{N-1} P(y_i|x_i)}$$

2.4 The discrete channel

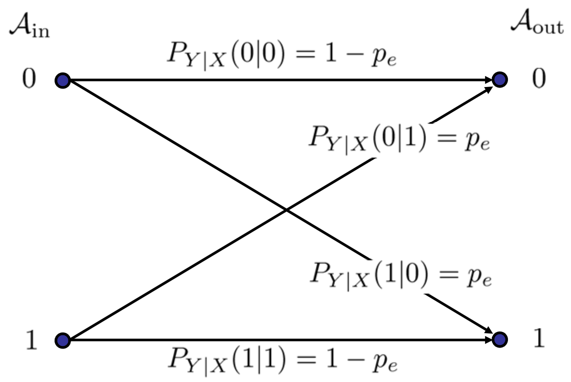
Symmetric hard decision DMC:

- symmetric transition probabilities
- $\mathcal{A}_{\text{in}} = \mathcal{A}_{\text{out}}$
- $P(Y|X)(y|x) = \begin{cases} 1 - p_e & \text{for} \quad x = y \\ \frac{p_e}{q-1} & \text{for} \quad x \neq y \end{cases}$, p_e : symbol error probability
- special case $q = 2$: **Binary symmetric channel (BSC)**

$$\boxed{P(Y|X)(y|x) = \begin{cases} 1 - p_e & \text{for} \quad y = x \\ p_e & \text{for} \quad y \neq x \end{cases}}$$

2.4 The discrete channel

Binary symmetric channel (BSC):



Example: Probability to receive 101 if 110 has been transmitted

$$P(101|110) = \underbrace{P(1|1)}_{1-p_e} \cdot \underbrace{P(0|1)}_{p_e} \cdot \underbrace{P(1|0)}_{p_e} = (1 - p_e) \cdot p_e^2$$

2.4 The discrete channel

Binary symmetric channel (BSC)

Important formulas:

1. Error event, P_{ee} , i.e., probability that within a sequence $\boldsymbol{x} = [x_0, x_1, ..., x_{N-1}]$ of length N at least one error occurs.

$$\boxed{P_{ee} = 1 - (1 - p_e)^n \approx n \cdot p_e \quad \text{for } n \cdot p_e \ll 1}$$

2. Probability that r specific bits are erroneous in a sequence of length n.

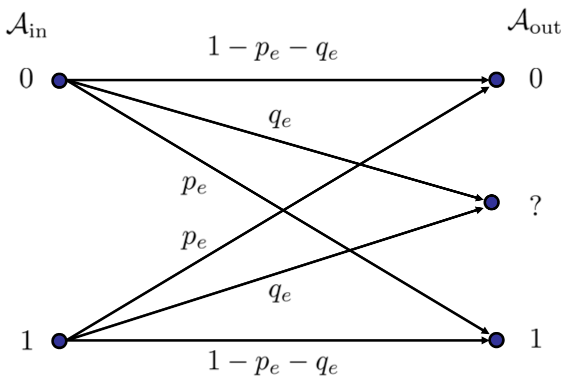
$$\boxed{P(\text{from } n \text{ bits are } r \text{ specific bits wrong}) = p_e^r \cdot (1 - p_e)^{n-r}}$$

3. Probability for r errors in a sequence of length n.

$$\boxed{P(\text{from } n \text{ bits are } r \text{ bits wrong}) = \underbrace{\binom{n}{r}}_{\text{combinations}} \cdot p_e^r \cdot (1 - p_e)^{n-r}}$$

2.4 The discrete channel

Binary symmetric erasure channel (BSEC):

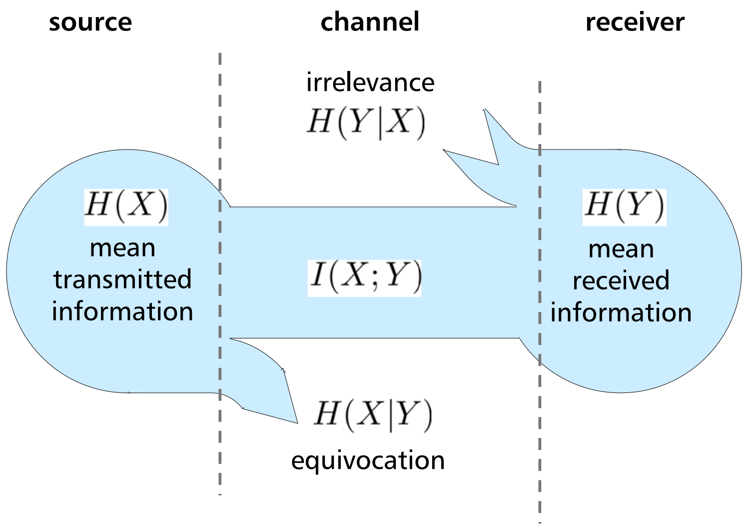


Simplest way of a soft-decision output

$$P(Y|X)(y|x) = \begin{cases} 1 - p_e - q_e & \text{for } \; y = x \\ q_e & \text{for } \; y = ? \\ p_e & \text{otherwise} \end{cases}$$

2.4 The discrete channel

Entropy diagram:



2.4 The discrete channel

Explaination:

- $H(X)$ source entropy, i.e., mean information emitted by the source
- $H(Y)$ mean information observed at the receiver
- $H(Y|X)$ irrelevance, i.e., the uncertainty over the output, if the input is known
- $H(X|Y)$ equivocation, i.e., the uncertainty over the input if the output is observed
- $I(X;Y)$ transinformation or mutual information, i.e., the information of the input which is contained in the output.

2.4 The discrete channel

Important formulas:

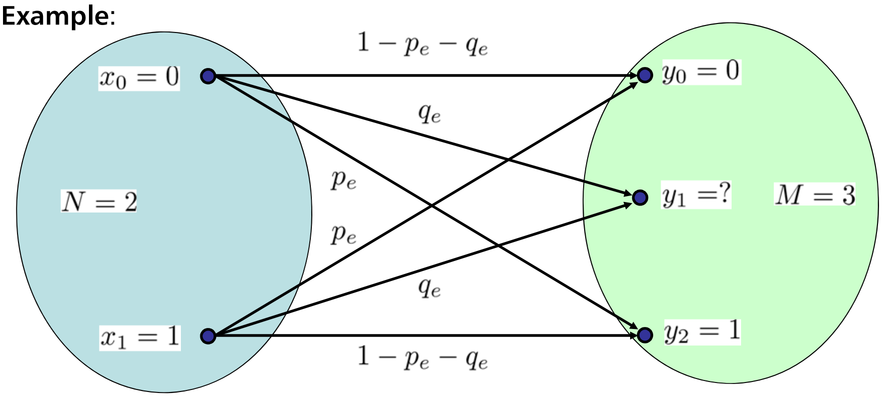
Input entropy

$$H(X) = -\sum_{i=0}^{N-1} p(x_i) \cdot \log_2(p(x_i))$$

output entropy

$$H(Y) = -\sum_{k=0}^{M-1} p(y_k) \cdot \log_2(p(y_k))$$

Example:



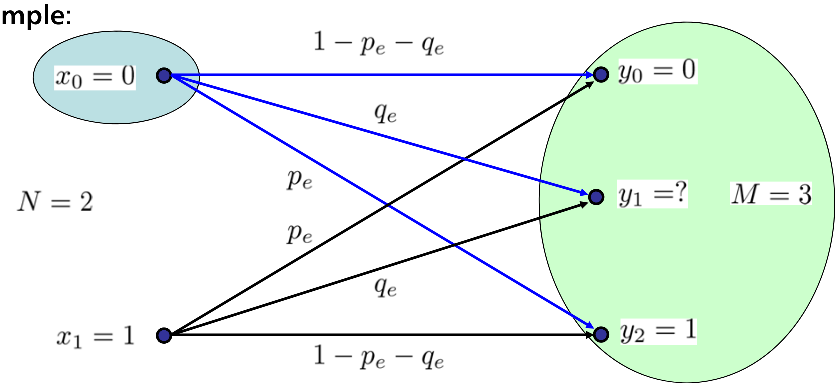
2.4 The discrete channel

irrelevance:

first consider only one input value x_i, $H(Y|X = x_i) = H(Y|x_i)$

$$H(Y|x_i) = -\sum_{k=0}^{M-1} p(y_k|x_i) \cdot \log_2(p(y_k|x_i))$$

Example:



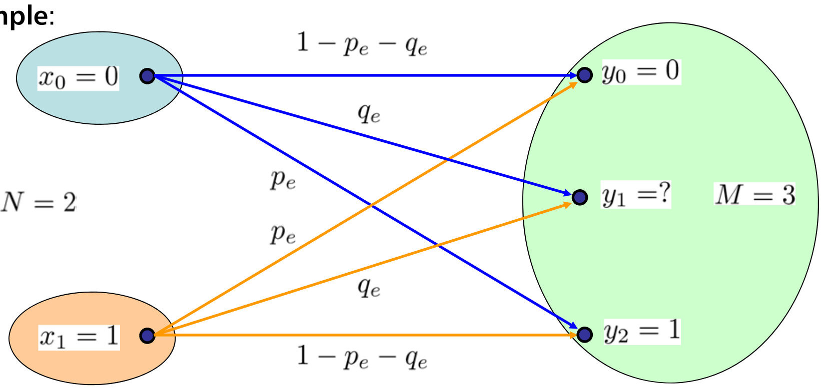
2.4 The discrete channel

irrelevance:

then take the mean for all possible input values

$$H(Y|X) = -\sum_{i=0}^{N-1} p(x_i) \sum_{k=0}^{M-1} p(y_k|x_i) \cdot \log_2(p(y_k|x_i))$$

Example:



2.4 The discrete channel

irrelevance:

$$H(Y|X) = -\sum_{i=0}^{N-1} p(x_i) \sum_{k=0}^{M-1} p(y_k|x_i) \cdot \log_2(p(y_k|x_i))$$

$$H(Y|X) = -\sum_{i=0}^{N-1} \sum_{k=0}^{M-1} \underbrace{p(x_i) \cdot p(y_k|x_i)}_{p(x_i, y_k)} \cdot \log_2(p(y_k|x_i))$$

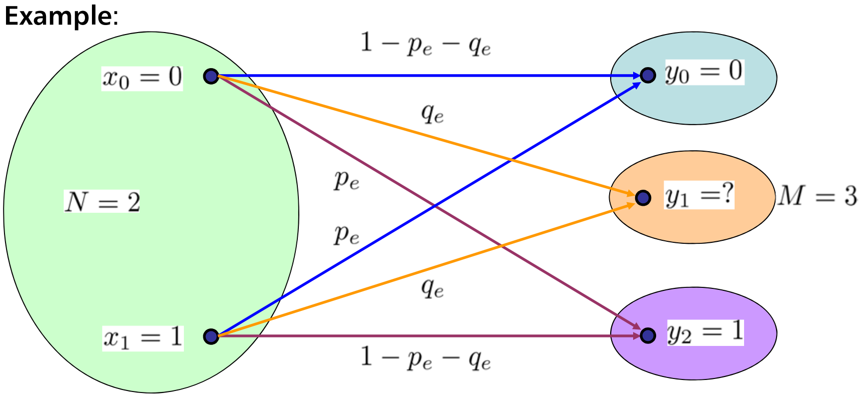
$$\boxed{H(Y|X) = -\sum_{i=0}^{N-1} \sum_{k=0}^{M-1} p(x_i, y_k) \cdot \log_2(p(y_k|x_i))}$$

2.4 The discrete channel

equivocation:

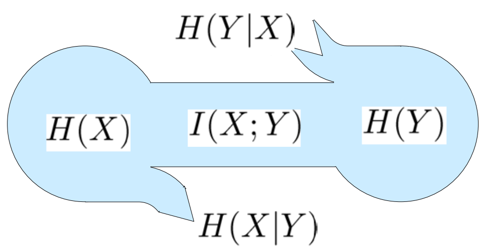
$$\boxed{H(X|Y) = -\sum_{i=0}^{N-1}\sum_{k=0}^{M-1} p(y_k, x_i) \cdot \log_2(p(x_i|y_k))}$$

Example:



2.4 The discrete channel

Mutual information:

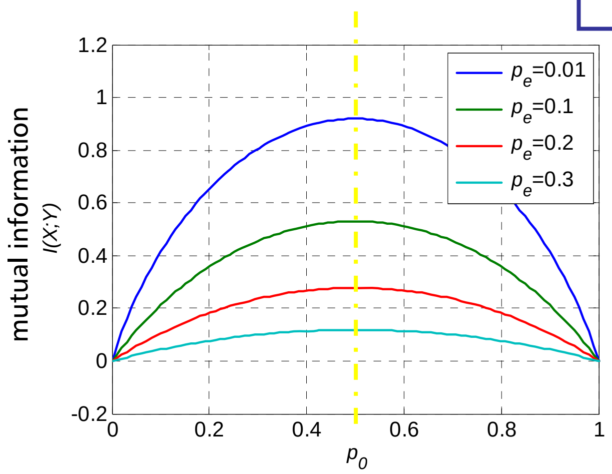


$$\boxed{I(X,Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)}$$

2.4 The discrete channel

Mutual information & channel capacity:

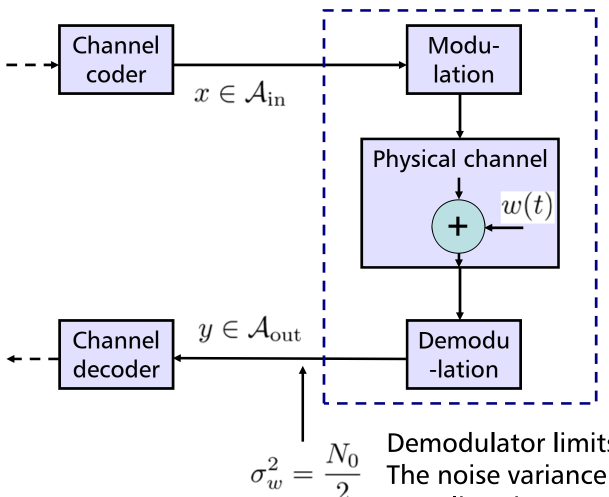
$$\boxed{C = \max_{p_0} \{I(X;Y)\}}$$



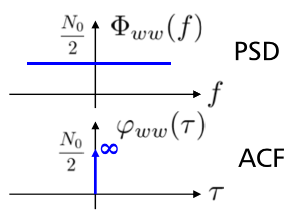
The maximum mutual information occurs for p_0=1/2, independent of p_e, i.e., for p_0=1/2 we can calculate the channel capacities for certain values of p_e.

2.5 The AWGN channel

AWGN (Additive White Gaussian Noise) Channel:



infinite bandwidth, therefore, infinite power: $\sigma_w^2 \rightarrow \infty$

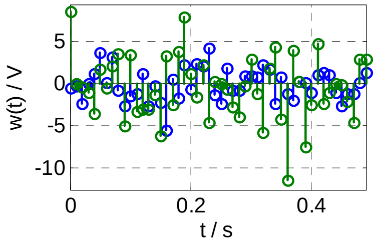


Demodulator limits bandwidth. The noise variance at the sampling times computes to $\frac{N_0}{2}$.

See "Communications Engineering" lecture for details.

2.5 The AWGN channel

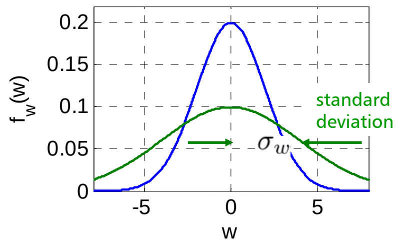
Noise example:



Sample realizations

$\sigma_w = 2$ V

$\sigma_w = 4$ V



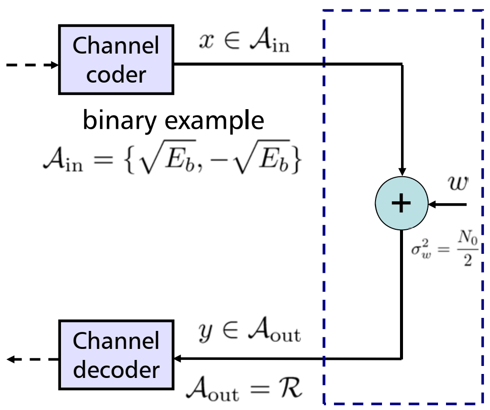
PDF of the amplitudes:

$$f_W(w) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \cdot \mathrm{e}^{-\frac{w^2}{2\sigma_w^2}}$$

variance

2.5 The AWGN channel

Simplified model:



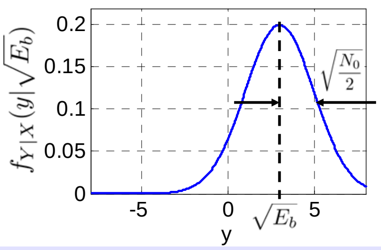
$$y = x + w$$

assume as statistically independent

conditional PDF

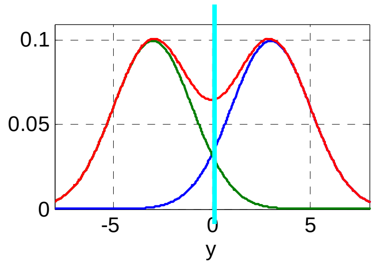
$$f_{Y|X}(y|x) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \cdot \mathrm{e}^{-\frac{(y-x)^2}{2\sigma_w^2}}$$

$$f_{Y|X}(y|\sqrt{E_b}) = \frac{1}{\sqrt{\pi N_0}} \cdot \mathrm{e}^{-\frac{(y-\sqrt{E_b})^2}{N_0}}$$



2.5 The AWGN channel

Error probability:



- $p(x=\sqrt{E_b})\cdot f_{Y|X}(y|\sqrt{E_b})$
- $p(x=-\sqrt{E_b})\cdot f_{Y|X}(y|-\sqrt{E_b})$
- $f_Y(y)$
- decision boundary

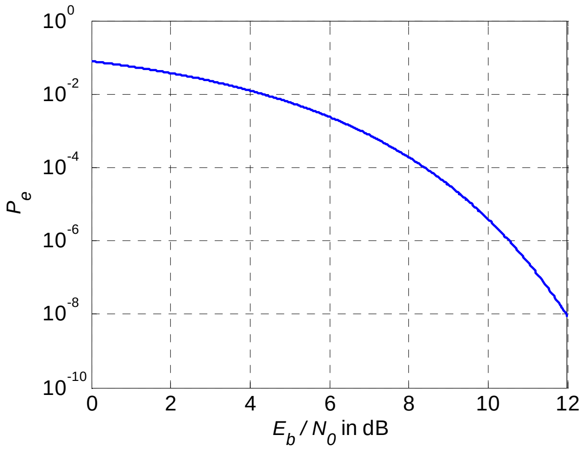
$$P_e = \underbrace{\frac{1}{2}}_{p(x=\sqrt{E_b})} \underbrace{P_{Y|X}(y<0|x=\sqrt{E_b})}_{\int_{-\infty}^{0} f_{Y|X}(y|x=\sqrt{E_b})\mathrm{d}x} + \underbrace{\frac{1}{2}}_{p(x=-\sqrt{E_b})} \underbrace{P_{Y|X}(y>0|x=-\sqrt{E_b})}_{\int_{0}^{\infty} f_{Y|X}(y|x=-\sqrt{E_b})\mathrm{d}x}$$

$$P_e = \int_0^\infty \frac{1}{\sqrt{\pi N_0}}\cdot \mathrm{e}^{-\frac{(y+\sqrt{E_b})^2}{N_0}}\mathrm{d}y$$

$$\boxed{P_e = \mathrm{Q}\left(\sqrt{\frac{2E_b}{N_0}}\right)}$$

2.5 The AWGN channel

AWGN Channel, binary input, BER performance (uncoded):



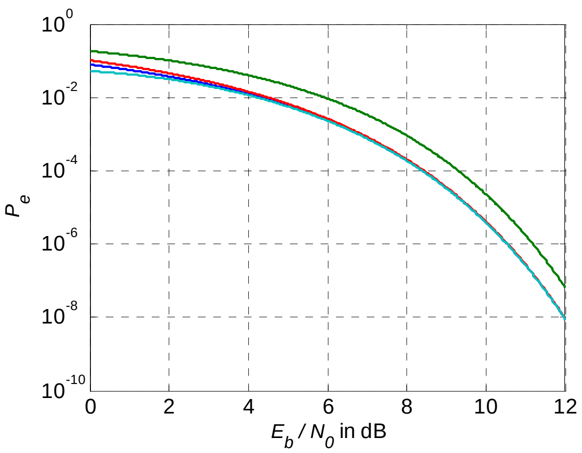
$$\boxed{P_e = \mathrm{Q}\left(\sqrt{\frac{2E_b}{N_0}}\right)}$$

$$\mathrm{Q}(x) = \frac{1}{2}\mathrm{erfc}\left(\frac{\mathrm{x}}{\sqrt{2}}\right)$$

$$\boxed{P_e = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)}$$

2.5 The AWGN channel

Bounds for the Q-function:



Exactly

$$P_e = \mathrm{Q}(x) \qquad x = \sqrt{\frac{2E_b}{N_0}}$$

Upper bounds

$$P_e \leq \frac{1}{2} \cdot \mathrm{e}^{-\frac{x^2}{2}}$$

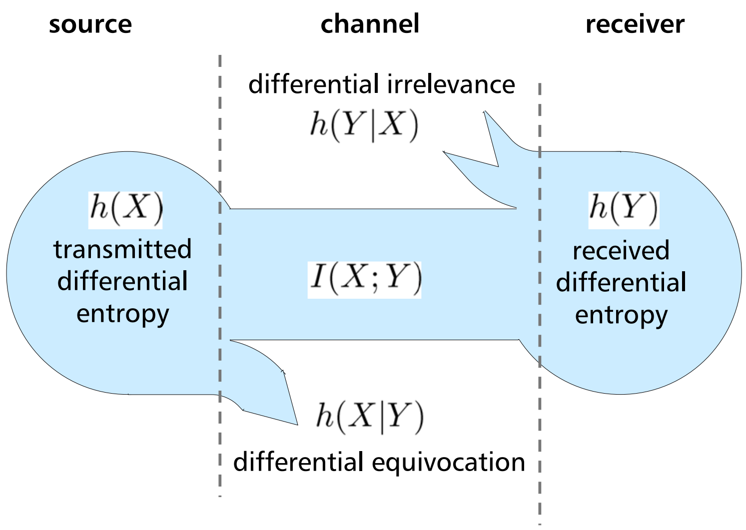
$$P_e \leq \frac{1}{x\sqrt{2\pi}} \cdot \mathrm{e}^{-\frac{x^2}{2}}$$

Lower bound

$$P_e \geq \left(1 - \frac{1}{x^2}\right) \frac{1}{x\sqrt{2\pi}} \cdot \mathrm{e}^{-\frac{x^2}{2}}$$

2.5 The AWGN channel

Entropy diagram for the continuous valued input and output:



2.5 The AWGN channel

Differential entropies:

$$h(X) = -\int_{-\infty}^{\infty} f_X(x)\log_2\left(f_X(x)\right)\mathrm{d}x \qquad h(Y) = -\int_{-\infty}^{\infty} f_X(y)\log_2\left(f_Y(y)\right)\mathrm{d}y$$

$$h(X|Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\log_2\left(\frac{1}{f_X(x|y)}\right)\mathrm{d}x\ \mathrm{d}y = E\left\{\log_2\left(\frac{1}{f_X(x|y)}\right)\right\}$$

$$h(Y|X) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\log_2\left(\frac{1}{f_X(y|x)}\right)\mathrm{d}x\ \mathrm{d}y = E\left\{\log_2\left(\frac{1}{f_X(y|x)}\right)\right\}$$

Mutual information:

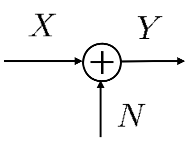
(i) $I(X;Y) = I(Y;X)$

(ii) $I(X;Y) \geq 0$

(iii) $I(X;Y) = h(X) - h(X|Y) = h(Y) - h(Y|X)$

2.5 The AWGN channel

AWGN Channel model:



X, Y, N: Random variables, containing the sampled values x, y, n of the input, output, and the noise process.

N: Gaussian distibuted with variance σ_N^2, $N \sim \mathcal{N}(0;\sigma_N^2)$

X: Input signal, power limited to $\mathrm{E}\left\{X^2\right\} = P$

Channel capacity:

$$\boxed{C = \max_{f_X(x)}\left\{I(X;Y) : \mathrm{E}\left\{X^2\right\} = P\right\}}$$

2.5 The AWGN channel

Mutual information:

$$I(X;Y) = h(Y) - h(Y|X)$$

X and N are statstically independent

$$Y = X + N$$

$$\Rightarrow h(Y|X) = h(N)$$

$$I(X;Y) = h(Y) - h(N)$$

maximization of $I(X;Y) \triangleq$ maximization of $h(Y)$,
since $h(N)$ does not depend on the p.d.f. of X

2.5 The AWGN channel

AWGN Channel capacity:

for $h(Y)$ to be maximum, Y has to be a Gaussian r.v.

$\Rightarrow$ since N is Gaussian, X must be Gaussian, too.

$\Rightarrow$ maximum is achieved if $X \sim \mathcal{N}(0;P)$

(i) variance of $Y : P + \sigma_N^2, \quad h(Y) = \frac{1}{2}\log_2(2\pi e(P + \sigma_N^2))$

(ii) $N \sim \mathcal{N}(0;\sigma_N^2), \quad h(N) = \frac{1}{2}\log_2(2\pi e\sigma_N^2)$

(iii) $C = h(Y) - h(N)$

2.5 The AWGN channel

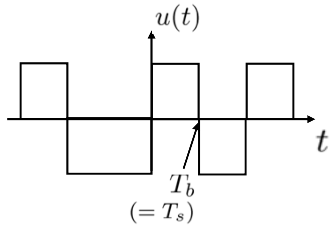
AWGN Channel capacity:

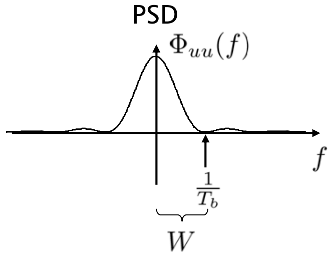
$$\boxed{C = \frac{1}{2}\log_2\left(1 + \frac{P}{\sigma_N^2}\right)}$$

in bits per transmission
or bits per channel use

AWGN Channel capacity as a function of the SNR and in bits per second?

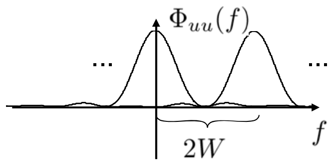
Example: Assume a transmission with a binary modulation scheme and bit rate $r_b = 1/T_b$ bit/s.



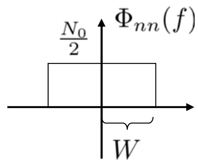


2.5 The AWGN channel

PSD of the sampled signal:



Band limited noise process:



Sampling at Nyquist rate of 2 *W*, i.e., we use the channel 2 *W* times per second

Noise power
$$\sigma_N^2 = 2 \cdot W \cdot \frac{N_0}{2} = N_0 W$$

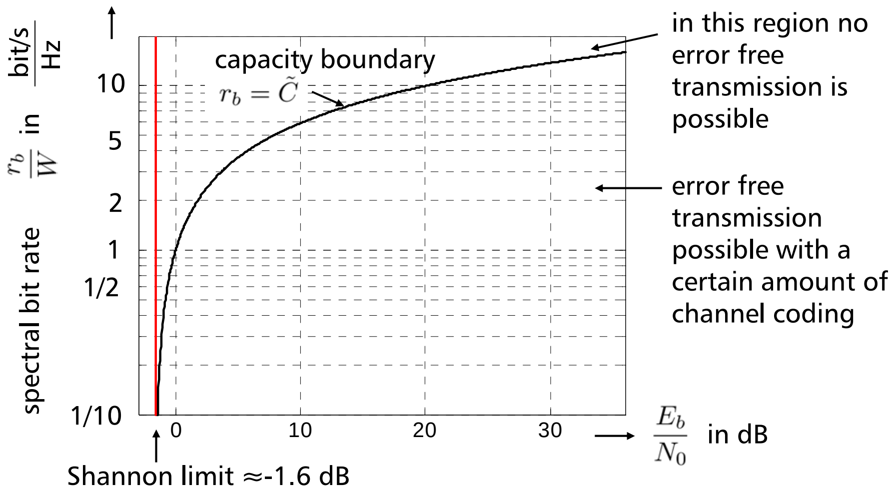
$$\tilde{C} = 2 \cdot W \cdot \frac{1}{2}\log_2\left(1 + \frac{P}{\sigma_N^2}\right)$$ in bits per second

channel uses per second

$$\boxed{\tilde{C} = W \cdot \log_2\left(1 + \frac{P}{N_0 W}\right) = W \cdot \log_2\left(1 + \frac{E_b}{N_0}\frac{r_b}{W}\right)}$$ in bits/second

2.5 The AWGN channel

Normalized capacity / spectral efficiency: $\boxed{\frac{\tilde{C}}{W} = \log_2\left(1 + \frac{E_b}{N_0}\frac{r_b}{W}\right) \quad \text{in} \quad \frac{\text{bit/s}}{\text{Hz}}}$



3 Channel Coding

Channel coding:

$\underbrace{\boldsymbol{u} = [u_0, \ldots, u_{k-1}]}_{\text{info word, length } k}$ $\xrightarrow{11,01,}$ $\boxed{\text{Channel coder}}$ $\xrightarrow{110,011,}$ $\underbrace{\boldsymbol{a} = [a_0, \ldots, a_{n-1}]}_{\text{code word, length } n}$

- add useful redundancy
- e.g., for FEC

Defines a (*n*,*k*) block code $\boxed{\text{code rate } R = k/n < 1}$

Example: (3,1) repetition code

$\boldsymbol{u} = [1] \rightarrow \underbrace{\boldsymbol{a} = [1\ 1\ 1]}_{\text{results in an increased data rate}}, \quad R = \frac{1}{3}$

results in an increased data rate $\longrightarrow$ $\underset{\text{code bit rate}}{r_c} = \underset{\text{info bit rate}}{r_b} \cdot \underset{\text{bandwidth expansion factor}}{\frac{1}{R}}$

3 Channel Coding

Code properties:

Systematic codes: Info words occur as a part of the code words

$$\boldsymbol{u} = [u_0, ..., u_{k-1}] \qquad \boldsymbol{a} = [a_0, ..., a_{n-1}]$$

$$\boxed{\begin{matrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix}} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \boxed{\begin{matrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix}} \boxed{\begin{matrix} 0 \\ 1 \\ 1 \\ 0 \end{matrix}}$$

Code space:
$\Gamma = \{000, 011, 101, 110\}$

Linear codes: The sum of two code words is again a codeword

$$\boldsymbol{a}_1, \boldsymbol{a}_2 \in \Gamma \rightarrow \boldsymbol{a}_1 + \boldsymbol{a}_2 = [a_1(0) + a_2(0), ..., a_1(n) + a_2(n)] \in \Gamma$$

$$\Gamma = \{000, 011, 101, 110\}$$

bit-by-bit modulo 2 addition without carry

$$\begin{array}{rccc} & 0 & 1 & 1 \\ + & 1 & 0 & 1 \\ \hline & 1 & 1 & 0 \end{array}$$

3 Channel Coding

Code properties:

Minimum Hamming distance:
A measure how different the most closely located code words are.
Example:

$$\begin{array}{c} d=2 \\ d=2 \\ \vdots \end{array} \quad \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{matrix} \quad \begin{array}{c} d=2 \\ d=2 \\ d=2 \end{array}$$

compare all combinations of code words

$$\boxed{d_{\min} = \min\{d(\boldsymbol{a}_i, \boldsymbol{a}_j), \ \forall\ \boldsymbol{a}_i, \boldsymbol{a}_j \in \Gamma, \ i \neq j\}}$$

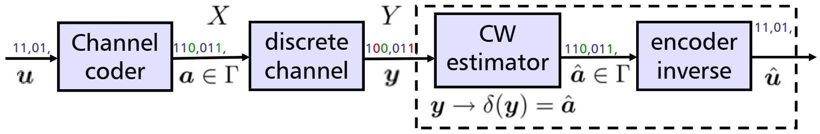
For linear codes the comparison simplifies to finding the code word with the lowest Hamming weight:

$$\boxed{d_{\min} = \min\{w_H(\boldsymbol{a}), \ \forall\ \boldsymbol{a} \in \Gamma\}}$$

3 Channel Coding

Maximum likelihood decoding (MLD):

Goal:
Minimum word error probability $\boxed{P_w = P(\hat{\boldsymbol{u}} \neq \boldsymbol{u}) = P(\hat{\boldsymbol{a}} \neq \boldsymbol{a}) \to \min}$



Code word estimator:
$\delta: \quad \boldsymbol{y} \to \delta(\boldsymbol{y}) = \hat{\boldsymbol{a}} \in \Gamma$
δ is the mapping from all 2^n possible received words to the 2^k possible code words in

Example: (7,4) Hamming code
$2^7 = 128$ possible received words
$2^4 = 16$ valid code words in Γ

3 Channel Coding

Decoding rule:
Assumption: equal apriori probabilities, i.e., all 2^k code words appear with probability $1/2^k$.

Probability for wrong detection if a certain cw $\boldsymbol{a}$ was transmitted:

$$P(\delta(\boldsymbol{y}) \neq \boldsymbol{a} \mid \boldsymbol{a} \text{ transmitted}) = \underbrace{\sum_{\substack{\boldsymbol{y} \\ \forall \delta(\boldsymbol{y}) \neq \boldsymbol{a}}} P(\boldsymbol{y} \text{ received} \mid \boldsymbol{a} \text{ transmitted}) = \sum_{\substack{\boldsymbol{y} \\ \forall \delta(\boldsymbol{y}) \neq \boldsymbol{a}}}}_{\text{Probability to receice a CW that yields an estimate } \neq \boldsymbol{a}} P_{Y|X}(\boldsymbol{y}|\boldsymbol{a})$$

Furthermore:

$$\sum_{\boldsymbol{a} \in \Gamma, \boldsymbol{y}} P_{Y|X}(\boldsymbol{y}|\boldsymbol{a}) = \sum_{\boldsymbol{a} \in \Gamma} \underbrace{P_{Y|X}(\text{any } \boldsymbol{y}|\boldsymbol{a})}_{=1} = \sum_{\boldsymbol{a} \in \Gamma} 1 = 2^k$$

3 Channel Coding

Example: (n=3,k=1) Repetition Code:

Assumption: equal apriori probabilities, i.e., each of the $2^k = 2^1 = 2$ code words (111,000) appear with probability $1/2^k = 1/2^1 = 1/2$

Probability for wrong detection if a certain cw $\boldsymbol{a}$ was transmitted:

$$P(\delta(\boldsymbol{y}) \neq \boldsymbol{a} \,|\, \boldsymbol{a} \text{ transmitted}) = \sum_{\substack{\boldsymbol{y} \\ \forall \delta(\boldsymbol{y}) \neq \boldsymbol{a}}} P(\boldsymbol{y} \text{ received} \,|\, \boldsymbol{a} \text{ transmitted}) = \sum_{\substack{\boldsymbol{y} \\ \forall \delta(\boldsymbol{y}) \neq \boldsymbol{a}}} P_{Y|X}(\boldsymbol{y}|\boldsymbol{a})$$

e.g., assume $\boldsymbol{a} = 111$ was transmitted over a BSC:

$$P(\delta(\boldsymbol{y}) \neq \boldsymbol{a} \mid \boldsymbol{a} = 111) = \sum_{\substack{\boldsymbol{y} \\ \forall \delta(\boldsymbol{y}) \neq \mathbf{111}}} P(\boldsymbol{y} \text{ received} \mid \boldsymbol{a} = 111) = p_e^3 + 3 \cdot p_e^2 \cdot (1 - p_e)$$

Transmitted, *a*	Possibly received, *y*	Decoded
111	000	000
	001	000
	010	000
	011	111
	100	000
	101	111
	110	111
	111	111

consider all received words that yield a wrong estimate

	Prob., e.g., if a BSC is considered
P(000|111)	$p_e\, p_e\, p_e$
P(001|111)	$p_e\, p_e\, (1-p_e)$
P(010|111)	$p_e\, (1-p_e)\, p_e$
P(100|111)	$(1-p_e)\, p_e\, p_e$

3 Channel Coding

Probability for a wrong detection (considering all possibly transmitted CWs now):

$$P(\delta(\boldsymbol{y}) \neq \boldsymbol{a}) = \underbrace{\sum_{\boldsymbol{a} \in \Gamma} \underbrace{P(\boldsymbol{a} \text{ transmitted})}_{=1/2^k = 2^{-k}}}_{\text{mean over all transmitted CWs}} \cdot \underbrace{P(\delta(\boldsymbol{y}) \neq \boldsymbol{a} \mid \boldsymbol{a} \text{ transmitted})}_{\sum_{\substack{\boldsymbol{y} \\ \forall \delta(\boldsymbol{y}) \neq \boldsymbol{a}}} P_{Y|X}(\boldsymbol{y}|\boldsymbol{a})}$$

$$= \sum_{\boldsymbol{a} \in \Gamma} 2^{-k} \cdot \sum_{\substack{\boldsymbol{y} \\ \forall \delta(\boldsymbol{y}) \neq \boldsymbol{a}}} P_{Y|X}(\boldsymbol{y}|\boldsymbol{a}) = 2^{-k} \cdot \underbrace{\sum_{\substack{\boldsymbol{a} \in \Gamma, \boldsymbol{y} \\ \forall \delta(\boldsymbol{y}) \neq \boldsymbol{a}}}}_{\text{wrong detection}} P_{Y|X}(\boldsymbol{y}|\boldsymbol{a})$$

combining the sums

$$= 2^{-k} \cdot \Bigg(\underbrace{\sum_{\boldsymbol{a} \in \Gamma, \boldsymbol{y}} P_{Y|X}(\boldsymbol{y}|\boldsymbol{a})}_{\substack{\text{any detection} \\ = 2^k}} - \underbrace{\sum_{\substack{\boldsymbol{a} \in \Gamma, \boldsymbol{y} \\ \forall \delta(\boldsymbol{y}) = \boldsymbol{a}}} P_{Y|X}(\boldsymbol{y}|\boldsymbol{a})}_{\text{correct detection}} \Bigg)$$

3 Channel Coding

Probability for wrong detection:

$$P_w = 1 - 2^{-k} \cdot \sum_{\substack{\boldsymbol{a}\in\Gamma,\boldsymbol{y} \\ \forall\delta(\boldsymbol{y})=\boldsymbol{a}}} P_{Y|X}(\boldsymbol{y}|\boldsymbol{a}) = 1 - 2^{-k} \cdot \sum_{\boldsymbol{y}} P_{Y|X}(\boldsymbol{y}|\underbrace{\delta(\boldsymbol{y})}_{\hat{\boldsymbol{a}}})$$

To minimize P_w choose $\delta(\boldsymbol{y}) = \hat{\boldsymbol{a}}$ for each received word such that $P_{Y|X}(\boldsymbol{y}|\hat{\boldsymbol{a}})$ gets maximized

$$\boxed{P_{Y|X}(\boldsymbol{y}|\hat{\boldsymbol{a}}) \geq P_{Y|X}(\boldsymbol{y}|\boldsymbol{b}) \quad \forall b \in \Gamma}$$

$P_{Y|X}(\boldsymbol{y}|\hat{\boldsymbol{a}})$ is maximized, if we choose a CW $\hat{\boldsymbol{a}}$ with the minimum distance d to the received word $\boldsymbol{y}$.

3 Channel Coding

MLD for hard decision DMC:
Find the CW with minimum Hamming distance.

$$\boxed{d_H(\boldsymbol{y},\hat{\boldsymbol{a}}) \leq d_H(\boldsymbol{y},\boldsymbol{b}) \quad \forall b \in \Gamma}$$

MLD for soft decision AWGN:

$$f_{Y|X}(\boldsymbol{y}|\hat{\boldsymbol{a}}) = \prod_{i=0}^{n-1} f_{Y|X}(y_i|\hat{a}_i) = \prod_{i=0}^{n-1} \frac{1}{\sqrt{\pi N_0}} \cdot \mathrm{e}^{-\frac{(y_i-\hat{a}_i)^2}{N_0}} = (\pi N_0)^{-\frac{n}{2}} \cdot \mathrm{e}^{-\frac{1}{N_0}\sum_{i=0}^{n-1}(y_i-\hat{a}_i)^2}$$

$$f_{Y|X}(\boldsymbol{y}|\hat{\boldsymbol{a}}) = (\pi N_0)^{-\frac{n}{2}} \cdot \mathrm{e}^{-\frac{1}{N_0}\underbrace{\|(\boldsymbol{y}-\hat{\boldsymbol{a}})\|^2}_{\text{Euklidean distance}}}$$

Find the CW with minimum Euklidean distance.

$$\boxed{\|\boldsymbol{y},\hat{\boldsymbol{a}}\| \leq \|\boldsymbol{y},\boldsymbol{b}\| \quad \forall b \in \Gamma}$$

3 Channel Coding

Coding gain:

Suitable measure: Bit error probability: $P_b = P(\hat{a}_i \neq a_i)$

(the bit error probability is considered only for the *k* info bits)

Code word error probability: $P_w = P(\hat{\boldsymbol{a}} \neq \boldsymbol{a})$

Example: Transmit 10 CWs and 1 bit error shall occur

$[\,\underbrace{1\,0\,1\,1}_{k\text{ info bits}}\,|\,1\,], [\,1\,1\,0\,0\,|\,0\,], \cdots$

1 bit wrong will yield 1 wrong code word $\Rightarrow$ $P_w = 1/10$

40 info bits have been transmitted $\Rightarrow$ $P_b = 1/40 = P_w/k$

As in general more than one error can occur in a code word, we can only approximate P_b

$$\boxed{\frac{1}{k} \cdot P_w \;\leq\; P_b \;\leq\; P_w}$$

3 Channel Coding

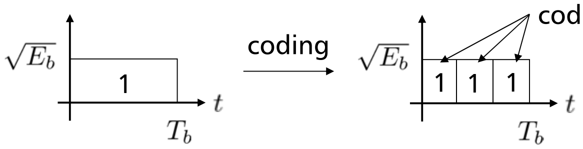
If we consider that a decoding error occurs only if $d_{\min}$ bits are wrong:

$$\boxed{P_b \approx \frac{d_{\min}}{k} \cdot P_w}$$

Comparison of codes considering the AWGN channel:

Energy per bit vs. energy per coded bit (for constant transmit power)

Example: (3,1) repetition code, $R = 1/3$



coded bits, energy E_c

$E_c = R \cdot E_b$

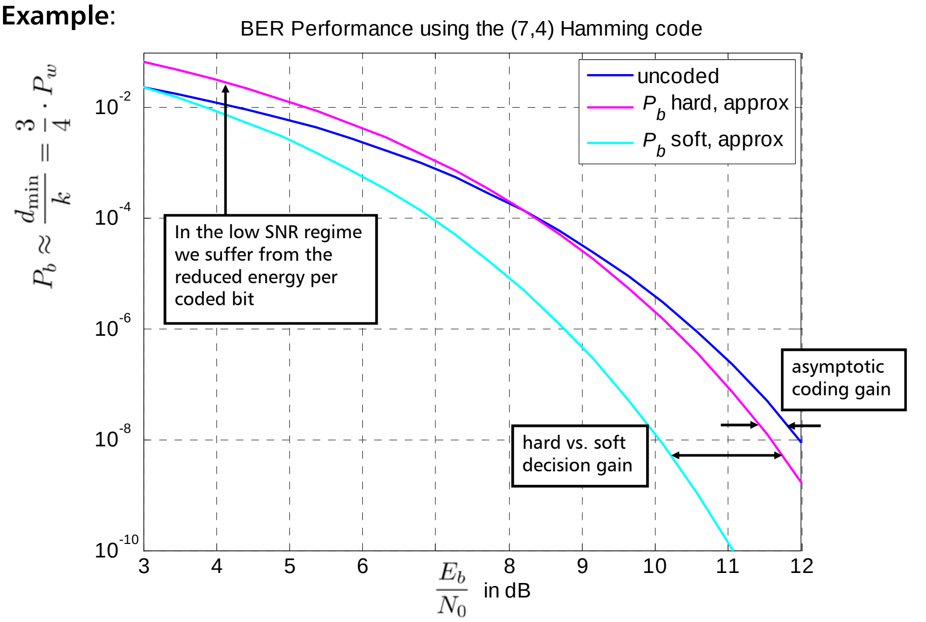
$r_c = 1/R \cdot r_b$

$W_c = 1/R \cdot W$

$$\frac{P}{\sigma_n^2} = \frac{E_b \cdot r_b}{N_0/2 \cdot 2\,W} = \frac{E_c}{N_0 \cdot R} \cdot \frac{R \cdot r_c}{R \cdot W_c} \qquad \boxed{\frac{E_c}{N_0} = R \cdot \frac{E_b}{N_0}}$$

3 Channel Coding

Example:



3 Channel Coding

Analytical calculation of the error probabilities:

Hard decision:

Example: (3,1) repetition code

Info word $\boldsymbol{u}$	code word $\boldsymbol{a}$	received word $\boldsymbol{y}$
1	1 1 1	0 0 0
		$\boxed{0\ 0\ 1}$
		$\boxed{0\ 1\ 0}$
		0 1 1
	$d_{\min} = 3$	$\boxed{1\ 0\ 0}$
		1 0 1
		1 1 0
		1 1 1
0	0 0 0	0 0 0
		⋮

$\binom{n}{r}$ combinations for r errors in a sequence of length n

$$\binom{n}{r} = \frac{n!}{r! \cdot (n-r)!}$$

1 combination for 3 errors

3 combinations for 1 error
will be corrected

3 combinations for 2 errors

$$\binom{3}{2} = \frac{1 \cdot 2 \cdot 3}{1 \cdot 2 \cdot 1} = 3$$

3 Channel Coding

$t = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor = \left\lfloor \frac{3-1}{1} \right\rfloor = 1$ error can be corrected

$$P_w = 3 \cdot p_e^2 \cdot (1-p_e)^2 + 1 \cdot p_e^3 \cdot (1-p_e)^0$$

3 combinations for 2 errors; 1 combination for 3 errors

general: $$\boxed{P_w = \sum_{r=t+1}^{n} \binom{n}{r} p_e^r \cdot (1-p_e)^{n-r}}$$

CW errors occur for more than t+1 wrong bits

combinations for r errors in a sequence of length n

probability for r errors

probability for n-r correct bits

3 Channel Coding

Approximation for small values of p_e

$$P_w \approx 3 \cdot p_e^2 \cdot \underbrace{(1-p_e)^2}_{\approx 1} + \underbrace{\cancel{1 \cdot p_e^3 \cdot (1-p_e)^0}}_{\approx 0}$$

only take the lowest power of p_e into account

general: $$\boxed{P_w \approx \binom{n}{t+1} p_e^{t+1}}$$ $$\boxed{P_b \approx \frac{d_{\min}}{k} \cdot P_w}$$

Example: (7,4) Hamming code, $d_{\min} = 3$ $\quad t = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor = \left\lfloor \frac{3-1}{1} \right\rfloor = 1$

$$P_w = \sum_{r=1+1}^{7} \binom{7}{r} p_e^r \cdot (1-p_e)^{7-r}$$

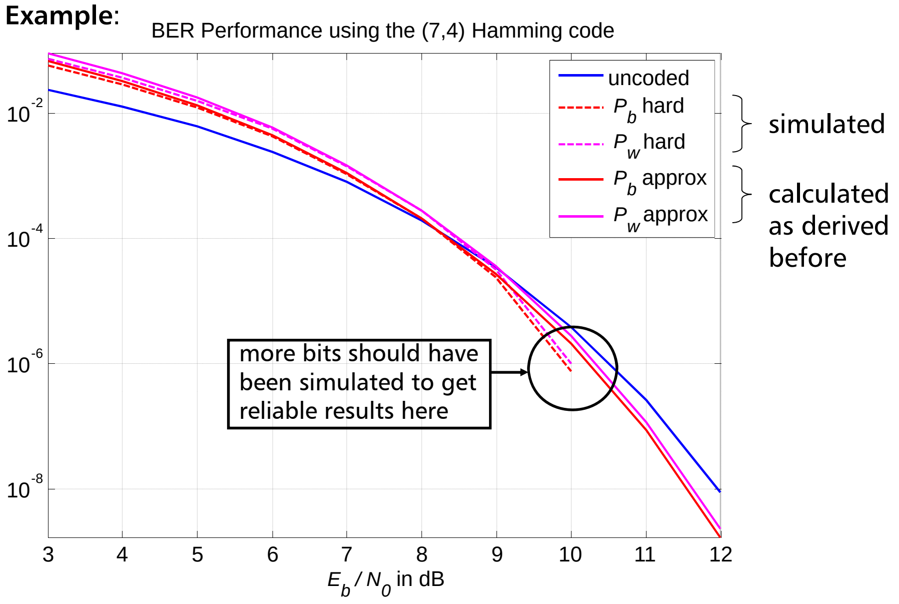
$$P_w = \binom{7}{2} p_e^2 \cdot \underbrace{(1-p_e)^{7-2}}_{\approx 1} + \cancel{\binom{7}{3} p_e^3 \cdot (1-p_e)^{7-3}} + \cancel{\binom{7}{4} p_e^4 \cdot (1-p_e)^{7-4}} + \ldots$$

$$P_w \approx \binom{7}{2} p_e^2 = 21 \cdot p_e^2$$

for a binary mod. scheme & AWGN channel $\rightarrow p_e = \mathrm{Q}\left(\sqrt{\frac{2E_c}{N_0}}\right) = \mathrm{Q}\left(\sqrt{\frac{2RE_b}{N_0}}\right)$

3 Channel Coding

Example:



3 Channel Coding

Asymptotic coding gain for hard decision decoding:

uncoded:

$$P_{b,u} = \mathrm{Q}\left(\sqrt{\frac{2E_{b1,u}}{N_0}}\right) \approx \text{const} \cdot \mathrm{e}^{-\frac{E_{b1,u}}{N_0}} \longleftarrow$$

good approximation for high SNR

coded:

$$P_{w,c} \approx \binom{n}{t+1} p_e^{t+1} \qquad P_{b,c} \approx \frac{d_{\min}}{k} \cdot P_{w,c}$$

$$P_{b,c} \approx \underbrace{\frac{d_{\min}}{k} \cdot \binom{n}{t+1}}_{\text{constant}} p_e^{t+1} \qquad p_e = \mathrm{Q}\left(\sqrt{\frac{2RE_{b2,u}}{N_0}}\right)$$

$$P_{b,c} \approx \text{const} \cdot \left[\mathrm{Q}\left(\sqrt{\frac{2RE_{b2,u}}{N_0}}\right)\right]^{t+1} \approx \text{const} \cdot \mathrm{e}^{-\frac{RE_{b2,u}}{N_0}(t+1)}$$

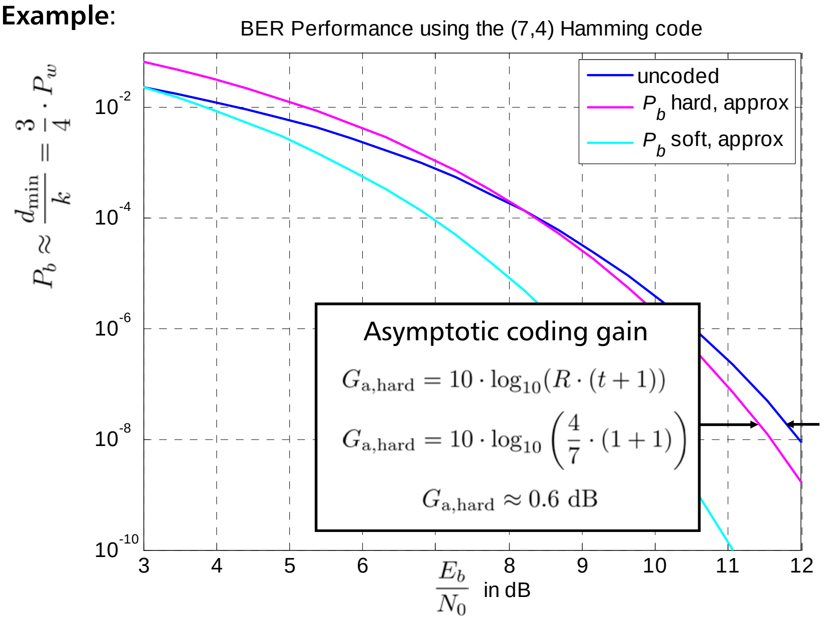
Assume constant BER and compare signal-to-noise ratios $P_{b,u} = P_{b,c}$

$$\text{const} \cdot \mathrm{e}^{-\frac{E_{b1,u}}{N_0}} \approx \text{const} \cdot \mathrm{e}^{-\frac{RE_{b2,u}}{N_0}(t+1)} \longrightarrow \frac{E_{b1,u}}{E_{b2,u}} = R \cdot (t+1)$$

$$\boxed{G_{\mathrm{a,hard}} = 10 \cdot \log_{10}\left(\frac{E_{b1,u}}{E_{b2,u}}\right) = 10 \cdot \log_{10}(R \cdot (t+1)) \quad \text{in dB}}$$

3 Channel Coding

Example:

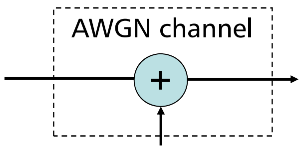


3 Channel Coding

Analytical calculation of the error probabilities:

Soft decision:

code word
$\boldsymbol{a} = [a_0, \ldots, a_{n-1}]$
$a_i \in \pm\sqrt{E_c}$



received word
$\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{n}$
$\boldsymbol{y} = [y_0, \ldots, y_{n-1}]$

$$\underbrace{\boldsymbol{n} = [n_0, n_1, \ldots, n_{n-1}]}$$

Noise vector: i.i.d. $\in \mathcal{N}\{0, N_0/2\}$

Example: (3,2) Parity check code

$$\Gamma = \{\boldsymbol{a}_0, \ldots \boldsymbol{a}_{2^k}\}$$

0 0 0	$\boldsymbol{a}_0 = [\ -1 \ \ -1 \ \ -1\ ]$		
0 1 1 $\rightarrow$	$\boxed{\boldsymbol{a}_1 = [\ -1 \ \ 1 \ \ 1\ ]}$	$\rightarrow$ + $\rightarrow$	$\boldsymbol{y} = [-1.7 \ \ 0.7 \ \ 1.2\ ]$
1 0 1	$\boldsymbol{a}_2 = [\ 1 \ \ -1 \ \ 1\ ]$	$\uparrow$	
1 1 0	$\boldsymbol{a}_3 = [\ 1 \ \ 1 \ \ -1\ ]$	$\boldsymbol{n} = [-0.7 \ \ -0.3 \ \ 0.2\ ]$	

$E_c = 1$

3 Channel Coding

Example continued

$$\|\boldsymbol{y}, \boldsymbol{b}\| = \sqrt{[-1.7-(-1)]^2 + [0.7-(-1)]^2 + [1.2-(-1)]^2}$$

$$\begin{array}{lll}
& \boldsymbol{a}_0 = [\ -1 \quad -1 \quad -1\ ] & 2.9 \\
\boxed{\hat{\boldsymbol{a}}} & \boxed{\boldsymbol{a}_1 = [\ -1 \quad 1 \quad 1\ ]} & \boxed{0.8} \\
& \boldsymbol{a}_2 = [\ 1 \quad -1 \quad 1\ ] & 3.2 \\
& \boldsymbol{a}_3 = [\ 1 \quad 1 \quad -1\ ] & 3.5
\end{array}
\qquad \boldsymbol{y} = [-1.7 \quad 0.7 \quad 1.2\ ]$$

ML decoding rule, derived before

$$\|\boldsymbol{y}, \hat{\boldsymbol{a}}\| \leq \|\boldsymbol{y}, \boldsymbol{b}\| \quad \forall b \in \Gamma$$

Pairwise error probability: Assume $\boldsymbol{a}_i$ has been transmitted. What is the probability that the decoder decides for a different CW $\boldsymbol{a}_j$?

$$P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = P(\|\boldsymbol{y} - \boldsymbol{a}_j\| \leq \|\boldsymbol{y} - \boldsymbol{a}_i\|)$$

The decoder will decide for $\boldsymbol{a}_j$ if the received word $\boldsymbol{y}$ has a smaller Euklidean distance to $\boldsymbol{a}_j$ as compared to $\boldsymbol{a}_i$.

$$P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = P(\|\boldsymbol{y} - \boldsymbol{a}_j\|^2 \leq \|\boldsymbol{y} - \boldsymbol{a}_i\|^2) \qquad \boldsymbol{y} = \boldsymbol{a}_i + \boldsymbol{n}$$

$$= P(\|\boldsymbol{a}_i + \boldsymbol{n} - \boldsymbol{a}_j\|^2 \leq \|\cancel{\boldsymbol{a}_i} + \boldsymbol{n} - \cancel{\boldsymbol{a}_i}\|^2)$$

$$= P(\|\boldsymbol{n} + (\boldsymbol{a}_i - \boldsymbol{a}_j)\|^2 \leq \|\boldsymbol{n}\|^2)$$

next: Evaluate the norm by summing the squared components

3 Channel Coding

$$P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = P\left(\sum_{r=0}^{n-1}\left[n_r^2 + 2n_r(a_{i,r} - a_{j,r}) + (a_{i,r} - a_{j,r})^2\right] \leq \sum_{r=0}^{n-1} n_r^2\right)$$

$$= P\left(\cancel{\sum_{r=0}^{n-1} n_r^2} + 2\sum_{r=0}^{n-1} n_r(a_{i,r} - a_{j,r}) + \sum_{r=0}^{n-1}(a_{i,r} - a_{j,r})^2 \leq \cancel{\sum_{r=0}^{n-1} n_r^2}\right)$$

$$= P\left(2\sum_{r=0}^{n-1} n_r(a_{i,r} - a_{j,r}) + \sum_{r=0}^{n-1}(a_{i,r} - a_{j,r})^2 \leq 0\right)$$

$$= P\left(\sum_{r=0}^{n-1} n_r(a_{i,r} - a_{j,r}) \leq -\frac{1}{2}\underbrace{\sum_{r=0}^{n-1}(a_{i,r} - a_{j,r})^2}\right)$$

$$a_{i,r}, a_{j,r} \in \pm\sqrt{E_c} \qquad a_{i,r} \neq a_{j,r} \to (a_{i,r} - a_{j,r})^2 = (2\sqrt{E_c})^2$$

$$a_{i,r} = a_{j,r} \to (a_{i,r} - a_{j,r})^2 = 0$$

For the whole CW we have $d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)$ different bits

$$\sum_{r=0}^{n-1}(a_{i,r} - a_{j,r})^2 = d_H(\boldsymbol{a}_i, \boldsymbol{a}_j) \cdot 4 \cdot E_c$$

3 Channel Coding

$$P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = P\left(\boxed{\sum_{r=0}^{n-1} n_r \underbrace{(a_{i,r} - a_{j,r})}}\le -2 \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j) \cdot E_c\right)$$

scales standard deviation

Gaussian rv with standard deviation $\sigma_n = \sqrt{\frac{N_0}{2}}$

sum of Gaussian rvs: The variance of the sum will be the sum of the individual variances.

$$\sigma^2 = \sum_{r=0}^{n-1} \underbrace{\left(\underbrace{\sqrt{\frac{N_0}{2}}(a_i,r - a_j,r)}_{\text{std. dev.}}\right)^2}_{\text{variance}} = \frac{N_0}{2} \cdot \underbrace{\sum_{r=0}^{n-1}(a_i,r - a_j,r)^2}_{d_H(\boldsymbol{a}_i,\boldsymbol{a}_j)\cdot 4 \cdot E_c}$$

Gaussian rv with zero mean and variance $\sigma^2 = 2 \cdot N_0 \cdot E_c \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)$

3 Channel Coding

$$P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = P\left(-\sum_{r=0}^{n-1} n_r(a_{i,r} - a_{j,r}) \ge 2 \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j) \cdot E_c\right)$$

multiplied with -1

Question: What is the probability that our Gaussian r.v. becomes larger than a certain value?

Answer: Integral over remaining part of the Gaussian PDF, e.g., expressed via the Q-function.

Q-Function: $\mathrm{Q}(\alpha) = P\left(\underbrace{\frac{x-\mu}{\sigma}} > \alpha\right)$ $\quad x \in \mathcal{N}(\mu, \sigma^2)$

normalized Gaussian rv $\in \mathcal{N}(0,1)$

Probability that a normalized Gaussian r.v. becomes larger than certain value α.

$$\mathrm{Q}(\alpha) = \frac{1}{2\pi}\int_{\alpha}^{\infty} \mathrm{e}^{-\frac{\epsilon^2}{2}} \mathrm{d}\epsilon$$

3 Channel Coding

$$P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = P\left(-\sum_{r=0}^{n-1} n_r(a_{i,r} - a_{j,r}) \geq 2 \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j) \cdot E_c\right)$$

$$\sigma = \sqrt{2 \cdot N_0 \cdot E_c \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)}$$

$$P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = P\left(\underbrace{-\frac{\sum_{r=0}^{n-1} n_r(a_{i,r} - a_{j,r})}{\sqrt{2 \cdot N_0 \cdot E_c \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)}}}_{\text{normalized Gaussian r.v.}} \geq \underbrace{\frac{2 \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j) \cdot E_c)}{\sqrt{2 \cdot N_0 \cdot E_c \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)}}}_{\alpha}\right)$$

$$P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = \mathrm{Q}\left(\frac{2 \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j) \cdot E_c)}{\sqrt{2 \cdot N_0 \cdot E_c \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)}}\right)$$

Pairwise error probability:

$$\boxed{P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = \mathrm{Q}\left(\sqrt{2 \cdot d_H(\boldsymbol{a}_i, \boldsymbol{a}_j) \cdot \frac{E_c}{N_0}}\right)}$$

3 Channel Coding

Example continued: e.g., for $E_b/N_0 = 5 \hat{=} 7\text{dB}$

transmitted

$\boldsymbol{a}_1 = [-1 \quad 1 \quad 1]$

$\boldsymbol{a}_i = \boldsymbol{a}_1$

$d_H(\boldsymbol{a}_1, \boldsymbol{a}_j)$		$P(\boldsymbol{a}_i \to \boldsymbol{a}_j)$
2	$\boldsymbol{a}_0 = [\ -1 \quad -1 \quad -1\ ]$	$3.9 \cdot 10^{-6}$
0	$\boldsymbol{a}_1 = [\ -1 \quad 1 \quad 1\ ]$	–
2	$\boldsymbol{a}_2 = [\ 1 \quad -1 \quad 1\ ]$	$3.9 \cdot 10^{-6}$
2	$\boldsymbol{a}_3 = [\ 1 \quad 1 \quad -1\ ]$	$3.9 \cdot 10^{-6}$

$d_{\min} = 2$

Number of CW within distance $d_{\min}$ → $A_{d_{\min}} = 3$

For $d_H(\boldsymbol{a}_1, \boldsymbol{a}_j) = 3$ we would get $P(\boldsymbol{a}_i \to \boldsymbol{a}_j) = 2.2 \cdot 10^{-8}$

The CWs with the minimum Hamming distance to the transmitted CW dominate the **CW error probability**

$$P_w \approx \underbrace{\sum_{i=0}^{2^k-1} p(\boldsymbol{a}_i)}_{\text{Mean over the transmitted CWs}} \cdot A_{d_{\min}} \cdot P(\boldsymbol{a}_i \to \boldsymbol{a}_j) \quad \forall i \neq j$$

3 Channel Coding

$$P_w \approx A_{d_{\min}} \cdot P(\boldsymbol{a}_i \rightarrow \boldsymbol{a}_j) \quad \forall i \neq j$$

$$\boxed{P_w \approx A_{d_{\min}} \cdot \mathrm{Q}\left(\sqrt{2 \cdot d_{\min} \cdot \frac{E_c}{N_0}}\right)} \qquad E_c = R \cdot E_b$$

$$1 \cdot \mathrm{Q}\left(\sqrt{2 \cdot d_{\min} \cdot \frac{E_c}{N_0}}\right) \leq A_{d_{\min}} \cdot \mathrm{Q}\left(\sqrt{2 \cdot d_{\min} \cdot \frac{E_c}{N_0}}\right) \leq (2^k - 1) \cdot \mathrm{Q}\left(\sqrt{2 \cdot d_{\min} \cdot \frac{E_c}{N_0}}\right)$$

Best case: only one CW within $d_{\min}$

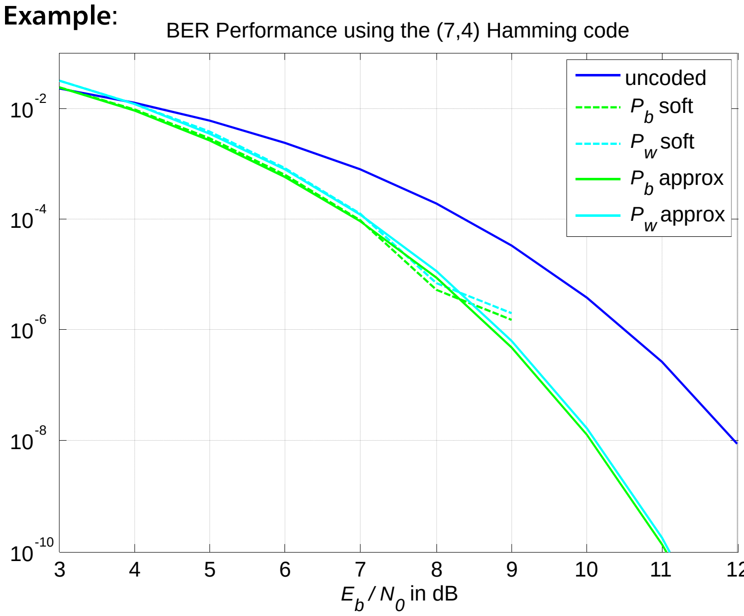
worst case: all CWs within $d_{\min}$

For high SNR or if $A_{d_{\min}}$ is unkown

$$\boxed{P_w \geq \mathrm{Q}\left(\sqrt{2 \cdot d_{\min} \cdot R \cdot \frac{E_b}{N_0}}\right)} \qquad \boxed{P_b \approx \frac{d_{\min}}{k} \cdot P_w}$$

3 Channel Coding

Example:



simulated

calculated as derived before using $A_{d_{\min}}$

3 Channel Coding

Asymptotic coding gain for soft decision decoding:

Derivation analog to the hard decision case

<u>uncoded:</u>

$$P_{b1} \approx \text{const} \cdot \mathrm{e}^{-\frac{E_{b1}}{N_0}} \quad \longleftarrow$$ good approximation for high SNR

<u>coded:</u>

$$P_{b2} \approx \text{const} \cdot \mathrm{e}^{-d_{\min} \cdot R \cdot \frac{E_{b2}}{N_0}}$$

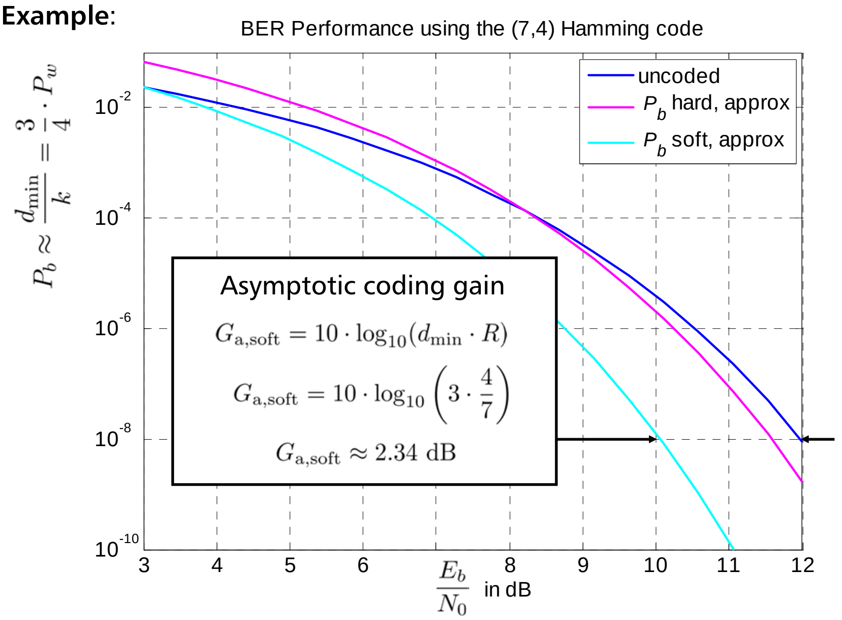
Assume constant BER and compare signal-to-noise ratios $P_{b1} = P_{b2}$

$$\cancel{\text{const}} \cdot \mathrm{e}^{-\frac{E_{b1}}{N_0}} \approx \cancel{\text{const}} \cdot \mathrm{e}^{-d_{\min} \cdot R \cdot \frac{E_{b2}}{N_0}} \longrightarrow \frac{E_{b1}}{E_{b2}} = d_{\min} \cdot R$$

$$\boxed{G_{\mathrm{a,soft}} = 10 \cdot \log_{10}\left(\frac{E_{b1}}{E_{b2}}\right) = 10 \cdot \log_{10}(d_{\min} \cdot R) \quad \text{in dB}}$$

3 Channel Coding

Example:



3 Channel Coding

Matrix representation of block codes:

Example: (7,4) Hamming code

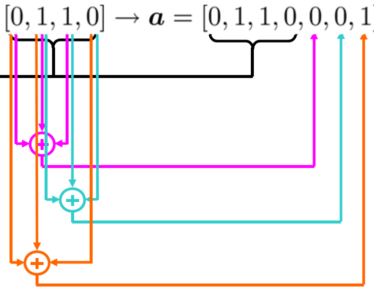
Encoding equation:

$$\boldsymbol{u} = [u_0, ..., u_3] \rightarrow \boldsymbol{a} = [a_0, ..., a_6]$$

$$\boldsymbol{u} = [0, 1, 1, 0] \rightarrow \boldsymbol{a} = [0, 1, 1, 0, 0, 0, 1]$$

$$\left.\begin{aligned} a_0 &= u_0 \\ a_1 &= u_1 \\ a_2 &= u_2 \\ a_3 &= u_3 \end{aligned}\right\} \text{systematic code}$$

$$\begin{aligned} a_4 &= u_0 + u_1 + u_2 \\ a_5 &= u_1 + u_2 + u_3 \\ a_6 &= u_0 + u_1 + u_3 \end{aligned}$$



bitwise modulo 2
sum without carry

3 Channel Coding

Introducing the **generator matrix** $\boldsymbol{G}$ we can express the encoding process as matrix-vector product.

$$\underbrace{\boldsymbol{a}}_{1\times 7} = \underbrace{\boldsymbol{u}}_{1\times 4} \cdot \underbrace{\boldsymbol{G}}_{4\times 7}$$

multiply and sum

$$\underbrace{[0\ 1\ 1\ 0]}_{\boldsymbol{u}} \quad \boldsymbol{G} = \left[\begin{array}{cccc:ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

$$\underbrace{\left[\begin{array}{cccc:ccc} 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]}_{\boldsymbol{a}}$$

$$= 0\cdot 1 + 1\cdot 1 + 1\cdot 1 + 0\cdot 0$$

The identity matrix is responsible that the code becomes a systematic code. It just copies the info word into the CW

Parity matrix $\boldsymbol{P}$

3 Channel Coding

General: For a (n,k) block code:

2^k info words $\underbrace{\boldsymbol{u}_i}_{1\times k} = [u_0, ..., u_{n-1}], \quad i = 0, ..., 2^k - 1$

2^k code words $\underbrace{\boldsymbol{a}_i}_{1\times n} = [a_0, ..., u_{n-1}], \quad i = 0, ..., 2^k - 1$

Encoding:

$$\boxed{\underbrace{\boldsymbol{a}_i}_{1\times n} = \underbrace{\boldsymbol{u}_i}_{1\times k} \cdot \underbrace{\boldsymbol{G}}_{k\times n}}$$

For systematic codes:

$$\boxed{\boldsymbol{G} = \left[\boldsymbol{I}_k \vdots \boldsymbol{P} \right]}$$

Set of code words:

$$\Gamma = \{\boldsymbol{a}_i\} = \{\boldsymbol{u}_i \cdot \boldsymbol{G}\}, \quad i = 0, ..., 2^k - 1$$

3 Channel Coding

Properties of the generator matrix G

- the rows of G shall be linear independent
- the rows of G are code words of Γ
- the row space is the number of linear independent rows
- the column space is the number of linear independent rows
- row space and column space are equivalent, i.e., the rank of the matrix
- as G has more columns than rows, the columns must be linear dependent

Example: (7,4) Hamming code

$$\boldsymbol{G} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}}_{n=7} \quad k = 4$$

easy to see:

- the rows are linear independent
- the last 3 columns can be written as linear comb. of the first 4 columns
- rank 4

3 Channel Coding

Properties of the generator matrix G

- rows can be exchanged without changing the code
- multiplication of rows with a scalar doesn't change the code
- sum of a scaled row with another row doesn't change the code
- exchanging columns will change the set of codewords but the weight distribution and the minimum Hamming distance will be the same

yields the same code:

$$G_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

each Generator matrix can be brought to the row echelon form, i.e., a systematic encoder

$$G = \left[I_k \,\vdots\, P \right]$$

3 Channel Coding

Properties of the generator matrix G

- as the all zero word is a valid code word, and the rows of G are also valid code words, the minimum Hamming distance must be less or equal the minimum weight of the rows.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad \rightarrow d_{\min} \leq 3$$

Parity check matrix H

The code can be also defined via the parity check matrix

$$\Gamma = \{ \, a \mid a \cdot H^T = 0 \, \}$$

$$0 = a \cdot H^T = u \cdot G \cdot H^T \;\rightarrow \boxed{G \cdot H^T = 0}$$

3 Channel Coding

Parity check matrix $\boldsymbol{H}$

If $\boldsymbol{G}$ is a systematic generator matrix, e.g.,

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & \vdots & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & \vdots & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & \vdots & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & \vdots & 0 & 1 & 1 \end{bmatrix} = \left[\boldsymbol{I}_k \vdots \boldsymbol{P} \right]$$

then

$$\boldsymbol{H} = \left[-\boldsymbol{P}^T \vdots \boldsymbol{I}_{n-k} \right] = \begin{bmatrix} 1 & 1 & 1 & 0 & \vdots & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & \vdots & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & \vdots & 0 & 0 & 1 \end{bmatrix} \quad n-k=3$$

$$n = 7$$

$\boldsymbol{H}$ can be used to check whether a received CW is a valid CW, or to determine what is wrong with the received CW (syndrom)

3 Channel Coding

Decoding:

ML decoding is trivial but computationally very complex as the received CW has to be compared with all possible CWs. Impractical for larger code sets.

Therefore, simplified decoding methods shall be considered.

Syndrom decoding using Standard Arrays (or Slepian arrays)

Assume an (n,k) code with the parity check matrix $\boldsymbol{H}$

The **Syndrom** for a received CW $\boldsymbol{y}$ is defined as:

$$\boxed{\underbrace{\boldsymbol{s}}_{1\times n-k} = \underbrace{\boldsymbol{y}}_{1\times n} \cdot \underbrace{\boldsymbol{H}^T}_{n\times n-k}}$$

with $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}$

valid CW + error word, error pattern

$$\boldsymbol{s} = (\boldsymbol{a}+\boldsymbol{e}) \cdot \boldsymbol{H}^T = \underbrace{\boldsymbol{a}\boldsymbol{H}^T}_{=\boldsymbol{0}} + \boldsymbol{e}\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T$$

For a valid received CW the syndrom will be **0**.

Otherwise the Syndrom only depends on the error pattern.

3 Channel Coding

As we get 2^k valid codewords and 2^n possibly received words there must be $2^n - 2^k$ error patterns. The syndrom is only of size $n-k$, therefore the syndroms are not unique.

E.g., (7,4) Hamming Code: 16 valid CWs, 128 possibly received CWs, 112 error patterns, $2^{(n-k)}=8$ syndroms.

Let the different syndroms be $\boldsymbol{s}_\mu, \mu = 0, ..., 2^{n-k}$.

For each syndrom we'll get a whole set of error patterns $\mathcal{M}_\mu$ (cosets), that yield this syndrom.

$$\mathcal{M}_\mu = \{\ \boldsymbol{e} \mid \boldsymbol{e}\boldsymbol{H}^T = \boldsymbol{s}_\mu\}$$

Let $\boldsymbol{e}, \boldsymbol{e}' \in \mathcal{M}_\mu$, i.e., they'll yield the same Syndrom $\boldsymbol{s}_\mu$

$$\boldsymbol{e}\boldsymbol{H}^T = \boldsymbol{e}'\boldsymbol{H}^T \ \rightarrow \ \underbrace{(\boldsymbol{e}' - \boldsymbol{e})}\cdot \boldsymbol{H}^T = \boldsymbol{0}$$

The difference of two error patterns in $\mathcal{M}_\mu$ must be a valid CW then.

3 Channel Coding

The set $\mathcal{M}_\mu$ can be expressed as one element $\boldsymbol{e} \in \mathcal{M}_\mu$ plus the code set Γ.

$$\boldsymbol{e} + \Gamma = \{\boldsymbol{e} + \boldsymbol{a} \mid \boldsymbol{a} \in \Gamma\} = \mathcal{M}_\mu$$

Within $\mathcal{M}_\mu$ each $\boldsymbol{e}$ can be chosen as **coset leader** $\boldsymbol{e}_\mu$ to calculate the rest of the coset.

$$\mathcal{M}_\mu = \boldsymbol{e}_\mu + \Gamma$$

The coset leader is chosen with respect to the minimum Hamming weight

$$w_H(\boldsymbol{e}_\mu) \leq w_H(\boldsymbol{e}), \quad \forall \boldsymbol{e} \in \mathcal{M}_\mu$$

Example: (5,2) Code

$$\Gamma = \{00000, 10110, 01011, 11101\}$$

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \qquad \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

3 Channel Coding

Syndrom 0 → valid CWs
$\Gamma = \{00000, 10110, 01011, 11101\}$

coset leader | coset | syndrom

μ	e_μ	$\mathcal{M}_\mu$				s_μ
0	00000	00000	10110	01011	11101	000
1	00001	00001	10111	01010	11100	001
2	00010	00010	10100	01001	11111	010
3	00100	00100	10010	01111	11001	100
4	01000	01000	11110	00011	10101	011
5	10000	10000	00110	11011	01101	110
6	11000	11000	01110	10011	00101	101
7	01100	01100	11010	00111	10001	111

e.g., $\mathcal{M}_4$, all error patterns that yield the syndrom 011

choose the pattern with minimum Hamming weight as coset leader

3 Channel Coding

Syndrom decoding

The same table as before only considering the coset leaders and the syndroms.

μ	s_μ	e_μ
0	000	00000
1	001	00001
2	010	00010
3	100	00100
4	011	01000
5	110	10000
6	101	11000
7	111	01100

resort for easier look-up. s_μ contains already the address information

syndrom table

μ	s_μ	e_μ
0	000	00000
1	001	00001
2	010	00010
3	011	01000
4	100	00100
5	101	11000
6	110	10000
7	111	01100

As the coset leader was chosen with the minimum Hamming distance, it is the most likely error pattern for a certain syndrom

3 Channel Coding

Example: (5,2) Code continued $\Gamma = \{00000, 10110, 01011, 11101\}$

Assume we receive $\boldsymbol{y} = [11110]$

Calculate the Syndrom ("what is wrong with the received CW?")

$$\boldsymbol{s} = \boldsymbol{y} \cdot \boldsymbol{H}^T \qquad \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{H}^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\rightarrow \quad \boldsymbol{s} = [011]$$

Look-up in the syndrom table at position 3 (011 binary).

$$\rightarrow \quad \boldsymbol{e}_3 = [01000]$$

Invert the corresponding bit to find the most likely transmitted CW.

$$\rightarrow \boxed{\hat{\boldsymbol{a}} = [10110]}$$

3 Channel Coding

Convolutional codes:

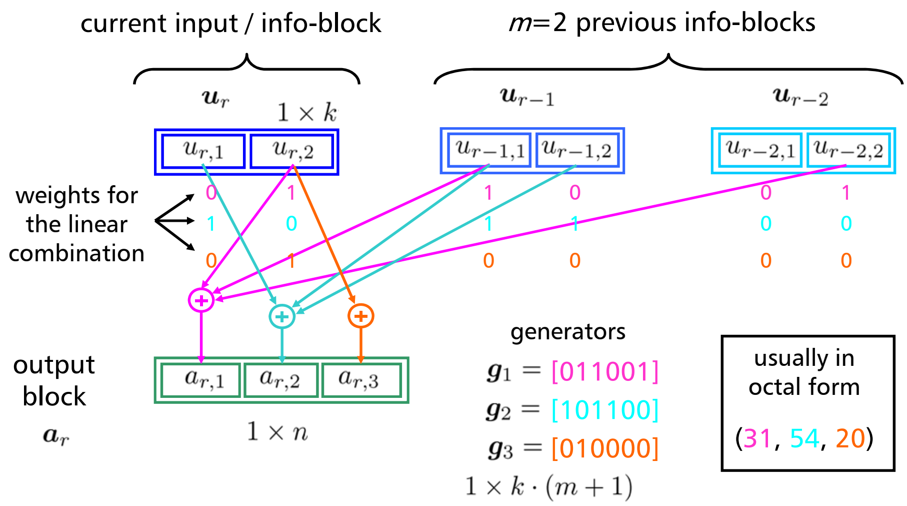
Features:

- No block processing; a whole sequence is convolved with a set of generator coefficients
- No analytic construction is known → good codes have been found by computer search
- Description is easier as compared to the block codes
- Simple processing of soft decission information → well suited for iterative decoding
- Coding gains from simple convolutional codes are similar as the ones from complex block codes
- Easy implementation via shift registers

3 Channel Coding

General structure:

Example: (n,k), e.g., (3,2) convolutional code with memory m=2 (constraint length K=m+1=3)



3 Channel Coding

Formal description:

Describes the linear combinations, how to compute the n output bits from the $k\,(m+1)$ input bits.

$$\boxed{a_{r,\nu} = \sum_{\mu=0}^{m} \sum_{\kappa=1}^{k} g_{\kappa,r,\mu} \cdot u_{r-\mu,\kappa}}$$

the ν^{th} bit from output block $\boldsymbol{a}_r$

the κ^{th} bit from input block $\boldsymbol{u}_{r-\mu}$

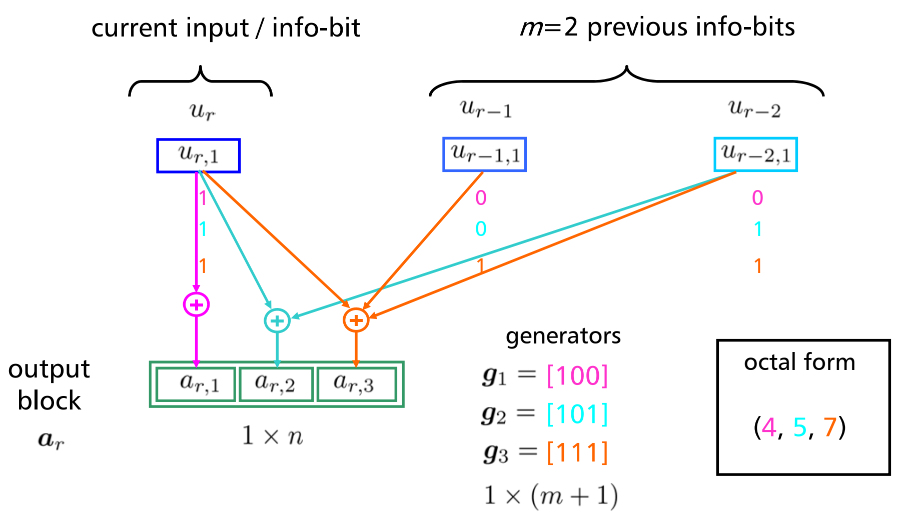
sum over the input blocks

corresponding weight, 0 or 1

sum over the bits of the input blocks

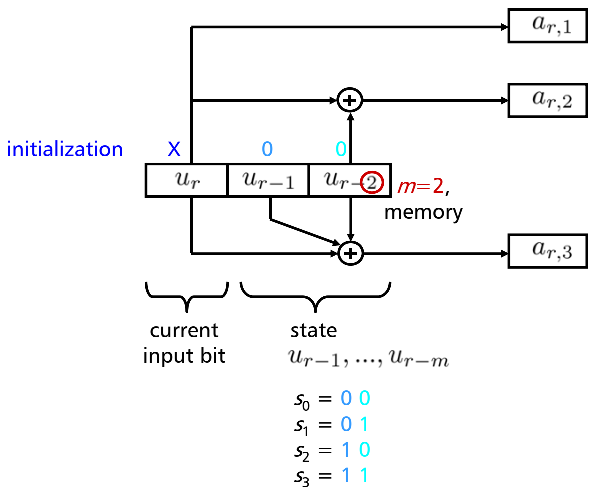
3 Channel Coding

General structure: often used, input blocks of size 1: (n,1), e.g., (3,1) convolutional codes



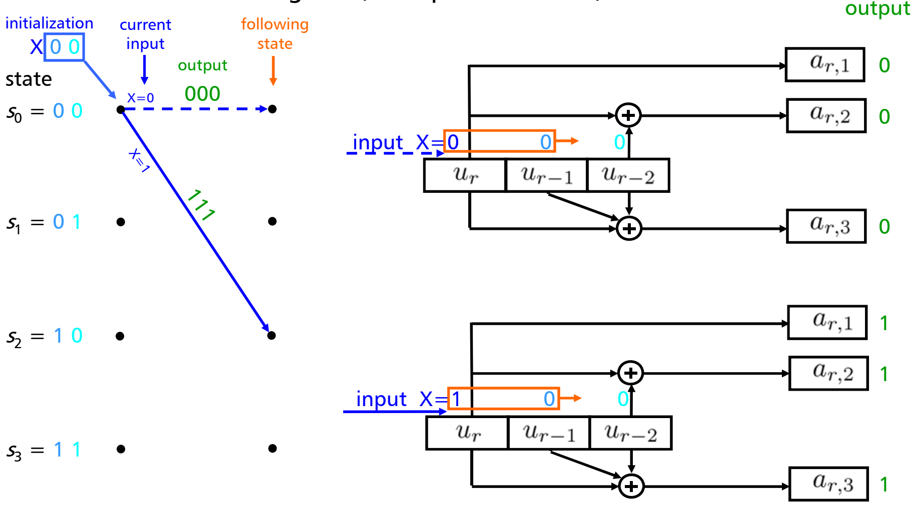
3 Channel Coding

General structure: visualization as shift register, e.g., (3,1) conv. code with generator (4,5,7), constraint length 3.



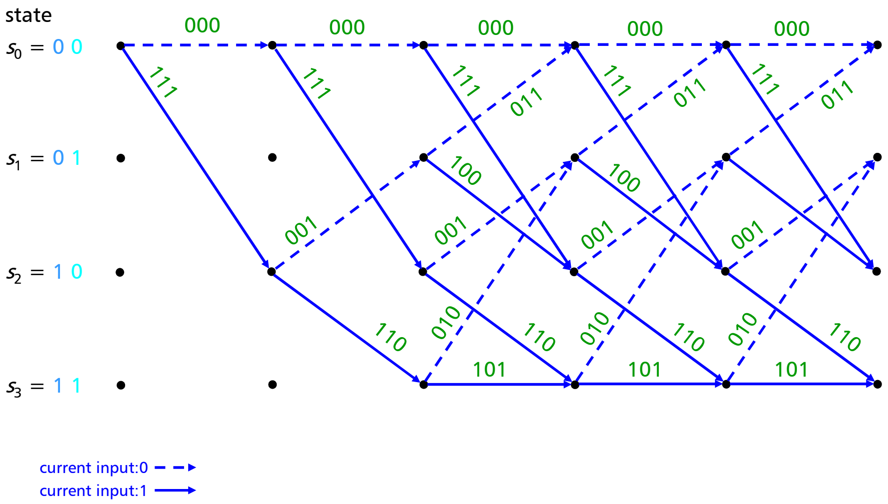
3 Channel Coding

Generation of Trellis diagram (example continued):



3 Channel Coding

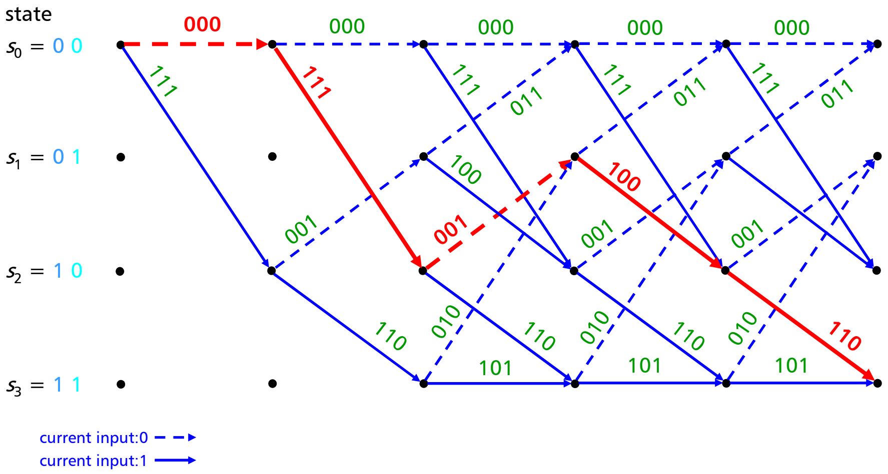
Trellis diagram (example continued):



3 Channel Coding

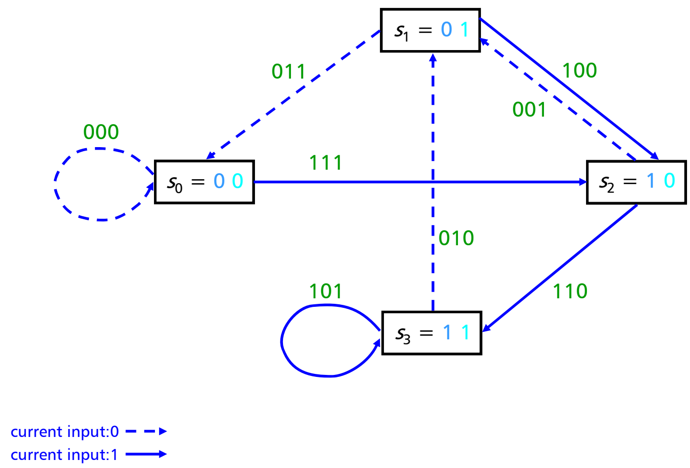
Encoding via the Trellis diagram (example continued):

Input seq.:	0	1	0	1	1	...
Output seq.:	000	111	001	100	110	...



3 Channel Coding

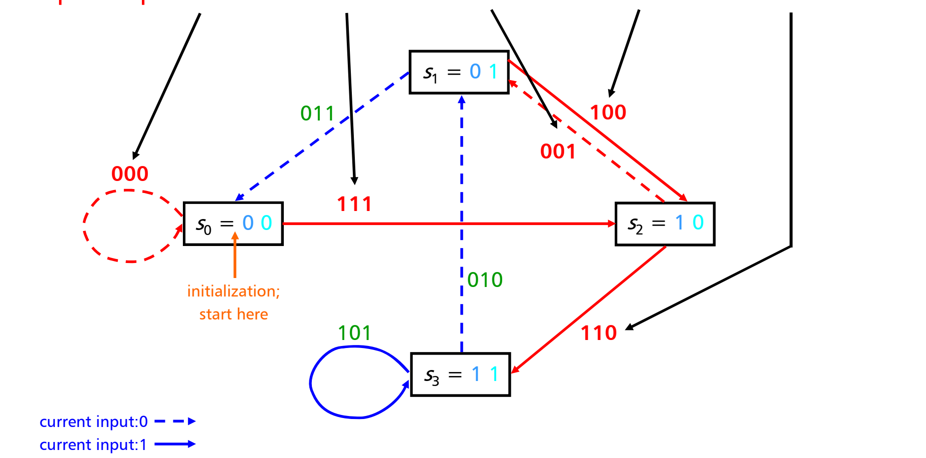
State diagram (example continued):
A more compact representation



3 Channel Coding

Encoding via state diagram (example continued):

Input seq.:	0	1	0	1	1	...
Output seq.:	000	111	001	100	110	...

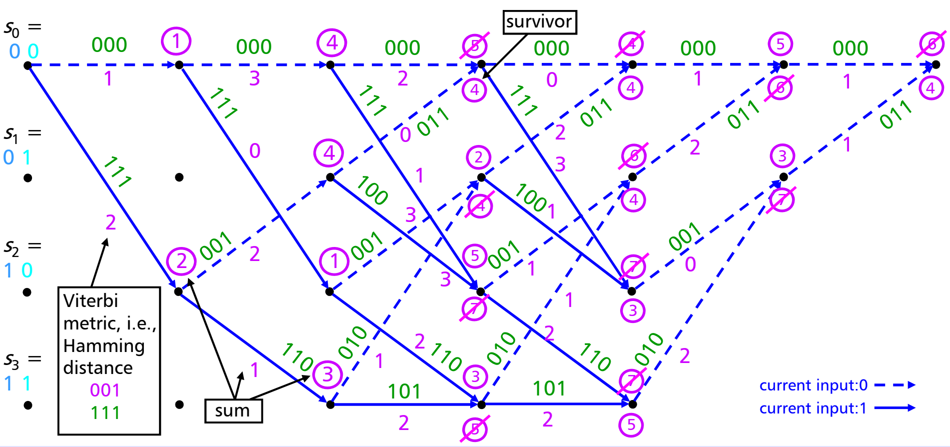


3 Channel Coding

Viterbi algorithm for hard decission decoding:

					termination / tail bits	
Info bits:	0	1	0	1	0	0
Transm.:	000	111	001	100	001	011
Received:	001	111	011	000	001	010

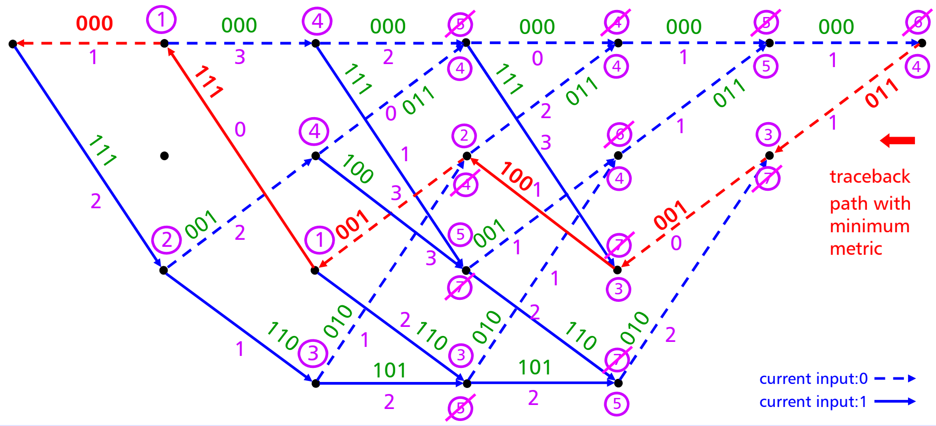
transmission errors



Viterbi algorithm for hard decission decoding:

termination / tail bits

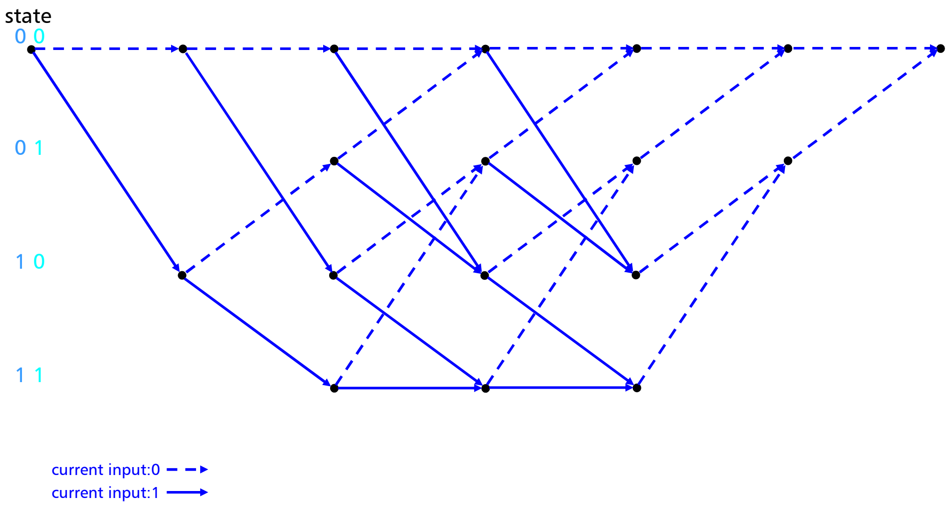
Info bits:	0	1	0	1	0	0
Transm.:	000	111	001	100	001	011
Received:	001	111	011	000	001	010
ML est.:	000	111	001	100	001	011
Decoded:	0	1	0	1	0	0



blank Trellis diagram:

termination / tail bits

0 0



3 Channel Coding

Summary: Viterbi algorithm for hard decission decoding:

- Generate the Trellis diagram depending on the code (which is defined by the generator)
- For any branch compute the Viterbi metrics, i.e., the Hamming distances between the possibly decoded sequence and the received sequence
- Sum up the individual branch metrics through the trellis (path metrics)
- At each point choose the suvivor, i.e., the path metric with the minimum weight
- At the end the zero state is reached again (for terminated codes)
- From the end of the Trellis trace back the path with the minimum metric and get the corresponding decoder outputs
- As the sequence with the minimum Hamming distance is found, this decoding scheme corresponds to the Maximum Likelihood decoding

Sometimes also different metrics are used as Viterbi metric, such as the number of equal bits. Then we need the path with the maximum metric.

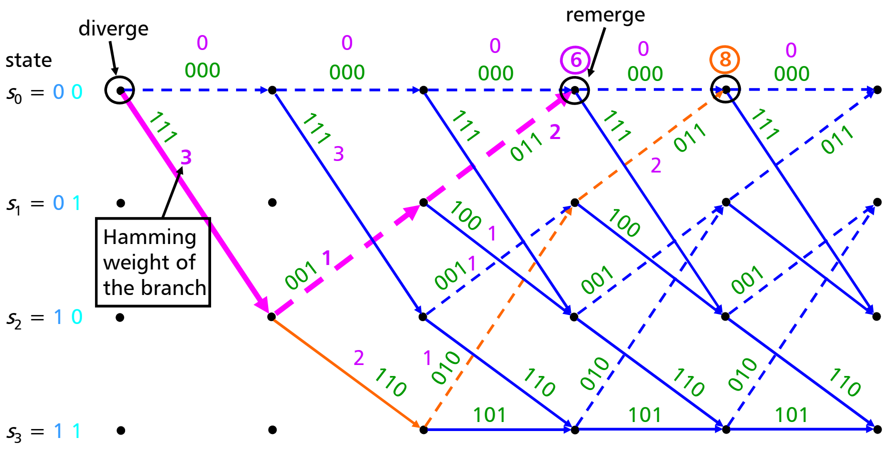
3 Channel Coding

How good are different convolutional codes?

- For Block codes it is possible to determine the minimum Hamming distance between the different code words, which is the main parameter that influences the bit error rate
- For convolutional codes a similar measure can be found. The free distance d_{free} is the number of bits which are at least different for two output sequences. The larger d_{free}, the better the code.
- A convolutional code is called optimal if the free distance is larger as compared to all other codes with the same rate and constraint length
- Even though the coding is a sequential process, the decoding is performed in chunks with a finite length (decoding window width)
- As convolutional codes are linear codes, the free distances are the distances between each of the code sequences and the all zero code sequence
- The minimum free distance is the minimum Hamming weight of all arbitrary long paths along the trellis that diverge and remerge to the all-zero path (similar to the minimum Hamming distance for linear block codes)

3 Channel Coding

Free distance (example recalled): (3,1) conv. code with generator (4,5,7).



The path diverging and remerging to all-zero path with minimum weight

$\boxed{d_{\text{free}} = 6}$

Note: This code is not optimal as there exists a better code with constraint length 3 that uses the generator (5,7,7) and reaches a free distance of 8

3 Channel Coding

How good are different convolutional codes?

- Optimal codes have been found via computer search, e.g.,

Code rate	Constraint length	Generator (octal)	Free distance
1 / 2	3	(5,7)	5
1 / 2	4	(15,17)	6
1 / 2	5	(23, 35)	7
1 / 3	3	(5,7,7)	8
1 / 3	4	(13,15,17)	10
1 / 3	5	(25,33,37)	12

Extensive tables, see reference: John G. Proakis, "Digital Communications"

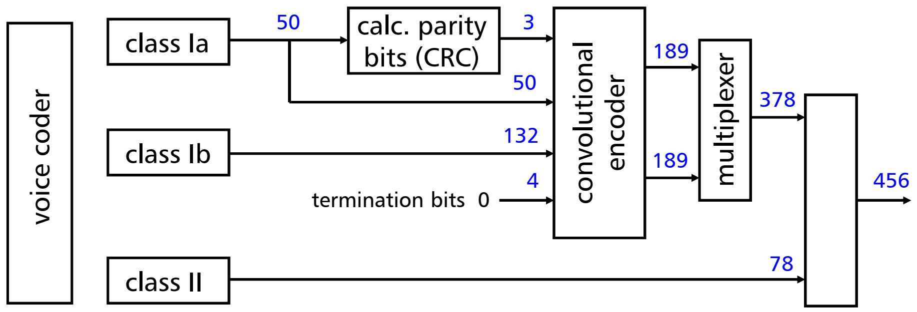
- As the decoding is done sequentially, e.g., with a large decoding window, the free distance gives only a hint on the number of bits that can be corrected. The higher the minimum distance, the more closely located errors can be corrected
- Therefore, interleavers are used to split up burst errors

3 Channel Coding

Application example GSM voice transmission

standardization 1982-1992
deployment starting 1992

- The speech codec produces blocks of 260 bits, from which some bits are more or less important for the speech quality
  - Class Ia: 50 bits most sensitive to bit errors
  - Class Ib: 132 bits moderately sensitive to bit errors
  - Class II: 78 bits least sensitive to bit errors



3 Channel Coding

Application example GSM voice transmission

- The voice samples are taken every 20ms, i.e., the output of the voice coder has a data rate of 260 bit / 20 ms = 12.7 kbit/s
- After the encoding we get 456 bits which means overall we get a code rate of about 0.57. The data rate increases to 456 bit / 20 ms = 22.3 kbit/s
- The convolutional encoder applies a rate ½ code with constraint length 5 (memory 4) and generator (23, 35), $d_{\text{free}} = 7$. The blocks are also terminated by appending 4 zero bits (tail bits).
- Specific decoding schemes or algorithms are usually not standardized. In most cases the Viterbi algorithm is used for decoding
- 2^4=16 states in the Trellis diagram
- In case 1 of the 3 parity bits is wrong (error in the most sensitive data) the block is discarded and replaced by the last one received correctly
- To avoid burst errors additionally an interleaver is used at the encoder output

3 Channel Coding

standardization 1990-2000
deployment starting 2001

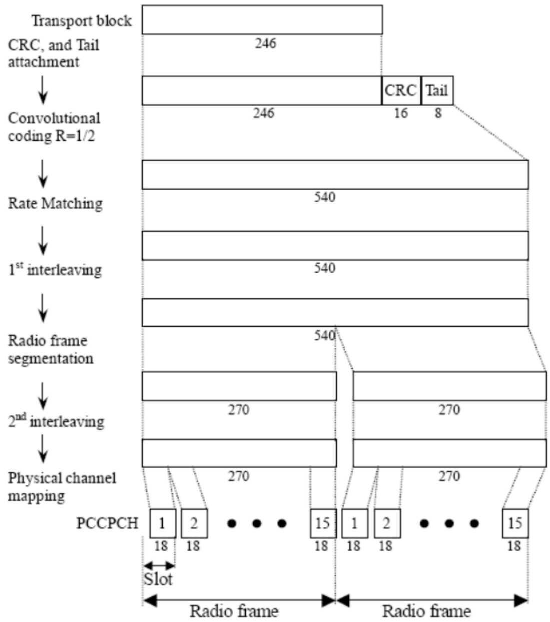
Application example UMTS:

- **Example: Broadcast channel (BCH)**
- Convolutional code:
  Rate ½
  Constraint length K=9 (memory m=8)
  generator (561,753),
  $d_{\text{free}} = 12$

→ 2^8=256 states in the Trellis diagram!

- Also Turbo codes got standardized

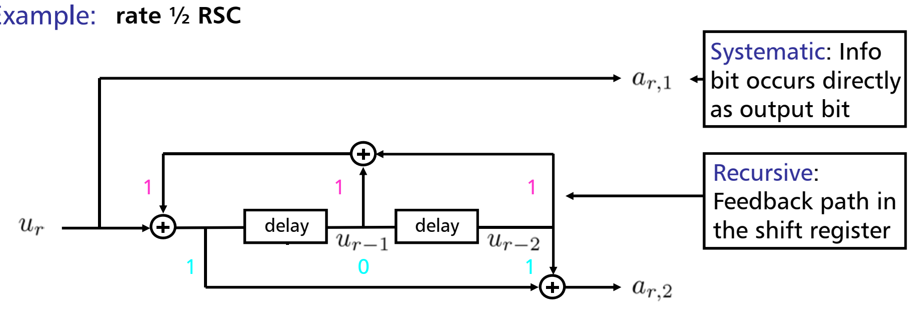
From: „Universal Mobile Telecommunications System (UMTS); Channel coding and multiplexing examples (ETSI 3GPP TR 25.944)“, 82 pages document



3 Channel Coding

Recursive Systematic Codes (RSC):

Example: **rate ½ RSC**



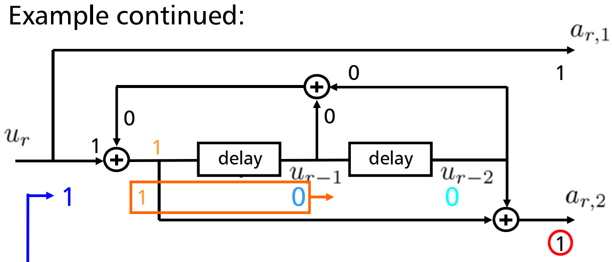
generators

feedback generator: $g_{1,fb} = [111]$ → $(7)_{\text{octal}}$

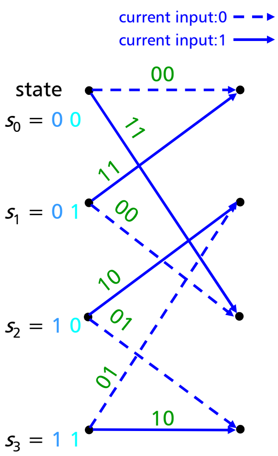
feedforward generator: $g_{1,ff} = [101]$ → $(5)_{\text{octal}}$

3 Channel Coding

Example continued:

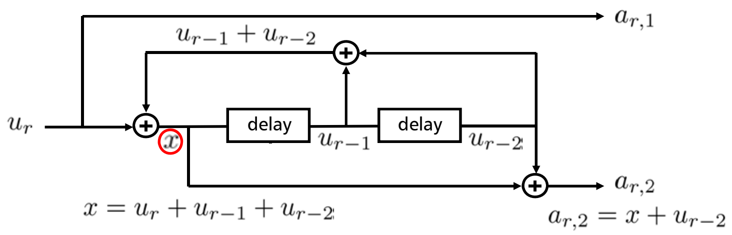


u_r	state	following state	$\boldsymbol{a}_r = [a_{r,1}\ a_{r,2}]$
0	00	00	00
1	00	10	11
0	01	10	00
1	01	00	11
0	10	11	01
1	10	01	10
0	11	01	01
1	11	11	10



3 Channel Coding

More detailed:

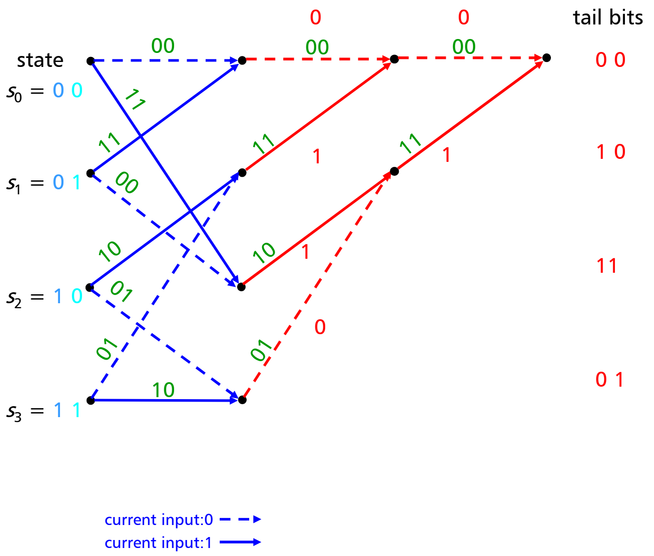


input u_r	state $[u_{r-1}\ u_{r-2}]$	x $u_r + u_{r-1} + u_{r-2}$	$a_{r,2}$ $x + u_{r-2}$	$a_{r,1}$ $= u_r$	output $\boldsymbol{a}_r = [a_{r,1}\ a_{r,2}]$	following state
0	00	0	0	0	00	00
1	00	1	1	1	11	10
0	01	1	0	0	00	10
1	01	0	1	1	11	00
0	10	1	1	0	01	11
1	10	0	0	1	10	01
0	11	0	1	0	01	01
1	11	1	0	1	10	11

3 Channel Coding

Tailbits for the terminated code?

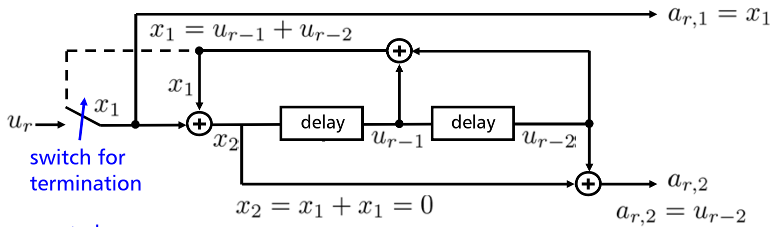
Depend on the state!



The tailbits are generated automatically by the encoder, depending on the encoded sequence

3 Channel Coding

How to terminate the code?



now generated from the state

will now be always zero, i.e., the state will get filled with zeros

input $x_1 = u_{r-1} + u_{r-2}$	state $[u_{r-1}\ u_{r-2}]$	$x_2 = x_1 + x_2$	$a_{r,2}$ u_{r-2}	$a_{r,1}$ $= x_1$	output $\boldsymbol{a}_r = [a_{r,1}\ a_{r,2}]$	following state
0	00	0	0	0	00	00
0	00	0	0	0	11	00
1	01	0	1	1	11	00
1	01	0	1	1	11	00
1	10	0	0	1	10	01
1	10	0	0	1	10	01
0	11	0	1	0	01	01
0	11	0	1	0	01	01

3 Channel Coding

Example: Termination if the last state has been „11“:

As the input is not arbitrary anymore, we get only 4 cases to consider

input $x_1 = u_{r-1} + u_{r-2}$	state $[u_{r-1}\ u_{r-2}]$	$x_2 = x_1 + x_2$	$a_{r,2}$ u_{r-2}	$a_{r,1}$ $= x_1$	output $\boldsymbol{a}_r = [a_{r,1}\ a_{r,2}]$	following state
0	00	0	0	0	00	00
1	$\boxed{01}$	0	1	1	11	$\boxed{00}$
1	10	0	0	1	10	01
0	$\boxed{11}$	0	1	0	01	$\boxed{01}$

From the state 11 we force the encoder back to the 00 state by generating the tail bits 0 1. The corresponding output sequence would be 01 11. See also the Trellis diagram for the termination.

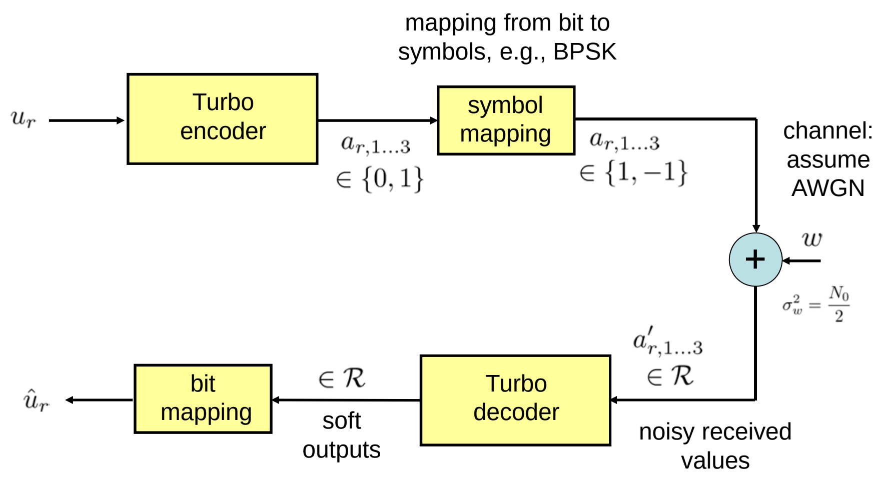
3 Channel Coding

Turbo codes:

- developed around 1993
- get close to the Shannon limit
- used in UMTS and DVB (Turbo Convolutional Codes, TCC)
  - parallel convolutional encoders are used
  - one gets a random permutation of the input bits
  - the decoder benefits then from two statistically independent encoded bits
  - slightly superior to TPC
  - noticeably superior to TPC for low code rates (~1 dB)
- used in WLAN, Wimax (Turbo Product Codes, TPC)
  - serial concatenated codes; based on block codes
  - data arranged in a matrix or in a 3 dimensional array
  - e.g., Hamming codes along the dimensions
  - good performance at high code rates
  - good coding gains with low complexity

3 Channel Coding

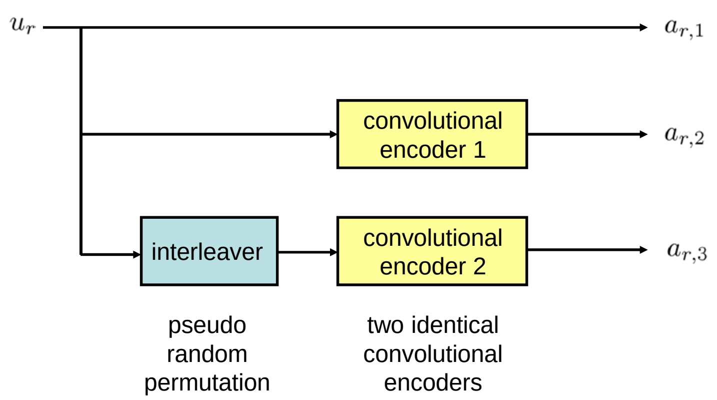
System overview:



3 Channel Coding

Turbo encoder (for Turbo Convolutional Codes, TCC):

Structure of a rate 1/3 turbo encoder

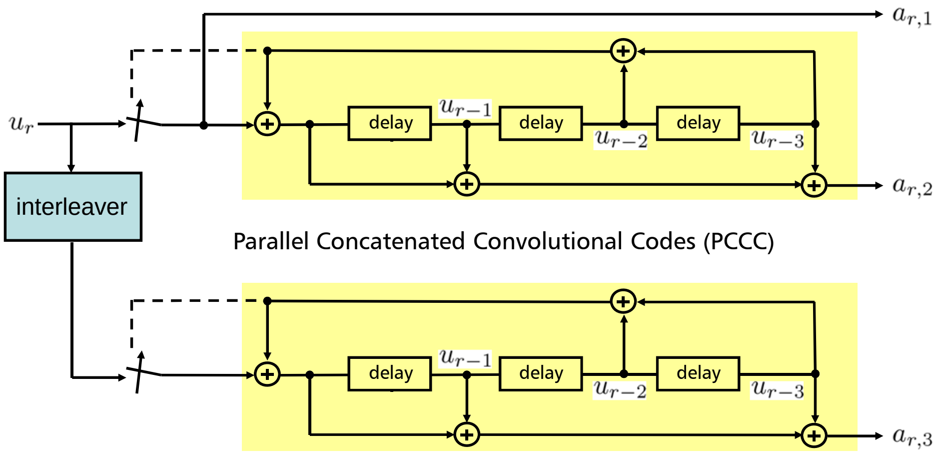


The turbo code is a block code, as a certain number of bits need to be buffered first in order to fill the interleaver

3 Channel Coding

Example: UMTS Turbo encoder:

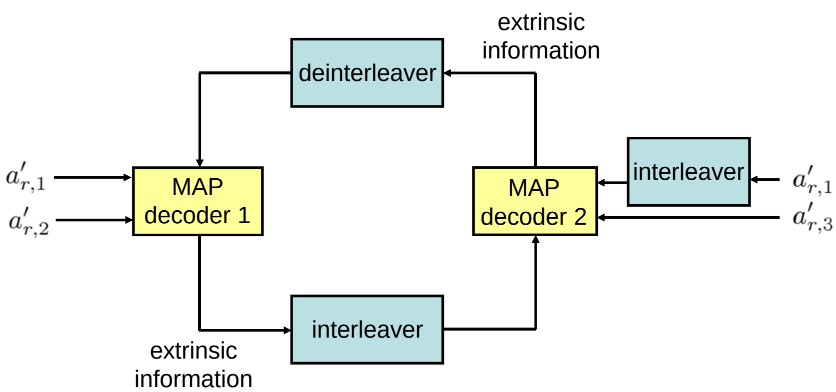
Rate 1/3, RSC with feedforward generator (15) and feedback generator (13)



3 Channel Coding

Turbo decoder:

Structure of a turbo decoder



The MAP decoders produce a soft output which is a measure for the reliability of their decission for **each of the bits**. This likelihood is used as soft input for the other decoder (which decodes the interleaved sequence). The process is repeated until there‘s no significant improvement of the extrinsic information anymore.

3 Channel Coding

MAP (Maximum a posteriori probability) Decoding:

- Difference compared to the Viterbi decoding:
  - Viterbi decoders decode a whole sequence (maximum likelihood sequence estimation). If instead of the Hamming distance the Euklidean distance is used as Viterbi metric we easily get the Soft-Output Viterbi algorithm (SOVA)
  - The SOVA provides a reliability measure for the decission of the whole sequence
- For the application in iterative decoding schemes a reliability measure for each of the bits is desirable, as two decoders are used to decode the same bit independently and exchange their reliability information to improve the estimate. The indepencence is artificially generated by applying an interleaver at the encoding stage.
- In the Trellis diagram the MAP decoder uses some bits before and after the current bit to find the most likely current bit
- MAP decoding is used in systems with memory, e.g., convolutional codes or channels with memory

3 Channel Coding

- Consider the transmission over an AWGN channel applying a binary modulation scheme (higher order modulation schemes can be treated by grouping bits).

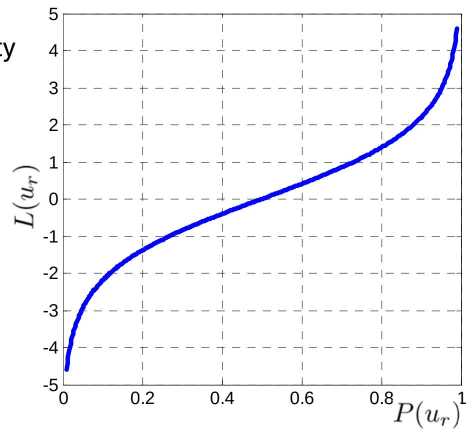
  Mapping: 0 → 1 and 1 → -1

- Suitable measure for the reliability

  Log-Likelihood Ratio (LLR)

$$L(u_r) = \ln\left(\frac{P(u_r=+1)}{P(u_r=-1)}\right)$$

$$L(u_r) = \ln\left(\frac{P(u_r=+1)}{1-P(u_r=+1)}\right)$$



3 Channel Coding

- The reliability measure (LLR) for a single bit at time r under the condition that a sequence y_1^N ranging from 1 to N has been received is:

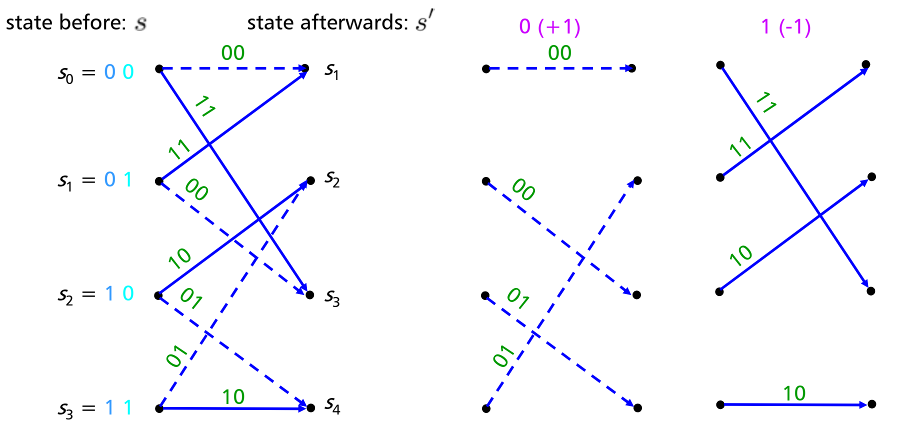
$$L(u_r) = \ln\left(\frac{P(u_r = +1) \mid y_1^N}{P(u_r = -1) \mid y_1^N}\right)$$

with Bayes rule: $\underbrace{P(A,B)}_{\substack{\text{joint}\\\text{probability}}} = \underbrace{P(A)}_{\substack{\text{a-priori}\\\text{probability}\\\text{of A}}} \cdot \underbrace{P(B|A)}_{\substack{\text{a-posteriori}\\\text{probability of}\\\text{B}}}$

$$L(u_r) = \ln\left(\frac{P(u_r = +1, y_1^N)/P(y_1^N)}{P(u_r = -1, y_1^N)/P(y_1^N)}\right) = \ln\left(\frac{P(u_r = +1, y_1^N)}{\underset{\substack{\uparrow\\\text{unknown}}}{P(u_r = -1}, \underset{\substack{\uparrow\\\text{known,}\\\text{observed}}}{y_1^N)}}\right)$$

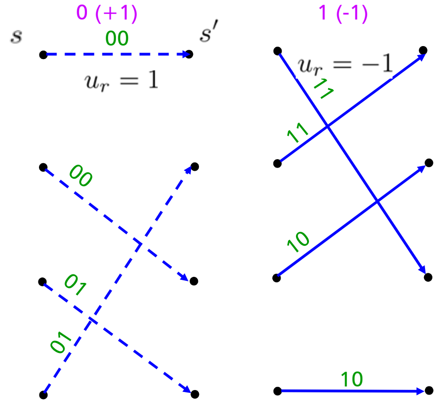
3 Channel Coding

- Example as used before Rate ½ RSC with generators 5 and 7:
- The probability that u_r becomes +1 or -1 can be expressed in terms of the starting and ending states in the trellis diagram



3 Channel Coding

$$P(u_r=+1) = \underbrace{P(s=s_0, s'=s_0)}_{\text{joint probability for a pair of starting and ending states}} + P(s=s_1, s'=s_2) + \ldots$$



$$P(u_r=+1) = \underbrace{\sum_{u_r=+1} P(s,s')}$$

probability for all combinations of starting and ending states that will yield a +1

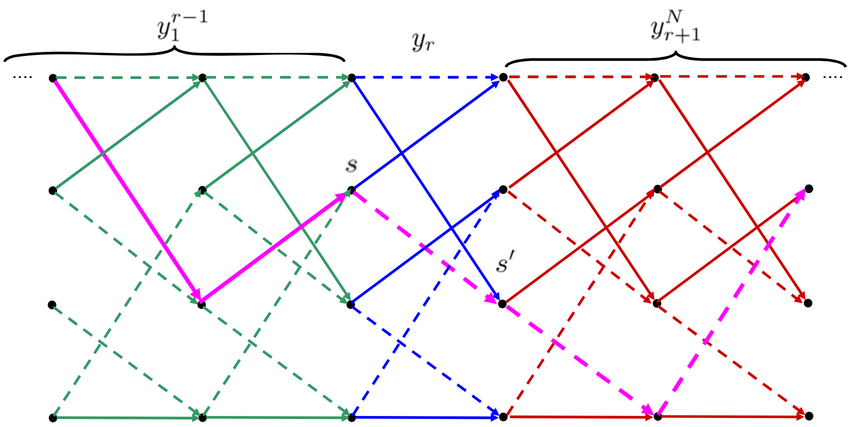
$$P(u_r=-1) = \underbrace{\sum_{u_r=-1} P(s,s')}$$

probability for all combinations of starting and ending states that will yield a -1

$$L(u_r) = \ln \frac{P(u_r=+1, y_1^N)}{P(u_r=-1, y_1^N)} = \ln \frac{\sum_{u_r=+1} P(s,s',y_1^N)}{\sum_{u_r=-1} P(s,s',y_1^N)}$$

3 Channel Coding

- The probability to observe a certain pair of states (s,s') depends on the past and the future bits. Therefore, we split the sequence of received bits into the past, the current, and the future bits



$$P(s,s',y_1^N) = P(s,s',y_1^{r-1},y_r,y_{r+1}^N)$$

3 Channel Coding

- Using Bayes rule to split up the expression into past, present and future

$$P(s,s',y_1^N) = P(s,s',y_1^{r-1},y_r,y_{r+1}^N)$$
$$P(s,s',y_1^N) = P(y_{r+1}^N \mid s,s',y_1^{r-1},y_r)\cdot P(s,s',y_1^{r-1},y_r)$$

- Looking at the Trellis diagram, we see the the future y_{r+1}^N is independent of the past. It only depends on the current state s'

$$P(s,s',y_1^N) = P(y_{r+1}^N \mid \cancel{s},s',\cancel{y_1^{r-1}},\cancel{y_r})\cdot P(s,s',y_1^{r-1},y_r)$$
$$P(s,s',y_1^N) = P(y_{r+1}^N \mid s')\cdot P(s,s',y_1^{r-1},y_r)$$

- Using again Bayes rule for the last probability

$$P(s,s',y_1^{r-1},y_r) = P(s',y_r \mid s,y_1^{r-1})\cdot P(s,y_1^{r-1})$$

- Summarizing

$$P(s,s',y_1^N) = P(y_{r+1}^N \mid s')\cdot P(s',y_r \mid s,y_1^{r-1})\cdot P(s,y_1^{r-1})$$

3 Channel Coding

- Identifying the metrics to compute the MAP estimate

$$P(s,s',y_1^N) = \underbrace{P(y_{r+1}^N \mid s')}\cdot\underbrace{P(s',y_r \mid s,y_1^{r-1})}\cdot\underbrace{P(s,y_1^{r-1})}$$

probability for a certain future given the current state, called **Backward metric**

$\beta_r(s')$

probability to observe a certain state and bit given the state and the bit before, called **Transition metric**

$\gamma_r(s',s)$

probability for a certain state and a certain past, called **Forward metric**

$\alpha_{r-1}(s)$

- Now rewrite the LLR in terms of the metrics

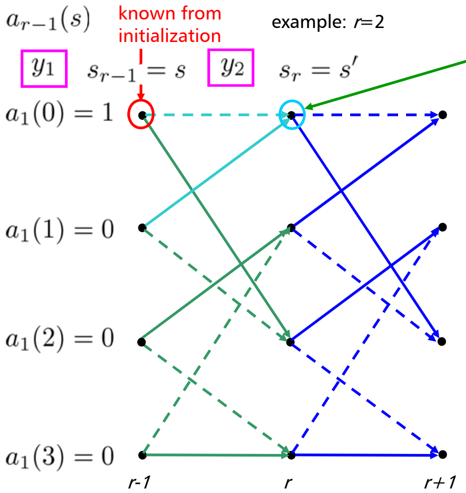
$$P(s,s',y_1^N) = \beta_r(s')\cdot\gamma_r(s',s)\cdot\alpha_{r-1}(s)$$

$$\boxed{L(u_r) = \ln\frac{\sum_{u_r=+1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot\gamma_r(s',s)}{\sum_{u_r=-1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot\gamma_r(s',s)}}$$

3 Channel Coding

- How to calculate the metrics? Forward metric $\alpha_r(s')$:

probability for a certain state and a certain past, called **Forward metric**	$\alpha_{r-1}(s) = P(s, y_1^{r-1})$



probability to arrive in a certain state and the corresponding sequence that yielded that state

$$\alpha_2(0) = \sum_s P(s, s' = 0, y_1^{r-1}, y_r)$$

$$\alpha_r(s') = \sum_s P(s, s', y_1^{r-1}, y_r)$$

using again Bayes rule and

$$\gamma_r(s', s) = P(s', y_r \mid s, y_1^{r-1})$$

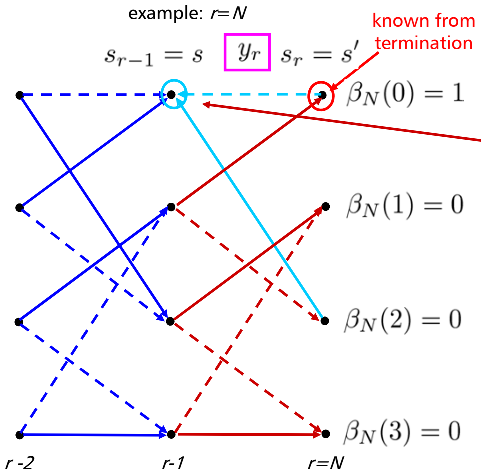
$$\alpha_{r-1}(s) = P(s, y_1^{r-1})$$

$$\boxed{\alpha_r(s') = \sum_s \gamma_r(s', s) \cdot \alpha_{r-1}(s)}$$

3 Channel Coding

- How to calculate the metrics? Back metric $\beta_r(s')$:

probability for a certain future given the current state, called **Backward metric**	$\beta_r(s') = P(y_{r+1}^N \mid s')$

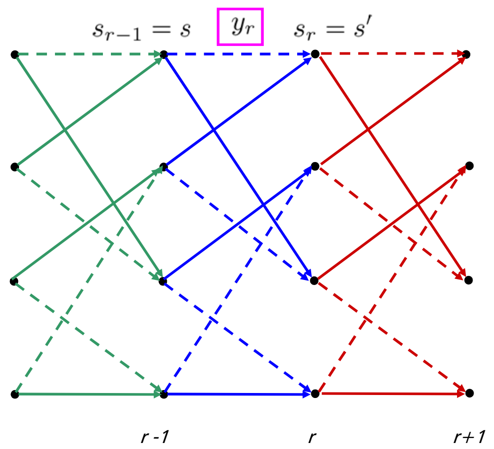


$$\boxed{\beta_{r-1}(s) = \sum_{s'} \gamma_r(s', s) \cdot \beta_r(s')}$$

3 Channel Coding

- How to calculate the metrics? Transition metric $\gamma_r(s, s')$:

probability to observe a certain state and bit given the state and the bit before, called **Transition metric**	$\gamma_r(s,s') = P(s', y_r \mid s, y_1^{r-1})$



$$\gamma_r(s,s') = P(s', y_r \mid s, \cancel{y_1^{r-1}})$$

for a given state s the transition probability does not depend on the past

$$\gamma_r(s,s') = P(s', y_r \mid s)$$

$$\gamma_r(s,s') = P(y_r|s', s) \cdot \underbrace{P(s' \mid s)}$$

$$\gamma_r(s,s') = P(y_r|s', s) \cdot P(u_r)$$

prob. to observe a received bit for a given pair of states

prob. for this pair of states, i.e., the a-priori prob. of the input bit

3 Channel Coding

- Now some math: $\gamma_r(s,s') = P(y_r|s', s) \cdot \boxed{P(u_r)}$ ← starting with this one

expressing the a-priori probability in terms of the Likelihood ratio

$$L(u_r) = \ln\left(\frac{P(u_r = +1)}{P(u_r = -1)}\right) = \ln\left(\frac{P(u_r = +1)}{1 - P(u_r = +1)}\right)$$

$$\exp[L(u_r)] = \frac{P(u_r = +1)}{1 - P(u_r = +1)}$$

$$P(u_r = +1) = \exp[L(u_r)] \cdot (1 - P(u_r = +1))$$

with

$$1 + \exp[L(u_r)] = 1 + \frac{P(u_r = +1)}{1 - P(u_r = +1)} = \frac{1 \cancel{- P(u_r = +1) + P(u_r = +1)}}{1 - P(u_r = +1)}$$

$$P(u_r = +1) = \frac{\exp[L(u_r)]}{1 + \exp[L(u_r)]}$$

$$P(u_r = +1) = \frac{1}{1 + \exp[-L(u_r)]}$$

3 Channel Coding

$$P(u_r=+1)=\frac{1}{1+\exp[-L(u_r)]}$$

$$P(u_r=-1)=1-P(u_r=+1)=1-\frac{1}{1+\exp[-L(u_r)]}=\frac{\exp[-L(u_r)]}{1+\exp[-L(u_r)]}$$

now combining the terms in a smart way to one expression

$$P(u_r=\pm 1)=\underbrace{\frac{\exp[-L(u_r)/2]}{1+\exp[-L(u_r)]}}_{A_r}\cdot\exp[\pm L(u_r)/2]$$

1 for '+' and $\exp[-L(u_r)]$ for '-'

we get the a-priori probability in terms of the likelihood ratio as

$$\boxed{P(u_r=\pm 1)=A_r\cdot\exp[\pm L(u_r)/2]\quad\text{with}\quad A_r=\frac{\exp[-L(u_r)/2]}{1+\exp[-L(u_r)]}}$$

3 Channel Coding

- Now some more math: $\gamma_r(s,s')=\boxed{P(y_r|s',s)}\cdot P(u_r)$ continuing with this one

$$P(y_r|s',s)=P(\underbrace{[a'_{r,1}\;a'_{r,2}]}_{\text{pair of observed bits}}\mid\underbrace{[a_{r,1}\;a_{r,2}]}_{\text{pair of transmitted coded bits, belonging to the encoded info bit } u_r})$$

$$P(y_r|s',s)=P(a'_{r,1}\mid a_{r,1})\cdot P(a'_{r,2}\mid a_{r,2})$$

example for code rate ½. Can easily be extended

noisy observation, disturbed by AWGN

$$P(y_r|u_r)=\frac{1}{\sqrt{2\pi\sigma_n^2}}\cdot\underbrace{\exp\left(-\frac{(a'_{r,1}-a_{r,1})^2}{2\cdot\sigma_n^2}\right)}\cdot\frac{1}{\sqrt{2\pi\sigma_n^2}}\cdot\exp\left(-\frac{(a'_{r,2}-a_{r,2})^2}{2\cdot\sigma_n^2}\right)$$

$$\exp\left(-\frac{a'^2_{r,1}-2\cdot a'_{r,1}\cdot a_{r,1}+a^2_{r,1}}{2\cdot\sigma_n^2}\right)=\exp\left(-\frac{a'^2_{r,1}+a^2_{r,1}}{2\cdot\sigma_n^2}\right)\cdot\exp\left(\frac{\cancel{+2}\cdot a'_{r,1}\cdot a_{r,1}}{\cancel{2}\cdot\sigma_n^2}\right)$$

+1 or -1 squared → always 1

3 Channel Coding

$$P(y_r|u_r) = \underbrace{\left(\frac{1}{\sqrt{2\pi\sigma_n^2}}\right)^2 \cdot \exp\left(-\frac{a_{r,1}'^2+1}{2\cdot\sigma_n^2} - \frac{a_{r,2}'^2+1}{2\cdot\sigma_n^2}\right)}_{B_r} \cdot \exp\left(\frac{a_{r,1}'\cdot a_{r,1}}{\sigma_n^2} + \frac{a_{r,2}'\cdot a_{r,2}}{\sigma_n^2}\right)$$

- Now the full expression: $\gamma_r(s,s') = P(y_r|s',s)\cdot P(u_r)$

$$\gamma_r(s,s') = B_r\cdot\exp\left(\frac{a_{r,1}'\cdot a_{r,1}}{\sigma_n^2} + \frac{a_{r,2}'\cdot a_{r,2}}{\sigma_n^2}\right)\cdot A_r\cdot\exp[\pm L(u_r)/2]$$

$$\sigma_n^2 = \frac{N_0}{2} \qquad a_{r,1} = \pm 1 \qquad a_{r,1}$$

$$\gamma_r(s,s') = A_r\cdot B_r\cdot\exp\left[\frac{2}{N_0}\left(a_{r,1}'\cdot a_{r,1} + a_{r,2}'\cdot a_{r,2}\right)\right]\cdot\exp[\underbrace{\pm L(u_r)}_{\text{a-priori information}}/2]$$

$$\gamma_r(s,s') = A_r\cdot B_r\cdot\exp\left[\frac{2}{N_0}\left(a_{r,1}'\cdot a_{r,1} + a_{r,2}'\cdot a_{r,2}\right)\right]\cdot\exp[\overbrace{a_{r,1}\cdot L_a(a_{r,1})}/2]$$

$$\gamma_r(s,s') = A_r\cdot B_r\cdot\exp\left[\frac{2}{N_0}\left(a_{r,1}'\cdot a_{r,1} + a_{r,2}'\cdot a_{r,2}\right) + \frac{1}{2}\cdot a_{r,1}\cdot L_a(a_{r,1})\right]$$

3 Channel Coding

$$\gamma_r(s,s') = A_r\cdot B_r\cdot\exp\left[\frac{4}{N_0}\left(\frac{1}{2}\cdot a_{r,1}'\cdot a_{r,1} + \frac{1}{2}\cdot a_{r,2}'\cdot a_{r,2}\right) + \frac{1}{2}\cdot a_{r,1}\cdot L_a(a_{r,1})\right]$$

$$\gamma_r(s,s') = A_r\cdot B_r\cdot\exp\left[\frac{1}{2}\cdot a_{r,1}\cdot L_a(a_{r,1}) + \frac{4}{N_0}\cdot\frac{1}{2}\cdot a_{r,1}'\cdot a_{r,1}\right]\cdot\exp\left[\frac{4}{N_0}\cdot\frac{1}{2}\cdot a_{r,2}'\cdot a_{r,2}\right]$$

$$\frac{4}{N_0} = \frac{2}{\sigma_n^2} = L_c$$ abbreviation

$$\gamma_r(s,s') = A_r\cdot B_r\cdot\exp\left[\frac{1}{2}\cdot a_{r,1}\cdot L_a(a_{r,1}) + L_c\cdot\frac{1}{2}\cdot a_{r,1}'\cdot a_{r,1}\right]\cdot\underbrace{\exp\left[L_c\cdot\frac{1}{2}\cdot a_{r,2}'\cdot a_{r,2}\right]}_{\gamma_r^e(s,s')}$$

from before:

$$L(u_r) = \ln\frac{\sum_{u_r=+1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot\gamma_r(s',s)}{\sum_{u_r=-1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot\gamma_r(s',s)} \qquad \alpha_r(s') = \sum_s \gamma_r(s',s)\cdot\alpha_{r-1}(s)$$

$$\beta_{r-1}(s) = \sum_{s'}\gamma_r(s',s)\cdot\beta_r(s')$$

3 Channel Coding

$$\gamma_r(s,s') = A_r\cdot B_r\cdot \exp\left[\frac{1}{2}\cdot a_{r,1}\cdot L_a(a_{r,1}) + L_c\cdot\frac{1}{2}\cdot a'_{r,1}\cdot a_{r,1}\right]\cdot\exp\left[L_c\cdot\frac{1}{2}\cdot a'_{r,2}\cdot a_{r,2}\right]$$

unknown at the receiver, but resulting from the corresponding branch in the Trellis diagram $s \rightarrow s'$

$$L(u_r) = \ln\frac{\sum_{u_r=+1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot A_r\cdot B_r\cdot\exp\left[\frac{1}{2}\cdot a_{r,1}\cdot L_a(a_{r,1}) + L_c\cdot\frac{1}{2}\cdot a'_{r,1}\cdot a_{r,1}\right]\cdot\exp\left[L_c\cdot\frac{1}{2}\cdot a'_{r,2}\cdot a_{r,2}\right]}{\sum_{u_r=-1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot A_r\cdot B_r\cdot\exp\left[\frac{1}{2}\cdot a_{r,1}\cdot L_a(a_{r,1}) + L_c\cdot\frac{1}{2}\cdot a'_{r,1}\cdot a_{r,1}\right]\cdot\exp\left[L_c\cdot\frac{1}{2}\cdot a'_{r,2}\cdot a_{r,2}\right]}$$

due to the assumptions

positive

negative

$$\ln\frac{\exp\left[\frac{1}{2}\cdot 1\cdot L_a(a_{r,1}) + L_c\cdot\frac{1}{2}\cdot a'_{r,1}\cdot 1\right]}{\exp\left[-\frac{1}{2}\cdot 1\cdot L_a(a_{r,1}) - L_c\cdot\frac{1}{2}\cdot a'_{r,1}\cdot 1\right]} = \ln\exp\left[L_a(a_{r,1}) + L_c\cdot a'_{r,1}\right]$$

$$L(u_r) = \left[L_a(a_{r,1}) + L_c\cdot a'_{r,1}\right] + \ln\frac{\sum_{u_r=+1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot\gamma_r^e(s,s')}{\sum_{u_r=-1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot\gamma_r^e(s,s')}$$

with $\gamma_r^e(s,s') = \exp\left[L_c\cdot\frac{1}{2}\cdot a'_{r,2}\cdot a_{r,2}\right]$

$$\alpha_r(s') = \sum_s \gamma_r(s',s)\cdot\alpha_{r-1}(s) \qquad \beta_{r-1}(s) = \sum_{s'}\gamma_r(s',s)\cdot\beta_r(s')$$

3 Channel Coding

- Interpretation:

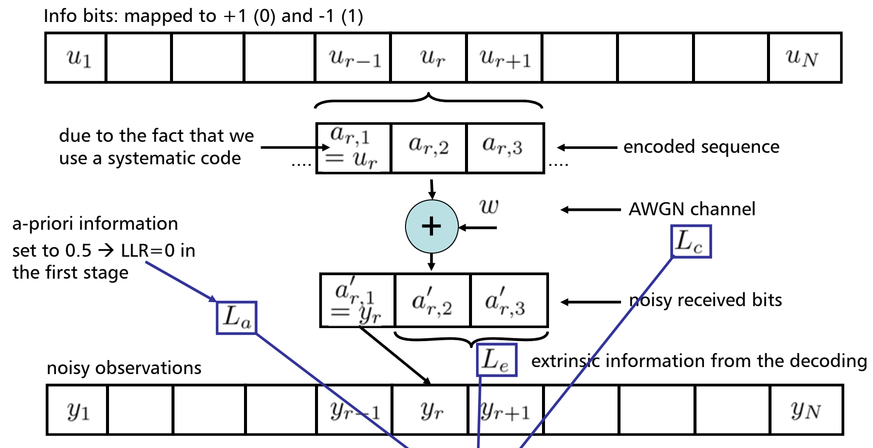
$$L(u_r) = \underbrace{L_a(a_{r,1})}_{\text{a-priori information about the transmitted bit, taken from an initial estimate before running the MAP algorithm}} + \underbrace{L_c\cdot a'_{r,1}}_{\text{information provided by the observation. Only depending on the channel; not on the coding scheme}} + \underbrace{\ln\frac{\sum_{u_r=+1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot\gamma_r^e(s,s')}{\sum_{u_r=-1}\alpha_{r-1}(s)\cdot\beta_r(s')\cdot\gamma_r^e(s,s')}}_{\text{a-posteriori (extrinsic) information. Gained from the applied coding scheme}}$$

$$\boxed{L(u_r) = L_a(a_{r,1}) + L_c\cdot a'_{r,1} + L_e(a_{r,1}) \qquad \hat{u}_r = \mathrm{sign}\{L(u_r)\}}$$

- In a Turbo decoder the extrinsic information of one MAP decoder is used as a-priori information of the second MAP decoder. This exchange of extrinsic information is repeated, until the extrinsic information does not change significantly anymore.

3 Channel Coding

- Summary:



yields the LLR and therefore, the bit estimate $\hat{u}_r$

$$\boxed{L(u_r) = L_a(a_{r,1}) + L_c \cdot a'_{r,1} + L_e(a_{r,1}) \qquad \hat{u}_r = \text{sign}\{L(u_r)\}}$$

3 Channel Coding

- Iterations:

$$\boxed{L(u_r) = L_a(a_{r,1}) + \underbrace{L_c \cdot a'_{r,1}}_{\text{constant over iterations} \rightarrow K} + L_e(a_{r,1}) \qquad \hat{u}_r = \text{sign}\{L(u_r)\}}$$

Iteration #1:

$$L_1(u_r) = 0 + K + L_{e,1,1}(a_{r,1})$$

first iteration, first decoder,a-priori LLR=0

$$L_2(u_r) = L_{e,1,1}(a_{r,1}) + K + L_{e,1,2}(a_{r,1})$$

first iteration, second decoder: uses extrinsic information from the first one as a-priori information

Iteration #2:

$$L_1(u_r) = L_{e,1,2}(a_{r,1}) + K + L_{e,2,1}(a_{r,1})$$

continuing in the same fashion with further iterations

$$L_2(u_r) = L_{e,2,1}(a_{r,1}) + K + L_{e,2,2}(a_{r,1})$$

Iteration #3:

$$L_1(u_r) = L_{e,2,2}(a_{r,1}) + K + L_{e,3,1}(a_{r,1})$$

$$L_2(u_r) = L_{e,3,1}(a_{r,1}) + K + L_{e,3,2}(a_{r,1})$$

reference:
see tutorials at www.complextoreal.com
or http://www.vashe.org/
Notes: We used a slightly different notation. The first tutorial has some minor errors but most cancel out

3 Channel Coding

Low-Density Parity Check (LDPC) codes:

- first proposed 1962 by Gallager
- due to comutational complexity neglegted until the 90s
- new LDPC codes outperform Turbo Codes
- reach the Shannon limit within hundredths decibel for large block sizes, e.g., size of the parity check matrix 10000 x 20000
- are used already for satellite links (DVB-S2, DVB-T2) and in optical communications
- have been adopted in IEEE wireless local areal network standards, e.g., 802.11n or IEEE 802.16e (Wimax)
- are under consideration for the long-term evolution (LTE) of third generation mobile telephony
- are block codes with parity check matrices containing only a small number of non-zero elements
- complexity and minimum Hamming distance increase linearly with the block length

3 Channel Coding

Low-Density Parity Check (LDPC) codes:

- not different to any other block code (besides the sparse parity check matrix)
- design: find a sparse parity check matrix and determine the generator matrix
- difference to classical block codes: LDPC codes are decoded iteratively

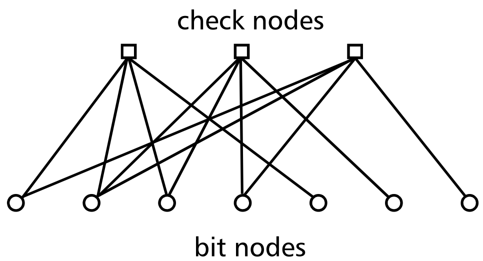
3 Channel Coding

Tanner graph

- graphical representation of the parity check matrix
- LDPC codes are often represented by the Tanner graph

Example: (7,4) Hamming code

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$



- n bit nodes
- n - k check nodes, i.e., parity check equations
- Decoding via message passing (MP) algorithm. Likelihoods are passed back and forth between the check nodes and bit nodes in an iterative fashion

3 Channel Coding

Encoding

- use Gaussian elimination to find $\boldsymbol{H} = \left[-\boldsymbol{P}^T \vdots \boldsymbol{I}_{n-k} \right]$
- construct the generator matrix $\boldsymbol{G} = \left[\boldsymbol{I}_k \vdots \boldsymbol{P} \right]$
- calculate the set of code words $\underbrace{\boldsymbol{a}_i}_{1\times n} = \underbrace{\boldsymbol{u}_i}_{1\times k} \cdot \underbrace{\boldsymbol{G}}_{k\times n}$

3 Channel Coding

Example:

- length 12 (3,4) regular LDPC code
  parity check code as introduced by Gallager

$$H = \left[\begin{array}{cccccccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hline
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
\hline
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{array}\right]$$

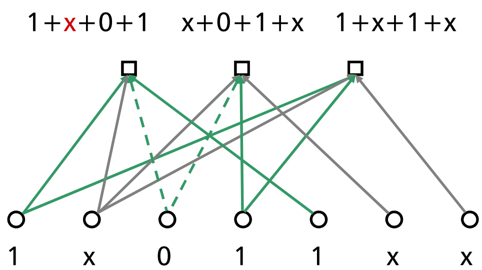
3 Channel Coding

Message Passing (MP) decoding

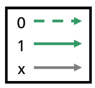
- soft- and hard decision algorithms are used
- often log-likelihood ratios are used (sum-product decoding)

Example: (7,4) Hamming code with a binary symmetric erasure channel

Initialization:

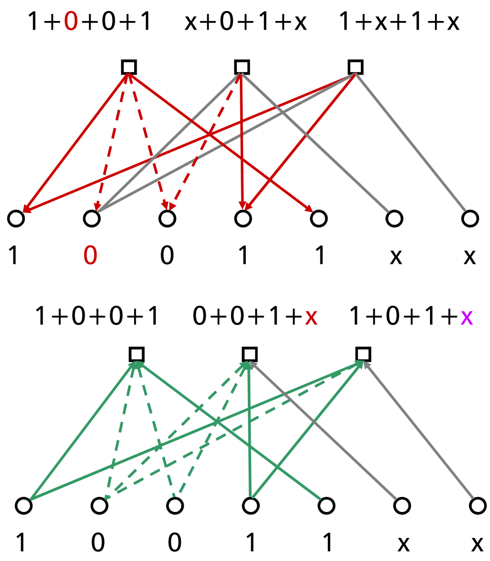


in order to be a valid code word, we want the syndrom to be zero.
Therefore, x must be 0.



3 Channel Coding

Message Passing (MP) decoding

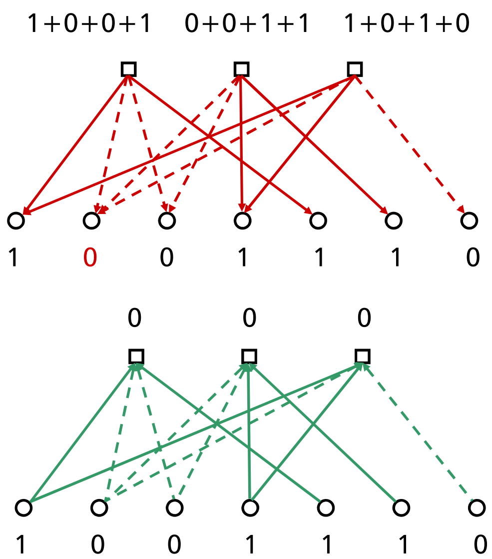


in order to be a valid code word, we want the sydrom to be zero.

Therefore, x must be 1 and x must also be 1.

3 Channel Coding

Message Passing (MP) decoding



Decoding result:
1 0 0 1 1 1 0

3 Channel Coding

Message Passing (MP) decoding

- sum-product decoding
- similar to the MAP Turbo decoding
- observations are used a a-priori information
- passed to the check nodes to calculate the parity bits, i.e., a-posteriory information / extrinsic information
- pass back the information from the parity bits as a-priori information for the next iteration
- actually, it has been shown, that the MAP decoding of Turbo codes is just a special case of LDPC codes already presented by Gallager

Robert G. Gallager,Professor Emeritus, Massachusetts Institute of Technology
und publications you'll also find his Ph.D. Thesis on LDPC codes
`http://www.rle.mit.edu/rgallager/`