Lecture 5
Channel Coding 2

M. Xiao
CommTh/EES/KTH

Lecture 5: Channel Coding 2
Advanced Digital Communications (EQ2410)[1]

M. Xiao
CommTh/EES/KTH

Monday, Feb. 3, 2016
10:00-12:00, B24

[1] Textbook: U. Madhow, *Fundamentals of Digital Communications*, 2008

Notes

Lecture 5
Channel Coding 2

M. Xiao
CommTh/EES/KTH

Overview

Lecture 4

- Low-density parity-check (LDPC) codes
- Iterative decoding on the factor graph
- Density evolution

Lecture 5: Modern Channel Coding 2

Notes

Overview

Information theory

- Random codes with infinite block length achieve the channel capacity.

Traditional block codes

- Fixed block length
- Algebraic code designs
- Often difficult to decode

Convolutional codes

- Encoding and decoding of sequences, no (or not necessarily) fixed block length.
- Efficient decoding with trellis based decoding (Viterbi or BCJR algorithm).
- Strong codes only with high constraint length ($\rightarrow$ high decoding complexity)

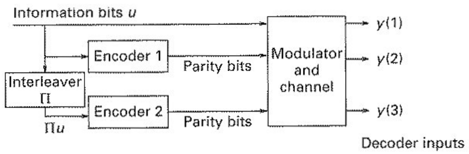
Traditional design goals

- Design codes with good distance properties (e.g., large minimum distance).
- Establish dependencies between a large number of bits (large memory).

Notes

Lecture 5
Channel Coding 2
M. Xiao
CommTh/EES/KTH

Turbo Codes
– Encoder Structure



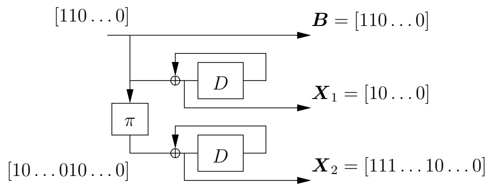
[U. Madhow, *Fundamentals of Dig. Comm.*, 2008]

- Two (or more) parallel component encoders encode permuted versions of the information word **u**.
- Component codes: recursive convolutional codes with rate $R_{c\,i}$ (often $R_{c\,i} = 1$).
- Interleaver
  - Pseudo-random permutation (Π) of the information bits u ($\rightarrow$ random coding).
  - Fixed interleaver length $\rightarrow$ Turbo codes are block codes.
  - Large memory, constraints between bits which are separated in time.
- Systematic bits and parity bits are multiplexed to the codeword of the Turbo code.
- Code rate $R = 1/(1 + \sum_i 1/R_{c\,i})$

Notes

Turbo Codes
– Encoder Structure

Example



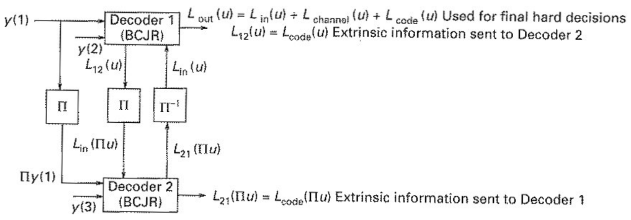
- Recursive codes have a long impulse response; they are necessary in order to be able to create high-weight codewords.
- Interleaver avoids that both codewords at the output of the component encoders have low weight.

→ Low-weight codewords can occur. However, the probability for this is low due to the interleaver.

Notes

KTH VETENSKAP OCH KONST
Lecture 5
Channel Coding 2
M. Xiao
CommTh/EES/KTH

Turbo Codes
– Decoder Structure



[U. Madhow, *Fundamentals of Dig. Comm.*, 2008]

- Component decoders
  - Two component decoders corresponding to the component encoders.
  - Soft-input/soft-output decoding; decoders generate *a posteriori* probabilities (APPs) for the information bits $\Pr(u|\mathbf{y})$.
- Factorization of the APP for bit u_i (Bayes' rule)

$$\Pr(u_i|\mathbf{y}) = \frac{\Pr(u_i,\mathbf{y})}{\Pr(\mathbf{y})} = \frac{\Pr(y_i|u_i)Pr(\mathbf{y}_{\setminus i}|u_i)\Pr(u_i)}{\Pr(\mathbf{y})}$$

- The same factorization expressed with log-likelihood ratios (LLRs)

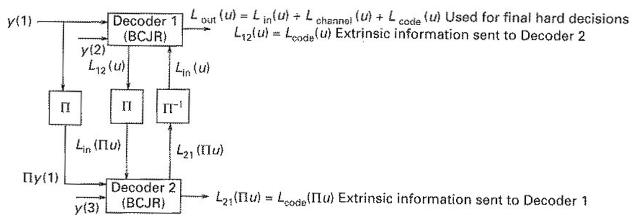
$$L_{out}(u) = \log\left(\frac{\Pr(u_i=0|\mathbf{y})}{\Pr(u_i=1|\mathbf{y})}\right) = \underbrace{L\left(\frac{\Pr(y_i|u_i=0)}{\Pr(y_i|u_i=1)}\right)}_{L_{channel}(u_i)} + \underbrace{L\left(\frac{\Pr(\mathbf{y}_{\setminus i}|u_i=0)}{\Pr(\mathbf{y}_{\setminus i}|u_i=1)}\right)}_{L_{code}(u_i)} + \underbrace{L\left(\frac{\Pr(u_i=0)}{\Pr(u_i=1)}\right)}_{L_{in}(u_i)}$$

Notes

Lecture 5
Channel Coding 2

M. Xiao
CommTh/EES/KTH

Turbo Codes
– Iterative Decoding



[U. Madhow, *Fundamentals of Dig. Comm.*, 2008]

- Output LLRs contain three contributions
  - $L_{in}(u)$: *a priori* distribution/information
  - $L_{channel}(u_i)$: LLR for the channel observation y_i of bit u_i (direct observation)
  - $L_{code}(u_i)$: extrinsic information on u_i provided by $\mathbf{y}_{\setminus i}$ (indirect observation); new information following from the code constraints.
- Decoding schedule
  - Run the decoders 1 and 2 and generate $L_{code}(u)$ for both decoders.
  - The decoders exchange the extrinsic information $L_{code}(u)$.
  - The extrinsic information $L_{code}(u)$ of the one decoder becomes after interleaving/de-interleaving the *a priori* information $L_{in}(u)$.
  - Start a new iteration and run the decoders again.

Notes

Lecture 5
Channel Coding 2

M. Xiao
CommTh/EES/KTH

Turbo Codes
– Example

1	0	1	0	0
0	1	1	1	1
0	1	1	0	0
1	0	0	0	1
0	0	0	1	

→ *BEC* →
(*binary erasure channel*)
7/24 bits erased
$C = 0.71$

x	0	1	0	0
x	1	x	1	1
0	x	x	0	0
1	0	x	x	1
0	0	0	1	

Parallel concatenated single-parity-check codes

- information word $\mathbf{B} = [1010\ 0111\ 0110\ 1000]$ is written in a matrix → *block interleaver*
- 2 single-parity-check codes (SPCCs) are applied to the columns and rows
- iterative decoding between the 2 SPCC decoders

1. *horiz. decoder*

1	0	1	0	0
x	1	x	1	1
0	x	x	0	0
1	0	x	x	1
0	0	0	1	

2. *vert. decoder*

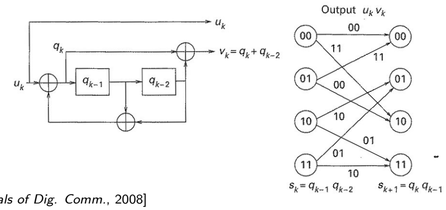
1	0	1	0	0
0	1	x	1	1
0	1	x	0	0
1	0	x	0	1
0	0	0	1	

3. *horiz. decoder*

1	0	1	0	0
0	1	1	1	1
0	1	1	0	0
1	0	0	0	1
0	0	0	1	

Notes

BCJR Algorithm



[U. Madhow, *Fundamentals of Dig. Comm.*, 2008]

- Named after the inventors (Bahl, Cocke, Jelinek, and Raviv).
- Soft-input/soft-output decoding algorithm for trellis codes; trellis based derivation of the *a posteriori* probabilities $\Pr(u|\mathbf{y})$.
- Observation: each state transition $s_k \to s_{k+1}$ defines uniquely a code symbol v_k and an information symbol u_k.
- APPs for the symbols u_k, v_k can be derived from the APPs of the state transitions $s' \to s$ (branch APPs)

$$\Pr(u_k = b|\mathbf{y}) = \frac{1}{Pr(\mathbf{y})} \sum_{(s',s)\in U_b} \Pr(s', s, \mathbf{y}) \text{ and } \Pr(v_k = b|\mathbf{y}) = \frac{1}{Pr(\mathbf{y})} \sum_{(s',s)\in V_b} \Pr(s', s, \mathbf{y})$$

with the sets U_b, V_b of state transitions (s', s) associated with the realization b of the respective bit.

Notes

BCJR Algorithm

- Factorization of the branch probability

$$\begin{aligned}\Pr(s', s, \mathbf{y}) &= \Pr(s_k = s', s_{k+1} = s, \mathbf{y}_1^{k-1}, y_k, \mathbf{y}_{k+1}^K) \\ &= \underbrace{\Pr(\mathbf{y}_{k+1}^K | s_{k+1} = s)}_{\beta_k(s)} \underbrace{\Pr(y_k, s_{k+1} = s | s_k = s')}_{\gamma_k(s', s)} \underbrace{\Pr(s_k = s', \mathbf{y}_1^{k-1})}_{\alpha_{k-1}(s')}\end{aligned}$$

  → $\beta_k(s)$ considers the future observations $\mathbf{y}_{k+1}^K$.
  → $\gamma_k(s', s)$ corresponds to the observation y_k of the current state transition (s', s).
  → $\alpha_{k-1}(s')$ considers the past observations $\mathbf{y}_1^{k-1}$.

- Forward recursion for deriving $\alpha_k(s)$

$$\begin{aligned}\alpha_k(s) &= \Pr(s_{k+1} = s, \mathbf{y}_1^k) = \sum_{s'} \Pr(s_{k+1} = s, s_k = s', y_k, \mathbf{y}_1^{k-1}) \\ &= \sum_{s'} \Pr(y_k, s_{k+1} = s | s_k = s') \Pr(s_k = s', \mathbf{y}_1^{k-1}) \\ &= \sum_{s'} \gamma_k(s', s)\alpha_{k-1}(s')\end{aligned}$$

with the initialization $\alpha_0(s') = 1$ for $s' = 0$ and $\alpha_0(s') = 0$ else.

Notes

Lecture 5
Channel Coding 2

M. Xiao
CommTh/EES/KTH

BCJR Algorithm

- Backward recursion for deriving $\beta_{k-1}(s')$ follows in a similar way:

$$\beta_{k-1}(s') = \sum_{s} \beta_k(s)\gamma_k(s', s)$$

with the initialization $\beta_K(s) = 1$ for $s = 0$ and $\alpha_0(s) = 0$ else.

- The γ-term can be derived as

$$\begin{aligned}\gamma_k(s', s) &= \Pr(y_k, s_{k+1}\!=\!s|s_k\!=\!s') \\ &\overset{(a)}{=} \Pr(y_k|s_{k+1}\!=\!s, s_k\!=\!s')\Pr(s_{k+1}\!=\!s|s_k\!=\!s') \\ &\overset{(b)}{=} \Pr(y_k|(u_k, v_k))\Pr(u_k)\mathbf{1}_{(s',s)}\end{aligned}$$

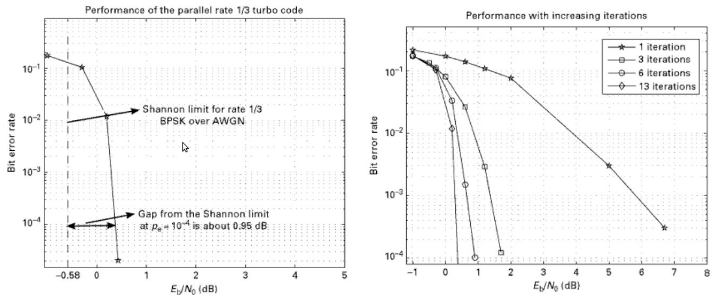
with $\mathbf{1}_{(s',s)} = 1$ if (s', s) is a valid state transition and $\mathbf{1}_{(s',s)} = 0$ else.
  - (a) follows from Bayes' rule,
  - (b) holds since (s', s) uniquely defines (u_k, v_k) and since for a given state s' the transition to s is driven by the information bit u_k.
- Implementation: BCJR is implemented in log-domain to avoid numerical instabilities due to low numbers.

Notes

Lecture 5
Channel Coding 2

M. Xiao
CommTh/EES/KTH

Performance Analysis
– Simulation



[U. Madhow, *Fundamentals of Dig. Comm.*, 2008]

Example

- Rate-1/3 Turbo code with recursive convolutional codes $(5, 7)_8$
- Waterfall region: can be predicted by analyzing the convergence of the iterative decoding.
- Error-floor region (at high SNR): analysis of the distance properties and union bound.

Notes

Performance Analysis
– Union Bound

- With the input/parity-weight enumerator function $A_{turbo}(w, p)$:

$$
\begin{aligned}
P_e &\leq \sum_w \sum_p \frac{w}{K} A_{turbo}(w,p) Q\left(\sqrt{\frac{2E_bR(w+p)}{N_0}}\right) \\
&\overset{(a)}{\leq} \sum_w \sum_p \frac{w}{K} A_{turbo}(w,p) e^{-\frac{E_bR}{N_0}w} e^{-\frac{E_bR}{N_0}p} \\
&\overset{(b)}{\leq} \sum_w \frac{w}{K} W^w A_{turbo}(P|w)\Big|_{W=P=e^{-\frac{E_bR}{N_0}}}
\end{aligned}
$$

and by using

(a) the upper bound $Q(x) \leq e^{-\frac{x^2}{2}}$;
(b) the conditional parity weight enumerator function

$$A_{turbo}(P|w) = \sum_p A_{turbo}(w,p) P^p$$

.

$\rightarrow$ How to derive $A_{turbo}(P|w)$?

Notes

Lecture 5
Channel Coding 2
M. Xiao
CommTh/EES/KTH

Performance Analysis
– Union Bound

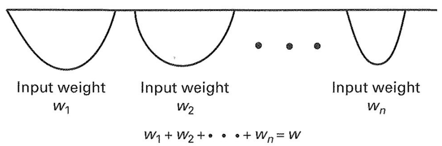
- Assumption: random interleaver that maps one weight-w info word into another weight-w info word with uniform probability $1/\binom{K}{w}$.
- With the conditional weight enumerator functions $A_1(P|w), A_2(P|w)$ of the component encoders we get:

$$A_{turbo}(P|w) = \frac{A_1(P|w)A_2(P|w)}{\binom{K}{w}}$$

- Approximation of the conditional parity weight enumerator function for convolutional codes:

$$A(P|w) \approx \sum_{n=1}^{n_{max}} A(P|w,n) \binom{K}{n}, \quad n_{max} \leq w \leq K$$

where $A(P|w,n)$ is the parity weight enumerator function for input weight w and codewords consisting of n error events.



Notes

Performance Analysis
– Union Bound

- By combining the results and using the approximation $\binom{K}{l} \approx K^l/l!$ for large K, we get:

$$A_{turbo}(P|w) \approx \sum_{n_1=1}^{n_{max}} \sum_{n_2=1}^{n_{max}} \frac{\binom{K}{n_1}\binom{K}{n_2}}{\binom{K}{w}} A_1(P|w,n_1)A_2(P|w,n_2)$$
$$\approx \sum_{n_1=1}^{n_{max}} \sum_{n_2=1}^{n_{max}} \frac{w!}{n_1!n_2!} K^{n_1+n_2-w} A_1(P|w,n_1)A_2(P|w,n_2)$$
$$\approx \frac{w!}{(n_{max}!)^2} K^{2n_{max}-w} A_1(P|w,n_{max})A_2(P|w,n_{max})$$

and with $A_1(P|w,n) = A_2(P|w,n) = A(P|w,n)$

$$P_e \lesssim \sum_{w=w_{min}}^{K} wW^w \frac{w!}{(n_{max}!)^2} K^{2n_{max}-w-1} A(P|w,n_{max})^2 \Big|_{W=P=e^{-\frac{E_bR}{N_0}}}$$

Dominating term at high SNR: $w = w_{min}$

- Interleaver gain if $P_e \sim K^{-k}$, $k > 0$.

Notes

Performance Analysis
– Union Bound

Non-recursive codes

- We have $w_{min} = 1$, $n_{max} = w$, and $A(P|w,n_{max}) = A(P|1,1)^w$, and it follows that

$$P_e \lesssim \sum_{w=1}^{K} \frac{K^{w-1}}{(w-1)!} W^w A(P|1,1)^{2w} \Big|_{W=P=e^{-\frac{E_bR}{N_0}}}.$$

→ The dominant term in the sum is $w = 1$.

- Observation: the performance does not improve with increasing block length K, no interleaver gain!

Recursive codes

- We have $w_{min} = 2$, $n_{max} = \lfloor w/2 \rfloor$
  - Case 1: $w = 2k \rightarrow n_{max} = k \rightarrow$ dominating term decays with K^{-1}. Furthermore, $A(P|w,n_{max}) = A(P|2k,k) = A(P|2,1)^k$.
  - Case 2: $w = 2k+1 \rightarrow n_{max} = k \rightarrow$ odd input weights decay with K^{-2} and can be neglected; $A(P|w,n_{max}) = A(P|2k+1,k)$.
- Case 1 gives us the effective free distance $p_{eff} = 2 + 2p_{min}$.
- Conclusion: Turbo codes have a low minimum free distance but the interleaver gain ($P_e \sim K^{-1}$) guarantees error probabilities around $10^{-5} \ldots 10^{-6}$.

Notes