Turbo Codes

Markus Feuerstein, Nikolaus Mutsam

Signal Processing and Speech Communication Laboratory
TU Graz

6-12-2010

Overview

Shannon Limit

- Claude Shannon, 1948
- Shannon Limit is the theoretical maximum information transfer rate of the channel for a particular noise level.
- Requires optimal channel encoding (hypothetical)

Shannon Limit

Channel capacity

$$C_S = B\, ld(1 + \frac{S}{N_0 B}) = B\, ld(1 + \frac{C_R}{B}\frac{E_b}{N_0})$$

$$\lim_{B \to \infty} C_S = \frac{S}{N_0} ld(e) \approx 1.44 \frac{S}{N_0}$$

Channel codes

Two common types of channel codes:

- **Block codes**
- **Convolutional codes**
- Systematic and non-systematic codes
- redundancy is added for error detection and correction

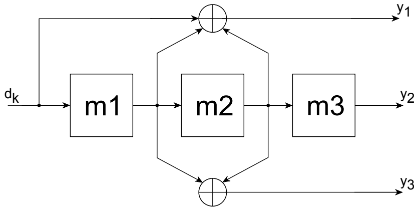
Block codes

- transforms a message into a sequence called codeword
- all codewords consist of the same amount n of symbols of a given alphabet
- in a linear block code each input message has a fixed length $k < n$
- code rate $R = \frac{k}{n}$ describes the redundancy

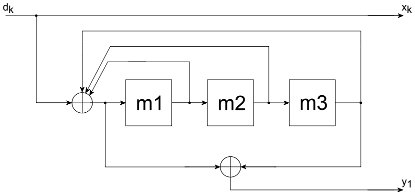
Convolutional Codes

- have a memory
- consecutive code words are not independent
- recursive and non-recursive codes
- decoding is based on estimation
- most convolutional codes are catastrophic

Convolutional Codes



(a) non-recursive

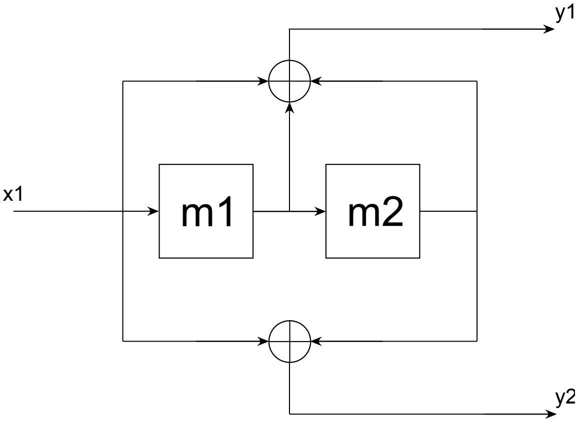


(b) recursive

Design of Convolutional Codes

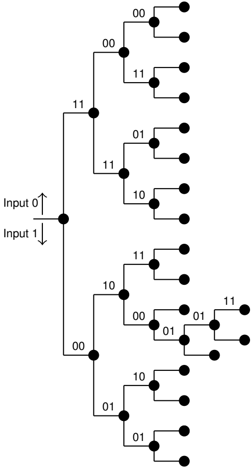
- no systematic design scheme
- relevant designs are usually found by computer simulations
- most designs lead to catastrophic convolutional codes

Encoder - Generator Polynomials

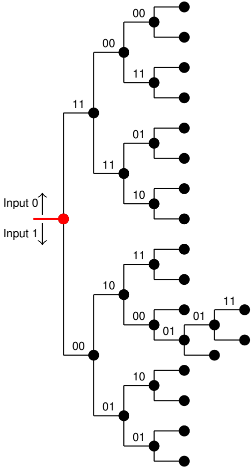


generator polynomials in octal notation: g1:7, g2:6

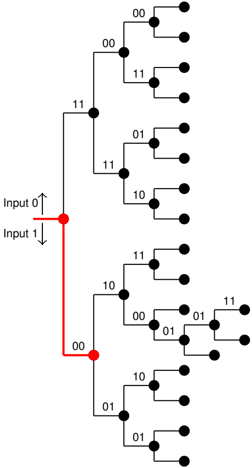
Encoder - Tree Diagram



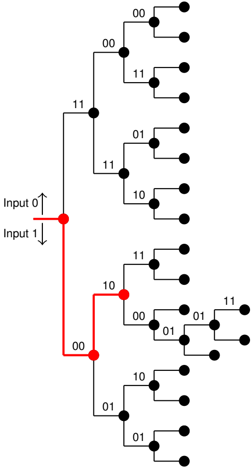
Encoder - Tree Diagram



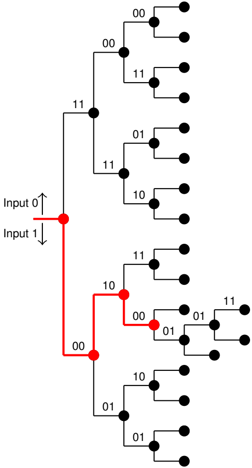
Encoder - Tree Diagram



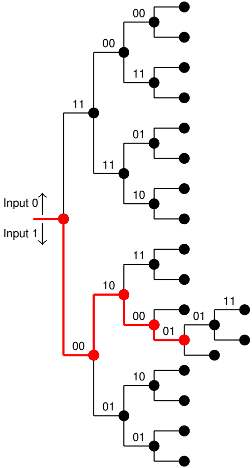
Encoder - Tree Diagram



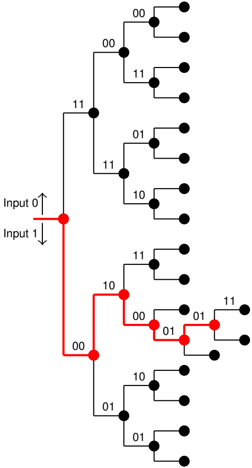
Encoder - Tree Diagram



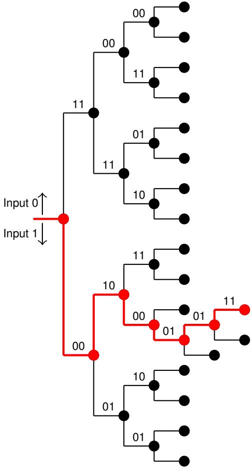
Encoder - Tree Diagram



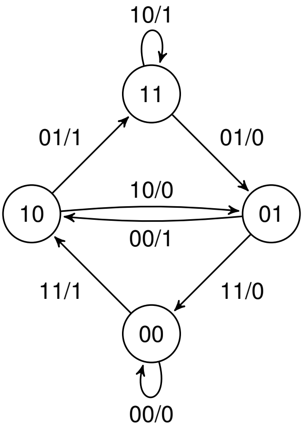
Encoder - Tree Diagram



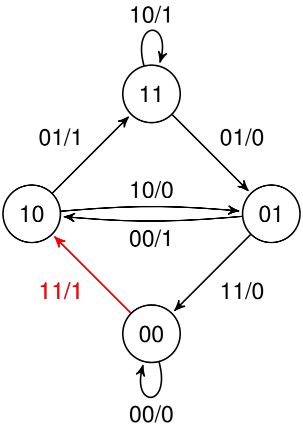
Encoder - Tree Diagram



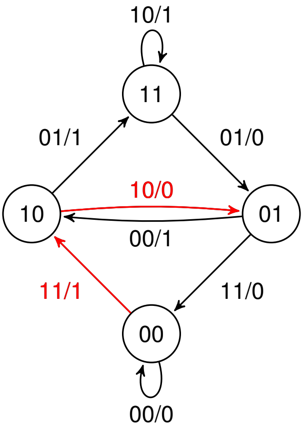
Encoder - State Diagram



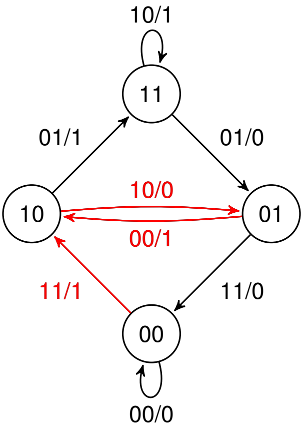
Encoder - State Diagram



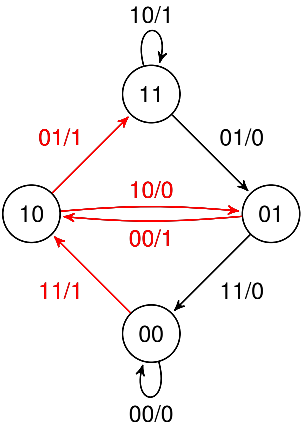
Encoder - State Diagram



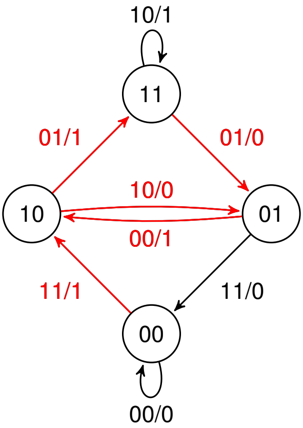
Encoder - State Diagram



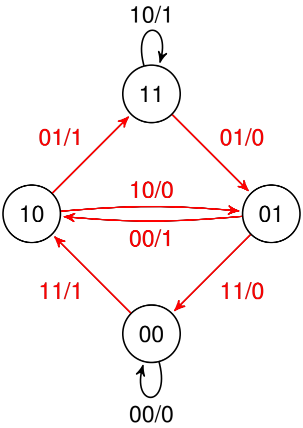
Encoder - State Diagram



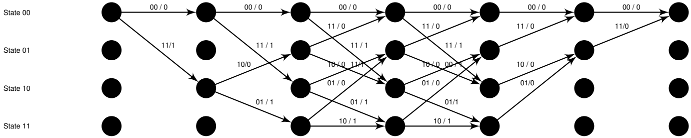
Encoder - State Diagram



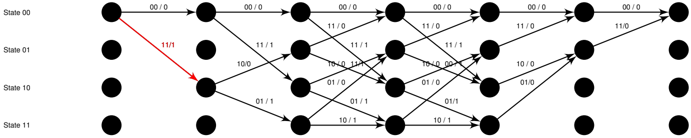
Encoder - State Diagram



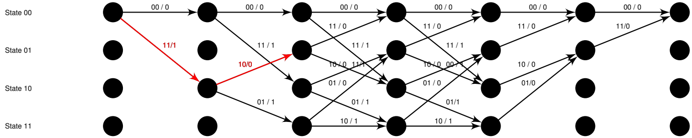
Encoder - Trellis Diagram



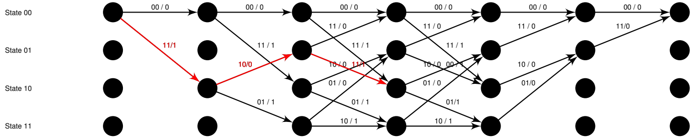
Encoder - Trellis Diagram



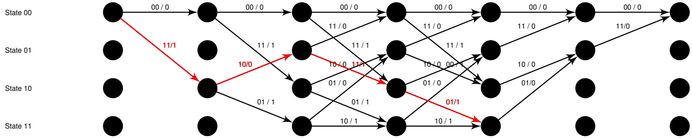
Encoder - Trellis Diagram



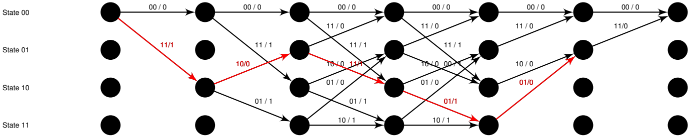
Encoder - Trellis Diagram



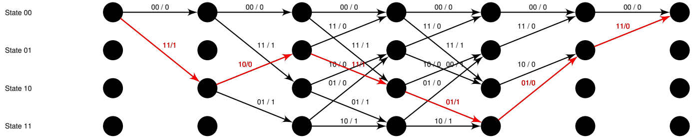
Encoder - Trellis Diagram



Encoder - Trellis Diagram



Encoder - Trellis Diagram



Encoder - Puncturing

- removing some of the parity bits after encoding
- decrease redundancy - increase code rate R
- encoder and decoder have to know the puncturing pattern

Encoder - Interleaving

- construct a permutation of the input sequence
- protection from burst error
- interleaver design has a big influence on code performance
- increases latency
- many different interleavers: matrix, random, semirandom, odd-even, circular shift,...

Encoder - Interleaving Example

transmission without interleaving

ThisIsAnExampleOfInterleaving...
ThisIs______pleOfInterleaving...

transmission with interleaving

ThisIsAnExampleOfInterleaving...
TIEpfeaghsxIlrv.iAaenli.snmOten.
TIEpfe______Irv.iAaenli.snmOten.
T_isI_AnE_amp_eOfInterle_vin_...

Encoder - Interleaving Example

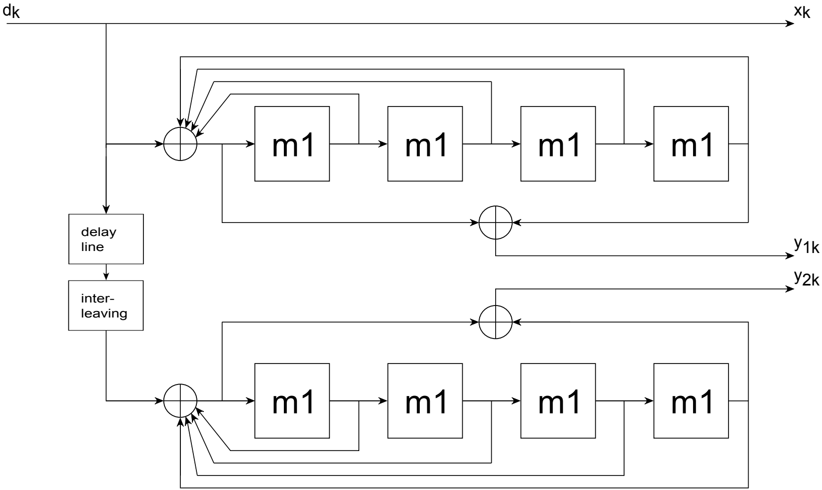
transmission without interleaving

ThisIsAnExampleOfInterleaving...
ThisIs______pleOfInterleaving...

transmission with interleaving

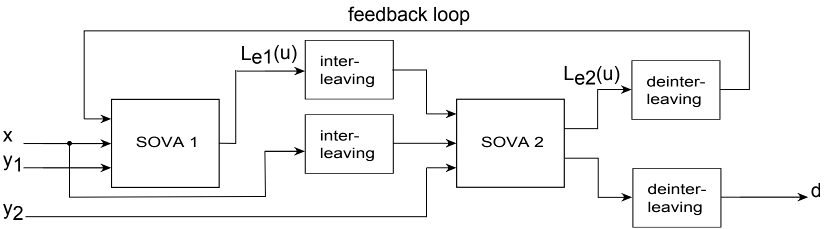
ThisIsAnExampleOfInterleaving...
TIEpfeaghsxlIrv.iAaenli.snmOten.
TIEpfe______Irv.iAaenli.snmOten.
T_isI_AnE_amp_eOfInterle_vin_...

Encoder - Turbo Code



Decoder

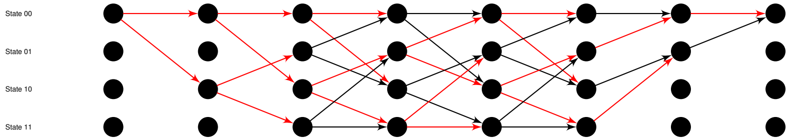
"historic" decoding scheme



Decoder - Viterbi Algorithm

- find symbol sequence with maximum likelihood
- soft decision: reliability of each symbol
- hard decision: estimated discrete symbol sequence
- applications: cell phones, DVB, satellite communication

Decoder - Trellis Diagram

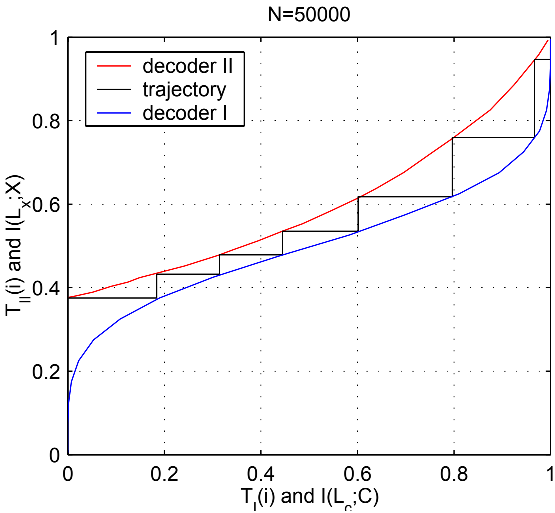


$$\text{Metric: } M_t^{(m)} = M_{t-1}^{(m)} + u_t^{(m)} \{L_c y_{S,t}\} + x_{1,t}^{(m)} L_c y_{1,t} + u_t^{(m)} L_{c2}(u_t)$$

EXIT - Charts

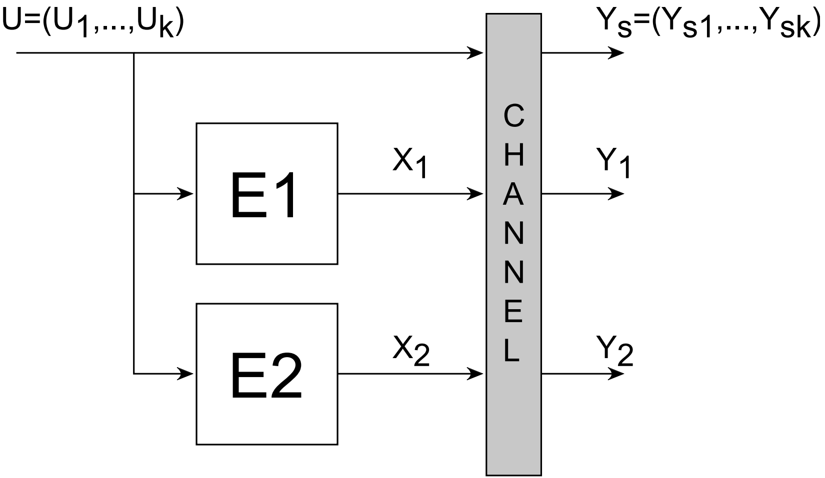
- **EX**trinsic **I**nformation **T**ransfer
- visualization of the asymptotic performance of a decoder where two components exchange extrinsic information

EXIT - Charts


N=50000
decoder II
trajectory
decoder I
0
0.2
0.4
0.6
0.8
1
$T_{II}(i)$ and $I(L_x;X)$
$T_I(i)$ and $I(L_c;C)$

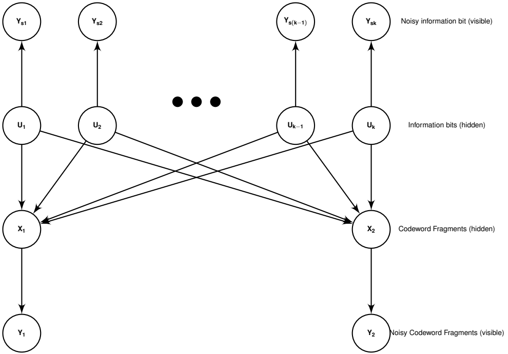
Decoder - Pearl's Belief

Coding Scheme



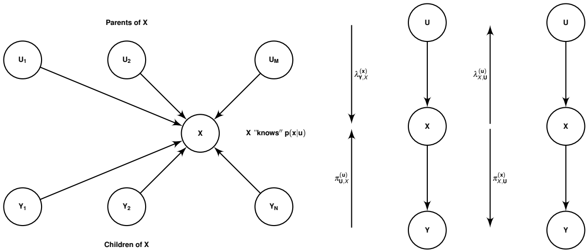
Decoder - Pearl's Belief

Bayesian Network interpretation



Decoder - Pearl's Belief

message passing



Decoder - Pearl's Belief

messages:

$$\pi_{\mathbf{U},X}(\mathbf{u}) = (\pi_{U_1,X}(u_1), \ldots, \pi_{U_M,X}(u_M))$$
$$\lambda_{\mathbf{Y},X}(x) = (\lambda_{Y_1,X}(x), \ldots, \lambda_{Y_N,X}(x))$$
$$\pi_{X,\mathbf{Y}}(x) = (\pi_{X,Y_1}(x), \ldots, \pi_{X,Y_N}(x))$$
$$\lambda_{X,\mathbf{U}}(\mathbf{u}) = (\lambda_{X,U_1}(u_1), \ldots, \lambda_{X,U_M}(u_M))$$

Internal quantities: $\mu_X(\mathbf{u})$, $\gamma_X(\mathbf{u})$

Soft decision: $BEL_X(x)$

Pearl's Belief - Initialization

	quantity (at X)	initially (evid.) $x = x_0$	initially (non. evid.)
1.	$\mu_X(\mathbf{u})$	–	–
2.	$\lambda_X(x)$	–	–
3.	$\pi_X(x)$	$\delta(x, x_0)^*$	$\begin{cases} p(x)^* & \text{if X is a source node} \\ -- & \text{otherwise} \end{cases}$
4.	$\gamma_X(\mathbf{u})$	$p(x_0 \mid \mathbf{u})$	1
5.	$BEL_X(x)$	$\delta(x, x_0)^*$	$\begin{cases} p(x) & \text{if X is a source node} \\ -- & \text{otherwise} \end{cases}$
6.	$\lambda_X, U(\mathbf{u})$	$\begin{cases} p(x_0 \mid u)^* & \text{if M = 1} \\ -- & \text{otherwise} \end{cases}$	1
7.	$\pi_{X,\mathbf{Y}}(x)$	$\delta(x, x_0)^*$	$\begin{cases} p(x) & \text{if X is a source node} \\ -- & \text{otherwise} \end{cases}$

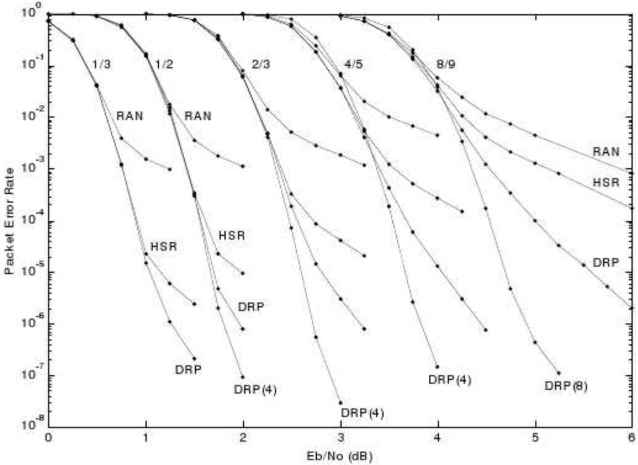
Pearl's Belief - Updates

	quantity (at X)	Type	Update Rule
1.	$\mu_X(\mathbf{u})$	A_U1 x...x $A_UM \to R^M$	$< \pi_{\mathbf{U},X}(\mathbf{u})$¿ (1 if X has no parents)
2.	$\lambda_X(x)$	$A_X \to R$	$< \lambda_{\mathbf{Y},X}(x)$¿ (1 if X has no children)
3.	$\pi_X(x)$	$A_X \to R$	$\sum_{\mathbf{u}} p(x\|\mathbf{u})\mu_X(\mathbf{u})$ (p(x) if X has no parents)
4.	$\gamma_X(\mathbf{u})$	A_U1 x...x $A_UM \to R$	$\sum_x \lambda_X(x)p(x\|\mathbf{u})$
5.	$BEL_X(x)$	$A_X \to R$	$\alpha\lambda_X(x)\pi_X(x)$
6.	$\lambda_X, U(\mathbf{u})$	A_U1 x...x $A_UM \to R^M$	$\pi_{\mathbf{U},x} \circ \gamma_X$
7.	$\pi_{X,\mathbf{Y}}(x)$	$A_X \to R^N$	$\pi_X(x)\prod_{i=1,i\neq j}^N \lambda_{Y_i,X}(x)$

Pearl's Belief - Convergence

- many loops in the belief network
- no guarantee for convergence for graphs with loops - stopping criterion needed
- convergence is only proven for loop-free graphs
- in many cases loops do not compromise the performance

Performance



Contact

Markus Feuerstein	markus.feuerstein@student.tugraz.at
Nikolaus Mutsam	nikolaus.mutsam@student.tugraz.at