Paper presentation – Ultra-Portable Devices

Paper:

J. Hagenauer, E. Offer, C. Measson, M. Mörz.

Decoding and Equalization with Analog Non-linear Networks.

EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS, EUREL PUBLICATION, MILANO, vol. 10, no. 6, November 1998 (1998-11), pages 659-680.

Presented by:

Reza Meraji

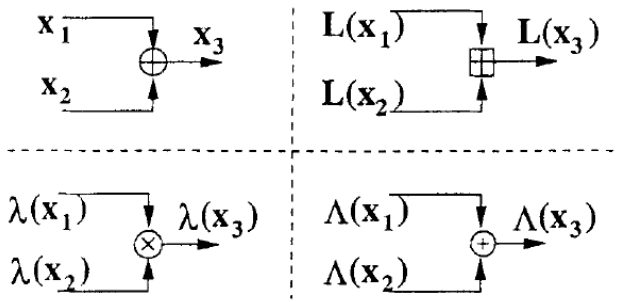
Outline

- Log-Likelihood Algebra
- Principles for the analog decoding design
- Factor graphs
- Building blocks for analog decoder networks
- Components for analog implementation
- Simple examples for analog decoders
- Networks based on simple analog networks for more complicated codes
- Summary

Log-Likelihood Algebra and Network elements

$$L(x) = \log \frac{P(x=+1)}{P(x=-1)}$$

$$\lambda(x) = \mathrm{E}\{x\} = (+1)\frac{e^{+L(x)}}{1+e^{+L(x)}} + (-1)\frac{e^{-L(x)}}{1+e^{-L(x)}}$$



Factor Graph

A graph that expresses how a 'global' function of many Variables factors into a product of 'local' functions

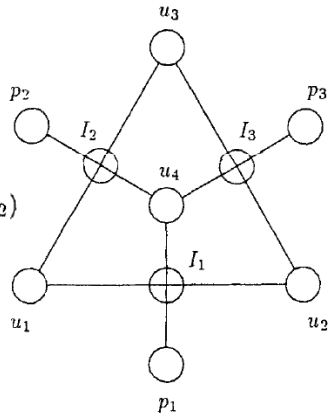
- Variable nodes
- Function nodes

Helpful in describing channel decoding algorithms

Hamming Code

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

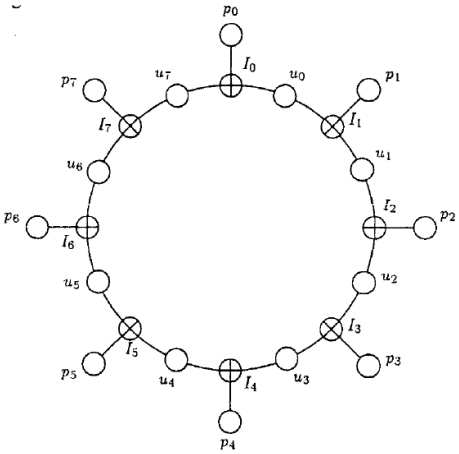
$$I_{\mathcal{C}_H}(\mathbf{x}) = I_1(u_1, u_2, u_4, p_1) \cdot I_2(u_1, u_3, u_4, p_2) \cdot I_3(u_2, u_3, u_4, p_3)$$



Memory one convolutional code

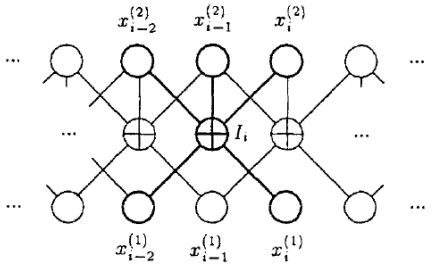
- Rate ½
- Code length 16
- Tailbiting structure

$$\mathbf{H}_{m1} = \begin{pmatrix} 1\,1 & & & & & & & 1\,0 \\ 1\,0\,1\,1 & & & & & & & \\ & 1\,0\,1\,1 & & & & & & \\ & & 1\,0\,1\,1 & & & & & \\ & & & 1\,0\,1\,1 & & & & \\ & & & & 1\,0\,1\,1 & & & \\ & & & & & 1\,0\,1\,1 & & \\ & & & & & & 1\,0\,1\,1 & \end{pmatrix}$$



Memory two convolutional code

- Rate ½
- [7,5] encoder
- N information bits



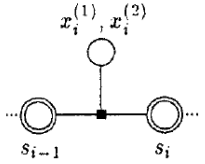
$$\mathbf{H}_{m2} = \begin{pmatrix} 1\,1 & & & & & & & 1\,1\,0\,1 \\ 0\,1\,1\,1 & & & & & & & 1\,1 \\ 1\,1\,0\,1\,1\,1 & & & & & & & \\ & 1\,1\,0\,1\,1\,1 & & & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & & & 1\,1\,0\,1\,1\,1 & & \\ & & & & & & 1\,1\,0\,1\,1\,1 & \end{pmatrix}$$

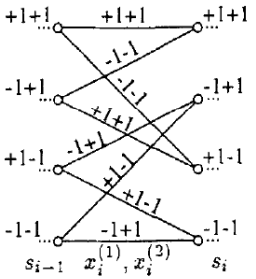
Trellis representation

- Tanner graphs:

  graphs with hidden or auxiliary variables and variables which are not necessarily binary

- States as hidden variables





Tailbiting convolutional codes

- Achieve the minimum distance of block codes for short and medium block sizes
- No need for termination bits
- Less complexity for analog decoder
- Certain constraint for block length
  - Ex. For memory two and rate = ½: Need for a circle size of at least 11 information bits

Code Block Length

- Long block lengths are a great challenge for the analog technique
- Analog decoding on short block length might be used as an inner decoder system to perform a first and quick clean up of errors

Building blocks for analog decoder networks

- BJT

$$I_C = I_S e^{\frac{qV_{BE}}{kT}}$$

- Gilbert Cell: a well known multiplier
- Boxplus operation:

  Differential output current of Gilbert Cell made of BJT transistors

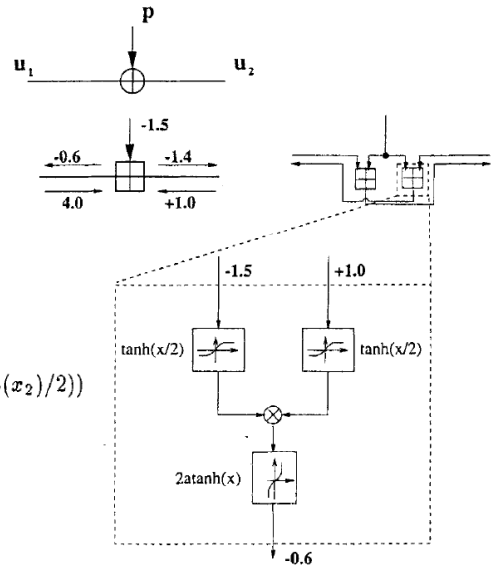
Building blocks for analog decoder networks, continue

- Another function needed in node variables: Algebraic sum of L-values
- Summing voltages: opamp with external resisotrs -> not suitable for analog VLSI
- An alternative: Using the Gilbert cells -> great advantage for layout
- Boxplus and variable sum circuits: each 9 transistors

Bulding Blocks: example

- Block length 3
- Check equation:

$$u_1 \oplus u_2 = p$$

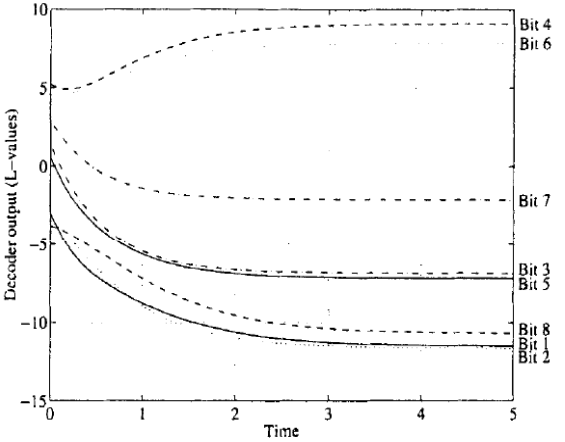


$$L(x_1 \oplus x_2) = \log \frac{1 + e^{L(x_1)} e^{L(x_2)}}{e^{L(x_1)} + e^{L(x_2)}}$$
$$= 2\,\mathrm{atanh}(\tanh(L(x_1)/2) \cdot \tanh(L(x_2)/2))$$

$$L(x_1) \boxplus L(x_2) \triangleq L(x_1 \oplus x_2)$$

Simple examples for analog decoders

- Memory 1
- Rate = ½
- Tailbiting convolutional code
- Circle size: 8 information bits



- u = (-1, -1, -1, +1, -1, +1, -1, -1)

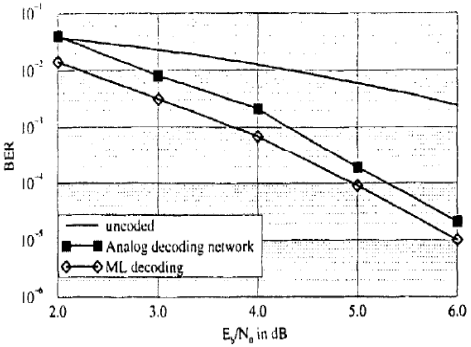
Simple examples for analog decoders

- A decoder network with two or three edges leaving one variable node needs additional information
- Case for Hamming code and memory two convolutional code

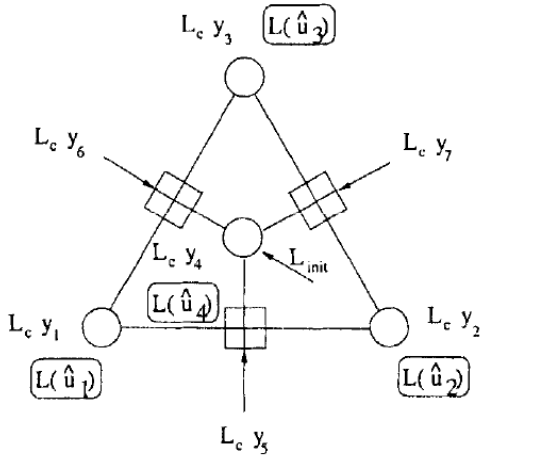
$$L(\hat{u}_i)_{\text{init}} = a \cdot \left(L_c y_{i+1}^{(1)} \boxplus L_c y_{i+1}^{(2)} \right)$$

Simple examples for analog decoders

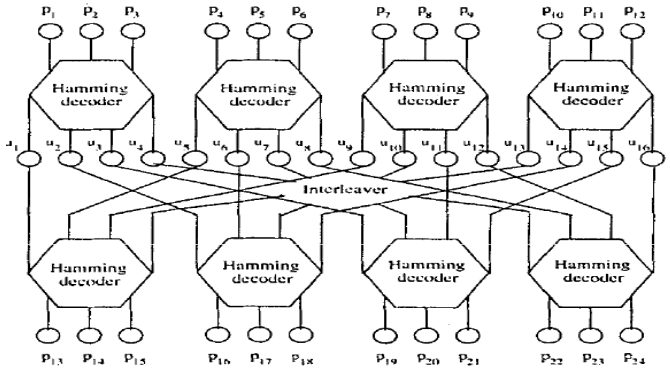
- Memory 2
- Rate = ½
- Tailbiting convolutional code
- Circle size: 16 information bits
- Eq. 35



Analog decoder for Hamming Code



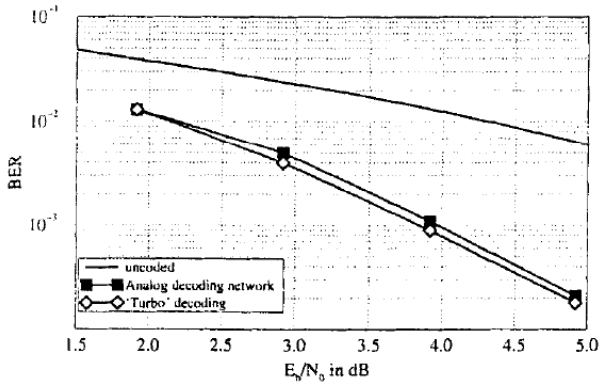
Hamming Turbo code



4 x 4 information bits in block interleaver

u_1	u_2	u_3	u_4	p_1	p_2	p_3
u_5	u_6	u_7	u_8	p_4	p_5	p_6
u_9	u_{10}	u_{11}	u_{12}	p_7	p_8	p_9
u_{13}	u_{14}	u_{15}	u_{16}	p_{10}	p_{11}	p_{12}
p_{13}	p_{16}	p_{19}	p_{22}			
p_{14}	p_{17}	p_{20}	p_{23}			
p_{15}	p_{18}	p_{21}	p_{24}			

Hamming Turbo Simulations



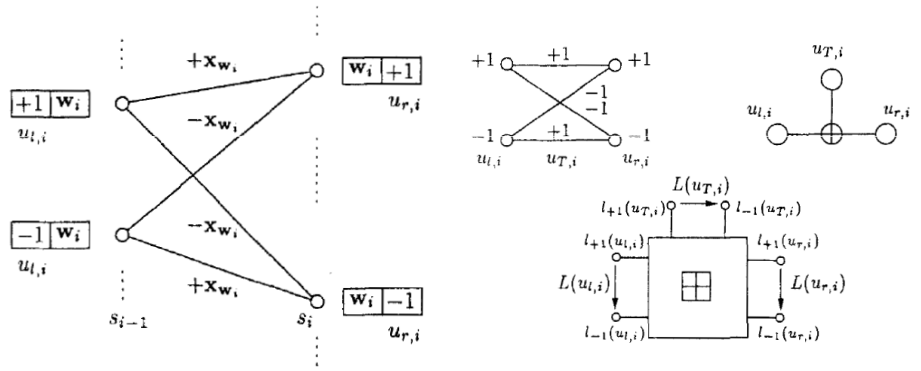
Codes with higher number of connections

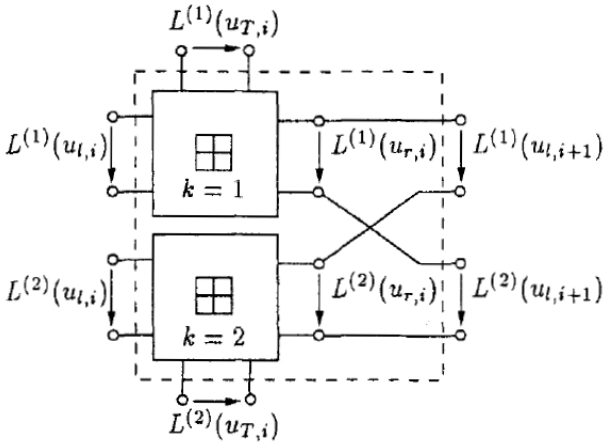
- ML decoding performance is only achieved when variable nodes are sparsely connected
- In general, no great achievement out of simulations
- However, most efficient coding schemes are characterized by sparsely connected variable nodes

Analog decoder networks based on Trellises

- Extension of box-plus operations
- As long as a trellis of the code preserves a ”butterfly” structure
- In this case message passing is equivalent to forward-backward algorithm (BCJR)

Analog decoder networks based on Trellises





Summary

- Analog, non-linear and highly parallel decoder networks instead of digital processors
- Box-plus element as the key element corresponding to binary addition in GF(2)
- Non-linearities of bipolar transistors utalized to build the key elements
- From factor graphs to simple analog VLSI implementations
- Soft values are needed as inputs
- Decoder network may not converge
- Mapping of binary trellis onto box-plus circuits