Analog Decoding

JASS’05
St. Petersburg
<Pavol.Hanus@tum.de>
Contents

• Introduction
  – History, Motivation, Advantages

• Basics
  – Analog Implementation of the Log-likelihood Algebra

• Analog Decoder
  – Graph based, Trellis based

• Analog Turbo Decoder
  – Interleaver as Cross-connecting Network, “Flooding"

• Conclusion
  – Results, Open Problems, Future Work
Historic Overview

- **Idea**
  - J. Hagenauer 1997 (TUM)

- **Implementation Aspects**
  - A. Loeliger, F. Lustenberger 1998 (ETH Zürich)

- **First Functioning Decoder Chip**
  - Matthias Mörz 1998/99 (TUM, Bell Labs)

- **Research Teams**
  - TUM, University of Utah, University of Alberta, University of Toronto, ETH Zürich, Politecnico di Torino, Università di Padova
  - Industry, Companies
Motivation

- World of „digital“ transmission is analog
  - Information is transmitted in form of analog waves
- Soft-in/Soft-out algorithms (SISO) use analog values
  - SOVA, APP, LDPC, …
  - Only analog is really soft
- Speed
  - Decoding speed limited only by the settling time
- Small Area
- Low Power Consumption
- Simple Circuit Design
  - Repetition of a few elementary circuits
Motivation for using Log-likelihood Algebra

A-posteriori LLR

\[ L(x \mid y) = L_c y + L(x) \]

Channel-state Information (Gaussian Fading Channel)

\[ L_c = 4a \frac{E_s}{N_o} \]

\[ x : \text{transmitted symbol} \]

\[ y : \text{Matched Filter-output} \]

\[ a : \text{Fading-Amplitude} \]
## Binary Log-likelihood Algebra Quantities using Analog Circuits (Voltages and Currents)

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Formula</th>
<th>Analog ((V,I))</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLRatio</td>
<td>(L(X) = \ln \frac{P_X (x = 0)}{P_X (x = 1)})</td>
<td>?</td>
</tr>
<tr>
<td>Probability</td>
<td>(P_X (x = 0) = \frac{1}{1 + e^{-L(X)}})</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>(P_X (x = 1) = \frac{e^{-L(X)}}{1 + e^{-L(X)}})</td>
<td></td>
</tr>
<tr>
<td>Soft-bit</td>
<td>(\lambda(X) = E{X} = \tanh \left[ \frac{L(X)}{2} \right])</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>(= (+1)P_X (x = 0) + (-1)P_X (x = 1))</td>
<td></td>
</tr>
</tbody>
</table>
Basic Circuits: Bipolar Junction Transistor (NPN-BJT)

BJT in forward active mode

\[ I_C = I_S \left( e^{\frac{V_{BE}}{V_T}} - 1 \right) \approx I_S \cdot e^{\frac{V_{BE}}{V_T}} \]

\[ V_T = \frac{kT}{q} \approx 26mV \] [Thermal Voltage]

\[ I_S \approx 10^{-16} \text{A} \] [Saturation Current]

BJT as a PN Diode

\[ I = I_S \cdot e^{\frac{V}{V_T}} \]

\[ V = V_T \ln\left( \frac{I}{I_S} \right) \]
Basic Circuits: Transistor Pairs

**Pair of Diodes**

\[
\frac{\Delta V}{V_T} = \frac{V_0 - V_1}{V_T} = 2 \tanh^{-1}\left(\frac{\Delta I}{I}\right) = L(X)
\]

\[
P \Rightarrow L
\]

**Differential Pair**

\[
\begin{align*}
\frac{I_1}{I} &= \frac{e^{-\Delta V/V_T}}{1 + e^{-\Delta V/V_T}} = P_x (x = 1) \\
\frac{I_0}{I} &= \frac{1}{1 + e^{-\Delta V/V_T}} = P_x (x = 0) \\
\frac{\Delta I}{I} &= \frac{I_0 - I_1}{I} = \tanh\left(\frac{1}{2} \frac{\Delta V}{V_T}\right) = \lambda(X)
\end{align*}
\]

\[
L \Rightarrow P
\]
## Binary Log-likelihood Algebra Quantities using Analog Circuits (Voltages and Currents)

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Formula</th>
<th>Analog ((V,I))</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLRatio</td>
<td>[ L(X) = \ln \frac{P_X(x = 0)}{P_X(x = 1)} ]</td>
<td>[ L(X) = \frac{\Delta V}{V_T} ]</td>
</tr>
<tr>
<td>Probability</td>
<td>[ P_X(x = 0) = \frac{1}{1 + e^{-L(X)}} ]  [ P_X(x = 1) = \frac{e^{-L(X)}}{1 + e^{-L(X)}} ]</td>
<td>[ P_X(x = i) = \frac{I_i}{I} ]</td>
</tr>
<tr>
<td>Soft-bit</td>
<td>[ \lambda(X) = E{X} = \tanh \left[ \frac{L(X)}{2} \right] ] [ = (+1)P_X(x = 0) + (-1)P_X(x = 1) ]</td>
<td>[ \lambda(X) = \frac{\Delta I}{I} ]</td>
</tr>
</tbody>
</table>
Binary Log-likelihood Algebra Operations using Analog Circuits (Voltages and Currents)

**BOX-Plus**

\[
L(X_3) = L(X_1) \oplus L(X_2) = \ln \frac{P_X(x_3 = 0)}{P_X(x_3 = 1)} =
\]

\[
= \ln \frac{(P_X(x_1 = 0)P_X(x_2 = 0) + P_X(x_1 = 1)P_X(x_2 = 1))}{(P_X(x_1 = 0)P_X(x_2 = 1) + P_X(x_1 = 1)P_X(x_2 = 0))}
\]

**L-Values Addition**

\[
L(X_3) = L(X_1) + L(X_2) = \ln \frac{P_X(x_1 = 0)}{P_X(x_1 = 1)} + \ln \frac{P_X(x_2 = 0)}{P_X(x_2 = 1)} =
\]

\[
= \ln \frac{(P_X(x_1 = 0)P_X(x_2 = 0))}{(P_X(x_1 = 1)P_X(x_2 = 1))}
\]
Basic Circuits: Probability Multiplication

\[ \frac{\Delta V_i}{V_T} = L(X_i) \]
\[ \frac{I_{i,j}}{I} = P_X(x_1 = i) \cdot P_X(x_2 = j) \]
Basic Circuits: Probability Addition

\[ P(X_1 = 0)P(X_2 = 0) + P(X_1 = 1)P(X_2 = 1) = \frac{I_{00} + I_{11}}{I} \]

\[ P(X_1 = 0)P(X_2 = 1) + P(X_1 = 1)P(X_2 = 0) = \frac{I_{01} + I_{10}}{I} \]
\[ \frac{\Delta V_i}{V_T} = L(X_i) \]

\[ L(X_3) = L(X_1) \oplus L(X_2) \]

\[ \frac{\Delta V_3}{V_T} = 2 \cdot \tanh^{-1} \left[ \tanh \left( \frac{\Delta V_1}{2V_T} \right) \cdot \tanh \left( \frac{\Delta V_2}{2V_T} \right) \right] \]
Basic Circuits: L-values Addition

\[
\frac{\Delta V_i}{V_T} = L(X_i)
\]

\[
L(X_3) = L(X_1) + L(X_2)
\]
Decoder Core Building Block with I/O

\[ L(X_3) = L(X_1) \quad \boxed{\text{or}} \quad L(X_2) \]

\[ L(X_3) = L(X_1) + L(X_2) \]
Soft-in / Soft-out Decoder  
(only analog is really soft)

Component decoder used in a turbo scheme

All values are Log-likelihood ratios (real numbers)

- Digital implementation works with quantized values
- Analog Decoder works with voltages and currents
- Analog sliding window decoder as component decoder
Analog Turbo Decoder
(soft-in, soft-out, soft-time)

Example Parallel Concatenated Codes:

Log-likelihood ratios

D: a posteriori decoder output
D = Z + A + E

Z: channel output
A: a priori input
E: extrinsic input
Check Node Decoder

\[ L(X_i) = \sum_{\substack{j=1 \atop j \neq i}}^{n} L(X_j) \]
Variable Node Decoder

\[ L(X_i) = \sum_{j=1}^{n} L(X_j) \quad \text{for } j \neq i \]
(7,4) Hamming Code
Parity Check Matrix

\[
H = \begin{pmatrix}
1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 \\
\end{pmatrix}
\]
Trellis Based Decoder

Special case: M=1 (memory)

\[ L(X_i) = \frac{\Delta V_i}{V_T}, \quad i = 1, 2, 3 \]
Trellis Based Decoder: Structure

S' – 2^M Transistors
R – 2^M2^N Transistors
S – 2^M Transistors

M=3

\( g^{(1)}(D) = 1 + D^2 + D^3 \)

\( g^{(2)}(D) = 1 + D + D^3 \)
Decoder Network

(16,8) tailbiting code, APP Decoder

16 differential input voltages

8 differential output voltages
Simulation Results - Settling Time

- Corrected channel error
Implementation Examples
# Implementation Examples

<table>
<thead>
<tr>
<th>TUM Lucent</th>
<th>ETH Endora</th>
<th>University of Utah</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>(16,8,3), m=1</td>
<td>(18,9,5), m=2</td>
</tr>
<tr>
<td>process</td>
<td>0.25μm BiCMOS</td>
<td>IBM6HP</td>
</tr>
<tr>
<td>chip area</td>
<td>1.68mm²</td>
<td>1.64mm²</td>
</tr>
<tr>
<td>transistor</td>
<td></td>
<td></td>
</tr>
<tr>
<td># bipolar</td>
<td>441</td>
<td>940</td>
</tr>
<tr>
<td># nMOS</td>
<td>356</td>
<td>-</td>
</tr>
<tr>
<td># pMOS</td>
<td>-</td>
<td>650</td>
</tr>
<tr>
<td>total # per inf.bit&amp;state</td>
<td>49.8</td>
<td>54.4</td>
</tr>
<tr>
<td>supply</td>
<td>3.3V</td>
<td>3.3V</td>
</tr>
<tr>
<td>bias per block</td>
<td>80μA (nMOS)</td>
<td>200μA (bipolar)</td>
</tr>
<tr>
<td>power consumption</td>
<td>20mW</td>
<td>150mW</td>
</tr>
<tr>
<td>power per inf.bit&amp;state</td>
<td>1.25mW</td>
<td>9.4mW</td>
</tr>
<tr>
<td>measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>channel</td>
<td>AWGN</td>
<td>AWGN</td>
</tr>
<tr>
<td>speed</td>
<td>160Mbit/s</td>
<td>10 Gbit/s</td>
</tr>
<tr>
<td>BER</td>
<td>yes</td>
<td>tbd.</td>
</tr>
</tbody>
</table>
Simulation Results (BER)

BPSK modulation - memoryless channel with additive white Gaussian noise

- analog decoder - measured data
- analog decoder - simulated data
- MAP decoder - simulated data
- uncoded

BER vs. Eb/No in dB
Conclusions

Open Questions
- Place interleaver cross-connecting network on chip
- Frame synchronization
- „Timing recovery“
- D/A conversion of the input values
- Storage of analog values

Usage
- Optical Transmission Systems
- Magnetic Recording
Thank you!

Any Questions?