Decoder Scheduling of Hybrid Turbo Codes

Neele von Deetzen

International University Bremen
School of Engineering and Science

ITG-Sitzung, München, 22. Mai 2006
Overview

- **System Model**: Hybrid Turbo Codes
- **Decoder Architecture**
  - Parallel Concatenation
  - Serial Concatenation
  - Hybrid Concatenation
- **Problem Statement**
- **Possible EXIT Charts**
- **Global (Multiple) EXIT Charts**
  - Decoder Scheduling
- **Evolution of Local EXIT Charts**
  - Relation between EXIT Charts
- **Conclusions**
**System Model: Hybrid Turbo Codes**

- Combined parallel/serial concatenation with interleavers
- Codes of rate $R_{11} = \frac{k}{k_1}$, $R_{12} = \frac{k_1}{n_1}$, $R_{21} = \frac{k}{k_2}$, and $R_{22} = \frac{k_2}{n_2}$
- Systematic component codes

\[ \rightarrow \mathbf{c} = \begin{pmatrix} c_{1,1}(1) & c_{1,2}(1) & c_{1,3}(1) & c_{2,2}(1) & c_{2,3}(1) & \cdots \\ \cdots & c_{1,1}(2) & c_{1,2}(2) & c_{1,3}(2) & c_{2,2}(2) & c_{2,3}(2) & \cdots \end{pmatrix} \]

- Overall rate: $R = \frac{k}{n_1 + n_2 - k}$
**Decoder Architecture**

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

**Parallel Concatenation**

**Serial Concatenation**
Decoder Architecture

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

Parallel Concatenation

Serial Concatenation
**Decoder Architecture**

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

**Parallel Concatenation**

![Parallel Concatenation Diagram]

**Serial Concatenation**

![Serial Concatenation Diagram]
**Decoder Architecture**

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

**Parallel Concatenation**

**Serial Concatenation**
**Decoder Architecture**

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

**Parallel Concatenation**

![Parallel Concatenation Diagram]

**Serial Concatenation**

![Serial Concatenation Diagram]
**Decoder Architecture**

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

**Parallel Concatenation**

**Serial Concatenation**
Decoder Architecture

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

Parallel Concatenation

Serial Concatenation
**Decoder Architecture**

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

**Parallel Concatenation**

- Encoding: \( u \rightarrow G_1 \rightarrow c_1 \) and \( u \rightarrow G_2 \rightarrow c_2 \)
- Decoding: \( \hat{u}_1, \hat{u}_2 \rightarrow \text{Dec} 1 \rightarrow L(\hat{c}_1), L(\hat{c}_2) \)

**Serial Concatenation**

- Encoding: \( u \rightarrow G_1 \rightarrow x \rightarrow G_2 \rightarrow c \)
- Decoding: \( \hat{x}' \rightarrow \text{Dec} 2 \rightarrow L(\hat{c}) \)
Decoder Architecture

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

Parallel Concatenation

Serial Concatenation
**Decoder Architecture**

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

**Parallel Concatenation**

- **Serial Concatenation**
Decoder Architecture

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

**Parallel Concatenation**

**Serial Concatenation**
**Decoder Architecture**

\[ L_e(\hat{u}) = L(\hat{u}) - L_a(\hat{u}) - L(r|u) \]

**Parallel Concatenation**

**Serial Concatenation**
**Decoder Architecture**

**Hybrid Concatenation**

![Diagram of Decoder Architecture showing Hybrid Concatenation](image)

- **Parallel Concatenation**: Shows the parallel combination of two decoder stages, Dec 1 and Dec 2, with input and output log-likelihood ratios (LLRs).
- **Upper Serial Concatenation**: Illustrates the serial concatenation of Dec 12 with Dec 1 and Dec 11 with Dec 1.
- **Lower Serial Concatenation**: Demonstrates the serial concatenation of Dec 22 with Dec 2 and Dec 21 with Dec 2.

Symbols used:
- $u_1$, $u_2$: Input symbols from the channels.
- $c_1$, $c_2$: Output symbols from the decoders.
- $x_0$, $x_1$, $x_2$: Intermediate variables.
- $G_{11}$, $G_{12}$, $G_{21}$, $G_{22}$: Encoder matrices.
- $\Pi_1$, $\Pi_2$: Interleavers.
- $L_a(u)$, $L_a(c)$, $L_a(x)$: Log-likelihood ratios.

Neele von Deetzen, IUB ITG-Sitzung, München, 22. Mai 2006
**Problem Statement**

- Decoding of upper branch for \( n_{l1} \) (local) iterations
- Passing information of upper branch to lower branch
- Decoding of lower branch with \( n_{l2} \) (local) iterations
- Passing information of lower branch to upper branch
- Repeat this for \( n_g \) (global) iterations

**Question**

How to choose \( n_{l1}, n_{l2}, \) and \( n_g \)?
EXtrinsic Information Transfer - EXIT Chart

- Tool for analysing information transfer in iterative decoding
- Mutual information:
  \[ I(X; L) = 1 - E \left\{ \log_2(1 + e^{-L}) \right\} \]
- Compute \( I_a = I(X; L_a) \), \( I_e = I(X; L_e) \)
- Transfer function \( I_e = T(I_a) \)

Goals
Reduce number of iterations and area between transfer curves
EXtrinsic Information Transfer - EXIT Chart

- Tool for analysing information transfer in iterative decoding
- Mutual information:
  \[ I(X; L) = 1 - E \left\{ \log_2(1 + e^{-L}) \right\} \]
- Compute \( I_a = I(X; L_a) \), \( I_e = I(X; L_e) \)
- Transfer function \( I_e = T(I_a) \)

Goals
Reduce number of iterations and area between transfer curves
EXIT CHARTS OF A HYBRID CONCATENATION

- 3 different EXIT charts:
  - Serial concatenation in 1st branch (local EXIT chart)
  - Serial concatenation in 2nd branch (local EXIT chart)
  - Parallel concatenation of the 2 branches (global EXIT chart)
**Multiple EXIT Chart - Scheduling Optimisation**

Consider global EXIT chart depending on local iterations.

\[ n_{l1} = n_{l2} = 1 \]
**Multiple EXIT Chart - Scheduling Optimisation**

Consider global EXIT chart depending on local iterations.

\[ n_{l1} = n_{l2} = 2 \]
Multiple EXIT Chart - Scheduling Optimisation

Consider global EXIT chart depending on local iterations.

\[ n_{l1} = n_{l2} = 3 \]
Multiple EXIT Chart - Scheduling Optimisation

Consider global EXIT chart depending on local iterations.
Multiple EXIT Chart - Scheduling Optimisation

Consider global EXIT chart depending on local iterations.
**Multiple EXIT Chart - Scheduling Optimisation**

Consider global EXIT chart depending on local iterations.

![EXIT Chart Diagram](image)
**Evolution of Local EXIT Charts**

**Encoder structure**

**Global EXIT chart**

**Local EXIT charts**

- $n_g = 1$
- $n_g = 2$
- $n_g = 3$
- $n_g = 4$
CONCLUSIONS

- Hybrid Turbo codes
- Nested iterative decoder
- Decoder scheduling by means of multiple EXIT chart
- Open Question: How to relate global and local EXIT charts?