다양한 부호율로 평가된 터보 부호의 성능 비교

지도교수: 송 홍엽

컴퓨터 응용 연구실 권 이경
Contents

- Motivation
- Turbo Code Structure
- Punctured Turbo Code
- Experimental Results
- Conclusion
Motivation

- Data communication system provide the ability of changing code rates with communication environment
  - Various rate codes are needed
  - Only rate 1/n turbo codes are known
  - Make turbo codes of various rates to provide general trend of performance
Turbo Code Structure

- **Characteristic**
  - Encoder: Parallel Concatenation Scheme
  - Decoder: Soft Output decoding + Iterative decoding

- **Encoder**
  - Rate 1/3 encoder
  - Two component encoder
  - One Interleaver

![Diagram of Turbo Code Structure]

- Encoder: Parallel Concatenation Scheme
- Decoder: Soft Output decoding + Iterative decoding
- Encoder
  - Rate 1/3 encoder
  - Two component encoder
  - One Interleaver

![Diagram of Turbo Code Structure]
Decoder

- **Viterbi Algorithm**
  - Optimal decoding method to minimize sequence error
  - But cannot produce information about each bit

→ **MAP (Maximum A Posteriori) Algorithm**
  - Produce soft information about each bit
Soft Input and Output Decoding

- $x_k$: systematic information added by noise
- $y_k$: parity information added by noise
- $L_a(d_k)$: a priori information of $d_k$
- $L_e(d_k)$: extrinsic information produced by decoder

$$L(d_k) = L_c x_k + L_a(d_k) + L_e(d_k)$$
Iterative Decoding

Use extrinsic information $L_e(d_k)$ of previous decoder as an a priori information of next decoder.
Concept of Puncturing

- **Definition**
  Systematic deleting of encoded bits using specific rules
  
  ➔ Higher/change the code rate

- **Advantage**
  One encoder/decoder pair can produce code of various rates
Turbo Encoder using Puncturing

Puncturing Period = 8

Interleaver

Puncturing Tables

111111111 $r = 1/3$

111111111 $r = 1/2$

Puncturing Period = 8

111111111 $r = 1/3$

10101010 $r = 1/2$

01010101 $r = 1/2$
Application of Puncturing to Turbo Code

- The effect of **Puncturing Pattern** on punctured turbo code performance

- Various rate turbo code generation using puncturing method
Puncturing Pattern

- The performance of Punctured turbo code vary with puncturing patterns
  ➔ **Provide general rule** to choose good puncturing pattern

- Puncturing Pattern of Turbo Code
  - Number of Puncturing bits from each encoder output
  - Distance of puncturing bit
  - Puncturing systematic bit or not
Various Rates Turbo Codes

- Generate puncturing pattern by proposed rule
  - Make the generated code by this puncturing pattern have the best performance in that code rate

- Suggest the performance of codes from rate 1/3 to rate 2/3 with best puncturing pattern
  - Predict the trend of performance variation with code rates
## Experimental Results

### System description

<table>
<thead>
<tr>
<th>Constraint Length (K)</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interleaver Type</td>
<td>Random Interleaver</td>
</tr>
<tr>
<td>Interleaver Length (N)</td>
<td>1024</td>
</tr>
<tr>
<td>Decoder Iteration Number</td>
<td>3 번</td>
</tr>
<tr>
<td>Modulation</td>
<td>BPSK, Baseband</td>
</tr>
<tr>
<td>Channel</td>
<td>AWGN</td>
</tr>
</tbody>
</table>

- Programmed by C
- Simulated in Linux Environment
Rate 1/2 Turbo Codes with Different Puncturing Pattern

Puncturing Matrix
From Rate 1/3 To Rate 1/2 Turbo Codes

\[ P_3 \] (frase)

\[ P_5 \] (frase)

\[ P_7 \] (frase)

 Eb/No (dB)

\[ r = \frac{8}{22} = \frac{4}{11} \]

\[ P_5 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \]

\[ \text{Eb/No} = 1.70 \]

\[ r = \frac{8}{18} = \frac{4}{9} \]

\[ P_7 = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \]

\[ \text{Eb/No} = 1.85 \]
From Rate 1/2 To Rate 2/3 Turbo Codes

<table>
<thead>
<tr>
<th>부호율</th>
<th>평처링 행렬</th>
</tr>
</thead>
<tbody>
<tr>
<td>r = 1/3</td>
<td></td>
</tr>
<tr>
<td>r = 1/2</td>
<td></td>
</tr>
<tr>
<td>r = 8/15</td>
<td></td>
</tr>
<tr>
<td>r = 8/13</td>
<td></td>
</tr>
<tr>
<td>r = 2/3</td>
<td></td>
</tr>
</tbody>
</table>

\[
P = \begin{bmatrix}
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

\[R = \frac{8}{15} = \frac{4}{7}\]

\[P_{10} = \begin{bmatrix}
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}\]

\[R = \frac{8}{13}\]

\[P_{12} = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}\]

\[R = \frac{8}{12} = \frac{2}{3}\]

\[P_{13} = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}\]
At $P_b = 10^{-4}$, the required $Eb/No$ increase almost linearly with code rate.
Conclusion

- **Results**
  - Suggest puncturing pattern choice guide by experiment
  - Provide the performance of various rate turbo codes

- **Future Research**
  - Suggested as a channel code in IMT-2000
    - Fading channel performance research
    - Performance bound prediction by theoretically
    - Hardware Implementation