An Efficient Technique for Bit Error Probability Estimation of MLSDA-Turbo Decoder

Presenter: Jia-Qin Pan (潘家鏘)
Advisory by Prof. Po-Ning Chen
Dept. of Communications Engineering
National Chiao Tung University

Outline

- Introduction and Background
- Preliminaries
  - Fixed-Length Codes and Maximum Metric Decoding
  - Importance Sampling
  - MLSDA for PCCC
- The Important Sampling Method on MLSDA-Turbo Decoder
- Simulation Results
- Conclusions
Introduction and Background

- MLSDA (Maximum-likelihood soft-decision sequential decoding algorithm) was proposed by Chen and Han in 1999. Later, it was applied to the decoding of PCCC codes. Unlike the iterative decoding algorithm, the algorithm turns out to be a maximum-likelihood decoding algorithm for PCCC.
- Unfortunately, for this ML decoder, brute-force MC often requires a very large number of simulation trials in order to achieve a meaningful estimate of system performance.

Importance Sampling (IS) is a modified MC technique which can efficiently reduce the number of simulation trials required to achieve a specified accuracy.

\[ I_s(x) = \begin{cases} \delta & \text{if } \text{decoded} \\ 0 & \text{otherwise} \end{cases} \]

\[ E_s(F(x)) = \frac{1}{T} \sum_{i=1}^{T} F(x_i) + \sum_{i=1}^{T} \delta_{s}(F(x_i)) \]
Introduction and Background

- In recent years, there has been considerable interest in applying IS to digital communications systems. Some of them are applied to ML decoder, e.g., Sadowsky developed an error event simulation (EES) method for the simulation of the Viterbi decoders; Khaled and Khurram developed a modified error event simulation (MEES) method for the simulation of the ZJ decoders.
- Here, a new application of the important sampling method for estimating BER of MLSDA-Turbo Decoder is presented.

Preliminaries

Fixed-length Codes and Maximum Metric Decoding

- System Model

Monte-Carlo Simulation

Stationary Memoryless Channel

Decoded Codeword

The joint density

The decision regions of a maximum metric decoder

$$f(y|x) = \prod p(y, x)$$

$$D(x) = \{ \hat{y} : \sum m(x, y) = \max \sum m(x, y) \}$$
Preliminaries
Fixed-length Codes and Maximum Metric Decoding

System Model

The indicator function of the decoding region: \( D(x') \)

\[ I_x(y) = \begin{cases} 1 & \text{if } y \in D(x') \\ 0 & \text{if } y \notin D(x') \end{cases} \]

The specific decoding error probability

\[ P(x'|x) = \int_{D(x')} f(y|x) dy = \int_{D(x')} I_x(y) f(y|x) dy \]

The average bit error probability

\[ P_b(x) = \frac{1}{m} E[N_x | x \text{ transmitted}] = \frac{1}{m} \sum n_x(x, x') P(x'|x) \]

\( n_x \): the number of information bits transmitted by a single codeword transmission.

\( n_x(x, x') \): the number of postdecoding information bit errors caused by decoding \( x' \) instead of \( x \).

\( m = \log_2(\text{codeword length}) \)

Preliminaries
Importance Sampling

How to apply importance sampling to coded communications systems?

Importance Sampling Estimator

\[ \hat{S}(x) = \frac{1}{m} \sum I_x(y) f(y|x) \]

Importance Sampling Weight

\[ w(x, x') = \frac{f(x'|x)}{f(x|x')} \]

\[ \hat{S}(x) = \frac{1}{m} \sum w(x', x) I_x(y) \]

\( x \) decoded otherwise

Codeword \( \tilde{x} \)

Channel (*)

Decoded Codeword \( \tilde{x} \)

2001/6/14
Preliminaries
Importance Sampling

How to choose a good simulation channel?

- The expectation operation for the simulation distribution
  \[ E^*(\hat{Y} | \tilde{Z}) = \frac{\int L(y) \pi^*(y | \tilde{z}) f(y | \tilde{z}) \text{d}y}{\int f^*(y | \tilde{z}) \text{d}y} = P(\tilde{Y} | \tilde{Z}) \]

- The variance of the estimator
  \[ \text{var}[\hat{Y} | \tilde{z}] = \frac{1}{L} \left\{ \int \left( \frac{f^*(y | \tilde{z})}{f^*(y | \tilde{z}) - P(\tilde{Y} | \tilde{Z})} \right)^2 \text{d}y \right\} \]

- The optimal importance sampling density
  \[ f^*(\tilde{y} | \tilde{z}) = \frac{f(\tilde{y} | \tilde{z}) L(\tilde{y})}{P(\tilde{y} | \tilde{z})} \]

How to estimate the precision of importance sampling estimates?

- The corresponding estimate of \( \text{var}[\hat{Y} | \tilde{z}] \)
  \[ \text{Var} = \frac{1}{L} \sum_{t=1}^{L} \left( \frac{f^*(y^t | \tilde{z})}{f^*(y^t | \tilde{z}) - P(\tilde{Y} | \tilde{Z})} \right)^2 \]

- The relative precision estimates of \( P(\tilde{Y} | \tilde{Z}) \)
  \[ \text{RPE}_{\text{rel}} = \frac{\text{Var}}{P^2(\tilde{Y} | \tilde{Z})} \]

- The relative precision estimates of \( P(\tilde{Y} | \tilde{Z}) \)
  \[ \text{RPE}_{\text{rel}} = \frac{\text{Var}}{P^2(\tilde{Y} | \tilde{Z})} \cdot \frac{1}{\sqrt{m \sum_{i=1}^{m} a_i(\tilde{z}, \tilde{Y}) P^2(\tilde{Y} | \tilde{Z})}} \]
Preliminaries
MLSDA for PCCC

- PCCC with $G_1=37, G_2=21$ as its component code.

```
1  2  3  4  5  6
7  8  9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29 30
31 32 33 34 35 36
```

Block interleaver with $N=36$

---

2001/6/14

---

Preliminaries
MLSDA for PCCC

- Tree-based **Maximum-likelihood** Soft-decision Sequential Decoding Algorithm on **AWGN** channels

- The conventional sequential decoding algorithm with a new metric

$$
\tilde{d}(j) = \min \{d(j) \mid d(j) \in D_j\}
$$

where $x_j^{(i)}$ and $r_j^{(i)}$ are respectively the $i$th bit of the $j$th transmitted block and $i$th received bit at $j$th received block, and

$$
y_j^{(i)} = \begin{cases} 
1 & \text{if } y_j^{(i)} < 0; \\
0 & \text{otherwise} 
\end{cases}
$$
Preliminaries
MLSDA for PCCC

- ML decoding algorithm
  - Step 1

```
STACK

Find the successors of the top path in the STACK
There could be 1 or 2 or 4 successors.
```

2001/6/14

---

Preliminaries
MLSDA for PCCC

- Step 2

```
STACK

Delete the top path.
Insert the successors into the STACK.
Reordering the STACK according to the metric.
```

2001/6/14
Preliminaries
MLSDA for PCCC

- Step 3
  If the length of STACK exceeds the Threshold, delete the path with Max metric in the stack.

```
<table>
<thead>
<tr>
<th>Min metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>2nd Min metric</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>2nd Max Metric</td>
</tr>
<tr>
<td>Max metric</td>
</tr>
</tbody>
</table>
```

Threshold

The Important Sampling Method on MLSDA-Turbo Decoder

- Simulation Flow

<table>
<thead>
<tr>
<th>Channel Output $\hat{Y}$</th>
<th>MLSE-A-Turbo Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data sequence $(00 \ldots 0 \ldots 0)$</td>
<td>Error sequence $(01 \ldots 1 \ldots 1)$</td>
</tr>
<tr>
<td>Codeword $(1-1 \ldots -1 \ldots -1)$</td>
<td>Turbo Encoder</td>
</tr>
</tbody>
</table>

Simulation Channel List

1. stationary channels
2. nonstationary channels

<table>
<thead>
<tr>
<th>Simulation Channel List</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$D$</th>
<th>$N_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
</tr>
</tbody>
</table>
The Important Sampling Method on MLSDA-Turbo Decoder

- **Stationary Channel**

  - The default channel
    \[ (-1+n_0) (-1+n_0) \cdots (-1+n_0) \]
    \[ n_0 = \text{Gauss}(0, \text{Var}_0) \]
  
  - The biased channel
    \[ (-1+n_1) (-1+n_1) \cdots (-1+n_1) \]
    \[ n_1 = \text{Gauss}(0, \text{Var}_1) \]

  \[ \text{Var}_1 > \text{Var}_0 \]

- **Nonstationary channels**
  - Channel I

  - Transmit \((00 \ldots 0)_m\) through the default channel
    \[ (-1+n_0) (-1+n_0) \cdots (-1+n_0) \]
    \[ n_0 = \text{Gauss}(0, \text{Var}_0) \]
  
  - Transmit \((00 \ldots 0)_m\) through the biased channel for the error event \((00\ldots01)_m\)
    \[ (-1+n_0) (-1+n_0) \cdots (-1+n_0) (-1+n_1) \]
    \[ n_1 = \text{Gauss}(0, \text{Var}_1) \]

  \[ \text{Var}_1 > \text{Var}_0 \]

2001/6/14
The Important Sampling Method on MLSDA-Turbo Decoder

- Channel II

- Channel III
Simulation Results

- Estimate the bit error probability of **SNR=8.0dB** by Biased Stationary Channels

![Graph showing simulation results for different SNR values with Monte Carlo simulation results.](image)

Simulation Results

- Time-spent during estimating the bit error probability of **SNR=8.0dB** by Biased Stationary Channels

![Graph showing time-spent during Monte Carlo simulation for different SNR values.](image)
Simulation Results

- Estimate the bit error probability of $\text{SNR}=3.0\text{dB}$ by Biased Stationary Channels

Simulation Results

- Time-spent during estimating the bit error probability of $\text{SNR}=3.0\text{dB}$ by Biased Stationary Channels
**Simulation Results**

- Estimate the bit error probability of **SNR=8.0dB** by the **first Biased Nonstationary Channels**

![Graph showing bit error probability](image)

- Time-spent during estimating the bit error probability of **SNR=8.0dB** by the **first Nonstationary Channels**

<table>
<thead>
<tr>
<th>P_0</th>
<th>MC</th>
<th>SNR=6.0</th>
<th>SNR=3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3e-06</td>
<td>6.2e-06</td>
<td>5.2e-06</td>
<td></td>
</tr>
<tr>
<td>Rpe %</td>
<td>5.80</td>
<td>5.75</td>
<td>5.84</td>
</tr>
<tr>
<td>Com</td>
<td>161673235</td>
<td>12534768</td>
<td>47607830</td>
</tr>
<tr>
<td>Ratio</td>
<td>0.7753</td>
<td>0.2945</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{Ratio} = \frac{\text{Com}_{\text{SNR}=6.0}}{\text{Com}_{\text{SNR}=3.0}}
\]
Simulation Results

- Estimate the bit error probability of **SNR=3.0dB** by the first Biased Nonstationary Channels

![Graph showing estimated SNR=3.0dB with different simulation times](image)

Simulation Results

- Time-spent during estimating the bit error probability of **SNR=3.0dB** by the first Nonstationary Channels

<table>
<thead>
<tr>
<th></th>
<th>MC</th>
<th>SNR=3.0</th>
<th>SNR=1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_i$</td>
<td>1.2e-03</td>
<td>1.1e-03</td>
<td>1.0e-03</td>
</tr>
<tr>
<td>Rpe %</td>
<td>7.74</td>
<td>17.4</td>
<td>17.2</td>
</tr>
<tr>
<td>Com</td>
<td>155713373</td>
<td>182182136</td>
<td>113213118</td>
</tr>
<tr>
<td>Ratio</td>
<td>1</td>
<td>1.17</td>
<td>0.7271</td>
</tr>
</tbody>
</table>

\[
\text{Ratio} = \frac{\text{Com}}{t_i}
\]
Simulation Results

- Estimate the bit error probability of **SNR=8.0dB** by the second Biased Nonstationary Channels

Simulation Results

- Time-spent during estimating the bit error probability of **SNR=8.0dB** by the second Nonstationary Channels

Simulation Results
Simulation Results

Estimate the bit error probability of **SNR=3.0dB** by the second Biased Nonstationary Channels

![Simulation Results Diagram]

Simulation Results

Time-spent during estimating the bit error probability of **SNR=3.0dB** by the second Nonstationary Channels

<table>
<thead>
<tr>
<th></th>
<th>MC</th>
<th>SNR=3.0</th>
<th>SNR=1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R</strong></td>
<td>1.2e-03</td>
<td>1.0e-03</td>
<td>9.8e-04</td>
</tr>
<tr>
<td>Rpe %</td>
<td>7.74</td>
<td>7.79</td>
<td>7.75</td>
</tr>
<tr>
<td>Com</td>
<td>155713373</td>
<td>76220279</td>
<td>139841615</td>
</tr>
<tr>
<td>Ratio</td>
<td>1</td>
<td>0.4895</td>
<td>0.8981</td>
</tr>
</tbody>
</table>

\[ \text{Ratio} = \frac{\text{Com}}{\text{MC}} \]
Simulation Results

- Estimate the bit error probability of **SNR=8.0dB** by the third Biased Nonstationary Channels

![Graph showing simulation results for different SNR values.]

**Simulation Results**

- Time-spent during estimating the bit error probability of **SNR=8.0dB** by the third Nonstationary Channels

<table>
<thead>
<tr>
<th></th>
<th>MC</th>
<th>SNR=6.0</th>
<th>SNR=3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_e$</td>
<td>5.3e-06</td>
<td>5.3e-06</td>
<td>5.4e-06</td>
</tr>
<tr>
<td>Rpe %</td>
<td>5.80</td>
<td>5.83</td>
<td>5.85</td>
</tr>
<tr>
<td>Com</td>
<td>161673235</td>
<td>76910196</td>
<td>36824995</td>
</tr>
<tr>
<td>Ratio</td>
<td>1</td>
<td>0.4757</td>
<td>0.2278</td>
</tr>
</tbody>
</table>

**Ratio** = \( \frac{\text{Com}}{\text{Ratio}} \)
Simulation Results

- Estimate the bit error probability of $\text{SNR}=3.0\text{dB}$ by the third Biased Nonstationary Channels

Simulation Results

- Time-spent during estimating the bit error probability of $\text{SNR}=3.0\text{dB}$ by the third Nonstationary Channels

<table>
<thead>
<tr>
<th></th>
<th>MC</th>
<th>SNR=3.0</th>
<th>SNR=1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_e$</td>
<td>1.2e-03</td>
<td>9.5e-04</td>
<td>9.8e-04</td>
</tr>
<tr>
<td>Com</td>
<td>155713373</td>
<td>171978318</td>
<td>148068555</td>
</tr>
<tr>
<td>Ratio</td>
<td>1</td>
<td>1.1045</td>
<td>0.9509</td>
</tr>
</tbody>
</table>

\[
\text{Ratio} = \frac{\text{Com}}{\text{R}_e}
\]
Simulation Results

- Estimate the bit error probability by the Best Biased Nonstationary Channel in order to achieve the same estimates of system performance

### Monte Carlo Simulation for recording at 300 bit errors

<table>
<thead>
<tr>
<th>SNR(dB)</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>2.5</th>
<th>3.0</th>
<th>3.5</th>
<th>4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pb</td>
<td>6.5e+00</td>
<td>6.5e+00</td>
<td>6.5e+00</td>
<td>6.5e+00</td>
<td>6.5e+00</td>
<td>6.5e+00</td>
<td>6.5e+00</td>
<td>6.5e+00</td>
<td>6.5e+00</td>
</tr>
<tr>
<td>Rpe</td>
<td>17.3</td>
<td>17.3</td>
<td>16.7</td>
<td>15.9</td>
<td>9.83</td>
<td>13.9</td>
<td>7.94</td>
<td>6.87</td>
<td>6.26</td>
</tr>
<tr>
<td>Com</td>
<td>22227635</td>
<td>19270343</td>
<td>20655576</td>
<td>21589806</td>
<td>20811909</td>
<td>18958455</td>
<td>15571357</td>
<td>94004594</td>
<td>65613107</td>
</tr>
</tbody>
</table>

### Importance Sampling Simulation for recording at the same RPE

<table>
<thead>
<tr>
<th>SNR(dB)</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>2.5</th>
<th>3.0</th>
<th>3.5</th>
<th>4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pb</td>
<td>1.3e+02</td>
<td>9.3e+01</td>
<td>4.2e+01</td>
<td>3.5e+01</td>
<td>2.8e+01</td>
<td>1.3e+01</td>
<td>1.0e+01</td>
<td>6.3e+00</td>
<td>4.3e+00</td>
</tr>
<tr>
<td>Rpe</td>
<td>17.3</td>
<td>17.3</td>
<td>16.7</td>
<td>15.9</td>
<td>9.83</td>
<td>13.9</td>
<td>7.94</td>
<td>6.87</td>
<td>6.26</td>
</tr>
<tr>
<td>Com</td>
<td>206142239</td>
<td>190742239</td>
<td>189590128</td>
<td>163451359</td>
<td>18958455</td>
<td>18958455</td>
<td>18958455</td>
<td>18958455</td>
<td>18958455</td>
</tr>
<tr>
<td>Gain</td>
<td>0.93011</td>
<td>0.98879</td>
<td>0.98879</td>
<td>0.98879</td>
<td>0.98879</td>
<td>0.98879</td>
<td>0.98879</td>
<td>0.98879</td>
<td>0.98879</td>
</tr>
</tbody>
</table>

2001/6/14 37

---

Simulation Results

- Estimate the bit error probability by the Best Biased Nonstationary Channel in order to achieve the same estimates of system performance

### Monte Carlo Simulation for recording at 300 bit errors

<table>
<thead>
<tr>
<th>SNR(dB)</th>
<th>4.5</th>
<th>5.0</th>
<th>5.5</th>
<th>6.0</th>
<th>6.5</th>
<th>7.0</th>
<th>7.5</th>
<th>8.0</th>
<th>8.5</th>
<th>9.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rpe</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td></td>
</tr>
<tr>
<td>Com</td>
<td>359870190</td>
<td>339017661</td>
<td>339022692</td>
<td>397153559</td>
<td>40464037</td>
<td>40464037</td>
<td>40464037</td>
<td>40464037</td>
<td>40464037</td>
<td></td>
</tr>
</tbody>
</table>

### Importance Sampling Simulation for recording at the same RPE

<table>
<thead>
<tr>
<th>SNR(dB)</th>
<th>4.5</th>
<th>5.0</th>
<th>5.5</th>
<th>6.0</th>
<th>6.5</th>
<th>7.0</th>
<th>7.5</th>
<th>8.0</th>
<th>8.5</th>
<th>9.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rpe</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td>1904850</td>
<td></td>
</tr>
<tr>
<td>Com</td>
<td>359870190</td>
<td>339017661</td>
<td>339022692</td>
<td>397153559</td>
<td>40464037</td>
<td>40464037</td>
<td>40464037</td>
<td>40464037</td>
<td>40464037</td>
<td></td>
</tr>
</tbody>
</table>

2001/6/14 38
Simulation Results

- Monte-Carlo VS Importance-Sampling

Monte Carlo Simulation Results:

- Estimate the bit error probability by the Best Biased Nonstationary Channel in order to achieve the same estimates of system performance.

![Graph showing bit error probability vs. SNR (dB)]

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>8.5</th>
<th>10.0</th>
<th>10.5</th>
<th>11.0</th>
<th>11.5</th>
<th>12.0</th>
<th>12.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pb</td>
<td>3.6e-07</td>
<td>1.1e-07</td>
<td>3.3e-08</td>
<td>7.9e-09</td>
<td>1.6e-09</td>
<td>3.6e-10</td>
<td>3.8e-11</td>
</tr>
<tr>
<td>Rpe</td>
<td>6.10</td>
<td>6.04</td>
<td>6.00</td>
<td>6.03</td>
<td>6.04</td>
<td>6.07</td>
<td>6.06</td>
</tr>
<tr>
<td>Com</td>
<td>1399725</td>
<td>1389150</td>
<td>1388576</td>
<td>1377392</td>
<td>1384587</td>
<td>1390125</td>
<td>1402518</td>
</tr>
</tbody>
</table>

Importance Sampling Simulation for recording at RPE=6%:

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>13.0</th>
<th>13.5</th>
<th>14.0</th>
<th>14.5</th>
<th>15.0</th>
<th>15.5</th>
<th>16.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pb</td>
<td>3.9e-12</td>
<td>3.3e-13</td>
<td>1.9e-14</td>
<td>8.5e-16</td>
<td>2.8e-17</td>
<td>5.3e-19</td>
<td>7.3e-21</td>
</tr>
<tr>
<td>Rpe</td>
<td>6.05</td>
<td>6.09</td>
<td>6.10</td>
<td>6.08</td>
<td>6.09</td>
<td>6.08</td>
<td>6.01</td>
</tr>
<tr>
<td>Com</td>
<td>1841599</td>
<td>1563825</td>
<td>1483578</td>
<td>1783187</td>
<td>1753832</td>
<td>1809472</td>
<td>1942098</td>
</tr>
</tbody>
</table>

2001/6/14
Simulation Results

- Monte-Carlo VS Importance-Sampling

![Graph showing comparison between Monte Carlo and Importance Sampling]

Conclusions

- There is no computational gain when we use the Importance Sampling Simulation by Biasd Stationary Channels. In fact, the noisier channel we use the more metric computations we pay, and it brings out the larger relative precision because of its larger variation than MC Simulation.
Conclusions

- The optimal distribution of the biased channel

\[ f_r(x, y) = \frac{f(x, y)P_r(x)}{P(x|y)} \]

can show why the second nonstationary channel causes the best performance among all channels.

Conclusions

- In this Simulation, the Importance Sampling can work well in high SNR, but less effectiveness in low SNR when we use the second biased channel. In fact, we gained 10-500 time-saving when we estimate bit error probability between SNR=6dB to 9dB, and gained 0-10 time-saving for SNR=0dB to 5.5dB.