Performance Comparison and Analysis of Serial Concatenated Convolutional Codes

Dongwon Lee, and Eon Kyeong Joo, Member, IEEE

Abstract—In this paper, the performance of three types of serial concatenated convolutional codes (SCCC) is compared and analyzed in additive white Gaussian noise (AWGN) channel. The BER performance of SCCC is varied according to constituent codes, interleaver type and size, and concatenation method [6], [7]. The variation due to constituent codes and interleavers has been researched in other papers, but it is not yet for the concatenation method. Therefore, in this paper, the BER performance of three types of SCCC with different concatenation scheme is compared and analyzed in additive white Gaussian noise (AWGN) channel. The distance spectra of the three types are used to analyze the performance.

Performance Comparison and Analysis of Serial Concatenated Convolutional Codes

Dongwon Lee, and Eon Kyeong Joo, Member, IEEE

Abstract—In this paper, the performance of three types of serial concatenated convolutional codes (SCCC) is compared and analyzed in additive white Gaussian noise (AWGN) channel. The BER performance of SCCC is varied according to constituent codes, interleaver type and size, and concatenation method [6], [7]. The variation due to constituent codes and interleavers has been researched in other papers, but it is not yet for the concatenation method. Therefore, in this paper, the BER performance of three types of SCCC with different concatenation scheme is compared and analyzed in additive white Gaussian noise (AWGN) channel. The distance spectra of the three types are used to analyze the performance.

This paper is organized as follows. Section II describes the encoder and the decoder of the three types of SCCC. In Section III, the MAP algorithm applied to SCCC is explained. Some simulation results are shown and analyzed by using the distance spectrum in Section IV. Finally, Section V concludes the paper.

II. CONCATENATION METHODS OF SCCC

In this paper, the model three types of SCCC are presented. The recursive systematic convolutional (RSC) code is used as inner and outer encoder and the overall code rate is 1/3. In order to improve the performance of SCCC, a bit interleaver between component encoders is used. The decoder of the presented SCCC is composed of two MAP modules, interleaver, and deinterleaver.

The first type (Type I) of SCCC is shown in Fig. 1, where I and DI denote interleaver and deinterleaver, respectively. And D is 1 bit shift register. As shown in Fig. 1, only the parity bits of the outer encoder are passed to the inner encoder through the interleaver. Therefore the interleaver size is the same as the length of an input frame. The extrinsic values of the information bits of the inner encoder are calculated at MAP1 and then passed to MAP2 through the deinterleaver. And the LLR values of the information bits and the extrinsic values of the parity bits of the outer encoder are calculated at MAP2. The former is used for detection and the latter is fed into MAP1 for iterative decoding.
The second type (Type II) of SCCC is shown in Fig. 2, where MUX and DEMUX is each multiplexer and demultiplexer.

As depicted in Fig. 2, both the information bits and the parity bits of the outer encoder are transferred to the inner encoder. Therefore, the multiplexer is necessary between the outer encoder and the interleaver and the interleaver size is two times the length of an input frame. In order to set the overall code rate to 1/3, some bits are eliminated by puncturing matrix $P$, at the last output stage of the encoder. The matrix $P$ is defined by

$$P = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$  

The decoder of Type II is similar to that of Type I. However, the multiplexer and the demultiplexer are required at the decoder. And the extrinsic values of the information bits as well as the parity bits of the outer encoder are also fed into MAP1.

The third type (Type III) of SCCC is shown in Fig. 3. As shown in Fig. 3, the structure of Type III is similar to that of Type II.

However, the 2/3-RSC code is used as inner encoder and the demultiplexer is used between the interleaver and the inner encoder in order to set the overall code rate to 1/3. The interleaver size is two times the length of an input frame like Type II. The decoder of Type III is similar to that of Type II excepting that the additional multiplexer and demultiplexer are required.

### III. MAP Algorithm to SCCC

In this paper, the MAP algorithm is used in the iterative decoder of SCCC. It is almost same with the MAP applied to PCCC. However, the LLR values of parity bits as well as information bits might be calculated in the outer decoder (MAP2) of SCCC.

For example, the trellis diagram of the 4-state RSC code with generator $[7, 5]$ is depicted in Fig. 4, where $m$ represents the state number and $S_k$ is the state at time $k$. The dotted line represents the state transition when the input bit is 0, and the solid line represents the state transition when the input bit is 1. At time $k$, the LLR for the information bit $u_k$, is defined as

$$A(u_k) = \log \frac{P(u_k = 1 | r^T)}{P(u_k = 0 | r^T)}.$$  

Fig. 4 Trellis diagram of the RSC code
where $r_k^l$ represents the set from the received symbol $r_l$ to $r_k$.

Using the forward and backward parameters [4], [5], (1) can be rewritten as follows:

$$A(u_k) = \log \sum_m a_k^1(m) \cdot \beta_k(m),$$

(2)

where

$$a_k^1(m) = P(u_k = i, S_k = m, r_k^l),$$

$$\beta_k(m) = P(r_{k+1}^l | S_k = m).$$

(3)

The forward and backward recursion are performed by

$$a_k^1(m) = \sum_{n=0}^{l} \gamma(n, n, n, m) \cdot a_{k-1}^1(n),$$

$$\beta_k(m) = \sum_{n=0}^{l} \gamma(n, n+1, n, m) \cdot \beta_{k+1}(n).$$

(4)

For example, if the information bit is equal to 0 at time $k$, then $a_k^0(m)$ is computed as

$$a_k^0(0) = \sum_{n=0}^{l} \gamma(n, 0, 0, 0) \cdot a_{k-1}^0(0),$$

$$a_k^0(1) = \sum_{n=0}^{l} \gamma(n, 1, 1) \cdot a_{k-1}^0(1),$$

$$a_k^0(2) = \sum_{n=0}^{l} \gamma(n, 1, 2) \cdot a_{k-1}^0(2),$$

$$a_k^0(3) = \sum_{n=0}^{l} \gamma(n, 1, 3) \cdot a_{k-1}^0(3).$$

(5)

At time $k$, the LLR for the parity bit $p_k$, can be obtained by substituting $p_k$ for $u_k$ in (1), (2), and (3). The forward and backward recursion are performed the same as in (4). However, the previous state number contributing to the forward parameter $a_k^1(m)$ and the post-state number contributing to the backward parameter $\beta_k(m)$ for the parity bits is each different to those for the information bits. For example, if the parity bit is equal to 0 at time $k$, then $a_k^0(m)$ can be calculated as

$$a_k^0(0) = \sum_{n=0}^{l} \gamma(n, 0, 0, 0) \cdot a_{k-1}^0(0),$$

$$a_k^0(1) = \sum_{n=0}^{l} \gamma(n, 1, 1) \cdot a_{k-1}^0(1),$$

$$a_k^0(2) = \sum_{n=0}^{l} \gamma(n, 1, 2) \cdot a_{k-1}^0(2),$$

$$a_k^0(3) = \sum_{n=0}^{l} \gamma(n, 1, 3) \cdot a_{k-1}^0(3).$$

(6)

IV. SIMULATION RESULTS

To evaluate the BER performance of the three types of SCCC in AWGN channel, some simulations have been performed. In these simulations, the overall code rate is 1/3 and the interleaver size is 1000 bits.

The performance of Type I is shown in Fig. 5. The BER performance of Type I is not improved above the ratio of bit energy to noise power spectral density, $E_b/N_0$, of 2.5dB over the five iterations. The performance of Type II and Type III are described in Fig. 6 and 7. As shown in Fig. 6 and 7, the BER performance of Type II is still improved over the five iterations at high $E_b/N_0$. The performance of Type III is similar to that of Type II. And Type III shows the better performance at high $E_b/N_0$ than Type II.
When the number of iterations is 7, the required $E_b/N_0$ to obtain the specific BER is shown in Table I. At BER $= 10^{-3}$, Type I has the coding gain of each 0.65 dB and 0.70 dB compared to Type II and Type III. The coding gain decreases to 0.10 dB and 0.05 dB at BER $= 10^{-5}$. Finally, the performance reversed; Type III has the coding gain of each 0.35 dB and 0.10 dB compared to Type I and Type II at BER $= 10^{-6}$.

The simulation results are analyzed using the distance spectrum which is used to calculate the specific upper bound on BER. The general upper bound on BER of concatenated codes in AWGN channel is [2], [8]

$$ P_b \leq \sum_{d=d_{free}}^{N} \left( \sum_{w=1}^{N} A_{w,d} \right) Q \left( \sqrt{\frac{2RE_b}{N_0}} \right), \quad (7) $$

where $N$ represents the interleaver size, $R$ is the overall code rate, and $d_{free}$ is the free distance of output codes. And $A_{w,d}$ is the number of output codes with hamming weight $d$ generated by input codes with hamming weight $w$. And

$$ Q(x) = \left[ 1 - \frac{1}{\pi} x + \frac{1}{\pi} \sqrt{x^2 + 2\pi} \right]^{-1} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \quad (8) $$

The distance spectra of the three types with $N = 500$ are shown in Table II, where $W_d$ is equal to $\sum_{w=1}^{N} A_{w,d}$ in (7).

<table>
<thead>
<tr>
<th>$W_d$</th>
<th>Type I (dB)</th>
<th>Type II (dB)</th>
<th>Type III (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0.90</td>
<td>1.55</td>
<td>1.60</td>
</tr>
<tr>
<td>10</td>
<td>1.40</td>
<td>1.80</td>
<td>1.80</td>
</tr>
<tr>
<td>12</td>
<td>2.00</td>
<td>2.10</td>
<td>2.05</td>
</tr>
<tr>
<td>13</td>
<td>2.55</td>
<td>2.30</td>
<td>2.20</td>
</tr>
</tbody>
</table>

Note that each $d_{free}$ of the three types is 9, 18, and 19. The upper bound on BER of the three types is obtained by substituting the values of Table II in (7) and (8). Fig. 8 shows the upper bound on BER and the practical BER performance of the three types with $R = 1/3$, $N = 500$, and the number of iterations $= 7$. Since $d_{free}$ has the greatest influence on the upper bound, the only $d_{free}$ term in (7) is depicted in Fig. 8. It is shown that the BER performance is closer to the upper bound as $E_b/N_0$ increases. Especially, the performance of Type I relatively approaches to the upper bound in the low $E_b/N_0$ range. Therefore, in the $E_b/N_0$ range from 0 to 2.5 dB, Type I has the best BER performance. On the other hand, above 3.0 dB, the performance is good in order of Type III, II, and I. The order is equal to that of large $d_{free}$.

### V. CONCLUSION

In this paper, the BER performance of the three types of SCCC was compared and analyzed in AWGN channel. The simulation results show that the performance of Type I cannot be improved in spite of the increase in the number of iterations at high $E_b/N_0$. However, the performance of Type II and Type III are still improved over the five iterations at high $E_b/N_0$ in the example. And the performance of Type III is better than Type II.
at high $E_b/N_0$. In sum, Type I shows the best error correction performance at low $E_b/N_0$ and Type III shows the best performance at high $E_b/N_0$. The distance spectrum was used to analyze the BER performance because the spectrum is strongly related to the BER upper bound. Especially, $d_{free}$ has the greatest influence on the upper bound. In the example, $d_{free, Type III} = 19$, $d_{free, Type II} = 18$, and $d_{free, Type I} = 9$, so the performance is good in the order at high $E_b/N_0$.

REFERENCES