Performance Analysis of Serial Concatenation Convolutional Codes Based on Amplitude Matrix

Eng. Balasem Salim Sumait
Electrical Engineering Department
College of Engineering, University of Anbar
E-mail: eng_blasm@yahoo.com

Dr. Salih Mohammed Salih
Electrical Engineering Department
College of Engineering, University of Anbar
E-mail: dr_salih_moh@hotmail.com

Abstract:
Serial Concatenation Convolutional Codes ($SCCC_s$) encoder is built using a serial concatenation of two Recursive Systematic Convolutional (RSC) encoders, separated by an interleaver. These two RSC encoders depending on the trellis termination criteria are software implemented and the performance of each one of them is analyzed under different conditions and circumstances. The output data from the encoder are multiplied by an amplitude matrix (AM) at the transmitter side and the Inverse of Amplitude Matrix (IAM) at the receiver side. The reliability estimation, log-likelihood algebra, and soft channel outputs for Soft Output Viterbi Algorithm (SOVA) are examined. Then the modified Viterbi metric that incorporates a-priori information used for SOVA decoding is derived. A low memory implementation of the SOVA decoder is presented. The iterative SOVA for $SCCC_s$ is described with illustrative examples. The behavior of the $SCCC_s$ encoder-decoder scheme is tested under different circumstances with AM and without AM at the AWGN and Rayleigh fading channels with unlike frame sizes (FS) and constraint length (K). The results show that the performance of system with AM outperforms the other conventional system that worked without AM.

Keywords:
Concatenation Codes, Frame size, RSC, SOVA, SCCCs, AM

1. INTRODUCTION

Recently, a near channel capacity error correcting code called Serial Concatenation Convolutional Codes (SCCCs) was introduced. This error correcting code is able to transmit information across the channel with arbitrary low bit error rate [1]. This code is a serial concatenation of two component convolutional codes separated by an interleaver. Random coding of long block lengths may also perform close to channel capacity, but this code is very hard to decode due to the lack of code structure. Without a doubt, the performance of $SCCC_s$ is partly due to an interleaver used to give the $SCCC_s$ a “random” appearance. However, one big advantage of a $SCCC_s$ is that there is enough code structure (from the convolutional codes) to decode it efficiently.

There are two primary decoding strategies for turbo codes. They are based on a Maximum A Posteriori (MAP) algorithm and a Soft Output Viterbi Algorithm (SOVA). Regardless of which algorithm is implemented, the turbo code decoder requires the use of two (same algorithm) component decoders that operate in an iterative manner. In this paper, the SOVA will be examined, because it is much less complex than MAP and it provides comparable performance results. Furthermore, SOVA is an extension of the Viterbi algorithm, and thus has an implementation advantage over MAP.

2. RSC ENCODER

The Recursive Systematic Convolutional (RSC) encoder is obtained from the nonrecursive nonsystematic (conventional) convolutional encoder by feeding back one of it is encoded outputs to it is input. **Fig. 1** represents a convolutional encoder [2, 3, 4].

The convolutional encoder is represented by the generator sequences $g^{(1)}=$ [1 1 1] and $g^{(2)}=$ [1 0 1] (where g represents the hardware connection between the shift register and modulo-2 adders) and can be equivalently represented in a more compact form as $G=[g^{(1)},g^{(2)}]$. The RSC encoder from this encoder is represented as $G=[1,g^{(2)}/g^{(1)}]$ where the first output (represented by $g^{(1)}$) is fed back to the input. In the above representation, 1 denotes the systematic output, $g^{(2)}$ denotes the feed forword output and $g^{(1)}$ is the feed back to the input of the RSC encoder. **Fig.2** shows the resulting RSC encoder.

It was suggested in [2] that good codes can be obtained by setting the feed back of the RSC encoder to a primitive polynomial, because the primitive polynomial generates maximum-length sequences which adds randomness to the concatenation codes. RSC encoding proceeds by first computation the feedback variable

$$r_i = x_i + \sum_{j=1}^{m} r_{i-j} g_j^{(1)} \tag{1}$$

and then finding the parity output

$$c^2 = \sum_{j=0}^{m} r_{i-j} g_j^{(2)} \tag{2}$$

In order to terminate the trellis of RSC code, the message input x_i must be chosen such that $r_i=0$ for $L-m \le i \le L-1$. Thus from eq.(1), the last m bits of the input message must satisfy

$$x_i = \sum_{j=1}^{m} r_{i-j} g_j^{(1)} \tag{3}$$

3. CONCATENATED CONVOLUTIONAL CODES

A concatenated code is composed of two separate codes that are combined to form a large code [1]. There are many types of concatenation, some of them parallel concatenation convolutional codes ($PCCC_s$) and serial concatenation convolutional codes ($SCCC_s$).

PCCCs, introduced by Berrou *et al* [2] in 1993, were a major breakthrough towards realizing Shannon's channel capacity limit. The performance of these codes achieves low bit error rates (10^{-5} – 10^{-6}) at very low signal to noise ratio, about 0.5 dB away from the theoretical capacity limit. PCCCs perform very well at low SNR, they give rise to a fairly high bit error rate (BER) floor at high Signal-to-Noise Ratio (SNR). In order to alleviate this problem, (SCCCs) were proposed by Benedetto, *et al* [5] using the concepts introduced by Forney [6].

4. ENCODING OF SCCCs

$PCCC_S$ generate parity bits by having several encoders operate in parallel on the same information sequence; one in it is original order, and one or more on interleaved versions of it. We can imagine, then, a serial structure where the information sequence is encoded by a convolutional encoder, interleaved, and then encoded again. Codes of such a structure, is known as $SCCC_S$. It has been observed from computer simulation that $PCCC_S$ outperform

$SCCC_S$ at low SNR_s, but that as the SNR increases; there is a cross-over point after which $SCCC_S$ perform better than $PCCC_S$ [7].
A general diagram for the serial concatenation encoder is given in **Fig. 3**. The $SCCC_S$ encoder is composed of two RSC encoders, which are usually identical. The first RSC encoder (outer RSC encoder) receives the data and then encoded it. But, the second RSC encoder (inner RSC encoder) receives the data after being permuted by an interleaver. As a $PCCC_S$, the interleaving that makes serially concatenated codes appears random. The output coded data from the inner RSC encoder divided into sub-blocks of length 32-bit and then it multiplied by an amplitude matrix of size (N*N), where N=32-bit. **Fig. 3** refers to the case of two convolutional codes, the outer code C_o with rate $R_c^o = q/p$, and the inner code C_i with rate $R_c^i = p/m$, joined by an interleaver of length N bits, generating an SCCC with rate $R_c = k/n$. Note that N must be an integer multiple of p. the input block size is $k = Nq/p$ and the output block size of SCCC is $n = Nm/p$.

5. INTERLEAVER DESIGN

For concatenated codes, an interleaver is used between the two component encoders. The interleaver is used to provide randomness to the input sequences [8, 9]. Also, it is used to increase the weights of the codewords. The interleaver affects the performance of turbo codes because it directly affects the distance properties of the code [10]. By avoiding low-weight codewords, the BER of a $SCCC_s$ can improved significantly. Thus, much research has been done on interleaver design. The following subsections show representative interleavers commonly used in $SCCC_s$ design.

5.1 Block Interleaver ("Row-Column" Interleaver)

Block interleaving or permutation interleaving is the simplest type of interleaver. The permutation of inputs to outputs is contained within one period in a block interleaver. The block interleaver is the most commonly used interleaver in communication systems. It is writes in row wise from left to right and top to bottom and reads out column wise from top to bottom and left to right. **Fig. 4** shows a block interleaver. From **Fig. 4**, the interleaver writes in [0 1 … 1 0 0 … 1 … 1 … 0 0 0 …1 1] and reads out [0 0 … 1 0 1 … 0 … 1 … 1 0 1 … 0 1].

5.2 Random "Pseudo-Random" Interleaver

These pseudo-random interleavers are defined by a pseudo-random number generator or a look-up table where all integers 1 to L (the block size to be interleaved) can be generated. This approach can be lead to good or bad interleavers, especially for small interleaver sizes, computer simulations can be used to find those permutations with good weight-distributions (e.g., [5, 10, 11, 12]). The only criterion for choosing between them is based on computer simulations; there seam to be no analytical criteria.
The random interleaver uses fixed random permutations and map the input sequence according to the permutation order. The length of the input sequence is assumed to be L.

5.3 Semirandom (S-type) Interleaver

The semirandom interleaver is a compromise between a random interleaver and "design" interleaver such as the block interleaver [13, 14]. It was shown [15] that weight-2 data sequences are an important factor in the design of the component codes. The weight of a data sequence, which is made of 0's and 1's, is the number of 1's in that sequence. If we randomly select an interleaver of size L, the probability that a particular weight-2 data sequence will be

permuted by the interleaver into another sequence of the same form is roughly *2/L* for large *L*. This probability is large for small *L*. Therefore, in order to avoid these identical permutations, the permutation algorithm for the semirandom interleaver is described below [14].
Step 1. Select a random index $i \in [0, L-1]$.
Step 2. Select a positive integer $S < \sqrt{\frac{L}{2}}$.
Step 3. Compare *i* to previous *S* integers. For each of the *S* integers, compare *i* to see if it lies within $\pm S$. If *i* does lie within the range, then go back to Step 1. Otherwise, keep *i*.
Step 4. Go back to Step 1 until all *L* positions have been filled.

6. DECODING Of SCCCs

The SCCC decoder is based on a modified Viterbi algorithm that incorporates reliability values to improve decoding performance. The modified Viterbi algorithm for SCCC decoding is the Soft Output Viterbi Algorithm (SOVA) is described. Finally, the decoding algorithm and implementation structure for a SCCC is presented.

7. THEORY OF SOVA FOR SCCCs

The SOVS for $SCCC_s$ is implemented with a modified Viterbi metric. A close examination of log-likelihood algebra and soft channel outputs is required before attempting to derive this modified Viterbi metric. **Fig. 5** shows the system model that is used to describe the above concepts.

7.1 Log-Likelihood Algebra

The log-likelihood algebra used for SOVA decoding of $SCCC_s$ is based on a binary random variable *u* in GF(2) with elements {+1,-1}, where +1 is the logic 0 element ("null" elements) and –1 is the logic 1 elements under ⊕ (modulo 2) addition [14, 16].
The log-likelihood ratio *L*(*u*) for a binary random variable *u* is defined to be:

$$L(u) = \ln \frac{p(u=+1)}{p(u=-1)} \qquad (4)$$

L(*u*) is often denoted as the "soft" value or L-value of the binary random variable u. The sign of *L*(*u*) is the hard dicision of u and the magnitude of *L*(*u*) is the reliability of this decision. The prbability of the random variable u may be conditioned on another random variable *z*. This forms the conditioned log-likelihood ratio *L*(*u*|*z*) and is defined to be:

$$L(u \mid z) = \ln \frac{p(u=+1 \mid z)}{p(u=-1 \mid z)} \qquad (5)$$

The probability of the sum of two binary random variables, say $p(u_1 \oplus u_2 = +1)$ is formed from

$$p(u_1 \oplus u_2 = +1) = p(u_1 = +1)\, p(u_2 = +1) + p(u_1 = -1)\, p(u_2 = -1) \qquad (6)$$

with the following relation

$$p(u = -1) = 1 - p(u = +1) \qquad (7)$$

the probability $p(u_1 \oplus u_2 = +1)$ becomes

$$p(u_1 \oplus u_2 = +1) = p(u_1 = +1)\, p(u_2 = +1) + (1 - p(u_1 = +1))(1 - p(u_2 = +1)) \quad (8)$$

From eq. (4)

$$p(u = +1) = \frac{e^{L(u)}}{1 + e^{L(u)}} \quad (9)$$

it can be shown that

$$p(u_1 \oplus u_2 = +1) = \frac{1 + e^{L(u_1)} e^{L(u_2)}}{(1 + e^{L(u_1)})(1 + e^{L(u_2)})} \quad (10)$$

The probability $p(u_1 \oplus u_2 = -1)$ can then be calculated as

$$p(u1 \oplus u2 = -1) = 1 - p(u1 \oplus u2 = +1) = \frac{e^{L(u_1)} + e^{L(u_2)}}{(1 + e^{L(u_1)})(1 + e^{L(u_2)})} \quad (11)$$

From the definition of log-likelihood ratio eq.(4), it follows directly that

$$L(u_1 \oplus u_2) = \ln \frac{p(u_1 \oplus u_2 = +1)}{p(u_1 \oplus u_2 = -1)} \quad (12)$$

using eqs.(10)and (11), $L(u_1 \oplus u_2)$ is found to be:

$$L(u_1 \oplus u_2) = \ln \frac{1 + e^{L(u_1)} e^{L(u_2)}}{e^{L(u_1)} + e^{L(u_2)}} \quad (13)$$

This result is approximated as in [16]:

$$L(u_1 \oplus u_2) \approx sign(L(u_1))sign(L(u_2)) \times \min(|L(u_1)|, |L(u_2)|) \quad (14)$$

The adition of two “soft” or L-value is denoted by [+] and is defined as

$$L(u_1)\ [+]\ L(u_2) = L(u_1 \oplus u_2) \quad (15)$$

With the following three properties

$$L(u)\ [+]\ \infty = L(u) \quad (16)$$

$$L(u)\ [+]\ -\infty = -L(u) \quad (17)$$

$$L(u)\ [+]\ 0 = 0 \quad (18)$$

By induction, it can be shown that

$$\sum_{\substack{[+]\\ j=1}}^{J} L(u_j) = L\left(\sum_{\substack{\oplus\\ j=1}}^{J} u_j\right) = \ln \frac{p\left(\sum_{\substack{\oplus\\ j=1}}^{J} u_j = +1\right)}{p\left(\sum_{\substack{\oplus\\ j=1}}^{J} u_j = +1\right)} = \ln \frac{\prod_{j=1}^{J}(e^{L(u_j)}+1) + \prod_{j=1}^{J}(e^{L(u_j)}-1)}{\prod_{j=1}^{J}(e^{L(u_j)}+1) - \prod_{j=1}^{J}(e^{L(u_j)}-1)} \tag{19}$$

By using the relation

$$\tanh\left(\frac{x}{2}\right) = \frac{e^x - 1}{e^x + 1} \tag{20}$$

the induction can be simplified to

$$\sum_{\substack{[+]\\ j=1}}^{J} L(u_j) = \ln \frac{1 + \prod_{j=1}^{J} \tanh\left(\frac{L(u_j)}{2}\right)}{1 - \prod_{j=1}^{J} \tanh\left(\frac{L(u_j)}{2}\right)} = 2\tanh^{-1}\left(\prod_{j=1}^{J} \tanh\left(\frac{L(u_j)}{2}\right)\right) \tag{21}$$

This value is very tedious for computing [14]. Thus, it can be approximated as before to

$$\sum_{\substack{[+]\\ j=1}}^{J} L(u_j) = L\left(\sum_{\substack{\oplus\\ j=1}}^{J} u_j\right) \approx \left(\prod_{j=1}^{J} sign(L(u_j))\right) \times \min_{j=1,\dots,J} \left\{ L(u_j) | \right\} \tag{22}$$

It can be seen from eq.(22) that the reliability of the sum of “soft” or L-values is miainly determined by the smallest “soft” or L-value of the terms. The encoded data is now divided into sub-blocks of length 32-bit (N=32), and it direcly multiplied by a generated amplitude matrix given by:

$$Amplitude\ Matrix = \begin{bmatrix} \mathrm{Real}\left(e^{-j(\frac{2\pi n}{N})k}\right) - A1 & if\ imag\left(e^{-j(\frac{2\pi n}{N})k}\right) < 0 \\ \mathrm{Real}\left(e^{-j(\frac{2\pi n}{N})k}\right) + A2 & if\ imag\left(e^{-j(\frac{2\pi n}{N})k}\right) \geq 0 \end{bmatrix} \tag{23}$$

Where N=32 (packet length), n=variable between 1 to N, k=variable between 0: N-1

$0 < A1 \leq 1, \qquad A1 \neq 0\ if\ \ A2 = 0$

$0 < A2 \leq 1 \qquad A2 \neq 0\ if\ \ A1 = 0$

Note that the values of A1 and A2 should not equal to zero, else the inverse of matrix be undetermined. Since the amplitude of signal will increase due to multiplying the transmitted data by AM values, and this will cause a peak to average power ratio (PAPR) problem (especially due to high values in the first column of matrix), so we convert the sign of first column of matrix from positive to negative value. This conversion will reduce the PAPR of the multiplied encoded data by amplitude matrix in the first value of each transmitted vector.

The output data from AM will send to the receiver through the channel which may be AWGN or fading channel.

7.2 Soft Channel Output

From the system model in **Fig. 5**, the information bit u is mapped to the encoded bits x. The encoded bits x are transmitted over the channel and received as y. From this system model, the log-likelihood ratio of x conditioned on y is calculated as [14, 11]:

$$L(x \mid y) = \ln \frac{p(x=+1 \mid y)}{p(x=-1 \mid y)} \tag{24}$$

By using Baye's Theorem, this log-likelihood ratio is equivalent to

$$L(x \mid y) = \ln\left(\frac{p(y \mid x=+1)\,p(x=+1)}{p(y \mid x=-1)\,p(x=-1)}\right) = \ln \frac{p(y \mid x=+1)}{p(y \mid x=-1)} + \ln \frac{p(x=+1)}{p(x=-1)} \tag{25}$$

The channel model is assumed to be flat fading with Gaussian noise. By using this Gaussian pdf $f(z)$

$$f(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-m)^2}{2\sigma^2}} \tag{26}$$

where m is the mean and σ^2 is the variance, it can be shown that

$$\ln \frac{p(y \mid x=+1)}{p(y \mid x=-1)} = \ln \frac{e^{-\frac{E_b}{N_o}(y-a)^2}}{e^{-\frac{E_b}{N_o}(y+a)^2}} = \ln \frac{e^{\frac{E_b}{N_o}2ay}}{e^{-\frac{E_b}{N_o}2ay}} = 4\frac{E_b}{N_o}ay \tag{27}$$

where $\frac{E_b}{N_o}$ is the signal to noise ratio per bit (directly related to the noise variance) and a is the fading amplitude. For nonfading Gaussian channel, a=1.

The log-likelihood ratio of x conditioned on y, $L(x|y)$, is the equivalent to

$$L(x|y) = L_c y + L(x) \tag{28}$$

Where L_c is defined to be the channel reliability

$$L_c = 4\frac{E_b}{N_o}a \tag{29}$$

Thus, $L(x|y)$ is just the weight received value ($L_c y$) summed with the log-likelihood value of x ($L(x)$).

8. SOVA COMPONENT DECODER FOR SCCCs

The SOVA component decoder estimates the information sequence using one of the two encoded streams produced by the SCCC$_s$ encoder. **Fig. 6** shows the inputs and outputs of the of the SOVA component decoder [14, 16].

The SOVA component decoder processes the (log-likelihood ratio) Inputs L(u) and $L_c y$, where L(u) is the a-priori sequence of the information sequence u and $L_c y$ is the weighted received sequence. The sequence y is received from the channel. However, the sequence L(u) is produced and obtained from the proceeding SOVA component decoder there are no a-priori values. Thus, the L(u) sequence is initialized to the all-zero sequence. The SOVA component decoder produces u' and L(u') as outputs where u' is the estimated information sequence and L(u') is the associated log-likelihood ratio ("soft" or L-value) sequence.

The SOVA component decoder operates similarly to the Viterbi decoder except the ML sequence is found by using modified metric. This modified metric, which incorporates the a-priori value, is derived below.

The fandamental Viterbi algorithm searches for the state sequence $S^{(m)}$ or the information sequence $u^{(m)}$ that maximizes the a-posteriori probability $p(S^{(m)}|y)$. For binary (k=1) trellises, m can be either 1 or 2 to denote the survivor and the competing path respectively. By using Bayes' theorem, the a-posteriori probability can be expressed as

$$p(S^{(m)} \mid y) = p(y \mid S^{(m)}) \frac{p(S^{(m)})}{p(y)} \tag{30}$$

since the received sequence y is fixed for metric computation and does not depend on m, it can be discarded. Thus, the maximization result to

$$\max_m p(y \mid S^{(m)}) p(S^{(m)}) \tag{31}$$

The probability of a state sequence terminated at time t is P($\mathbf{S}_t$). This probability can be calculated as [14]

$$P(\mathbf{S}_t) = p(S_{t-1})\, p(S_t) = p(S_{t-1}) p(u_t) \tag{32}$$

where $p(S_t)$ and $p(u_t)$ denote the probability of the state and the bit at time t respectively. The maximization can then be expanded to

$$\max_m p(y \mid S^{(m)}) p(S^{(m)}) = \max_m \left\{ \prod_{i=0}^{t} p(y_i \mid S_{i-1}^{(m)}, S_i^{(m)}) p(S_t^{(m)}) \right\} \tag{33}$$

where $(S_{i-1}^{(m)}, S_t^{(m)})$ denotes the state transition between time i-1 and time i and y_i denotes the associated received channel value for the state transition. After substituting and rearranging,

$$\max_m p(y \mid S^{(m)}) p(S^{(m)}) = \max_m \left\{ p(S_{t-1}^{(m)}) \prod_{i=0}^{t-1} p(y_i \mid S_{i-1}^{(m)}, S_i^{(m)}) p(y_t \mid S_{t-1}^{(m)}, S_t^{(m)}) \right\} \tag{34}$$

Note that

$$p(y_t \mid S_{t-1}^{(m)}, S_t^{(m)}) = \prod_{j=0}^{N} p(y_{t,j} \mid x_{t,j}^{(m)}) \tag{35}$$

Thus, the maximization becomes

$$\max_m \left\{ p(S_{t-1}^{(m)}) \prod_{i=0}^{t-1} p(y_i \mid S_{i-1}^{(m)}, S_i^{(m)}) p(u_t^{(m)}) \prod_{j=1}^{N} p(y_{t,i} \mid x_{t,j}^{(m)}) \right\} \tag{36}$$

This maximization is not changed if algorithm is applied to the whole expression, multiplied by 2, and added two constants that are independent of m. This leads to

$$\max_m \{M_t^{(m)}\} = \max_m \left\{ \begin{array}{l} M_{t-1}^{(m)} + \left[2\ln P(u_t^{(m)}) - C_u\right] + \\ \sum_{j=1}^{N} \left[2\ln P(y_{t,j} \mid x_{t,j}^{(m)}) - C_y\right] \end{array} \right\} \tag{37}$$

Where

$$\frac{M_{t-1}^{(m)}}{2} = \ln\left(p(S_{t-1}^{(m)}) \prod_{i=0}^{t-1} p(y_i \mid S_{i-1}^{(m)}, S_i^{(m)}) \right) \tag{38}$$

and for convenience, the two constants are

$$C_u = \ln p(u_t = +1) + \ln p(u_t = -1) \tag{39}$$

$$C_y = \ln\,(p(y_{t,j} \mid x_{t,j} = +1)) + (p(y_{t,j} \mid x_{t,j} = -1)) \tag{40}$$

After substitution of these two constants, the SOVA metric is obtained as [14]

$$M_t^{(m)} = M_{t-1}^{(m)} + \sum_{j=1}^{N} x_{t,j}^{(m)} L_c y_{t,j} + u_t^{(m)} L(u_t) \tag{41}$$

For systematic codes, mthis can be modified to become

$$M_t^{(m)} = M_{t-1}^{(m)} + u_t^{(m)} L_c y_{t,1} + \sum_{j=2}^{N} x_{t,j}^{(m)} L_{ct,j} y_{t,j} + u_t^{(m)} L(u_1) \tag{42}$$

As seen from eqs. (41 and 42), the SOVA metric incorporates values from the past metric, the channel reliability, and the source reliability (a-priori value).

9. SOVA ITERATIVE SCCC$_S$ DECODER

The received signal from the channel is multiplied by the Inverse of Amplitude Matrix (IAM) given by eq. (23). The output bits after multiplication (y) will now processed by SCCC$_s$. The iterative SCCC$_s$ decoder is composed SOVA component decoders. **Fig. 7** shows the SCCC$_s$ decoder schematic with IAM [7, 11, 16, 17].

Consider the SCCC encoder to be composed of two RSC encoders of rate 1/2. Let $u = [u_1\ u_2\ u_3 \ldots u_k]$ be the information bits input to a constituent RSC encoder and $c = [c_1\ c_2\ c_3 \ldots c_n]$ be the code bits at the output of the encoder. Let $L_a^b(x_c)$ be the LLR generated in the SCCC decoder where a, b, x and c can take on the following values.
a: (i for inner, o for outer) denoting the inner or outer decoder,
b: (a for a priori, in for input, and e for extrinsic and o for output) denoting kind of LLR information,
x: (u for information, c for code) distinguishing between information and codeword bits
c: value denoting the number of bits (information or code bits, depending on x).

The input to the conventional SCCCs decoder is a noisy version of the multiplexed systematic and parity bits of the inner encoder represented as $y = [y_s^{'} y_p^{'}]$. The input y to the decoder is scaled by the channel reliability factor Lc. The order of operation of the decoders is the reverse of that of the encoders. Hence, the scaled input is fed into the inner SOVA decoder first. The operation of the SCCC decoder is iterative and every iteration is composed of two half-iterations (one for each constituent SOVA decoder). During the first half-iteration, the inner decoder accepts the input frame (size≈4*k*) and computes the extrinsic information $L_e^i(u_{2k})$ of the information bits only. The a priori information to the inner decoder is set to zero for the first iteration. The extrinsic information generated by the inner decoder is de-interleaved and forms the input $L_{in}^o(c_{2k})$ to the outer decoder. It is clear that extrinsic information is generated by a decoder with the help of information that is not available to the other decoder (inner encoder parity is not available to the outer decoder) [18].
The a priori information to the outer decoder is always set to zero and is not used. The outer decoder not only produces extrinsic information of information bits $L_e^o(u_k)$ but also computes LLR of code word bits $L_e^o(c_{2k})$. The computation of extrinsic information of codeword bits is the only major difference between the functioning of the SCCC outer decoder as compared to the SCCC inner decoder or any PCCC decoder. The extrinsic information of the information and codeword bits is interleaved and fed back to the inner decoder forming the a priori information for the next iteration. The decoders keep sharing extrinsic information of information and codewords bits in the fashion described above for a number of pre-set iterations. After the last iteration is performed, the outer decoder computes the complete output LLR $L_o^o(u_k)$ of the information bits. A hard-decision is performed on these bits and an estimate of the original information sequence $u^{'}$ is produced [18].

10. PERFORMANCE ANALYSIS

Simulation results for $SCCC_s$ are based on bit error rate (BER) performance over a range of Eb/No with AWGN and Rayleigh fading channels, also in the same graphs, the performance of SCCCs (named Conventional) will be compared with its performance when an amplitude matrix (AM) added at the transmitter and the receiver sides) . The following table shows the AWGN channel and rate 1/2 component RSC codes used in the simulation results.

Fig. 8, represents the performance of SCCCs with iteration number is 10 and frame size (FS) is 100 and different memory size, where BER decreases with SNR increases at increases the encoder memory size. The performance of such system was improved when AM added at the transmitter and the received parts. In all range of SNR, the perfomance of conventional system with AM ouperforms the conventional model without AM.

Fig. 9, represents the SCCCs BER performance for FS=10 and constraint length (K=3) with different number of decoding iterations. From this figure it can be seen that the performance of the conventionl with AM and without AM has been improved as the iteration number incresed. The system with AM has lower BER in all range of SNR than conventional model.
Fig. 10, shows SCCCs BER performance for constraint length (K=3) and 10 decoding iterations with different frame size (with and without AM), where in these figures, with increases SNR, the BER decreases when frame size, encoder memory size, or number of decoding iteration increases. The model with AM still outperforms the conventional model.
Fig. 11 shows that the Rayleigh fading channel degrades BER performance for all SCCCs schemes with FS=1024, K=3, decoding iterations is 10 and semi-random interleaver. Also the performance of system with AM decreases but it stays has better performance than conventional model. The two systems in AWGN channel have less BER than their performances in Rayleigh fading channel.

11. CONCLUSION

The simulation results showed many interesting properties about conventional $SCCC_s$ with and without AM. Some of these important results are listed below:

1. For a fixed $SCCC_s$ encoder, its performance improves as the frame size increases, but the latency of decoding process increases also.
2. For a fixed $SCCC_s$ encoder, its performance improves as the constraint length increases.
3. For a fixed frame size, the $SCCC_s$ performance increases, for a fixed code rate, an increase in constraint length improves the performance.
4. Substantial decoding gain is observed if more than one decoding iteration is used.
5. Rayleigh fading channel degrades BER performance for all $SCCC_s$ schemes.
6. The system performance of conventional SCCCs with amplitude matrix outperforms the performance of system without amplitude matrix in all points given above.

12. REFERENCES

[1] J. G. Proakis, “Digital Communications,” 3rd edition, New York, USA: McGraw-Hill, 1995.

[2] C. Berrou, A. Glavieux, and P. Thitimashima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” in *proceeding of IEE*E *International Conference on communication*, pp. 1064-1070, May 1993.

[3] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: turbo-codes,” IEEE Trans. Commun., vol. 44, pp. 1261-1271. Oct. 1996.

[4] M. C. Valenti, “Turbo Codes and Iterative Processing,” Proc. IEEE New Zealand wireless communication Symposium 98, Nov.1998.

[5] S. Benedetto, D. Divsalar, D. Montorsi, and F. Pollara, “Serial concatenation of interleaved codes: Performance analysis, design and iterative decoding,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 909-926, May 1998.

[6] G. D. Forney, “Concatenated Codes,” Cambridge, MA: MIT Press, 1966.

[7] N. Chandran, “Hybrid ARQ with Parallel and Serial Concatenated Convolutional Codes for Next Generation Wireless Communications,” M.Sc. thesis, West Virginia University, 2001.

[8] C. Fragouli, "Turbo Code Design for High Spectral Efficiency", Ph.D. thesis, University of California, Los Angeles, 2000.

[9] Kuhn and Desmond, "Near-Shannon-limit channel coding using woven convolutional codes," Melbourne University, Nov. 13, 2000.

[10] P. Jung and M. Nasshan, "Dependence of the error performance of turbo-codes on the interleaver structure in short frame transmission system," *Electronics Letters*, vol. 30, pp. 287-288, Feb. 17th 1994.

[11] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429-445, Mar. 1996.

[12] D. Divsalar, and F. Pollara, "Turbo Codes for Deep-Space Communications," *JPL TDA Progress Report 42-120* , Feb. 15, 1995.

[13] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and non-random permutations," TDA Progress Report 42-122, JPL, Aug. 1995.

[14] F. H. Huang, "Evaluation of Soft Output Decoding for Turbo Codes," M.Sc. thesis, Virginia Polytechnic Institute and State University, May 29, 1997.

[15] D. Divsalar and F. Pollara, "Multiple Turbo Code for Deep-Space Communication," *JPL TDA Progress Report* 42-121, JPL, PASADENO, Ca. USA, pp. 66-76, May 1995.

[16] J. Hagenauer, P. Robertson and L. Papke, "Iterative (turbo) decoding of systematic convolutional codes with the MAP and SOVA algorithms," *ITG Conf.*, Frankfurt, Germany, pp. 1–9, Oct. 1994.

[17] Chuan Xiu Huang and Ali Ghrayeb, "A simple remedy for the exaggerated extrinsic information produced by the SOVA algorithm," *IEEE Transactions on Wireless Communications.* vol. 5, no. 5, pp. 996-1002, May 2006.

[18] A.Tripathy, S.S.Pathak and S.Chakrabarti, "SOVA-based Turbo Equalization for ISI Corrupted Indoor Wireless Channels," *IEEE Int. Conf. on Communications, Signal Processing and Networking,* Madras Institute of Technology, Anna University in association with AU-KBC Research Centre, pp.106-111, accessible through *IEEEXplore* database. Feb. 22-24, 2007.

[19] Aruna T., Sant S. Pathak, and Saswat CH., "A low complexity approach to Turbo Equalizer," WSEAS Transactions on Signal Processing, Issue 5, vol. 4, May 2008.

Table (1): Rate ½ RSC Component Codes Used in Simulation Results

Constraint Length (K)	Feed forward Generator (in Octal)	Feedback Generator (in Octal)
3	5	7
4	15	17
5	23	37

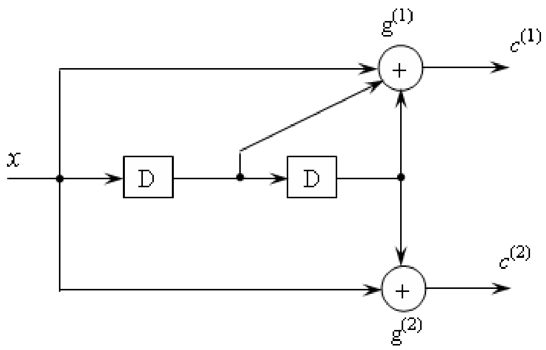

Fig. (1): Atypical rate ½ linear non-systematic convolutional encoder

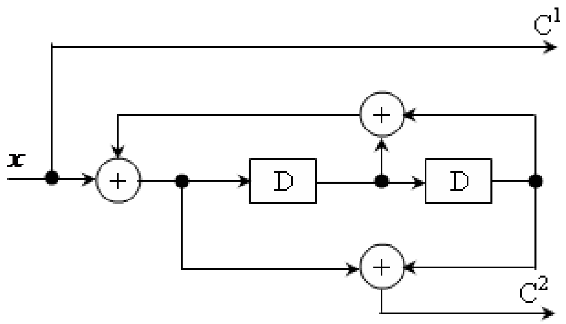

Fig. (2): The RSC encoder obtained from Figure 1 with r =1/2 and k=3

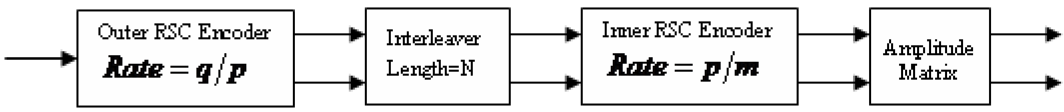

Fig. (3): Serial Conctenated Convolutional Codes with Amplitude Matrix

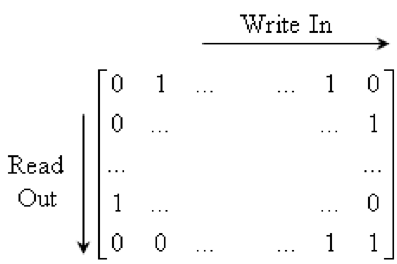

Fig. (4): Block interleaver

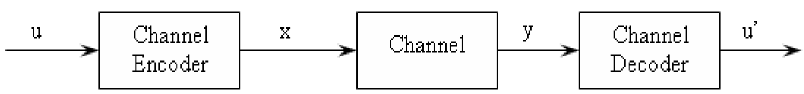

Fig. (5): System model for SOVA derivation

Fig. (6): SOVA component decoder

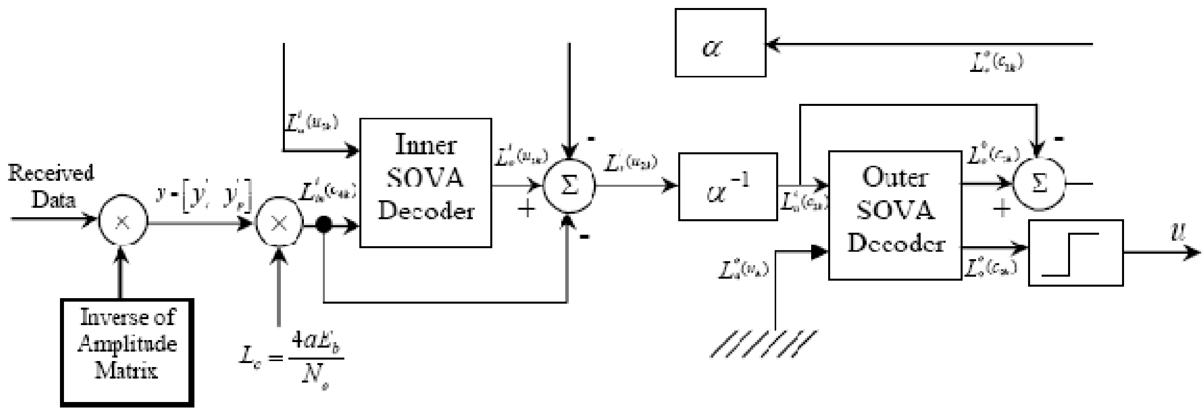

Fig. (7): A SCCC Decoder

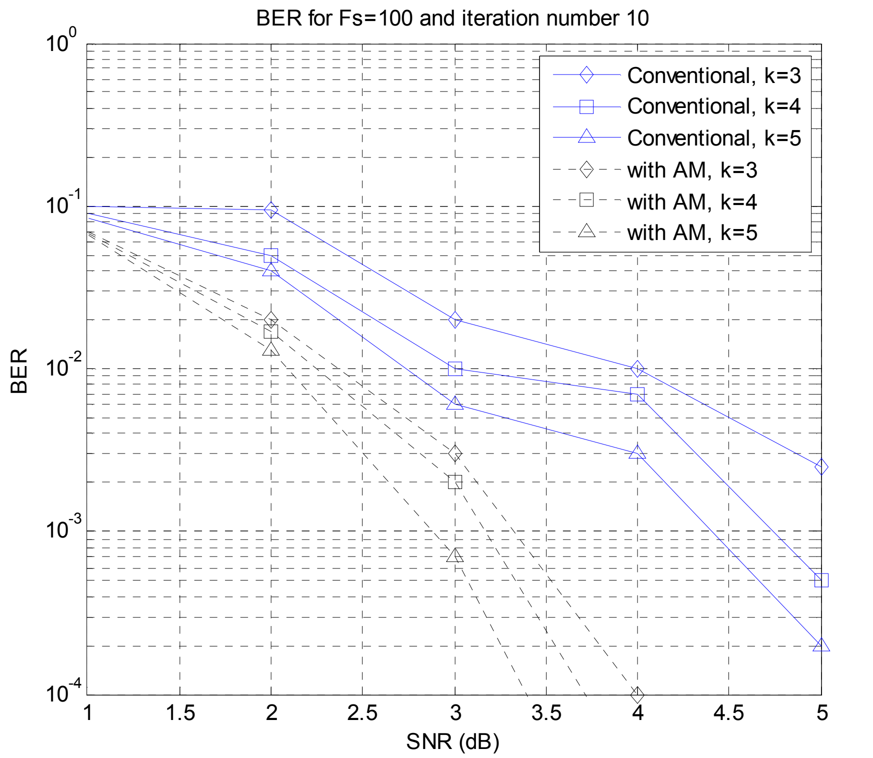

Fig. (8): Performance of Conventional SCCCs (with and without AM) with increases the encoder memory size

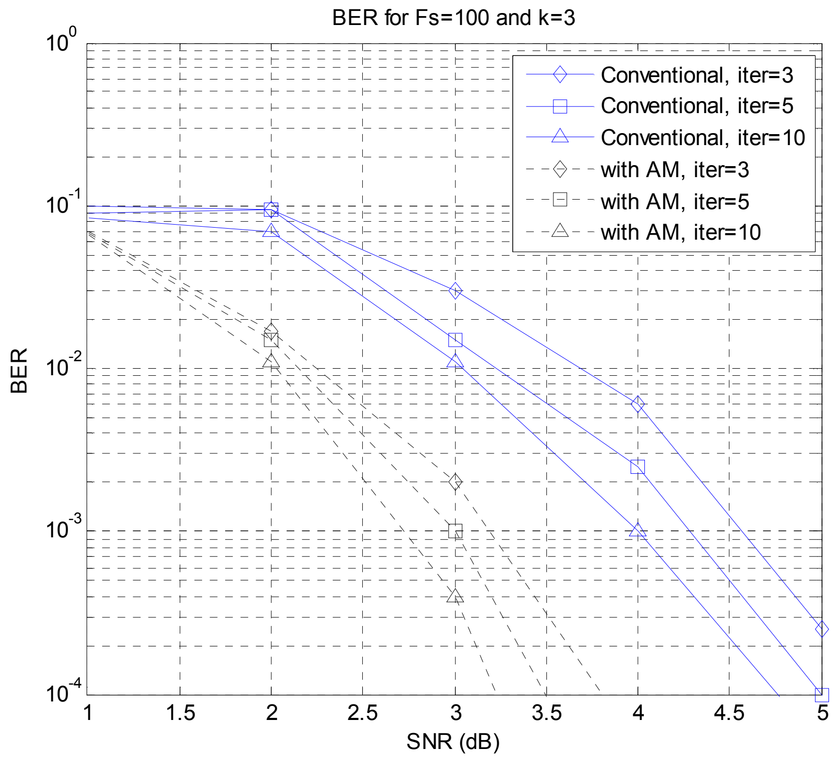

Fig. (9): Performance of conventional SCCCs (with and without AM) with increases the number of iterative decoding

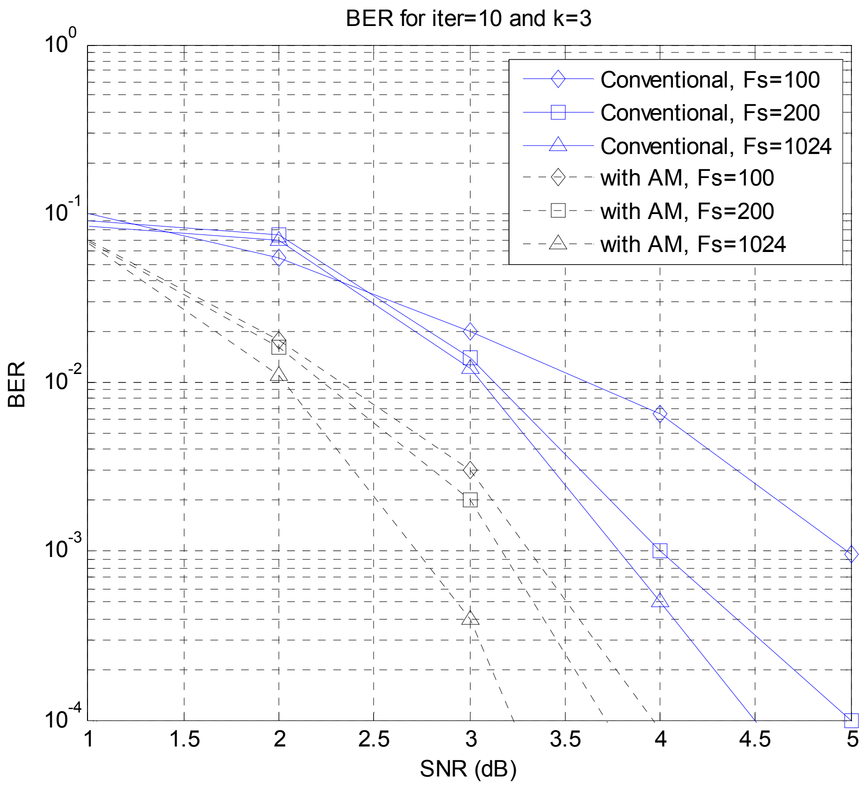

Fig. (10): Performance analysis of SCCCs (with and without AM) with increases the frame size

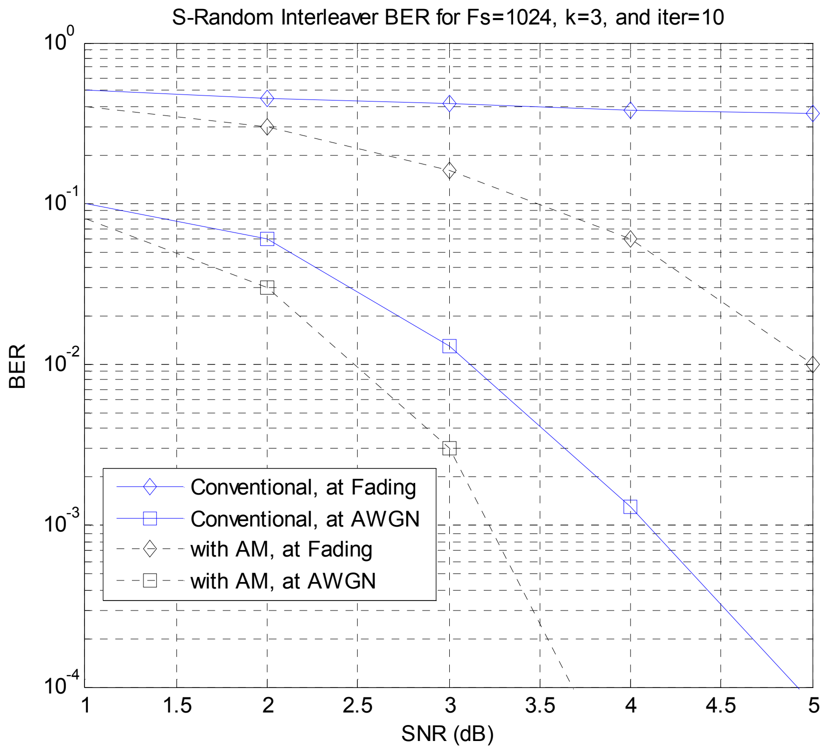

Fig. (11): Performance analysis of SCCCs at the AWGN and fading channel (with and without AM)

تحليل اداء السلسلة المتسلسلة الملتفة المشفرة بالاعتماد على المصفوفة السعوية

م. بلاسم سالم سميط
قسم الهندسة الكهربائية، كلية الهندسة، جامعة الانبار

د. صالح محمد صالح
قسم الهندسة الكهربائية، كلية الهندسة، جامعة الانبار

الخلاصة:

السلسلة المتسلسلة الملتفة المشفرة (SCCCs) تم بنائها باستخدام سلسلة متسلسلة مكونة من اثنان من المنظّمات التكرارية الملتفة (RSC) للمشفرات، ومفصولتان بمبدل مواقع الادخالات (interleaver). ان الاثنان من مشفرات (RSC) يعتمدان على معايير انهاء التعريشة قد تم بنائهما وتقييم ادائهما تحت مختلف الضروف والشروط المتعلقة بادائهما. البيانات الخارجة من المشفر تم ضربها بمصفوفة سعوية (AM) بجهة الارسال، ومعكوس المصفوفة السعوية بجهة الاستلام. تخمين الوثوقية، الامكانية الجبرية المحتملة، الخوارزمية المرنة لقناة الاخراج لفيتربي (SOVA) تم اختبارها. ثم ان خوارزمية Viterbi المُعَدَّل المترية التي تَدْمجُ المعلوماتَ الإستنتاجيةَ إستعملتْ لتَرْجَمَة مُشْتَقَّةُ ال SOVA. أي تطبيق ذاكرةِ منخفض مِنْ جهاز فكّ رموز SOVA مُقَدَّمُ. ان SOVA التكراري الى SCCCs مَوْصُوفُ بالأمثلةِ التوضيحية. إنّ سلوكَ مخططِ جهاز فكّ رموز SCCCs للمشفر مُجرّبُ تحت الظروفِ المختلفةِ مَع مصفوفة السعة وبدونها في حالة كون قناة نقل الاشارة هي قناة الضوضاء الابيض التراكمية لكاوز (AWGN) وقنوات الخبو Rayleigh بمختلف القيم لحجوم الإطارِ (Fs) وطول قيدِ (k). بينت النَتائِجُ ان أداءُ النظامِ مَع مصفوفة السعة قد فاق اداء النظام التقليدي الآخر الذي عَملَ بدون مصفوفة السعة.