On the Error Floor Performance of SCTCM Systems with Non-Recursive Inner Codes

Anders Nilsson, Student Member, IEEE, and Tor M. Aulin, Fellow, IEEE
Chalmers University of Technology, Department of Computer Science and Engineering
Telecommunication Theory Group
SE-412 96 Gothenburg, Sweden
Email: {anders,tor} @ce.chalmers.se

Abstract—It has previously been shown that serially concatenated trellis coded modulation (SCTCM) with a non-recursive inner convolutional code can achieve convergence close to capacity [1]. In this paper, we supplement the analysis by comparing recursive and non-recursive inner codes with respect to the bit error rate (BER) of the error floor region. We show that the high error floor, commonly attributed to serially concatenated codes using a non-recursive inner code, can be reduced significantly through careful code design. The considered system is iteratively decoded SCTCM with an outer single parity check (SPC) code and a two-state rate 3/4 inner convolutional code whose output is mapped to 8PSK. Simulation results are shown for the additive white Gaussian noise (AWGN) channel.

I. INTRODUCTION

The use of non-recursive inner convolutional codes in SCTCM was first considered in [1], where it was shown that one can obtain close to capacity performance through serial concatenation of an SPC code and a non-recursive convolutional code. The proposed SCTCM system of [1], which has a throughput of 2 bits/symbol and uses 8PSK signaling, achieves convergence at a signal-to-noise ratio (SNR) below 3dB. However, serially concatenated codes using a non-recursive inner code are often impaired by a high error floor. Hence, in this paper we supplement the analysis of [1] by investigating how the error floor can be reduced.

The structural difference between a recursive and a non-recursive encoder is illustrated in Fig. 1. Recursive encoders are characterized by a feedback connection in the encoder, which non-recursive encoders are without. The recursive and non-recursive encoders convey different code properties, which becomes especially evident when studying the error events of the two code structures. By definition, an error event starts when the detected path through the trellis diagram departs from the correct path, and ends when the detected path again merges with the correct path. In our case, each trellis transition of the inner code is associated with three input bits and an 8PSK output signal. Hence, the error events are associated with two distances, the input Hamming distance of the output Euclidean distance from the correct path. The error events of the inner code are of fundamental importance for the BER performance of the error floor region. The larger the input and output distances from the correct path, the less likely the error event is to occur. Error events of recursive codes, with few exceptions, always have an input distance of at least $d_{Z} = 2$, whereas for non-recursive codes error events of input distance $d_{Z} = 1$ always exist. In [2], Benedetto et al. correctly explained that non-recursive inner codes will not experience as much interleaving gain as recursive inner codes. Interleaving gain enables improved performance with larger input block sizes (i.e., the interleaver size). The interleaving gain for non-recursive inner codes is often insignificant even for smaller block sizes. Hence, the general conception in coding theory has been that the inner code of a serially concatenated code has to be recursive. However, inner codes with minimum input distance $d_{Z_{\text{min}}} = 1$ have proven to yield good performance in SCTCM [1]. Likewise, bit interleaved coded modulation with iterative decoding (BICM-ID), which can be seen as SCTCM with a one-state non-recursive inner code, can achieve impressive coding gains [3]. The problem with non-recursive inner codes is that the $d_{Z} = 1$ error events induce an irreducible error floor, for which an increased input block size cannot improve the performance. In this paper, we will show that the unwanted effects of $d_{Z} = 1$ error events can be made negligible for short and medium sized codes. We will see that input distance $d_{Z} = 1$ error events will not contribute significantly to the BER of the proposed system unless the input block size is very large.

1The theoretical capacity of 2 bits/symbol 8PSK is just below 2.8 dB [8]
2Note that the input distance is commonly referred to as the input weight.
3Error events of input distance $d_{Z} = 1$ can exist for recursive codes with parallel transitions.
II. SYSTEM MODEL

In this paper, random variables are denoted by bold face letters, realizations of random variables by non-bold italic letters, and vectors by underlined letters.

The block diagram of the considered system is illustrated in Fig. 2. The first component of the SCTCM encoder is an outer (3,2) single parity check (SPC) encoder. At each time instant we have two input bits \((u_{j,t})_{j=0}^{3} = (u_0, u_1, u_2, u_3)\). The SPC encoder produces three output bits \((y_{j,t})_{j=0}^{3} = (y_0, y_1, y_2)\), for which \(y_0 = u_0, y_1 = u_1, y_2 = u_2 \oplus u_1\). The input and output bits can also be represented in symbol form. For example, input symbol \(U = 1100\) and output symbol \(Y = 011\) are outcomes of the uniformly distributed binary random variables \((u_{j,t})_{j=0}^{3}\) and \((y_{j,t})_{j=0}^{3}\), respectively. Likewise the corresponding bits \((\hat{u}_{j,t})_{j=0}^{3}\) and \((\hat{y}_{j,t})_{j=0}^{3}\) are outcomes of the uniformly distributed binary random variables \((c_{j,t})_{j=0}^{3}\) and \((d_{j,t})_{j=0}^{3}\), respectively.

The decoder computes the conditional probability functions of the random variables above given the received vector \(R\). For easier notation we define the new symbol random variables \(U_t, Y_t, Z_t, Z_t, C_t, C_t\), and analogously the binary random variables \((\hat{u}_{j,t})_{j=0}^{3}, (\hat{y}_{j,t})_{j=0}^{3}, (\hat{z}_{j,t})_{j=0}^{3}\) and \((c_{j,t})_{j=0}^{3}\) to correspond to these conditional probability functions.

The inner and outer decoders are realized by the soft-input soft-output (SISO) module [6]. The straightforward implementation of the SISO module works at symbol level, hence, to make bit deinterleaving feasible we have to transform each symbol random variable \(Z_t\) to the binary random variables \((\hat{y}_{j,t})_{j=0}^{3}\). With a slight change of notation from [6] we have

\[
p_{\hat{y}_{j,t}}(x) = H_{\hat{y}_{j,t}} \sum_{Z: \hat{z}_{j,t} = x} p_{Z|\hat{y}_{j,t}}(\hat{z}_{j,t}) \prod_{i=0}^{j} p_{\hat{z}_{i,t}}(\hat{z}_{i,t}),
\]

where \(H_{\hat{y}_{j,t}}\) normalizes the sum of all probabilities to add to one. To obtain the inputs of the SISO modules we perform bit-to-symbol transformation. Statistical independency is assumed due to the deinterleaving, and thus we have

\[
p_{\hat{y}_{j,t}}(\hat{y}_j) = \prod_{j=0}^{2} p_{\hat{y}_{j,t}}(\hat{y}_j).
\]

The transformations necessary for the reinterleaving in the feedback are obtained analogously. By letting the transformations be denoted \(S2B\) (symbol-to-bit) and \(B2S\) (bit-to-symbol), we illustrate the iterative decoder as in Fig. 2.

The SPC encoder problem is denoted \(S2B\) (symbol-to-bit) and \(B2S\) (bit-to-symbol), to distinguish the SISO input sequences from the SISO output sequences.

Given the received vector \(R = S + N\), a posteriori probabilities (APP’s) for each 8PSK symbol is computed. Demapping of the 8PSK symbols gives a sequence of probability functions \(p_{C_j | R}(C_0, p_{C_j | R}(C_1), \ldots)\) that is fed to the iterative decoder.

The decoder views the inputs and outputs of each constituent encoder as outcomes of discrete random variables. The encoder symbols \(U_t, Y_t, Z_t, C_t\) are outcomes of the uniformly distributed random variables \(U_t, Y_t, Z_t, C_t\). The input sequence is described by a two-dimensional vector \((\hat{y}_{j,t})_{j=0}^{3}\), where the mapping between bits and symbol is binary to decimal conversion. The input at the discrete time instant \(t\) is denoted \(U_t\) or \((u_{j,t})_{j=0}^{3}\). Hence, the input sequence is described by \(U = (U_0, U_1, \ldots, U_{k-2}), U_{k-2}\), where \(k\) is the input block size in bits. The encoder output sequence \(Y\) is fed to an in-line bit interleaver [4], that permutes the order of the bits. The advantages of in-line interleaving is thoroughly explained in [5]. The interleaver output sequence \(Z\) is encoded by an inner CE of rate 3/4, whose output \(C\) is mapped to an 8PSK symbol. Since the inner code rate is 3/4, each 8PSK symbol will be associated with two output symbols \(C_i\).

The 8PSK symbol is transmitted with signal energy \(E_s\) over the AWGN channel. There are two information bits per 8PSK symbol, and thus the energy per bit is \(E_s = E_s/2\). The AWGN waveform channel can be converted into an equivalent vector channel model [7]. The 8PSK symbol at time \(t\) is then described by a two-dimensional vector \(z_t = (s_{1,t}, s_{2,t})\). The transmitted sequence of 8PSK symbols is represented by the vector \(S\). The transmission is disturbed by an additive white Gaussian noise \(N\), for which each element is an outcome of a two-dimensional Gaussian random variable \(u_t\), with the density function

\[
f_{U}(u_t) = \frac{1}{\pi N_0 e^{\frac{|u_t|^2}{N_0}}},
\]

where \(\pi N_0\) normalizes the sum of all probabilities to add to one. To obtain the inputs of the SISO modules we perform bit-to-symbol transformation. Statistical independency is assumed due to the deinterleaving, and thus we have

\[
p_{\hat{y}_{j,t}}(\hat{y}_j) = \prod_{j=0}^{2} p_{\hat{y}_{j,t}}(\hat{y}_j).
\]

The transformations necessary for the reinterleaving in the feedback are obtained analogously. By letting the transformations be denoted \(S2B\) (symbol-to-bit) and \(B2S\) (bit-to-symbol), we illustrate the iterative decoder as in Fig. 2.

Note that the outer SPC code is a block code, but can be seen as a one-state convolutional code with four parallel
transitions assigned to the symbols $Y \in \{0, 3, 5, 6\}$. The extrinsic feedback probabilities for this outer code are

$$p_{Y_t}(\hat{Y}) = \begin{cases} 0.25, & Y = 0, 3, 5 \text{ and } 6 \\ 0, & Y = 1, 2, 4 \text{ and } 7 \end{cases}.$$  \hspace{1cm} (4)

The extrinsic probabilities are constant, and therefore not necessary to compute. We can therefore remove the feedback of the outer SISO module, as in Fig. 2. Note, however, that the constraints of the outer code is available in the iterative decoder via the feedback probabilities of (4) combined with the S2B transformation of the feedback loop.

III. RECURSIVE VS. NON-RECURSIVE INNER CODES

The recursive and the non-recursive inner codes that were illustrated in Fig. 1 can also be viewed as state dependent mappers, where the input symbol $Z$ is mapped to an 8PSK symbol depending on the encoder state $\sigma$. The state dependent mappers for the two considered codes are illustrated in Fig. 3. By using this description we obtain an immediate overview of the important properties of the code.

As we have mentioned, each error event is associated with two distances, namely the input Hamming distance $d_Z$ and the output Euclidean distance $d_C$ from the correct path. The trellis sections of Fig. 3 show the error events that correspond to the minimum input distance $d_{Z_{min}}$. The error events with the minimum input distance $d_{Z_{min}}$ are the most likely to occur. These error events will become less likely to occur if they are associated with a large output distances $d_C$ from the correct path. For easier notation we now define the conditional output distance $d_{C|Z}$. Hence, the distance $d_{C|Z=1}$ refers to the output distance associated to error events of input distance $d_Z = 1$.

A. Code Design

A design rule for the recursive inner code of a serially concatenated convolutional code was proposed in [2]. The design rule states that the free effective distance of the recursive inner code should be maximized, in order to obtain maximum interleaving gain. By definition, the effective free distance is the same as the conditional minimum output distance $d_{C_{min}|Z=2}$. Similarly, design criteria for the inner mapping of BICM-ID were proposed in [3], where the conditional minimum output distance $d_{C_{min}|Z=1}$ of the inner mapping is maximized, in order to obtain low error floors. These two design principles have in common that they both maximize the conditional minimum output distance $d_{C_{min}|Z_{min}}$.

The principle of maximizing $d_{C_{min}|Z_{min}}$ can be directly used for any non-recursive inner convolutional code. However, we should be aware that if $d_{C_{min}|Z=1}$ is large, the error floor might not always be dominated by $d_Z = 1$ error events. Therefore we might also need to consider the output distance $d_{C_{min}|Z=2}$ of the inner code, especially when input block size is small. In Section III-B, we investigate further how error events of different input distance $d_Z$ contribute to the error floor for various block sizes.

Let us now analyze the design of the two inner codes of Fig. 3. Note first that both inner codes A and B have the input symbols $Z_1 = \{0, 3, 5, 6\} = \{000_2, 011_2, 101_2, 110_2\}$ and $Z_2 = \{1, 2, 4, 7\} = \{001_2, 010_2, 100_2, 111_2\}$ assigned to the respective parallel transitions. By using these sets of input symbols, parallel transitions will always differ in two bit positions and consequently we avoid input distance $d_Z = 1$ error events over one symbol interval.

Inner code A is recursive and has the minimum input distance $d_{Z_{min}} = 2$. To obtain maximum interleaving gain, i.e. the lowest possible error floor, we need to associate the $d_Z = 2$ error events with a large output distance $d_C$. For recursive convolutional codes there exists a large number of error events that have input distance $d_Z = 2$, and this number increases linearly with the block size. Luckily, however, we will only need to consider the error events that will be associated with the smallest output distances, and we can therefore restrict ourselves to error events within two symbol intervals, as illustrated in the trellis section of Fig. 3a.

The 8PSK signal set is divided into two subsets of four signals, that each correspond to four parallel transitions. These subsets are illustrated in Fig. 3a by the filled and empty signal points of the state dependent mapper. The subsets are chosen to maximize the distance $d_{C|Z=2}$ between parallel transitions. The mappings of state 0 and 1 of inner code A, can now
be obtained by maximizing the conditional output distance $d_{C|Z=2}$ over two symbol intervals.

Inner code B is non-recursive and has the minimum input distance $d_{Z_{min}} = 1$. Hence, first and foremost we need a large conditional output distance $d_{C_{min}|Z=1}$. Note that the inputs $Z_1 \in \{000_2, 011_2, 101_2, 110_2\}$ always drive the encoder to state 0 and the inputs $Z_2 \in \{001_2, 010_2, 100_2, 111_2\}$ to state 1. The error events can therefore merge with the correct path again without any further increase of the input distance $d_Z$ for non-recursive codes. In order to maximize $d_{C_{min}|Z=1}$, the distance between an input symbol $Z$ in the mapping of state 0 and 1 therefore needs to be as large as possible. For 8PSK this is achieved by rotating the mapping of state 0 by $\pi$ radians to obtain the mapping of state 1.

The conditional output distance $d_{C|Z=2}$ is also of interest and therefore inner code B uses the same partitioning of the 8PSK signal constellation as inner code A does. This is done in order to maximize $d_{C|Z=2}$ over one symbol interval. Note that the only $d_Z = 2$ error events that exist for the non-recursive inner code are those corresponding to the parallel transitions.

The design of inner codes A and B have now been explained. However, to emphasize the importance of the conditional minimum output distance $d_{C_{min}|Z=1}$ and the effect it has on the error floor, we will consider yet another non-recursive inner code. Inner code C is illustrated by its state dependent mapper in Fig. 4. It is very similar to inner code B, however, the mapping of state 1 is here obtained by rotating the mapping of state 0 by $\pi/2$ radians rather than $\pi$ radians. Hence, the conditional minimum output distance $d_{C_{min}|Z=1}$ for inner code C is not maximized.

### B. Error Event Occurrence

In the design of the non-recursive inner encoder, we had to associate the error events of input distances $d_Z = 1$ with a large output distance $d_C$ in order to avoid a high error floor. We will now evaluate the effect of this by looking at the occurrence of the $d_Z = 1$ error events in the error floor region for various input block sizes $L$.

Let us start by describing the methodology we have used in the error event analysis. To find out the occurrence of different error events we simulate a large number of blocks for the SCTCM systems in question. We are only interested in the error events that contribute to the BER of the error floor region and therefore we do the simulations at an $E_b/N_0$ of $4\text{dB}$. At the occurrence of a bit error, we record the error events that caused the actual bit error. From the recorded results we can evaluate the contribution of error events with different input distance $d_Z$. Our aim is to find out how the $d_Z = 1$ error events contribute to the error floor. Hence, we will only simulate the non-recursive inner codes B and C, since the recursive inner code A has no $d_Z = 1$ error events. The error floor for the SICTCM system using inner code A will be dominated by $d_Z = 2$ error events.

Through simulation we obtain the curves of Fig. 5. The two curves indicate the contribution of $d_Z = 1$ error events to the BER of the error floor for inner codes B and C. Empirically, one can see that only error events of $d_Z \leq 2$ contribute significantly to the error floor of the considered block sizes. Hence, we can interpret what is above the curves as the fraction of $d_Z = 2$ error events, and what is below as the fraction of $d_Z = 1$ error events. For inner code B it is evident that the large conditional output distances $d_{C|Z=1}$ reduce the occurrence of $d_Z = 1$ error events. In fact, $d_Z = 2$ error events are clearly dominating the BER of the error floor region for smaller block sizes. For inner code C we had intentionally associated the error events of input distance $d_Z = 1$ with a smaller output distance $d_C$. Analogously to BICM-ID, this will result in a higher error floor [3]. As the dashed curve of Fig. 5 indicate, the error floor is here dominated by $d_Z = 1$ error events for all input block sizes $L$ that we have considered.

### IV. Simulation Results

The outer SPC code in concatenation with inner codes A, B and C, are here referred to as the SICTCM systems SPC-A, SPC-B and SPC-C. Transmission over the AWGN channel is considered. Three input block sizes $L$ are used, 10000 bits, 100000 bits and 1000000 bits. The decoder uses a maximum of 300 iterations, although only a small fraction of these are required in the error floor region$^1$.

Fig. 6 shows the simulated BER performance of systems SPC-A and SPC-B for various input block sizes. Both inner codes A and B are designed to obtain the lowest possible error floor. For input block size $L = 10000$, the error floor performance of systems SPC-A and SPC-B is essentially equal. The slightly higher error floor of system SPC-B could be explained by the fact that the recursive inner code A has a larger number of $d_Z = 2$ error events that are associated with

$^1$A stopping criterion was used to speed up the simulations.
a small output distance $d_C$. The only $d_Z = 2$ error events of non-recursive inner code B are those corresponding to parallel transitions.

If we now look at the BER performance for the input block sizes $L = 10000$ bits and $L = 1000000$ bits, we can see that the error floor of system SPC-A continues to reduce with an increased block size. A reduction of the error floor of system SPC-B due to the increased block size is also evident\(^6\). However, for SPC-B this reduction comes to an end when the $d_Z = 1$ error events start to dominate the BER in the error floor region (compare with Fig. 5).

Fig. 7 shows the simulated BER performance of system SPC-C. As was intended, the smaller conditional minimum output distance $d_{C_{min}}|Z=1$ results in a higher error floor. For system SPC-B, we had a quite significant reduction of the error floor when the input block size was increased from $L = 10000$ bits to $L = 100000$ bits. This reduction does not occur for system SPC-C, where the error floor remains firm despite the increased block size. This can easily be explained by the curves of Fig. 5. For inner code B the fraction of $d_Z = 1$ error events is small at input block size $L = 10000$ bits, whereas for inner code C, the error floor is already dominated by $d_Z = 1$ error events. The irreducible error floor has thus already been reached at an input block size of 10000 bits, and no further increase of the block size will be able to reduce it.

Finally, one should note that both inner codes A and B have been designed mainly with respect to the error floor region. The waterfall region has not been considered in the design of the two inner codes, and hence one should not draw conclusions based on the earlier convergence of system SPC-B. In general, however, non-recursive inner codes can attain a little bit earlier convergence than recursive inner codes. For example, the proposed SCTCM system of [1], which uses a non-recursive inner code, has convergence below 3dB. The earlier convergence of the non-recursive codes can intuitively be explained by the fact that an input symbol $Z$ will always drive the non-recursive encoder to the same state (see Fig. 3b).

\(^6\)Note that the block error rate of SPC-B with 1000000 bits is very high. The large block size is used to illustrate the irreducible error floor of SPC-B. The block error rate of SPC-B for 10000 bits and 100000 bits is, however, comparable to that of SPC-A.

The state probabilities obtained from the forward recursion of the inner SISO module will therefore distinguish the two states from each other at a lower $E_b/N_0$.

V. CONCLUSIONS

We have in this paper shown that an SCTCM system using non-recursive convolutional codes can obtain low error floors, matching those of recursive inner codes for short and medium sized codes. We do not refute the theory of [2], instead we have shown that by careful design of the non-recursive inner code one can largely avoid the unwanted effects of $d_Z = 1$ error events. The simulation results have shown that the $d_Z = 1$ error events will not contribute significantly to the error floor region until a BER of about $10^{-6}$. This would require input block sizes well above $L = 100000$ bits for the considered system. We have further shown that low error floors can be obtained by two very simple component codes. The outer SPC code allows us to leave the outer SISO module out from the iterative decoding cycle. Similarly to BICM-ID our system then decreases the decoding complexity by using only one SISO module.

REFERENCES


