Turbo Decoding of Concatenated Channel Coding and Trellis Shaping for Peak Power Controlled Single-Carrier Systems

Makoto Tanahashi* and Hideki Ochiai†
Department of Electrical and Computer Engineering,
Yokohama National University
79-5 Tokiwadai, Hodogaya, Yokohama, Kanagawa 240-8501, Japan
Email: *makoto@ochiailab.dnj.ynu.ac.jp, †hideki@ynu.ac.jp
†Corresponding author

This paper was in part presented at the IEEE Global Communications Conference 2007 (GLOBECOM 2007), Washington D.C., November 2007.

Abstract

The trellis shaping (TS) has found its application in the peak power control of band-limited single-carrier signals. Our recent work has demonstrated that a well-designed TS can control the symbol transitions such that the output signal has almost constant envelope, which significantly alleviates the linearity requirements of power amplifiers. Compared to transmission without constellation shaping, however, the TS involves signal constellation expansion exclusively for peak power control. Therefore, unlike the trellis coded modulation (TCM) that increases the minimum Euclidean distances (MED), the TS decreases the MED, thus incurring the increase in signal-to-noise power ratio (SNR) required for achieving a certain error rate. In this paper, in order to overcome this drawback, we propose a serial concatenation of coding and shaping together with an effective decoding algorithm that utilizes the memory effect (i.e., error correcting capability) of the shaped symbols. The achievable performance of the proposed system is analyzed in terms of the average mutual information. The simulation results demonstrate that the iterative decoding of the proposed concatenated system with outer convolutional codes and inner trellis shaping offers significant performance gain.

Index Terms
I. INTRODUCTION

Controlling the peak-to-average power ratio (PAR) of transmission signal has been one of the important issues when designing practical communication systems. In order to linearly amplify the signal with a large dynamic range, the power amplifier (PA) should be operated with a large back-off, which substantially reduces its power-added efficiency (PAE). Since an inefficient operation of the PA is prohibitive for battery-driven transmitters, the system with a large PAR, such as orthogonal frequency-division multiplexing (OFDM), has been rarely adopted for mobile terminals.

The PAR issue has been well recognized in the framework of OFDM systems, but this is also the case for the band-limited single-carrier systems. Even with phase shift keying (PSK) modulation, the single-carrier signal exhibits the PAR typically higher than 3 dB due to the existence of the pulse-shaping filter. In the case of the single-carrier systems, the occurrence of high PAR can be avoided by judiciously controlling the symbol transition patterns. The use of trellis shaping (TS), originally proposed by Forney [1] for the purpose of reducing the average power of high order quadrature amplitude modulation (QAM) systems, is one approach that can control this symbol transition efficiently by the Viterbi algorithm. The use of trellis shaping for peak power control has been studied in the framework of OFDM [2, 3] as well as the single-carrier systems [4, 5]. In our previous work [6], we have demonstrated that the TS can control the envelope of the band-limited single-carrier PSK signals with an almost constant dynamic range, even in the presence of a narrow pulse-shaping filter (with roll-off factor as low as 0.1).

One major drawback of the TS for the purpose of peak power control is the reduced transmission rate due to the constellation redundancy imposed by shaping process. If one attempts to compensate for this rate reduction by expanding the signal constellation, the signal-to-noise power ratio (SNR) required for a certain bit error rate (BER) in turn increases. Consequently, the reduced transmission rate is translated into the SNR penalty. In a typical case, the TS adds one-bit redundancy to every transmitted symbol, which implies that the constellation size is doubled and the increase in terms of the required SNR is as large as 3 dB.

Overcoming this SNR penalty is the main subject of this paper. To this end, we first propose
a decoding method that exploits the memory effect of the trellis shaped symbols. The peak controlling TS imposes a constraint on successive symbol transition patterns such that the resulting signal has low PAR. As a result, symbol transitions with small phase difference are observed very frequently, whereas antipodal transitions are very unlikely [6]. In other words, the output of the TS encoder has a memory, similar to convolutionally encoded sequences. This memory allows us to correct erroneous data and we may expect some coding gain, similar to the concept of the constraint gain [7].

Nevertheless, it turns out that the coding gain achieved by the TS is relatively small. Therefore, we propose a serial concatenation of the channel coding and TS as sketched in Fig. 1 (a), which enables us to fully exploit the inherent coding effect of the TS. In many practical communication systems, the use of channel coding is necessary, and capacity-approaching error-correcting codes such as turbo codes [8] and its serial concatenation counterpart [9] are being employed. In the literature, coding and shaping are often designed to be separated as depicted in Fig. 1 (b) [5, 10]. If the trellis shaping is used for average power reduction of high-order QAM signals, this parallel design is reasonable since the coding part guarantees nearly reliable channel so that the shaping part just minimizes transmission energy for a fixed information rate, i.e., coding gain and shaping gain are regarded as separable entities [1]. However, in the scenario of this paper, the primary role of shaping is to control the signal peak power and we wish to exploit this shaping process also for error correction. This motivates us to adopt a serial concatenation of the channel coding and TS together with maximum a posteriori probability (APP) estimation based on iterative decoding [9]. Simulation results demonstrate that the proposed serially concatenated system exhibits bit error performance with a so-called waterfall and a large amount of coding gain, similar to that of the turbo codes.

In order to justify the significant coding gain observed in the simulation results, we also analyze an achievable information rate of the transmission system with the presence of a given shaping constraint based on the average mutual information (AMI). It will be shown that the waterfall region of the BER obtained by simulation is close to the performance limit indicated by the AMI.

There are several related studies found in the literature. Comparisons of parallel and serial concatenation designs are given in [11], which observes the optimality of the serial concatenation suggested by Gallager [12]. In [11, 13], signal mappers for the (average power reducing) shaping
serially concatenated with channel coding are studied, together with its joint iterative processes. More recently, in [14], concatenation of an outer LDPC code and an inner trellis code which matches Markov input constraints is studied. However, the main subject of these studies is on the inter-symbol interference (ISI) or partial response channels. To the best of the authors' knowledge, there have been no studies that exploit symbol transition control for both PAR reduction and error correction in the single-carrier signaling framework.

This paper is organized as follows: In Section II, we briefly review the notation of trellis shaping and describe a specific serially concatenated coding and shaping system considered throughout the paper. Section III introduces a Markov model of the trellis shaped symbols, which is used as an underlying principle throughout the paper. Section IV analyzes the performance of the uncoded shaping system in terms of the BER and that of the coded shaping system in terms of the achievable information rate based on the AMI. The proposed decoder for trellis shaped symbols is presented in Section V, followed by the simulation results in Section VI. Finally, concluding remarks are given in Section VII.
II. SYSTEM DESCRIPTION

Throughout the paper, for simplicity, we focus on the specific 8-PSK trellis shaping system concatenated with an outer channel coding shown in Fig. 2. In this system, a binary information sequence \( b \) is first encoded to \( c \) by rate \( r_c = 1/2 \) channel encoder, and the subsequent pseudo-random interleaver \( \Pi \) permutes \( c \) to another binary sequence \( s \), which is then input to the TS encoder. The TS encoder transforms \( s \) into \( z \) with an addition of one bit redundancy such that \( z \) satisfies a certain property. In our scenario, \( z \) should be chosen such that the signal waveform after modulation and pulse shaping filtering exhibits low PAR. In this specific example, the shaping rate is \( r_s = 2/3 \) and the overall coding/shaping rate is equal to \( r = r_s r_c = 1/3 \). Finally, \( z \) is mapped onto 8-PSK signal constellation. In this paper, an \( M \)-PSK constellation is represented by

\[
S_M = \{ S^i = e^{j2\pi i/M} | 0 \leq i < M \}.
\]  

(1)

Since each symbol carries 3 bits, the transmission rate in this case is one information bit per symbol. The following subsections briefly describe the principle of TS and its conventional decoding method.
A. Principle of Trellis Shaping for Peak Power Control

We summarize the principle and notation of the TS. First, we obtain three kinds of convolutional encoders: shaping encoder \(G_s\), its syndrome former \(H^T_s\), and its left inverse \((H^{-1}_s)^T\) (hereafter called inverse syndrome). For a given \(G_s\), the matrices \(H^T_s\) and \((H^{-1}_s)^T\) are determined such that

\[
G_sH^T_s = 0, \quad (H^{-1}_s)^TH^T_s = I
\]

where \(0\) and \(I\) are corresponding zero vector and identity matrix, respectively. In addition, these matrices should be designed such that \(H^T_s\) has no recursive term. Otherwise, catastrophic error propagation will occur at the receiver [1, 4, 5]. In [1], it is shown that there always exists \(H^T_s\) with no recursive term.

In the case of the shaping system given in Fig. 2, these three matrices are arranged as sketched in Fig. 3. In this scenario, if the modulation format is \(M\)-ary PSK, the sizes of the matrices \(G_s\), \(H^T_s\), and \((H^{-1}_s)^T\) are \(1 \times m\), \(m \times (m - 1)\), and \((m - 1) \times m\), respectively, where \(m = \log_2 M\) denotes the number of bits that constitute one symbol. At the transmitter, an input sequence is
encoded by the inverse syndrome to generate

\[ \tilde{s} = s(H^{-1}_s)^T. \] (3)

Since the inverse syndrome is an \((m - 1) \times m\) matrix, one bit redundancy is imposed here. Next, an arbitrary bit stream \(x\) is fed into \(G_s\), and its output \(v\) is modulo-2 added to \(\tilde{s}\) to form \(z = \tilde{s} + v\). Finally, \(z\) is mapped into complex symbol sequence \(S\) to transmit. Note that since the sequence \(x\) (or its codeword \(v\)) can be chosen arbitrary, we wish to choose them such that \(S\) turns into a waveform having low PAR after pulse-shaping filtering. An efficient search algorithm that achieves this goal for single-carrier PSK systems is proposed in [6]. Finally, the received symbol \(R\), which is a noisy version of \(S\), is processed to the decoder as shown in Fig. 3.

Note that in the 8-PSK example of Fig. 2, according to our previous work [6] the following set of matrices has a good peak power reduction capability:\(^1\):

\[
G_s = \begin{bmatrix} 1 & 1 + D^2 & 0 \end{bmatrix}, \quad H_s^T = \begin{bmatrix} 1 + D^2 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (H_s^{-1})^T = \begin{bmatrix} \frac{1}{1+D^2} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \] (4)

**B. Conventional Decoding of Trellis Shaping**

In the conventional trellis shaping framework [1], the input sequence \(s\) can be retrieved at the receiver by first demapping the received symbol sequence \(R\) to \(\hat{z}\) with hard decision and then performing convolution with \(H_s^T\), based on the fact that [1]

\[ zH_s^T = (\tilde{s} + v)H_s^T = (s(H_s^{-1})^T + v)H_s^T = s(H_s^{-1})^T H_s^T + xG_sH_s^T = sI + x0 = s. \] (5)

This indicates that if the received and demapped symbol \(\hat{z}\) in Fig. 3 is correct (which can be achieved by proper set-partition mapping with an assumption of high SNR channel), the information input \(s\) can be easily retrieved. This may be a desirable property for a well-designed coding and shaping system with parallel structure of Fig. 1 (b). However, in this decoding structure, the shaped bits that are separated from coded bits cannot enjoy the gain by joint (iterative) decoding.

---

\(^1\)The rightmost 0 of \(G_s\) implies that the second bit of \(s\) is invariant before and after shaping.
III. MARKOV MODEL FOR TRELIS SHAPED SYMBOLS

In the conventional single-carrier transmission, it is often assumed that the transmitted sequence $S$ follows independent and identical distribution (i.i.d.). In contrast, since the TS for peak power control restricts symbol transition patterns so as to generate a signal with low dynamic range, the successive symbols of $S = \{S_1, S_2, \ldots\}$ are highly correlated. The sequence of these correlated output symbols can be approximated by Markov process. We describe this model in this section.

Let us consider the baseband signal after pulse shaping filter at the transmitter expressed as

$$s(t) = \sum_{n=1}^{N} S_n g(t - n T_s),$$

(6)

where $N$ is the number of the transmitted symbols, $T_s$ is the Nyquist interval, and $g(t)$ is the impulse response of a pulse-shaping filter. The optimal output (or input $x$) in terms of PAR reduction is the one such that $s(t)$ of (6) has low dynamic range throughout its whole range of $t$. This means that the statistical property of the output symbols $S_n$ is characterized exactly only by the $N$-dimensional joint probability, which becomes prohibitive to compute as $N$ increases. To make this correlation property tractable, we model the shaping output symbols as an $m_m$th-order Markov process ($m_m \geq 1$). Specifically, upon applying the chain rule of the joint probability:

$$\Pr(S_1, \ldots, S_N) = \Pr(S_N|S_1, \ldots, S_{N-1}) \Pr(S_{N-1}|S_1, \ldots, S_{N-2}) \cdots,$$

(7)

we assume, for any $n > m_m$, that

$$\Pr(S_n|S_1, \ldots, S_{n-1}) = \Pr(S_n|S_{n-m_m}, \ldots, S_{n-1})$$

(8)

where

$$\Gamma_m^{(n)} \triangleq (S_{n-m_m+1}, S_{n-m_m+2}, \ldots, S_n),$$

(9)

is the state of the $m_m$th order Markov process for a given symbol sequence up to $S_n$. Note that the subscript of $m_m$ and $\Gamma_m^{(n)}$ refers to the fact that we invoke a Markov assumption of the generated symbol sequence.

We further assume the stationarity of the shaped symbols $S$ so that the transition probability of (8) does not depend on the particular symbol index $n$. Then, according to (8) we obtain

$$\Pr(S_1, \ldots, S_N) = \Pr(S_1, \ldots, S_{m_m}) \prod_{n=m_m+1}^{N} \Pr(S_n|\Gamma_m^{(n-1)}),$$

(10)
TABLE I
FIRST ORDER TRANSITION PROBABILITIES FROM $S^0$.

<table>
<thead>
<tr>
<th>$S^x$</th>
<th>$S^0$</th>
<th>$S^1$</th>
<th>$S^2$</th>
<th>$S^3$</th>
<th>$S^4$</th>
<th>$S^5$</th>
<th>$S^6$</th>
<th>$S^7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr($S^x</td>
<td>S^0$)</td>
<td>0.25</td>
<td>0.26</td>
<td>0.12</td>
<td>5.9 x 10^{-5}</td>
<td>0</td>
<td>5.9 x 10^{-5}</td>
<td>0.12</td>
</tr>
</tbody>
</table>

where $Pr(S_1, \ldots, S_{m_m}) \triangleq Pr(\Gamma_{m_m})$ represents the $m_m$-dimensional joint probability.

For a given assumed order $m_m$ with specified trellis shaping parameters and metric, these joint probabilities and their associated conditional (transition) probabilities can be easily obtained by computer simulation \textit{a priori}. As an example, typical symbol transition probabilities based on the first order approximation (i.e., $m_m = 1$), obtained from the shaping system proposed in [6], are listed in Table I.

IV. PERFORMANCE ANALYSIS

In this section, we analyze the performance of the trellis shaping system. We start by deriving BER expressions for uncoded and coded system without exploiting shaping for error correction. Then we derive the expressions for the AMI of the proposed concatenated system and calculate them by Monte-Carlo method.

A. Performance of Conventional Decoding without Channel Coding

First, we derive the BER of conventional decoding of the system in Fig. 2 without channel coding. Extension of analysis to general shaping systems is straightforward. The syndrome former in (4) implies the following relationship between $s_n$ and $z_n = (z_{0,n} \ z_{1,n} \ z_{2,n})$:

$$s_n = \begin{bmatrix} s_{0,n} \\ s_{1,n} \end{bmatrix} = z_n H_s^T = \begin{bmatrix} z_{0,n} + z_{0,n-2} + z_{1,n} \\ z_{2,n} \end{bmatrix},$$  \hspace{1cm} (11)

where the subscript $n$ denotes the index of the transmission symbol sequence. Assuming Gray mapping at the transmitter and hard decision at the receiver, each of demapped bits $\hat{z}_{i,n}, (i = 0, 1, 2)$ has an identical bit error probability, denoted by $p_z$. Since $\hat{s}_{1,n} = \hat{z}_{2,n}$, the bit error probability of $\hat{s}_{1,n}$ is equal to $p_z$. On the other hand, that of $\hat{s}_{0,n}$ can be calculated as

$$\binom{3}{1} p_z (1 - p_z)^2 + \binom{3}{3} p_z^3 = 3p_z - 6p_z^2 + 4p_z^3.$$  \hspace{1cm} (12)
Thus the average bit error probability of $s_n$, denoted by $p_s$, is given by

$$p_s = \frac{1}{2} \left( 3p_z - 6p_z^2 + 4p_z^3 + p_z \right) = 2p_z - 3p_z^2 + 2p_z^3.$$  \hspace{1cm} (13)

Note that, for high SNR, the error probability $p_s$ is dominated by $2p_z$. In practice, the shaping matrices should be determined such that $p_s$ is as small as possible. In this perspective, the use of shaping matrices with short constraint length is preferable. However, the use of shaping matrices with a larger constraint length enhances shaping capability (i.e. peak power control capability) as observed in [6]. Therefore, in the framework of conventional decoding, the constraint length of the shaping matrix offers a trade off between the peak power control capability and BER performance.

In the case of an additive white Gaussian noise (AWGN) channel and Gray-mapped 8PSK modulation, $p_z$ is given by

$$p_z = \frac{2}{3} Q \left( \sqrt{2E_b/N_0 \sin \left( \frac{\pi}{8} \right)} \right).$$ \hspace{1cm} (14)

The signal-to-noise ratio per information bits $E_b/N_0$ can be expressed as

$$E_b/N_0 = (\log_2 M - \eta) \frac{E_s}{N_0},$$  \hspace{1cm} (15)

where $E_s$ denotes the energy of PSK symbol, $N_0$ is a single-sided power spectral density of the AWGN process, and $\eta$ denotes the number of the redundant bits imposed by the TS (which is equal to 1 in our example).

B. Performance of Soft-Decision Decoding with Channel Coding

As a reference, we also analyze the BER of the serially concatenated system without exploiting the shaping constraint. It is assumed that the TS demapper passes the soft decision to the channel decoder. In this case, based on the transfer function technique and Chernoff bound [15], the union bound of the BER is calculated as

$$P_{TS, CC} < \frac{\partial T(D, B)}{\partial B} \bigg|_{B=1, D=\frac{1}{2}p_z},$$ \hspace{1cm} (16)

where $T(D, B)$ is a transfer function of a given convolutional codes.
C. Achievable Performance of Joint Decoding of Coding and Shaping

Capacity analysis of peak power-limited channel has been found in [16, 17] as well as in the original work of Shannon [18]. However, none of these studies takes into account the envelope fluctuation of band-limited waveforms and thus cannot be applied to the analysis of our peak power controlling shaping systems.

Recall that the TS embeds correlation (or memory) in the modulated symbols as a consequence of peak power control. For a given transmission sequence \( S \) and its received noisy version \( R = S + W \), where \( W \) is a corresponding additive noise process, the performance limit is characterized by its AMI, denoted by \( I(S; R) \). Since \( S \) has a memory, \( I(S; R) \) should take into account this memory effect. Let us express \( I(S; R) \) in terms of entropies as:

\[
I(S; R) = H(R) - H(R|S)
\]

(17)

\[
= H(R) - H(W).
\]

(18)

Throughout this paper, \( W \) is assumed to be an AWGN with the variance equal to \( \sigma^2 = N_0/2 \). The entropy is then given by [15]

\[
H(W) = \log_2(\pi e N_0).
\]

(19)

On the other hand, the entropy of the stationary sequence \( R \) is expressed as

\[
H(R) = - \lim_{n \to \infty} \frac{1}{n} E[\log_2 p(R_1, \cdots, R_n)]
\]

(20)

where \( p(X_1, \cdots, X_n) \) denotes the \( n \)-dimensional joint probability density function (PDF) associated with the random variables from \( X_1 \) to \( X_n \). The entropy (20) can be numerically calculated by means of the forward-recursion computation procedure similar to the BCJR algorithm [19, 20]. Specifically, let us rewrite \( p(R_1, \cdots, R_n) \) as a summation of joint PDFs with a state variable:

\[
p(R_1, \cdots, R_n) = \sum_{\text{all possible } \Gamma^{(n)}} p(R_1, \cdots, R_n, \Gamma^{(n)}),
\]

(21)

where the state \( \Gamma^{(n)} \) is defined as a combination of all \( n \) output symbols. If we express \( \alpha_n(\Gamma^{(n)}) \triangleq p(R_1, \cdots, R_n, \Gamma^{(n)}) \), then it satisfies the following recursive equation:

\[
\alpha_n(\Gamma^{(n)}) = \sum_{\text{all possible } \Gamma^{(n-1)}} p(R_n|\Gamma^{(n)}, \Gamma^{(n-1)}) \text{Pr}(\Gamma^{(n)}|\Gamma^{(n-1)}) \alpha_{n-1}(\Gamma^{(n-1)})
\]

\[
= \frac{1}{\pi N_0} \sum_{\text{all possible } \Gamma^{(n-1)}} \exp\left(-\frac{|R_n - S_n|^2}{N_0}\right) \Pr(S_n|\Gamma^{(n-1)}) \alpha_{n-1}(\Gamma^{(n-1)}),
\]

(22)
where we have used the fact that a transition from $\Gamma^{(n-1)}$ to $\Gamma^{(n)}$ is identical to the observation of the corresponding one symbol $S_n$. In general, calculation of (20) is prohibitively complex as the number of states increases exponentially with that of the observation symbols $n$. Therefore, we resort to the Markov model assumption in Section III. Let $\phi(S; m_m)$ denote the sequence of $S$ approximated by the Markov process with the order $m_m$. Since $R$ and $\phi(S; m_m)$ are both functions of $S$, the data processing theorem [21] states that

$$I(S; R) \leq I(\phi(S; m_m); R) \triangleq I_{m_m}.$$ \hspace{1cm} (23)

Therefore, the AMI obtained based on the Markov assumption serves as an upper bound for the exact AMI. Furthermore, since the gap between $\phi(S; m_m)$ and $S$ reduces as the order $m_m$ increases, one may obtain arbitrarily close approximation of $I(S; R)$. In this case, the calculation of (22) can be done by simply replacing $\Gamma^{(n)}$ with the state $\Gamma_m^{(n)}$ defined in (9).

The state and branch of the corresponding trellis for calculating $I_{m_m}$ can be defined as follows.

- **State**: a state holds a possible combination of a set of the recent $m_m$ symbols as given by (9). Since there exist several symbol patterns whose associated stationary probabilities are zero, i.e., symbol patterns that are never encountered, the number of states is not necessarily equal to $M^{m_m}$.

- **Branch**: every state has at most $M$ branches associated with one symbol observation. Each branch also has transition probability $\Pr(S_n|\Gamma_m^{(n-1)})$ as a weight of this branch. This does not depend on particular time index $n$ due to the stationarity of $S$, i.e., $\Pr(S_n|\Gamma_m^{(n-1)}) = \Pr(S_{n-1}|\Gamma_m^{(n-2)})$ holds for any $n$. If the weight is zero, then one can prune this branch. Thus the number of branches is less than or equal to $M$.

It is interesting to note that another simulation-based approach for estimating the AMI appears in [22, 23]. The method in this paper decomposes the mutual information into (18) and the forward-recursion is applied to calculate $H(R)$. On the other hand, the approach of [22, 23] decomposes the AMI as

$$I(S; R) = H(S) - H(S|R),$$ \hspace{1cm} (24)

and BCJR algorithm calculates $H(S|R)$. These two computation methods essentially lead to the same result, but they treat the memory in different ways. Specifically, the former implies that the memory lies in source and the channel is memoryless. On the other hand, the latter renders the
memory be a part of channel and thus the source can be considered to be i.i.d. In this paper, we take the former approach, as the memory induced by the TS can be assumed deterministic and need not be estimated, unlike other dynamic channels such as those with the ISI. Furthermore, they are also different from the viewpoint of computation. The former makes the computation of $H(R|S) = H(W)$ easy, due to the memoryless channel assumed. Conversely, the latter makes the computation of $H(S)$ easy due to the i.i.d. source.

D. Achievable Performance of Disjointed Decoding

In the serial concatenation framework of shaping and coding, we also consider the achievable performance when we do not take the correlation of the shaping output into account for simple implementation. In what follows, we refer to this decoding approach as a disjointed decoding. In this case, the shaping redundancy of $\eta$ bit is used exclusively for non-error-correcting purpose. Thus, similar to the binary case found in [7], the AMI in this case is upper-bounded by

$$I_{\text{dis}} \leq \max \left( \sup_{S} I(S;R) - \eta, 0 \right)$$

(25)

where the max operation compensates for the fact that the AMI cannot be negative. Since $I(S;R)$ achieves maximum value when $S$ follows i.i.d. for a given $M$-ary input constellation, the above upper bound for $M$-PSK symbol can be expressed as

$$I_{\text{dis}} \leq \max \left( C_{M\text{-PSK}} - \eta, 0 \right),$$

(26)

where $C_{M\text{-PSK}}$ is the AMI of the $M$-PSK symbol over AWGN with i.i.d. input.

To avoid confusion, we distinguish the AMI considering symbol correlation, (18), from $I_{\text{dis}}$ by explicitly referring to the former as a joint AMI in what follows.

E. Numerical Evaluation of AMI

By Monte-Carlo simulation, we have numerically evaluated the joint AMI upper bound $I_{mm}$ of (23) using (20)-(22) and the disjointed AMI upper bound (26). The symbol transition probabilities depend on the specific shaping metrics. In the following numerical example, we have adopted the moment method of [6] with three external memories. Upon generating the signal waveform, a square-root raised-cosine filter with roll-off factor 0.4 is used for pulse shaping and the length of effective (non-zero) impulse response is chosen to be 10 symbols. The shaping matrices are
given by (4). Upon calculating (20), we have set the length of the symbols 5000. The AMIs are first computed as a function of $E_s/N_0$ where $E_s$ is energy per transmitted symbol, and then converted into a function of $E_b/N_0$ by dividing $E_s/N_0$ by the corresponding AMI value. The results are plotted in Fig. 4.

1) Joint AMI: First, let us examine the joint AMI curves. As we have seen, the approximation of the joint AMI of the underlying TS system can be made accurate by increasing the order of Markov model $m_m$. In Fig. 4, the AMI curves using Markov model with the orders up to $m_m = 4$ are plotted. From this figure, we observe that as $m_m$ increases the gap becomes smaller with a convergent behavior especially when they are measured around low $E_b/N_0$ region. Therefore, in the vicinity of the AMI around 1 bit per symbol, which is the rate of the examined system, the Markov model with $m_m = 4$ may be expected to serve as a practically accurate estimate of

---

Fig. 4. Joint AMIs with respect to various Markov orders $m_m$ and a disjointed AMI.
the actual AMI.

2) **Disjointed AMI:** The behavior of the disjointed AMI suggests several interesting issues from the viewpoint of coding system design. From Fig. 4, it is observed that unlike usual AMI characteristics where the required $E_b/N_0$ monotonically decreases as the achievable information rate decreases, the disjointed design tends to increase the required $E_b/N_0$ as the information rate becomes below a certain level. (It may be interesting to mention that in [15, 24] similar behavior is reported in the noncoherent transmission schemes.) In addition to this unusual behavior, there is a degradation in the required $E_b/N_0$ of more than 6 dB compared to that of the i.i.d. QPSK system, even when they are evaluated at the point with the minimum $E_b/N_0$ gap (around the information rate of 0.75).

3) **Practical Remarks:** When evaluated at the information rate of 1 bit per symbol, the gap of the required $E_b/N_0$ between i.i.d. QPSK and the joint AMI is about 1.5 dB, revealing that there is an unrecoverable loss of $E_b/N_0$ due to the use of the TS. On the other hand, in [6], we have observed that the TS with the aforementioned parameters is able to reduce the PAR by about 1.8 dB over i.i.d. QPSK system. If the PAR and $E_b/N_0$ is compared in the same framework, the overall gain is only 0.3 dB. However, in practice we should take into account the fact that the reduced PAR in turn results in substantial enhancement of the PAE of the PA [25], and the benefit of this improvement is not directly comparable with the loss in terms of the required $E_b/N_0$. In the scenario of a typical practical high transmission power (i.e., long distance) communication system, the power consumption of the PA saved by the PAR reduction of 1.8 dB is beneficial even at the price of the increase of the average transmit power of 1.5 dB so as to compensate for the resulting $E_b/N_0$ loss.

The question is how to develop the system that can approach the performance limit suggested by the joint AMI. In the next section, we describe a joint decoding strategy that can achieve this goal.

V. **A New Decoder for Trellis Shaping**

The previous section has demonstrated the benefit of the joint design of coding and shaping by exploiting the memory effect embedded in the modulated symbols. In this section, we extend the conventional decoder of TS in Section II-B so as to adopt the Markov property efficiently.
Furthermore, the new decoder has a form of soft-input soft-output (SISO) decoding, which can thus serve as a component code of a so-called turbo system.

A. Soft-Input Soft-Output (SISO) Decoding of Trellis Shaping

Let us consider shaping and modulation as one entity, similar to that of coded modulation. The relation of the input bits and output symbols can be expressed in a form of mapping function \( \tau(\cdot) \) as (see Fig. 3)

\[
S = \tau(s, x).
\]  
(27)

On the other hand, since the above mapping is one to many in terms of \( s \rightarrow S \), the inverse conversion can be defined as

\[
s = \tau^{-1}(S).
\]  
(28)

We observe that the forward conversion (27) does not have a specific trellis structure in terms of input/output relation (i.e., \( s \rightarrow S \)). Conversely, the inverse conversion (28) has a trellis structure with a finite number of states, since \( H_s^T \) is a convolutional code in a non-recursive form. Let us rewrite the expression of \( \tau^{-1}(\cdot) \) element-wise as

\[
s_n = \tau^{-1}(S_n-m_s, \ldots, S_n),
\]  
(29)

where \( m_s \) denotes the number of memories in \( H_s^T \), and \( s_n \) and \( S_n \) are the \( n \)th element of \( s \) and \( S \), respectively. Since we consider rate \( 2/3 \) TS, \( s_n \) is a binary vector with length 2 and given by \( \begin{pmatrix} s_{0,n} & s_{1,n} \end{pmatrix}^T \). The trellis that represents (29) is characterized by the following state and branch.

- **State**: a state at the \( n \)th symbol index holds a set (repeated permutation) of the previous \( m_s \) symbols

\[
\Gamma_s^{(n)} \triangleq (S_{n-m_s+1}, S_{n-m_s+2}, \ldots, S_n),
\]  
(30)

each of which takes a symbol from the signal constellation \( S \). Thus the number of state is \( M^{m_s} \).

- **Branch**: every state has \( M \) branches, each of which represents one symbol observation. A branch connecting \( \Gamma_s^{(n-1)} \) and \( \Gamma_s^{(n)} \) is also associated with the binary output \( s_n = \tau^{-1}(\Gamma_s^{(n-1)}, S_n) \).
(Note that the subscript of \( m_s \) and \( \Gamma_s^{(n)} \) emphasizes that these memory and states are of the shaping decoder.)

Recall that, in addition to the above trellis structure, \( S \) by itself has a trellis structure due to the peak power control process at the transmitter as discussed in Section IV-C. To effectively exploit the memory effect of the shaped symbols, we merge these two trellises into a single trellis that can be used for the proposed decoding as illustrated in Fig. 5. In this figure, \( m_t \) denotes the size of the symbol array stored in each state, which is chosen as

\[
m_t = \max(m_s, m_m).
\]

(31)

Note that the order of Markov process, \( m_m \), which captures the memory effect of the symbol outputs, can be chosen independent of \( m_s \). As will be shown later, increasing \( m_m \) yields better decoding performance, but it also causes exponential increase of the trellis states. There is thus a trade off between decoding complexity and performance gain. Nevertheless, in practice, it is always beneficial to choose \( m_m \geq m_s \) (and thus \( m_t \geq m_s \)), since at least \( M^{m_s} \) states are necessary for successful decoding of \( s_n \). (In Fig.5, we have shown the merged trellis with \( m_s > m_m \) just for illustration purpose.)

The proposed decoding algorithm is an application of BCJR algorithm [26] to the trellis of Fig. 5. To describe the BCJR algorithm, it may be sufficient to describe the definitions of \( \alpha, \beta, \) and \( \gamma \) [26]. Note that some of the notations below coincides with Section IV-C.

Associated with each state of the merged trellis

\[
\Gamma^{(n)}_t \triangleq (S_{n-m_t+1}, S_{n-m_t+2}, \ldots, S_n),
\]

(32)

the branch metric, i.e., the state transition probability \( \gamma \) at the \( n \)th symbol index where \( n = 1, 2, \ldots, N \) is defined as

\[
\gamma(\Gamma^{(n)}_t, \Gamma^{(n-1)}_t) \triangleq \Pr \left( R_n, s_n, S_n | \Gamma^{(n-1)}_t \right) \\
= \Pr(R_n | S_n) \cdot \Pr(s_n = \tau^{-1}(\Gamma^{(n-1)}_s, S_n)) \cdot \Pr(S_n | \Gamma^{(n-1)}_m),
\]

(33)

where

- the channel transition term \( P_c \) denotes a channel transition probability and given by in the case of AWGN

\[
\Pr(R|S) = C \exp \left( -\frac{|R - S|^2}{N_0} \right),
\]

(34)
with \( C \) a normalization constant,

- the a priori information term \( P_a \) is the a priori probability of the input bit \( s \), which can incorporate the soft outputs from the outer code for iterative decoding, and
- the transition probability term \( P_t \) denotes the symbol transition probability based on the Markov model approximation of \( S \).

The forward and backward state probabilities, \( \alpha \) and \( \beta \), are recursively calculated by

\[
\alpha(\Gamma_t^{(n)}) \triangleq \Pr(\Gamma_t^{(n)}|R_1, \cdots, R_n) \\
= \sum_{\text{all possible } \Gamma_t^{(n-1)}} \gamma(\Gamma_t^{(n)}, \Gamma_t^{(n-1)}) \alpha(\Gamma_t^{(n-1)}),
\]

\[
\beta(\Gamma_t^{(n)}) \triangleq \Pr(R_1, \cdots, R_n|\Gamma_t^{(n)}) \\
= \sum_{\text{all possible } \Gamma_t^{(n)}} \gamma(\Gamma_t^{(n+1)}, \Gamma_t^{(n)}) \beta(\Gamma_t^{(n+1)}),
\]

(35)
To compute $\alpha$ and $\beta$ based on (35), their end values $\alpha(\Gamma_t^{(m_t)})$, $\beta(\Gamma_t^{(N-m_t)})$, where $N$ is the length of $S$, are necessary. However, unlike usual convolutional codes, the initial and final states are unknown in the scenario of the TS. It is thus appropriate to assign them the stationary probabilities with respect to each state. Finally, the following formula gives a posterior probabilities for each input bit $s_{i,n}$:

$$
\Pr(s_{i,n} = x | R) = \sum_{\Gamma_t^{(n)}, \Gamma_t^{(n-1)}} \alpha(\Gamma_t^{(n-1)}) \gamma(\Gamma_t^{(n)}, \Gamma_t^{(n-1)}) \beta(\Gamma_t^{(n)}).
$$

(36)

**B. Iterative Decoding of Trellis Shaping**

The proposed decoding algorithm is an instance of the soft-input soft-output (SISO) decoder, as it can take into account the a priori information with respect to the binary shaping inputs $s$. Thus, the system depicted in Fig. 2 is capable of applying iterative decoding as shown in Fig. 6. We describe the iterative algorithm by log-likelihood ratio (LLR) [27] to simplify the interchange of information between SISO components, as in the turbo decoding literature.

The a priori and a posterior LLRs in Fig. 6 are respectively defined as

$$
L_a(\xi) \triangleq \log \frac{\Pr(\xi = 0)}{\Pr(\xi = 1)},
$$

(37)

$$
L_d(\xi) \triangleq \log \frac{\Pr(\xi = 0 | y)}{\Pr(\xi = 1 | y)},
$$

(38)

where $\xi$ can be either input or coded bit and $y$ be either the detected symbols $R$ or a priori LLRs of coded bits $c$. In what follows, if the argument is a bold letter then they represent an associated sequence of LLRs. For trellis codes, the BCJR algorithm calculates the a posterior LLR from a given a priori LLR. The extrinsic LLR, which serves as the a priori LLR to the other SISO decoders, is given by

$$
L_e(\xi) = L_d(\xi) - L_a(\xi).
$$

(39)

Prior to the BCJR decoding, a SISO decoder transforms a priori LLR into the probability domain as

$$
\Pr(\xi = 0) = \frac{e^{L_a(\xi)}}{1 + e^{L_a(\xi)}}, \quad \Pr(\xi = 1) = \frac{1}{1 + e^{-L_a(\xi)}}.
$$

(40)

To the contrary, after the completion of the BCJR algorithm, the a posterior probabilities are transformed to LLR based on (36) and (38).
First, the TS SISO decoder (based on the trellis described in the previous subsection) calculates the \textit{a posterior} LLR \( L_d(s) \) from the detected symbols \( R \). It is then deinterleaved to \( L_a(c) \), which in turn serves as the \textit{a priori} LLR for the SISO decoder of the outer channel coding. At the initial stage, the \textit{extrinsic} LLR \( L_e(s) \) is identical to \( L_d(s) \) since the \textit{a priori} LLR is absent. Next, the SISO decoder of the outer channel coding calculates two kinds of \textit{a posterior} LLRs, \( L_d(c) \) and \( L_d(b) \), with respect to the codeword \( c \) and the information \( b \), respectively. The SISO decoding of the outer code is similar to those given in [9, 28]. For the next iteration, \( L_e(c) \) is subtracted from \( L_d(c) \), and becomes the \textit{extrinsic} LLR \( L_a(c) \). The interleaved version of this LLR, \( L_a(s) \), serves as the \textit{a priori} LLR to the TS SISO decoder. At the second iteration or later, \( L_a(s) \) is fed to the TS SISO decoder. After an arbitrary number of iterations, decision is made based on \( L_d(b) \) to determine the hypothesis of the transmitted information \( \hat{b} \).

**VI. Simulation Results**

In this section, we evaluate the BER of the proposed system by simulation and compare it with the analytical bounds derived in Section IV. In the following simulations, the optimal convolutional code (CC) within the class of constraint length \( K = 7 \) given in [15] is used for the outer channel code in Fig. 2. The S-random interleaver [29] (with the parameter \( S = 10 \)) is used for the pseudo random interleaver \( \Pi \). The orders of Markov model used for decoding of...
shaped symbols are chosen from $m_m = 1, 2, \text{ and } 4$. As for the transmitter side, the same peak power control TS system as that described in Section IV-E is employed.

A. Performance without Markov Model

We first evaluate the performance of the TS system without channel coding. Fig. 7 shows the simulation results, along with the theoretical curve from (13) and (14). It is observed that the theoretical curve closely matches the simulation result.

In the case of the convolutional coding, its transfer function is given by

$$T(D, B) = D^{10}B^{36} + D^{12}B^{211} + D^{14}B^{1404} + D^{16}B^{11633} + D^{18}B^{177433} + \cdots$$

(41)

from which the union bound can be calculated with (16). The simulation result in Fig. 7 also matches the theoretical bound. Note that in this simulation, we have not taken into account the shaping symbol correlation, but the TS decoder is still able to pass the soft decision output to the outer Viterbi decoder by assigning the equiprobable transition probabilities (i.e., $m_m = 0$).

Referring to the disjointed AMI bound of Fig. 4 at the information rate of 1 bit per symbol, it is observed that the required minimum $E_b/N_0$ is around 5.7 dB if the decoder does not make any use of knowledge about the Markov nature of the shaped symbols. To verify this limit, we have investigated a serial concatenation of TS and a turbo code [8] (the outer convolutional code in Fig. 2 is now replaced by a turbo code). This turbo code is based on the original rate-1/3 turbo code [8] and punctured so as to match the coding rate $r_c = 1/2$. Hence, the overall rate $r$ is the same as that of the baseline system (1 bit per symbol). We have set the interleaver length to 10,000 and the constraint length of component codes to 5 such that it can approach within 1 dB from the channel capacity for BPSK.

It is observed from Fig. 7 that the BER curve and the disjointed AMI limit well agree, which may justify the argument presented in Section IV-D. This result also suggests the importance of the use of Markov model at the receiver; even with a near optimal turbo code, the system cannot reduce the required $E_b/N_0$ below 5.7 dB in this scenario.

B. Performance with Markov Model and Iterative Decoding

Fig. 8 shows the BER using Markov model with and without channel coding. Various cases are plotted for joint decoding. As observed, the coding gain achieved by the TS only is insignificant,
even with the order as high as $m_m = 4$. Nevertheless, the serial concatenation in conjunction with the iterative decoding achieves significant coding gain. The result shows that the BER performance is improved gradually but steadily, with increasing $m_m$, the number of iteration, and interleaver length. Also, it is remarkable that the combination of $m_m = 4$, 20 iterations, and a large interleaver exhibits so-called water fall behavior and coding gain of 10 dB compared to the conventional decoding of TS. Furthermore, the gap between the simulated BER and the joint AMI limit is within 1.5 dB in this case.

In the case of $m_m = 4$, there is no error floor observed at least in the simulated BER region, whereas the BER curves associated with $m_m = 1$ and 2 have a noticeable error floor. This error floor may be alleviated by a carefully designed interleaver. However, the beginning of water fall region between $m_m = 2$ and $m_m = 4$ is almost identical. Thus, considering the number of
Fig. 8. Bit error rate with Markov model and iterative processes. The attached numbers in parenthesis indicate 1) the order of Markov model $m_m$, 2) the maximum number of iteration, and 3) the interleaver length, from left to right.

required trellis states in the case of $m_m = 4$, we conclude that $m_m = 2$ (and even $m_m = 1$) is advantageous to $m_m = 4$ from the viewpoint of decoding complexity.

In this simulation, we examined the case with $m_m = 1$ for the purpose of demonstrating the effect of the Markov model on the resulting BER. It is interesting to note the simulated system with that $m_m = 2$ virtually outperforms that with $m_m = 1$ in terms of complexity as well as BER: In addition to the fact that in these cases $m_t$ is fixed to 2 by the syndrome memory constraint of $m_s = 2$, prunable states also increase as $m_m$ increases. Hence, $m_m = 2$ has eventually lower complexity than $m_m = 1$, and yet achieves better BER performance. For example, in this shaping setup, the number of states (after eliminating the prunable states) is 64, 56, and 1079 for $m_m = 1, 2,$ and 4, respectively.
VII. Conclusion

In this paper, we have proposed a new approach for the trellis shaping designed for peak power control. It has been shown by both theoretical analysis and computer simulation that the serial concatenation structure of coding and shaping, together with the Markov model approximation of the shaped symbols and iterative decoding, can achieve a substantial coding gain. Thus, one can enjoy the large amount of peak power reduction of TS without noticeable degradation of BER performance.

REFERENCES


