Iterative Decoding of Concatenated Channel Coding and Trellis Shaping Based on Markov Model

Makoto Tanahashi
Department of Electrical and Computer Engineering
Yokohama National University
Yokohama, Kanagawa 240-8501, Japan
Email: makoto@ochialab.dnj.ynu.ac.jp

Hideki Ochiai
Department of Electrical and Computer Engineering
Yokohama National University
Yokohama, Kanagawa 240-8501, Japan
Email: hideki@ynu.ac.jp

Abstract—The trellis shaping (TS) is an effective technique not only for average power reduction of high-order QAM signals, but also for peak power reduction of band-limited single-carrier PSK signals. Our recent work has demonstrated that signals with almost constant envelope can be generated. Compared to the unshaped PSK system, however, this shaping process introduces signal constellation expansion exclusively for peak power reduction and thus reduces the minimum Euclidean distance of the PSK signal constellation, rendering the overall system sensitive to channel noise. To overcome this drawback, this paper proposes a new decoding algorithm of the TS that exploits the Markov process property of the shaped symbols. This decoding algorithm can work alone, but becomes more effective when the shaping encoder is serially concatenated with a channel coding. The simulation results demonstrate that this serial concatenation together with an iterative decoding between shaping and coding can achieve a significant amount of coding gain.

I. INTRODUCTION

Reducing peak power of communication signals has been one of the major issues in communication system engineering. In particular, linear amplification of wideband signals with a large dynamic range is still challenging, since the power amplifier (PA) should be operated with a large back-off. This operation substantially reduces DC-RF conversion efficiency of the PA, which is a critical bottleneck for battery driven communication terminals.

The trellis shaping (TS), originally proposed by Forney [1] for average power reduction of high-order QAM, has found its applications in peak power reduction of communication systems. Recently, the authors have proposed a newly designed shaping metric for single-carrier systems and demonstrated that the TS can control the envelope of the band-limited PSK signals with almost constant level [2]. An example waveform of this trellis shaping is depicted in Fig. 1.

In this paper, to overcome this SNR penalty, we propose a new decoding algorithm for the peak power controlling TS. The underlying principle for the proposed decoding is to make use of the Markov nature of the symbol transitions caused by the TS (similar to the constraint gain in [3]). More specifically, the peak power reduction process is equivalent to constraining the successive phase shifts so as not to generate envelope with large fluctuation and, as a result, the symbol transition with small phase difference occurs very frequently, whereas antipodal transitions would never occur as observed in [2]. By modeling this constraining process as a Markov chain and formulating a trellis structure, we apply BCJR algorithm [4] to decoding of this shaping process. The proposed decoder can work alone, but becomes much more effective when it is serially concatenated with a channel coding as sketched in Fig. 2. In fact, Fig. 2 can be seen as an instance of serially concatenated convolutional codes [5]. It is demonstrated that the proposed serial concatenation of trellis shaping and channel coding, along with the well designed iterative decoding, can achieve significant performance gain that can even approach the coded performance without shaping.

In order to justify the significant coding gain observed in the simulation results, we also analyze the achievable information rate based on average mutual information (AMI) with a given TS constraint. It will be shown that the simulation results are close to the achievable rate obtained by the AMI, suggesting that the proposed decoding approach is nearly optimal.
II. System Description

Throughout the paper, for simplicity, we focus on the specific 8-PSK trellis shaping system concatenated with an outer channel coding of 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 interleaver II permutes \( c \) to \( s \), which is then input to the TS encoder. The TS encoder transforms \( s \) into \( z \) with adding one bit redundancy such that \( z \) generates a low-peak waveform when modulated and pulse-shaped. In this case, 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 \( S = \{S_i | 0 \leq i < 8 \} \) (see Fig. 2). Since each symbol carries 3 bits, the transmission rate in this case is one information bit per symbol. The next subsection describes the principle of the TS and its conventional decoding method.

A. Trellis Shaping

We restrict our attention to the TS design described in [2]. The combination of this TS and 8-PSK achieves almost constant envelope as observed in Fig. 1, and transition probabilities between successive symbols become nonuniform. An example case is shown in Table I.

We briefly describe the principle and notation of the TS. (The details can be found in [1,6].) First, we need three kinds of convolutional encoders: shaping encoder \( G_s \), its syndrome former \( H^{-1}_s \), and its left inverse \( (H^{-1}_s)^T \) (called inverse syndrome). They must satisfy \( G_s H_s^T = 0 \) and \( (H^{-1}_s)^T H_s^T = I \). In addition, these matrices should be designed such that \( H_s \) has no recursive term. Otherwise catastrophic error propagation will occur at the receiver [1,6].

In our TS framework, the above matrices are arranged as sketched in Fig. 3. If the modulation format is \( M \)-ary PSK, then \( G_s \) is \( 1 \times m \), \( H_s^T \) is \( m \times (m - 1) \), and \( (H^{-1}_s)^T \) is \( (m - 1) \times m \), where \( m = \log_2 M \) denotes the number of bits that constitute one symbol. In the case of 8-PSK modulation \((m = 3)\), we use the following matrices set:\(^1\)

\[
G_s = \begin{bmatrix} 1 & 1 + D^2 & 0 \end{bmatrix} \tag{1}
\]

\[
H_s^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2}
\]

\[
(H^{-1}_s)^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3}
\]

At the transmitter, an input sequence is fed into the inverse syndrome to generate \( \hat{s} = s (H^{-1}_s)^T \). 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 \( \hat{s} \) to form \( z = \hat{s} + v \). Finally, \( z \) is mapped into complex symbol sequences \( S \) to transmit. Note that \( x \) or its codeword \( v \) can be chosen arbitrary, and therefore it should be determined such that \( S \) turns into waveforms with low peak power after pulse-shaping filtering. An efficient search algorithm of \( x \) for single-carrier PSK system is given in [2].

To retrieve the input sequence \( s \) at the receiver, the received symbol sequence \( R \) is demapped to \( \hat{z} \) (the estimate of \( z \)), and the convolution with \( H_s^T \) is performed. Then the following equation is satisfied as long as the channel noise is negligible, i.e., the estimation of \( z \) is perfect.

\[
z H_s^T = (\hat{s} + v) H_s^T = (s (H^{-1}_s)^T + v) H_s^T = s (H^{-1}_s)^T H_s^T + x G_s H_s^T = s 1 + x 0 = s. \tag{4}
\]

Note that this decoding, though simple to implement, does not generate soft decision output. Soft output is necessary for the concatenated coding and shaping system to exert its full error correcting capability. Therefore, we propose a new decoding algorithm which is capable of passing a soft output for this system in the subsequent section.

III. Proposed Decoding Algorithm

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

Let us consider shaping and modulation as one entity, similar to the concept of TCM. Then the input/output relation can be expressed as (see Fig. 3)

\[
S = \tau(s, x). \tag{5}
\]

\(^1\)The rightmost 0 of \( G_s \) implies that the second bit of \( s \) is invariant before and after shaping.
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

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

We observe that the forward conversion (5) does not have a specific trellis structure in terms of input/output relation (i.e., \( s \rightarrow S \)). Conversely, the inverse conversion (6) has a trellis structure with a finite number of states, since \( H_s^T \) is a convolutional code in non-recursive form. Let us rewrite the expression of \( \tau^{-1}(\cdot) \) elementwise as

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

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, the \( s_n \) is a binary vector with length 2 and given by \( (s_{0,n}^T, s_{1,n}^T) \). The trellis that represents (7) is characterized by the following state and branch.

- **state**: a state holds a set (repeated permutation) of \( m_s \) symbols
  \[
  (S^{x(i)}, S^{x(i+1)}, \ldots, S^{x(i+m_s-1)})
  \]
  where \( 0 \leq x(i) < M, \)

  \[
  (8)
  \]
  each of which takes from signal constellation \( S \). Thus the number of states is \( M^{m_s} \).

- **branch**: every state has \( M \) branches, each of which represents one symbol observation. A branch is also associated with the output value \( s_n \) as a result of the corresponding state transition.

Recall that, due to the peak power reduction shaping, \( S \) by itself has a trellis structure that represents its Markov property. If we model \( S \) as a Markov process with the \( m_m \)-th order, then this trellis has:

- **state**: a state holds a possible combination of a set of the past \( m_m \) symbols similar to (8). It should be noted that the number of states is not necessarily equal to \( M^{m_m} \) since there exist several symbol transition patterns that never occur (i.e., stationary probability is zero).

- **branch**: every state has at most \( M \) branches associated with one symbol observation. Each branch also has associated transition probability as a weight of this branch. If the weight is zero, then one can prune this branch. Thus the number of branches is not always equal to \( M \).

Merging the above two trellises, we obtain the proposed decoding trellis, which is illustrated in Fig. 4. In this figure, \( m_t \) denotes the length of symbol array stored in the state, and chosen to be

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

![Fig. 4. An example trellis section of the proposed TS decoder in the case of 8-PSK (\( M = 8 \)), \( m_m = 1, m_s = 2 \), and therefore \( m_t = 2 \).](image)

Note that the order of assumed Markov process \( m_m \) can be chosen independent of \( m_s \). It will be shown that increasing \( m_m \) yields better decoding performance, but it also causes exponential increase of trellis states. Hence there is a trade off between complexity and performance gain. Nevertheless, it is always beneficial to choose \( m_m \geq m_s \) (and thus \( m_t = m_m \)) in practice, since at least \( M^{m_m} \) trellis states are required for the proposed decoding process. Note that the shaping metric in [2] is designed based on the impulse response of pulse-shaping filter, and thus the order \( m_m \) must be comparable to the effective length of the impulse response, in order to fully capture the Markov property of shaped symbols.

The proposed decoding algorithm is an application of BCJR algorithm [4] to the trellis of Fig. 4. To describe the BCJR algorithm, it may be sufficient to describe the definition of \( \alpha \), \( \beta \), and \( \gamma \) in our TS scenario.

The branch metric, i.e., state transition probability \( \gamma \) represents observation of one symbol \( S^k \) for a given state

\[
\Gamma = (S^{x(i)}, S^{x(i+1)}, \ldots, S^{x(i+m_t-1)}
\]

at the \( n \)th trellis section and is defined as

\[
\gamma_n(k, \Gamma) = \Pr\left(R_{n+1}, \delta_{n+1}, S^k|\Gamma\right) = P_c(n) \cdot P_o(n, k|\Gamma) \cdot P_t(k|\Gamma),
\]

where
channel transition term $P_c(n) \triangleq \Pr(R_{n+1}|S_{n+1})$ denotes a channel transition probability. In the case of AWGN, it is given by

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

(11)

where $C$ is a normalization constant.

- \textit{a priori} information term $P_a(n,k|\Gamma) \triangleq \Pr(\hat{s}_n = \mathcal{L}^{-1}(S^k, S^{x(i)}, S^{x(i+1)}, \ldots, S^{x(i+m_m-1)}))$ is a priori probability about the input bit $s[n]$. This term accepts soft outputs from the outer code for iterative decoding.

- transition probability term $P_t(k|\Gamma) \triangleq \Pr(S^k|S^{x(i)}, S^{x(i+1)}, \ldots, S^{x(i+m_m-1)})$ denotes symbol transition probability based on Markov assumption of $S$.

The forward and backward state probabilities, $\alpha$ and $\beta$, are recursively calculated from $\gamma$. To compute the forward and backward state probabilities, $\alpha$ and $\beta$, their end values $\alpha_0(\Gamma), \beta_N(\Gamma)$ are necessary. However, unlike regular convolutional codes, the initial and final states are unknown in the scenario of TS. Thus, we assign them the stationary probabilities corresponding to each state. Finally, the following formula gives a posterior probabilities for each estimate of input bit $\hat{s}_i,n$:

$$Pr(\hat{s}_i,n|R) = \sum_{k=0}^{N-1} \alpha_{n-i}(\Gamma') \gamma_n(k, \Gamma') \beta_n(\Gamma),$$

(12)

where $\Gamma'$ is a departure state which merges into $\Gamma$ for the branch associated with $S^k$.

**B. Iterative Decoding of Trellis Shaping**

The proposed decoding algorithm is an instance of soft-input soft-output (SISO) decoder, as it can take into account the \textit{a priori} information about $s$. Thus, the system depicted in Fig. 2 is capable of applying iterative decoding. For space limitation, we omit details and summarize the iterative process in Fig. 5. The notations used here are based on log-likelihood ratio (LLR): $L_a(\cdot)$ denotes \textit{a priori} LLR, $L_d(\cdot)$ denotes \textit{a posterior} LLR, and $L_e(\cdot)$ denotes extrinsic LLR.

**IV. SIMULATION RESULT**

**A. BER of The Proposed Decoding**

Throughout the simulation, we use a rate 1/2 convolutional code (CC) which is optimal within the class of constraint length $K = 7$ as an outer code. The interleaver $\Pi$ is an S-random interleaver [7] with $S = 10$. Assumed Markov processes are $m_m = 1, 2, 4$ with respect to the shaping context: shaping metric is based on the moment method and external memory is 3 (refer to [2] for these shaping parameters). The roll-off factor of pulse-shaping filter is 0.4 and the shaping matrices are those given in Section II-A. Hence $m_m = 2$. It is shown in [2] that the shaping system with the aforementioned parameters can achieve peak-to-average power ratio (PAR) reduction of 1.8 [dB].

Fig. 6 shows the bit error rate (BER) of the concatenated TS and CC system, in conjunction with the case of TS alone. The curve labeled with $m_m = 0$ corresponds to the TS system which does not take the symbol correlation into account by means of assigning all the transition probabilities equiprobable. In this case, the TS decoder does not exploit the Markov property, but is still able to pass the soft-output to the outer decoder. As a result, the iterative processing does work, but yields no significant gain.

As observed, the coding gain achieved by the TS without iterative decoding is insignificant, even when the long order ($m_m = 4$) is considered. On the other hand, the iterative decoding together with the outer code achieves significant coding gain. The result shows that BER is improved gradually but steadily, with increasing $m_m$, the number of iteration, and interleaver length. It is remarkable that the combination of $m_m = 4$, 20 iterations and a long interleaver exhibits so-called water fall behavior and achieves a coding gain of more than 10 [dB] compared to the conventional decoding of TS. Furthermore, in this case, there is no error floor observed at least in the simulated BER region, whereas BERs associated with $m_m = 1$ and 2 have noticeable error floor. This error floor may be alleviated by a carefully designed interleaver.

On the other hand, the beginning of water fall region between $m_m = 2$ and $m_m = 4$ is almost identical. Thus, considering the tremendous number of 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 trade-off between decoding complexity and error rate performance.

In this simulation, we examined $m_m = 1$ for the purpose of demonstrating the effect of assumed Markov order on the resulting BER. It is interesting to note that $m_m = 2$ virtually outperforms $m_m = 1$ 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 $H_s$ of $m_s = 2$, prunable states also increase as $m_m$ increasing. Hence, $m_m = 2$ has rather lower complexity than $m_m = 1$, and yet achieves better BER performance. For example, in this shaping setup, the number of states (excluding the prunable states) is 64, 56, and 1079 for $m_m = 1, 2,$ and 4, respectively.

B. Average Mutual Information

It is well recognized that, for a given modulation format, the AMI is maximized if and only if the modulated symbols follow i.i.d. and equiprobable distribution. (The resulting AMI is called constellation-constrained channel capacity.) On the other hand, in the presence of the TS, the AMI would be reduced due to the memory embedded in the modulated symbols. To quantitatively derive the AMI-$E_b/N_0$ characteristic in this scenario, we employed a simulation-based approach given in [8]. This approach executes a procedure similar to BCJR algorithm to compute the entropy at the receiver, given that the order of Markov process $m_m$ is computationally tractable (we considered up to $m_m = 4$). In general, the higher the inherent Markov order, the smaller the resulting AMI, and thus the AMI obtained from this computation serves as an upper bound of the actual AMI. (Recall that we stated in Section III-A that higher order is required to precisely capture the Markov process of the shaped symbols.)

Fig. 7 shows the computed AMI in the presence of TS with the same shaping parameters as given in the previous subsection. Comparing the four AMI-$E_b/N_0$ curves associated with $m_m = 1, 2, 3,$ and 4, we can state that the curve with $m_m = 4$ is sufficiently convergent to approximate the actual AMI-$E_b/N_0$ curve at least around AMI = 1 [bit/symbol] which equals to the shaping/coding system discussed in this paper.

To verify the coding gain shown in Fig. 6, we plot the $E_b/N_0$ required for BER = $10^{-5}$ with respect to the overall rate (the star symbol in Fig. 7). As observed, the gap between the star and AMI is as small as 1.5 [dB], which demonstrates the validity of the proposed decoding.

For comparison, coding system without shaping is also examined. For example, Berrou’s turbo code [9] (rate $r_c = 1/3$) concatenated to 8-PSK via interleaver (BICM) is plotted with circle in the figure. This system has the same overall rate as the considered shaping/coding system. From this result, we see that there remains a gap in the required $E_b/N_0$ of 2 [dB] compared to the case without shaping. However, since the proposed TS [2] reduces the PAR by 1.8 [dB], the overall power efficiency is typically more than compensated for. It should be mentioned that PAR reduction enhances the efficiency of PA substantially [10], and this efficiency improvement is not explicitly comparable with the loss in terms of the required $E_b/N_0$.

V. CONCLUSION

In this paper, we have proposed a new iterative decoding of trellis shaping for peak power reduction by making use of the Markov model formed by shaped symbols. Simulation results have demonstrated the significant improvement of BER. Thus, we can enjoy the PAR reduction of TS without noticeable degradation of BER performance.

REFERENCES