Performance of a Novel Hybrid ARQ Scheme for Trellis Coded Modulation in Wireless Networks

Qian Huang, Li Ping, Sammy Chan, King-Tim Ko

Abstract—A novel trellis-coded modulation-based hybrid ARQ scheme which provides multiple levels of error correction is proposed in this paper for efficient and reliable data transmission. The analytical throughput bounds of the proposed scheme are derived and verified by simulations. Numerical results demonstrate that an improved throughput is achieved by the proposed scheme over other recently proposed hybrid ARQ schemes.

Index Terms—Automatic repeat request (ARQ), Hybrid ARQ, wireless networks.

I. INTRODUCTION

With the growing demands of TCP/IP-based applications in the next generation of mobile networks comprising both wired and wireless systems, the mechanisms providing efficient and reliable transmission over error-prone and bandwidth-limited wireless links become very important, because packet loss caused by transmission errors over wireless links will lead to serious end-to-end TCP throughput degradations. Hybrid Automatic Repeat reQuest (ARQ), which combines the advantages of ARQ and forward error correction (FEC) for reliable transmission, is a local error recovery solution to alleviate the adverse impacts from wireless links on TCP performance.

Commonly used hybrid ARQ can be classified into three types. In conventional Type-I hybrid ARQ, e.g., the scheme of [1], a copy of the encoded data sequence is retransmitted and decoded separately. Some Type-I hybrid ARQ schemes are enhanced by code combining and/or trellis-coded modulation (TCM) in order to improve efficiency, such as the scheme of [2] which uses both TCM and code combining. The performance of code-combining depends on the combining strategies used. Schmitt suggested in [3] to use soft combining of the squared distances between the soft detector output and the possible signals. If $\mathbf{r}_i = (r^{(1)}_i, r^{(2)}_i, \ldots, r^{(N)}_i)$ is the $i$-th received version of a packet and $r^{(n)}_i$ are the $2^{m+1}$ dimensional vectors representing the $n$-th symbol, the resulting packet obtained by combining $L$ received versions of the packet is $\mathbf{z}_i = (z^{(1)}_i, z^{(2)}_i, \ldots, z^{(N)}_i)$ where $z^{(j)}_i = \sum_{l=1}^{L} r^{(j)}_{i_l}$. This combining strategy has better throughput performance than that of the average diversity combining (ADC) scheme [4] which simply averages the soft decision values of the coordinates of each received symbol in all packets.

In Type-II hybrid ARQ, the incremental redundancy bits are retransmitted and combined with previously received bits of the same data packet for decoding, e.g., the schemes proposed in [5] and [6]. Obviously, Type-II hybrid ARQ has better throughput performance than Type-I hybrid ARQ. However, most proposed TCM-based hybrid ARQ schemes are Type-I, since the information bits and parity bits in conventional TCM codes cannot be separated. A near TCM-based Type-II hybrid ARQ is proposed in [7], which only retransmits redundancy in smaller increments instead of complete encoded packets. Its objective is to reduce bandwidth usage for signal-to-noise ratio (SNR) in which such additional amount of redundancy is sufficient for decoding. In this scheme, when retransmission is required, partial packets of $1/k$ of the original packet length, formed by mapping only every $k$-th channel symbol label, are transmitted. The obtained squared distances are soft combined with the corresponding squared distances of the first transmission. The resulting TCM scheme has improved distance properties than that in [3]. If the combined packet cannot be decoded correctly, another partial packet is formed and transmitted to the receiver. This process continues until the packet is decoded successfully or $k$ different partial packets have been sent. After $k$ retransmissions, subsequent retransmission will involve a complete copy of the packet.

Type-III hybrid ARQ scheme is defined by the 3GPP specifications for Radio Link Protocol (RLP) of the third generation (3G) mobile networks [8]. It is similar to Type-II that different sequences of the same data packet are retransmitted alternatively, but differs in that each retransmitted sequence contains both data information and parity bits and can be self-decoded. The hybrid ARQ scheme proposed in [9] is similar to Type-III. In order to improve the performance of code-combining
in hybrid ARQ, two different TCM codes are used for retransmissions. For initial transmission, the message is encoded by the first TCM code. If the received copy contains uncorrectable errors, the message is retransmitted, but encoded by the second TCM code. For further retransmissions, the first and second TCM codes are used alternatively. It has been shown that this scheme improves the rate at which the squared free distance of the combined codes increases.

In this paper, a Type-III hybrid ARQ scheme for TCM is proposed. This scheme is detailed in Section II, which is followed by performance evaluation by comparisons between the proposed scheme and other recently proposed hybrid ARQ schemes in Section III. Finally, we present our concluding remarks in Section IV.

II. SCHEME DESCRIPTION

The coding algorithm for which our hybrid ARQ scheme is designed is a concatenated two-state trellis-coded modulation (CT-TCM) [10]. Compared with the existing turbo-TCM schemes, CT-TCM codes have significantly reduced decoding complexities and still demonstrate improved performances, such as better waterfall behavior, lower error floor and higher spectral efficiency for high-order modulation.

Fig. 1 shows the encoder structure of the CT-TCM algorithm. \(d = \{d_k\} (k = 1, 2, \ldots, N)\) represents the input \(2^n\)-ary information sequence containing \(N\) symbols. Each symbol \(d_k\) comprises \(n\) information bits and has \(d_k = (d_{k,0}, \ldots, d_{k,n-1})\). The information symbols \(d_k\) is sent to \(M\) encoders interconnected with the CT-TCM scheme. \(z(i)\) represents symbol-based random interleaving rule with the constrain: \(z(i) \mod M = i \mod M\). The output of each tree encoder is a sequence of \(2^{(n+1)}\)-ary coded symbols, each of the form \(c_k = (d_k, q_k)\), where \(q_k = q_{k-1} + d_k \cdot g_k\). \(g_k = (g_{k,0}, g_{k,1}, \ldots, g_{k,n-1})^T\) is an indication vector defined by,

\[
g_{k,j} = \begin{cases} 
1 & \text{if } d_{k,j} \text{ participates in parity check}, \\
0 & \text{otherwise}.
\end{cases}
\]

The coded symbols \(\{c_k\}\) from tree encoders are mapped to appropriate \(2^{n+1}\)-ary constellations, generating \(M\) groups of modulated symbol sequences \(\{x_k^{(0)}\}, \{x_k^{(1)}\}, \ldots, \{x_k^{(M-1)}\}\). For each input information symbol \(d_k\), its modulated symbol \(x_k\) contains \(M\) components, which are denoted as \(x_k^{(0)}, x_k^{(1)}, \ldots, x_k^{(M-1)}\). Let \(\{\Gamma_i\}, i = 1, 2, \ldots, M\), indicate the set of puncture patterns related to a CT-TCM processor containing \(M\) encoders. The puncture processing performs as follows. Given the puncture pattern \(\Gamma_i\), \(i\) components of each modulated symbol \(x_k\) are selected to use in the proposed ARQ scheme, and the rest of components is punctured before transmission.

Table I shows an example of the selected sequences generated by different puncture patterns in the case of 8PSK-based CT-TCM deployed, that is \(M = 4\). In this example, each modulated symbol of \(d_k\) contains four components. Denote the \(i\)th modulated component as \(x_k^{(i)}, i = 0, 1, 2, 3\). Provided the puncture pattern, for each modulated symbol \(x_k\), its components corresponding to “1” in Table I are selected to use and the components corresponding to “0” are punctured. For example, the selected sequence related to puncture pattern \(\Gamma_2\) follows \(x_k^{(0)}, x_k^{(2)}, x_k^{(1)}, x_k^{(3)}, x_k^{(0)}, x_k^{(2)}, x_k^{(1)}, x_k^{(3)}\). If the puncture pattern \(\Gamma_4\) is used, all components \(\{x_k^{(i)}, i = 0, 1, 2, 3\}\) are preserved to use in the proposed ARQ scheme. Totally, four patterns of selected modulated-sequences are obtained using 8PSK-based CT-TCM.

Different from the Type-II hybrid ARQ schemes in [5], [6] and the TCM-based hybrid ARQ scheme in [9],
which provide only two levels of error correction, the proposed ARQ scheme based on CT-TCM coding provides multiple levels of error correction capability. For a CT-TCM processor with \( M \) branches of encoders, input data sequence \( s \), \( M \) patterns of selected modulated-sequences are produced and provide \( M \) levels of error correction capability. Denote \( M \) patterns of selected modulated-sequences produced by such CT-TCM as \( s_1, s_2, \ldots, s_i, \ldots, s_M \), corresponding to puncture pattern \( \Gamma_1, \Gamma_2, \ldots, \Gamma_i, \ldots, \Gamma_M \) respectively. The ARQ process for a data sequence \( s \) are presented as follows:

**Level 1:** Initially, the transmitter adopts puncture pattern \( \Gamma_1 \) to obtain the selected modulated-sequence \( s_1 \) of \( s \) for the first transmission. The receiver decodes the received sequence \( s'_1 \) using CT-TCM decoding algorithm and checks whether it contains error. If no error detected, the receiver returns an acknowledgement (ACK) to the transmitter. If errors are detected, the receiver stores \( s'_1 \) and returns a negative acknowledgement (NAK) to the transmitter, then it moves to **Level 2**.

**Level \( i \), \( 2 \leq i \leq M \):** At level \( i \), puncture pattern \( \Gamma_i \) is used to obtain the selected modulated-sequence \( s_i \) of \( s \). The differential components between the newly produced modulated-sequence \( s_i \) and the previously produced \( s_{i-1} \) related to \( \Gamma_{i-1} \), denoted as \( \delta_i \), is transmitted to the receiver. The received differential sequence \( \delta'_i \) is combined with the previously received sequences \( \delta'_1, \delta'_2, \delta'_3, \ldots, \delta'_{i-1} \) of the same data \( s \) for decoding. Note that the combined sequence of \( \delta'_i \) and previous \( s'_1, \delta'_2, \delta'_3, \ldots, \delta'_{i-1} \) approximates the selected modulated-sequence \( s_i \), and thus is decoded by \( \Gamma_i \) at the receiver. Then the receiver checks whether the decoded data contains error. If it is error free, data \( s \) is successfully received. Otherwise, the receiver stores the received \( \delta'_i \) and returns NAK to the transmitter. Then, it moves up to **Level \((i+1) \mod M\)**. In other words, after level \( M \), the next retransmission of data \( s \) will start from \( \Gamma_1 \) again.

### III. PERFORMANCE EVALUATION AND COMPARISONS

We consider the retransmission based on selective repeat. Let \( P(C_i^{(j)}) \), \( P(U_i^{(j)}) \), \( P(D_i^{(j)}) \) represent the probabilities of “decoded sequence contains no error”, “decoded sequence contains undetected errors”, “decoded sequence contains detected errors”, respectively, for the \( j \)th transmission of the same data by \( \Gamma_i \). Let \( P(E_i^{(j)}) \) represent the error probability of decoding the \( j \)th transmission of the same data sequence related to \( \Gamma_i \). Clearly, it has \( P(E_i^{(j)}) = 1 - P(C_i^{(j)}) = P(U_i^{(j)}) + P(D_i^{(j)}) \). Actually, the values of \( P(E_i^{(j)}) \) for each received sequence are obtained from the simulation results of the CT-TCM decoding algorithm [10]. Assuming \( P(U_i^{(j)}) \) negligible as in [6], [9], we have \( P(D_i^{(j)}) \approx P(E_i^{(j)}) \). Let \( \Lambda \) denote the average number of transmissions of a data sequence. Because each sequence produced by the CT-TCM transmitter for each transmission contains same number of symbols \( N \) as the input data sequence, the throughput efficiency of the proposed scheme, denoted as \( \eta \) and defined as the average number of information bits per transmitted channel symbol, is given by \( \eta = n/\Lambda \), where \( n \) is the number of information bits carried by each channel symbol. Based on the analysis in [6], the throughput bounds of our hybrid ARQ scheme can be derived as follows.

### TABLE I

**SELECTED SEQUENCES VS. PUNCTURE PATTERNS (\( M = 4 \))**

<table>
<thead>
<tr>
<th>Puncture pattern</th>
<th>modulated component ( i )</th>
<th>( \delta_{k} )</th>
<th>( \delta_{k+1} )</th>
<th>( \delta_{k+2} )</th>
<th>( \delta_{k+3} )</th>
<th>( \delta_{k+4} )</th>
<th>( \delta_{k+5} )</th>
<th>( \delta_{k+6} )</th>
<th>( \delta_{k+7} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Gamma_1 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( \Gamma_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( \Gamma_3 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( \Gamma_4 )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>


Fig. 2 shows the performance of frame error rate (FER) towards average bit energy to noise power spectral density ratio \((E_b/N_0)\) by the CT-TCM algorithm. It demonstrates that proper puncture pattern selected for transmission can enhance the error recovery capability and lead to reliable data transmissions over wireless channels.

![Fig. 2. The frame error rate performance for CT-TCM algorithm using different puncture patterns.](image)

We develop a link layer simulation model by OPNET [11] to verify the analytical throughput bounds in (1). The performance of the proposed hybrid ARQ scheme in the case of additive white Gaussian noise (AWGN) channel is considered. For simplicity, we also assume that the feedback channel is error free and thus the acknowledgement packet returned by the receiver can be always be correctly received by the sender.

Fig. 3 shows that, for selective repeat, the upper and lower bounds of the throughput obtained by (1), and the throughput obtained by simulation are effectively overlapped. This means that the bounds are very tight and accurate. For comparisons, also included in Fig. 3 are the simulation results of the throughput performance of the code combining based Type-II hybrid ARQ scheme [6] and the 8PSK-TCM based Type-III hybrid ARQ scheme [9] under the same assumption of AWGN channel. We observe that the proposed hybrid ARQ scheme provides higher throughput over the other two.

![Fig. 3. Throughput performance comparisons.](image)

In Fig. 4, we compare the performance of throughput against average symbol energy to noise power spectral density ratio \((E_s/N_0)\) of our scheme with other recently proposed TCM-based hybrid ARQ schemes. Here, the throughput is measured by the number of bits per symbol (bits/symbol). We observe that the proposed scheme achieves higher efficiency over a large range of \(E_s/N_0\).
values than the near TCM-based Type-II scheme of [7] and other TCM-based Type-I schemes [2] and [3].

IV. CONCLUSION

A hybrid ARQ scheme based on the concatenated two-state trellis-coded modulation (CT-TCM) algorithm has been proposed in this paper for efficient and reliable data transmission over wireless links. The analytical throughput bounds of the proposed scheme are derived and verified by simulation results based on the link layer model we developed. Both analysis and simulation results show that higher improvement of the throughput is achieved by the proposed TCM-based hybrid ARQ scheme than other recently proposed hybrid ARQ schemes under AWGN channel. The study of the impacts from the proposed scheme to TCP performance is in progress. Relevant results will be presented in our future works.

ACKNOWLEDGMENTS

The work described in this paper was jointly supported by a grant from City University of Hong Kong (Project No. 7001124) and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No., CityU 1190/02E].

REFERENCES