Optimum Subpacket Transmission for Hybrid ARQ Systems

Yiqing Zhou, *Member, IEEE*, and Jiangzhou Wang, *Senior Member, IEEE*

***Abstract*—Hybrid automatic-repeat-request (ARQ) is a flexible and efficient technique for data transmissions. In hybrid ARQ, subpacket schemes are more attractive for systems with burst errors than complete packet schemes. Although subpacket schemes were proposed in ARQ systems, optimum subpacket transmission is more effective to maximize throughput in a dynamic channel. Since convolutional codes have properties of burst errors in decoding, the optimum subpacket can be applied to convolutional codes. This paper investigates the performance of subpacket transmission for convolutionally coded systems. An efficient method is proposed to estimate the optimum number of subpackets, and adaptive subpacket schemes, i.e., schemes that enable a system to employ different optimum numbers of subpackets under various conditions, are suggested to achieve the maximum throughput of the system. Numerical and simulation results show that the adaptive subpacket scheme is very effective for the convolutionally coded hybrid ARQ system, and it can provide higher throughput, smaller delay, and lower dropping rate than complete packet schemes. Moreover, the adaptive subpacket scheme can be flexibly used with packet combining techniques to further improve the system throughput.**

***Index Terms*—Convolutional codes, cyclic redundancy check (CRC) codes, hybrid automatic-repeat-request (ARQ) systems, packet-error patterns, packet schemes, punctured codes.**

VARIABLES LIST

$A_{w,d}$	Number of convolutional codewords with Hamming weight d, generated from input sequence with Hamming weight w.
d	Hamming weight of convolutional codewords.
L_{cb}	Length of input bits to convolutional encoder.
L_{crc}	Number of parity bits in cyclic redundancy check (CRC) codes.
L_f	Length of output bits after convolutional encoding.
N	Number of subpackets.
N_{opt}	Optimum number of subpackets.
M	Maximum transmission time for a subpacket.
P_p	Tangential sphere bound on the packet error rate (PER).
$P_{\text{sp}}(N)$	Subpacket error rate (SPER) for the N-subpacket scheme.
$P_w(N)$	Whole PER for the N-subpacket scheme.
R_{eff}	Effective code rate including the redundancy introduced by CRC codes.
S_d	Total number of convolutional codewords with Hamming weight d.
T	Time needed to encode, transmit, and decode a subblock of data.
α_d, $\beta_d(z_1)$, r_{z_1}	Variables used in the tangential sphere bound.
δ_d	Euclidean distance between two codewords in d different positions.
$\overline{\gamma}(a, x)$	Normalized incomplete gamma function.
$\Gamma(x)$	Gamma function.
η	Throughput.

I. INTRODUCTION

THE automatic-repeat-request (ARQ) technique was introduced in the early days of data communications as a consequence of the development of parity-check codes [1], [2]. The main advantage of this technique is that it can adapt to the channel conditions at low complexity. Further, forward-error correction (FEC) and error detection were introduced into ARQ protocols, and the resultant schemes are known as hybrid ARQ protocols today. The hybrid systems can achieve throughput similar to the pure FEC systems, and provide the good reliability and flexibility of pure ARQ protocols.

In data communication systems, information sequences are transmitted usually in packets with fixed length. At a receiver, error correction and detection are carried out in the whole packet. If the packet is found in error, the receiver sends a request to the transmitter via a feedback channel, and then the whole packet is retransmitted. However, such a conventional complete-packet ARQ scheme is inefficient in the presence of burst errors. As shown in Fig. 1, when the whole packet is divided into a few subpackets, all or majority of the bit errors may be located in one subpacket, leaving other subpackets error-free. But in the complete packet scheme, the whole packet will be retransmitted no matter how bit errors are distributed.

Paper approved by Y. Fang, the Editor for Wireless Networks of the IEEE Communications Society. Manuscript received February 7, 2003; revised July 27, 2005.

Y. Zhou is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong (e-mail: yqzhou@eee.hku.hk).

J. Wang is with the Department of Electronics, University of Kent, Canterbury, Kent CT2 7NT, U.K. (e-mail: J.Z.Wang@kent.ac.uk).

Digital Object Identifier 10.1109/TCOMM.2005.863724

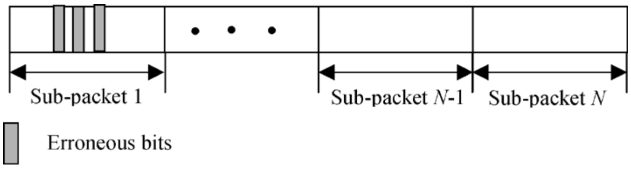

Fig. 1. Burst packet-error patterns.

This situation can be improved if subpacket schemes are employed. In subpacket transmissions, only those subpackets that include errors need to be retransmitted.

Many powerful FEC codes have been considered for hybrid ARQ systems, such as turbo codes [3] and convolutional codes [4], [5]. Turbo codes have been shown to have excellent performance [6]. However, turbo decoding is very complicated. Convolutional codes are still attractive in real applications, due to its simple realization. Convolutional codes can be obtained by shift-register techniques, and exhibit good error-correction capability by use of the maximum-likelihood decoding scheme, i.e., Viterbi decoding scheme [7]. Convolutional codes have been widely employed in many communication systems, such as deep-space and satellite communication systems, second generation (2G) and third generation (3G) mobile communication systems [8], [9], and so on. A typical convolutional encoder is involved with one-bit input, a few-bit output, and constraint length m_c. The performance of convolutional codes is determined by the Viterbi decoding, which selects survival paths in making decisions. Thus, convolutional codes have a property of burst errors.

Although subpacket schemes were proposed in ARQ systems [10], there is no paper discussing the optimum number of subpackets, and a fixed subpacket scheme only provides improved throughput in a small range of signal-to-noise ratio (SNR). Ascertaining the optimum number of subpackets in one transmission is the objective of this paper. Exploiting the burst-error property of convolutional codes, an adaptive subpacket scheme is proposed to provide the highest throughput in a dynamic channel. The paper is organized as follows. In Section II, the system model and subpacket schemes are introduced. In Section III, packet error rate (PER) performance of FEC codes is analyzed. Section IV is devoted to finding the optimum number of subpackets for a system to maximize the throughput. An efficient algorithm is proposed to estimate the optimum number of subpackets at variable SNR. Representative analytical and simulation results of throughput, delay, and dropping rate are also presented. Finally, conclusions are drawn in Section V.

II. System Overview

In a hybrid ARQ system, the information data blocks are encoded with error detection and FEC codes. The coded bit stream modulates a carrier using binary phase-shift keying (BPSK). After passing through a channel (additive white Gaussian noise (AWGN) channel) and being demodulated, the baseband signals are decoded by *error correction* and checked by *error detection*. Finally, acknowledgment signals, i.e., positive acknowledgment (ACK) and negative acknowledgment (NAK), are transmitted back to the transmitter over an ideal feedback channel that is assumed to be error-free.

Suppose that an information data block of L-bit length is transmitted with ARQ techniques. The whole information data block is segmented into N subblocks, as illustrated in Fig. 2(a), which consists of N error-detection encoders and one FEC encoder. Each detection code is an (n, k) systematic block code, where n is the length of one codeword (one subpacket) and k is the length of information bits in each subblock ($k = L/N$ is an integer). That is, each subblock passes through an error-detection encoder that produces a subpacket of n bits. All of the subpackets are then time-multiplexed into a whole packet with length $L_{cb} = n \cdot N$ for the convolutional encoder with possible punctuation. The final output is a channel-coded word of length L_f. The subpacket structure is shown in Fig. 2(b), where each subpacket includes a systematic error-detection code of $n - k$ bits. In the following analysis, two different PERs are used in the N subpacket schemes. One is the SPER, denoted as $P_{\text{sp}}(N)$, and another one is the whole PER, i.e., the codeword error rate of the FEC code, denoted as $P_w(N)$. Note that the conventional complete-packet scheme can be treated as a special case of the subpacket scheme with $N = 1$. That is, the PER of the complete-packet scheme is simply represented by $P_w(1)$ or $P_{\text{sp}}(1)$.

A systematic CRC code with 16 parity bits ($L_{\text{crc}} = 16$) in Consultative Committee of International Telephone and Telegraph (CCITT) standards is used as the error-detection code [2], [9]. Its generator polynomial is $\mathrm{g}(D) = D^{16} + D^{12} + D^5 + 1$, where D stands for bit delay operation. The CRC code is easy to implement, but it is not easy to evaluate its performance in traditional methods, such as weight distribution and minimum distance. However, there are some other meaningful measures for the performance of CRC codes, such as the "burst-error detection capability" and "error-detection coverage (λ)." A binary CRC code with L_{crc} parity bits can detect all burst errors of length L_{crc} or less, and detect the fraction $\lambda = 1 - 2^{-L_{\text{crc}}}$ of all the error patterns. On a binary symmetric channel, when the code dimension k and the error probability are large, the undetected error probability of CRC codes approaches $2^{-L_{\text{crc}}}$. Thus, the performance of CRC codes is mainly determined by L_{crc}. The longer L_{crc}, the better the performance. The CRC code with $L_{\text{crc}} = 16$ can provide adequate detection for most applications.

A 1/2-rate convolutional code is employed with constraint length $m_c = 4$ (or memory length of $m_c - 1 = 3$) and generator polynomial $\mathrm{G}(D) = [1 + D + D^3, 1 + D + D^2 + D^3]$, or (64, 74) [2]. Higher rate codes are obtainable by puncturing from the low-rate code [11]. For example, the compatible 2/3-rate and 5/6-rate codes can be obtained from the 1/2-rate code by means of the puncturing matrices shown in the first equation at the bottom of the page, respectively, where "1" means that the code bit will be transmitted, and "0" means that the code bit will be dropped and not transmitted, as shown in the first equation at the bottom of the next page.

Consider the ARQ protocol of subpacket schemes. As shown in Fig. 2(b), subpackets are independent of each other in CRC coding processes. The N subpackets named $\{\mathrm{SP}_1, \mathrm{SP}_2, \ldots, \mathrm{SP}_N\}$ are time-multiplexed into a single whole packet to be encoded by the FEC coder. At the receiver side,

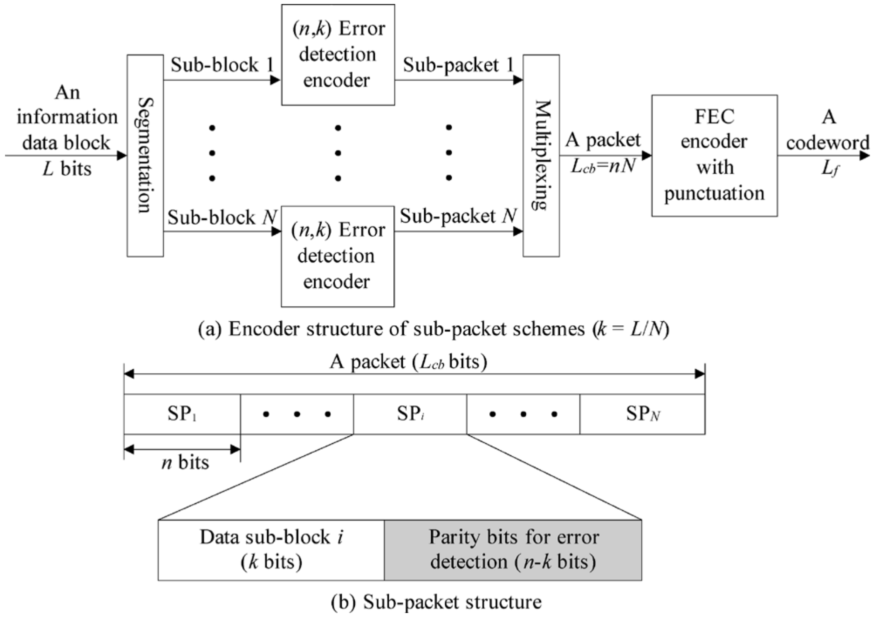
An information data block
L bits
Segmentation
Sub-block 1
Sub-block N
(n,k) Error detection encoder
(n,k) Error detection encoder
Sub-packet 1
Sub-packet N
Multiplexing
A packet
$L_{cb}=nN$
FEC encoder with punctuation
A codeword
L_f
(a) Encoder structure of sub-packet schemes (k = L/N)
A packet (L_{cb} bits)
SP_1
SP_i
SP_N
n bits
Data sub-block i (k bits)
Parity bits for error detection (n-k bits)
(b) Sub-packet structure

Fig. 2. Subpacket schemes. (a) Encoder structure of subpacket schemes $(k = L/N)$. (b) Subpacket structure.

Viterbi decoding is carried out, and the syndrome of CRC codes is calculated for each subpacket SP_i $(i = 1, 2, \ldots, N)$. If there is no error in a subpacket, the corresponding decoded subblock is stored in a buffer and an ACK is sent to the transmitter. However, if there are some erroneous subpackets after the decoding is finished, then the corresponding NAK is sent to the transmitter for retransmission of the erroneous subpackets. All correctly detected subblocks are stored in a buffer, and sequentially forwarded to the higher layer. At the transmitter, since the codeword of the previous transmission has been stored, the corresponding parts in the codeword of the subpackets to be retransmitted can be obtained and retransmitted to the receiver. At the receiver, the signals of the previous transmission are needed. The retransmitted parts will replace the erroneous parts in the previous codeword, and the whole codeword is decoded again. The maximum transmission times for a particular subpacket is limited to M. If the subpacket is still in error after M transmissions, then it will be dropped out.

III. Performance Bound

In this paper, the tangential sphere bound is used, since it is a tight bound for binary linear codes [12]–[14]. Compared with the union bound [1], [2], the tangential sphere bound gets much closer to the actual performance at low-to-moderate SNR. The total number of codewords with Hamming weight d is defined as

$$S_d = \sum_{w=0}^{L_{cb}} A_{w,d}, \quad d = 0, 1, \ldots, L_f \tag{1}$$

where $A_{w,d}$ is the number of codewords with Hamming weight d, generated from information bits with Hamming weight w. Assuming that the variance of the additive channel noise is σ^2, the tangential sphere bound on the PER P_p [14] is given by (2), shown at the bottom of the page, where the sum is only taken

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 1 & 0 & 1, & 1 & 0 & 0 & 1, & 1 & 1 & 0 & 1, & 1 & 0 & 0 & 1, & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0, & 0 & 1 & 1 & 1, & 0 & 1 & 1 & 0, & 0 & 1 & 1 & 1, & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$P_p \le \int_{-\infty}^{+\infty} \frac{dz_1}{\sqrt{2\pi}\sigma} e^{-z_1^2/2\sigma^2} \cdot \left\{ 1 - \overline{\gamma}\left(\frac{L_f - 1}{2}, \frac{r_{z_1}^2}{2\sigma^2}\right) + \sum_{\substack{d \\ \delta_d/2 < \alpha_d}} S_d \left[Q\left(\frac{\beta_d(z_1)}{\sigma}\right) - Q\left(\frac{r_{z_1}}{\sigma}\right)\right] \overline{\gamma}\left(\frac{L_f - 2}{2}, \frac{r_{z_1}^2 - \beta_d^2(z_1)}{2\sigma^2}\right) \right\} \tag{2}$$

over $\{d\}$ which satisfy $\delta_d/2 < \alpha_d$, and variables r_{z_1}, $\beta_d(z_1)$, and α_d are further defined as

$$\begin{cases} r_{z_1} = \left(1 - \dfrac{z_1}{\sqrt{L_f E_s}}\right) r \\ \beta_d(z_1) = \dfrac{r_{z_1}}{\sqrt{1 - \dfrac{\delta_d^2}{4 L_f E_s}}} \cdot \dfrac{\delta_d}{2r} \\ \alpha_d = r\sqrt{1 - \dfrac{\delta_d^2}{4 L_f E_s}} \end{cases} \tag{3}$$

where E_s is the energy per coded bit. Thus, the total energy of a codeword is $L_f E_s$. δ_d is defined as the Euclidean distance between two codewords in d different positions $(d \leq L_f)$. For antipodal signals, $\delta_d = 2\sqrt{dE_s}$. Furthermore, $\overline{\gamma}(a, x)$ in (2) is the normalized incomplete gamma function, given by

$$\overline{\gamma}(a, x) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t} dt, \quad a > 0, x > 0 \tag{4}$$

where $\Gamma(x)$ is the gamma function, given by

$$\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt, \quad x \geq 0. \tag{5}$$

In (2), $Q(x)$ is the Q function, expressed as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-t^2/2dt}, \quad x \geq 0. \tag{6}$$

By setting the derivative of (2) to zero, an optimal r can be found to achieve the tightest upper bounds [14].

To calculate the tangential sphere bound, it is necessary to evaluate (1) first. Although many efforts have been made by experts to find the distance spectra [15]–[18], it remains a problem for most codes. Fortunately, for convolutional codes with small constraint lengths, (1) can be quickly calculated [18] when the codeword is not too long. Since the computation complexity increases exponentially with the constraint length, the tangential sphere bound is only calculated for the 1/2-rate convolutional code with constraint length of four. Given $L_{cb} = 1200$, the length of the codeword is 2406, including 6 tail bits. Otherwise, an overflow will occur because the calculation involves very large values. Convolutional codes with other code rates, constraint lengths, and code lengths will be studied as well.

IV. Analysis and Simulation

As explained in the Introduction, subpacket schemes are good schemes in the presence of burst errors. This section is devoted to finding the optimum subpacket schemes to maximize the system throughput. Consider an N-subpacket scheme with an (n, k) error-detection code. The effective code rate of the packet scheme is defined as

$$R_{\text{eff}} = \frac{L}{L_f} = \frac{N \cdot k}{L_f} = \frac{N \cdot (n - L_{\text{crc}})}{L_f} = \frac{L_{cb} - N \cdot L_{\text{crc}}}{L_f} \tag{7}$$

which accounts for the redundancy introduced by error-detection codes. Define $\text{SNR}_b = E_b/N_0$ as the SNR per information bit, where E_b is the energy per information bit, and N_0 is the power spectrum density of channel noise, and $\text{SNR}_c = R_{\text{eff}} \cdot \text{SNR}_b$ as the SNR per channel bit. Obviously, when all the other parameters are fixed, SNR_c (or R_{eff}) decreases as N increases. Given $L_{\text{crc}} = 16$ and N, one way to alleviate the effect of N on SNR_c is to apply subpacket schemes in long-packet-length systems (large L_{cb}).

A. *Optimum Subpacket Schemes*

Throughput is chosen as the criterion for optimizing the number of subpackets N. Consider a truncated system that employs a selective repeat-ARQ (SR-ARQ) protocol with M maximum transmissions. The throughput is defined as [1], [2]

$$\eta = R_{\text{eff}} \cdot [1 - P_{\text{sp}}(N)] = \frac{L_{cb} - N \cdot L_{\text{crc}}}{L_f} \cdot [1 - P_{\text{sp}}(N)]. \tag{8}$$

When L_{cb}, L_{crc}, L_f (or the code rate), and SNR_b are given, the throughput only depends on N. When the number of subpackets N is very large, R_{eff} tends to be small, so that the throughput η becomes small. However, when N is very small, the SPER $P_{\text{sp}}(N)$ tends to be large, so the throughput η becomes small, as well. Therefore, there must exist a value of N at which the throughput takes a maximum value. The optimum number of subpackets (N_{opt}) that achieves the highest throughput can be obtained by setting the derivative $d\eta/dN$ to zero. N_{opt} can then be obtained from

$$N = \frac{L_{cb} \cdot \dfrac{dP_{\text{sp}}(N)}{dN} + L_{\text{crc}} \cdot [1 - P_{\text{sp}}(N)]}{L_{\text{crc}} \cdot \dfrac{dP_{\text{sp}}(N)}{dN}}. \tag{9}$$

If $P_{\text{sp}}(N)$ is given, then N_{opt} can be calculated from (9) directly. Unfortunately, it is difficult to obtain a closed form of $P_{\text{sp}}(N)$. Under the assumption of burst packet-error patterns and long packet lengths, N_{opt} can be obtained from (9) by approximating $P_{\text{sp}}(N)$ with its average upper bound as follows.

For the hybrid ARQ system employing an N-subpacket scheme, the error rate of the whole packet, $P_w(N)$, can be approximated by the tangential bound (2). Note that the input sequence to the FEC encoder is not pure information bits, but N time-multiplexed CRC codewords. CRC encoders will change the weight distribution of the input sequence. As S_d in (1) is the number of FEC codewords with weight d, it is a function of the number of subpackets. In addition, the effective code rate is a function of N. For long packet lengths, the effect of N on weight distribution and code rate are not significant. Moreover, the PER of convolutional codes is not sensitive to the change in

SNR. Thus, $P_w(N) \approx P_w(1)$. Furthermore, since bit errors are in burst, $P_{\rm sp}(N)$ can be approximated by $P_w(N)/N$

$$P_{\rm sp}(N) \approx \frac{P_w(N)}{N} \approx \frac{P_w(N)}{N} \approx \frac{P_p}{N}. \tag{10}$$

Then, $N_{\rm opt}$ can be obtained from (9) and (10)

$$N_{\rm opt} \approx \sqrt{\frac{L_{cb} \cdot P_p}{L_{\rm crc}}}. \tag{11}$$

Equation (11) clearly describes the relationship between $N_{\rm opt}$ and other system parameters. First, $N_{\rm opt}$ is limited by the redundancy $(L_{\rm crc})$ introduced by error-detection codes. When CRC is long (large $L_{\rm crc}$), $N_{\rm opt}$ may take a value as small as one, which means that there is no benefit from subpacket schemes due to too much redundancy introduced by CRC. When $L_{\rm crc}$ is very small, $N_{\rm opt}$ becomes large. Thus, a small $L_{\rm crc}$ is required to render a subpacket scheme advantageous. However, for CRC codes, $L_{\rm crc}$ should be large enough to ensure the detection capability. Therefore, to balance both CRC detection capability and the subpacket scheme, an appropriate value of $L_{\rm crc}$ should be chosen. Second, when $L_{\rm crc}$ and $P_w(1)$ are given, $N_{\rm opt}$ increases with L_{cb}, which means more benefit from large-N subpacket schemes in long packet-length systems. Third, a lower bound of $P_w(1)$ can be obtained from (11) by setting $N_{\rm opt} = 1$, and is given by

$$\underline{P_w} = \frac{L_{\rm crc}}{L_{cb}}. \tag{12}$$

Thus, a longer packet length L_{cb} can result in a smaller $\underline{P_w}$, which will increase the effective SNR$_b$ region of subpacket schemes [see (7)]. In the SNR$_b$ region where $P_w(1) < \underline{P_w}$, $N_{\rm opt}$ is set to one, which means that complete-packet schemes are the best packet schemes in the region, instead of subpacket schemes.

Given $L_{\rm crc}$ and L_{cb}, the optimum subpacket scheme can be constructed as follows. First, a PER versus SNR$_b$ curve is obtained from the tangential sphere bound (or simulation). Then $\underline{P_w}$ can be found from the curve, and the corresponding SNR$_b$ is taken as a threshold. For SNR$_b$ where $P_w(1) \le \underline{P_w}$, $N_{\rm opt}$ should be set to one, whereas for smaller SNR$_b$ where $P_w(1) > \underline{P_w}$, $N_{\rm opt}$ is calculated using (11).

The proposed method of constructing adaptive subpacket schemes is verified with convolutional codes. Consider a convolutional code with $L_f = 2406$, $L_{cb} = 1200$, $m_c = 4$, and code rate of 1/2, whose generator polynomial is (64, 74). The tangential sphere bound of PER is obtained, and $\underline{P_w}$ is calculated from (12) with the corresponding SNR$_b$ of about 5.0 dB. Therefore, when SNR$_b$ is smaller than 5.0 dB, the optimum $N_{\rm opt}$ can be found by using (11), whereas when SNR$_b \ge 5$ dB, $N_{\rm opt} = 1$. The adaptive subpacket scheme (or $N_{\rm opt}$) can be derived. Simulation results are also depicted in Fig. 3 for comparison. It is seen that analytical $N_{\rm opt}$ derived from the tangential sphere bound is close to the simulated $N_{\rm opt}$ obtained from simulations at low or high SNR$_b$. However, at moderate SNR$_b$, it is larger than the simulated $N_{\rm opt}$. This is because when SNR$_b$ is low or high, the tangential sphere bound provides good estimation for PER, while at moderate SNR$_b$, this upper bound is much larger than the simulated PER. It is seen from Fig. 3 that the optimum subpacket scheme is not a constant-N scheme. Instead, $N_{\rm opt}$ changes with SNR$_b$. The scheme that enables a system to employ different values of $N_{\rm opt}$ under variable SNR$_b$ is called "*an adaptive subpacket scheme*." Using a predefined table of $N_{\rm opt}$ at different SNR$_b$, the adaptive subpacket scheme can be easily realized.

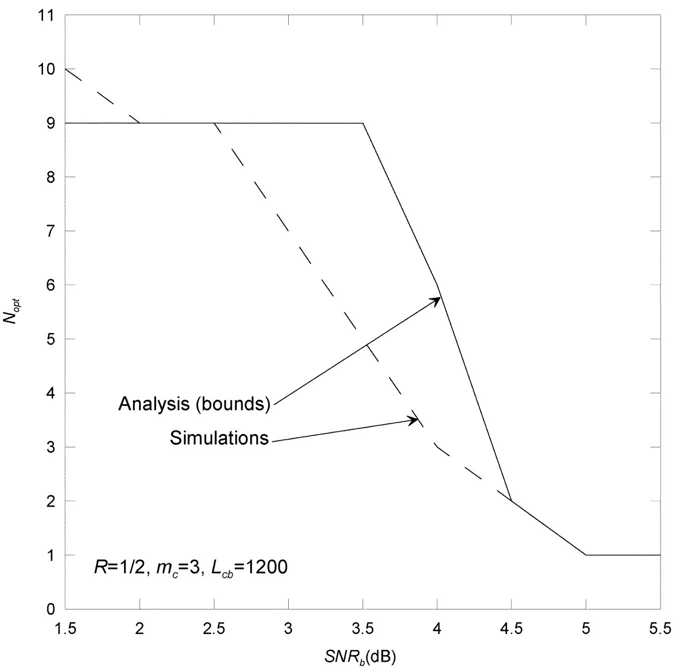

Fig. 3. Optimum number of subpackets in the adaptive subpacket schemes.

Since at moderate SNR$_b$, $N_{\rm opt}$ derived from the tangential sphere bound provides a rough estimation for the real $N_{\rm opt}$, it is necessary to investigate the system throughput with the adaptive optimum subpacket scheme. The throughput of the complete packet scheme obtained from simulation is also plotted for comparison. Note that the optimum number of subpackets is dependent on the SNR$_b$. It can be seen from Fig. 4 that when SNR$_b$ is small or moderate (SNR $_b \le 3$ dB), the adaptive subpacket scheme can significantly improve the system throughput, compared with the complete packet scheme. Analytical and simulation results are very close, even at moderate SNR$_b$, where analytical $N_{\rm opt}$ is larger than simulated $N_{\rm opt}$ (see Fig. 3). That is, performance degradation is small when $N_{\rm opt}$ is slightly overestimated.

Simulated throughput is illustrated in Fig. 5 with fixed-N and adaptive subpacket schemes. It is seen that the adaptive subpacket scheme always provides the highest throughput, whereas fixed-N subpacket schemes provide the same throughput as adaptive only in narrow ranges of SNR$_b$. Specifically, when SNR$_b$ is from 2 to 3 dB, or from 3 to 4 dB, or from 4 to 5 dB, the optimum number of subpackets are about $N_{\rm opt} = 8, 4, 2$, respectively. When SNR$_b \ge 5$ dB, the complete packet scheme $(N_{\rm opt} = 1)$ performs the best. This is consistent with Fig. 3.

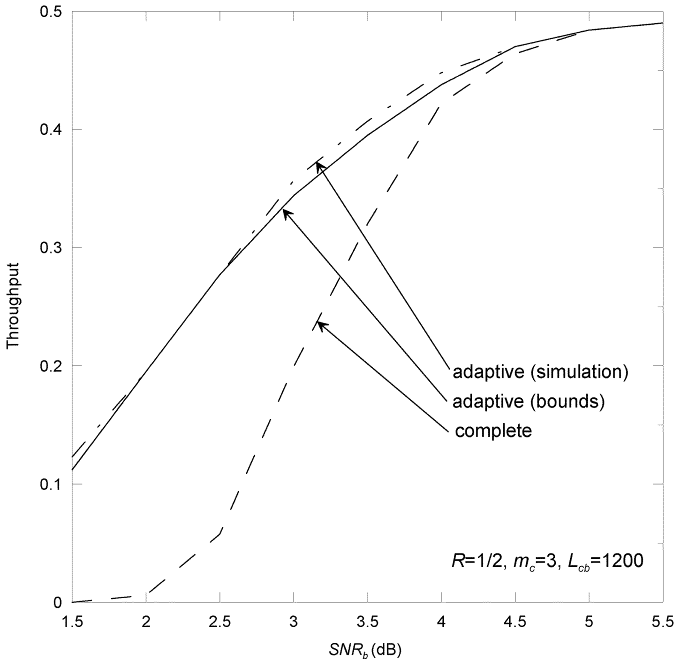

Fig. 4. Throughput comparison of simulation and analytical results.

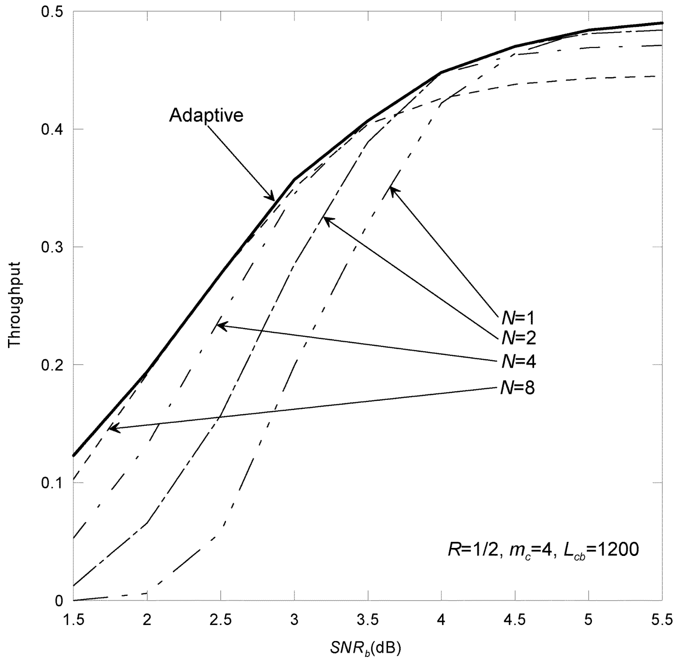

Fig. 5. Throughput of fixed subpacket and adaptive subpacket schemes.

In summary, for convolutionally coded hybrid ARQ systems, the adaptive subpacket scheme improves the throughput significantly.

The performance of adaptive subpacket schemes will be further investigated with different code rates and constraint lengths. As mentioned in the previous section, it is complicated to obtain the distance spectra of convolutional codes with punctuation and long constraint lengths. In order to provide the performance comparison of the adaptive subpacket and complete packet schemes in more detail, the following results are obtained by means of simulations.

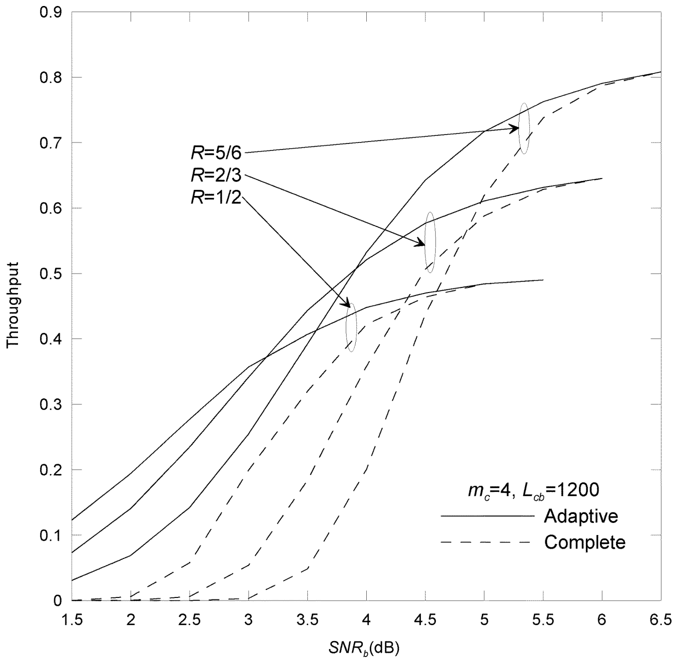

Fig. 6. System throughput of adaptive subpacket and complete packet schemes for different code rates.

Fig. 6 illustrates the system throughput as a function of SNR_b for different code rates generated by puncturing. It can be seen that for a given scheme (adaptive or complete), the convolutional code with code rate of 1/2 provides the highest throughput at small SNR_b, while the code with code rate of 5/6 performs best at high SNR_b. For a given code rate, although adaptive and complete schemes perform the same when SNR_b is large, the former performs much better than the latter when SNR_b is small or moderate. It is also interesting to note that in the SNR_b range of interest, the throughput of the 5/6-rate code with the adaptive scheme is always higher than that of the 1/2- or 2/3-rate code with the complete scheme. Although the SPER of the 5/6-rate code is higher than the complete PER of the 1/2-rate code, the former code still provides better throughput than the latter one, due to the higher code rate. In summary, for the adaptive scheme, the best code rates should be 1/2, 2/3, and 5/6 for SNR_b less than 3 dB, from 3 to 4 dB, and greater than 4 dB, respectively.

In order to investigate the effect of packet length on the optimum subpacket scheme, Fig. 7 shows the throughput (η) as a function of the number of subpackets (N) for $L_{cb} = 1200$, 1800, respectively. It is seen that each curve takes a peak value at a specific value of N, which is $N_{\mathrm{opt}} = 9, 12$ for $L_{cb} = 1200$, 1800, respectively. N_{opt} is roughly proportional to L_{cb}. Although the peak values of the two curves are very close, the curve of $L_{cb} = 1800$ is more robust to N. Therefore, a large value of L_{cb} is preferable.

B. Delay

Assuming that T is the time needed to encode, transmit, and decode a subblock of data, the system delay is T plus the additional delay caused by retransmission. For $N = 1$, i.e., in complete packet schemes, if the packet is error-free, then the system delay is T. If the packet is in error, then additional time is

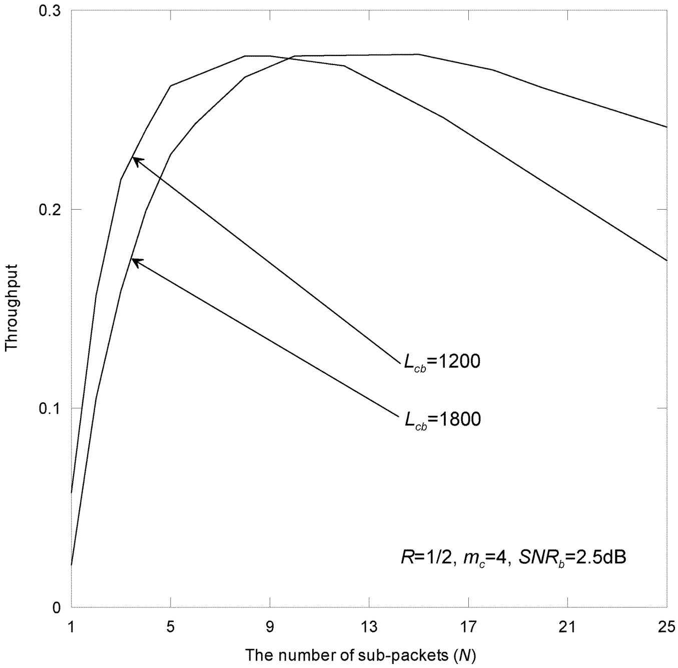

Fig. 7. System throughput as a function of the number of subpackets for different packet lengths.

needed for retransmission. Assuming that feedback delay is negligible, the average additional system delay for complete packet schemes is given by

$$\text{Delay} = T\cdot P_{\rm sp}(1)+T\cdot P_{\rm sp}(1)^2+\cdots+T\cdot P_{\rm sp}(1)^{M-1}. \quad (13)$$

When $N > 1$, situations are quite different. For the received subpackets SP_1 to SP_N, if the first subpacket SP_1 is in error, then the other subpackets must wait for it even if they are error-free, because data blocks must be sent to the upper-layer protocol in sequence. Hence, the system delay also depends on the position of the erroneous subpacket. Consider a simple case of $M = 2$, where all new subpackets are transmitted. At the receiver side, if the first subpacket is in error, then all N subpackets must wait for another T before being sent to the upper layer or dropped, and the average additional delay is $P_{\rm sp}(N)\cdot T$. If the first subpacket is error-free and the second is in error, then the first is sent to the upper layer, and the other $N-1$ subpackets wait in the buffer for another T. Hence, the average additional delay is $[1-P_{\rm sp}(N)]\cdot P_{\rm sp}(N)\cdot(N-1)/N\cdot T$. Similarly, when the first i subpackets are error-free and the $(i+1)$th is in error, the average additional delay is $[1-P_{\rm sp}(N)]^i\cdot P_{\rm sp}(N)\cdot(N-i)/N\cdot T$. Then, the additional delay caused by retransmission for N-subpacket

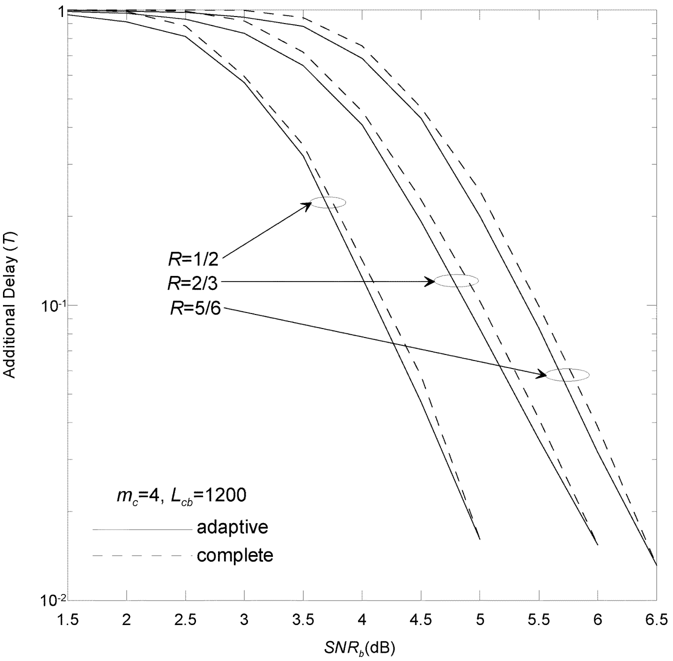

Fig. 8. Additional delay in adaptive subpacket and complete packet schemes.

schemes is given by (14), shown at the bottom of the page. When $N = 1$, (14) reduces to (13).

Fig. 8 illustrates the additional delay of complete and adaptive schemes for convolutionally coded systems. For all considered code rates, the adaptive subpacket scheme always provides less additional delay than the complete packet scheme, although the improvement is relatively small. In this figure, L_{cb} is set to 1200, and the packet length varies with code rates. Note that, in fact, T should change with packet lengths. However, the same T is used in this figure, since the focus is on the performance difference between complete packet schemes and adaptive subpacket schemes for each code rate.

C. Dropping Rate

When a subpacket is unsuccessfully transmitted M times, it will be dropped out. Therefore, in a truncated ARQ system with M maximum transmissions, the dropping rate of a subpacket is simply a power function of $P_{\rm sp}(N)$

$$P_{\rm drop} = \underbrace{P_{\rm sp}(N)\cdot P_{\rm sp}(N)\cdots P_{\rm sp}(N)}_{M} = P_{\rm sp}^M(N). \quad (15)$$

For $M = 2$, the dropping rates of complete packet and adaptive subpacket schemes are shown in Fig. 9. It can be seen that for a given code rate, the dropping rate of the adaptive subpacket

$$\begin{aligned}\text{Delay} &= \sum_{\rm i=0}^{N-1}[1-P_{\rm sp}(N)]^i\cdot P_{\rm sp}(N)\cdot\frac{N-i}{N}\cdot T\\ &= T\cdot\left(P_{\rm sp}(N)+\frac{[1-P_{\rm sp}(N)]^{N+1}-[1-N\cdot P_{\rm sp}(N)]\cdot[1-P_{\rm sp}(N)]}{N\cdot P_{\rm sp}(N)}\right)\end{aligned} \quad (14)$$

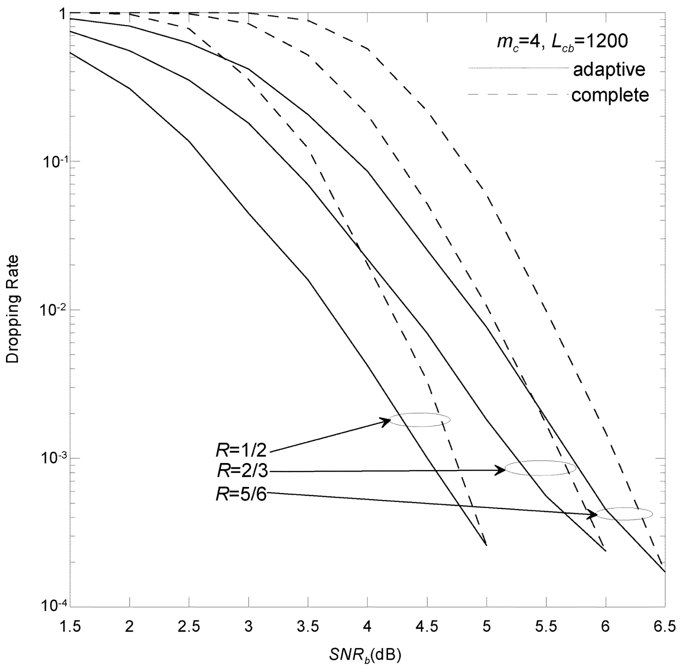

Fig. 9. Dropping rate in adaptive subpacket and complete packet schemes.

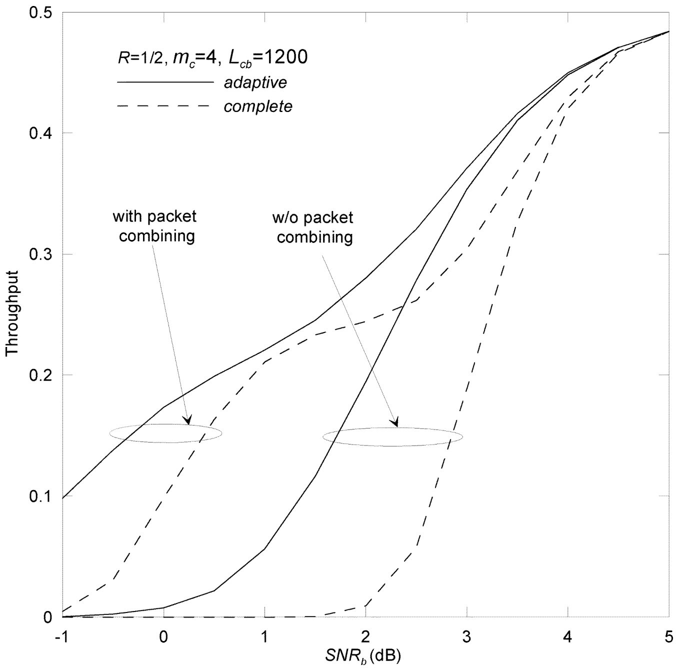

Fig. 10. Performance of adaptive subpacket schemes with packet combining.

is much smaller than that of the complete packet, especially when SNR_b is moderate.

D. Subpacket Combining

By combining a previously received erroneous packet and a new packet, the system throughput can be improved significantly [19], [20]. The adaptive subpacket scheme can be used with packet combining to further improve the system performance. It is assumed that totally $M = 2$ transmissions are permitted. At the first transmission, the data signals are transmitted using the optimum subpacket scheme according to the channel condition. At the receiver, all subpackets are stored, and NAK is sent to the transmitter for retransmission of the erroneous subpackets. The received signals of the previous erroneous subpackets and the corresponding subpackets in the second transmission are combined, and the whole codeword is decoded again. If the channel is static during the two transmissions, then the combining is simply the half sum of the two received signals. In the Gaussian channel, the SNR of the combined signal is improved by 3 dB, and the SPER can be reduced significantly. If the subpacket is still in error after packet combining, it will be dropped.

Fig. 10 shows the throughput performance of the adaptive scheme with and without subpacket combining. As a comparison, the performance of the complete scheme with and without packet combining is also shown. It can be seen that for either the adaptive or complete scheme, the system throughput is enhanced considerably by the combining algorithm, especially at low SNR. With the combining algorithm, the adaptive scheme still outperforms the complete scheme, but the performance gap is reduced when SNR_b is around 1 dB. This is because when SNR_b is small, the SPER and PER of the first transmission is high, and the system throughput is mainly determined by the SPER and PER after combining. Hence, due to two transmissions, the throughput is roughly equal to the half of the throughput without combining by shifting the SNR by -3 dB. Therefore, with combining, the performance gap between the adaptive and complete schemes reduces to a small value as SNR_b increases. However, when SNR_b is further increased beyond 1 dB, the throughput in the first transmission increases for the adaptive scheme, while it is still very small for the complete scheme. Therefore, after combining, the throughput gap between the two schemes increases as SNR_b increases from 1 dB. Then, when SNR_b is moderate or high, after combining, the packet is error-free in most cases. Hence, the throughput is mainly determined by the SPER and PER at the first transmission, and the throughput difference between the two schemes reduces as SNR_b is high.

It is interesting to note that although the adaptive scheme without combining is inferior to the complete scheme with combining at low SNR, the former outperforms the latter when $\mathrm{SNR}_b > 2$ dB. As explained before, when SNR_b is moderate or high, the throughput after combining is mainly determined by the PER of the complete scheme at the first transmission. Since the SPER of the adaptive scheme is much lower than the PER of the complete scheme when $\mathrm{SNR}_b > 2$ dB, it is noticeable that the adaptive scheme without combining has higher throughput than the complete scheme with combining. In summary, the adaptive subpacket scheme can be flexibly used with the packet-combining technique, and still outperforms the complete scheme with combining.

V. Conclusion

This paper presents an optimum subpacket scheme for convolutionally coded ARQ systems. By means of theoretical analysis and computer simulations, we show the following.

1) Subpacket schemes are suitable for convolutionally coded ARQ systems at low-to-moderate SNR_b. Significant improvement in the system throughput is obtainable with optimum subpacket schemes.
2) The optimum number of subpackets N_{opt} can be derived under the assumption of burst packet-error patterns. N_{opt} decreases with increasing SNR_b. Adaptive subpacket schemes provide the best performance in dynamic channels (variable SNR_b).
3) The optimum subpacket scheme outperforms complete packet schemes in the additional delay and dropping rate.
4) The optimum subpacket scheme can be flexibly used with the packet-combining technique to further improve the system throughput.

References

[1] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[2] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[3] K. R. Narayanan and G. L. Stuber, "A novel ARQ technique using the turbo coding principle," *IEEE Commun. Lett.*, vol. 1, no. 3, pp. 49–51, Mar. 1997.

[4] H. Yamamoto and K. Itoh, "Viterbi decoding algorithm for convolutional codes with repeat request," *IEEE Trans. Inf. Theory*, vol. IT-26, no. 9, pp. 540–547, Sep. 1980.

[5] B. A. Harvey and S. B. Wicker, "Packet combining systems based on the Viterbi decoder," *IEEE Trans. Commun.*, vol. 42, no. 4, pp. 1544–1557, Apr. 1994.

[6] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes (1)," in *Proc. Int. Conf. Commun.*, May 1993, pp. 1064–1070.

[7] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 4, pp. 260–269, Apr. 1967.

[8] K. Higuchi, T. Ikeda, S. Fukumoto, M. Sawahashi, and F. Adachi, "Experimental evaluations of high rate data transmission using turbo/convolutional coding in W-CDMA mobile radio," in *Proc. 12th Int. Conf. Wireless Commun.*, Jul. 2000, pp. 480–486.

[9] *Multiplexing and Channel Coding (FDD)*, 3rd Generation Partnership Project, Tech. Spec. Group Radio Access Network, ETSI TS 25212 V3.5.0 (Rel. 1999), 2000.

[10] N. Guo, F. Khaleghi, A. Gutierrez, J. Li, and M.-H. Fong, "Transmission of high speed data in CDMA2000," in *Proc. Wireless Commun. Netw. Conf.*, 1999, vol. 3, pp. 1442–1445.

[11] J. Hagenauer, "Rate compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

[12] H. Herzberg and G. Poltyrev, "The error probability of M-ary PSK block coded modulation schemes," *IEEE Trans. Commun.*, vol. 44, no. 4, pp. 427–433, Apr. 1996.

[13] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes via their spectra," *IEEE Trans. Inf. Theory*, vol. 40, no. 7, pp. 1282–1292, Jul. 1994.

[14] I. Sason and S. Shamai, "Improved upper bounds on the ML decoding error probability of parallel and serial concatenated turbo codes via their ensemble distance spectrum," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 24–47, Jan. 2000.

[15] L. R. Bahl, C. D. Cullum, W. D. Frazer, and F. Jelinek, "An efficient algorithm for computing the free distance," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 11, pp. 437–439, Nov. 1972.

[16] K. L. Larsen, "Comments on "An efficient algorithm for computing the free distance" by Bahl," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 7, pp. 577–579, Jul. 1973.

[17] M. Cedervall and R. Johannesson, "A fast algorithm for computing distance spectrum of convolutional codes," *IEEE Trans. Inf. Theory*, vol. 35, no. 11, pp. 1146–1159, Nov. 1989.

[18] R. J. McEliece, "How to compute weight enumerators for convolutional codes," in *Communication and Coding*, M. Darnell and B. Honary, Eds. Taunton/Somerset, U.K.: Research Studies Press, 1998, pp. 121–141.

[19] D. Chase, "Code combining—a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets," *IEEE Trans. Commun.*, vol. COM-33, no. 5, pp. 385–393, May 1985.

[20] B. A. Harvey and S. B. Wicker, "Packet combining systems based on the Viterbi decoder," in *Proc. IEEE MILCOM*, Oct. 1992, pp. 757–762.

Yiqing Zhou (M'04) was born in Zhejiang Province, China, in 1975. She received the B.S. degree in communication and information engineering and the M.S. degree in signal and information processing from Southeast University, Nanjing, China, in 1997 and 2000, respectively. In February 2004, she received the Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Hong Kong.

Since June 2004, she has been with the Department of Electrical and Electronic Engineering, University of Hong Kong, as a Postdoctoral Fellow. Her research interests include coding theory, spread spectrum, OFDM systems, interference cancellation, hybrid ARQ, and other transmission techniques for wireless high-speed data communications.

Jiangzhou Wang (M'91–SM'94) received the B.S. and M.S. degrees from Xidian University, Xian, China, in 1983 and 1985, respectively, and the Ph.D. degree (with Greatest Distinction) from the University of Ghent, Ghent, Belgium, in 1990, all in electrical engineering.

He is currently a Professor and a Chair in the Department of Electronics, University of Kent, Canterbury, U.K. From 1995 to 2005, he was with the University of Hong Kong, Hong Kong, where he is still serving as an Honorary Professor. From 1992 to 1995, he was a Senior System Engineer with Rockwell International Corporation (now Conexant), CA. From 1990 to 1992, he was a Postdoctoral Fellow with the University of California at San Diego, San Diego, CA. He has held a Visiting Professor position with NTT DoCoMo, Japan. He has published over 140 papers, including 42 IEEE TRANSACTIONS/JOURNAL papers in the areas of wireless mobile and spread spectrum communications. He has written/edited two books, *Broadband Wireless Communications* (Boston, MA: Kluwer, 2001) and *Advances in 3G Enhanced Technologies for Wireless Communications* (Norwood, MA: Artech House, 2002). He holds one U.S. patent in the GSM system.

Dr. Wang is an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS and a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in 2001 and 2006. He is listed in *Who's Who in the World* (New Providence, NJ: Marquis Who's Who). He was a Technical Chairman of the IEEE Workshop in 3G Mobile Communications in December, 2000.