Joint Design of Fixed-Rate Source Codes and Multiresolution Channel Codes

Andrea J. Goldsmith, *Member, IEEE*, and Michelle Effros, *Member, IEEE*

***Abstract*—We propose three new design algorithms for jointly optimizing source and channel codes. Our optimality criterion is to minimize the average end-to-end distortion. For a given channel SNR and transmission rate, our joint source and channel code designs achieve an optimal allocation of bits between the source and channel coders. Our three techniques include a source-optimized channel code, a channel-optimized source code, and an iterative descent technique combining the design strategies of the other two codes. The joint designs use channel-optimized vector quantization (COVQ) for the source code and rate-compatible punctured convolutional (RCPC) coding for the channel code. The optimal bit allocation reduces distortion by up to 6 dB over suboptimal allocations and by up to 4 dB relative to standard COVQ for the source data set considered. We find that all three code designs have roughly the same performance when their bit allocations are optimized. This result follows from the fact that at the optimal bit allocation the channel code removes most of the channel errors, in which case the three design techniques are roughly equivalent. We also compare the robustness of the three techniques to channel mismatch. We conclude the paper by relaxing the fixed transmission rate constraint and jointly optimizing the transmission rate, source code, and channel code.**

***Index Terms*—Joint source/channel coding, optimal bit allocation, RCPC channel code, vector quantization.**

I. INTRODUCTION

AN END-TO-END communication system is composed of a system encoder, which maps the source symbols into channel inputs, and a system decoder, which maps the channel outputs into noisy reproductions of the original source symbols. The system encoder can be further broken down into a source encoder, which maps the source symbols into an intermediate alphabet, typically a set of binary strings, and a channel encoder, which maps the binary strings into coded bits or waveforms for transmission over the channel. Similarly, the system decoder can be broken down into a channel decoder and a source decoder corresponding to the respective channel and source encoders. Any system encoder–decoder pair can be represented in this manner, although the breakdown is not unique.

Shannon's classical separation result states that we can optimize the end-to-end system design by separately optimizing the source encoder–decoder pair and the channel encoder-decoder pair [1]. However, this result holds only in the limit of infinite source code dimension and infinite channel code block length. Shannon theory does not provide a design algorithm for good channel codes with finite block length. In addition, Shannon theory does not address the design of good source codes when the probability of channel error is nonzero, which is inevitable for finite-length channel codes. Thus, for practical systems, a joint source and channel code design may reduce distortion, as well as complexity and delay.

Previous work in the area of joint source and channel coding falls into several broad categories: source-optimized channel coding or modulation, channel-optimized source coding, and iterative algorithms, which combine these two code designs. In source-optimized channel coding, the source code is designed for a noiseless channel. A channel code is then designed for this source code to minimize end-to-end distortion over the given channel, which is typically a binary symmetric channel (BSC), an additive white Gaussian noise (AWGN) channel with a given modulation, or a time-varying channel. Modestino and Daut provide an early treatment of source-optimized channel coding using differential pulse code modulation (DPCM) for image coding followed by convolutional channel coding [7]. They conclude that near optimal performance can be achieved using equal-error-protection channel codes. They also indicate that more flexible code rates are needed for source and channel code designs, leading to subsequent work using rate-compatible punctured convolutional (RCPC) channel codes. In [8], an RCPC channel code is matched to a subband source code for images. The optimal bit allocation and corresponding RCPC channel code are determined by an exhaustive search over all possible channel code choices. A similar technique is used in [9], where a tree-structured vector quantizer (TSVQ) designed for a noiseless channel is followed by an RCPC channel code, with the bit allocation and channel code optimized via an exhaustive search. Subband speech coding followed by a matched RCPC channel code designed for a Rayleigh fading channel is studied in [10]. A more comprehensive treatment of matching RCPC channel codes or multilevel quadrature amplitude modulation (MQAM) to DPCM, subband, and linear predictive speech coding (LPC), in both AWGN and Rayleigh fading channels, can be found in [11]. This study also determines the effect of different bit allocations between the source and channel codes. More recently, Sherwood and Zeger describe a source-optimized channel code for the BSC in which outputs of a progressive

Paper approved by E. Ayanoglu, the Editor for Communication Theory and Coding Applications of the IEEE Communications Society. Manuscript received July 27, 1997; revised February 13, 1998. The work of A. J. Goldsmith was supported by the Office of Naval Research under Grant NAV-5X-N149510861 and by the National Science Foundation under CAREER Award NCR-9501452. The work of M. Effros was supported by the National Science Foundation under CAREER Award MIP-9501977 and by a grant from the Powell Foundation and donations from Intel's 2000 program. This paper was presented in part at the IEEE International Communications Conference, Montreal, P.Q., Canada, June 1997.

The authors are with the Department of Electrical Engineering, M/C 136-93, California Institute of Technology, Pasadena, CA 91125 USA.

Publisher Item Identifier S 0090-6778(98)07770-8.

Said-Pearlman image coder are packetized into fixed-length packets, protected by a fixed number of check-sum bits, and then followed by RCPC channel coding [12].

In source-optimized modulation, the source code is designed for a noiseless channel and then the modulation is optimized to minimize end-to-end distortion. Two recent papers address source-optimized modulation through an energy allocation strategy. In [13], a vector quantizer (VQ) is followed by multicarrier modulation, where the modulation provides unequal error protection to the different source bits by assigning different powers to each subcarrier. This work indicates that using modulation to obtain unequal error protection provides significant performance improvement. The use of energy allocation for unequal error protection is also examined in [14], where unequal error protection for the output bits of a multistage VQ is provided via a time-varying energy allocation.

Channel-optimized source coding is another approach to joint source and channel coding. In this scenario, the source code maps the set of source symbols (scalars or vectors from the source alphabet) to binary strings for transmission over the channel. In fixed-rate source coding, the effect of a channel error is the receipt of an erroneous fixed-length binary string at the source decoder, which may cause the string to be decoded as an incorrect codeword. One way to achieve some degree of robustness to channel errors in such a setting (without allocating any bits to error protection or modifying the VQ codewords) is by careful assignment of the binary description (or index) associated with each codeword. Intuitively, the proper index assignment is the index assignment for which similar codewords have similar binary descriptions. In this case, the increase in distortion caused by likely codeword errors is kept small. Papers treating the index assignment problem include [15]–[17].

An alternate approach to the same problem is to redesign the source codebook to take into account the index crossover probabilities imposed by a noisy channel. Examples of work taking this approach include the channel-optimized vector quantizer (COVQ) and its scalar variation [2], [3], [18]. COVQ is a vector quantizer that has been optimized for a given set of crossover probabilities of the source codeword indices [2], [3]. These codeword indices are generally mapped to binary strings, and the crossover probabilities are then functions of the channel's bit error probability. The COVQ design involves iteratively optimizing the encoding regions (source encoder) and the codewords (source decoder) for the given set of index crossover probabilities. The result of this process is a locally optimal encoding rule for mapping source vectors to fixed-length binary indices and a corresponding decoding rule for mapping binary indices to their associated codewords, which may or may not be unique. Thus, inherent in the COVQ design algorithm is a block channel code design with hard-decision decoding. It is interesting to note that while index assignment would be unnecessary if the COVQ achieved the globally optimal source code design [3], attention to the index assignment problem in practical COVQ design does yield some performance gains by finding a better (local) optimum [19].

Source-optimized channel coding and modulation can be combined with channel-optimized source coding using an iterative design. In [20], a variation of the generalized Lloyd algorithm is used to iteratively design a COVQ encoder, a modulation signal set, and a linear decoder. A similar approach for the joint design of a COVQ and multicarrier modulation appears in [21]. Neither of these approaches uses channel coding. In contrast, [17] uses an iterative technique to jointly design a multistage VQ and trellis-coded modulation. In this work, the VQ is designed for a noiseless channel and then combined with index optimization. The effect of channel errors is assumed negligible, so the bit allocation between the source and channel codes is implicitly chosen to force a small channel bit-error-probability.

Other iterative design approaches to joint source and channel coding use trellis-coded quantization (TCQ), a form of source coding motivated by Ungerboeck's formulation of trellis-coded modulation (TCM) [22]. Ayanoğlu and Gray propose an iterative design for the encoder and decoder of a trellis-based source code, assuming a fixed channel bit-error-probability [23]. A subsequent paper [24] combines TCQ and TCM by first designing the TCQ for a noiseless channel, then matching the trellis of the TCM to that of the TCQ. This technique is somewhat similar in spirit to index assignment. A more powerful approach jointly optimizes the TCQ and TCM designs [25]. Simulated annealing and asymmetrical signal sets are added to a TCQ/TCM design in [26]. Most of the performance results for TCQ/TCM show performance gains relative to COVQ. However, this may be due to the power of the TCM alone. Since TCQ is a suboptimal source code, it is not clear that combining TCQ and TCM leads to better performance than, for example, combining COVQ with TCM.

The three design techniques that we propose include a source-optimized channel code, a channel-optimized source code and a joint iterative design of a source-optimized channel code, and a channel-optimized source code. The source-optimized channel code and channel-optimized source code use the traditional approaches of channel coders and source coders, respectively, with the addition of optimal bit allocation. The iterative design is a locally optimal technique combining the two traditional approaches along with optimal bit allocation. We consider all three approaches to: 1) determine how optimal bit allocation affects the performance of each technique; 2) compare the performance of the traditional approaches; and 3) determine the performance gain of a truly optimal joint code design.

The iterative joint code design uses COVQ for the source code and RCPC coding for the channel code.[1] Since the design of COVQ includes an inherent block channel code with hard-decision decoding, it should be clarified why additional channel coding may be beneficial. First of all, hard-decision decoding imposes an approximate 2–dB channel code performance penalty relative to soft-decision decoding. In addition, COVQ complexity grows exponentially with block length, which precludes the use of large block lengths for the implicit

[1] The COVQ is a full-search VQ rather than a TSVQ. While TSVQ's designed for noiseless channels are sometimes more robust to channel errors than are similarly designed VQ's, this advantage is lost when the channel statistics are incorporated in the source code design through index or codeword optimization.

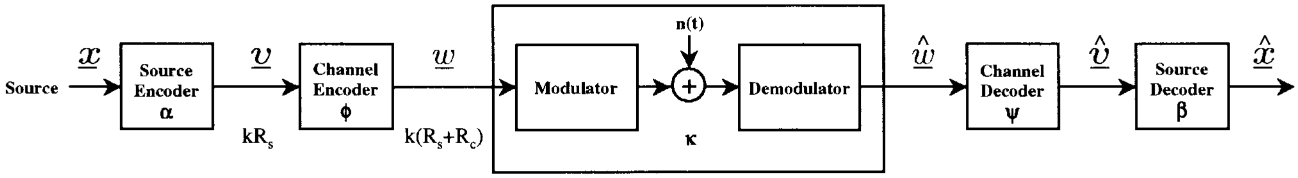


Fig. 1. System model.

COVQ channel code. Block channel codes may also exhibit worse performance than more powerful channel codes such as convolutional codes, trellis codes, and turbo codes, depending on the code rate and the complexity constraints. Since it is not clear how to modify COVQ to inherently design more powerful types of channel codes or to include soft-decision decoding, we combine COVQ with a separate channel code and optimize the bit allocation between these two codes. Since bit errors in different locations of a binary string cause different amounts of distortion, it is desirable for the channel code to provide different levels of error protection for different bits, which can be done using a multiresolution (unequal-error-protection) channel code. The design of the multiresolution channel code should be matched to the COVQ design to minimize distortion. Multiresolution channel coding can be implemented using multiplexed convolutional or block codes, trellis codes [4], or RCPC codes [5]. We use RCPC codes for our channel code, although our design technique is easily extended to other forms of multiresolution channel coding. We can also generalize the technique to any fixed-rate source code that can be optimized to a multiresolution channel code or, equivalently, to a set of channel bit-error-probabilities.

The designs of the COVQ and RCPC codes are not independent. The optimal COVQ is the COVQ matched to the index crossover probabilities determined by the RCPC channel code. Likewise, the optimal RCPC code is the RCPC code that minimizes the expected distortion of the COVQ. Ideally, the COVQ and the RCPC should be designed simultaneously. However, this simultaneous design is difficult to do in practice, leading to our three sequential design techniques. The first technique is a channel-optimized source code: COVQ optimized for a given equal-error-protection convolutional channel code. The second technique is a source-optimized channel code: RCPC channel coding optimized for a VQ design based on a noiseless channel. The third technique is an iterative algorithm combining the design strategies of the previous two. All of our joint design algorithms surpass the performance of their corresponding predecessors by optimizing the bit allocation between the source and channel codes for the given signal-to-noise ratio (SNR). The iterative design algorithm goes a step beyond, achieving a joint optimization of the source and channel codes using an iterative descent technique reminiscent of the generalized Lloyd algorithm [6] and guaranteeing convergence to a locally optimal code.

All three of our proposed joint codes exhibit roughly the same distortion performance on the data set considered. The codes differ, however, in their complexity and their robustness to channel estimation errors. Our codes exhibit a performance gain of roughly 4 dB relative to standard COVQ and more than 6 dB of gain relative to joint code designs with suboptimal bit allocations. We also find that for the optimized bit allocation, essentially all of the end-to-end distortion is contributed by the source code. This explains why the distortions of the three code designs are approximately the same: at the optimal bit allocation the channel appears approximately noiseless and therefore all the joint code designs are roughly equivalent.

BPSK modulation is used for transmission of the joint code designs over the channel. Further distortion reduction can be achieved by optimizing the modulation to match the source and channel codes. We consider this effect by optimizing the channel transmission rate (BPSK symbol time) jointly with the source and channel codes. In our experiments, a suboptimal choice of transmission rate increases distortion by up to 4 dB.

We see in our performance results under channel mismatch that a code designed for a channel with high SNR performs quite poorly when the SNR decreases. This indicates that our joint code design may not perform well in fading channels, where the SNR varies over time. However, we can easily modify our joint code design to improve its performance in fading. The modification will depend on whether or not an accurate estimate of the channel fade level can be made available to the transmitter. If so, then we can divide the fading range into a finite number of disjoint sets and design a joint source and channel code optimized for each set. When the fading channel SNR is estimated to be in a given set, we transmit the joint code design optimized for that set. If an accurate estimate of the channel fading is not available at the transmitter then we can use the same code design as for an AWGN channel with the distortion measure averaged over the fading distribution. This will likely lead to a higher allocation of bits to the channel encoder for error protection.

The remainder of this paper is organized as follows. The end-to-end system model is presented in Section II. The COVQ, RCPC channel codes, and our joint code design algorithms are described in Section III. Experimental results appear in Section IV, where we also determine the distortion reduction obtained by optimizing the channel transmission rate along with the joint code design. Section V summarizes our results and conclusions.

II. System Model

A block diagram of the end-to-end communication system is shown in Fig. 1. We assume a discrete-time, real-valued, stationary source. The source encoder maps each possible k-dimensional source vector $\underline{x}$ to a binary string $\underline{v}$. An example of a source vector would be a block of k pixels from an image. The binary strings can be fixed-length or variable-length, corresponding to a fixed-rate or variable-rate source code. Due to rate constraints, the source encoder typically introduces some distortion. While variable-rate source codes achieve better compression for a given distortion and source vector dimension, they typically perform poorly when channel errors are introduced, due to synchronization errors [27]. For this reason we consider only fixed-rate vector quantizers.

The fixed-rate vector quantizer produces fixed-length binary strings $\underline{v}$ of length kR_s for each source vector $\underline{x}$, where R_s is a parameter of the source code design. The channel encoder operates on the vector $\underline{v}$ of kR_s bits to obtain $k(R_s+R_c)$ coded bits $\underline{w}$. These coded bits (channel symbols) are then modulated and transmitted sequentially over an AWGN channel. We use BPSK modulation at a rate of one channel symbol per T seconds, where T is the channel symbol time. The SNR per channel symbol is $E_s/N_0 = PT/N_0$, where P is the received signal power and N_0 is the spectral noise density. After demodulation, the received channel symbol vector $\hat{\underline{w}}$ may differ from the transmitted vector $\underline{w}$ due to channel noise. We denote the mapping of channel symbols through the BPSK modulator, AWGN channel, and demodulator by κ, which is parameterized by E_s/N_0. These noisy channel symbols pass through the channel decoder, which may correct some (or all) of the channel errors to obtain a noisy reproduction $\hat{\underline{v}}$ of the original binary string $\underline{v}$. Finally, $\hat{\underline{v}}$ is passed through the source decoder to obtain a noisy reproduction $\hat{\underline{x}}$ of the original source vector $\underline{x}$.

We measure the total end-to-end distortion of the system as the squared-error between the source vector $\underline{x}$ and source reproduction vector $\hat{\underline{x}}$: $d(\underline{x}, \hat{\underline{x}}) = \| \underline{x} - \hat{\underline{x}} \|^2$. For a given R_s, the total distortion decreases as R_c increases. Similarly, for a given R_c, the total distortion decreases as R_s increases. For $R = R_s + R_c$ fixed, there is an optimal way to divide the transmission rate R between the source and channel rates R_s and R_c to minimize total distortion. This optimal allocation depends on the SNR per channel symbol E_s/N_0 and the source statistics. In particular the R_s value corresponding to the optimal bit allocation will generally increase as a function of E_s/N_0.

The information rate I_r of the system is defined as the number of source symbols transmitted per second. If the channel symbol rate is $1/T$ channel symbols per second, then the source symbol rate $I_r = [T(R_s + R_c)]^{-1}$ source symbols per second. Thus, $R = R_s + R_c$ is constrained by the required source symbol rate and the channel symbol time. The source symbol rate I_r is generally fixed. We initially assume that the modulation is also fixed, so T and therefore R are fixed parameters. We later allow T to vary, which gives one more degree of freedom in the joint source and channel code design.

III. Joint Code Design Algorithms

In this section we describe our three joint source and channel code design algorithms: a source-optimized channel code using a VQ and an RCPC channel code (VQ-RCPC), a channel-optimized source code using COVQ and an equal-error-protection (equal-weight) convolutional channel code (COVQ-EW), and a joint iterative design of a COVQ and an RCPC channel code (COVQ-RCPC). For all three code designs, we optimize the bit allocation between the source and channel codes. The joint iterative design will always have an expected distortion at least as low as the expected distortion of the other two techniques, since the codes resulting from these other design techniques are considered during the joint iterative design process and improved through this process when possible. We therefore begin by describing the joint iterative design algorithm. We will then describe the other design techniques, which are subsets of this algorithm.

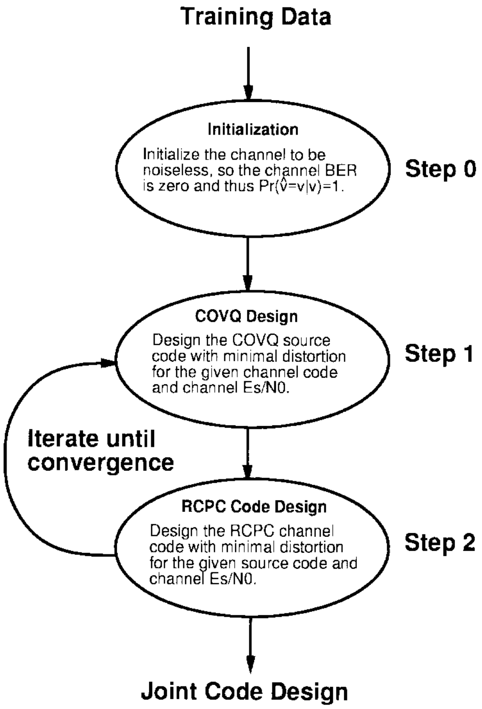

Training Data
Initialization
Initialize the channel to be noiseless, so the channel BER is zero and thus Pr(v̂=v|v)=1.
Step 0
COVQ Design
Design the COVQ source code with minimal distortion for the given channel code and channel Es/N0.
Step 1
Iterate until convergence
RCPC Code Design
Design the RCPC channel code with minimal distortion for the given source code and channel Es/N0.
Step 2
Joint Code Design

Fig. 2. Iterative code design process (R_s fixed).

A. Joint Iterative Design (COVQ-RCPC)

The goal of our joint code design is to minimize the expected distortion $D = E[d(\underline{X}, \hat{\underline{X}})]$ of the COVQ and RCPC codes, where the expectation is taken with respect to a source data training set and the channel noise.[2] We use an iterative design technique to obtain this minimization. Specifically, for a given R_s value, we alternately optimize the COVQ for the existing RCPC code and then the RCPC code for the new COVQ. The flow chart for our design algorithm is shown in Fig. 2. This iterative approach is guaranteed to converge to a locally optimal solution since each design step gives a global optimum. The design process is repeated for each R_s value in the range $0 \le R_s \le R$, where R is the channel transmission rate per source symbol. For $R_s = 0$, the source is represented by a single codeword, so there is no channel distortion and the total distortion corresponds to the source variance. For $R_s = R$, all redundant bits are allocated to the source code, so there is no channel coding. Intermediate values of R_s correspond to a redundancy tradeoff in the source and channel codes. The joint code design corresponding to the

[2] For fading channels, the expected distortion can also be averaged over the fading distribution.

(R_s, R_c) pair with minimal distortion is chosen as the final joint code design. We now describe each of the algorithm steps in Fig. 2.

1. Step 0—Initialization:

We initialize the system such that $\Pr(\hat{v} = v|v) = 1$. This initialization is equivalent to assuming a noiseless channel, so that the bit-error-probability for any RCPC channel code is zero.

2. Step 1—COVQ Design:

In Step 1 of the design algorithm, the COVQ (α, β) is optimized for the bit-error-probability of the given RCPC channel code. These probabilities result either from the initialization of Step 0 or from the RCPC channel code (ϕ, ψ) found in Step 2 and the channel E_s/N_0. The optimal α and β are obtained through the COVQ design algorithm described in [3]. For any fixed RCPC channel code, this design technique is itself an iterative algorithm which successively redesigns α for a given β and β for a given α. We now describe the optimization process in greater detail.

The optimal COVQ (α, β) is the COVQ that minimizes the expected distortion $D = E[d(\underline{X}, \hat{\underline{X}})]$ between a random source input vector $\underline{X}$ and its reproduction $\hat{\underline{X}}$ at the receiver. Expanding D as a function of a fixed source and channel code $(\alpha, \phi, \kappa, \psi, \beta)$ gives

$$\begin{aligned} D &= Ed(\underline{X}, \beta(\psi(\kappa(\phi(\alpha(\underline{X})))))) \\ &= E_x \sum_{\underline{\hat{v}}} \Pr(\underline{\hat{v}}|\alpha(\underline{X}))d(\underline{X}, \beta(\underline{\hat{v}})) \end{aligned} \tag{1}$$

where E_x is the expectation with respect to the source distribution and

$$\Pr(\underline{\hat{v}}|\underline{v}) = \Pr(\psi(\kappa(\phi(\underline{v}))) = \underline{\hat{v}}) \tag{2}$$

is the probability that channel errors cause the binary string $\underline{v}$ output by the source encoder to be received as $\underline{\hat{v}}$ by the source decoder. We call $\{\Pr(\underline{\hat{v}}|\underline{v})\}$ the index crossover probabilities, which are functions of the RCPC channel code. In particular, given an RCPC code, modulation, and channel E_s/N_0, we can find the bit-error-probability P_j of the jth bit in $\underline{v}$ and then expand the index crossover probability as

$$\Pr(\underline{\hat{v}}|\underline{v}) = \prod_{j=1}^{kR_s} \left[P_j 1(\hat{v}_j \neq v_j) + (1 - P_j)1(\hat{v}_j = v_j) \right] \tag{3}$$

where $1(\cdot)$ denotes the indicator function and $\hat{v}_j$ and v_j denote the jth elements of the vectors $\underline{\hat{v}}$ and $\underline{v}$, respectively. In theory, the expectation in (1) should be taken with respect to the true underlying source distribution. For the purpose of practical code design, the true distribution must be approximated by the empirical distribution of some finite representative training set.

The optimal source encoder α is the source encoder that minimizes for each $\underline{x}$ the expected distortion between $\underline{x}$ and its reproduction $\hat{\underline{x}}$ at the source decoder. Thus, given a fixed source decoder β and the index crossover probabilities $\{\Pr(\underline{\hat{v}}|\underline{v})\}$ associated with a fixed RCPC code, the optimal source encoder $\alpha^\star$ must satisfy

$$\alpha^\star(\underline{x}) = \arg\min_{\underline{v}} \sum_{\underline{\hat{v}} \in \{0,1\}^{kR_s}} \Pr(\underline{\hat{v}}|\underline{v})d(\underline{x}, \beta(\underline{\hat{v}})). \tag{4}$$

Associated with the source encoder $\alpha^\star(\underline{x})$ are a set of encoding regions $\{S_{\underline{v}}\colon \underline{v} \in \{0, 1\}^{kR_s}\}$, where $S_{\underline{v}} = \{\underline{x}\colon \alpha(\underline{x}) = \underline{v}\}$. Notice that the optimal source encoder is no longer the traditional VQ nearest neighbor encoder: the binary string to which $\alpha^\star$ maps a given $\underline{x}$ vector may not be the one with the closest reproduction $\beta(\cdot)$, due to the fact that the binary string may be corrupted in transmission.

For a given source encoder α and crossover probabilities $\{\Pr(\underline{\hat{v}}|\underline{v})\}$ of a fixed RCPC channel code, the optimal source decoder $\beta^\star$ must have codewords that satisfy

$$\begin{aligned} \beta^\star(\underline{\hat{v}}) &= \arg\min_{\hat{\underline{x}}} E[\Pr(\underline{\hat{v}}|\alpha(\underline{X}))d(\underline{X}, \hat{\underline{x}})] \\ &= E_{\underline{x}}[\underline{X}|\psi(\kappa(\phi(\alpha(\underline{X})))) = \underline{\hat{v}}] \\ &= \frac{\sum_{\underline{v}} \Pr(\underline{v}) \Pr(\underline{\hat{v}}|\underline{v}) E_x[\underline{X}|\alpha(\underline{X}) = \underline{v}]}{\sum_{\underline{v}} \Pr(\underline{v}) \Pr(\underline{\hat{v}}|\underline{v})} \end{aligned} \tag{5}$$

for each $\underline{\hat{v}} \in \{0, 1\}^{kR_s}$, where $\Pr(\underline{v}) = \Pr(\underline{x}\colon \underline{x} \in S_{\underline{v}})$ and we have assumed the squared error distortion measure. Just as the optimal encoding regions have changed due to the channel errors, the optimal codewords of the decoder design are no longer at the centroids of their decoding cells. Instead, each codeword is a weighted average of the centroids of all of the encoding cells. Specifically, in finding the codeword $\beta(\underline{\hat{v}})$ associated with the codeword index $\underline{\hat{v}}$, we weight the centroid of the encoding region $S_{\underline{v}}$ by the probability that the received vector $\underline{\hat{v}}$ came from transmitting $\underline{v}$ through the channel.

Nonetheless, the encoding region centroids will be extremely useful to us in optimizing our joint code. For each $\underline{v} \in \{0, 1\}^{kR_s}$ define the centroid $\gamma(\underline{v})$ of $S_{\underline{v}}$ by

$$\gamma(\underline{v}) = E[\underline{X}|\underline{x} \in S_{\underline{v}}]. \tag{6}$$

Then, by (1), we may rewrite the expected distortion associated with a given source code (α, β) as

$$\begin{aligned} &E[d(\underline{X}, \hat{\underline{X}})] \\ &\quad = E_x \left[\sum_{\underline{\hat{v}} \in \{0,1\}^{kR_s}} \Pr(\underline{\hat{v}}|\alpha(\underline{X})) ||\underline{X} - \beta(\underline{\hat{v}})||^2 \right] \\ &\quad = \sum_{\underline{v}, \underline{\hat{v}} \in \{0,1\}^{kR_s}} \Pr(\underline{v}) \Pr(\underline{\hat{v}}|\underline{v}) E[||\underline{X} - \beta(\underline{\hat{v}})||^2 | \alpha(\underline{X}) = \underline{v}] \\ &\quad = \sum_{\underline{v}, \underline{\hat{v}} \in \{0,1\}^{kR_s}} \Pr(\underline{v}) \Pr(\underline{\hat{v}}|\underline{v}) \\ &\qquad \cdot \left[E[||\underline{X} - \gamma(\underline{v})||^2 | \alpha(\underline{X}) = \underline{v}] + ||\gamma(\underline{v}) - \beta(\underline{\hat{v}})||^2 \right] \\ &\quad = E_x[||\underline{X} - \gamma(\alpha(\underline{X}))||^2] \\ &\qquad + \sum_{\underline{v}, \underline{\hat{v}} \in \{0,1\}^{kR_s}} \Pr(\underline{v}) \Pr(\underline{\hat{v}}|\underline{v}) ||\gamma(\underline{v}) - \beta(\underline{\hat{v}})||^2 \\ &\quad = E_x[||\underline{X} - \gamma(\alpha(\underline{X}))||^2] + E_{\underline{v}, \underline{\hat{v}}}[||\gamma(\underline{v}) - \beta(\underline{\hat{v}})||^2] \end{aligned} \tag{7}$$

where $E_{\underline{v}, \underline{\hat{v}}}$ denotes the expectation with respect to the joint distribution $\Pr(\underline{v}, \underline{\hat{v}}) = \Pr(\underline{v}) \Pr(\underline{\hat{v}}|\underline{v})$. Equation (7) will be useful in designing the optimal RCPC code in Section III-A-3.

The optimal design algorithm is an iterative descent technique analogous to the generalized Lloyd algorithm. Each iteration proceeds through the following two steps:

i) *Optimize each encoder* for the given decoder and RCPC channel code. The optimal encoder is given by (4).

ii) *Optimize each decoder* for the given encoder and RCPC channel code. The optimal decoder is given by (5).

Notice that while the algorithm as a whole guarantees only a locally optimal (α, β), each of the above steps finds a globally optimal solution. In the first step, we find a globally optimal encoder for the given decoder and RCPC channel code, and in the second step, we find a globally optimal decoder for the given encoder and RCPC channel code.

3. Step 2—RCPC Channel Code Design:

In Step 2 of the algorithm, the RCPC code with minimal distortion for the COVQ designed in Step 1 is found. We use the set of channel code rates from Table I in [5] to obtain our RCPC code candidates. Each candidate RCPC code provides a set of kR_s bit-error probabilities for the kR_s bits in $\underline{v}$. We represent each candidate code by a vector of length kR_s, where the jth vector element corresponds to the channel code rate applied to the jth bit in $\underline{v}$. For example, for $kR_s = 4$, the RCPC code (1/2, 1/4, 1, 1) applies a rate 1/2 convolutional code to the first bit in $\underline{v}$, a rate 1/4 convolutional code to the second bit in $\underline{v}$, and no channel code to the third and fourth bits in $\underline{v}$. Associated with the convolutional code applied to the jth bit is a bit-error-probability P_j which depends on the modulation and the channel E_s/N_0. A plot is shown in [5, Fig. 5] of bit-error-probability versus E_s/N_0 for each element of an RCPC channel code vector under our BPSK modulation assumption.

The minimal-distortion RCPC code is obtained as follows. For a given COVQ and corresponding R_s value, we search over all RCPC code vectors that satisfy the transmission rate constraint $R = R_s + R_c$. For a given R_s, an RCPC code vector $(c_1, \cdots, c_{kR_s})$ satisfies this constraint if $\sum_{i=1}^{kR_s}(c_i)^{-1} \leq kR$. We call an RCPC code that satisfies this constraint an eligible channel code. Note that the channel code vector (1/2, 1/4, 1, 1) satisfies this constraint for $kR = 8$, as would the code vectors (1/2, 1/2, 1/2, 1/2) and (1/3, 1, 1, 1/3), along with many others. Since the COVQ does not order its codeword bits relative to their error sensitivity, we must search over all eligible channel codes so that the best protection levels under the transmission rate constraint R are applied to $\underline{v}$. The search for the optimal RCPC channel code must consider all eligible channel codes in the set of $(14)^{R_s}$ possible channel codes.[3]

The optimal RCPC code is the code with the error protection levels that minimize the expected distortion (7). The first term in (7) depends only on the source code and source training data set. Thus, while this term is expensive to calculate, it does not change as a function of the RCPC channel code. Therefore, our search for the optimal RCPC channel code need only find the RCPC code that minimizes the second term of (7):

$$E_{\underline{v},\,\hat{\underline{v}}}[||\gamma(\underline{v}) - \beta(\hat{\underline{v}})||^2]. \tag{8}$$

Equation (8) is a function of the index crossover probabilities $\{\Pr(\hat{\underline{v}}|\underline{v})\}$, which in turn depend on the bit-error-probabilities $\{P_j\colon j = 1, \cdots, kR_s\}$ of the RCPC code, as indicated in (3). However, (8) is not a function of the training set and therefore can be efficiently computed. After computing (8) for all eligible channel codes, we select the RCPC code with minimal distortion. This code choice corresponds to the initial channel code (ϕ, ψ) in Fig. 1.

4. Step 3—Design Iteration:

At the conclusion of Step 2, a new channel code (ϕ, ψ) has been designed for the COVQ obtained in Step 1. The iterative design process then returns to Step 1 to determine the COVQ (α, β) for this new RCPC channel code (ϕ, ψ). Once a new COVQ is obtained, this source code is passed to Step 2 of the algorithm to obtain a new RCPC channel code. Successive application of Steps 1 and 2 results in a sequence of source codes $\{(\alpha, \beta)\}$ and corresponding channel codes $\{(\phi, \psi)\}$ for which the average distortions (1) form a positive nonincreasing sequence which has to converge. At convergence, the source code $(\alpha, \beta)^*_{R_s}$ and channel code $(\phi, \psi)^*_{R_s}$ with minimal distortion for the given R_s value are obtained. The design process is repeated for each R_s value $0 \leq R_s \leq R$, and the source code $(\alpha, \beta)^*$ and channel code $(\phi, \psi)^*$ corresponding to the R_s value with minimal distortion comprise the final joint code.

B. Source-Optimized Channel Code Design (VQ-RCPC)

Our second joint design is a source-optimized channel code which uses a VQ designed for a noiseless channel followed by an RCPC channel code matched to the VQ to minimize distortion. This code differs from previous source-optimized channel codes in its use of an optimal bit allocation algorithm. For a given value of R_s this source and channel code is designed by following Steps 0–2 of Fig. 2 and then stopping, so the iterative design process is eliminated. This process is repeated for each R_s value, $0 \leq R_s \leq R$, and the joint source and channel code corresponding to the ++R_s value with minimal distortion is used for the final joint code. The advantage of this technique relative to the joint iterative design is its simplicity: the source code design is independent of all aspects of the channel code, modulation, and channel SNR. The disadvantages of this design will be discussed in Section IV.

C. Channel-Optimized Source Code Design (COVQ-EW)

Our last joint code design is a channel-optimized source code using COVQ and an equal-weight channel code. As with VQ-RCPC, COVQ-EW differs from earlier channel-optimized source codes in its use of an optimal bit allocation. The COVQ-EW design for each R_s value uses only Step 1 of Fig. 2. The channel code used in this design is restricted so that the bit-error probabilities $\{P_j\colon j = 1, \cdots, kR_s\}$ for all kR_s bits in $\underline{v}$ are the same. For each R_s value there is only one equal-weight channel code with maximal error protection, and this is the channel code used in the design. Note that for a given R_s value, the RCPC channel code in [5] is not a rich enough set to always use all of our channel code redundancy for equal-

[3] The search complexity is significantly reduced for source codes where the bits are ordered relative to their error sensitivity.

TABLE I
SQCNR COMPARISON (dB)

Code	E_s/N_0 0 dB	2 dB	4 dB
COVQ-RCPC	13.5	15.8	17.3
COVQ-EW	13.2	15.7	17.2
VQ-RCPC	13.2	15.7	17.2
COVQ	9.3	13.0	15.3
VQ-RCPC (10^{-6} Ch. BER)	10.0	13.7	16.5

weight channel coding. We therefore use interpolation to find the error probability corresponding to a channel code using the full redundancy. For example, if $kR_s = 3$ and $R = 8$ then we require a rate 3/8 channel code to use all the redundant bits. However, the RCPC code of [5] has a rate 4/10 code and a rate 4/11 code, but no rate 3/8 code. Therefore, in our numerical results, we assume a rate 3/8 code with error probability derived by interpolating between the error probabilities of the rate 4/10 code and the rate 4/11 code.[4] The design process is carried out for all R_s values, with the final code design based on the R_s value with minimal distortion.

IV. Experimental Results

The three joint source and channel code designs described in Section III were implemented in C^{++} and run for a range of channel E_s/N_0 values. Although the design processes are computationally complex,[5] they are done off-line. Thus, the code design complexity does not impact the system operation. For a channel with variable E_s/N_0, code designs corresponding to several E_s/N_0 values can be stored so that a code matched to the channel quality is always available.

Our experimental results are for a test data set of five magnetic resonance images (MRI's) described by the joint code designed from a training data set of 20 other MRI's. The expected distortion is computed with the expectation taken over the test data set and the channel noise. We define the ratio of signal power to quantization and channel noise power (SQCNR) as $\mathrm{SQCNR(dB)} = 10\log(\sigma^2/\mathrm{D})$, where D is the expected distortion (1) of the joint code and σ^2 is the expected distortion of a rate zero ($R_s = 0$) VQ for the same test data set. We use vector dimension $k = 4$ and consider $0 \leq R_s \leq 2$ bits per pixel (bpp) and $0 \leq E_s/N_0 \leq 4$ dB. We choose these ranges since, as we will see in the numerical results below, there is little distortion reduction for $R_s > 2$ bpp or $E_s/N_0 > 4$ dB.

Table I shows the performance of each of our three joint design techniques, along with the performance of standard COVQ (no additional channel coding). We also show the performance of a VQ followed by a matched RCPC channel code with R_s chosen to achieve an approximate channel BER of 10^{-6}, which is a common target BER for image transmission over noisy channels. Notice that the three joint design algorithms exhibit up to 4 dB of performance gain over COVQ and up to 3.5 dB of gain relative to VQ-RCPC with R_s chosen based on a target channel BER. Notice also that the performance of the three joint design techniques proposed in Section III is nearly the same, with the joint iterative design performing slightly better in all cases. To understand the similarity in performance of our joint code designs, we now investigate these three algorithms in more detail.

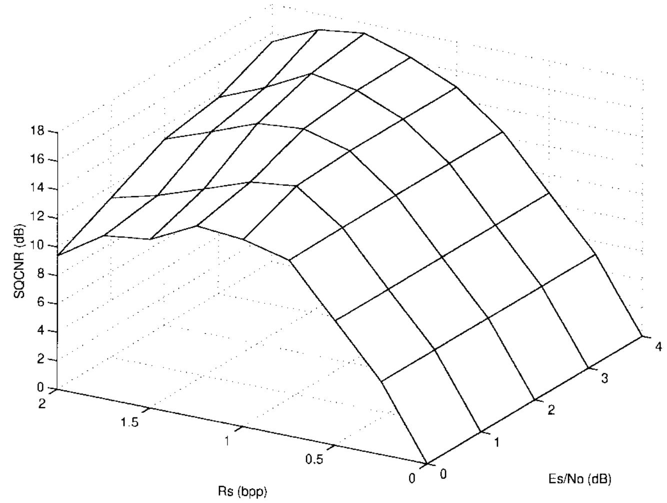


Fig. 3. SQCNR versus E_s/N_0 and R_s for COVQ-RCPC.

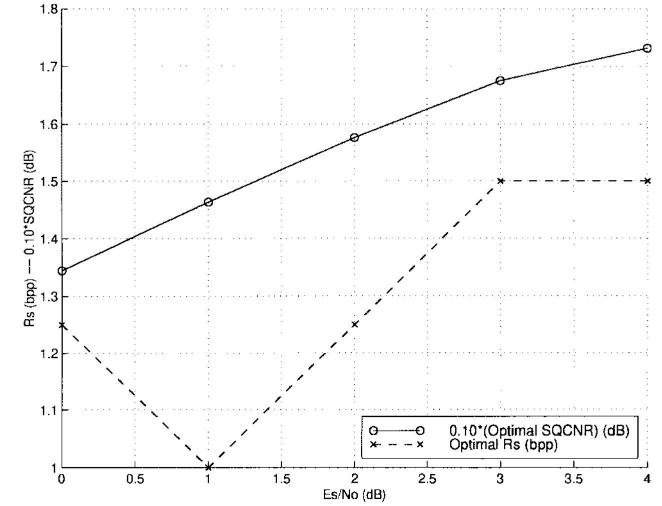
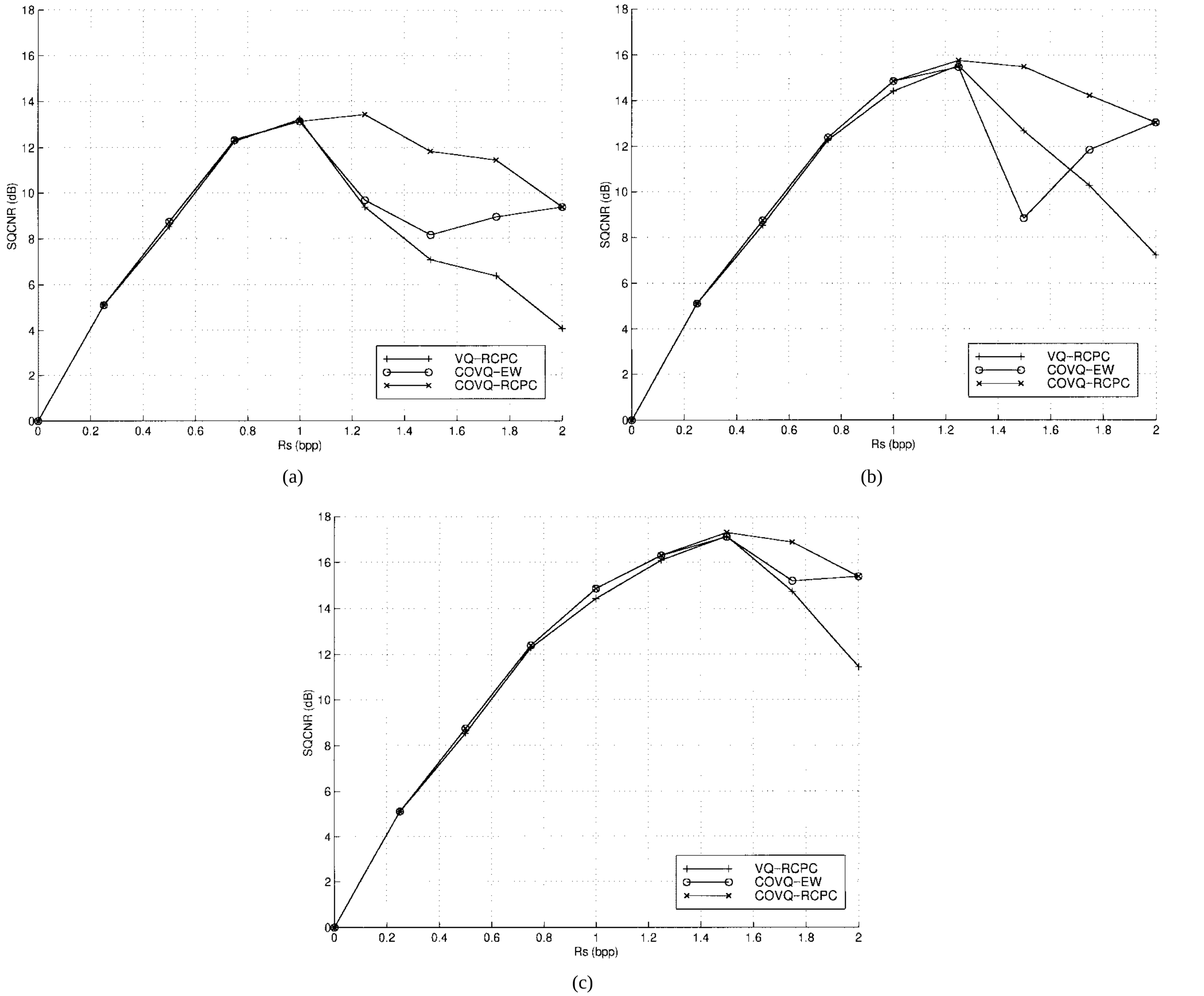


Fig. 4. SQCNR and optimal R_s versus E_s/N_0 for COVQ-RCPC.

In Fig. 3 we show a three-dimensional plot of SQCNR for COVQ-RCPC as a function of the bit allocation R_s and the channel E_s/N_0. We see from this figure that improper choice of R_s can reduce the SQCNR by more than 10 dB. As expected, the R_s value that maximizes SQCNR (minimizes distortion) increases as the channel E_s/N_0 increases, since fewer redundant bits are needed for channel coding. This trend is illustrated more clearly in Fig. 4, where we plot both the optimal R_s and the SQCNR of the COVQ-RCPC joint code as a function of E_s/N_0. An exception to this trend occurs at $E_s/N_0 = 1$ dB, where the optimal R_s value decreases. This exception is due to the behavior of RCPC codes at low E_s/N_0 values, where code rates in the RCPC code with one or two redundant bits exhibit a negative coding gain [5, Fig. 5]. We see that the SQCNR increases monotonically as E_s/N_0

[4] We can achieve this interpolated probability in practice by time-sharing between the rate 4/10 and rate 4/11 codes. Alternatively, we can use any rate 3/8 convolutional code with parameters chosen to yield roughly the same error probability as our interpolated code.

[5] Approximately five iterations are required for convergence of the COVQ-RCPC and COVQ-EW code designs.



(a)

(b)

(c)

Fig. 5. SQCNR versus R_s for all design techniques. (a) $E_s/N_0 = 0$ dB. (b) $E_s/N_0 = 2$ dB. (c) $E_s/N_0 = 4$ dB.

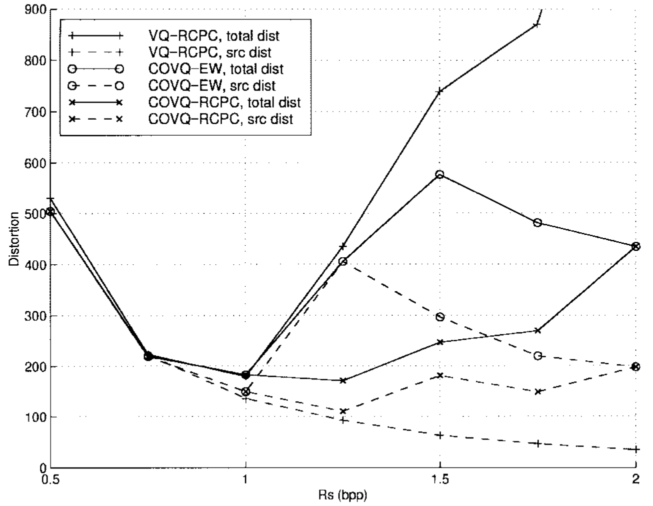
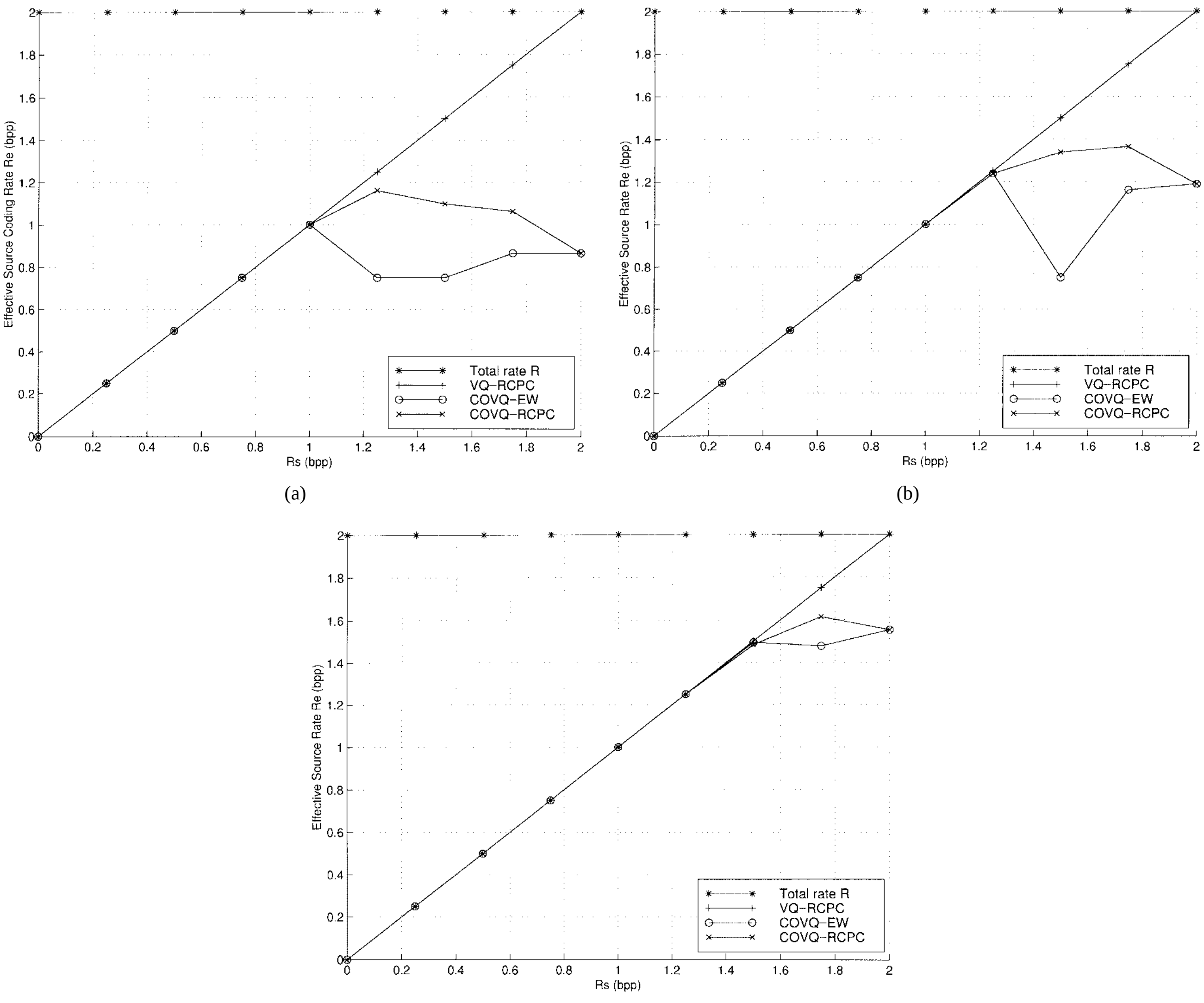


Fig. 6. Distortion versus R_s for all design techniques. $E_s/N_0 = 0$ dB.

increases, with most of the transmission rate R allocated to the source code at $E_s/N_0 = 4$ dB. Our other joint design techniques exhibit similar behavior.

A comparison of SQCNR versus R_s for all three of our joint code designs is shown in Fig. 5 for $E_s/N_0 = 0, 2,$ and 4 dB. Notice that the optimal bit allocation for all three codes is almost the same, with the optimal R_s value for COVQ-RCPC always larger than that of the other codes. This result is expected, since the COVQ-EW uses a less flexible channel code (equal-error-protection versus unequal-error-protection), so more channel code bits are needed to get the same level of error protection. For VQ-RCPC, the source encoder is designed with respect to a noiseless channel, so the encoded bits are more sensitive to channel errors, and therefore need stronger channel coding for a higher level of error protection. Fig. 5 confirms the observation of Table I that for this range of E_s/N_0 the SQCNR's for all three of our joint code designs are approximately equal. As is evident from Fig. 5, the most important aspect of a joint source and channel code design is choosing the appropriate bit allocation between the source and channel codes, yet it is precisely this step that is ignored in most previous joint source and channel code designs. For all of our joint design techniques the SQCNR can decrease by more than 10 dB over the entire range of R_s: $0 \leq R_s \leq 2$ bpp.



(a)

(b)

(c)

Fig. 7. Effective source code rate versus R_s for all design techniques. (a) $E_s/N_0 = 0$ dB. (b) $E_s/N_0 = 2$ dB. (c) $E_s/N_0 = 4$ dB.

However, some of these values (e.g., $R_s = 0$) would never be used in a real code design. For a reasonable allocation of (R_s, R_c), $0.5 \le R_s \le 1.5$, the SQCNR of the joint code designs decreases by more than 6 dB relative to the SQCNR at the optimal bit allocation.[6] Note that the SQCNR for the COVQ-EW and VQ-RCPC code designs falls off sharply as R_s increases above its optimal value, whereas COVQ-RCPC is less sensitive to a suboptimal R_s choice.

The distortion D at $E_s/N_0 = 0$ dB for each technique is shown by the solid lines in Fig. 6. The dashed lines in Fig. 6 show the distortion contribution of the source code alone, based on the test data set and a noiseless channel. Note that for all design techniques, most of the distortion at the optimal bit allocation (R_s, R_c) is contributed by the source code, even though $E_s/N_0 = 0$ dB is an extremely poor channel. Similar distortion results were obtained for $0 \le E_s/N_0 \le 4$ dB. This explains why the minimal distortions of all our joint code designs are roughly the same at all of the E_s/N_0 values. Specifically, for the optimal bit allocation the channel bit-error-probabilities $\{P_j\}$ of the optimal RCPC channel code are approximately zero. Therefore, since the source code of VQ-RCPC is designed for $P_j = 0$ for all j, this code design is approximately optimal. In addition, if the $\{P_j\}$ of the RCPC channel code are all approximately zero, then the same will be true for the equal-weight channel code in the COVQ-EW code design. Therefore, this joint source and channel code design is also approximately optimal.

As discussed in Section III-A-2, the optimal codewords $\{\beta(\underline{\hat{v}})\}$ associated with a COVQ will change as a function of the index crossover probabilities $\{\Pr(\underline{\hat{v}}|\underline{v})\}$. In particular, the codewords will tend to be closer together when the probability of an index error is high than when that same probability is low. For sufficiently high index error probabilities, the codewords will actually merge, meaning that two or more codeword indices (e.g., $\underline{\hat{v}_1}$ and $\underline{\hat{v}_2}$) will map to the same reproduction vector [e.g., $\beta(\underline{\hat{v}_1}) = \beta(\underline{\hat{v}_2})$]. A source code with two or more identical codewords effectively uses some of the source coding bits for redundancy or channel error

[6] This range of (R_s, R_c) allocations is taken from the values considered in [8].

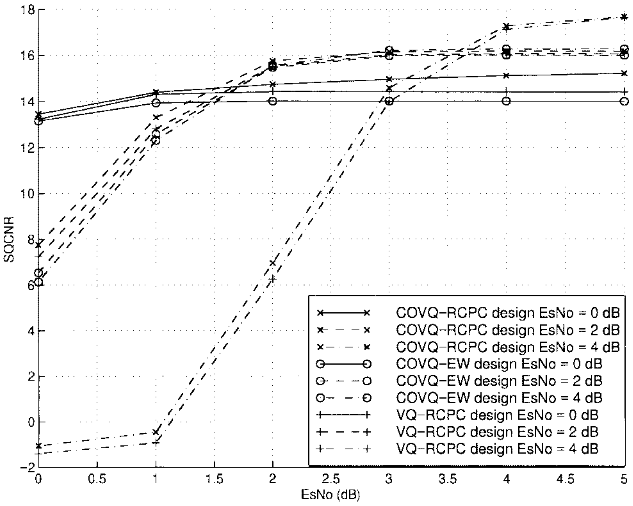


Fig. 8. Effect of channel mismatch.

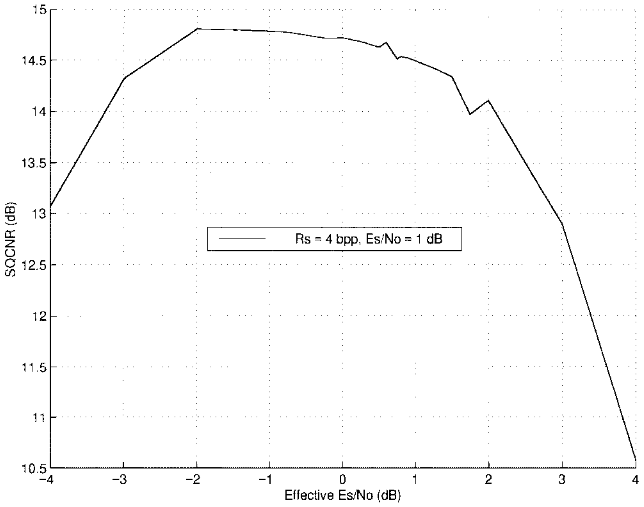


Fig. 9. SQCNR versus effective E_s/N_0.

protection. In this case we say that the effective source coding rate R_e is lower than the source coding rate R_s, since the source code applies some of its rate $R_s - R_e$ to the job of error protection. Therefore the total number of bits applied to channel error protection is $R_c + (R_s - R_e)$: we call this the effective channel coding rate. Given a COVQ with N distinct codewords, we will calculate the effective source coding rate as $R_e = \log N/k$. Fig. 7 shows the breakdown of the transmission rate R between the effective source coding rate R_e and the effective channel coding rate $R_c+(R_s-R_e) = R-R_e$ for all three of our joint design techniques. In the case of the VQ-RCPC code, the source code is designed under a noiseless channel assumption, and thus,, R_e is everywhere equal to R_s. However, for COVQ-RCPC and COVQ-EW as R_s increases the channel coding rate R_c decreases, and thus, the index crossover probabilities increase. This will eventually cause some of the source coding rate to be used for error protection so that, above some critical R_s value, R_e will drop below R_s, as we observe in the figure. In COVQ-EW, the channel coding rate R_c is not used to its maximum effect since we restrict the channel code to equal-error-protection, and thus, more of the source coding rate is applied to channel error protection than in the COVQ-RCPC code design.

By comparing Figs. 5 and 7 we notice that the optimal rate allocation (R_s, R_c) corresponds approximately to the point at which R_e diverges from R_s. This is because the soft-decision RCPC channel code is a more effective channel code than the hard-decision block channel code inherent to the COVQ design. Thus, the optimal rate allocation corresponds to the maximal R_s for which most or all of the source coding rate is being used for source coding ($R_s \approx R_e$). Therefore, for the COVQ-RCPC and COVQ-EW codes, rather than calculate the performance at each possible R_s value to find the optimal (R_s, R_c), we can perform the joint design at a mid-range value of R_s and then efficiently search up or down in R_s as appropriate to find the point at which the two curves diverge.

The optimal bit allocation between the source and channel codes and their respective designs depend on the channel E_s/N_o. Thus, when this value is estimated in error, the performance of the source and channel code will suffer. The effect of channel mismatch, where the source and channel code design is based on an incorrect channel E_s/N_o, is plotted in Fig. 8 for all three of our joint code designs. We see that the COVQ-EW code designed for $E_s/N_o = 4$ dB has much better performance at low E_s/N_o values than the COVQ-RCPC and VQ-RCPC codes. That is because some members of the RCPC code family in the latter two code designs exhibit negative coding gain at these low E_s/N_o values. The COVQ-EW will also exhibit negative coding gain under some mismatch conditions. Thus, the main conclusion to draw from this figure is that, for any of the code design techniques, it is best to design the source and channel code for a pessimistic estimate of the channel E_s/N_o in order to avoid obtaining a negative coding gain due to mismatch in the channel code design.

The joint source and channel code designs described in Section III assume that E_s/N_0 is a fixed parameter of the channel. We can vary the $E_s/N_0 = PT/N_0$ for a fixed transmit power P by changing the symbol time T to $\hat{T}$. However, to maintain the same source symbol rate $I_r = (TR)^{-1} = (\hat{T}\hat{R})^{-1}$ source symbols per second, we must also change the channel symbol rate from R to $\hat{R} = RT/\hat{T}$. In Fig. 9 we show the change in SQCNR for COVQ-RCPC when we vary the symbol rate $\hat{T}$ over a range of values or, equivalently, vary the *effective* $E_s/N_0 = P\hat{T}/N_0$. The calculations were done for $E_s/N_0 = 1$ dB. Reducing the symbol time ($\hat{T} < T$) causes a corresponding increase in R. This typically increases the SQCNR, since the extra bits available for the source and channel codes more than compensate for the decrease in effective E_s/N_0. However, if $\hat{T}$ is reduced too far below its optimal value then the SQCNR decreases dramatically. Increasing the symbol duration well above the optimal value also dramatically decreases SQCNR. Note that the SQCNR is not a smooth function. We attribute this to the fact that the number of eligible codes changes at discrete values of $\hat{T}$, whereas the effective E_s/N_0 is a continuous function of $\hat{T}$.

It is clear from the numerical results in this section that the choice of which joint code design to use should be based on criteria other than SQCNR, which is roughly the same for all three codes. Therefore the choice comes down to one of robustness versus complexity. For a given R_s value the COVQ-RCPC code design is more complex than the other two designs, although this design process is done off-line and does not impact system operation. The optimal choice of (R_s, R_c) is easier to determine for the COVQ-RCPC and COVQ-EW codes than for the VQ-RCPC code since it corresponds to the point at which the effective source code rate diverges from the actual source code rate. In terms of system operation, it is slightly easier to implement a single-rate equal-error-protection channel code than the multirate RCPC channel code. Finally, the COVQ-RCPC code design exhibits more robustness to a suboptimal bit allocation than the other techniques. Since it also has slightly better SQCNR in all cases, this code design seems to be the best choice.

V. Summary

We describe three joint source and channel code techniques to minimize end-to-end distortion and analyze their performance. The first technique, VQ-RCPC, is a VQ followed by an RCPC channel code where the channel code is designed relative to the VQ to minimize distortion. The second technique, COVQ-EW, uses COVQ and an equal-error-protection convolutional channel code with soft-decision decoding. The last technique, COVQ-RCPC, combines COVQ with an RCPC channel code in an iterative design procedure. All three code designs optimize the bit allocation between the source and channel coders. The optimal bit allocation reduces distortion by up to 6 dB relative to a suboptimal bit allocation. Our joint code designs also reduces distortion by up to 4 dB relative to standard COVQ.

The distortions of all three of our proposed code designs are approximately the same. That is because the optimal bit allocation for each code design results in a channel code which removes most of the channel errors, in which case all of the design procedures are roughly equivalent. thus, the most important aspect for any of these source and channel code designs is the optimal bit allocation between the source and channel coders. This optimal allocation depends on the source statistics, the channel quality, and the modulation. We determine an efficient method for finding the optimal bit allocation for the COVQ-RCPC and COVQ-EW codes without searching over all possible values.

Imperfect knowledge of the channel E_s/N_0 or the source statistics will generally degrade performance. Our experiments indicate that all three codes have roughly the same sensitivity to channel mismatch, and that COVQ-RCPC is less sensitive than the other coding techniques to a suboptimal bit allocation. This robustness is achieved at the expense of higher off-line design complexity. However, all three codes have approximately the same run-time complexity. Based on our performance results under channel mismatch, it is best to design the source and channel code for a pessimistic channel E_s/N_0 when this value is not precisely known.

Our distortion results indicate that multiresolution channel coding does not significantly reduce distortion relative to equal-error-protection channel coding when the bit allocation is optimized. We attribute this to the fact that, after optimizing the bit allocation between the source and channel codes, the bits in the encoded source vector have approximately the same sensitivity to channel errors. We are now investigating the use of two-stage source codes like the weighted universal vector quantizer in our design algorithm [28]. Since the codebook index (first stage), which is very sensitive to channel errors, is sent along with the codewords (second stage), we expect multiresolution channel coding to have more of an impact for this source code. We are also studying joint optimization of the source code, channel code, and modulation for AWGN channels and for fading channels.

Acknowledgment

The authors gratefully acknowledge the anonymous reviewers for their detailed critiques. Their comments and suggestions helped to greatly improve the clarity of the paper.

References

[1] C. E. Shannon, “Coding theorems for a discrete source with a fidelity criterion,” in *IRE Nat. Conv. Rec, Part 4,* 1959, pp. 142–163.

[2] H. Kumazawa, M. Kasahara, and T. Namekawa, “A construction of vector quantizers for noisy channels,” *Electron. Eng. in Japan,* vol. 64B, no. 4, pp. 39–47, 1984.

[3] N. Farvardin and V. Vaishampayan, “On the performance and complexity of channel-optimized vector quantizers,” *IEEE Trans. Inform. Theory,* vol. 37, pp. 155–160, Jan. 1991.

[4] A. R. Calderbank and N. Seshadri, “Multilevel codes for unequal error protection,” *IEEE Trans. Inform. Theory,* vol. 39, pp. 1234–1248, July 1993.

[5] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their applications,” *IEEE Trans. Commun.,* vol. 36, pp. 389–400, Apr. 1988.

[6] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* New York: Kluwer Academic, 1992.

[7] J. W. Modestino and D. G. Daut, “Combined source-channel coding of images,” *IEEE Trans. Commun.,* vol. COM-27, pp. 1644–1659, Nov. 1979.

[8] N. Tanabe and N. Farvardin, “Subband image coding using entropy-coded quantization over noisy channels,” *IEEE J. Select. Areas Commun.,* vol. 10, pp. 926–943, June 1992.

[9] H. Jafarkhani, P. Ligdas, and N. Farvardin, “Adaptive rate allocation in a joint source/channel coding framework for wireless channels,” in *Proc. IEEE VTC’96,* Apr. 1996, pp. 492–496.

[10] R. V. Cox, J. Hagenauer, N. Seshadri, and C.-E. W. Sundberg, “Variable rate sub-band speech coding and matched convolutional channel coding for mobile radio channels,” *IEEE Trans. Signal Processing,* vol. 39, pp. 1717–1731, Aug. 1991.

[11] W. C. Wong, R. Steele, and C.-E. W. Sundberg, *Source-Matched Mobile Communications.* London, U.K.: Pentech, New York: IEEE Press, 1995.

[12] P. G. Sherwood and K. Zeger, “Progressive image coding on noisy channels,” in *Proc. IEEE Data Compress. Conf.,* Mar. 1997, pp. 72–81.

[13] K.-P. Ho and J. M. Kahn, “Transmission of analog signals using multicarrier modulation: A combined source-channel coding approach,” *IEEE Trans. Commun.,* vol. 44, pp. 1432–1443, Nov. 1996.

[14] S. Gadkari and K. Rose, “Unequally protected multistage vector quantization for time-varying channels,” in *Proc. IEEE Int. Conf. Commun.,* pp. 786–790, June 1998.

[15] N. Farvardin, “A study of vector quantization for noisy channels,” *IEEE Trans. Inform. Theory,* vol. 36, pp. 799–809, July 1990.

[16] K. Zeger and A. Gersho, “Pseudo-Gray coding,” *IEEE Trans. Commun.,* vol. 38, pp. 2147–2158, Dec. 1990.

[17] S. Gadkari and K. Rose, “Vector quantization with transmission energy allocation for time-varying channels,” *IEEE Trans. Commun.,* submitted for publication.

[18] N. Farvardin and V. Vaishampayan, “Optimal quantizer design for noisy channels: An approach to combined source-channel coding,” *IEEE Trans. Inform. Theory,* vol. 33, pp. 827–838, Nov. 1987.

[19] M. Effros, “Robustness to channel variation in source coding for transmission across noisy channels,” in *Proc. IEEE ICASSP,* Apr. 1997, pp. 2961–2964.

[20] V. A. Vaishampayan and N. Farvardin, “Joint design of block source codes and modulation signal sets,” *IEEE Trans. Inform. Theory,* vol. 38, pp. 1230–1248, July 1992.

[21] K.-P. Ho and J. M. Kahn, “Combined source-channel coding using channel-optimized quantizer and multicarrier modulation,” in *Proc. IEEE ICC’96,* June 1996, pp. 1323–1327.

[22] G. Ungerboeck, “Channel coding with multi-level/phase signals,” *IEEE Trans. Inform. Theory,* vol. IT-28, pp. 55–67, Jan. 1982.

[23] E. Ayanoğlu and R. M. Gray, “The design of joint source and channel trellis waveform coders,” *IEEE Trans. Inform. Theory,* vol. IT–33, pp. 855–865, Nov. 1987.

[24] T. R. Fischer and M. W. Marcellin, “Joint trellis coded quantization/modulation,” *IEEE Trans. Commun.,* vol. 39, pp. 172–176, Feb. 1991.

[25] M. Wang and T. R. Fischer, “Trellis-coded quantization designed for noisy channels,” *IEEE Trans. Inform. Theory,* vol. 40, pp. 1792–1802, Nov. 1994.

[26] H. A. Aksu and M. Salehi, “Joint optimization of TCQ-TCM systems,” *IEEE Trans. Commun.*, vol. 44, pp. 529–533, May 1996.

[27] W. M. Lam and A. R. Reibman, “An error concealment algorithm for images subject to channel errors,” *IEEE Trans. Image Processing.,* vol. 4, pp. 533–542, May 1995.

[28] P. A. Chou, M. Effros, and R. M. Gray, “A vector quantization approach to universal noiseless coding and quantization,” *IEEE Trans. Inform. Theory,* vol. 42, pp. 1109–1138, July 1996.

Andrea J. Goldsmith (S’90–M’93) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California at Berkeley in 1986, 1991, and 1994, respectively.

From 1986 to 1990, she was affiliated with Maxim Technologies, where she worked on packet radio and satellite communication systems, and from 1991–1992, she was affiliated with AT&T Bell Laboratories, where she worked on microcell modeling and channel estimation. She joined the faculty of the California Institute of Technology, Pasadena, in 1994, where her current research includes work in wireless communication theory, joint source and channel coding, and resource allocation in cellular systems.

Dr. Goldsmith is a recipient of the National Science Foundation CAREER Development Award, a National Semiconductor Faculty Development Award, an IBM Graduate Fellowship, and the David Griep Memorial Prize from UC Berkeley. She is a member of IEEE Information Theory, IEEE Communications, and IEEE Vehicular Technology Societies, and an editor for the IEEE TRANSACTIONS ON COMMUNICATIONS and the *IEEE Personal Communications Magazine*.

Michelle Effros (S’93–M’95) received the B.S. degree with distinction in 1989, the M.S. degree in 1990, and the Ph.D. degree in 1994, all in electrical engineering from Stanford University, Stanford, CA.

During the summers of 1988 and 1989, she worked at Hughes Aircraft Company, researching modulation schemes, real time implementations of fast data rate error correction schemes, and future applications for fiber optics in space technology. Since 1994, she has been an Assistant Professor of Electrical Engineering at the California Institute of Technology. Her research interests include information theory, data compression, communications, pattern recognition, and image processing.

Dr. Effros received Stanford’s Frederick Emmons Terman Engineering Scholastic Award in 1989, the Hughes Masters Full-Study Fellowship in 1989, the National Science Foundation Graduate Fellowship in 1990, the AT&T Ph.D. Scholarship in 1993, the NSF CAREER Award in 1995, the Charles Lee Powell Foundation Award in 1997, and the Richard Feynman-Hughes Fellowship in 1997. She is a member of Tau Beta Pi, Phi Beta Kappa, Sigma Xi, and IEEE Signal Processing, IEEE Information Theory, and IEEE Communications Societies and is currently the Editor of the IEEE Information Theory Society Newsletter and a Member of the Board of Governors of the IEEE Information Theory Society.