Joint Source-Channel Decoding of Predictively and Nonpredictively Encoded Sources: A Two-Stage Estimation Approach

David J. Miller, *Member, IEEE*, Elias S. G. Carotti, *Student Member, IEEE*, Yu-Wei Wang, and Juan Carlos De Martin, *Member, IEEE*

***Abstract*—A common joint source-channel (JSC) decoder structure for predictively encoded sources involves first forming a JSC decoding estimate of the prediction residual and then feeding this estimate to a standard predictive decoding (synthesis) filter. In this paper, we demonstrate that in a JSC decoding context, use of this standard filter is suboptimal. In place of the standard filter, we choose the synthesis filter coefficients to give a least-squares (LS) estimate of the original source, based on given training data. For first-order differential pulse-code modulation, this yields as much as 0.65-dB gain in reconstructing first-order Gauss–Markov sources. More gains are achieved with modest additional complexity by increasing the filter order. While performance can also be enhanced by increasing the source's Markov model order and/or the decoder's lookup table memory, complexity grows exponentially in these parameters. For both predictive and nonpredictive coding, our LS approach offers a strategy for increasing the estimation accuracy of JSC decoders while retaining manageable complexity.**

***Index Terms*—Differential pulse code modulation (DPCM), joint source-channel (JSC) decoding, least-squares (LS) estimation, residual redundancy.**

I. Introduction

IN [21], for a differential pulse-code modulation (DPCM) system, a suboptimal predictor was used, introducing statistical dependencies within the sequence of quantization indexes output by the DPCM encoder. The authors proposed to exploit this redundancy at the decoder to improve error resilience in much the same way that controlled redundancy inserted via channel coding is exploited. They developed a source-channel decoder that finds the most likely sequence of transmitted indexes, given a sequence of noisy received indexes. Their technique requires that the decoder has access to a simple model for the quantized source (e.g., a first-order Markov model) as well as a channel model. It was demonstrated in [21] that significant error resilience could be achieved by this approach. Following this paper, there has been substantial further activity in *decoding based on residual redundancy*. While [21] maximized a sequence-likelihood criterion, [18] developed a decoder based on maximum *a posteriori* (MAP) sequence detection. Both approaches are suboptimal if the true objective is minimum mean-squared error (MMSE) estimation. MMSE decoding was also considered in [18], with a more general MMSE approach developed in [14] and [22]. While these papers focused on "1-D" signals, i.e., time series, several papers have developed extensions for digital images [8], [13], [16]. Some recent research has also focused on systems that use variable-length encoding (VLC). Approximate decoders for systems that use Huffman coding have been developed, e.g., [6] and [15]. Methods for introducing redundancy in arithmetic-coding systems have also been developed [4], [7], as well as decoding techniques for these systems, e.g., [17].

While some recent work has developed methods tailored for VLCs, there is another fundamental coding paradigm that has not, to date, been optimally accounted for by existing JSC decoding methods—*predictive coding*. The most basic predictive system, to be addressed in this paper, is DPCM. DPCM is generally not a competitive lossy compression system on its own. However, it is often used within more powerful coding frameworks, e.g., in transform coding of images, for encoding low-frequency coefficients. Moreover, the predictive quantization paradigm in DPCM is a fundamental one that appears in many practical systems and standards, e.g., motion-compensated prediction in video coding. Thus, techniques for enhancing the error resilience of DPCM may have general applicability to the applications (in image, video, and speech coding) where predictive quantization is routinely used. Error resilience for DPCM and more general predictive coding systems has been previously addressed in several different ways, e.g., by optimizing the predictor coefficients to enhance robustness [3], by optimizing the quantizer given knowledge of the noisy channel [1], by conventional forward-error correction techniques, e.g., [20], and by error-concealment techniques.

Decoding based on residual redundancy has been considered for DPCM systems in a number of papers [6], [8], [12], [13], [16], [21]. The predictive setting is arguably the most appropriate one for exploiting residual redundancy. The most common cause of residual redundancy is, in fact, suboptimal encoding of a source with memory. However, if the source has memory, the most appropriate lossy coding paradigm is predictive quantization. In all prior techniques exploiting residual redundancy for DPCM, excepting [12], the following decoding strategy was taken: form a (JSC decoding) estimate

Paper approved by M. Skoglund, the Editor for Source-Channel Coding of the IEEE Communications Society. Manuscript received August 12, 2003; revised January 9, 2004.

D. J. Miller is with the Electrical Engineering Department, Pennsylvania State University, University Park, PA 16802 USA (e-mail: millerdj@ee.psu.edu).

Y.-W. Wang is with Hewlett-Packard, Fort Collins, CO 80525 USA.

E. S. G. Carotti is with the Politecnico di Torino, 10129 Torino, Italy.

J. C. De Martin is with the IEIIT-CNR, Politecnico di Torino, 10129 Torino, Italy.

Digital Object Identifier 10.1109/TCOMM.2004.833064

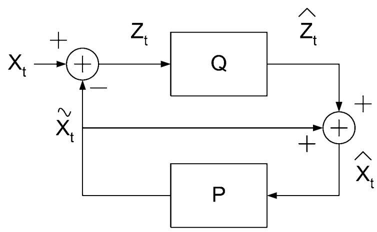


Fig. 1. DPCM encoder.

of the encoder's prediction residual, given the received (noisy) quantization indexes; and feed this estimate as input to a *standard* (noise-free) DPCM decoder (synthesis) filter. We will refer to this strategy as *standard JSC decoding for predictively encoded sources*. Note that the first step does not specify whether the estimate is obtained using an MMSE decoder [14], [22] or a MAP-based decoder [18]. Regardless, in this paper, we will demonstrate that the standard synthesis filter used in the second step is suboptimal, and that via a simple training-based technique, one can design a better synthesis filter for JSC decoding of predictively encoded sources. Moreover, this improved performance can be achieved without *any* increase in decoder complexity, relative to that of the standard approach. Alternatively, additional performance can be achieved with modest increases in complexity. We will refer to our method as a "two-stage" estimation approach because it follows a conventional JSC decoding estimator by a (second-stage) least-squares (LS) filter.

In [12], the suboptimality of standard predictive JSC decoding was explicitly asserted. The authors proposed an alternative decoder, which in some cases, outperforms the standard decoder. However, as will be discussed in Section II, in other cases, the decoder proposed in [12] is actually equivalent to the standard decoder. In Section IV, we will demonstrate improved performance over the decoder from [12] as well as over the standard decoder. Beyond the case of predictive coding, our work suggests a practical approach, in general, to the fundamental performance/complexity dilemma associated with JSC decoding techniques—how to effectively increase the model "order" (and thus, model accuracy) while retaining practical implementation complexity. This will be discussed in the following.

In the next section, we review relevant prior work. In Section III, we develop our new technique for the predictive-coding case and also discuss its more general applicability. In Section IV, we give experimental results and comparisons with other decoders. Finally, the paper concludes with a summary and the identification of some future work.

II. Prior Work on JSC Decoding for DPCM

A. Preliminaries

Consider the basic DPCM encoder shown in Fig. 1. The source sample X_t can be written $X_t = \tilde{X}_t + Z_t$, where $\tilde{X}_t$ is the prediction of the current sample based on past quantized values of the source, e.g., $\hat{X}_{t-1}$ in the first-order case, and

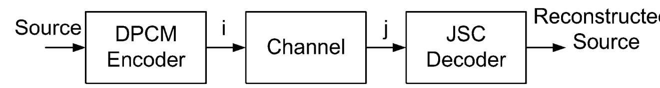


Fig. 2. Communications system model considered in this paper, with DPCM encoding, transmission over a memoryless channel, and JSC decoding to recover an estimate of the source.

Z_t is the prediction residual.[1] The decoder reconstructs the source via $\hat{X}_t = \hat{\tilde{X}}_t + \hat{Z}_t$, with $\hat{Z}_t$ the quantized prediction residual and $\hat{\tilde{X}}_t$ the prediction of X_t formed at the decoder. A fundamental law of predictive quantization, e.g., [10], is that the mean-squared reproduction error is the mean-squared error (MSE) in quantizing the prediction residual. In order to achieve this law, the decoder chooses $\hat{\tilde{X}}_t = \tilde{X}_t$, whereby $E[(X_t - \hat{X}_t)^2] = E[(Z_t - \hat{Z}_t)^2]$. The key here is the use of a common predictor at the encoder and decoder; otherwise, there would be an additional distortion term and error propagation. For the case of first-order DPCM, the decoder's reconstruction thus takes the form $\hat{X}_t = a\hat{X}_{t-1} + \hat{Z}_t$, with a the prediction coefficient.

Now consider the noisy-channel case. The communications system model is shown in Fig. 2. For clarity's sake, we only consider the memoryless-channel case here. The source encoder takes $X_t \in \mathcal{R}$ and produces an output index $I_t \in \mathcal{I}$, with $\mathcal{I}$ the quantization index set $\{0, 1, \ldots, N-1\}$. The index is transmitted over a discrete channel, resulting in a possibly corrupted index J_t, from the same set $\mathcal{I}$, according to the (assumed constant) channel-transition probabilities $\{P[J = j | I = i]\}$. We define the data source, from a discrete-time random process, as the sequence $\underline{X} \equiv (X_0, X_1, \ldots, X_{T-1})$, the encoder's index sequence $\underline{I} \equiv (I_0, I_1, \ldots, I_{T-1})$, and the received sequence, $\underline{J} \equiv (J_0, J_1, \ldots, J_{T-1})$. As in prior work [8], [14], [18], [21], we assume a Markovian model for the sequence of transmitted indexes, e.g., in the first-order case, $P[I_0 = i_0, I_1 = i_1, \ldots, I_t = i_t] = P[I_0 = i_0] \prod_{l=1}^{t} P[I_l = i_l | I_{l-1} = i_{l-1}]$ $\forall t$. All the probabilities $\{P[J = j | I = i]\}, \{P[I_0 = i_0]\}, \{P[I_t = i_t | I_{t-1} = i_{t-1}]\}$ are assumed known at the decoder. In practice, the source probabilities are estimated based on an encoded training set. In the case of a binary symmetric channel (BSC), the channel is completely specified by a single parameter, ϵ, the channel bit-error rate (BER). This value could be obtained from measurements. Alternatively, a robust value could be selected, based on an assumed distribution for ϵ, as, e.g., in [9].

The JSC decoder objective is to produce an approximation of the source $\underline{X}$, denoted $\hat{\underline{x}}^{(\mathrm{dec})} \equiv (\hat{x}_0^{(\mathrm{dec})}, \hat{x}_1^{(\mathrm{dec})}, \ldots, \hat{x}_{T-1}^{(\mathrm{dec})})$, given a realization of the noisy index sequence, $\underline{j} \equiv (j_0, j_1, \ldots, j_{T-1})$.[2] The knowledge brought to bear by the decoder consists of the source and channel probability models, as well as knowledge of the predictor and quantizer used by the encoder. For clarity's sake, in the following, we consider the case of a first-order DPCM encoder. Moreover, solely

[1]Random variables are denoted by capitals, with the corresponding lowercase symbols representing their realizations.

[2]Without knowledge of $\underline{j}$, the decoder output $\hat{X}_t^{(\mathrm{dec})}$ is treated as a random variable. However, the decoding rule is a deterministic function of the received sequence $\underline{j}$. For purpose of retaining compact notation, we do not explicitly indicate the dependence on $\underline{j}$ in $\hat{\underline{x}}^{(\mathrm{dec})}$.

for clarity's sake, it is assumed that knowledge of the entire received sequence $\underline{j}$ is used for decoding at each sample instant t. We refer to this as the "infinite-delay" case.[3] While our explicit development is for the "infinite-delay" case, we emphasize that our decoder formulation is straightforwardly specialized for the cases where at each t, only the causal subsequence $(j_0, j_1, \ldots, j_t)$ is used [18], and a noncausal subsequence is used at each t, but with finite delay τ, i.e., $\underline{j}_{\tau}^{(t)} \equiv (j_0, j_1, \ldots, j_{t+\tau})$ [14]. In fact, in our results, we evaluate decoders for both the infinite- and finite-delay cases.

B. Decoding Based on Estimating the Prediction Residual

As aforementioned, a common predictive JSC decoding strategy is to estimate the prediction residual and then feed this estimate to a standard (noise-free) DPCM decoder filter [6], [8], [16], [21]. In the case of a sequence MAP strategy [18] for estimating the residual, one first estimates the transmitted index sequence via $\hat{\underline{i}} = \arg\max_{\underline{i}'} P[\underline{I} = \underline{i}' | \underline{J} = \underline{j}]$. This is achieved via the dynamic programming algorithm. The JSC decoding rule is then

$$\hat{x}_t^{(\text{dec})} = a\hat{x}_{t-1}^{(\text{dec})} + Q_{\text{dec}}^{-1}(\hat{i}_t) \tag{1}$$

with $Q_{\text{dec}}^{-1}(\cdot)$ the inverse operation to the encoder's quantization, i.e., a table lookup for the quantization level. A reasonable choice is $Q_{\text{dec}}^{-1}(i) = E[Z_t | I_t = i] = Q_{\text{enc}}^{-1}(i)$, i.e., the same lookup table is used at both the encoder and decoder. In the case of a sequence-based approximate MMSE (SAMMSE) approach [14], [16], one first forms a conditional mean estimate of the prediction residual, denoted $E[Z_t|\underline{j}]$, based on the source and channel models, i.e.,

$$E[Z_t|\underline{j}] = \sum_{l=0}^{N-1} Q_{\text{dec}}^{-1}(l) P[I_t = l|\underline{j}] \tag{2}$$

where

$$P[I_t = l|\underline{j}] = \frac{\sum_{\underline{i}:i_t=l} P[\underline{J} = \underline{j}|\underline{I} = \underline{i}]P[\underline{I} = \underline{i}]}{\sum_{\underline{i}} P[\underline{J} = \underline{j}|\underline{I} = \underline{i}]P[\underline{I} = \underline{i}]}, \quad l \in \mathcal{I} \tag{3}$$

and where, again, we choose $Q_{\text{dec}}^{-1}(l) = E[Z_t|I_t = l]$. The *a posteriori* probabilities (APPs) (3) are computed efficiently via the Forward/Backward algorithm [14], [19]. The decoding rule is then [16]

$$\hat{x}_t^{(\text{dec})} = a\hat{x}_{t-1}^{(\text{dec})} + E[Z_t|\underline{j}]. \tag{4}$$

Note that in both (1) and (4), the decoded value consists of an estimate of the prediction $(a\hat{x}_{t-1}^{(\text{dec})})$ plus an estimate of the prediction residual. Some heuristic justification for this standard approach can be obtained, as follows. The ultimate objective is to choose the value $\hat{x}_t^{(\text{dec})}$ to minimize the MSE $D_t \equiv E[(X_t - \hat{x}_t^{(\text{dec})})^2|\underline{j}]$. We can represent X_t as $X_t = \tilde{X}_t^{(\text{enc})} + Z_t$, with $\tilde{X}_t^{(\text{enc})}$ the prediction produced by the encoder. Further, without loss of generality, we can represent the decoder's estimate, given $\underline{j}$, as the sum $\hat{x}_t^{(\text{dec})} = \tilde{x}_t^{(\text{dec})} + \hat{z}_t^{(\text{dec})}$. Then the distortion can be written as

$$\begin{aligned} D_t = E\left[\left(\tilde{X}_t^{(\text{enc})} - \tilde{x}_t^{(\text{dec})}\right)^2 \middle| \underline{j}\right] + E\left[\left(Z_t - \hat{z}_t^{(\text{dec})}\right)^2 \middle| \underline{j}\right] \\ +2E\left[\left(\tilde{X}_t^{(\text{enc})} - \tilde{x}_t^{(\text{dec})}\right)^T \left(Z_t - \hat{z}_t^{(\text{dec})}\right) \middle| \underline{j}\right]. \end{aligned} \tag{5}$$

Now, although it is unknown whether it is a reasonable assumption, suppose that the last (cross) term vanishes. Then, the total expected distortion is just the sum of the distortions in estimating the prediction[4] and in estimating the prediction residual. The distortion is thus minimized by choosing $\tilde{x}_t^{(\text{dec})}$ and $\hat{z}_t^{(\text{dec})}$, *separately*, to minimize their respective distortion terms. In the absence of other information, the most reasonable choice for the first term is $\tilde{x}_t^{(\text{dec})} = a\hat{x}_{t-1}^{(\text{dec})}$. Likewise, the choice $\hat{z}_t^{(\text{dec})} = E[Z_t|\underline{j}]$ minimizes the second term if the statistical model, on which the expectation is based, is accurate. Based on these choices, the standard decoder for predictively encoded sources can be seen to approximately minimize the distortion D_t.

While the above provides *heuristic* support for the decoding rule (4), in this paper, we will demonstrate that this rule is, in fact, suboptimal and that significant performance improvement can be achieved with *no* increase in complexity, simply by choosing different coefficients than a and 1 on the respective terms $\hat{x}_{t-1}^{(\text{dec})}$ and $E[Z_t|\underline{j}]$. Thus, through the performance of our new decoder, we will demonstrate that there is suboptimality stemming from neglecting the last term in D_t and/or from inaccuracy in the estimates of the prediction residual (due, e.g., to inaccuracy in the Markov model).

There are several ways to improve the estimate of the prediction residual $E[Z_t|\underline{j}]$. One strategy is to increase the order of the Markov model for $\{I_t\}$. However, the complexity of the forward and backward recursions needed to calculate the APPs (3) grows exponentially with the Markov order K, i.e., the complexity is $O(N^{(K+1)}T)$. This limits the practical feasibility of this approach. A second strategy is to use "memory-enhanced" decoding, as suggested in [14]. This approach is based on the fact that in deriving the SAMMSE decoder, one approximates $E[Z_t|\underline{I} = \underline{i}]$ by $E[Z_t|I_t = i_t]$. Memory-enhanced decoding uses a higher resolution decoder lookup table and thus obtains a more accurate conditional mean estimate. We will not derive this decoder here, but refer the reader to [14]. We simply give the decoder's form for the case of second-order (enhanced memory) decoding

$$E[Z_t|\underline{j}] = \sum_l \sum_m Q_{\text{dec}}^{-1}(l,m) P[I_t = l, I_{t-1} = m|\underline{j}] \tag{6}$$

where $Q_{\text{dec}}^{-1}(l,m) \equiv E[Z_t|I_t = l, I_{t-1} = m]$. It should be emphasized that the decoder memory can be usefully chosen to be second or higher order *irrespective of* the order of the Markov model for $\{I_t\}$.[5] Unfortunately, the number of summations increases linearly, and thus, the number of summands *exponentially*, with the order of the decoder memory. Moreover, the

[3]The sequence length T could correspond to the row length of an image or to the length of a block of samples from a sampled waveform. Blocking may be performed, e.g., to mitigate signal nonstationarity or to facilitate packetized transmission.

[4]This is reflective of the mismatch, now inherent for the noisy channel case, between the predictions used at the encoder and decoder.

[5]It is also not necessary for the conditioning context to be causal. It could also be noncausal, e.g., $E[Z_t|I_t = m, I_{t-1} = l, I_{t+1} = n]$.

complexity of calculating the APPs $P[I_t, I_{t-1}, \ldots, I_{t-K_d+1}|\underline{j}]$ *also* increases exponentially with the decoder memory K_d.

While increasing both the Markov order and the decoder memory may improve performance, there is a heavy price in complexity. In the next section, alternatively, we will demonstrate that the standard JSC decoder for predictive coding can be improved without *any* increase in complexity. Before developing this new approach, we first discuss [12], which attacks the same problem.

C. Recent Work on JSC Decoding for DPCM

In [12], the authors recognized the suboptimality of existing systems for the predictive case. They aimed to develop an improved decoder, directly estimating the source, rather than the prediction residual. For brevity's sake, we will not derive the authors' decoding rule. We will simply present their rule and discuss it. The decoder in [12] was developed by assuming a finite-memory prediction model, i.e., it was assumed that the current quantized prediction residual $\hat{Z}_t$ is a function of the last $\mu+1$ quantization indexes $I_t, I_{t-1}, \ldots, I_{t-\mu}$ [12]. For DPCM based on a *moving average* predictor of length μ, this is precisely valid. However, for autoregressive prediction, considered both here and in [12], this assumption is an approximation. Based on this assumption, the authors developed and proposed the following decoding rule for the case of finite delay τ:

$$\hat{x}_t^{(\mathrm{dec})} = a^{\mu+1}\hat{x}_{t-\mu-1}^{(\mathrm{dec})} + \sum_{i_t=0}^{N-1}\cdots\sum_{i_{t-\mu}=0}^{N-1}\left(\sum_{k=0}^{\mu} a^k E[Z_{t-k}|i_{t-k}]\right) \times P\left[I_t=i_t, I_{t-1}=i_{t-1}, \ldots, I_{t-\mu}=i_{t-\mu}\Big|\underline{j}_\tau^{(t)}\right]. \quad (7)$$

While (7) as written has complexity growing exponentially in μ, we show in the Appendix that this rule can be rewritten in the simplified form

$$\hat{x}_t^{(\mathrm{dec})} = a^{\mu+1}\hat{x}_{t-\mu-1}^{(\mathrm{dec})} + \sum_{k=0}^{\mu} a^k E\left[Z_{t-k}\Big|\underline{j}_\tau^{(t)}\right]. \quad (8)$$

This form gives a significant reduction in implementation complexity, so long as $\mu > K$. Likewise, for the "infinite-delay" case, the decoder from [12] [(7) with $\underline{j}_\tau^{(t)}$ replaced by $\underline{j}$], can be written in the simplified form

$$\hat{x}_t^{(\mathrm{dec})} = a^{\mu+1}\hat{x}_{t-\mu-1}^{(\mathrm{dec})} + \sum_{k=0}^{\mu} a^k E[Z_{t-k}|\underline{j}]. \quad (9)$$

In the next section, this decoding rule will be discussed in relation to our new decoder.

In [12], the authors experimentally compared their decoder with the standard decoding rule

$$\hat{x}_t^{(\mathrm{dec})} = a\hat{x}_{t-1}^{(\mathrm{dec})} + E\left[Z_t\Big|\underline{j}_\tau^{(t)}\right]. \quad (10)$$

This is simply (4) with $\underline{j}$ replaced by $\underline{j}_\tau^{(t)}$. In [12], the authors demonstrated that their decoder outperforms (10) for a high-order Gauss–Markov (GM) source, with results given for the case $\tau = 0$. However, in the Appendix, we show that for the "infinite-delay" case, the decoder from [12] is actually *equivalent to* the standard decoding rule (4). Also in Section IV, we give experimental results showing, for the same high-order source, how the performance of (10) and (7) quickly approach each other as τ is increased. For a first-order GM source, our results in Section IV show similar performance for these two decoders, even for very small τ. To summarize, the Appendix and our experiments suggest that the decoder from [12] may only give improved performance over the standard decoder in some cases. Alternatively, we next develop a new decoder, which, in our experiments, outperforms the standard decoder for both the finite- and infinite-delay cases, for both low- and high-order GM sources, and does so without any increase in complexity. Moreover, in the finite-delay case, we will demonstrate experimentally that our new decoder gives better performance than [12].

III. New JSC Decoder for DPCM-Encoded Sources

In JSC decoding, e.g., [14], [18], and [21], the approach often taken is to assume an optimality criterion (such as MMSE or MAP) and a statistical model, and then to analytically derive a closed-form decoder expression. However, as indicated in Section II-B, it is difficult to give an accurate analytical formulation in the predictive case. Alternatively, we take our cue from *training-based* approaches to quantizer design, proposing a training-based approach to the design of a JSC decoder. The ultimate performance is the MSE $E[(X - \hat{X}^{(\mathrm{dec})})^2]$, with the expectation taken with respect to both the source and channel distributions. Now, as is often done in practice without proof, e.g., [5], let us suppose that an *ergodicity property* holds in our case, i.e.,

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\left(x_t - \hat{x}_t^{(\mathrm{dec})}(\underline{j})\right)^2 = E\left[\left(X - \hat{X}^{(\mathrm{dec})}\right)^2\right] \quad (11)$$

where we have emphasized the rule's dependence on $\underline{j} = (j_0, j_1, \ldots, j_{T-1})$. The motivation behind this is that, if (11) holds, then one can choose the rule $\hat{x}_t^{(\mathrm{dec})}(\underline{j})$ to minimize a LS cost based on a large training set *and a single realization of the channel*, $\underline{j}$, with the reasonable expectation that one is then (approximately) choosing the decoder to minimize the MSE $E[(X - \hat{X}^{(\mathrm{dec})})^2]$. A similar approach was taken in [2] for a different source-coding context and for optimization at the *encoder*, not the decoder.[6] The reasonableness of our ergodicity assumption will be substantiated by our experimental results. In particular, it will be seen that the decoding performance optimized over the training set closely agrees with the performance on multiple independent test sets.

Thus, from the perspective of a *least-squares error* (LSE) performance criterion, $\sum_{t=0}^{T-1}(x_t - \hat{x}_t^{(\mathrm{dec})}(\underline{j}))^2$, we now observe the

[6]One might wonder why we are emphasizing an ergodicity property here, since source-coder design for noisy channels is typically based on training sets, e.g., in channel-optimized vector quantizer (COVQ) design [9]. The difference is that in COVQ, the decoder is explicitly designed to minimize the expected distortion over the training set, with the effect of the channel analytically accounted for in the expected distortion expression. This cannot be precisely done in our case. Thus, we generate a single channel realization and assume an ergodicity property.

following: $\hat{x}_t^{(\mathrm{dec})}$ as given in (4) can be viewed as a *linear estimator* of x_t, based on (what we accordingly now recognize as) the derived "observations" $\hat{x}_{t-1}^{(\mathrm{dec})}$ and $E[Z_t|\underline{j}]$. This raises the question of whether the coefficients $(a, 1)$ are *optimal* (or nearly so) in the LS sense, as the weights for these observations.

Consider the more general estimator

$$\hat{x}_t^{(\mathrm{newdec})} = \alpha \hat{x}_{t-1}^{(\mathrm{dec})} + \beta E[Z_t|\underline{j}] \quad (12)$$

with $\hat{x}_{t-1}^{(\mathrm{dec})}$ obtained from the standard decoding rule (4). Suppose (4) is first applied, yielding $\{\hat{x}_t^{(\mathrm{dec})},\ t = 0, 1, \ldots, T-1\}$, with the optimal coefficients then sought to weight the values $\hat{x}_{t-1}^{(\mathrm{dec})}$ and $E[Z_t|\underline{j}]$ in forming a *new* decoding estimate $\{\hat{x}_t^{(\mathrm{newdec})},\ t = 0, 1, \ldots, T-1\}$. For the given training set $\underline{x} = (x_0, x_1, \ldots, x_{T-1})^T$ and a realization of the channel $\underline{j}$, we seek the pair (α, β) optimal, in the LS sense, for combining the "data observations" $\hat{x}_{t-1}^{(\mathrm{dec})}$ and $E[Z_t|\underline{j}]$. This is a standard estimation problem, e.g., [11], with solution given in the form

$$(\alpha \beta)^T = (\mathcal{X}^T \mathcal{X})^{-1} \mathcal{X}^T \underline{x} \quad (13)$$

based on the "data matrix"

$$\mathcal{X}^T = \begin{bmatrix} \hat{x}_{-1}^{(\mathrm{dec})} & \hat{x}_{0}^{(\mathrm{dec})} & \cdots & \hat{x}_{T-2}^{(\mathrm{dec})} \\ E[Z_0|\underline{j}] & E[Z_1|\underline{j}] & \cdots & E[Z_{T-1}|\underline{j}] \end{bmatrix} \quad (14)$$

and with $\hat{x}_{-1}^{(\mathrm{dec})}$ an initial value. Our new approach to JSC decoding for predictively encoded sources is to use the LS-optimal coefficients in the decoding rule (12). Since our new LS-optimal decoder forms standard JSC decoding values (4) using an assumed statistical model and statistical inferences (APPs) estimated based on the model and then optimally weights, in the LS sense, these values and the estimated prediction errors, we refer to our approach as a "two-stage" estimation technique.

Consider an example of a first-order GM process $\{X_t\}$ with correlation coefficient 0.95, $N = 8$ quantization levels, a first-order Markov model for the sequence $\{I_t\}$, and a BSC with BER $\epsilon = 0.05$. Residual redundancy is introduced by mismatching the prediction coefficient, relative to the source correlation. Suppose the prediction coefficient is chosen as $a = 0.45$. Experimentally, we have found that this choice leads to good decoding performance, both for our system and for the standard JSC decoder [which uses $(a, 1)$]. For this choice, we find that the LS-optimal pair (based on a training set of 10^6 samples), is $(\alpha, \beta) = (0.55, 0.79)$, quite different from the values (0.45, 1) that give the standard rule. Morever, averaged over three test sets of size 50 000 samples, the reduction in distortion of this LS-optimal decoder over the standard JSC decoder is $10 \log_{10}$ (MSE of standard JSC/MSE of new JSC) $= 0.41$ dB. Clearly, the standard JSC rule is *not*, in general, the optimal way to combine the "observations" $E[Z_t|\underline{j}]$ and $\hat{x}_{t-1}^{(\mathrm{dec})}$.

Increasing the Filter Order: Even better performance is achieved by increasing the order of the filter, i.e., forming

$$\hat{x}_t^{(\mathrm{newdec})} = \sum_{k=1}^{L_c} \alpha_k \hat{x}_{t-k}^{(\mathrm{dec})} + \beta E[Z_t|\underline{j}], \quad L_c \geq 1. \quad (15)$$

The decoding *form* in (15) is equivalent to the form in (9) if we choose $L_c = \mu$. In particular, by using (4) to expand each term $\hat{x}_{t-k}^{(\mathrm{dec})}$ in (15) and then simplifying further, it can be shown that

$$\hat{x}_t^{(\mathrm{newdec})} = \beta E[Z_t|\underline{j}] + \sum_{k=1}^{\mu} \left(\sum_{l=1}^{k} \alpha_l a^{k-l} \right) \times E[Z_{t-k}|\underline{j}] + \alpha_\mu a \hat{x}_{t-\mu-1}^{(\mathrm{newdec})}. \quad (16)$$

The Appendix shows that the decoder (9) is equivalent to the standard JSC decoder (4). Thus, the difference between the standard JSC decoder and (16) is just the choice of weights applied to each of the terms $\{E[Z_{t-k}|\underline{j}], k = 0, 1, \ldots, \mu\}$ and $\hat{x}_{t-\mu-1}$. While the standard decoder (4) effectively prescribes the respective weights $1, a, a^2, \ldots, a^\mu, a^{\mu+1}$, we take a LS, training-based approach to choosing them. Note also that, rather than choosing the decoder form (15), we could instead choose the form

$$\hat{x}_t^{(\mathrm{newdec})} = \alpha \hat{x}_{t-\mu-1}^{(\mathrm{dec})} + \sum_{k=0}^{\mu} \beta_k E[Z_{t-k}|\underline{j}]. \quad (17)$$

This is *explicitly* equivalent to (9). Although we do not prove that LS training of the two models (15) and (17) yields identical decoders, this is at least plausible.[7] To give some additional insight, we have performed LS training for both (15) and (17). We found that these two forms gave decoded sequences with samples agreeing up to the third or fourth fractional decimal digit. The difference may be attributable to finite numerical precision.

Causal and Anticausal Terms: While (15) and (17) only include the *causal* terms $\{\hat{x}_{t-k}^{(\mathrm{dec})},\ k \geq 0\}$ and $\{E[Z_{t-k}|\underline{j}],\ k \geq 0\}$, respectively, our method can also be applied to design a decoder that uses *anticausal* terms $\{\hat{x}_{t+k}^{(\mathrm{dec})},\ k \geq 0\}$, with the potential for additional gains in performance. In particular, consider the rule

$$\hat{x}_t^{(\mathrm{newdec})} = \sum_{k=1}^{L_c} \alpha_k \hat{x}_{t-k}^{(\mathrm{dec})} + \sum_{k=1}^{L_{nc}} \alpha_{-k} \hat{x}_{t+k}^{(\mathrm{dec})} + \beta E[Z_t|\underline{j}] \quad (18)$$

with L_c the maximum causal lag and L_{nc} the maximum anticausal advance.[8] The benefit of including these additional terms in the decoding estimate will be explored in the next section.

LS Decoding for Nonpredictive Systems: In [14] and [18], JSC decoding was considered for the case of a nonpredictive encoder (a basic scalar or vector quantizer). In [14], for the SAMMSE decoder, it was noted that while, in principle, performance could be improved by increasing the order of the Markov model for $\{I_t\}$, this order is generally kept quite small to retain manageable complexity. Let us denote the SAMMSE decoder output (based, e.g., on a first-order Markov model for $\{I_t\}$) by the sequence $\{\hat{x}_t^{(\mathrm{SAMMSE})}\}$. One potential consequence of keeping the order low is that there may be unexploited partial correlation between the source X_t and both causal $(\hat{x}_{t-k}^{(\mathrm{SAMMSE})},\ k \geq 1)$ and anticausal $(\hat{x}_{t+k}^{(\mathrm{SAMMSE})},\ k \geq 1)$

[7]The LS cost is convex, and (16) demonstrates that (15) and (17) perform linear filtering using the same set of "observations".

[8]The anticausal terms can always be computed when $\underline{j} = (j_0, j_1, \ldots, j_{T-1})$. For the finite-delay case where, at time t, $\underline{j}_\tau^{(t)} = (j_0, j_1, \ldots, j_{t+\tau})$, they can be computed if $\tau \geq L_{nc}$.

terms. This again suggests improving the decoding result via a two-stage estimation procedure

$$\hat{x}_t^{(\text{newdec})} = \sum_{k=0}^{L_c} \alpha_k \hat{x}_{t-k}^{(\text{SAMMSE})} + \sum_{k=1}^{L_{nc}} \alpha_{-k} \hat{x}_{t+k}^{(\text{SAMMSE})} \quad (19)$$

with the α_k coefficients again optimized for a LS training-set cost. Low-order (e.g., $K = 1$) SAMMSE decoding followed by the filtering in (19) is *much* less complex than SAMMSE decoding based on a high-order Markov model. Thus, this approach can potentially improve the performance of a low-order JSC decoder while retaining modest complexity.

IV. Experimental Results

We compared our proposed decoder against: naive DPCM $[\hat{x}_t^{(\text{dec})} = a\hat{x}_{t-1}^{(\text{dec})} + Q_{\text{enc}}^{-1}(j_t)]$; standard JSC decoding (4); and the decoder from [12]. We also conducted experiments comparing our method against SAMMSE decoding [14] for the nonpredictive case.

Predictive Coding Results: We evaluated all the methods for a first-order GM source $\{X_t\}$ with correlation coefficient $\rho = 0.95$ (denoted GM-1(0.95) in the tables) and with input white-noise variance $\sigma_w^2 = 1.0$. We assumed a BSC with BER of 5×10^{-2} (denoted BSC(0.05)), both for decoder design and testing. A training set of 1 million samples was used for all decoder designs. We also generated three independent test sets, each of size 50 000 samples.[9] We assumed first-order DPCM at three bits/sample. A uniform quantizer was chosen, based on the dynamic range of the prediction residual. While optimizing the quantizer consistent with the channel and the decoding scheme, perhaps along the lines of [2], could further improve system performance, the main goal of our experiments was to compare decoding techniques in a common setting. For SAMMSE estimation of the prediction residual, i.e., $E[Z_t|\underline{j}]$, we assumed a first-order Markov model for $\{I_t\}$. The choice of the predictor is very important in JSC decoding for the DPCM case. A value for a too close to ρ will not introduce sufficient residual redundancy, while a value too small relative to ρ will not adequately capture the DPCM prediction gain. A robust choice for a was suggested in [3]. However, we have taken an empirical approach, trying a number of values for a in the interval [0.1, 0.95] and measuring the decoders' performances for each. We first conducted experiments for the infinite-delay case.

Table I gives signal-to-quantization-noise ratio ($\text{SQNR} = 10\log_{10}(\sigma_x^2/\text{MSE})$) results, with the MSE and σ_x^2 both averaged over the three test sets, for naive decoding, standard JSC decoding (4), and our LS decoder (12). The SQNR is evaluated for the indicated choices of a, with the associated LS (α, β) pair shown. The column "Gain" is the gain in SQNR (in decibel units) of our LS decoder, relative to the standard decoder. This gain ranges from 1.01 dB $(a = 0.1)$ down to 0.17 dB $(a = 0.65)$. The best performance for both decoders is achieved at $a = 0.45$, where the LS decoder is 0.41 dB better than the standard decoder. The best performance for the naive decoder occurs at $a = 0.72$, the coefficient choice from [3]. However, the naive decoder is still about 9 dB worse than the best-performing LS decoder. Note also that the choice $a = 0.72$ is far from optimal for standard and LS decoding. Consider the design at $a = 0.35$. The training set SQNR is 12.46 dB, while the average test set SQNR is 12.40 dB. Similar small differences between training and test set performance (less than 0.1 dB) were consistently observed in our experiments. Finally, note that in all our experiments, for all a values, we found that the LS design chooses $\alpha > a$ and $\beta < 1$, i.e., increased weight is given to the prediction and diminished weight to the estimated prediction residual, compared with the standard decoder.[10] This is a fundamental characteristic observed in our LS solutions.

TABLE I
SQNR Performance of Naive, Standard, and LS Decoding for DPCM Encoding of a GM-1(0.95) Source, as a Function of the Prediction Coefficient, for the Case of "Infinite-Delay" Decoding. A BSC(0.05) Channel Was Used. Also Shown Are LS Coefficients

Prediction coefficient	Naïve Dec. Avg. SQNR	Std. Dec. Avg. SQNR	LS Dec. Avg. SQNR	Gain vs. Std. dec.	α	β
0.10	-0.97	10.19	11.20	1.01	0.35	0.68
0.35	1.07	11.76	12.40	0.64	0.49	0.75
0.45	1.88	12.01	12.42	0.41	0.55	0.79
0.55	2.72	11.99	12.22	0.23	0.63	0.83
0.65	3.41	11.34	11.50	0.17	0.72	0.88
0.72	3.43	10.61	10.84	0.24	0.81	0.91
0.85	2.48	8.41	8.98	0.57	1.01	1.00

TABLE II
SQNR Performance of "Infinite-Delay" Standard and LS Decoding for DPCM Encoding of a GM-1(0.95) Source as a Function of the Prediction Coefficient. A BSC(0.05) Channel Was Used. In This Case, LS Decoding Uses One Causal and One Anticausal Sample. Also Shown Are LS Coefficients

Prediction coefficient	Std. Dec. Avg. SQNR	LS Dec. Avg. SQNR	Gain	α_1	β	α_{-1}
0.10	10.19	11.75	1.56	0.30	0.50	0.23
0.35	11.76	12.85	1.09	0.41	0.58	0.20
0.45	12.01	12.77	0.76	0.47	0.62	0.19
0.55	11.99	12.46	0.48	0.54	0.67	0.17
0.65	11.34	11.66	0.32	0.63	0.72	0.16
0.72	10.61	10.97	0.36	0.70	0.76	0.16
0.85	8.41	9.06	0.65	0.88	0.85	0.16

Table II gives results comparing the standard decoder (4) and our LS decoder (18) with $L_c = L_{nc} = 1$. In this case, the gain of our decoder at $a = 0.45$ is 0.76 dB, and at $a = 0.35$, the gain is 1.09 dB, i.e., $\sim$ 0.4-dB additional gain is attributable to the $\hat{x}_{t+1}^{(\text{dec})}$ term. Table III shows that, at least in the first-order GM case, the benefit of using additional causal terms diminishes rapidly. The additional gain from 10 terms [$L_c = 10$ in (15)] is < 0.2 dB, relative to (12) [which is $L_c = 1$ in (15)].

We also conducted experiments for the finite-delay case, comparing with [12]. We first considered both $\tau = 0$ and $\tau = 1$. We evaluated performance both for the first-order GM source and for the tenth-order GM source from [12]. For the first-order source, we chose $a = 0.35$. For the tenth-order

[9]We found that this size test set was adequate to give representative results. In particular, increasing the test set length to 500 000 samples gave results (averaged over the three independent test sets) within 0.02 dB of those based on 50 000 samples, for the several cases we tried.

[10]The value $\beta = 1.00$ for $a = 0.85$ in Table I was rounded up from 0.996.

TABLE III
SQNR PERFORMANCE OF NAIVE, STANDARD, AND LS DECODING FOR DPCM ENCODING OF A GM-1(0.95) SOURCE AS A FUNCTION OF THE PREDICTION COEFFICIENT, FOR THE CASE OF "INFINITE-DELAY" DECODING. A BSC(0.05) CHANNEL WAS USED. ALSO SHOWN ARE LS COEFFICIENTS. IN THIS CASE, THE LS DECODER USES 10 CAUSAL SAMPLES

Prediction coefficient	Std. Dec. Avg. SQNR	LS Dec. Avg. SQNR	Gain vs Std. Dec.
0.35	11.76	12.53	0.77
0.45	12.01	12.54	0.53
0.55	11.99	12.35	0.37

TABLE IV
PERFORMANCE OF STANDARD DECODER, [12], AND LS DECODER FOR THE GM-1(0.95) AND 10TH-ORDER GM SOURCES FOR $\tau = 0, 1$. A BSC(0.05) CHANNEL WAS USED

	1^{th} order source			10^{th} order source		
Delay	Std. Dec.	[12]	LS Dec.	Std. Dec.	[12]	LS Dec.
0	10.96	10.91	11.69	14.44	15.30	15.78
1	11.57	11.57	12.71	15.82	16.13	16.76

source, we chose $a = 0.83$ as in [12]. For $\tau = 0$, we compared (7) [implemented efficiently via (8)] with the LS decoder (17), but with $\underline{j}$ replaced by $\underline{j}_{\tau}^{(t)}$. For both decoders, how to calculate the quantities $E[Z_{t-k}|\underline{j}_{\tau}^{(t)}]$, i.e., SAMMSE decoding in the finite-delay case, is described in [14]. For both decoders, we used $\mu = 3$. These decoders have the same complexity. We also compared with the standard decoder (10). For $\tau = 1$, we compared (8) for $\mu = 4$ against our LS decoder, written in the form

$$\hat{x}_t^{(\text{newdec})} = \alpha \hat{x}_{t-\mu-1}^{(\text{dec})} + \sum_{k=-1}^{3} \beta_k E\left[Z_{t-k}|\underline{j}_{\tau}^{(t)}\right]. \quad (20)$$

Note that these decoders have essentially the same complexity. Further, note that both decoders take advantage of the delay ($\tau = 1$), while our decoder also efficiently includes an anticausal term $E[Z_{t+1}|\underline{j}_{\tau}^{(t)}]$. The decoder from [12], (8) can only include similar dependence by using second-order memory ($E[Z_t|I_t, I_{t+1}]$) or third-order memory ($E[Z_t|I_{t-1}, I_t, I_{t+1}]$). However, as already discussed, this would entail a large increase in complexity. The results are shown in Table IV, with our LS decoder best in all cases and with the decoder from [12] only better than the standard decoder for the tenth-order GM source. To further explore results for the first-order source, we also tried choosing $a = 0.72$ (based on [3]), instead of $a = 0.35$. In this case, we found (8) achieved 9.85 and 10.24 dB for $\tau = 0, 1$, respectively, compared with 9.50 and 10.18 dB for the standard decoder. While (8) does gain over the standard decoder for $a = 0.72$, the results for both decoders are worse (by more than 1 dB) than for the choice $a = 0.35$. Also, for the tenth-order source, we evaluated (8) for $\mu = 4$ against the standard decoder (10) as τ is increased. Fig. 3 shows that the performance of (10) and (8) approach each other fairly quickly, with the two curves indistinguishable by $\tau = 6$. Thus, for this source, the "infinite-delay" equivalence of [12] and the standard decoder is borne out at a fairly small τ value. For the first-order GM source, the convergence of the two decoders is more rapid, with the performance of the two decoders comparable even at $\tau = 1$.

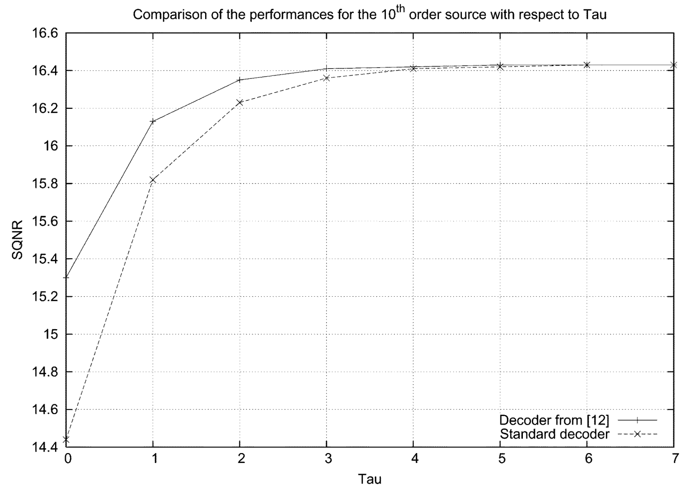


Fig. 3. SQNR performance of the standard decoder and the decoder from [12] for the tenth-order GM source, as a function of τ. A BSC(0.05) channel was used.

TABLE V
SQNR PERFORMANCE OF NAIVE, STANDARD, AND LS DECODING FOR DPCM ENCODING OF GM-1(0.95) SOURCE UNDER CHANNEL MISMATCH CONDITIONS, FOR THE CASE OF "INFINITE DELAY." STANDARD AND LS DECODERS WERE DESIGNED FOR BER $\epsilon = 0.05$; $a = 0.35$. THE NAIVE DECODER USED $a = 0.72$. THE LS DECODER USES ONE CAUSAL AND ONE ANTICAUSAL SAMPLE. LAST COLUMN GIVES LS DECODER PERFORMANCE MATCHED TO THE CHANNEL (FOR $a = 0.35$)

BER	Naïve Dec. Avg. SQNR	Std. Dec. Avg. SQNR	LS Dec. Avg. SQNR	Gain vs. Std. Dec.	LS-opt Dec. Avg. SQNR
0.001	15.79	12.98	14.34	1.36	14.77
0.010	9.83	12.76	14.06	1.30	14.30
0.050	3.43	11.76	12.85	1.09	12.85
0.100	0.63	10.25	11.15	0.90	11.41

Finally, Table V gives results of some mismatch experiments for the infinite-delay case, where the decoders were all designed for BER $\epsilon = 0.05$, but where the true $\epsilon = 0.001,\ 0.01$, and 0.1. Here, the naive decoder used $a = 0.72$, while the other decoders used $a = 0.35$. There are several observations to make here. First, it is seen that the LS decoder retains a substantial SQNR advantage over standard JSC decoding under channel mismatch. Second, note that for BER $= 0.001$, naive decoding actually gives better results than the mismatched LS decoder. Finally, the last column gives performance of an LS decoder matched to the channel condition. This gives an indication of the loss in LS decoder performance, attributable to channel mismatch.[11]

Computational Complexity: The number of multiplications and additions for our decoding approach are $O(N^{K+1}T + PT)$, with P the order of the LS filter. The complexity of standard JSC decoding is $O(N^{K+1}T)$. For the choices of N and P considered here, the complexity of our approach is only slightly greater than that of standard JSC decoding. The complexity of both our approach and standard JSC decoding are much greater

[11]The reason that the matched LS decoder performance is worse than the naive decoder performance for BER $= 0.001$ is the choice of a. When the matched LS decoder also uses $a = 0.72$, it achieves 17.68 dB, outperforming the naive decoder at BER $= 0.001$.

than for naive decoding; running C under Linux on a 1.4-GHz machine, we found that for $N = 8,\ K = 1$, naive decoding processes 10^6 samples in 2 s, while standard JSC decoding and our LS approach require more than 3 min (a more than 90 times increase). However, this is compensated by (much better) JSC decoding performance for high-BER channels.

Nonpredictive Coding Results: We have also investigated our LS optimization for improving nonpredictive SAMMSE decoding [14], based on (19). As a source, we chose the second discrete cosine transform (DCT) coefficient (based on zigzag scan order) from 8×8 blocks of a gray-scale image. A 1-D source (time series) was created by applying the 8×8 DCT transform to the image blocks and then scanning the coefficients using a fixed row-by-row scanning order. We chose a scalar quantizer with eight quantization levels, designed via the Lloyd algorithm using 23 gray-scale images. The initial quantizer was randomly selected from the training data. Again, we chose $\epsilon = 0.05$ for both training and testing. We formed SAMMSE decoders based on both first- and second-order Markov models for the quantization indexes. The Markov probabilities were obtained from frequency counts based on encoding of the source. SAMMSE decoding based on a first-order Markov model gave an SQNR, measured over the same 23-image source, of 2.99 dB. For a second-order Markov model, SAMMSE decoding gave 4.52 dB. We next designed an LS-optimal decoder that filters the first-order SAMMSE decoding result, according to (19). For this LS-optimal decoder, we chose $L_c = 3,\ L_{nc} = 2$ and optimized the $\{\alpha_k\}$ for each of the 23 images. This was done because we found that only small performance gains are achieved if a single "universal" filter $\{\alpha_k\}$ is used for all the images. A small amount of side information to the decoder is required, to specify the coefficients $\{\alpha_k\}$ for each image. The resulting LS-optimal decoder performance, averaged over the 23-image source, was 4.12 dB–∼1.1 dB better than first-order SAMMSE, and only ∼0.4 dB worse than second-order SAMMSE. The LS decoder complexity $(O(N^2T + PT))$ was ∼8% higher than first-order SAMMSE $(O(N^2T))$ and roughly seven times less than second-order SAMMSE $(O(N^3T))$.

V. Conclusion

We first reviewed prior work on JSC decoding for DPCM, including [12]. Next, we proposed a new decoder that adds degrees of freedom, allowing one to select the relative weights given to the prediction and prediction-error terms within the decoding rule. This decoder is designed on training data to optimize a LS criterion. This new decoder was found to improve upon the standard JSC decoder and [12] without any increase in complexity. Moreover, additional gains over the standard decoder can be achieved with quite modest increases in complexity. The flexibility of our LS procedure was also demonstrated, via extension, to include anticausal terms in the estimation. Finally, we applied our design procedure even more generally, to yield better decoders for the nonpredictive case as well, while retaining practical complexity. This approach was demonstrated for decoding image-transform coefficients. An open question is whether the use of *nonlinear* estimators, e.g., neural networks, could further improve upon our results. Future work may consider an extension for systems that use both predictive coding and variable length coding, a 2-D extension specifically for decoding digital images, as well as extension for the case of predictive vector quantization.

Appendix

Here, we consider the infinite-delay case and show that the decoding rule (7) under infinite delay (i.e., with $\underline{j}_{\tau}^{(t)}$ replaced by $\underline{j}$) is actually equivalent to the standard decoder. First, we rewrite (7) assuming infinite delay, i.e.,

$$\hat{x}_t^{(\mathrm{dec})} = a^{\mu+1}\hat{x}_{t-\mu-1}^{(\mathrm{dec})} + \sum_{i_t=0}^{N-1}\cdots\sum_{i_{t-\mu}=0}^{N-1}\left(\sum_{k=0}^{\mu} a^k E[Z_{t-k}|i_{t-k}]\right) \times P[I_t = i_t, I_{t-1} = i_{t-1}, \ldots, I_{t-\mu} = i_{t-\mu}|\underline{j}]. \quad (21)$$

Now, note that if $\mu > K$, the Markov model order, then *irrespective of the order of the Markov model for the quantized source* (so long as it is first order or greater), the computational and memory complexity of (21) [and of (7)] is $O(TN^{\mu+1})$. The Forward and Backward recursions used to calculate the *joint* probability $P[I_t = i_t, I_{t-1} = i_{t-1}, \ldots, I_{t-\mu} = i_{t-\mu}|\underline{j}]$ require computations on a trellis with $N^{\mu+1}$ states. However, this can be simplified as follows. Noting that $E[Z_{t-k}|i_{t-k}]$ is a function solely of i_{t-k}, we can simply get that

$$\begin{aligned}
&\sum_{i_t=0}^{N-1}\cdots\sum_{i_{t-\mu}=0}^{N-1}\left(\sum_{k=0}^{\mu} a^k E[Z_{t-k}|i_{t-k}]\right) \\
&\quad \times P[I_t = i_t, I_{t-1} = i_{t-1}, \ldots, I_{t-\mu} = i_{t-\mu}|\underline{j}] \\
&= \sum_{k=0}^{\mu}\sum_{i_t=0}^{N-1}\cdots\sum_{i_{t-\mu}=0}^{N-1} a^k E[Z_{t-k}|i_{t-k}] \\
&\quad \times P[I_t = i_t, I_{t-1} = i_{t-1}, \ldots, I_{t-\mu} = i_{t-\mu}|\underline{j}] \\
&= \sum_{k=0}^{\mu} a^k \left(\sum_{i_{t-k}=0}^{N-1} E[Z_{t-k}|i_{t-k}]P[I_{t-k} = i_{t-k}|\underline{j}]\right) \\
&= \sum_{k=0}^{\mu} a^k E[Z_{t-k}|\underline{j}]. \quad (22)
\end{aligned}$$

The second step was achieved by explicit marginalization of the joint probability mass function (pmf) $P[I_t = i_t, I_{t-1} = i_{t-1}, \ldots, I_{t-\mu} = i_{t-\mu}|\underline{j}]$. The third step was obtained by using (2) to replace the term in parentheses. Thus, we simply obtain the decoding rule

$$\hat{x}_t^{(\mathrm{dec})} = a^{\mu+1}\hat{x}_{t-\mu-1}^{(\mathrm{dec})} + \sum_{k=0}^{\mu} a^k E[Z_{t-k}|\underline{j}]. \quad (23)$$

This rule (first given as (9) in Section II-C) gives the same decoded sequence as (21). Note that $E[Z_{t-k}|\underline{j}]$ is obtained from (3) and (2), i.e., from the SAMMSE decoder [14]. The complexity of SAMMSE decoding is $O(TN^{K+1})$. If $\mu > K$, there is a large reduction in complexity for (9), relative to (21). Moreover, even in the finite-delay case, the marginalization applied in (22) can be used to reduce the complexity of (7), again so long as $\mu > K$. In particular, a similar derivation leads, in the finite-delay case, to the decoder

$$\hat{x}_t^{(\mathrm{dec})} = a^{\mu+1}\hat{x}_{t-\mu-1}^{(\mathrm{dec})} + \sum_{k=0}^{\mu} a^k E\left[Z_{t-k}\left|\underline{j}_{\tau}^{(t)}\right.\right] \quad (24)$$

where $E[Z_{t-k}|\underline{j}_{\tau}^{(t)}]$ is given by (2).

While we have shown that for $\mu > K$, the complexity of (21) and (7) can be reduced, we next consider the standard JSC decoder (4) and demonstrate that (9) [and thus, (21)] is, in fact, *equivalent* to this rule. In particular

$$\begin{aligned}\hat{x}_t^{(\text{dec})} &= a\hat{x}_{t-1}^{(\text{dec})} + E[Z_t|\underline{j}] \\ &= a\left(a\hat{x}_{t-2}^{(\text{dec})} + E[Z_{t-1}|\underline{j}]\right) + E[Z_t|\underline{j}] \\ &= a^2\hat{x}_{t-2}^{(\text{dec})} + aE[Z_{t-1}|\underline{j}] + E[Z_t|\underline{j}] \\ &\cdot \\ &= a^{\mu+1}\hat{x}_{t-\mu-1}^{(\text{dec})} + \sum_{k=0}^{\mu} a^k E[Z_{t-k}|\underline{j}]. \end{aligned} \tag{25}$$

We have just shown that for the "infinite-delay" case, (21) from [12] is equivalent to (9), which, in turn, is equivalent to the standard rule (4).

In the finite-delay case, the standard rule and the rule from [12] are not the same. In particular, the standard decoding rule is obtained by replacing $\underline{j}$ by $\underline{j}_\tau^{(t)}$ in (4). This can be shown to be equivalent to the rule

$$\hat{x}_t^{(\text{dec})} = a^{\mu+1}\hat{x}_{t-\mu-1}^{(\text{dec})} + \sum_{k=0}^{\mu} a^k E\left[Z_{t-k}\,\middle|\,\underline{j}_{\tau-k}^{(t)}\right]. \tag{26}$$

Comparing with (8), we see that (8) uses $E[Z_{t-k}|\underline{j}_\tau^{(t)}]$, while (26) uses $E[Z_{t-k}|\underline{j}_{\tau-k}^{(t)}]$, i.e., for $k > 0$ (8) conditions on more (anticausal) received indexes than (26).

Acknowledgment

The authors would like to thank F. Lahouti and Prof. A. Khandani for a helpful discussion relating to [12].

References

[1] E. Ayanoglu and R. M. Gray, "The design of joint source and channel trellis waveform coders," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 855–865, May 1987.
[2] K.-H. Chei and K.-P. Ho, "Design of optimal soft decoding for combined trellis coded quantization/modulation in Rayleigh fading channel," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 2000, pp. 2633–2636.
[3] K.-Y. Chang and R. W. Donaldson, "Analysis, optimization, and sensitivity study of differential PCM systems operating on noisy communication channels," *IEEE Trans. Commun.*, vol. COM-20, pp. 338–350, Mar. 1972.
[4] J. Chou and K. Ramchandran, "Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 861–867, June 2000.
[5] J. R. Deller, Jr., J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*. New York: Macmillan, 1993, p. 47.
[6] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes," in *Proc. Data Comp. Conf.*, 1998, pp. 139–148.
[7] G. F. Elmasry, "Joint lossless-source and channel coding using automatic repeat request," *IEEE Trans. Commun.*, vol. 47, pp. 953–955, July 1999.
[8] S. Emami and S. Miller, "DPCM picture transmission over noisy channels with the aid of a Markov model," *IEEE Trans. Image Processing*, vol. 4, pp. 1473–1479, Nov. 1995.
[9] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. Inform. Theory*, vol. 36, pp. 799–809, Apr. 1990.
[10] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.
[11] M. Hayes, *Statistical Digital Signal Processing and Modeling*. New York: Wiley, 1996.
[12] F. Lahouti and A. K. Khandani, "Sequence MMSE source decoding over noisy channels using the residual redundancies," in *Proc. Allerton Conf. Communication, Control, Computing*, 2001, pp. 1306–1315.
[13] R. Link and S. Kallel, "Markov model aided decoding for image transmission using soft-decision feedback," *IEEE Trans. Image Processing*, vol. 9, pp. 190–196, Feb. 2000.
[14] D. J. Miller and M. Park, "A sequence-based approximate MMSE decoder for source coding over noisy channels using discrete hidden Markov models," *IEEE Trans. Commun.*, vol. 46, pp. 222–231, Feb. 1998.
[15] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Trans. Commun.*, vol. 48, pp. 1–6, Jan. 2000.
[16] ——, "Improved image decoding over noisy channels using minimum mean-squared estimation and a Markov mesh," *IEEE Trans. Image Processing*, vol. 8, pp. 863–867, June 1999.
[17] B. D. Pettijohn, M. W. Hoffman, and K. Sayood, "Joint source/channel coding using arithmetic codes," *IEEE Trans. Commun.*, vol. 49, pp. 826–836, May 2001.
[18] N. Phamdo and N. Farvardin, "Optimal detection of discrete Markov sources over discrete memoryless channels—Application to combined source channel coding," *IEEE Trans. Inform. Theory*, vol. 40, pp. 186–193, Jan. 1994.
[19] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[20] R. C. Reininger and J. D. Gibson, "Backward adaptive lattice and transversal predictors in ADPCM," *IEEE Trans. Commun.*, vol. COM-33, pp. 74–82, Jan. 1985.
[21] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. Commun.*, vol. 39, pp. 838–846, June 1991.
[22] M. Skoglund, "Soft decoding for vector quantization over noisy channels with memory," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1293–1307, Apr. 1999.

David J. Miller (M'95) received the B.S.E. degree from Princeton University, Princeton, NJ, in 1987, the M.S.E. degree from the University of Pennsylvania, Philadelphia, in 1990, and the Ph.D. degree from the University of California, Santa Barbara, in 1995, all in electrical engineering.

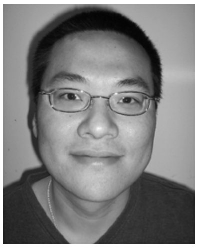
From January 1988 through January 1990, he was with General Atronics Corp., Wyndmoor, PA. From September 1995 to July 2001, he was Assistant Professor of Electrical Engineering at The Pennsylvania State University, University Park. He is now an Associate Professor of Electrical Engineering at The Pennsylvania State University. His research interests include statistical pattern recognition, machine learning, source coding, coding over noisy channels, and image and video coding.

Dr. Miller received the National Science Foundation CAREER Award in 1996. Since 1997, he has been a Member of the Neural Networks for Signal Processing Technical Committee within the IEEE Signal Processing Society. He was General Cochair for the 2001 IEEE Workshop on Neural Networks for Signal Processing. In 2004, he was appointed an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING.

Elias Carotti (S'01) was born in Cuneo, Italy, in 1976. He received the Laurea degree in computer engineering from the Politecnico di Torino, Torino, Italy, in 2000. He is currently working toward the Ph.D. degree in the Dipartimento di Automatica e Informatica, Politecnico di Torino.

In 2000, he was with S.S.B. in Turin, Italy. His primary research interests include image and video coding and coding over noisy channels. He collaborates with the Center for Wireless Multimedia Communications (CERCOM) of the Politecnico di Torino.

Mr. Carotti is a student member of the IEEE Signal Processing Society.



Yu-Wei Wang received the B.S.E.E. degree from Tennessee Technological University, Cookeville, in 1997, and the M.S.E.E. and Ph.D. degrees from Pennsylvania State University, State College, in 1999 and 2004, respectively, all in electrical engineering.

He is currently a Digital Color Image Scientist with Hewlett-Packard, Fort Collins, CO. His research interests include source coding, coding over noisy channels, and image and video coding.

Dr. Wang has been a member of Eta Kappa Nu (electrical engineering honor society) and Kappa Mu Epsilon (mathematics honor society) since 1995.

Juan Carlos De Martin (S'93–M'96) received the Laurea degree in electronic and computer engineering in 1991, and the Ph.D. degree in electronic and telecommunications engineering in 1996, from the Politecnico di Torino, Torino, Italy.

Between 1993 and 1995, he was a Visiting Researcher at the Signal Compression Laboratory, University of California, Santa Barbara, where he worked on low bit-rate speech coding. From 1996 to 1998, he was a Member of Technical Staff at the Media Technologies Laboratory, Texas Instruments Inc., Dallas, where he carried out research in the areas of speech coding, voice over IP, and satellite telephony. In 1998, he joined the IEIIT-CNR (National Research Council of Italy) at the Politecnico di Torino, where he is currently a Principal Research Scientist and an Adjunct Professor. His main research interests are in source and channel coding, speech, audio and video processing and compression, and multimedia transmission over packet networks.

Dr. De Martin is a member of the Multimedia Communications Technical Committee within the IEEE Communications Society, and a member of the IEEE Signal Processing Education Technical Committee.