NOISE ROBUST SPEECH CODING AT VERY LOW BIT RATES

Xiaoqiang Xiao* *and Robert M. Nickel*$^\#$

Department of Electrical Engineering*
The Pennsylvania State University
University Park, PA 16802, USA
xxx106@psu.edu

Department of Electrical Engineering$^\#$
Bucknell University
Lewisburg, PA 17837, USA
robert.nickel@bucknell.edu

ABSTRACT

We propose a new method for noise robust encoding of speech at very low bit rates. The method constitutes an extension to common speech-recognition/speech-resynthesis schemes, which have become feasible in recent years due to advances in speech recognition and artificial speech synthesis. Most such methods, however, suffer from a significant performance degradation in acoustic environments with background noise.

Our proposed procedure is novel since speech enhancement capabilities are built directly into the coding paradigm. Denoising and coding are accomplished jointly by utilizing a statistical description of the parameter space of an underlying speech model (i.e. speech inventory). We conducted experiments with a dedicated speaker in acoustic environments with a signal-to-noise ratio of 10dB. The proposed method was able to improve the perceptual quality of the encoded speech signal by 30% in PESQ measure at an average rate of just under 1.5 kbit/sec.

1. INTRODUCTION

Most known algorithms for the encoding of speech signals at low bit rates fall into either of two categories: (1) parametric coders and (2) waveform inventory coders. Parametric coders analyze the incoming speech signal according to a parametric speech production model (such as an autoregressive production model or a time-varying sinusoidal signal model) and encode and transmit the model parameters across the channel. At the receiver the signal is resynthesized from the encoded parameters through the model.

Classic examples for parametric coders are linear prediction based schemes [1] and harmonic decomposition based schemes [2]. Technically feasible coding methods with a reliable performance according to these two fundamental coding paradigms were developed in the 1980s. Prominent examples are the code excited linear prediction (CELP) approach by Schroeder and Atal [1] and the sinusoidal decomposition approach by McAulay and Quatieri [2]. Since the 1980s many improvements upon these two fundamental methods have been accomplished. Notable examples are the 2400 bits/sec mixed excitation LPC vocoder (MELP) by McCree and Barnwell [3] and Supplee *et. al.* [4] and the NATO-STANAG 4479 improvement of the LPC-10 approach by Mouy *et. al.*, which operates at bit rates as low as 800 bits/sec [5].

More recently waveform inventory coders have become technically feasible at low bit rates [6, 7]. Waveform inventory coding is motivated by a speech-recognition/speech-resynthesis paradigm with an inventory style speech-resynthesis mechanism [7]. Prominent examples for waveform inventory coding are the 1000 bits/sec scheme developed by Lee and Cox [6] and the 400 bits/sec scheme proposed by Baudoin and El Chami [8]. The advantage of the waveform inventory approach is that, if the codec is trained for a dedicated speaker, the resulting speech is of significantly more natural quality than for parametric approaches.

The disadvantage of many very low bit rate speech codecs is, however, that their performance degrades rapidly with increasing levels of background noise. The approach proposed in this paper improves the noise robustness of a very low rate speech codec by building a denoising method into the heart of the procedure. Our method is motivated by the waveform inventory based codec proposed by Lee and Cox in 2001 [6] and a novel speech enhancement procedure published by our research group in 2008/2009 [9, 10]. The disadvantage of the method of Lee and Cox is that it relies on a prosodic analysis of the incoming speech signal, which is potentially problematic under noisy conditions. In our method we have removed the prosodic analysis in favor of a more refined speech inventory, in conjunction with a statistical model of the underlying parameter space. As a result, our method has the ability to jointly encode and enhance an incoming speech signal. The price for the speech enhancement capability is a moderate increase in average bit rate from around 1000 bits/sec to just under 1500 bits/sec.

2. METHODS

To discuss the encoding and decoding methods concisely it is necessary to introduce some mathematical notation. We assume that we have a large record, i.e. an *inventory*, of prerecorded speech $s[n]$ from our targeted speaker. The data is sampled at 8 kHz with a fine quantization granularity. We define a segment vector $\mathbf{s}[n]$ as a collection of 160 successive samples starting at *any* arbitrary time index n.

$$\mathbf{s}[n] = [\quad s[n] \quad s[n+1] \quad \dots \quad s[n+159] \quad]^{\mathrm{T}} \tag{1}$$

We also assume that we have a mapping $k = \mathrm{cmap}(n)$ that assigns every frame $\mathbf{s}[n]$ (for *every* time index n) *uniquely* to one of 50 frame clusters $\mathbb{S}_k$. Each frame cluster collects all inventory frames that belong to a cluster-specific *phonemic* function[1]. The clusters can be generated with an automated design procedure from the given inventory[2]. A detailed description of how to define $\mathrm{cmap}(n)$ and how to generate the clusters can be found in [9].

$$\mathbb{S}_k = \{\, \mathbf{s}[n] \,|\, k = \mathrm{cmap}(n) \,\} = \{\, \mathbf{s}_1^k, \mathbf{s}_2^k, \dots, \mathbf{s}_{M_k}^k \,\} \tag{2}$$

[1] We are using the term *phonemic function* in reference to a general, function carrying unit of a language. The group *may* or *may not* match with an actual *phoneme* defined for that language.

[2] The design procedure is fully automated and does not require any manual tuning and/or other human intervention.

It is assumed that the set of all frames $\mathbf{s}_m^k$ of cluster $\mathbb{S}_k$ is organized in an unspecified but fixed sequential order. The number M_k of frames in each cluster may vary. In our experiments (see section 3) the average number of frames-per-cluster was around 400,000.

Besides the organization of our inventory into clusters $\mathbb{S}_k$ we also need column vectors $\bar{\mathbf{c}}_k$ of *mel* frequency cepstral coefficients (MFCCs, [9]) that represent an average of the MFCCs of the frames of cluster k under noisy conditions. Again, the details of the computation of the $\bar{\mathbf{c}}_k$ is comprehensively described in [9].

In our coding approach we assume that we are operating on a noisy input signal $x[n]$ that has been contaminated with a known noise type at a reasonably constant signal-to-noise ratio (10dB jet cockpit noise in our experiments). The availability of training noise for our procedure allows for the preprocessing of the input signal $x[n]$ with a pre-whitening filter. The details are summarized in [10]. We use $\hat{x}[n]$ to denote the output of the pre-whitening filter. For the sake of a concise discussion we will also make use of the notation $\tilde{x}[n]$ to indicate the *unknown* underlying *clean* speech input.

Unlike the segmentation of our inventory, which operates on a 159 samples overlap, we are using only an 80 samples overlap (i.e. a 50% overlap) to segment our input signals.

$$\mathbf{x}_i = [\quad x[80i] \quad x[80i+1] \quad \ldots \quad x[80i+159] \quad]^{\mathrm{T}} \qquad (3)$$

Index $i = 0, 1, 2, \ldots$ indicates the input frame number. Symbols $\hat{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}$ are defined analogously with respect to the pre-whitened signal $\hat{x}[n]$ and the clean signal $\tilde{x}[n]$.

The coding paradigm of our approach is best summarized with the following diagram:

$$x[n] \xrightarrow[\text{filtering}]{\text{framing}} \hat{\mathbf{x}}_i \xrightarrow{\text{analysis}} g_i \cdot \mathbf{s}_{m_{\text{opt}}(i)}^{k_{\text{opt}}(i)} \xrightarrow{\text{resynthesis}} y[n] \qquad (4)$$

An incoming noisy input signal $x[n]$ is pre-whitened and segmented into a sequence of $\hat{\mathbf{x}}_i$ vectors. An analysis procedure then finds a frame $\mathbf{s}_{m_{\text{opt}}(i)}^{k_{\text{opt}}(i)}$ in our inventory that is, in a probabilistic sense, similar to the underlying clean speech frame $\tilde{\mathbf{x}}_i$. A gain factor g_i is estimated to account for possible magnitude discrepancies between $\tilde{\mathbf{x}}_i$ and $\mathbf{s}_{m_{\text{opt}}(i)}^{k_{\text{opt}}(i)}$. The parameters $k_{\text{opt}}(i)$ (cluster index), $m_{\text{opt}}(i)$ (sub-frame index), and g_i (gain) are encoded and sent across the channel. At the receiver we are concatenating the scaled inventory frames with a 50% overlap and a post-processing procedure to resynthesize the desired output $y[n]$. A block diagram of the proposed coding procedure is shown in figure 1.

In the following three subsections we discuss the computation and encoding of the three parameters $k_{\text{opt}}(i)$, $m_{\text{opt}}(i)$, and g_i. The resynthesis step at the receiver is considered in section 2.4.

2.1 Cluster Index Computation and Encoding

We begin by computing *mel* frequency cepstral coefficients $\mathbf{c}_i = \text{MFCC}\{\mathbf{x}_i\}$ for every incoming noisy frame $\mathbf{x}_i$. The MFCCs are then compared to every MFCC cluster representative $\bar{\mathbf{c}}_k$ with the following distance measure[3]:

$$d(i,k) = \|\mathbf{c}_i\| \cdot \left(1 - \frac{\mathbf{c}_i^{\mathrm{T}}\,\bar{\mathbf{c}}_k}{\|\mathbf{c}_i\| \cdot \|\bar{\mathbf{c}}_k\|}\right) \qquad (5)$$

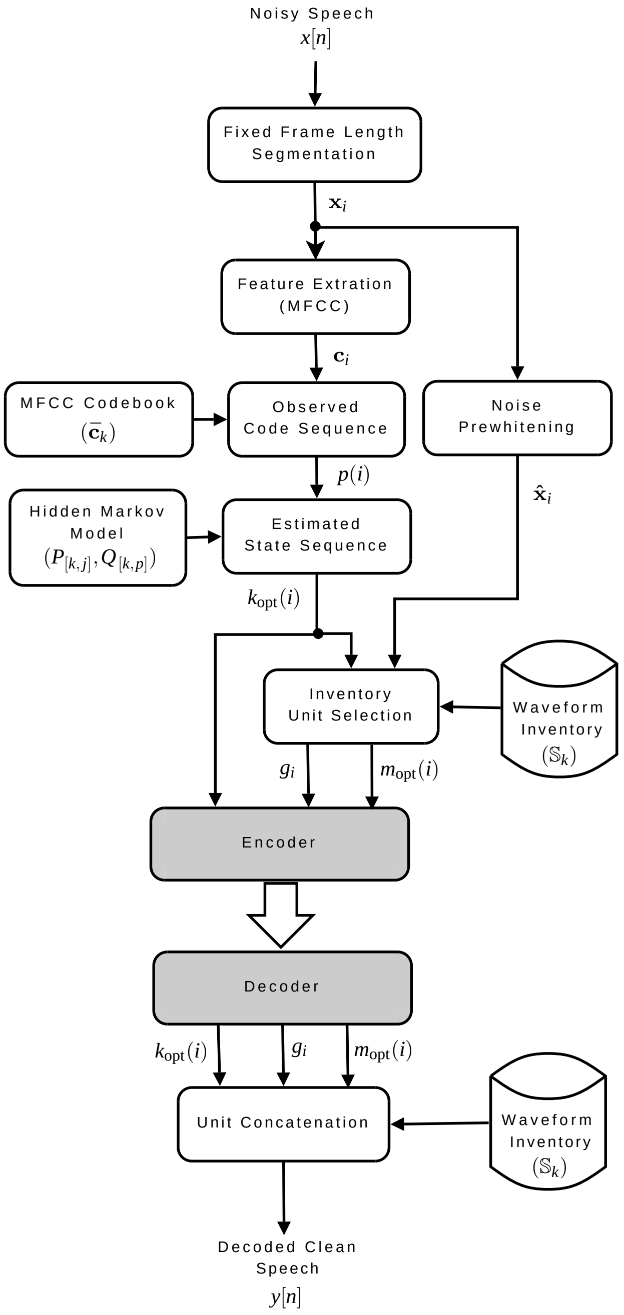

Figure 1. A block diagram of the proposed speech coding method.

The MFCCs are used to formulate a cluster membership hypothesis by finding the cluster p (as a function of i) with the smallest distance d between $\mathbf{c}_i$ and $\bar{\mathbf{c}}_p$.

$$\mathbf{x}_i \to p(i) \quad \text{if} \quad d(i,p) = \min_{k=1\ldots50} d(i,k) \qquad (6)$$

We refer to a sequence Φ of observed hypothetical cluster memberships as the *observed code sequence.*

$$\Phi = [\, \mathbf{x}_0 \to p(0),\ \mathbf{x}_1 \to p(1),\ \mathbf{x}_2 \to p(2),\ \ldots\,] \qquad (7)$$

Similarly, we can argue that there exists a "true" sequence Ψ of underlying cluster memberships of the corresponding clean frames $\tilde{\mathbf{x}}_i$.

$$\Psi = [\, \tilde{\mathbf{x}}_0 \to k(0),\ \tilde{\mathbf{x}}_1 \to k(1),\ \tilde{\mathbf{x}}_2 \to k(2),\ \ldots\,] \qquad (8)$$

[3] The distance measure proposed in equation (5) is more robust under the considered noise conditions than Euclidean distances [9].

We refer to Ψ as the *true underlying state sequence.* The true underlying state sequence is, of course, not known. We can, however, estimate it by maximizing the *a posteriori* probability $\text{Prob}[\Psi\,|\,\Phi]$.

$$\hat{\Psi}_{\text{opt}} = \underset{\text{over all }\Psi}{\arg\max}\ \text{Prob}[\Psi\,|\,\Phi] \tag{9}$$

The computation of $\text{Prob}[\Psi\,|\,\Phi]$ becomes possible if we know the *state transition probabilities*

$$P_{[k,j]} = \text{Prob}[\tilde{\mathbf{x}}_{i+1} \rightarrow j\,|\,\tilde{\mathbf{x}}_i \rightarrow k] \tag{10}$$

and the *observation code probabilities*

$$Q_{[k,p]} = \text{Prob}[\mathbf{x}_i \rightarrow p\,|\,\tilde{\mathbf{x}}_i \rightarrow k] \tag{11}$$

for $k, j, p = 1 \ldots 50$. Both $P_{[k,j]}$ and $Q_{[k,p]}$ can be estimated from our inventory under clean and noisy conditions. The details are described in [9]. The maximization of equation (9) is readily accomplished with the Viterbi algorithm. The estimated hidden state sequence

$$\hat{\Psi}_{\text{opt}} = [\ \tilde{\mathbf{x}}_0 \rightarrow k_{\text{opt}}(0),\ \tilde{\mathbf{x}}_1 \rightarrow k_{\text{opt}}(1),\ \ldots\ ] \tag{12}$$

provides us with the desired cluster indices $k_{\text{opt}}(i)$.

Encoding of the $k_{\text{opt}}(i)$ can be accomplished in a recursive fashion. With $P_{[k,j]}$ we know the followup probability from every state at frame i to the next state at frame $i+1$. Instead of defining a fixed code word for each state, we are defining a flexible length code word for each of the $50 \times 50 = 2500$ possible followup scenarios. For a fixed k we can use the probabilities $P_{[k,j]}$ for j=1…50 to design a Huffman code [11] that minimizes the expected rate. As a result, we obtain a 50×50 "bit-matrix" of codewords that can be used at the transmitter for encoding and at the receiver for decoding. The code word for the cluster index $k_{\text{opt}}(i+1)$ at frame $i+1$ is therefore a function of the cluster index $k_{\text{opt}}(i)$ at frame i, as defined through the "bit-matrix."

In our experiments (see section 3) we found that the expected rate for the encoding of the $k_{\text{opt}}(i)$ under the assumption of a uniform cluster probability was 2.817 bits/frame. The average rate measured on our testing sets, however, was *significantly lower* due to highly non-uniform cluster probabilities.

2.2 Sub-Frame Index Computation and Encoding

The computation of the sub-frame indices $m_{\text{opt}}(i)$ requires the definition of a concatenation similarity between two frame vectors $\mathbf{x}$ and $\mathbf{s}$ as:

$$\text{csim}(\mathbf{x},\mathbf{s}) = \frac{\sum_{m=1}^{80}[\mathbf{x}]_{80+m}\cdot[\mathbf{s}]_m}{\sqrt{\sum_{m=1}^{80}([\mathbf{x}]_{80+m})^2\cdot\sum_{m=1}^{80}([\mathbf{s}]_m)^2}}. \tag{13}$$

We use the notation $[\mathbf{x}]_i$ to indicate the i^{th} element of vector $\mathbf{x}$. The concatenation similarity is normalized between -1 and +1 and provides information about structural similarity between the second half of a frame $\mathbf{x}$ and the first half of a followup frame $\mathbf{s}$.

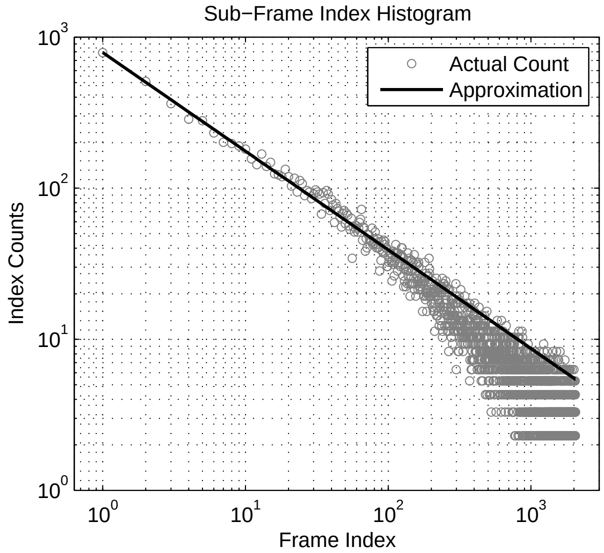

Figure 2. A logarithmic representation of the event count histogram of sub-frame indices.

Similarly to the encoding of $k_{\text{opt}}(i)$, we are using a recursive strategy for the encoding of $m_{\text{opt}}(i)$ which relies on the availability of the decoded frame vector $\mathbf{s}^{k_{\text{opt}}(i-1)}_{m_{\text{opt}}(i-1)}$ from the previous frame $i-1$. Given $m_{\text{opt}}(i-1)$, $k_{\text{opt}}(i-1)$, and $k_{\text{opt}}(i)$ we can rearrange the sequence of elements in $\mathbb{S}_{k_{\text{opt}}(i)}$ with a permutation function $\mu(q)$ for $q = 1 \ldots M_{k_{\text{opt}}(i)}$ such that the concatenation similarity with the known previous frame is monotonically decreasing[4].

$$\text{csim}(\mathbf{s}^{k_{\text{opt}}(i-1)}_{m_{\text{opt}}(i-1)}, \mathbf{s}^{k_{\text{opt}}(i)}_{\mu(q)}) > \text{csim}(\mathbf{s}^{k_{\text{opt}}(i-1)}_{m_{\text{opt}}(i-1)}, \mathbf{s}^{k_{\text{opt}}(i)}_{\mu(q+1)}) \tag{14}$$

We, furthermore, generate an ordered subset $\bar{\mathbb{S}}_{k_{\text{opt}}(i)}$ of the total inventory $\mathbb{S}_{k_{\text{opt}}(i)}$ in cluster $k_{\text{opt}}(i)$ via

$$\bar{\mathbb{S}}_{k_{\text{opt}}(i)} = \{\,\mathbf{s}^{k_{\text{opt}}(i)}_{\mu(1)}\,,\,\mathbf{s}^{k_{\text{opt}}(i)}_{\mu(2)}\,,\,\ldots\,,\,\mathbf{s}^{k_{\text{opt}}(i)}_{\mu(2048)}\,\}. \tag{15}$$

Note that the subset $\bar{\mathbb{S}}_{k_{\text{opt}}(i)}$ and the permutation function $\mu(q)$ are both available at the transmitter *and* the receiver. We found that a limitation of the full inventory[5] to only 2048 best matches in concatenation similarity is sufficient for the targeted coding quality.

We proceed by identifying the frame $\mathbf{s}^{k_{\text{opt}}(i)}_{\mu(q)}$ in $\bar{\mathbb{S}}_{k_{\text{opt}}(i)}$ that best resembles the pre-whitened input frame $\hat{\mathbf{x}}_i$. Given an inventory vector $\mathbf{s}$ and a matrix $\mathbf{H}$ that models the pre-whitening filter operation[6] we can define a similarity measure between $\hat{\mathbf{x}}_i$ and $\mathbf{s}$ as follows:

$$\sigma(\hat{\mathbf{x}}_i,\mathbf{s}) = \frac{\hat{\mathbf{x}}_i^{\text{T}}\,\mathbf{H}\,\mathbf{s}}{\|\,\mathbf{H}\,\mathbf{s}\,\|} \tag{16}$$

The frame in $\bar{\mathbb{S}}_{k_{\text{opt}}(i)}$ that best matches $\hat{\mathbf{x}}_i$ is chosen to represent the input frame i.

$$q_{\text{opt}} = \underset{q=1\ldots2048}{\arg\max}\,\sigma(\hat{\mathbf{x}}_i, \mathbf{s}^{k_{\text{opt}}(i)}_{\mu(q)})$$

[4] The probability of two different frames in a cluster to have the exact same concatenation similarity is zero.

[5] The full inventory contains in average 400,000 frames per cluster.

[6] See [10] for the details on matrix $\mathbf{H}$.

The optimal sub-frame index $m_{\text{opt}}(i)$ for frame i is then found as $m_{\text{opt}}(i) = \mu(q_{\text{opt}})$. Note that we can encode $m_{\text{opt}}(i)$ indirectly through q_{opt}, since $\mu(q)$ is also available at the receiver. The advantage of encoding q_{opt} is two fold: (1) we only need to consider a fixed range of $q_{\text{opt}} = 1 \ldots 2048$, and (2) the probability mass function (PMF) of q_{opt} is non-uniform and we can therefore gain further compression via a Huffman code [11].

Figure 2 shows a representation of the event counts (histogram) for q_{opt} from our experimental training data (see section 3). It is readily visible that, in the double logarithmic representation chosen for figure 2, the data points fall approximately onto a straight line. The deviation from the line at higher values of q_{opt} is explained with the increased variance of the event counts in these areas. An appropriate estimate for a PMF of q_{opt} can be accomplished with a weighted leased squares (WLS) fit between a straight line and the event counts in figure 2. We use $\xi(q_{\text{opt}})$ to indicate the event counts as a function of q_{opt}. A logarithmic index vector γ and a logarithmic event count vector χ are defined as

$$\gamma = [\, \log_{10}(1)\ \log_{10}(2)\ \ldots\ \log_{10}(2048)\,]^{\mathrm{T}} \quad \text{and} \qquad (17)$$
$$\chi = [\, \log_{10}(\xi(1))\ \log_{10}(\xi(2))\ \ldots\ \log_{10}(\xi(2048))\,]^{\mathrm{T}}$$

We, furthermore, need a diagonal weight matrix $\mathbf{W}$ with an exponentially decaying weight on its main diagonal, i.e. $[\mathbf{W}]_{kk} = 10^{1-k}$ for $k = 1 \ldots 2048$. Symbol ι indicates a 2048 dimensional vector in which each element is equal to 1. The slope α and the offset β of the line are estimated as follows:

$$\beta = \frac{\chi^{\mathrm{T}}\mathbf{W}\chi \cdot \gamma^{\mathrm{T}}\mathbf{W}\iota - \chi^{\mathrm{T}}\mathbf{W}\iota \cdot \gamma^{\mathrm{T}}\mathbf{W}\chi}{\iota^{\mathrm{T}}\mathbf{W}\iota \cdot \chi^{\mathrm{T}}\mathbf{W}\chi - (\chi^{\mathrm{T}}\mathbf{W}\iota)^2} \qquad (18)$$

$$\alpha = \frac{\gamma^{\mathrm{T}}\mathbf{W}\chi - \chi^{\mathrm{T}}\mathbf{W}\iota \cdot \beta}{\chi^{\mathrm{T}}\mathbf{W}\chi} \qquad (19)$$

From our training data (see section 3) we found $\alpha = -0.6529$ and $\beta = 2.8974$. The probability mass function for the estimated distribution of q_{opt} can then be written as

$$\mathrm{PMF}(q_{\text{opt}}) = \lambda\, 10^{\beta + \alpha \log_{10}(q_{\text{opt}})} \qquad (20)$$

in which λ is an appropriately chosen constant such that $\sum_{q_{\text{opt}}=1}^{2048} \mathrm{PMF}(q_{\text{opt}}) = 1$. The PMF of q_{opt} can be used to design a Huffman codeword for each q_{opt} index.

In our experiments we found that the average rate of 10.357 bits/frame for the encoding of $m_{\text{opt}}(i)$ via the Huffman code was only slightly less than the rate of a corresponding fixed rate scheme at 11 bits/frame (for $q_{\text{opt}} = 1 \ldots 2048$). The rather modest increase in additional compression may not warrant the complexity of encoding and decoding with a Huffman scheme. However, this small rate decrease enabled us to push the average overall rate below 1.5 kbits/sec.

2.3 Gain Computation and Encoding

The last parameter that needs to be estimated and encoded is the appropriate segment gain g_i. We choose g_i such that the frame energy of the scaled inventory frame matches the estimated energy of the underlying clean signal, i.e. $\|\, g_i \cdot \mathbf{H}\,\mathbf{s}_{m_{\text{opt}}(i)}^{k_{\text{opt}}(i)} \,\|^2 = \|\, \hat{\mathbf{x}}_i \,\|^2 - V^2$ in which V^2 is the expected frame energy of the pre-whitened noise[7]. Coding of g_i can be accomplished in a recursive fashion by considering the half frame norm $E'_{(i-1)}$ of the previous frame and the half frame norm $E''_{(i)}$ of the current frame.

Table I

Huffman Code Lengths of Gain Ratio Codes.

Gain Ratio ε_i	Estimated Probability (%)	Huffman Code Length (bits)
0.25	0.32	8
0.5	1.45	5
0.75	11.66	2
1	70.75	1
1.25	11.5	3
1.5	2.49	4
2	0.88	6
2.5	0.42	8
3	0.18	9
3.5	0.13	9
4	0.23	8

$$E'_{(i-1)} = \sqrt{\sum_{m=1}^{80} \left(\left[g_{i-1} \cdot \mathbf{s}_{m_{\text{opt}}(i-1)}^{k_{\text{opt}}(i-1)} \right]_{80+m} \right)^2} \qquad (21)$$

$$E''_{(i)} = \sqrt{\sum_{m=1}^{80} \left(\left[\mathbf{s}_{m_{\text{opt}}(i)}^{k_{\text{opt}}(i)} \right]_{m} \right)^2} \qquad (22)$$

We define a gain ratio ε_i as:

$$\varepsilon_i = E'_{(i-1)} / (g_i \cdot E''_{(i)}) \qquad (23)$$

The gain ratio can be used as a vehicle to transmit the gain information to the receiver. It is possible to quantize the gain ratio ε_i with only 11 quantization levels without much of a loss in perceptual quality of the reconstructed speech at the receiver. A complete list of the ε_i-quantization levels employed in our experiments, as well as a probability estimate for each level from our training data is listed in table I. Since the probability distribution for each level is highly non-uniform we can, again, use a Huffman code for the transmission of the ε_i information. The lengths of the resulting Huffman codewords for each level are listed in table I as well. At the receiver we reconstruct the targeted frame gain g_i from the decoded ε_i with $g_i = E'_{(i-1)} / (\varepsilon_i \cdot E''_{(i)})$.

From our training data we estimated the expected rate of the gain encoding to be at 1.616 bits/frame. The actual average rate observed with our testing data, however, was slightly higher (see section 3).

2.4 Speech Signal Resynthesis

After decoding the parameters $k_{\text{opt}}(i)$, $m_{\text{opt}}(i)$, and g_i for frame i at the receiver we can begin with the speech signal resynthesis. In a first step we are reconcatenating the segments $g_i \cdot \mathbf{s}_{m_{\text{opt}}(i)}^{k_{\text{opt}}(i)}$ with a simple 50% overlap crossfading procedure. The resulting reconstructed speech signal $\hat{y}[n]$ exhibits still a significant amount of musical noise. The musical noise is (in part) due to phase discontinuities at the frame transition boundaries. To reduce the amount of musical noise we employ a sinusoidal analysis/resynthesis procedure which gracefully interpolates the phases and frequencies from one

[7] We are assuming that the signal and the noise are approximately orthogonal. We set $g_i = 0$ if $V^2 \geq \|\, \hat{\mathbf{x}}_i \,\|^2$. See [10] for details.

frame to the next. The procedure is comprehensively described in [10]. For the purpose of discussion we will call the post-processed output of our coding scheme $y[n]$.

3. EXPERIMENTAL RESULTS

We evaluated the performance of the proposed methods with experiments over the `CMU_ARCTIC` database from the Language Technologies Institute at Carnegie Mellon University[8]. The database was generated specifically for the design of (inventory based) speech synthesis systems. The corpus subset that is used for our study stems from the *US English* male speaker with identifier `BDL`. It contains 1132 phonetically balanced English utterances, most of which are between one and four seconds long. The data is appropriately low-pass filtered and downsampled to a processing sampling rate of 8kHz. We divided the data into three strictly disjoint sets. 1002 utterances were used for the inventory design process (equation 2, [9, 10]). A separate set of 100 utterances was used for the estimation of the gain ratio probabilities (see table I) and the sub-frame index statistics (see figure 2). The remaining 30 utterances were used for codec testing.

Additive noise was taken from the `NOISEX` database from the Institute for Perception-TNO, The Netherlands Speech Research Unit, RSRE, UK[9]. For our experiments we used additive *buccaneer jet cockpit* noise at a signal-to-noise ratio of 10 dB.

The coding results and average rates for the proposed scheme are summarized in table II. We are reporting the estimated rates from the training data and the actually obtained rates from the testing data (in bold face). The total average rate for the proposed scheme is just below 1500 bits/sec with a variation between 1382 bits/sec (low end) and 1512 bits/sec (high end) across different utterances. As can be seen, the estimated total rate (training) and the actually observed total rate (testing) are quite similar. The estimated rate for the cluster index was somewhat higher than the actual rate. The discrepancy is due to an (unrealistic) assumption of equal cluster probability for the estimated rate. Similarly, there is a seemingly significant discrepancy between the estimated rate for the gain and its actual rate. It should be pointed out, though, that the rate difference amounts to less than one bit per frame, which is well within the expected variability for a flexible length code.

Lastly, an objective quality assessment was performed with the *Perceptual Evaluation of Speech Quality* (PESQ) measure. The PESQ measure is an ITU recommendation developed by Rix *et. al.* [12]. It is reported to correlate very well with *subjective quality* of speech. In our experiments, the average PESQ measure across all input testing utterances $x[n]$ amounted to 2.02. The average PESQ measure across all corresponding output signals $y[n]$ amounted to 2.64. We observed, therefore, an improvement of around 30% in perceptual quality as measured by the PESQ standard.

4. CONCLUSIONS

We presented a new method for joint coding and denoising of speech at an average bit rate of 1500 bits/sec. The approach is based on an *inventory style* speech analysis/resynthesis scheme that utilizes a statistical analysis of the underlying parameter space. The required statistical descriptions are obtained from noise enrollment and from speaker enrollment. With experiments we have shown that the proposed method significantly improves the perceptual quality of the coded signal.

Table II
Average Bit Rates for Each Parameter.

Parameters	Bit Rate (bits/sec)	
	Training	Testing
Cluster $k_{\mathrm{opt}}(i)$	281.7	**195.8**
Sub-Frame $m_{\mathrm{opt}}(i)$	1023.0	**1035.7**
Gain g_i	161.6	**226.8**
Total Rate	1466.3	**1458.3**

REFERENCES

[1] M. Schroeder and B. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," *Proceedings of ICASSP*, vol. 10, pp. 937–940, Apr 1985.

[2] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744–754, Aug 1986.

[3] A. V. McCree and T. P. Barnwell III, "A mixed excitation LPC vocoder model for low bit rate speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 242–250, Jul 1995.

[4] L. M. Supplee, R. P. Cohn, J. S. Collura, and A. V. McCree, "MELP: the new federal standard at 2400 bps," *Proceedings of ICASSP*, vol. 2, pp. 1591–1594, Apr 1997.

[5] B. Mouy, P. De La Noue, and G. Goudezeune, "NATO STANAG 4479: a standard for an 800 bps vocoder and channel coding in HF-ECCM system," *Proceedings of ICASSP*, vol. 1, pp. 480–483, May 1995.

[6] K. S. Lee and R. V. Cox, "A very low bit rate speech coder based on a recognition/synthesis paradigm," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 482–491, Jul 2001.

[7] K. S. Lee and R. V. Cox, "A segmental speech coder based on a concatenative TTS," *Speech Communication*, vol. 38, pp. 89–100, 2002.

[8] G. Baudoin and F. El Chami, "Corpus based very low bit rate speech coding," *Proceedings of ICASSP*, vol. 1, pp. 792–795, 2003.

[9] X. Xiao, P. Lee, and R. M. Nickel, "Inventory based speech denoising with hidden Markov models," *Proceedings of EUSIPCO*, 2008.

[10] X. Xiao, P. Lee, and R. M. Nickel, "Inventory based speech enhancement for speaker dedicated speech communication systems," *Proceedings of ICASSP, Taipei, Taiwan*, 2009.

[11] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, 1996.

[12] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," *Proceedings of ICASSP*, vol. 2, pp. 749–752, 2001.

[8] The corpus is available at <http://www.festvox.org/cmu_arctic>.

[9] The noise is available at <http://spib.rice.edu/spib/select_noise.html>.