

A Novel Approach to Variable Dimension Vector Quantization of Harmonic Magnitudes

Wai C. Chu

DoCoMo USA Labs – Mobile Media Laboratory
181 Metro Dr, Suite 300, San Jose, CA 95110, U.S.A.
wai@docomolabs-usa.com

Abstract

Harmonic coders explore the harmonic structure of a given signal to represent it efficiently, and have become quite successful recently where several standardized speech coders are based upon it. Quantization of harmonic magnitudes is an integral component of harmonic coders. Since the harmonic magnitude sequences can be considered as vectors of variable dimension, the technique of variable dimension vector quantization (VDVQ) is highly suitable for their representation. In this paper a novel implementation of VDVQ is described; experimental data show the superiority of the technique which is compared to existent methods.

1. Introduction

The term of *harmonic coding* is probably first introduced by Almeida and Tribolet [1], where a speech coder operating at a bit-rate of 4.8 kbit/s is described. For the purpose of this paper we define a *harmonic coder* as any coding scheme that explicitly transmits the fundamental frequency and harmonic magnitudes as part

of the encoded bit-stream. We use the term of *harmonic analysis* to signify the procedure in which the fundamental frequency and harmonic magnitudes are extracted from a given signal.

The harmonic model is an attractive solution to many signal coding applications, with the objective being an economical representation of the underlying signal. Fig.1 shows a popular configuration for harmonic coding, where harmonic analysis is performed to the excitation signal obtained by inverse filtering the input signal through a linear prediction (LP) analysis filter [2]. In the present work we will consider exclusively this configuration of harmonic coder since it has achieved remarkable success and is adopted by various speech coding standards.

In a typical harmonic coding scheme, LP analysis is performed on a frame-by-frame basis; the prediction-error (excitation) signal is computed, which is windowed and converted to the frequency domain via fast Fourier transform (FFT). An estimate of the fundamental period T (or frequency) is found either from the input signal or the prediction error, and is used to locate the magnitude peaks in frequency domain, leading to the sequence

$$x_j, j = 1, 2, \dots, N(T) \quad (1)$$

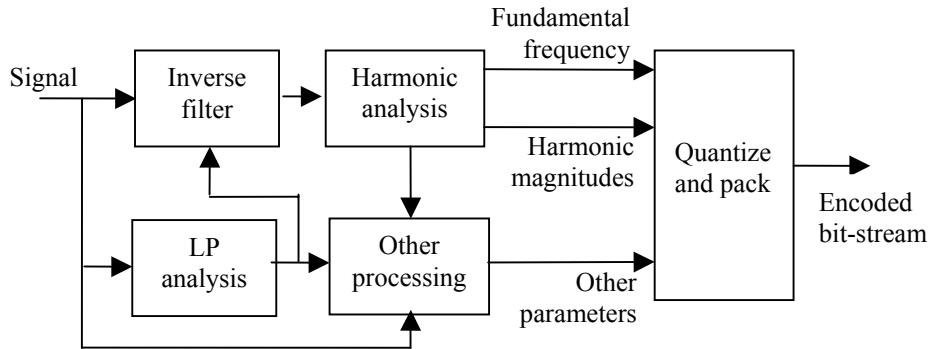


Figure 1. Block diagram of a harmonic encoder.

containing the magnitude of each harmonic; $N(T)$ is the number of harmonics given by

$$N(T) = \lfloor \alpha \cdot T / 2 \rfloor \quad (2)$$

where α is a constant and sometime selected to be slightly lower than one so that the harmonic component at $\omega = \pi$ is excluded. The corresponding frequency values for each harmonic component are

$$\omega_j = 2\pi j / T; j = 1, 2, \dots, N(T). \quad (3)$$

As we can see from (2), $N(T)$ depends on the fundamental period; a typical range for speech coding is [20, 147] encodable with 7 bits, leading to $N(T) \in [9, 69]$ when $\alpha = 0.95$.

Various approaches can be deployed for the quantization of the harmonic magnitude sequence (1). Scalar quantization, for instance, quantize each element individually; however, vector quantization (VQ) is the preferred approach for modern coding paradigms due to improved efficiency [3]. Traditional VQ design is targeted to fixed-dimension vectors. More recently researchers have looked into variable dimension designs, which possess many interesting variations.

Harmonic modeling has exerted a great deal of influence to the development of speech coding algorithms. The federal standard linear prediction coding (LPC) algorithm [4], for instance, is a crude harmonic coder where all harmonic magnitudes are equal for a certain frame; the federal standard mixed excitation linear prediction (MELP) algorithm [5], on the other hand, uses VQ where only the first ten harmonic magnitudes are quantized; the MPEG4 harmonic vector excitation coding (HVXC) algorithm employs an interpolation-based dimension conversion method where the variable dimension vectors are converted to fixed dimension before quantization [6]. Even though the mentioned schemes can be implemented efficiently and produce reasonable quality, unnecessary distortion is introduced to the quantization outcomes since the input vectors are subjected to some sort of transformation which in general create losses. The VDVQ scheme gets around the problem by transforming the codevectors of the quantizer's codebook instead of the input vectors, its principles are explained in the next section.

2. Variable Dimension Vector Quantization

VDVQ [7] [8] represents an alternative for harmonic magnitude quantization, and has many advantages with respect to existent techniques. In this section the basic concepts are presented with an exposition of the nomenclatures involved; the next section contains an

improved quantizer which is a variant of the basic VDVQ.

2.1. Codebook Structure

The codebook of the quantizer is comprised by N_c codevectors \mathbf{y}_i , $i = 0$ to $N_c - 1$ with

$$\mathbf{y}_i^T = [y_{i,0} \quad y_{i,1} \quad \dots \quad y_{i,N_v-1}] \quad (4)$$

where N_v is the dimension of the codevector. Consider the harmonic magnitude vector \mathbf{x} associated with pitch period T having the dimension $N(T)$ given by (2); assuming full search, the following distances are computed

$$d(\mathbf{x}, \mathbf{u}_i); i = 0, \dots, N_c - 1 \quad (5)$$

where

$$\mathbf{u}_i^T = [u_{i,1} \quad u_{i,2} \quad \dots \quad u_{i,N(T)}] \quad (6)$$

and

$$u_{i,j} = y_{i,\text{index}(T,j)}; j = 1, \dots, N(T), \quad (7)$$

with

$$\text{index}(T, j) = \text{round}\left(\frac{(N_v - 1)\omega_j}{\pi}\right) = \text{round}\left(\frac{2(N_v - 1)j}{T}\right); \quad (8)$$

for $j = 1$ to N , $\text{round}(x)$ converts x to the nearest integer. The scheme works as follows: a vector \mathbf{u}_i having the same dimension as \mathbf{x} is extracted from the codevector \mathbf{y}_i by calculating a set of indices using the associated pitch period. These indices point to the positions of the codevector where elements are extracted. The operation is summarized with

$$\mathbf{u}_i = \mathbf{C}(T)\mathbf{y}_i \quad (9)$$

with $\mathbf{C}(T)$ the selection matrix associated with the pitch period T and has the dimension $N \times N_v$. The selection matrix is specified with

$$\mathbf{C}(T) = c_{j,m}^{(T)} \mid_{j=1, \dots, N(T); m=0, \dots, N_v-1}, \quad (10)$$

and

$$c_{j,m}^{(T)} = \begin{cases} 1; & \text{if } \text{index}(T, j) = m \\ 0; & \text{otherwise} \end{cases} \quad (11)$$

In a full search mode, the N_c distances

$$d(\mathbf{x}, \mathbf{u}_i) = d(\mathbf{x}, \mathbf{C}(T)\mathbf{y}_i); i = 0 \text{ to } N_c - 1 \quad (12)$$

are computed, the codevector that minimizes the distance is selected to represent the input vector \mathbf{x} . Thus, the

codebook of the VDVQ scheme contains codevectors that are average spectral magnitudes.

2.2. Codebook Generation

Let's assume that the set of training data

$$\{\mathbf{x}_k, T_k\}, k = 0 \text{ to } N_T-1 \quad (13)$$

is available, with N_T the size of the training set. The set contains vectors of all dimensions associated with a pitch period. The N_c codevectors divide the whole space into N_c cells. The vector \mathbf{x}_k is said to pertain to the i th cell if

$$d(\mathbf{C}(T_k)\mathbf{y}_i, \mathbf{x}_k) \leq d(\mathbf{C}(T_k)\mathbf{y}_j, \mathbf{x}_k) \quad (14)$$

for all $j \neq i$. Thus, given a codebook, partition to the training vectors can be performed so as to obtain the following data

$$\{\mathbf{x}_k, T_k, i_k\}, k = 0 \text{ to } N_T-1 \quad (15)$$

with i_k the index of the cell that \mathbf{x}_k pertains. The task of obtaining (15) is referred to as *nearest neighbor search* [3]. The objective of codebook generation is to minimize the sum of distortion at each cell

$$D_i = \sum_{k, i_k=i} d(\mathbf{x}_k, \mathbf{C}(T_k)\mathbf{y}_i) \quad (16)$$

by optimizing the codevector \mathbf{y}_i , the process is referred to as *centroid computation*. Nearest neighbor search together with centroid computation are the key steps of the Generalized Lloyd algorithm (GLA [3]) and can be used to generate the codebook. Consider the distance definition

$$d(\mathbf{x}_k, \mathbf{C}(T_k)\mathbf{y}_i) = \|\mathbf{x}_k - \mathbf{C}(T_k)\mathbf{y}_i + \mathbf{g}_k \mathbf{1}\|^2 \quad (17)$$

it is assumed in (17) that all elements of the vectors \mathbf{x}_k and \mathbf{y}_i are in the logarithmic domain, and hence (17) is proportional to the spectral distortion (SD [2]) measure, defined by

$$SD = \sqrt{\frac{1}{N(T)} \sum_{j=1}^{N(T)} (20 \log_{10} x_j - 20 \log_{10} y_j)^2} = \sqrt{\frac{1}{N(T)} \sum_{j=1}^{N(T)} (f(x_j) - f(y_j))^2} \quad (18)$$

with x and y the magnitude sequences to be compared and $f(\bullet) = 20 \log(\bullet)$. Note that in (17) $\mathbf{1}$ is a vector whose elements are all 1s with dimension $N(T)$. It can be shown that the optimal gain satisfies

$$\mathbf{g}_k = \frac{1}{N(T_k)} (\mathbf{y}_i^T \mathbf{C}(T_k)^T \mathbf{1} - \mathbf{1}^T \mathbf{x}_k) = \mu_{\mathbf{C}(T_k)\mathbf{y}_i} - \mu_{\mathbf{x}_k} \quad (19)$$

where $\mu_{\mathbf{x}}$ denotes the mean of the elements of the vector \mathbf{x} ; therefore, the optimal gain is given by the difference of the mean between the two vectors. To compute the centroid, we minimize (16) leading to

$$\sum_{k, i_k=i} \mathbf{C}(T_k)^T \mathbf{C}(T_k) \mathbf{y}_i = \sum_{k, i_k=i} \mathbf{C}(T_k)^T \mathbf{x}_k + \mathbf{g}_k \mathbf{C}(T_k)^T \mathbf{1} \quad (20)$$

which can be represented with

$$\Phi_i \mathbf{y}_i = \mathbf{v}_i \quad (21)$$

where

$$\Phi_i = \sum_{k, i_k=i} \mathbf{C}(T_k)^T \mathbf{C}(T_k) \quad (22)$$

and

$$\mathbf{v}_i = \sum_{k, i_k=i} \mathbf{C}(T_k)^T \mathbf{x}_k + \mathbf{g}_k \mathbf{C}(T_k)^T \mathbf{1}. \quad (23)$$

hence the centroid is given by

$$\mathbf{y}_i = \Phi_i^{-1} \mathbf{v}_i. \quad (24)$$

Since Φ_i is a diagonal matrix, its inverse is relatively easy to find. Nevertheless, elements of the main diagonal of Φ_i might contain zeros; in that case alternative methods must be used to solve the centroids.

3. A Novel Approach to VDVQ

The VDVQ system described in the last section is based on finding the index of the codevectors' elements through (8). A more natural way is to use the expression

$$\text{index}(T, j) = \frac{2(N_v - 1)j}{T}; j = 1, \dots, N \quad (25)$$

where rounding is omitted. Thus, the frequency index contains a fractional part and cannot be used to recover the elements of the codevectors. However, it is possible to use interpolation among the elements of the codevectors when the indices contain a nonzero fractional part. We propose to use a first-order linear interpolation method where the vector \mathbf{u} in (6) is found with

$$u_{i,j} = \begin{cases} y_{i, \text{index}(T, j)}; & \text{if } \lceil \text{index}(T, j) \rceil = \lfloor \text{index}(T, j) \rfloor \\ \left(\text{index}(T, j) - \lfloor \text{index}(T, j) \rfloor \right) y_{i, \lceil \text{index}(T, j) \rceil} + \\ \left(\lceil \text{index}(T, j) \rceil - \text{index}(T, j) \right) y_{i, \lfloor \text{index}(T, j) \rfloor}; & \text{otherwise} \end{cases} \quad (26)$$

that is, interpolation is performed between two elements of the codevector whenever the index contains a nonzero fractional part. The operation can also be captured in matrix form as in (9), with the elements of the matrix given by

$$c_{j,m}^{(T)} = \begin{cases} \text{index}(T, j) - \lfloor \text{index}(T, j) \rfloor & \text{if } \lceil \text{index}(T, j) \rceil = m \\ \lceil \text{index}(T, j) \rceil - \text{index}(T, j) & \text{if } \lfloor \text{index}(T, j) \rfloor = m \\ 0; & \text{otherwise} \end{cases} \quad (27)$$

for $j = 1$ to $N(T)$ and $m = 0$ to $N_v - 1$. The same procedure as described in the previous section can be applied for codebook design.

4. Experimental Results

This section summarizes the experimental results in regard to VDVQ as applied to harmonic magnitudes quantization. In order to design the vector quantizers, a set of training data must be obtained. We have selected 100 sentences from the TIMIT database (downsampled to 8 kHz). The sentences are LP-analyzed at 160-sample frames with the prediction-error found. An autocorrelation-based pitch period estimation algorithm is designed. The prediction-error signal is mapped to the frequency domain via 256-sample FFT after Hamming windowing. Harmonic magnitudes are extracted only for the voiced frames according to the estimated pitch period, which has the range of [20, 147] at steps of 0.25; thus, fractional values are allowed for the pitch periods. There are approximately 20000 training vectors in total. A testing data set is similarly extracted from 12 sentences leading to roughly 2500 vectors.

4.1. VDVQ Results

Using the training data set, we designed a total of 30 quantizers at a resolution of $r = 5$ to 10 bits, and codevector dimension $N_v = 41, 51, 76, 101,$ and 129. The average SD is employed as performance measure. For each combination of resolution and dimension, GLA is applied to randomized initial codebooks, a total of 10 random initializations are performed with each followed by 100 epochs of training (one epoch consists of nearest neighbor search followed by centroid computation), the codebook associated with the lowest distortion sum is kept.

Fig.2 shows the SD results for the quantizers. As expected, performances of the quantizers improve with increase in resolution. Training performance can normally be raised with dimension augmentation, however, for low

resolution (such as $r = 5$ to 7), SD has remained almost constant when N_v changes from 76 to 129.

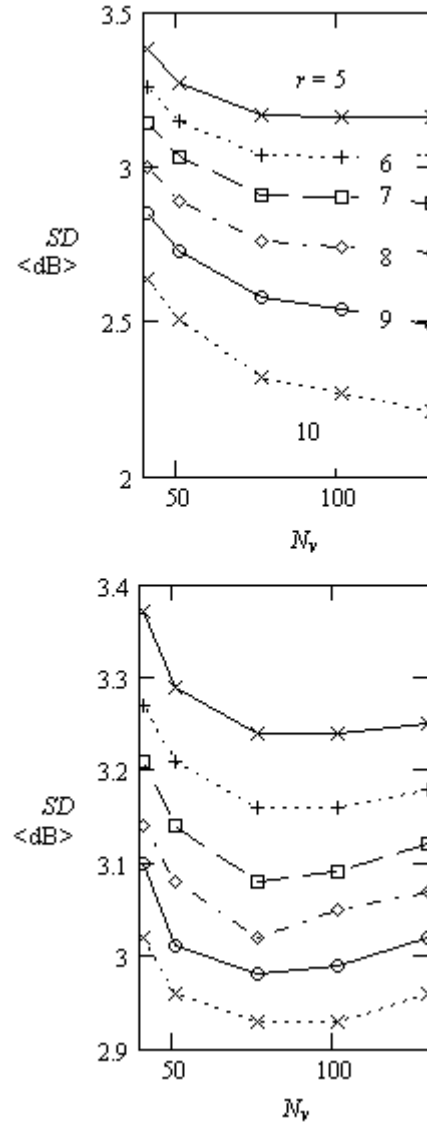


Figure 2. Spectral distortion (SD) as a function of the resolution (r) and codevector dimension (N_v) in VDVQ: training performance (top) and testing performance (bottom).

The testing performance curves show a generalization problem when N_v increases. In most of the cases, increasing N_v beyond 76 leads to a degradation in performance. The phenomenon can be explained from the fact that over-fitting happens for higher dimension, which limits the generalization capability to data outside the training set. For high dimension codevectors, each element of the codevectors are obtained from less training

data, since the ratio between the number of codevectors' elements and the amount of training data is lower; this creates an over-fitting situation where the codevectors tend to over-adapt to some particular training vectors, limiting their ability to generalize to data outside the training set. At lower dimension, the elements of the codevectors are obtained from more training data, producing a final set of codevectors that better represents the average behavior of the data source.

4.2. VDVQ with Interpolation Among the Elements of the Codevectors

The same values of resolution and dimension as for the basic VDVQ are used to design the quantizers where interpolation among the elements of the codebook is incorporated. Fig.3 shows the difference between the SD values, obtained by subtracting the results from the present scheme to that of the basic VDVQ. As we can see, by introducing interpolation among the elements of the codevectors, there is always a reduction in average SD for the training data set, and the amount of reduction tends to be higher for low dimension and high resolution. For testing performance, reduction (by introducing interpolation among the elements of the codevectors) in average SD is lower for high resolution; in certain cases (such as $N_v = 101$ and 129) the average SD actually increased.

4.3. Comparison with Techniques from Standardized Coders

Previously we have seen that by incorporating interpolation among the elements of the codevectors, performance can be improved for VDVQ. However, how well they are when compared to those adopted by standardized coders remain to be seen. In order to compare the various techniques, we implemented some of the quantization methods adopted by the LPC, MELP, and HVXC coders with their performances measured.

For both the LPC and MELP approaches we assume infinite resolution with the performance measured on the training and testing data sets, and is done by following the constraints set forth by the respective standards. In LPC each vector is approximated by its mean leading to a vector where all elements are equal. In MELP only the first ten magnitudes are transmitted.

The HVXC dimension conversion method is also implemented, where full search vector quantizers of dimension equal to 44 at resolution from 5 to 10 bits are designed; to accomplish this task, the variable-dimension training vectors are interpolated to 44 elements, which are used to train the quantizers where GLA is applied, ten random initializations are performed with each followed

by 100 epochs of training; at the end only the best codebook is kept.

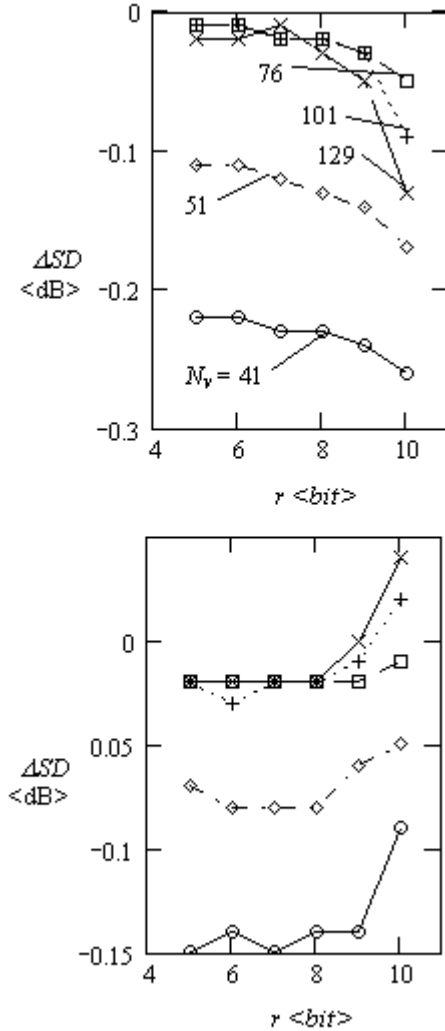


Figure 3. Difference in distortion (ΔSD) as a function of the resolution (r) and codevector dimension (N_v) between two VDVQ schemes: training data (top) and testing data (bottom).

Plots of SD for the discussed schemes appear in Fig.4, where we can see that at $N_v = 41$, the VDVQ schemes have far lower SD. This value of dimension is slightly lower than the value of 44 for the HVXC coder, hence the complexity is lower. Notice that performance curves for LPC and MELP are plotted for reference only, they represent the best possible outcomes under the constraints set forth by each coding algorithm, reachable only at infinite resolution.

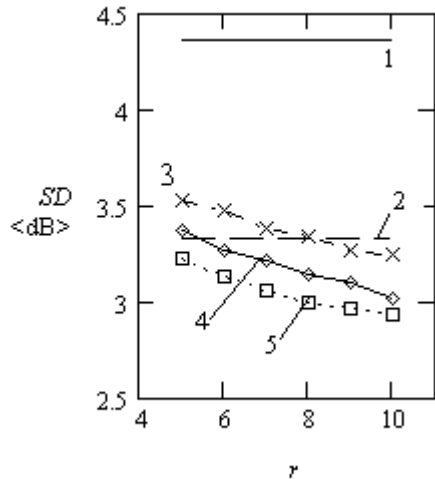


Figure 4. Comparison of testing performance for five schemes: LPC (1), MELP (2), HVXC (3), VDVQ – basic (4), VDVQ – with interpolation among the elements of the codevectors (5). $N_v = 41$ for VDVQ.

5. Conclusion

The advantage of VDVQ comes from the fact that no transformation and hence distortion is introduced to the input vector prior to distance computation during encoding. In both configurations of VDVQ studied, the codebook of the quantizer is adjusted so as to reproduce as accurate as possible a given input vector. This is in sharp contrast to other propositions where the input vector is subjected to some sort of transformation, such as partial quantization in the MELP standard and interpolation for dimension conversion for the HVXC coder. The transformation step introduces a low bound for the distortion that cannot be surpassed with augmentation in resolution.

Another virtue of VDVQ lies in its simplicity, allowing the incorporation of various kinds of structures so as to reduce computational cost. For instance, the popular structures for split VQ, multi-stage VQ, and predictive VQ can be imposed to the VDVQ framework [3]. Moreover, different types of weighting for distance computation can also be incorporated during codebook design. By utilizing interpolation among the elements of the codevectors to the basic VDVQ structure, the performance can generally be raised. The price to pay is a moderate increase in computation; nonetheless it is comparable to that involved with dimension conversion for the HVXC coder.

It is shown that higher resolution normally brings an improvement in quality. However, an excessively long codevector might actually be counter-productive, since

over-fitting conditions arrive with an impairment to the ability to generalize to data outside the training set. The optimal codevector dimension can be found experimentally; and according to the present work, it is near 70 for the basic-VDVQ, and close to 50 when interpolation among the elements of the codevectors is included.

6. References

- [1] L. B. Almeida and J. M. Tribolet, "Nonstationary Spectral Modeling of Voiced Speech", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol.ASSP-31, No.3, pp.664-678, June 1983.
- [2] W. B. Kleijn and K. K. Paliwal, *Speech Coding and Synthesis*, Elsevier Science, 1995.
- [3] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1995.
- [4] T. Tremain, "The Government Standard Linear Predictive Coding Algorithm: LPC-10", *Speech Technology Magazine*, pp.40-49, April 1982.
- [5] A. W. McCree, L. M. Supplee, R. R. Cohn, and J. S. Collura, "MELP: The New Federal Standard at 2400 BPS", *IEEE ICASSP*, pp.1591-1594, 1997.
- [6] M. Nishiguchi, A. Inoue, Y. Maeda, and J. Matsumoto, "Parametric Speech Coding – HVXC at 2.0 – 4.0 KBPS", *IEEE Speech Coding Workshop*, pp.84-86, 1999.
- [7] A. Das, A. Rao, and A. Gersho, "Variable-Dimension Vector Quantization", *IEEE Signal Processing Letters*, Vol.3, No.7, pp.200-202, July 1996.
- [8] A. Das and A. Gersho, "Variable Dimension Spectral Coding of Speech at 2400 BPS and Below with Phonetic Classification", *IEEE ICASSP*, pp.492-495, 1995.