QUANTIZATION OF NON-LINEAR PREDICTORS IN SPEECH CODING

*Jes Thyssen, Henrik Nielsen, and Steffen Duus Hansen**

Tele Danmark Research, DK-2970 Hørsholm, Denmark
*) Electronics Institute, Technical University of Denmark, DK-2800 Lyngby, Denmark

ABSTRACT

In this paper we focus on how to exploit the non-linearities in speech with the main purpose of improving the prediction in speech coders. If non-linearities are absent from speech the linear technique is sufficient, but if non-linearities are present the technique is inadequate and more sophisticated predictors are called for. In our ICASSP-94 paper [1] we gave evidence for non-linearities in speech and presented two non-linear short-term predictors that both were superior to the linear predictor *without* quantization. In this paper we present methods to design vector quantizers for the non-linear predictors and investigate how vector quantization of the non-linear predictors affects prediction. Furthermore, we compare the performance of the quantized non-linear predictors to the performance of traditional quantized linear predictors. The experiments show that 10-bit VQ of the non-linear predictor leads to similar performance as 20-bit state-of-the-art split VQ of the LSP-parameters.

1. INTRODUCTION

This work focusses on how to improve the prediction of speech, with the main purpose of improving speech quality versus bit-rate in predictive coders. By improving the prediction the excitation signal will contain less information and will therefore be easier to code at the same bit rate. Or, the number of bits allocated to the predictor can be lowered and leave more bits to code the excitation with the same predictor performance.

Several different authors have reported and documented non-linearities in speech [2], [3], [1]. In [3] physiological evidence for non-linearities in speech is presented. Keeping that in mind and the experiments in [4], where it was found that Gaussian noise passed through a linear predictor produces *unvoiced speech* of high quality, it seems that there is only need for non-linear prediction of *voiced speech*. Accordingly, in [1] we found that the prediction of *voiced speech* is improved significantly (5 dB) with the use of non-linear short-term predictors (see figure 1; notice that although the non-linear predictor is short-term it is capable of removing most of the pitch information).

These experiments are the work reported in our ICASSP-94 paper [1] and were carried out without

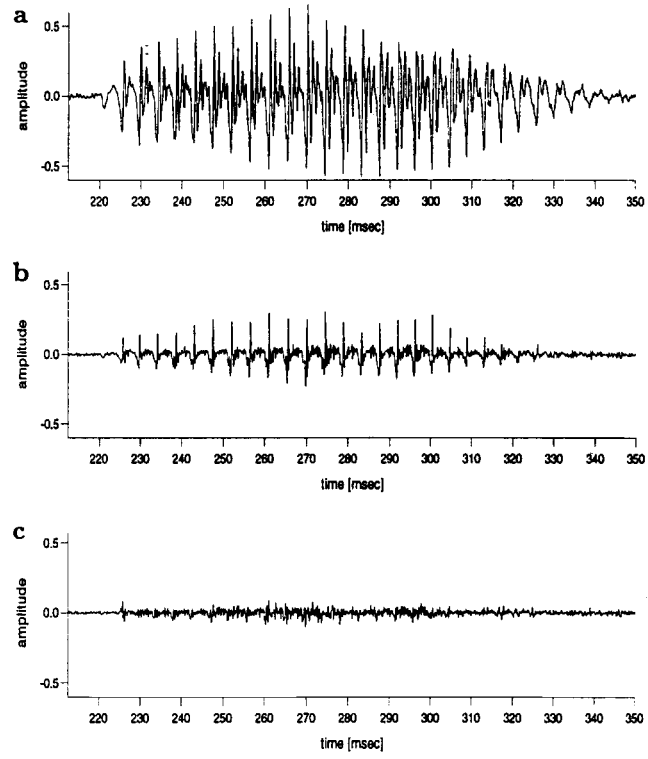

Figure 1: *a: original speech, b: linear short-term residual, c: non-linear short-term residual.*

quantization of the predictors to bring out the potential of non-linear prediction. However, in a speech coder the predictor must be quantized. Accordingly, the performance of the quantized predictors should be examined to allow a fair comparison between linear and non-linear prediction with respect to speech coding.

Throughout this paper the prediction order P is 10 samples and the frame size L is 200 samples, i.e. 25 ms at 8 kHz sampling frequency, which all are parameter values commenly used in speech coding. Furthermore, *closed test* denotes evaluation of a codebook with speech used for training, and *open test* denotes evaluation with speech not used for training. The open test set consists of 60 sentences in total including Danish, British and American English, where speakers and sentences differ from the training set.

0-7803-2431-5/95 $4.00

The paper is organized as follows. In section 2 and 3 the two non-linear predictors, based on the Volterra filter and the neural network, respectively, are presented. This includes both the least squares solutions and design of vector quantizers for the predictors. Section 4 deals with experiments comparing the performance of the quantized short-term predictors. Finally, section 5 contains the conclusions.

2. THE VOLTERRA FILTER

The first non-linear predictor is based on a second-order Volterra filter. The prediction of the speech sample, $x(n)$, is:

$$\begin{aligned}\hat{x}(n) &= \mathbf{H_1}[x(n)] + \mathbf{H_2}[x(n)] \\ &= \sum_{i=1}^{P} h_i \cdot x(n-i) + \sum_{i=1}^{P}\sum_{j=i}^{P} h_{i,j} \cdot x(n-i) \cdot x(n-j)\end{aligned} \tag{1}$$

Like the linear predictor the Volterra filter is a polynomial filter, but in addition to the linear part it has a non-linear part, which in this case is the second-order kernel. The second-order kernel was chosen as a compromise between the number of filter coefficients and the prediction gain.

The least squares solution, i.e. the predictor that minimizes the mean squared error between the original speech, $x(n)$, and the predicted speech, $\hat{x}(n)$, is derived in [1]. The solution is a matrix equation (2):

$$\underline{a}_{opt} = \underline{\underline{M}}^{-1}\underline{m} \tag{2}$$

where $\underline{a}_{opt}$ contains the parameters for $\mathbf{H_1}$ and $\mathbf{H_2}$. The matrix, $\underline{\underline{M}}$, contains second, third, and fourth-order cumulants, and the vector, $\underline{m}$, contains second and third-order cumulants of the time series $\{x(1), x(2), \cdots, x(L)\}$.

2.1. VECTOR QUANTIZER DESIGN

The algorithm to generate codebooks for the Volterra predictor is based on the LBG algorithm (Linde, Buzo, and Gray) [6]. An obvious way to use the LBG algorithm is to let the training set consist of predictors derived from a speech database, use the Euclidean distance measure between predictors when assigning the training set to the codebook, and simply average predictors when calculating new centroids. However, to design codebooks that performs optimally in terms of prediction error we use the following modifications which ensure minimum prediction error in the various steps of the LBG algorithm:

- **The training set** consists of speech frames.
- **The distance measure** between the training vectors (speech frames) and the codebook (predictors) is the residual energy.
- **The new centroids** are calculated to minimize the sum of the residual energies from the training vectors assigned to the given centroid.

The two first items are straightforward. The *centroid calculation* will be derived below. Let $x_{k,1}, x_{k,2}, \ldots, x_{k,N_k}$ be the speech frames that has been matched to the code vector (predictor) $\underline{c}_k$ and thus will be used in the calculation of the new code vector, $\underline{c}_k^{new}$. That is the one that minimizes the sum of the residual energies, $E_{k,tot}$, (3).

$$\begin{aligned}E_{k,tot} &= \sum_{i=1}^{N_k} E_{k,i} \\ &= \sum_{i=1}^{N_k}\sum_{j=1}^{L} [x_{k,i}(j) - \hat{x}_{k,i}(j)]^2 \\ &= \sum_{i=1}^{N_k} E_{x_{(k,i)}} - 2\underline{c}_k^{new}\underline{m}_{(k,i)} + \underline{c}_k^{new}\underline{\underline{M}}_{(k,i)}\underline{c}_k^{new}\end{aligned} \tag{3}$$

where $E_{k,i}$ is the energy of the i^{th} residual, $E_{x_{(k,i)}}$ is the energy, and $\underline{\underline{M}}_{(k,i)}$ and $\underline{m}_{(k,i)}$ are the cumulant matrix and vector, respectively, of the i^{th} speech frame, $x_{(k,i)}$, assigned to $\underline{c}_k$. The cumulant matrix and vector are the same as in (2). The derivative of $E_{k,tot}$ with respect to $\underline{c}_k^{new}$ can be calculated from (3) as:

$$\begin{aligned}\frac{\partial E_{k,tot}}{\partial \underline{c}_k^{new}} &= \sum_{i=1}^{N_k}\left(-2\underline{m}_{(k,i)} + 2\underline{\underline{M}}_{(k,i)}\underline{c}_k^{new}\right) \\ &= 0 \\ &\Rightarrow \left(\sum_{i=1}^{N_k}\underline{\underline{M}}_{(k,i)}\right)\underline{c}_k^{new} = \sum_{i=1}^{N_k}\underline{m}_{(k,i)}\end{aligned} \tag{4}$$

The new centroid is calculated from (4) as:

$$\underline{c}_k^{new} = \left(\sum_{i=1}^{N_k}\underline{\underline{M}}_{(k,i)}\right)^{-1}\sum_{i=1}^{N_k}\underline{m}_{(k,i)} \tag{5}$$

3. THE NEURAL NETWORK

The second non-linear predictor is based on a neural network. Figure 2 shows the architecture of the neural network used as a predictor. It has P input nodes, two hidden layers with 2 nodes in each, and one output node. Each node has a sigmoid transfer function (6), except the output node, which is linear.

$$f(z) = \frac{1}{1 + e^{-\beta z}} \tag{6}$$

The argument z in (6) is a weighted sum of the outputs of the preceding layer, and β is a constant.

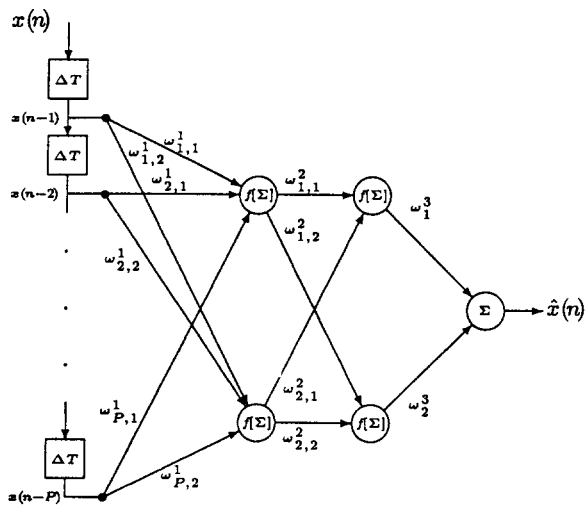

Figure 2: *The Neural Network Predictor.*

The least squares solution cannot be expressed analytically as for the Volterra predictor. Instead, it has to be found iteratively, e.g. with the backpropagation algorithm [5]. We use a minimization algorithm requiring the Jacobian, which can be derived using the idea of the backpropagation algorithm. It should be noted that in a future speech coder no online training of the neural network will be necessary. The predictors in a codebook can simply be applied one by one to the speech frame and finally transmit the index of the predictor which minimizes the residual energy of the frame.

3.1. VECTOR QUANTIZER DESIGN

Like the Volterra predictor the vector quantizer design for the neural network is based on the LBG algorithm. However, using the obvious scheme described at the begining of section 2.1 leads to poor performance of the codebooks. This is mainly because averaging and using the Euclidean distance measure is unsuitable here too. When the predictor is linear these circumstances do not affect the generated codebook to the same extent as for the non-linear predictor. The effect is much larger on the design of codebooks for the neural network predictor than on the codebook design for the Volterra predictor. This is because the Volterra predictor is non-linear in the input (signal values) only, while the neural network predictor is non-linear in both input and coefficients.

To improve the vector quantizer design for the neural network predictor modifications similar to those made in section 2.1 are introduced. The *centroid calculation* is the only item that will be described. Unlike the Volterra predictor in section 2.1 it is impossible to express the solution analytically. Yet letting the cost function be the sum of the residual energies the backpropagation algorithm or an optimization algorithm can be used to calculate the new centroids. Let $x_{k,1}, x_{k,2}, \ldots, x_{k,N_k}$ be the speech frames that have been matched to the code vector (predictor) $\underline{c}_k$ and therefore used to calculate the new code vector, $\underline{c}_k^{new}$. That is the one minimizing the cost function, $E_{k,tot}$, (7).

$$\begin{aligned} E_{k,tot} &= \sum_{i=1}^{N_k} E_{k,i} \\ &= \sum_{i=1}^{N_k} \sum_{j=1}^{L} [x_{k,i}(j) - \hat{x}_{k,i}(j)]^2 \\ &= \sum_{i=1}^{N_k} \sum_{j=1}^{L} \left[x_{k,i}(j) - F(\underline{c}_k^{new}, \underline{x}_{k,i}(j))\right]^2 \end{aligned} \quad (7)$$

where $F(\underline{c}, \underline{x}(n))$ is the output from the neural network described by $\underline{c}$ when the input is the signal samples $\underline{x}(n) = \{x(n-1), x(n-2), \ldots, x(n-P)\}$.

The performance of codebooks generated with the obvious design method and the modified method is shown in figure 3, where the prediction gain versus bits/predictor, i.e. bits/25ms, for both closed and open tests can be seen. From the figure it is obvious that

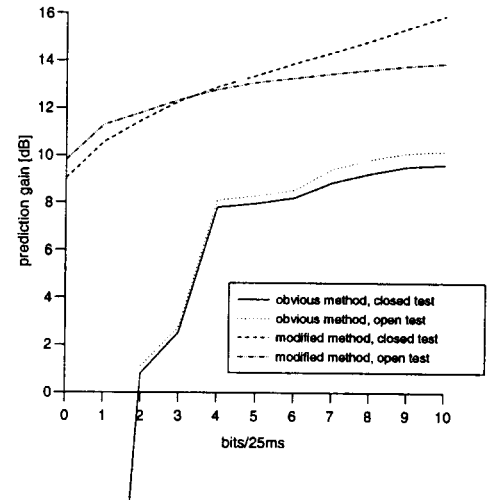

Figure 3: *Obvious and modified VQ performance.*

the modified LBG method performs significantly better than the obvious one. Both methods use a training set of 140 sentences in total with Danish, British and American English utterances from female and male speakers.

4. RESULTS

The first experiment focusses on the size of the training set. For the Volterra predictor the dependency is strong. As the training set is varied from 24 to 210

sentences the improvement in open test prediction gain at 8 bits/predictor is 1.6 dB. For the neural network the dependency is weaker, still the prediction gain improves 0.60 dB, while the linear predictor shows almost no dependency. The dependency increases with the number of coefficients (linear: 10, neural network: 26, Volterra: 65). This confirms what should be expected: the more coefficients the more training examples are needed to achieve a good generalization.

The second experiment compares the performance of the non-linear methods to traditional linear methods at equal bit rate (same number of bits/predictor). In figure 4 the open test performance for single stage VQ of Volterra, neural network, and linear predictors as a function of bits/predictor can be seen. Also a two-stage split VQ of LSP parameters is included to make a comparison to a linear state-of-the-art predictor possible. In this experiment all predictor codebooks are trained with 210 sentences totally, including Danish, British and American English. The cost function used when designing the codebooks is the same for all predictors except for the split VQ of the LSP parameters. In that case the weighted Euclidean distance measure between LSP parameters from [7] is used. Several in-

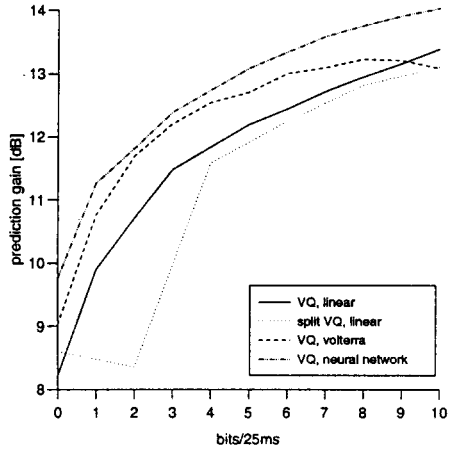

Figure 4: *Open test performance.*

teresting points should be noted from the experiment.

- The open test performance of the non-linear Volterra predictor degrades as the bits/predictor exceed 8. This indicates that the codebook is a poor generalizer.
- The open test performance of the non-linear neural network predictor is superior to the open test performance of all linear methods at any bit rate.
- The open test performance of the non-linear predictor at 10 bits/predictor is identical to 20 bits/predictor split VQ of LSP parameters. Note, 20 bits/predictor is not shown in the figure.

Especially the last point is interesting, since a 20-bit split VQ of LSP parameters is considered sufficient for high quality speech coding. However, it should be noted that the non-linear predictors are incapable of removing the pitch to the same extent as without quantization, see figure 1. Furthermore, the open test of the neural network shows some saturation at higher bit rates as compared to the closed test, see figure 3. This indicates that the VQ of the neural network might be improved.

5. CONCLUSIONS

In this paper we have presented methods to design vector quantizers for two non-linear predictors. One of the main results is that the non-linear predictors require much larger training sets to generalize properly than linear predictors. While the two non-linear predictors perform similarly without quantization, the neural network predictor performs significantly better than the Volterra predictor when quantization is introduced. Thus, the neural network predictor is preferable to the Volterra predictor with respect to speech coding. The better quantization properties of the neural network is largely due to the smaller number of coefficients in the neural network predictor. Furthermore, a 10-bit VQ of the neural network performs approximately 0.75 dB better than a 10-bit VQ of the linear predictor, and most importantly, the performance is similar to a 20-bit state-of-the-art split VQ of the LSP parameters as regards prediction gain. The experiments have shown that the performance of the quantized non-linear predictors might be improved by further work focussing on the generalization. We are currently implementing the non-linear neural network predictor in a low rate speech coder.

6. REFERENCES

[1] Jes Thyssen, Henrik Nielsen, and Steffen Duus Hansen. *Non-Linear Short-Term Prediction in Speech Coding.* Proceedings ICASSP, Vol. 1, 1994, pp. II.185-II.188.

[2] Brent Townshend. *Nonlinear Prediction of Speech.* Proceedings ICASSP, Vol. 1, 1991, pp. 425-428.

[3] H.M. Teager and S.M. Teager. *Evidence for Nonlinear Sound Production Mechanishms in the Vocal Tract.* Speech Production and Speech Modelling. NATO ASI series Vol. 55, Kluwer Academic Publishers 1990.

[4] Gernot Kubin, Bishnu S. Atal, and W. Bastiaan Kleijn. *Performance of Noise Excitation for Unvoiced Speech.* Proceedings IEEE Workshop on Speech Coding for Telecommunications, 1993, pp. 35-36.

[5] Don R. Hush and Bill G. Horne. *Progress in Supervised Neural Networks.* IEEE Signal Processing Magazine, January 1993.

[6] Yoseph Linde, Andrés Buzo, and Robert M. Gray. *An Algorithm for Vector Quantizer Design.* IEEE Transactions on Communications, Vol. 28, January 1980.

[7] K.K. Paliwal and B.S. Atal. *Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame.* Proceedings ICASSP, Vol. 1, 1991, pp. 661-664.