Proc. Biennial Symp. Commun. (Kingston, Ont.), pp. 3-7, June 1998

Improved Pitch Modeling for Low Bit-Rate Speech Coders

Costantinos Papacostantinou and Peter Kabal

Department of Electrical Engineering
McGill University
Montreal, Quebec H3A 2A7

Abstract

The poor modeling of the pitch pulse waveforms during voiced speech contributes to the degradation of CELP coded speech. This problem becomes more acute with the reduction of the number of pulses and other constraints imposed on the fixed codebook part of the excitation. We have developed a Pitch Pulse Averaging (PPA) algorithm to enhance the periodicity of such segments, where during steady state voicing the pitch pulse waveforms in the excitation signal evolve slowly in time. The PPA algorithm extracts a number of such pitch pulse waveforms from the past excitation, aligns them, and then averages them to produce a new pitch pulse waveform with reduced quantization noise. We have simulated and tested our algorithm on a floating point C-simulation of the G.729 8 kbps CS-ACELP coder. The results we present verify that the algorithm has generally improved the periodicity of voiced segments by reducing the average of the weighted mean-squared error.

1 Introduction

In Code-Excited Linear Prediction (CELP) [1] coders the excitation signal is constructed by a linear combination of an adaptive codebook and a fixed codebook contribution [2] which model the periodic and noisy (or stochastic) part of the excitation, respectively. In simple terms, the excitation signal $\hat{r}[n]$ for a given subframe of length N is given as

$$\hat{r}[n] = \beta\hat{r}[n-d] + Gc[n], \qquad n = 0, \ldots, N-1, \quad (1)$$

where $\hat{r}[n-d]$ and $c[n]$ are the adaptive and fixed codebook signals. The constants β and G are their respective gain factors. The adaptive and fixed codebook are searched sequentially to select the codebook entries and scaling factors that minimize the weighted mean-squared difference between the original and synthesized signal. Finally, the resulting optimal excitation, is expressed in vector notation as

$$\hat{\mathbf{r}}_{opt} = \beta_{opt}\boldsymbol{\upsilon}_{opt} + G_{opt}\mathbf{c}_{opt}. \quad (2)$$

During steady state voiced speech, the adaptive codebook contributes a large fraction to the resulting excitation. The purpose of the fixed excitation is to provide the missing part. However, the need for lower bit-rates reduces the number of bits available, thus restricting the fixed codebook to smaller sizes. This results in less accurate waveform matching. It is possible that the selected fixed codebook vector will disturb the periodicity of the resulting excitation signal, or alternatively increase the quantization noise. After the selection of the two contributions, the "noisy" resulting optimal excitation is fed back to the adaptive codebook. As a result, the adaptive codebook is populated with a "noisy" residual and no longer provides the intended purely periodic signal.

Our approach [3] is focused on the adaptive codebook contribution. Based on observations on which the principles of Waveform Interpolation [4] and Pitch Pulse Evolution model reported in [5] are based, we concluded that during steady state voicing, the adaptive codebook should supply a pitch waveform, whose shape changes slowly from pulse to pulse. Thus the abrupt changes that the pitch pulse in the adaptive codebook contribution may undergo, have to be removed before is added to the fixed codebook contribution to form the resulting excitation. This can be achieved by allowing the adaptive codebook to be populated the usual way, and remove the quantization noise from the current optimal codevector based on a relatively long history of pitch pulses.

2 The Pitch Pulse Averaging Technique

In practice, the adaptive codebook is specified as an array of samples, of length at least as large as $D_{max}+N$. The first D_{max} samples represent past constructed optimal excitation and the next N samples represent the excitation for the current subframe, as illustrated in Fig. 1. Setting the length

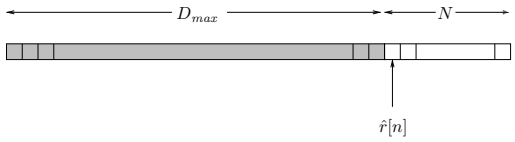


Fig. 1 Adaptive codebook state before excitation selection.

of the past excitation to the longest possible pitch period D_{max}, implies that the past excitation contains at least one pitch pulse waveform. In our approach, the past excitation is extended to contain a number of pitch pulse waveforms, even for the case of having a signal with pitch period as long as D_{max} . The PPA technique can be divided into two steps. First, the evolution of the current pitch pulse waveform is extracted from the relatively long excitation history and second, the noisy component is removed from the current waveform by averaging its evolution.

2.1 Extraction of pitch pulses

The evolution of the pitch pulses is found by identifying and extracting the best match to the target vector of the current subframe, then in a similar way find the match to the first match and so on. Here, a match to an arbitrary time instant is defined as the sample from the past that was selected to minimize the weighted mean-squared error of the synthesized speech at that time instant. This sample is identified by the optimal delay found for that time instant. By identifying the subframe that each time instant belongs to, we could assign an optimal delay to each one of them. This information could then be used to extract the aligned consecutive pitch pulse waveforms and form the pitch pulse evolution.

For example, the simplest case would be to identify the best match to the current subframe, i.e., at time instants $n = 0, \ldots, N-1$, when the delay is greater than the subframe length. All the time instants in this period are assigned the same delay $D^{(0)}$, thus their matched samples are found $D^{(0)}$ samples back. Thus, the best match for time instant $n = q,\ q = 0, \ldots, N-1$ is sample $\hat{r}[q - D^{(0)}]$. This can also be written as

$$S_0[n] = \hat{r}[n - D^{(0)}], \qquad \text{for } n = 0, \ldots, N-1, \tag{3}$$

where $D^{(0)}$ is the optimal integer delay found for the current subframe, after a closed loop search of the adaptive codebook, and $S_0[n]$ is the first extracted waveform. In order to find the second waveform, the match to the time instants that constitute the first match, i.e. time instants $n = q - D^{(0)},\ q = 0, \ldots, N-1$, have to be identified. Here, there might be a case where different time instants are assigned with different delays because they belong into different subframes. Once the subframe(s) that those time instants belong to are identified, their matches can be found by using their assigned delays.

Depth-First search procedure using integer delay values

The approach for extracting the pitch pulse waveforms introduced above uses a *breadth-first* search procedure, where all the elements of one waveform are extracted before moving to the next waveform. This makes this approach rather complex to implement. An alternative approach is to use a *depth-first* search procedure, where all the elements having the same index n in each waveform i are extracted before advancing the index to the next element. The pitch pulse waveforms $S_i[n]$ are now given as:

$$S_i[n] = \hat{r}[n - P(i)], \qquad \begin{array}{l}\text{for } i = 0, \ldots, L, \\ \text{and } n = 0, \ldots, N-1,\end{array} \tag{4}$$

where

$$P(i) = P(i-1) + D^{(k)}, \qquad \text{with } P(0) = D^{(0)}, \tag{5}$$

and

$$k = \begin{cases} 0 & \text{if } (-P(i-1) + n) > -1, \\ \left\lfloor \dfrac{P(i-1) - n - 1}{N} \right\rfloor + 1 & \text{otherwise.} \end{cases} \tag{6}$$

In Eq. (5), $D^{(k)}$ denotes the optimal delay found for the k-th past subframe. The integer L denotes the total number of waveforms extracted. This approach is shown in Fig. 2 for the first and last elements of the first three waveforms in the previous example.

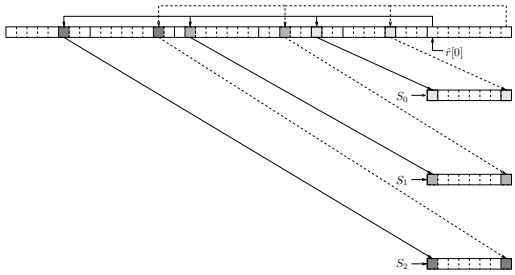


Fig. 2 Pitch pulse waveform extraction using a *depth-first* search procedure. Assumed values: $N = 8$, $D^{(k)} = 11, 10, 12, 12$, for $k = 0, 1, 2$ and 3, respectively.

So far we have assumed that the delay at the current subframe is greater than the subframe length. When the optimal delay found for the current subframe is smaller than the subframe length, $N - D^{(0)}$ samples need to be read from the current subframe in order to completely define the first pitch pulse waveform $\mathbf{S}_0$ (see Eq. (4)). Unfortunately, the excitation at the current subframe is still unknown and thus those samples are not defined. To remedy this problem, we let the G.729 algorithm [6] first form the optimal adaptive codebook vector and then copy this vector to the current subframe. This approach was chosen for its practicality and it was found through simulations that it gives similar results as if the waveform was simply repeated for the undefined samples.

Depth-First search procedure using fractional delay values

The *depth-first* search procedure described earlier, employs integer delays to extract the pitch pulses. Better matching can be achieved if delays of higher resolution are used. Thus, the procedure described above has been modified to operate with fractional delays. The introduction of fractional delays can be done either by using polyphase filters or by interpolating the excitation buffer. Since the first method is rather cumbersome to implement and requires a large amount of computation the latter was preferred.

Fractional delays of resolution I can be implemented by interpolating the excitation buffer by a factor I. Thus, a fractional delay expressed as an integer delay T and a fraction $t/I,\ t = 0, 1, \ldots, I-1$, in the original excitation, is equivalent to an integer delay of $IT + t$ samples in the interpolated excitation. Following this principle, the pitch pulse waveforms are now given as

$$S_i[n] = \hat{r}_{int}[-P(i) + nI], \qquad \text{for } i = 0, \ldots, L, \quad \text{and } n = 0, \ldots, N-1, \tag{7}$$

where

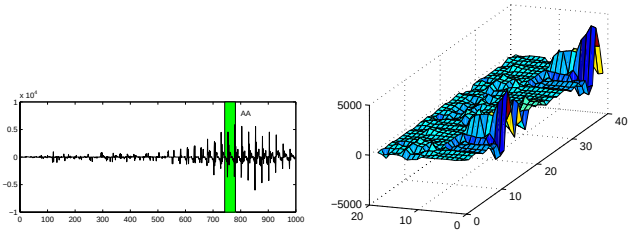
$$P(0) = IT^{(0)} + t^{(0)},$$
$$P(i) = P(i-1) + (IT^{(k)} + t^{(k)}), \tag{8}$$

and

$$k = \begin{cases} 0 & \text{if } (-P(i-1) + In) \geq 0, \\ \left\lfloor \dfrac{P(i-1) - In - 1}{IN} \right\rfloor + 1 & \text{otherwise.} \end{cases} \tag{9}$$

In Eq. (7), $\hat{r}_{int}$ denotes a pointer to the beginning of the interpolated current subframe in the interpolated excitation; that is, to the sample point in the interpolated excitation that corresponds to the first sample of the current subframe in the original excitation. The integer and fractional part of the delay in the k-th subframe are denoted as $T^{(k)}$ and $t^{(k)}$ respectively.

Figures 3(a) and 3(b) show a voiced segment of the excitation corresponding to a female spoken utterance and the extracted waveforms at the indicated subframe respectively. During unvoiced segments of speech, the extracted



(a) Excitation segment.

(b) Extracted waveforms at subframe "AA".

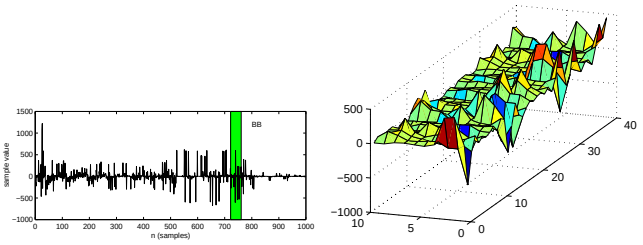
Fig. 3 Extraction of waveforms during voiced segments.

waveforms are more noise-like since pitch periodicity is absent. This case is demonstrated in Figures 4(a)and 4(b).

After the waveforms have been extracted, they are normalized in energy so that the pitch pulses in these waveforms have more uniform amplitudes and thus the weighting applied to each extracted waveform during the averaging procedure is more effective.

2.2 Averaging of pitch pulses

After the pitch pulse waveforms have been extracted and normalized, the noisy component can be removed from the intended adaptive codebook vector by averaging these waveforms. The adaptive codebook vector found for the current



(a) Excitation segment.

(b) Extracted waveforms at subframe "BB".

Fig. 4 Extraction of waveforms during unvoiced segments.

subframe, resembles very closely the first pitch pulse waveform extracted. This is the most recent waveform in the evolution of the pitch pulses and thus should be emphasized most. As the waveforms age in time, their relevance decreases and therefore they should be given less emphasis. Since it is very important that we are be able to control the number of waveforms that are emphasized most, the weighting function is required to have a varying shape. This requirement led us to the choice of a Kaiser window. In continuous time, a Kaiser window is specified by the following equation:

$$w(t) = \begin{cases} \dfrac{I_0(\alpha\sqrt{(1-t^2)})}{I_0(\alpha)} & \text{for } -1 \leq t \leq 1, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

The discrete-time one-sided window of length N_w is obtained by setting

$$t = \frac{n}{(N_w - 1)}, \qquad \text{for } n = 0, \ldots, N_w - 1, \tag{11}$$

which provides the weights for the N_w extracted waveforms. The independent window parameter α determines the shape of the window and its value is estimated empirically. The value of α not only affects the averaged adaptive codebook vector but the population of the adaptive codebook as well, since this vector is replacing the optimum adaptive codebook vector originally estimated by the original algorithm of the coder.

The normalized pitch pulse waveforms $\bar{\mathbf{S}}_i$ are now weighted with their corresponding weights $w[i]$ and added together to form the averaged waveform $\boldsymbol{v}_{av}$, given as

$$\boldsymbol{v}_{av} = \sum_{i=0}^{N_w - 1} w[i]\bar{\mathbf{S}}_i. \tag{12}$$

Before forming the new adaptive codebook vector for the current subframe, the gain difference between the averaged waveform $\boldsymbol{v}_{av}$ and the one intended to be supplied by the original coder, $\boldsymbol{v}_{opt}$, has to be compensated. Note, for the purposes of this paper, the term "original coder" refers to

the unmodified coder. Since the extracted waveforms are not orthogonal, normalization of the energy of the weights does not solve the problem. A simple way to compensate for this difference is to multiply the averaged waveform with a gain-scaling factor given as

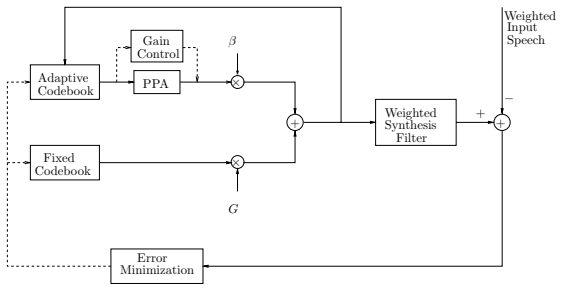
$$g = \sqrt{\frac{\sum_{i=0}^{N-1} v_{opt}^2[n]}{\sum_{i=0}^{N-1} v_{av}^2[n]}}. \tag{13}$$

The gain-scaled averaged waveform $\tilde{\boldsymbol{v}}_{av}$ is given by

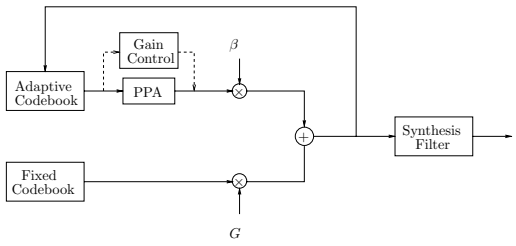
$$\tilde{\boldsymbol{v}}_{av} = g\boldsymbol{v}_{av}. \tag{14}$$

Finally, the vector $\tilde{\boldsymbol{v}}_{av}$ replaces the originally estimated optimal adaptive codebook vector $\boldsymbol{v}_{opt}$ and is subsequently used to calculate the gain β_{opt} and form the adaptive codebook contribution for the current subframe.

The block diagram of Fig. 2.2 indicates how the original codec structure has been modified to accomodate our algorithm.



(a) Encoder structure.



(b) Decoder structure.

Fig. 5 Modified codec structure.

3 Simulation results

The performance of the algorithm was measured using a number of objective measures, such as the cross correlation coefficient between the filtered adaptive codebook vector (zero-state response of the weighted synthesis filter $h[n]$ to $v_{opt}[n]$) and target vector, and the variations of the fixed codebook gain. Since these two objective measures are not very well correlated, i.e., an increase in the correlation coefficient does not always result in a decrease in the fixed codebook gain, more emphasis is given to minimizing the weighted mean-squared error between the synthesized and weighted input speech vectors.

Simulations indicate that the weighted MSE improves during steady-state voicing but not during onsets and end of voicing segments. The algorithm does best during steady state voiced speech because the similarity between successive waveforms is higher during such segments. Any contribution from the fixed codebook which disturbs the periodicity of such segments by inserting pulses of unwanted amplitudes to unnecessary positions, is reduced by the averaging procedure. The best value of α during these segments is low, which implies that the fixed codebook is introducing unwanted noise to the final excitation waveform and there is more need to remove it by the averaging procedure. The situation is reversed during the onsets. In this case the algorithm hinders the attempt of the coder to compensate for the increasing amplitude pitch pulses and ringing at the same time, while having only 4 pulses available for the fixed codevector. Similarly, during unvoiced segments the adaptive codebook essentially acts as a second stochastic codebook, helping the fixed codebook compensate for the highly noisy character of such segments. The averaging of such segments, reduces the noisy character of the adaptive codebook, thus leaving more work to the fixed codebook. Thus for such segments the value of α is high and the overall performance of the algorithm drops below that of the original coder.

The performance of the algorithm during these segments is illustrated in Fig. 6 where the weighted mean squared error is plotted for the original codebook and that modified by the PPA algorithm. During steady state the weighted

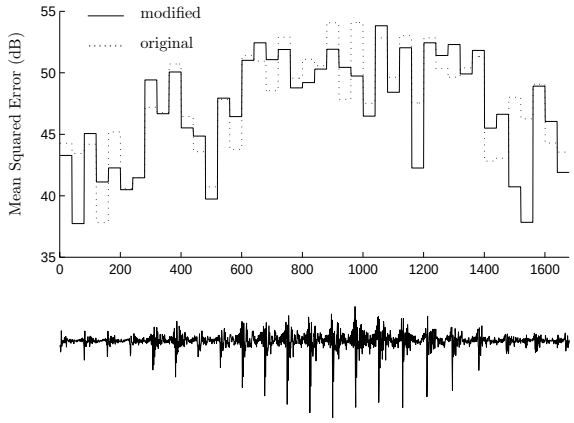


Fig. 6 Weighted Mean Squared Error for a voiced segment.

MSE is lower for the majority of subframes, but this is not always the case during the onset and end of the voiced segment. Nevertheless, the average weighted MSE of the segment shown is lower than the original.

The problem of poorer performance during the onsets and unvoiced segments has been reduced by introducing a simple mechanism to activate the algorithm only during voiced segments, skipping the first few pitch pulses at the onsets. The problem at the ends of voiced segments has not been solved though, since there was no simple and efficient way to detect such segments. Nevertheless the overall performance of the algorithm has improved.

4 Conclusions

In this paper we presented the design and simulation of the Pitch Pulse Averaging (PPA) technique whose goal is to reduce the noise in the pitch pulse waveforms supplied by the adaptive codebook during steady state voiced speech segments. Objective tests verified that the algorithm does best during steady state voiced speech because of the similarity between successive waveforms in such segments.

Better results could be obtained by switching between the PPA technique and the original coder depending on which of the two performs best at each subframe. Using a simple decision mechanism, the cost in terms of complexity would be nearly the same as if the algorithm was continuously active. But an extra bit per subframe should be transmitted to inform the decoder which case should be used at each subframe to be synthesized. For a 5 ms subframe and an 8 kHz input sampled speech, such a scheme would increase the coding rate by 200 bps. The next generation low bit-rate coders will operate at even lower bit-rates, that is at 4 kbps and below. At such rates it will be harder to achieve high quality. Our PPA technique along with a jointly optimized fixed codebook may be the key to providing the desired speech quality.

References

[1] B. S. Atal and M. R. Schroeder, "Stochastic coding of speech signals at very low bit rates," in *Proc. IEEE Int. Conf. Commun.*, (Amsterdam), pp. 1610–1613, May 1984.

[2] W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum, "Improved speech quality and efficient vector quantization in SELP," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (New York), pp. 155–158, Apr. 1988.

[3] C. Papacostantinou, "Improved Pitch Modeling for Low Bit-Rate Speech Coders," Master's thesis, McGill University, Montreal, Canada, 1997.

[4] W. B. Kleijn and J. Haagen, "Waveform interpolation for coding and synthesis," in *Speech Coding and Synthesis* (W. B. Kleijn and K. K. Paliwal, eds.), pp. 175–208, Elsevier, 1995.

[5] J. Stachurski, *A Pitch Pulse Evolution Model for Linear Predictive Coding of Speech*. PhD thesis, McGill University, Montreal, Canada, May 1997.

[6] R. Salami, C. Laflamme, J.-P. Adoul, A. Kataoka, S. Hayashi, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, and Y. Shoham, "Description of the proposed ITU-T 8-kb/s speech coding standard," in *Proc. IEEE Workshop on Speech Coding for Telecom.*, (Annapolis), pp. 3–4, Sept. 1995.