IJICTR International Journal of Information & Communication Technology Research ITRC

Volume 7- Number 3- Summer 2015 (11-19)

8 kbps Speech Coding using KLMS Prediction, Look-Ahead Adaptive Quantization and Pre-Emphasized Noise Reduction

Ghasem Alipoor
Electrical Engineering Department
Hamedan University of Technology
Hamedan, Iran
alipoor@hut.ac.ir

Mohammad Hassan Savoji
Electrical and Computer Engineering Faculty
Shahid Beheshti University
Tehran, Iran
m-savoji@sbu.ac.ir

Received: December 11 , 2014-Accepted: May 10, 2015

Abstract—**A new scheme is developed, in this paper, within the framework of the ADPCM-based waveform coding technique for low bit rate encoding of speech signals. The essential feature of this scheme consists of replacing the commonly used linear filter with nonlinear processing based on kernel methods. Our previously reported study, conducted on various emerging kernel adaptive algorithms, shows the usefulness of the kernel LMS (KLMS) algorithm in this framework. However, two original strategies are incorporated into this scheme, in the current study, to further improve its performance. The first strategy is based on improving the adaptive scalar quantization of the residual samples by employing a look-ahead concept to find the best possible quantization levels using the Viterbi algorithm. The second strategy is to apply a pre-emphasized noise reduction filter. This filter is implemented in a closed-loop form along with an inverse filter, so as to minimize the destructive effects of the noise reduction filter. Simultaneous employment of these strategies in the main scheme with the nonlinear processing provided by the KLMS algorithm brings about a waveform encoder that reconstructs speech with PESQ measure of 2.5 at low bit rate of 1 bit per sample.**

Keywords- Kernel Least Mean Square, Look-Ahead Quantization, Low Bit Rate Speech Coding, Pre-Emphasized Noise Reduction

I. INTRODUCTION

Despite the existence of a large number of efficient speech coding methods and even in spite of the recent departure from narrow-band to wide-band speech, there are always great demands for coding algorithms at lower rates. A key objective of many state-of-the-art speech and audio coding algorithms [1, 2] and modern standard encoders, e.g. MPEG-4 audio [3] and ITU g729.1 [4] standards, is still to deliver the best possible features at low bit rates. However, apart from sinusoidal coders that are directly applied to the speech waveforms, low bit rate speech coding techniques are mostly based on the source-filter model [5]. Speech is synthesized, in this model, by passing an excitation signal through a linear filter that represents the spectral contents of the speech signal. This simple paradigm has met with a considerable success and received a great popularity

*This article has been extracted from a PhD thesis carried out by Gh. Alipoor under supervision of Professor M. H. Savoji.

in a variety of applications. Nevertheless, this model suffers from some well-understood shortcomings, mainly due to its severe dependency to the nature of the signal. First of all, the frame-based linear prediction (LP) analysis, embodied in this model, implies a delay which can be intolerable in many applications. Furthermore, performance of these algorithms seriously degrades in the presence of background noise or any other non-speech signal. This in turn makes them very sensitive to tandem connection. This problem is usually alleviated by employing a speech enhancement unit prior to the coding scheme [6, 7]. Although this method is found to be useful in a variety of applications, its operability is generally restricted to slowly varying noises. Moreover, this preprocessor in turn deteriorates the performance of the speech-specific coding methods [8].

These problems mainly stem from the rigid dependency of the adopted source-filter model to the speech signals' characteristics. To address this issue, the well-known adaptive differential pulse code modulation (ADPCM) technique with backward prediction is used in the current study for developing a low bit rate speech coding scheme. ADPCM coders are classified as waveform coding algorithms that benefit from some appealing advantages, e.g. robustness against background noise, less degradation in tandem connection, having low delay and being independent from the nature of the signals. However, they are generally accepted as coding algorithms operating at moderate bit rates [5, 9]. By contrast, a scheme is developed, in this paper, for low bit rate coding of speech signals within this framework. The essential feature of this scheme consists of replacing the commonly used linear filter with nonlinear processing based on emerging kernel methods to account for nonlinear characteristics inherent in speech signals. In kernel methods, linear algorithms are applied on the transformed data in reproducing kernel Hilbert spaces (RKHS) that are nonlinearly related to the original input space [10, 11]. Reproducing property of the new spaces makes it possible to calculate inner products in these implicit high, or possibly infinite, dimensional spaces by means of the kernel functions evaluated in the low-dimensional input space. Therefore, in spite of linearity and convexity in RKHSs, resultant algorithms, possessing the property of universal nonlinear approximation, can be solved in a reasonable complexity. Our previously reported study, conducted on various kernel adaptive algorithms, shows the usefulness of the kernel LMS (KLMS) algorithm in this framework [12]. Nonetheless, as will be emphasized here below, the operability of the resultant algorithm is limited to the minimum rate of 2 bits per sample. But, there is room for further improvement and that is the issue undertaken in the study reported here. To that end, two original strategies are incorporated into our previously proposed KLMS backward ADPCM speech coding scheme to make it possible for the coding algorithm to operate at as lower rates as 1 bit per sample. In fact, this improvement is achieved by reducing the quantization noise and alleviating its effect on the quality of the reconstructed speech. Reducing the quantization noise has been always of great interest and considerable attempts are always made to alleviate its effect. The most recent study is reported in [13].

Inspired by the Viterbi algorithm, which is vastly used in the context of hidden Markov models and convolutional channel coding, a novel technique is developed to increase the accuracy of the adaptive scalar quantization used. This technique is based on a look-ahead concept to find the best possible quantization levels for representing the residual signal, i.e. to minimize the total reconstruction error calculated on the present and future samples. In this way, the decision-making is postponed, for any residual sample to be quantized, so as to take its effect on the future samples (in terms of the adaptive quantization step size and the KLMS prediction filter to be used) and the impact of the quantized future samples on the total quantization error into account. This is done by considering more than one quantization level for each residual sample and finding the best possible quantization sequence in a multipath search manner. This general idea has a long history of success in source coding in algorithms generally known as trellis coding [14-16]. A special form of these algorithms is the trellis coded quantization (TCQ) motivated by the trellis coded modulation concept [17]. This scheme and its variants, e.g. predictive trellis coded quantization [18] and trellis coded vector quantization [19], make use of the Ungerboeck's notion of set partitioning. In summary, to quantize one sample with b bits, the 2^{b} codewords used in the traditional adaptive quantization are doubled (to 2^{b+1} quantization levels) and then partitioned into $2^{\tilde{b}+1}$ subsets, where $\tilde{b}$ is an integer less than or equal to b. $\tilde{b}$ of the input bits are expanded by a rate $\tilde{b}/\tilde{b}+1$ convolutional code and used to select the subsets the quantization level for the current sample will be chosen from. The remaining $b-\tilde{b}$ bits are used to select one of the $2^{b-\tilde{b}}$ codewords in the selected subset. Viterbi decoding is used to find the sequence of codewords which minimizes the distortion caused by quantization. The convolutional code and set partitions are chosen in such a manner as to increase the Euclidean distance between allowable sequences of codewords [14-16].

Our technique is straightforward and utilizes the correlation that exists among subsequent samples in the coding algorithm and has clear differences with all these approaches. This technique is incorporated in the coding scheme along with a pre-emphasized noise suppression strategy. This strategy is based on cutting the quantization error back by means of a very simple one-tap low-pass filter similar to the one used in the frame-based analysis-by-synthesis coding algorithms for spectral tilt correction [20, 21]. This filtering is implemented in a closed-loop form along with its inverse, placed prior to the quantization block in the encoder, to minimize its destructive effect on the quantized speech residual signal. In the decoder, the low-pass filtering is carried out directly on the received quantized signal.

The paper is organized as follows. The KLMS algorithm and its employment within the ADPCM technique for speech coding are briefly described in section II, following a general introduction to kernel methods. Strategies adopted to improve the performance of the resultant codec are addressed in section III and sectionIV is dedicated to simulation results. Finally some conclusion remarks are presented in sectionV.

II. Employing KLMS Prediction in Speech Coding

The core part of the proposed scheme is an ADPCM-based coding algorithm with adaptive backward prediction. The quantization is carried out, in this technique, on the residual or what remains of the speech signal when its predictable parts have been removed adaptively. Linear prediction is the simplest choice in this paradigm where prediction is performed by a linear combination of a finite number of past samples. However, several researchers have investigated, theoretically and experimentally, the presence of nonlinearities in speech signals [22, 23]. These nonlinearities, which are mainly due to amplitude-dependent vocal folds oscillation and interaction between the vocal folds and the vocal tract, can be observed, for example, from higher order statistics measures and chaotic behavior of speech signals. Therefore, replacing the linear model with nonlinear models should enable us to obtain a more accurate description of the speech signal. This in turn may lead to a better performance of practical speech processing applications. On the other hand, our previously reported study shows the usefulness of nonlinear processing based on emerging methods of kernel adaptive filtering in this context [12]. In fact as mentioned later an improvement of up to 3.4 dB in the SNR of the decoded speech is achieved when employing the kernel LMS (KLMS) algorithm, which is judged the best for this purpose in that study. The KLMS algorithm is briefly introduced in this section, but further details can be found in [12].

A. KLMS Adaptive Algorithm

It can be shown that for any RKHS $\mathcal{H}$ with the kernel function *K*, one can imagine a space, known as the feature space, in which the inner product can be calculated through evaluating its kernel function *K* in the original input space [10, 11]. The mapping that projects the input vector $\boldsymbol{x} \in \mathcal{X}$ as the function $\phi(\boldsymbol{x})(\cdot) = K(\boldsymbol{x},\cdot) \in \mathcal{H}$is termed feature mapping and denoted by ϕ . In other words, representing the function $\phi(\boldsymbol{x})(\cdot)$as $\phi(\boldsymbol{x})$, the kernel K corresponds to a feature mapping ϕ for which:

$$K(\boldsymbol{x},\boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle, \quad \boldsymbol{x},\boldsymbol{y} \in \mathcal{X} \qquad (1)$$

Equation (1) is known as the kernel trick and states that the inner product in the feature space can be expressed in terms of the kernel function evaluation. Kernel trick has the central role in kernel methods based on which all linear inner-product-based algorithms can be implicitly applied to the feature space while remaining in the input space. Therefore, one can implicitly extend linear algorithms, such as those used in optimization problems, to a high-dimensional feature space while performing all calculations in the low-dimensional input space. The resultant algorithms possess the properties of convexity and universal nonlinear approximation. Furthermore, nonlinear kernel methods are quite flexible so that one can change the nonlinear model just by changing the kernel function used. In addition to successful applications of kernel methods in batch mode, developing kernel adaptive algorithms for online applications, e.g. the situation entailed in the backward ADPCM technique, have also recently witnessed a significant attention [24]. Extending linear adaptive algorithms to RKHSs are mostly based on reformulating the original algorithms in terms of inner products and then replacing the inner products with the kernel function evaluations. This will be equivalent to implicitly solving the linear adaptive algorithms in the feature spaces induced by the kernel functions, where transformed signals are more likely to be linearly related to the so called desired signal.

The milestone in the evolution of kernel adaptive algorithms is the kernel LMS (KLMS) algorithm which is a straightforward extension of the linear least mean square (LMS) algorithm into RKHS [25]. In the framework of ADPCM speech coding with backward prediction, we aim at predicting the current speech sample $s(i)$ based on P past samples of the reconstructed speech $\hat{s}$. Using the normalized LMS (NLMS) algorithm, the weight update equation, at instant *i*, is:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \frac{\mu x_i \hat{e}(i)}{\tilde{\sigma}_{s_i}^2} \qquad (2)$$

$\boldsymbol{x}_i = [\hat{s}(i-1) \quad \cdots \quad \hat{s}(i-P)]^T$ and $\hat{e}(i)$ are the input vector and the quantized value of the prediction error at instant i, respectively. $0 < \mu \ll 1$ is the convergence parameter to control the memory span of the predictor filter and therefore the convergence speed of the algorithm and ${\tilde{\sigma}_{s_i}}^2$ is an estimate of the input signal variance. The KLMS algorithm [24, 25] is derived by employing the NLMS algorithm to predict $s(i)$ based on the transformed input $\boldsymbol{\varphi}_i = \phi(\boldsymbol{x}_i)$. Denoting by $\boldsymbol{\omega}$ the estimated value of the filtering coefficients in the feature space and assuming $\boldsymbol{\omega}_0 = \boldsymbol{0}$, it is easily seen that:

$$\tilde{s}(i) = \boldsymbol{\omega}_{i-1}^T \boldsymbol{\varphi}_i = \mu \sum_{j=1}^{i-1} \frac{\hat{e}(j)}{{\tilde{\sigma}_{\varphi_j}}^2} K\left(\boldsymbol{x}_j, \boldsymbol{x}_i\right) \qquad (3)$$

$${\tilde{\sigma}_{\varphi_j}}^2 = \alpha {\tilde{\sigma}_{\varphi_{j-1}}}^2 + (1-\alpha) K\left(\boldsymbol{x}_j, \boldsymbol{x}_j\right) \qquad (4)$$

${\tilde{\sigma}_{\varphi_j}}^2$is an estimate of the variance of the transformed data at instant *j* and α is the forgetting factor in this estimation. In conclusion, adaptive NLMS filtering can be implicitly carried out in the high-dimensional feature space without direct access to the feature map and the filtering coefficients. More interestingly, it has been shown that the KLMS algorithm possesses the property of self-regularization that makes an extra regularization unnecessary [25]. In addition to simplifying the implementation, this property improves the performance because regularization biases the optimal solution.

As one can see from (3), the size of the network over which the signal is expanded or the number of past samples based on which the signal is estimated, called the dictionary, increases with the size of the data. This dictionary, at any time, consists of all previous input data, i.e. $\boldsymbol{x}_j$ vectors as well as all previous normalized residual samples $\frac{\hat{e}(j)}{\tilde{\sigma}_j^2}$. Alleviating this problem is the main implementational challenge in online applications where the number of observations continuously increases. In practice, redundancy among input data makes it possible to drastically reduce the size of the network, at the cost of a negligible effect on the quality of the model. This is generally carried out based on selecting the most informative data and discarding the others from the dictionary. This procedure is termed sparsification and many approaches have been proposed for this purpose in both batch and online modes. One of the first and still widely used measures is the novelty criterion (NC) proposed in [26] which acts based on a simple distance measure in the input space. In this approach, at iteration *i*, the minimum distance of the new input vector $\boldsymbol{x}_i$ to all the vectors retained in the dictionary $\mathcal{C}_{i-1}$ (i.e. $\min_{\boldsymbol{x}_j \in \mathcal{C}_{i-1}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$) is calculated. The new input vector will be accepted as a new element of the dictionary only if this measure is larger than a preset threshold, and the quantized prediction residual $\hat{e}(i)$ is also larger than another predefined constant. Sparsification drastically reduces the complexity of the online algorithm. This in turn makes kernel adaptive filtering a competitive candidate for nonlinear adaptive signal processing.

B. *Utilizing KLMS Prediction in the Framework of the ADPCM technique*

The main source of performance improvement for the ADPCM coder is the reduced dynamic range of the quantizer's input signal. This reduction is achieved by removing the short term redundancy of the speech waveform that is, in turn, accomplished by subtracting an adaptively predicted signal from the input signal. In backward prediction, used in this work, the coding parameters, i.e. the kernel adaptive predictor and adaptive quantizer's step size, are sequentially estimated from the past quantized residual signal, also available at the decoder. This scheme is shown in Fig 1 for the encoder. Prediction is usually performed linearly. But, speech is inherently nonlinear and nonlinear filters with higher ability to cope with this nonlinearity ought to be used. Volterra filters are nonlinear models widely used for this purpose [27, 28]. However, in addition to their inherent instability, the fact that their computational complexity grows exponentially with the memory size and the degree of nonlinearity involved is the major obstacle for their practical use.

Nonlinear adaptive Volterra filtering can be also accomplished using kernel adaptive algorithms [29]. Corresponding to the quadratic Volterra filter with memory span *P*, a kernel function is adopted in [12] as:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) := (\boldsymbol{x}_i{}^T \boldsymbol{x}_j) + (\boldsymbol{x}_i{}^T \boldsymbol{x}_j)^2 \qquad (5)$$

The relevant mapping function ϕ, that constitutes an RKHS, transforms the input vector $\boldsymbol{x}_i \in \mathbb{R}^P$ to $\phi(\boldsymbol{x}_i) \in \mathbb{F} = \mathbb{R}^{P+P(P+1)/2}$ that is a vector containing all possible first and second order permutations of the elements of $\boldsymbol{x}_i$. In contrast to the Volterra filter, the estimation complexity is now linearly dependent on the input dimensionality. The selected polynomial kernel exactly implements the quadratic Volterra filter. But, implementing this adaptive filter in the lower-dimensional input space avoids some instability characteristics the Volterra filters suffer from. Moreover, the self-regularization property of the KLMS algorithm makes numerical solutions more reliable.

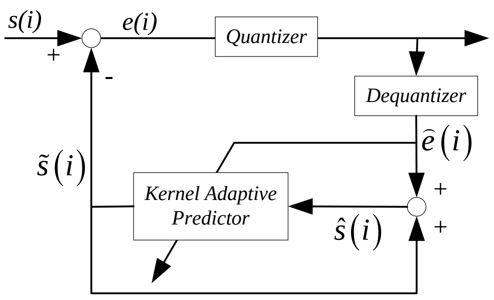

Fig 1 General scheme of the backward ADPCM encoder utilizing adaptive backward prediction

III. Strategies to Improve the Performance of the Encoding Algorithm

Although the KLMS algorithm results in a considerable improvement over the LMS algorithm, the operability of the resultant algorithm is only limited to the minimum rate of 2 bits per sample. But, there is still room to further improve the performance of this coding scheme. Two strategies devised for this purpose are described in this section.

A. *Look-Ahead Adaptive Quantization based on the Viterbi Algorithm*

Since the quantization error is of critical importance, an adaptive scalar quantizer is used to quantize the residual samples. The adaptive memoryless quantizer, at any given time, is assumed to have a symmetric uniform transfer characteristic with a fixed scheme and an unknown variable step size Δ_i. The optimum step size, Δ_{opt}, is related to the residual's standard deviation σ_e via a parameter, say ρ, that depends only on and is mmse optimized for the input probability density function (pdf) and the number of bits per sample (bps) used. For a nonstationary input, σ_e is time-varying and adaptive quantization means estimating it continuously. Therefore, the operation of an adaptive quantizer can be defined in the form of $\Delta_i = \rho\sigma_{e_i}$, where σ_{e_i} is an adaptive estimate of σ_e at time *i*. With the adaptive backward prediction, this estimation is also performed in backward manner [9].

Traditionally scalar quantization is nothing more than selecting a value among the codewords that best represents the current sample of the residual signal, independent of others. This memoryless method can be improved if one can consider the effect of the current decision on the following samples and also its subsequent impact on the total reconstruction error in

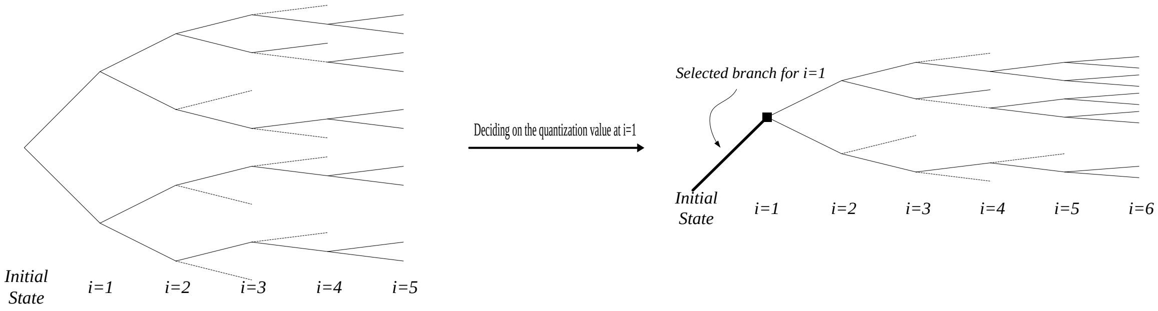

Fig 2 Tree formed by the look-ahead scalar quantization technique, for D_t=2, M_t=5 and L_t=5. Pruned leaves are shown with dashed lines

a look-ahead procedure. This can be easily performed by picking more than one codeword, at each time instant, to keep the future evolution of this error in sight. This way the decision-making among these picked candidates is carried out, after a short delay, based on their effect on the subsequent samples. This is done by forming a so-called tree and finding the best possible quantization sequence among the possible paths. This is the essence of the method used in this study to improve on the traditional memoryless adaptive quantizer described before.

All parameters of the codec are assumed to have started from known initial values based on which the first speech sample is estimated which in turn yields the first residual sample. The KLMS starts with an empty dictionary whereas the two energy estimates, used in normalizing the KLMS algorithm and adapting the quantization step size, are initialized with a small positive value. In the proposed look-ahead adaptive quantization (LAQ) technique, the D_t (to be called the tree depth) closest codewords are picked as the candidates for representing the first residual sample. The encoding algorithm subsequently steps forward a sample, considering all possible quantized values, resulting in D_t different residual samples for the second time instant. It should be noted that, in the backward scheme, values of the residual samples and all other parameters depend on the previous quantization levels. Therefore there will be D_t possible residual values for $i=2$. Each possible residual sample, at the second time instant, is in turn represented using D_t different quantization levels. This branching continues for the subsequent samples resulting in a tree, as illustrated in Fig 2. Each path through the tree represents a possible encoding sequence for the corresponding sequence of speech samples. In other word, there is a one-to-one correspondence between each path and an encoding sequence. These paths can be retrieved by saving the quantization levels as well as all other signals and parameters belonging to each path.

Decision is made, for each node, after a delay of L_t (to be called the trace-back length) samples, using the Viterbi algorithm. By doing so, at time instant L_t+i the best path, among the retained paths of the tree, which results in the best reconstructed speech sequence is found. The root branch of this path is chosen at instant i as the branch to be selected and all corresponding signals (to form the predictor or the sparsed dictionary) and parameters are therefore substantiated for that instant. Selection of the best path is, in turn, carried out on the basis of a merit criterion assigned to each path. This criterion shows the distortion caused by going through the path and is defined as the cumulative difference between the value of the speech samples and their reconstructed counterparts available at the encoder. That is, the criterion $C_{i,k}$ assigned at instant i to the path k, is defined as:

$$C_{i,k} = C_{i-1,k} + |s(i) - \hat{s}_k(i)|$$

$\hat{s}_k$ is the reconstructed speech signal following the kth path.

In practice, if the trace-back length L_t is sufficiently large, most of the surviving paths emerge from the root branch that leads to the selected best path. Our experiments showed that this is the case, on average, for more than 90% of decision time instances. However, once the selected path is chosen only those paths that emerge from the selected branch are kept and all other existent paths are cut away as they are no more valid. This notion is also depicted in Fig 2. To control the size of the resultant tree and hence the complexity and storage, the number of surviving paths is limited, at each instant, to a maximum value of M_t (to be called the tree mass). This is done by keeping the M_t best paths, with less cumulative distortion, and truncating the others. It should be noted that, at each instant, all processes (including analysis, quantization and reconstruction) should be carried out over all nodes and hence restricting the number of the survivors drastically reduces the computational complexity as well as the storage. It is noted that in addition to the increased complexity, the LAQ procedure implies a short delay of L_t samples. Notice that the LAQ is implemented in the encoder whilst the decoder uses a conventional dequantizer without delay. It is important to state that this quantization scheme is quite general and can be considered as a novel and efficient adaptive quantization method that could be used, in principle, in any other relevant application, especially when it is accompanied by the pre-emphasized noise reduction outlined next.

B. *Pre-Emphasized Noise Reduction*

A very simple noise reduction technique is adopted in this study to reduce the effect of the quantization error. In its simplest form the quantization error $q(i)$ can be modeled as an additive noise, i.e. $\hat{e}(i) = e(i) + q(i)$. For the employed uniform quantizer, the quantization error q can be modeled as a white noise with a fairly flat power spectrum. This is while the residual signal e still bears some similarity to the speech signal s and hence can be considered somewhat low-band. The proposed noise reduction technique is based on low-pass filtering the quantized residual signal $\hat{e}$ so as to attenuate the quantization noise q while keeping the residual component e unaffected, as much as possible. This is done by applying a one-tap integrator $(1+\alpha z^{-1})$ on the quantized residual signal.

The main feature of this method is that the effect of this noise reduction filter on the original signal is compensated beforehand by applying an inverse high-pass filter $(1+\alpha z^{-1})^{-1}$ to the residual signal before quantization. Although quantization is a nonlinear function and hence the superposition property is not applicable, simultaneous deployment of the two filters, in the encoder, results in low-pass filtering of the quantization error with minimum effect on the residual signal. Therefore, this strategy gives rise to suppressing the reconstruction error due to quantization noise while minimizing its destructive effect on the speech signal. This scheme is depicted in Fig 3. Notice that incorporating this noise reduction scheme is achieved by a slight increase in the complexity as now the two filters, whose relevant data must be kept as others for each path, are part of the LAQ implemented by Viterbi algorithm in the encoder. The low-pass noise reduction filter is applied, in the decoder, on the received quantized residual signal.

IV. Results

Results reported throughout this paper are averaged over all 504 SI speech signals in the test set of the DARPA TIMIT [30]. These signals have an average length of about 3.5 seconds containing each a whole sentence in English, uttered by both male and female speakers. They were originally sampled at 16 kHz and are down-sampled to 8 kHz, after applying a 20th order anti-aliasing low-pass filter, and then quantized uniformly at 16 bps. These results are obtained by setting the adaptive filtering memory span to P=10 samples. The quality assessment of reconstructed speech signals is based on two objective criteria of signal to noise ratio (SNR) and perceptual evaluation of speech quality (PESQ). PESQ evaluation is conducted as suggested by ITU-T P.862 recommendation [31] that has a good correlation with the subjective measure of mean opinion score (mos).

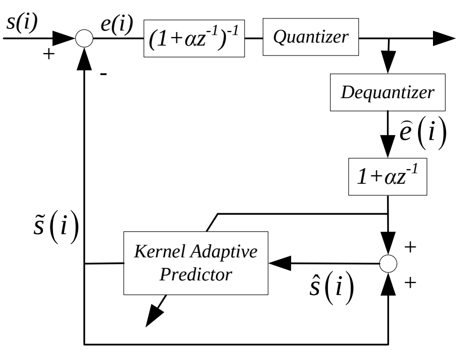

Fig 3 General scheme used for deploying the noise reduction filter along with its inverse filter in the encoder

It was shown in [12] that utilizing the KLMS algorithm in the framework of the backward-prediction ADPCM coding results in a considerable improvement in the overall performance of the encoder, as compared to its linear counterpart. This improvement is up to 3.4 dB in the SNR of the decoded speech. Moreover, it was seen in that study that the linear LMS-based coding algorithm reveals instability for bit-rates less than 3 bps whereas the KLMS-based codec's stability is restricted to bps values greater than 1. To further improve the performance of the nonlinear scheme, the use of the proposed LAQ as well as the well-known TCQ techniques with adaptive scalar quantizer are investigated in this scheme. These tests are conducted with four bps values of 1, 2, 3 and 4. The LAQ technique is implemented as described in section III in which the Viterbi parameters are set as D_t=4, L_t =7 and M_t=5.

Furthermore, the rate-1/2 feedback-free convolutional encoder whose block diagram is depicted in Fig 4 is used for the TCQ coder. This coder was also tested with some other convolutional encoders and best results are reported in this paper. In any case, decision is again made with a delay of 7 samples. Incorporating the TCQ technique in this structure makes the encoder stable for bps=1. On the other hand, even though the LAQ encoder is still unstable for bps=1, both LAQ and TCQ techniques increase the quality of the reconstructed speech. This improvement is achieved at the cost of an increased complexity and introducing a short delay of L_t samples. Overall quality of the reconstructed speech utilizing these techniques is tabulated in Table 1, along with the results achieved using the memoryless adaptive quantizer. These results reveal that the proposed LAQ technique outperforms the TCQ technique for bps values greater than 1. The averaged processing time is comparable for both techniques.

The incorporation of the proposed pre-emphasized noise reduction filter is also studied, using both LAQ and TCQ techniques as well as the original adaptive memoryless quantizer. Results, tabulated in Table 2, show that this noise reduction filtering does not

Table 1 Overall quality of the reconstructed speech using the KLMS algorithm, with different quantization techniques

	bps=1	bps=2	bps=3	bps=4
(a) SQ				
PESQ	Unstable	2.76	3.56	3.98
SNR (dB)	Unstable	8.37	19.21	24.03
(b) TCQ				
PESQ	2.2287	2. 79	3. 36	3. 84
SNR (dB)	7.99	12.68	17.80	22.28
(c) LAQ				
PESQ	Unstable	2.93	3.66	4.03
SNR (dB)	Unstable	11.04	20.42	24.33

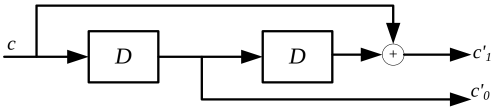

Fig 4 Feedback-free convolutional encoder used in the TCQ encoder. D stands for delay

increase the overall quality of the TCQ encoder. However, this technique improves significantly the performance of the LAQ encoder for bps=1 and 2. The LAQ encoder is now stable even at 1 bit per sample quantization and the resultant scheme again outperforms, in this structure, the well-known TCQ technique. However, it was noted that the noise reduction technique cannot stabilize the scheme by itself i.e. without being used as part of the LAQ. It is worth mentioning that the α parameter used in the noise reduction filtering is adjusted for each coding scheme separately on the basis of the averaged PESQ measure calculated on a training database. These best values are also included in Table 2. It can be seen that the positive effect of this technique is considerable for LAQ coder with bps=1. This effect diminishes with increasing bit-rates expecting less noise reduction. Using the TCQ technique with bps values of 2 and 3 best results are achieved with α=0 i.e. bypassing the noise reduction filter. Therefore, results corresponding to this scheme are the same in both tables. It is should be noted that the LAQ parameters are selected so as the algorithm leads to the best possible results. As an example, Fig 5 shows the average PESQ measure against the trace-back length for the LAQ algorithm with the pre-emphasized noise suppression filter. It can be seen that increasing the L_t parameter beyond 7 has no significant effect on the codec's quality.

In any case, the main achievement is that utilizing the noise reduction strategy along with the look-ahead quantization results in a waveform encoding algorithm that reveals a good performance in terms of average PESQ measure of about 2.5 at the rate of 1 bit per sample. This result is even better than that of the TCQ technique. Nonetheless, the algorithm suffers from high complexity. Moreover, as a backward adaptive all-pole filter is used to model the speech signal, the codec has high sensitivity to transmission errors. In addition to resorting to pole-zero models, the sensitivity of the developed encoder to channel errors can be reduced by including a leakage factor in the prediction adaptation algorithm [5]. Leakage allows the system to forget past values of the dictionary contents. Our tests showed that including the leakage factor considerably increases the robustness of the encoder against transmission error with a negligible effect on its transmission noise-free performance. It is noted that the inclusion of a leakage factor, introduced here in passing, is by itself a novelty in the context of kernel based methods. The issue of robustness to channel error and its remedy is not dwelled on here as this problem is beyond the scope of this paper. It is only noted that it may be mitigated by way of utilizing the proposed Viterbi algorithm in a joint source and channel coding scheme.

Table 2 Overall quality of the reconstructed speech incorporating noise reduction (NR) in the coding scheme

	bps=1	bps=2	bps=3	bps=4
(a) SQ				
PESQ	Unstable	2. 90	3. 59	3.98
SNR (dB)	Unstable	10.76	18.89	24.03
α	Unstable	0.2	0.15	0
(b) TCQ				
PESQ	2.25	2.80	3.36	3.84
SNR (dB)	7.99	12.65	17.80	22.28
α	0.1	0.025	0	0
(c) LAQ				
PESQ	2.47	3.07	3.67	4.03
SNR (dB)	8.56	14.4	20.59	24.39
α	0.97	0.4	0.2	0.05

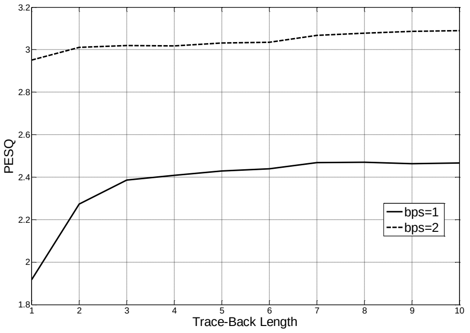

Fig 5 Averaged PESQ vs. trace-back length for the LAQ algorithm

V. Conclusion

Despite the proven usefulness of the celebrated KLMS algorithm in ADPCM based backward speech coding, its operability was limited to bit-rates values of 2 bps, i.e. bit-rate of 16 kbps for 8 KHz sampling frequency. Two original strategies are investigated in the current study to improve the performance of this coding algorithm so as to develop a waveform encoder able to operate at low bit-rate of 1 bit per sample, i.e. bit-rate of 8 kbps for 8 KHz sampling frequency. The first developed strategy is based on the Viterbi algorithm to refine the adaptive scalar quantization of the residual samples. This method is based on a look-ahead concept to consider the effect of the current quantization level on the following samples and the impact of future samples in the total reconstruction error. Although this quantization technique increases significantly the quality of the reconstructed speech and outperforms the well-known trellis coded quantization, the resultant coding scheme does not still operate at 1bps. The performance of the scheme is further improved by applying a noise reduction filter. The main feature of this procedure is that the low-pass filtering is carried out in a closed-loop form, in the encoder, along with an inverse filter to minimize its destructive effect on the reconstructed speech signal. Simultaneous deployment of these strategies brings about a waveform encoder that operates at low bit rates of 1 bit per sample.

This basic study in turn shows the usefulness of the proposed strategies and paves the way for further study and improvements. To the best of our knowledge, this is the first proposed low delay and low bit rate ADPCM-based speech coding algorithm. As a waveform coding algorithm, the developed scheme is expected to benefit from some appealing advantages of ADPCM coders e.g. robustness against background noise, less degradation in tandem connection, having low delay and being independent from the nature of the signals. Extending the KLMS algorithm to block processing and combining the proposed LAQ strategy with vector quantization, to represent the residual signal more efficiently, constitute the main line of our future research work.

References

[1] M. Holters, *et al.*, "Delay-free audio coding based on ADPCM and error feedback," in *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx08), Espoo, Finland*, 2008, pp. 221-225.

[2] Z. Perić and J. Nikolić, "An adaptive waveform coding algorithm and its application in speech coding," *Digital Signal Processing*, vol. 22, pp. 199-209, 2012.

[3] K. Brandenburg and M. Bosi, "Overview of MPEG audio: Current and future standards for low-bit-rate audio coding," *Journal of Audio Engineering Society*, vol. 45, pp. 4-21, 1997.

[4] ITU-T, "g729.1: g.729 based embedded variable bit-rate coder: An 8-32 kbit/s scalable wideband coder bitstream interoperable with g.729," in *ITU-T Recommendation*, ed, 2006.

[5] L. Hanzo, et al., Voice and Audio Compression for Wireless Communications: John Wiley & Sons Ltd, 2007.

[6] Y. Ephraim and R. M. Gray, "A unified approach for encoding clean and noisy sources by means of waveform and autoregressive model vector quantization," *Information Theory, IEEE Transactions on*, vol. 34, pp. 826-834, 1988.

[7] R. Martin, *et al.*, "A noise reduction Preprocessor for mobile voice communication," *EURASIP Journal on Applied Signal Processing*, vol. 2004, 1, pp. 1046-1058 January 2004.

[8] G. Guilmin, *et al.*, "Study of the influence on noise pre-processing on the performance of low bit rate parametric speech coder," in *Eurospeech*, 1999, pp. 2367–2370.

[9] N. S. Jayant and P. Noll, Digital Coding of Waveforms: Principles and Applications to Speech and Video. New Jersey: Prentice Hall, 1984.

[10] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, UK: Cambridge University Press 2004.

[11] B. Schölkopf and A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond Cambridge, MA: MIT Press, 2002.

[12] G. Alipoor and M. H. Savoji, "Wide-Band Speech Coding using Kernel Methods and Bandwidth Extension based on Parametric Stereo," in *Accepted for Publication in the 20th European Signal Processing Conference, EUSIPCO 2012*, To be held in Bucharest, Romania, 2012.

[13] B. W.-K. Ling, *et al.*, "Reduction of quantization noise via periodic code for oversampled input signals and the corresponding optimal code design," *Digital Signal Processing*, vol. 24, pp. 209-222, 2014.

[14] R. Gray, "Time-invariant trellis encoding of ergodic discrete-time sources with a fidelity criterion," *Information Theory, IEEE Transactions on*, vol. 23, pp. 71-83, 1977.

[15] J. Anderson and S. Mohan, "Sequential Coding Algorithms: A Survey and Cost Analysis," *Communications, IEEE Transactions on*, vol. 32, pp. 169-176, 1984.

[16] E. Ayanoglu and R. Gray, "The Design of Predictive Trellis Waveform Coders Using the Generalized Lloyd Algorithm," *Communications, IEEE Transactions on*, vol. 34, pp. 1073-1080, 1986.

[17] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *Communications, IEEE Transactions on*, vol. 38, pp. 82-93, 1990.

[18] M. W. Marcellin, *et al.*, "Predictive trellis coded quantization of speech," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, pp. 46-55, 1990.

[19] T. R. Fischer, *et al.*, "Trellis-coded vector quantization," *Information Theory, IEEE Transactions on*, vol. 37, pp. 1551-1566, 1991.

[20] C. Juin-Hwey and A. Gersho, "Adaptive postfiltering for quality enhancement of coded speech," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, pp. 59-71, 1995.

[21] B. Bessette, *et al.*, "The adaptive multirate wideband speech codec (AMR-WB)," *Speech and Audio Processing, IEEE Transactions on*, vol. 10, pp. 620-636, 2002.

[22] S. Teager, "Evidence for nonlinear sound production mechanisms in the vocal tract," in *Speech production and speech modelling*, ed: Springer, 1990, pp. 241-261.

[23] M. Faundez-Zanuy, *et al.*, "Nonlinear speech processing: overview and applications," *Control and intelligent systems*, vol. 30, pp. 1-10, 2002.

[24] W. Liu, *et al.*, *Kernel Adaptive Filtering: A Comprehensive Introduction*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2010.

[25] W. Liu, *et al.*, "The Kernel Least Mean Square Algorithm," *IEEE Transactions on Signal Processing*, vol. 56, pp. 543–554, 2008.

[26] J. Platt, "A Resource Allocating Network for Function Interpolation," *Neural Computation*, vol. 3, pp. 213–225, 1991.

[27] G. L. Sicuranza, "Quadratic Filters for Signal Processing.," *IEEE Proceedings*, vol. 80, pp. 1263-1285, 1992.

[28] G. Alipoor and M. H. Savoji, "Employing Volterra Filters in the ADPCM Technique for Speech Coding: A Comprehensive Investigation," *European Transactions on Telecommunications*, vol. 22, pp. 81-92, 2011.

[29] M. O. Franz and B. Schölkopf, "A Unifying View of Wiener and Volterra Theory and Polynomial Kernel Regression," *Neural Computation*, vol. 18, pp. 3097-3118, 2006.

[30] TIMIT. “DARPA TIMIT-Acoustic-Phonetic Continuous Speech Corpus,” [Online].

[31] ITU-T, "P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," in *ITU-T Recommendation*, ed. Geneva, Switzerland, 2001.

Ghasem Alipoor received his B.Sc. degree, in telecommunication engineering, from Tabriz University, in 2002, his M.Sc. and Ph.D. , in electronic engineering, from Shahid Beheshti University, in 2005 and 2012, respectively. Now, he is an assistant professor in Electrical Engineering Department of the Hamedan University of Technology. His main research interests include utilizing statistical methods and new algorithms in digital speech and image signals processing.

Mohammad H. Savoji finished his B.Sc. and M.Sc. studies, in Electrical Engineering, in Sharif University of Technology, in 1972 and 1975 respectively. He received his Ph.D. from INPG in Electronics and Telecommunications in 1979. He carried out a post-doctorate in Oxford University in 1981. He worked at various European Universities and Research Centers between 1981 and 1995. Now, he is a professor of electronics and telecommunications in Electrical Engineering Faculty of Shahid Beheshti University. Some of his interests include signal processing, image and video processing, speech processing, adaptive and non-linear filters.