A Scalable Delay Speech Codec at 8kbps using Local Cosine Transform IV

A. Averbuch¹*, A. Magner²

¹School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
²Dept. of Electrical Engineering Systems
Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract

A new low bit rate, 8kbps speech codec algorithm is proposed. This codec is based on the ITU-T G722.1 standard wideband codec that operates at 24kbps and 32kbps bit rates. The ITU-T G722.1 standard uses the Modulated Lapped Transform (MLT) as its kernel. The suggested algorithm reduces the ITU-T G722.1 codec bit rate to 8kbps reducing its bandwidth from 7kHz audio signals to 3.5kHz speech signals. The new algorithm uses the Local Cosine Transform type IV (LCT-IV) as its kernel. The LCT-IV allows a scalable approach for choosing the codec delay between 40ms to 20 ms algorithmic delay by changing the overlapping and the window sizes. The delay reduction introduces reasonable speech quality trade-offs. It was shown to have a high perceptual quality at diverse background noises making it very robust and noise immune. The new codec speech perceptual quality and intelligibility were tested at very difficult audible scenarios where very complex background sounds were added. The new codec outperformed the popular ITU-T G729 standard in most of these tests. The G729 codec performed better with clean speech. The codec has low complexity.

EDICS={1-COD1,2-ACOD,3-JPRO}

1 Introduction

Transform coding is very common among audio applications with high quality requirements (2 bits/sample and above). For speech coding at low bit rates (1 bit/sample and below), codecs that are using speech production physical model are very common, among these codecs the G729 and G723.1 standards are very popular especially in telecom applications. These speech compression codecs are known as vocoders. The main disadvantage of these speech codecs is that they rely on an explicit speech model parameters, thus their performance decrease significantly when the encoded signal contains additional audio information besides a single speaker speech. Signals with multiple speakers, noisy background
environment and especially music signals might result with poor speech quality reconstruction when using vocoders. Lately many hybrid codecs were suggested, which combines transform coding and vocoders' mechanism for improving the vocoders' quality with complex signals. The hybrid codecs yields higher bit rates than vocoders (around 16kbps) but with improved immunity to complex signals.

In this work we suggest a new algorithm which is based on transform coding that operates at 1 bit/sample. The algorithm is based on a bit rate reduction of the ITU-T G722.1 standard that operates at 24kbps and 32kbps bit rates (for audio signal with 7kHz bandwidth) to 8kbps bit rate (for speech signals with 3.5kHz bandwidth). The algorithm is investigated with two kinds of transforms that use overlapping constructions for handling the blocking and ringing artifacts of the block based transform coding: 1. Modulated Lapped Transform (MLT), as used in the G722.1 standard, which is based on a Modulated Quadrature Filter (QMF) bank. The MLT is asymptotically optimal lapped transform for coding an AR(1) signal with a high inter-sample correlation coefficients. 2. Local Cosine Transform type IV (LCT-IV) is a transform with local trigonometric basis functions that overlaps adjacent blocks. Both transforms are implemented using DCT-IV. For the LCT-IV transform a suitable bell (window function) for optimizing the performance of the new codec was chosen by adjusting the bell properties to the encoding process of the codec.

The LCT-IV introduces a scalable approach for reducing the algorithmic delay with reasonable audio quality tradeoffs. The MLT based codec yields 40 ms algorithmic delay. Using the LCT-IV based codec produces different delays that can vary from 40 ms to 20 ms depending on the window and overlapping sizes. Three different LCT-IV based codecs were investigated with 22.5ms, 30ms, 40 ms algorithmic delays. At the end the LCT-IV 40 ms and 22.5ms based codecs were tested against the popular ITU-T G729 standard codec. These subjective tests aimed to determine the perceptual quality and the intelligibility of the speech codecs when speech signals are used in a noisy environment or with multiple diverse audible signal components. The new codec was also tested with clean single speaker speech. The LCT-IV based codec with 40ms delay produced better results in noisy backgrounds, multiple speakers and music signals than the G729 codec both in the intelligibility and perceptual quality tests.

When clean speech was used with only one speaker the G729 produced better results than the LCT-IV based codec. The LCT-IV based algorithm, with 22.5ms delay, presents reasonable degradation preserving the speech intelligibility at difficult composed signals and yielding good perceptual quality, slightly worse than the LCT-IV based coder, with 40 ms delay. The main difference was with clean speech. The new codec is robust with high immunity to background noise signals and multiple sources speech signals. It also has low complexity for implementation. In addition, it produces scalable delay with reasonable quality, which makes it very versatile for usage in many applications.

The structure of this paper is as follows: section 2 contains an overview of the G722.1 algorithm.
Section 3 describes the DCT-IV transform and the blocking effect associated with it. Section 4 describes the MLT transform and its implementation in the G722.1 standard. Section 5 describes the theory of local trigonometric bases and its application to the new algorithm. Different bells and their properties for the LCT-IV are examined. The LCT-IV and the MLT are compared in the context of the new algorithm. Section 6 introduces the new codec structure with its different variations and its reasoning. Section 7 describes the performance of the new algorithm.

2 Algorithm overview of the G722.1 standard

The G722.1 is a wideband speech codec algorithm that provides an audio bandwidth of 50Hz to 7kHz, operating at a bit rate of 24kbit/s or 32kbit/s. The codec gets input samples at 16kHz. This codec is recommended for use in hands-free applications. It may be used with either speech or music inputs. The algorithm delay is 40 ms (not including the computational and transmission delays in the channel). The algorithm also supports a rate of 16kbit though this rate is not included in the G.722.1 recommendation yet.

2.1 Encoder - general description

The G722.1 algorithm is based on a transform coding using a Modulated Lapped Transform (MLT) [7] as the kernel of the algorithm. The audio source is divided into equally spaced frames and the encoder process a frame at a time. The MLT operates on 40 ms frames with a fixed 50% overlapping between adjacent frames. Each time the next 20 ms (320 samples) frame is read and the most recent 640 time domain samples are fed to the MLT. Each application of the MLT transform produces a frame of 320 MLT coefficients and each frame of MLT coefficients is coded independently. The allotment of bits per frame varies according to the compression bit rate: 640 bits for 32kbit/s, 480 bits for 24kbit/s, and 320 bits for 16kbit/s. The MLT is decomposed into a windowing and an overlap operation followed by a DCT type IV transform.

The MLT transform coefficients in each frame, or what we will refer to as the MLT spectrum, are divided into 14 regions. Each region contains 20 MLT coefficients that represent a bandwidth of 500Hz. MLT coefficients that represent frequencies above 7kHz are ignored (regions 15 and 16). For each region the amplitude envelope is computed. The amplitude envelope in the region \( r \) is defined as the RMS (root-mean-square) value of the MLT coefficients in the region, and is computed as

\[
\text{rms}(r) = \sqrt{\frac{1}{20} \sum_{n=0}^{19} \text{mlt}(20r + n)^2}.
\]

The amplitude envelope is a coarse representation of the MLT spectrum. The amplitude envelope for each region is quantized, then differentially coded, and entropy coded with Huffman code. The bits remaining after quantization and coding of the amplitude envelope are used to encode the MLT
coefficients.

Quantization and coding of the MLT coefficients is done using a categorization process. The process of categorization assigns a category to each of the regions. A category assigned to a region determines the quantization and coding parameters for that region, and the expected total number of bits required to represent the region’s MLT coefficients. There are eight categories labeled 0-7. Sixteen different sets of categorizations are computed and only one is selected for transmission.

3 DCT-IV

The Discrete Cosine Transform includes four different commonly known types of transforms: DCT-I, DCT-II, DCT-III, and DCT-IV [1]. The DCT can be efficiently implemented in many ways. For example, see [16] to have an FFT implementation of DCT-II and DCT-IV. We will focus on the definitions and properties of DCT-IV.

The 1-dimensional (1D) DCT-IV of a discrete function \( f(j) \quad j = 0, 1, \ldots, N - 1 \) is defined as:

\[
F(u) = \sqrt{\frac{2}{N}} \sum_{j=0}^{N-1} f(j) \cos \left( \frac{(2j + 1)(2u + 1)\pi}{4N} \right) \quad u = 0, 1, \ldots, N - 1.
\] (3.1)

The 1-dimensional Inverse 1D IDCT-IV is defined as: \( f(j) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} F(u) \cos \left( \frac{(2j + 1)(2u + 1)\pi}{4N} \right) \quad j = 0, 1, \ldots, N - 1. \)

Unlike DCT-II, no coefficient of DCT-IV is the average of the function \( f \). Thus, applying DCT-IV on a flat constant function will result in many non-zero transform coefficients. This is of course a disadvantage of DCT-IV when it is used in transform coding. An important difference between DCT-II and DCT-IV lies in the symmetry of the transform basis functions at the boundaries of their intervals. Observe that the cosine element in the basis function of the 1-D DCT-II has the term \( \frac{(2j+1)\pi}{2N} \), which can also expressed as \( \frac{(j+\frac{1}{2})\pi}{N} \), while the cosine element in the 1-D DCT-IV has the term \( \frac{(2j+1)(2u+1)\pi}{4N} \), which can also expressed as \( \frac{(j+\frac{1}{2})(u+\frac{1}{2})\pi}{N} \). Thus, the difference between them is the addition of \( \frac{1}{2} \) to the term \( u \). This difference totally changes the symmetry behavior of the basis functions at the boundaries. By expanding the range of \( j \) from \( 0 \ldots N - 1 \) to \( -N \ldots 2N - 1 \) we get that the basis functions of DCT-II are even at both sides, e.g., even with respect to \( -\frac{1}{2} \) and \( N - \frac{1}{2} \). By doing the same to DCT-IV we get that the basis functions of DCT-IV are even at the left side with respect to \( -\frac{1}{2} \) and odd at the right side with respect to \( N - \frac{1}{2} \).

3.1 Blocking effects

DCT block based encoders in audio and image processing usually divide the input signal into small predetermined blocks, which are quantized and encoded independently. Two types of reconstruction
artifacts are typical in block based transform codecs in general and in DCT based coders in particular, mainly at low bit rates: blocking (or tilling) and ringing. Blocking artifacts arise because the concatenation of the reconstructed blocks generates signal discontinuities across block boundaries. Ringing artifacts arise because the quantization errors on the transform coefficients generate signal reconstruction errors that last for the entire block. Another fact that contributes to the blocking effect (especially for small blocks) is that the transform processing in a block assumes no correlation between adjacent blocks. In audio signals these degradation are perceived as periodic clicking while for images an artificial boundary discontinuities among blocks is noticed as tilling effect. Methods that reduce the blocking effect after the application of a transform coding which is DCT based are: overlapping [6], filtering [2], Lapped orthogonal transform (LOT) [2]-[4] and local Fourier bases [17].

4 Modulated Lapped Transform (MLT)

As was described in section 3.1 the blocking effect occurs when transform block coding is being used. The MLT is a derivative of the lapped orthogonal transform (LOT) [3] that is being used as the kernel in the transform coding of the G722.1 compression algorithm. The main difference between the LOT and the standard DCT is that it uses basis function that extends beyond the block boundaries. The LOT maps a block of \( L \) samples into \( M \) coefficients \( (L > M) \) while maintaining an overlap of \( L - M \) samples with the adjacent block. The new basis functions have the property that they decay to zero near the boundaries, so that the discontinuity from zero to the boundary values is much lower than of DCT. Due to the fact that the LOT can be derived using a DCT it can be implemented efficiently. In this paper we will focus on the MLT, which is a form of the LOT transform, using \( 2M \) samples that are mapped into \( M \) coefficients each time with \( M \) samples overlap. In the MLT, a window of \( 2M \) samples from two consecutive blocks undergoes a cosine transform yielding \( M \) transform coefficients. The window is then shifted by \( M \) samples and the next set of \( M \) transform coefficients is computed. Thus, each window overlaps the last \( M \) samples of the previous window. The overlap ensures the continuity of the reconstructed samples despite the variations in the transform coefficients due to quantization [9]. The MLT has even fewer blocking effects than the LOT does. This is because the MLT window forces its basis functions to decay asymptotically to zero at their boundaries [11]. The design of the MLT is based on a multirate signal processing or wavelet representation scheme, which were discussed in Malvar’s book [10].

The M-channel MLT was defined as a particular instance of the oddly stacked [12] cosine modulated filter bank (known as the time domain alias cancellation – TDAC):

\[
p_a(n,k) = h_a(n) \sqrt{\frac{2}{M}} \cos \left[ (n + \frac{M + 1}{2})(k + \frac{1}{2}) \frac{\pi}{M} \right] \quad n = 0, 1, ..., 2M - 1
\]
\[ p_n(n, k) = h_s(n) \sqrt{\frac{2}{M}} \cos \left( (n + \frac{M+1}{2}) (k + \frac{1}{2}) \frac{\pi}{M} \right) \quad k = 0, 1, ..., M - 1 \]

where \( p_n(n, k) \) is the \((n,k)\) element of direct transform matrix \( P_a \), and \( p_s(n, k) \) is the \((n,k)\) element of the inverse transform matrix \( P_s \). The modulating cosine functions are windowed by \( h_s(n) \) for the forward transform (analysis filter bank) and by \( h_s(n) \) for the inverse transform (synthesis filter bank).

Assuming symmetric and identical windows \( h_s(n) = h_s(n) = h_s(2M - 1 - n) \). The filter bank in the above equation achieves perfect reconstruction (which leads to orthogonal basis functions) if the Princen-Bradley condition is satisfied \([10, 12]\) \( h_s^2(n) + h_s^2(n + M) = 1 \).

The MLT was defined by a unique window that applies \( \sum p_n(n, k) = 0 \) for all \( k \neq 0 \) (that is, DC signals are captured entirely by the first basis function), which is a necessary condition for maximum coding gain \([10]\). This window is given by \( h(n) = \sin[(n + \frac{1}{2}) \frac{\pi}{2M}] \) for \( n = 0, 1, ..., 2M - 1 \). And finally, the forward MLT was given by \( X(k) = \sum_{n=0}^{2M-1} x(n)p_{n,k} \quad k = 0, 1, ..., M - 1 \) where \( p(n, k) = p_n(n, k) = p_s(n, k) \). The Inverse MLT was given by \( x(n) = \sum_{k=0}^{M-1} [X(k)p_{n,k} + X'(k)p_{n,M,k}] \) where \( X'(k) \) is the previous block of the MLT coefficients.

The MLT can be computed using DCT-IV. The relationship between the MLT and the DCT-IV can be seen if we express the coefficients, \( p_{n,k} \), in the following way: \( p_{n,k} = h(n) \sqrt{\frac{2}{M}} \cos \left( \frac{(2n+1)(2k+1)}{4M} \pi + (2k + 1) \frac{\pi}{2} \right) \).

We notice that after multiplying by the windowing function \( h(n) \) we get DCT-IV with a phase shift \((2k + 1)\frac{\pi}{4}\) in basis function, which ensures aliasing cancellation for perfect reconstruction.

### 4.1 Implementation in G722.1

The inputs to each MLT are the most recent 640 audio samples, \( x(n) \), where \( x(0) \) is the oldest sample, and \( 0 \leq n < 640 \). The MLT outputs 320 transform coefficients. We denote \( \text{mlt}(m) \) to be the MLT output:

\[ \text{mlt}(m) = \sum_{n=0}^{639} \sqrt{\frac{2}{320}} \sin \left( \frac{\pi}{640} (n + 0.5) \right) \cos \left( \frac{\pi}{320} (n - 159.5)(m + 0.5) \right) x(n) \]

where \( 0 \leq m < 320 \). The MLT can be decomposed into a window, overlap and add operation followed by DCT-IV. The windowing, overlap and add operation is given by \( v(n) = w(159 - n)x(159 - n) + w(160 + n)x(160 + n) \) and \( v(n + 160) = w(319 - n)x(320 + n) - w(n)x(639 - n) \) \( 0 \leq n \leq 159 \) where \( w(n) = \sin \left( \frac{\pi}{160} (n + 0.5) \right) \) \( 0 \leq n < 320 \). By combining \( v(n) \) with DCT-IV we derive the expression \( \text{mlt}(m) = \sum_{m=0}^{319} \sqrt{\frac{2}{320}} \cos \left( \frac{\pi}{320} (n + 0.5)(m + 0.5) \right) v(n) \) for the MLT. The inverse MLT (IMLT) is performed in the same way. Each IMLT operation operates on 320 coefficients to produce 320 time domain audio samples. The IMLT can be decomposed into a DCT-IV followed by a window, overlap and add operation.
The DCT-IV is given by $u(n) = \sum_{m=0}^{319} \sqrt{\frac{2}{320}} \cos \left( \frac{\pi}{320} (m + 0.5) (n + 0.5) \right) ml(t)$ where $u(n)$ are the reconstructed samples of the IMLT. The window, overlap and add operation used half of the samples from the DCT output of the current frame with half of those from the DCT output of the previous frame $y(n) = w(n) u(159 - n) + w(319 - n) u_{old}(n)$ and $y(n + 160) = w(160 + n) u(n) - w(159 - n) u_{old}(159 - n) \quad 0 \leq n \leq 159$ where $w(n) = \sin \left( \frac{\pi n}{319} (n + 0.5) \right) \quad 0 \leq n \leq 319$. The unused half of $u()$ was stored as $u_{old}()$ for use in the next frame: $u_{old}(n) = u(n + 160) \quad 0 \leq n \leq 159$.

5 Trigonometric bases

We now present the local trigonometric bases, which replaces the MLT transform, in the proposed algorithm. Let $f$ be a function defined on $R$ and assume that we have a partition of the line into a set of disjoint adjacent intervals. If for each interval we have an expansion of $f$ in terms of an orthonormal base of this interval, then we say that this is a “windowed base” expansion of $f$ on $R$ [17]. We present a smooth orthogonal basis with an arbitrary partition of the line. The bases consist of cosine (or sine) multiplied by a smooth cutoff function (bell function) that overlaps adjacent intervals. The cosine functions used in the expansion are similar to the basis functions of DCT-IV and DCT-II, thus, these local trigonometric bases are called Local Cosine Transform (LCT). Using DCT-IV or DCT-II with the appropriate LCT will be called LCT-IV or LCT-II, respectively. We will show that the multiplication by the bell function can be implemented as folding the overlapping parts of the bell into the original intervals. In the discrete case, this fact allows us to perform the LCT in two successive steps: 1. Application of the folding operation on the original audio frame (preprocessing). 2. Application of the DCT-IV on the folded audio frame. The Inverse LCT (ILCT) can be performed in a similar way using an unfolding operation in which the signal samples are unfolded back to the overlapped bells and then multiplied by the bells functions.

5.1 Theory of Local Trigonometric Bases

The basic construction of smooth orthogonal local trigonometric bases was proposed in [7]. In this section we present some useful facts concerning this construction.

**Definition 1:** A function $f$ is called even (odd) with respect to $a$ on $[a - \varepsilon, a + \varepsilon]$ iff $f(2a - x) = f(x)$ ($f(2a - x) = -f(x)$). If $f$ is even (odd) with respect to some point then we say that $f$ has positive (negative) polarity with respect to this point.

Suppose we have an arbitrary function $f$ defined on the interval $I = [0, 1]$. We extend the function to $[-\varepsilon, 1 + \varepsilon]$ such that it will be even with respect to 0 and odd with respect to 1. Clearly, it means that $f$ can be expanded into a Fourier series by means of the orthonormal base $\sqrt{2} \cos \left( (k + \frac{1}{2}) \pi x \right)$. If we extend $f$ to be even with respect to 0 and even with respect to 1 then $f$ can be expanded in terms
of orthonormal base $\sqrt{2} \cos(k\pi x)$. Switching between even and odd will result in replacing cosine by sine. By applying the translation operator this can be extended to any interval $I_j = [a_j, a_{j+1}]$, with lengths $l_j = a_{j+1} - a_j$. This leads to the following:

**Proposition 1:** \( \sqrt{2} \frac{b_j}{l_j} \cos \left( k + \frac{1}{2} \right)_{\frac{x-a_j}{l_j}} \) \( k = 0, 1, 2, \ldots \) form an orthonormal base of \( L^2[a_j, a_{j+1}] \).

Let \( \{a_j\}, j \in Z \), be a sequence of numbers such that for all \( j, a_j < a_{j+1} \) and \( \lim_{j \to \pm \infty} a_j = \pm \infty \). Let \( \{I_j\} \) be the set of disjoint adjacent intervals \( [a_j, a_{j+1}] \) with lengths \( l_j \), and let \( \{\varepsilon_j\} \) be a sequence of positive numbers such that \( \varepsilon_j + \varepsilon_{j+1} \leq l_j \), or more clearly, \( a_j + \varepsilon_j \times a_{j+1} - \varepsilon_{j+1} \). We define window functions or *bells* over the intervals \( I_j \) by:

1. \( 0 \leq b_j(x) \leq 1 \) for all \( x \).
2. \( b_j(x) = 1 \quad x \in [a_j + \varepsilon_j, a_{j+1} - \varepsilon_{j+1}] \).
3. \( b_j(x) = 0 \quad x \in [a_j - \varepsilon_j, a_{j+1} + \varepsilon_{j+1}] \).
4. \( b_{j-1}(x) = b_j(2a_j - x) \) and \( b_j^2(x) + b_{j-1}^2(x) = 1 \quad x \in [a_j - \varepsilon_j, a_j + \varepsilon_j] \).

The last condition implies that the two bells \( b_{j-1}(x) \) and \( b_j(x) \) are supported over the adjacent intervals \( I_j \) and \( I_{j+1} \) respectively, are "orthogonal" and have a mutual symmetry with respect to \( a_j \). The bells can be of different widths and that they are not necessarily symmetric around their center.

The construction of the Local Trigonometric Base is based on the following:

**Theorem 1:** The series \( u_j,k(x), j \in Z, k \in N \), defined by

\[
 u_{j,k}(x) = \sqrt{\frac{2}{l_j}} b_j(x) \cos \left( k + \frac{1}{2} \right)_{\frac{x-a_j}{l_j}} 
\]

is an orthonormal base for \( L^2(R) \).

### 5.2 LCT-IV - Local cosine transform type IV

Theorem 1 shows us how to construct an orthonormal base for \( L^2(R) \) consisting of smooth functions supported on the intervals \( [a_j - \varepsilon_j, a_{j+1} + \varepsilon_{j+1}] \). Let \( \beta(x) \) be a continuous function defined on \( R \) with the following properties:

1. \( \beta(x) = 0 \quad x \leq -1 \)
2. \( \beta(x) = 1 \quad x \geq 1 \) \( \quad (5.3) \)
3. \( \beta(x)^2 + \beta(-x)^2 = 1 \) for all \( x \).

In particular, we can use the function \( \beta(x) \) defined in Eq. 5.4 and illustrated in Fig. 1.

\[
 \beta(x) = \begin{cases} 
 0 & x < -1 \\
 \sin \frac{\pi}{2}(1 + \sin \frac{\pi}{2} x) & -1 \leq x \leq 1 \\
 1 & x > 1.
\end{cases} 
\]

\[
(5.4) 
\]

8
Let \( \{ \varepsilon_j \} \) be a sequence such that \( a_j + \varepsilon_j = a_{j+1} - \varepsilon_{j+1} \). We can now define a bell function \( b_j(x) \) supported on \([a_j - \varepsilon_j, a_{j+1} + \varepsilon_{j+1}]\) as follows:

\[
b_j(x) = \begin{cases} 
\beta \left( \frac{x - a_j}{\varepsilon_j} \right) & x \in [a_j - \varepsilon_j, a_j + \varepsilon_j] \\
\beta \left( \frac{x + a_j - 2a_{j+1}}{\varepsilon_{j+1}} \right) & x \in [a_{j+1} - \varepsilon_{j+1}, a_{j+1} + \varepsilon_{j+1}] .
\end{cases}
\] (5.5)

![Graph of bell function](image)

Figure 1: The function \( \beta(x) = \sin \left( \frac{\pi}{2} (1 + \sin(\frac{\pi}{2} x)) \right) \) for \(-1 \leq x \leq 1\).

In this definition the bell has two parts. The left part is an increasing function from 0 to 1 and the right part is a decreasing function from 1 to 0. At the intersection point between these parts the bell has the value \( \sqrt{2}/2 \). In addition, \( b_j(x) \) also satisfies the conditions that are required from the bells as defined in Eq. 5.1. Finally, we define:

\[
u_{j,k}(x) = \sqrt{\frac{2}{l_j}} b_j(x) \cos \left( (k + \frac{1}{2}) \pi \frac{x - a_j}{l_j} \right)
\] (5.6)

where \( l_j = a_{j+1} - a_j \) and \( b_j(x) \) is the bell defined above. Theorem 1 yields that \( u_{j,k}(x) \) is an orthonormal base of \( L^2(R) \), thus, we construct a local cosine transform (LCT) basis function.

### 5.3 Discretization of LCT-IV

The LCT basis functions, \( u_{j,k}(x) \), have discrete analogues which form a basis for \( l^2(Z) \). For each \( i \in Z \), we define a discrete bell function \( \hat{b}_j(i) \) as \( \hat{b}_j(i) = b_j(i + \frac{1}{2}) \), \( i \in Z \). We define also a discrete version of the cosine set as \( \hat{f}_{j,k}(i) = \sqrt{\frac{2}{l_j}} \cos \left( (k + \frac{1}{2}) \pi \frac{i + \frac{1}{2} - a_j}{l_j} \right) \). Finally, we define the discrete sequences:

\[
\hat{u}_{j,k}(x) = \hat{b}_j(i) \hat{f}_{j,k}(i) = \sqrt{\frac{2}{l_j}} b_j(i + \frac{1}{2}) \cos \left( (k + \frac{1}{2}) \pi \frac{i + \frac{1}{2} - a_j}{l_j} \right)
\] (5.7)

where \( a_j - \varepsilon_j \leq a_{j+1} + \varepsilon_{j+1} \) and \( 0 \leq k \leq \frac{1}{2}(a_{j+1} + \varepsilon_{j+1} - (a_j - \varepsilon_j)) \). \( \hat{u}_{j,k}(i) \) is the discrete LCT basis function. For each \( j \), the functions \( \hat{f}_{j,k}(i) \) are evidently the basis functions for the DCT-IV transform over the interval \([a_j, a_{j+1}]\). For this reason we call Eq. 5.7 the discrete LCT-IV basis function.
5.4 Implementation of LCT-IV by folding

From the proof of theorem 1 we know that:

\[
\int_{a_j - \varepsilon_j}^{a_{j+1} + \varepsilon_{j+1}} b_j(x) f(x) \sqrt{\frac{2}{l_j}} \cos \left( k + \frac{1}{2} \pi \frac{x - a_j}{l_j} \right) dx = \int_{a_j}^{a_{j+1}} \tilde{F}_j(x) \sqrt{\frac{2}{l_j}} \cos \left( k + \frac{1}{2} \pi \frac{x - a_j}{l_j} \right) dx
\]  

(5.8)

where \( \tilde{F}_j(x) \) is defined as:

\[
\tilde{F}_j(x) = b_j(x) f(x) + b_j(2a_j - x) f(2a_j - x) - b_j(2a_{j+1} - x) f(2a_{j+1} - x).
\]  

(5.9)

The discrete analogues of Eq. 5.8 is:

\[
\sum_{i=a_j - \varepsilon_j}^{a_{j+1} + \varepsilon_{j+1}} b_j(i + \frac{1}{2}) f(i + \frac{1}{2}) \sqrt{\frac{2}{l_j}} \cos \left( k + \frac{1}{2} \pi \frac{i + \frac{1}{2} - a_j}{l_j} \right) = \sum_{i=a_j}^{a_{j+1}} \tilde{F}_j(i + \frac{1}{2}) \sqrt{\frac{2}{l_j}} \cos \left( k + \frac{1}{2} \pi \frac{i + \frac{1}{2} - a_j}{l_j} \right)
\]  

(5.10)

(5.11)

where \( \tilde{F}_j(i) = \tilde{F}_j(i + \frac{1}{2}) \). Eq. 5.10 tells us that in the discrete case rather than calculating the inner products with the sequence \( \tilde{a}_{j,k}(i) \), we can preprocess the data so that the standard DCT-IV algorithm can be used. This may be visualized as folding the overlapping parts of the bells back into the interval. The folding can be transposed onto the data, and the result is disjoint intervals of samples. Since the folding operation is completely reversible these samples can be later unfolded to reproduce the smooth overlapping segments. Therefore, the discrete LCT-IV will be implemented in two steps. First, we perform the folding operation on the source data and then applying the DCT-IV on the folded data.

Assume that we wish to fold the function \( f(x) \) across the points \( a_j \) and \( a_{j+1} \) using the bell \( b(x) = b_j(x) \). In other words, we want to fold the overlapping parts \( [a_j - \varepsilon_j, a_j] \) and \( [a_{j+1}, a_{j+1} + \varepsilon_{j+1}] \) back into the interval \( [a_j, a_{j+1}] \). By the translation operator we can see it as if we have to fold the function \( f(x) \) across 0, onto the intervals \( [-\varepsilon_j, 0] \) and \( (0, \varepsilon_j] \), using the bell \( b(x) \). Then, folding replaces the function \( f(x) \) with the left and right parts \( f_-(x) \) and \( f_+(x) \) as follows:

\[
\begin{align*}
    f_-(x) &= b(-x) f(x) - b(x) f(-x) \quad x \in [-\varepsilon_j, 0] \\
    f_+(x) &= b(x) f(x) + b(-x) f(-x) \quad x \in [0, \varepsilon_j].
\end{align*}
\]  

(5.12)

The symmetry of the bell allows us to use \( b(-x) \) instead of using the bell attached to the left interval. The folding function has the same polarity as the basis functions of DCT-IV, e.g., positive (even) at the left side and negative (odd) at the right side:
\[
\begin{align*}
  f_-(x) &= \frac{1}{2}f_+(x) + \frac{1}{2}f_-(x) \\
  f_+(x) &= f_+(x) \\
  f_-(x) &= \frac{1}{2}f_+(x) - \frac{1}{2}f_-(x)
\end{align*}
\]

(5.13)

The formulas for the unfolding can be reached by solving two sets of two equations. For example, consider the next set of equations:

\[
\begin{align*}
  f_-(-x) &= -b(-x) + b(x) f(-x) \\
  f_+(x) &= b(x) + b(-x) f(-x)
\end{align*}
\]

(5.14)

By multiplying the upper equation by \(b(x)\) and the lower by \(-b(-x)\) and then summing the resulting equations we have

\[
b(x)f_+(x) - b(-x) f_-(x) = (b(x)^2 + b(-x)^2)f(x).
\]

(5.15)

The fact that the bell \(b(x)\) function satisfies the orthogonality property \(b^2(x) + b^2(-x) = 1\) implies:

\[
f(x) = b(x)f_+(x) - b(-x)f_-(x) \\
  x \in [0, \varepsilon_j].
\]

(5.16)

A similar set of equations and a similar argument will give us the formula for the case \(x \in [-\varepsilon_j, 0]\). Thus, we found that the unfolding operation reconstructs \(f(x)\) from \(f_-\) and \(f_+\) with the following formulas:

\[
f(x) = \begin{cases} 
  b(x)f_+(x) + b(-x)f_-(x) & x \in [-\varepsilon_j, 0] \\
  b(x)f_+(x) - b(-x)f_-(x) & x \in (0, \varepsilon_j]
\end{cases}
\]

(5.17)

Therefore, the unfolding operation will be used to implement the Inverse LCT (ILCT).

### 5.5 LCT-IV bell properties

The bell function that is being used by the LCT-IV may affect the performance of the codec. The bell function can change the coding gain of the LCT-IV and the properties of the basis functions and thus affect the audio quality after compression. Besides the coding gain, the bell shape significantly affects the spectrum of the signal after the application of the LCT-IV. If we take into account the fact that the G722.1 encodes the signal, after the transform coding, according to the obtained spectrum bands, we should design the bell to best fit the G722.1 encoder properties. Many bells can be used which satisfy the condition in Eq. 5.3. In section 5.2 we defined a bell function for the LCT-IV which is based on \(\beta(x)\) that was defined in Eq. 5.4. We will present a way to modify this particular bell function to create a new bell function that has different properties and still satisfies the conditions in Eq. 5.3. The bell function \(\beta(x)\) is smooth on \((-1, 1)\) with vanishing derivatives at the boundary.
points, thus the bell function has a continuous derivatives on $R$. We can modify $\beta(x)$ to have more continuous derivatives by iterating the innermost sine. The result is that the function will be flatter and close to zero when $x$ reaches $-1$ and $1$. More specifically, the modified $\beta(x)$ function for LCT-IV can be defined as:

$$\beta_0(x) = \sin \left( \frac{\pi}{4}(1 + x) \right) \quad \text{for} \quad x \in [-1, 1].$$

(5.18)

Windows with higher regularity are constructed with the profile $\beta_k(x)$ defined for $k \geq 0$ by induction:

$$\beta_{k+1}(x) = \beta_k \left( \sin \frac{\pi x}{2} \right) \quad \text{for} \quad x \in [-1, 1].$$

(5.19)

For any $k \geq 0$, one can verify that $\beta_k$ satisfies the conditions in Eq. 5.3 and has $2^k - 1$ vanishing derivatives at $x = \pm 1$. The resulting $\beta(x)$ is therefore $2^k - 1$ times continuously differentiable.

![Figure 2: Bell function $\beta(x)$ (Eq. 5.19) with different values of $k$.](image)

As $k$ becomes larger, the bell becomes flatter until the left half side of the bell has value 0 and the right half side has the value 1. In this case the folding and unfolding operations does not affect. Figure 2 shows the bell functions based on Eq. 5.18 and 5.19 for different $k$. Figure 3 shows the spectrum of the bells for four different $k$ after the application of the FFT. For $k=1$ we have a window which is similar to an impulse response in the frequency domain. It has a narrow main lobe (less than 0.2 wide) and afterwards an attenuation of more than 40dB. For $k=3$ we get a very wide main lobe that declines slowly. In its edges it has a very large attenuation of more than 100dB. For $k=6$ and $k=10$ we have many side lobes and a small main lobe with less than 20dB attenuation. The bell with $k=1$ has the advantage of being most similar to an impulse response in the frequency domain. Thus, when it will be convoluted in the frequency domain with the target signal it will preserve better the localization in the frequency domain. Another advantage of this bell is that it has the smoothest shape in the frequency domain among all the bells. Convoluting with a smooth shape in the frequency domain will yield a smoother result. Since the quantization is done in the spectral domain it will have
a smaller quantization error when applied to speech signals that are concentrated in a few harmonics (voiced phonemes for example).

![Figure 3: The spectrum of the bells (based on Eq. 5.18) with different values of k after the application of the FFT.](image)

We will demonstrate this advantage by applying LCT-IV with each bell on a 1kHz sine signal. We can see the results in Fig. 4. It is obvious that the bell with \( k = 1 \) has the best spectral localization. It yields a better compaction of the coefficients. The quantization error in this case will be smaller due to the fact that the energy of the signal is concentrated in only a few coefficients. The small coefficients that cause a large quantization error disappears for \( k = 1 \), improving the quantization performance. Figure 5 shows a comparison between the MLT 40 ms transform and the LCT-IV 40 ms transform (with \( k = 1 \)) after they were applied on a 1kHz sine signal. The MLT has even better localization property in the spectrum than that of the LCT-IV with \( k = 1 \), though this advantage is almost negligible as can be seen in Fig. 5. The localization property will be noticeable especially with speech signals that contain few base harmonics like voiced signals that have high formants in only few points. To understand better how the different bells cope with complex signals we will use a white noise signal input.
Figure 4: Initial 60 coefficients after the application LCT-IV on a 1kHz sine input signal, using different $k$ in the LCT-IV.

Figure 5: Initial 60 coefficients after the application: 1. LCT-IV with $k = 1$ on a 1kHz sine input signal. 2. MLT on a 1kHz sine input signal.

We apply all the different LCT-IV with different $k$ values on a white noise (after averaging many frames to slightly smooth the spectrum). We recall in section 2 that the encoding stage includes a categorization process. In this categorization process the bit allotment for each region is determined mainly according to the RMS values in each region. We compare the RMS values in all the regions after the application of applying the LCT-IV with different $k$ values on a white noise. Comparing the RMS values for each bell in each region shows the difference between the bells from the encoding process point of view. The RMS values are shown in Fig. 6. We see that the LCT-IV with $k=1$ emphasizes slightly better the low regions 1-4 (0-2000Hz). In speech signals most of the energy, in most of the cases, is concentrated in the range 0-2000Hz. Thus, the bell with $k=1$ has the potential to emphasize these spectrum bands, yielding better quantization performance. For complex signals the energy is spread over all the spectrum range. In this case we cannot predetermine which bell will perform better, but we definitely see that each bell emphasizes differently the regions. Finally, we
compare between the application of LCT-IV with $k=1$ and MLT both with a 40 ms bell length on a white noise (as in Fig. 7). We see that the MLT emphasizes better the lower bands while the LCT-IV emphasizes better the higher bands.

We summarize that all the bells were tested for different scenarios that represent similar components to those contained in speech signals. The white noise represents wide band unvoiced phonemes in the speech and complex signal like many different background noises (white noise, music etc.). The 1kHz sine signal represents a component of voiced phonemes. Usually, in voiced phonemes there are more than one single harmonic. In this case the behavior is a superposition of the responses to single tones. Finally, we tested the distortion for each bell in the new 8kbps codec with different audio segments. We measured the Segmental SNR (SSNR) and compared the results among different $k$ values. In all the scenarios the SSNR was better for $k=1$ especially with clean speech signals. It had a SSNR result of 1.5-2dB better than the bell with $k=6$ and about 1dB better results than the bell with $k=3$. For $k=10$ the results were poor (more than 3dB less than for bell with $k=1$). A detailed comparison between the LCT-IV codec with $k=1$ and the MLT codec is presented in sections 6.5 and 7.1.

![Figure 6: RMS values for all the regions using bells with different $k$ (based on Eqs.5.18 and 5.19).](image)

![Figure 7: RMS values for all the regions using: 1. LCT-IV with $k=1$. 2. MLT transform.](image)
6 The new algorithm

6.1 The challenges in audio transform coding at low bit rate

Transform codecs perform very well on bit rates equal or higher than 2 bits/sample. In these cases the perceptual degradation is almost not noticeable to the ear. At low bit rates, less than 2 bits/sample, the performance of the transform codecs is degraded rapidly. The quantization error grows and becomes more significant due to the fact that perfect reconstruction conditions are violated, aliasing between basis function generates significant errors that are noticeable to the ear. This last problem makes the basis function design (the filters design) very critical in order to minimize the effect of the quantization error. These effects are less noticeable with complex or mixed signal such as music because of the higher masking levels associated with such signals. With clean single speaker speech transform codecs at low bit rates might not be able to reproduce the fine harmonic structure. The MLT transform has a large algorithmic delay because of the 50% adjacent frame overlap that it requires. On the other hand it has the advantage of a high coding gain. This advantage is less noticeable at low bit rates codecs.

We will show how the use of the LCT-IV at a low bit rate encoding partially solves the two problems above and yields a reduction of the algorithmic delay, which becomes critical in low bit rates applications (e.g. telecom), with reasonable degradation of the audio quality.

6.2 Choosing transform coding strategy

One of the important issues in transform codecs is to determine the frame (block) size strategy. Two opposite requirements arise. The first one is to get a good frequency resolution, which means that long enough block size should be employed, 30ms – 60ms is common. The other requirement is that the block size should not be too long when transient signals occur, thus leading to a noise lasting the entire block known as the pre-echo effect [15]. In other words coarse resolution in frequency is preferred at high frequencies speech signals. Then, a short block size is preferred. At low frequencies speech signals, high resolution is needed and in that case long block size should be used. In our proposed algorithm, we will compromise in the frequency resolution and use a fixed length blocks of 20ms - 40ms. We will show that with the LCT-IV as the transform coding, even a small block size such as 20 ms produces a reasonable perceptual quality with the big advantage of having a short algorithmic delay. Because we are handling a low bit rate coding, the quantization error of the coefficients becomes more significant and thus the advantage of the MLT, from a coding gain aspect point of view, is less noticeable.
6.3 Bandwidth and bit rate reduction

We now suggest a new compression scheme that is based on the G722.1 with lower bit rate that is suitable for telecom applications. This new scheme is for a 3.5kHz (8kHz sampling rate) analog bandwidth, which is used in telecom applications. The main advantage of transform codec based compression at low bit rates is that it has a better immunity to background noises and it handles better multiple speakers at the same time, yielding a more robust compression.

The new codec replaces the MLT with the LCT-IV while uses the same quantization and entropy coding as in the G722.1. It reduces the number of regions from 14 to 7 which represents spectrum frequencies up to 3.5kHz. The new codec operates at 8kbps. The encoding process of the G722.1 was adjusted to the new bit rate and the quantizer and encoding parameters were changed to fit the new codec. The categorization process is done as in the G722.1 codec, using 16 categorization with 8 categories and the same bit allotment. The categorization parameters were optimized to incorporate 7 regions instead of 14. The number of coefficients after the application of the LCT-IV is 160 (instead of 320) in the new algorithm. The total number of bits allocated for a frame is 160 bits.

The MLT transform replacement with the LCT-IV allows the reduction of the algorithmic delay with reasonable audio quality tradeoffs. According to the system requirements the algorithmic delay can be reduced to 22.5ms with acceptable performance.

6.4 Replacing the MLT with the LCT-IV in the new algorithm

We replaced the MLT with the LCT-IV in the encoder and decoder. Using a LCT-IV transform with 160 coefficients replaces the MLT in a transparent way. The same quantization and entropy coding scheme as in the G722.1 were used in the new codec. Therefore, the quantization and the coding process were not tailored to accommodate the special behavior and features of the LCT-IV. Future research can optimize the quantization and coding process to fit the LCT-IV.

The compression scheme using the MLT is described in Fig. 8.

![Diagram](image)

Figure 8: MLT codec

We notice that after windowing there is a gain control, which is applied to improve the dynamic range of the computation. The gain determines a scaling factor, which is encoded, and transferred to the decoder. The computation of the gain control was changed to fit the 8kbps compression.
The replacement of the MLT with the LCT-IV is illustrated in Fig. 9. The folding operation on both left and right overlapping sides is done in each frame and then the gain control computes the appropriate scaling factor. Afterwards, the DCT-IV is applied yielding the LCT-IV coefficients. The LCT-IV coefficients and the scaling factor are quantized and encoded. At the decoder side an unfolding operation is done based on Eq. 5.17 as was shown in section 5.4.

![Figure 9: LCT-IV encoder](image)

We used identical bells for all the frames and a symmetric overlapping on the left and right sides. This is based on Eqs. 5.18 and 5.19 with \( k = 1 \), which produced the best performance. We call the size of the overlapping part \( \text{reach} \). Figure 10 illustrates the bell’s parameters.

![Figure 10: Structure of the bells and their associated parameters](image)

We used three different overlapping sizes with the following parameters: 1. Bell Length = 320, \( \text{reach} = 80 \). This is a full overlap. The algorithmic delay is 40 ms. 2. Bell Length = 240, \( \text{reach} = 40 \). This is partial overlap. The algorithmic delay is 30 ms. 3. Bell Length = 180, \( \text{reach} = 10 \). This is a partial overlap. The algorithmic delay is 22.5 ms.

### 6.5 The influence of the window and overlapping size on the codecs performance

In this section we analyze the effects of using different overlapping sizes (different bell sizes) in the LCT-IV transform on the new codec’s performance for different speech components. We will refer to
the following three different codecs: 1. 8kbps MLT based codec with window length of 40 ms and 50% overlapping with the adjacent frame. 2. 8kbps LCT-IV based codec with bell length of 40 ms and overlapping parts of 10 ms on each side. 3. 8kbps LCT-IV based codec with bell length of 22.5 ms and overlapping parts of 1.25 ms in each side.

First, we will examine the influence of different bell sizes and different overlapping sizes on voiced phonemes that contain few basic harmonics. Figure 11 shows the reconstructed signal of the syllable /ng/ extracted from the last part of the word “young” with the three different codecs. The syllable /ng/ is a voiced phoneme.

![Figure 11: The syllable /ng/ waveform of the original signal. From top to down: original signal, the reconstructed signal using codec no. 1 (MLT based codec with 40 ms delay), the reconstructed signal using codec no. 2 (LCT-IV based codec with 40 ms delay), the reconstructed signal using codec no. 3 (LCT-IV based codec with 22.5 ms delay).](image)

We notice that using longer bell size with longer overlapping parts yields better reconstruction results. We see in Fig. 11 that there is a noticeable degradation in the LCT-IV 22.5ms window based codec in the reconstructed signal. The reason is that when using a longer bell the transform presents a wider view of the signal’s spectrum. In this case the harmonics will have a better emphasis in the LCT-IV coefficients.

Figure 12 illustrates the differences between the LCT coefficients in one frame in the /ng/ syllable for 22.5 ms and 40 ms windows codecs. We see that many coefficients are zero and the energy is concentrated only in a few coefficients in the 40 ms LCT-IV transform. Thus, the quantization distortion is smaller. On the other hand, when we use a short bell the LCT-IV coefficients are spread over more coefficients expressing a less localized spectral view of the signal. Many small coefficients cause a larger quantization error.
When voiced signal contains more than a few harmonics the advantage of a long bell almost disappears. Figure 13 shows the waveform of the reconstructed signal of the three codecs and the original signal of the syllable /you/ which is the first part of the word “young”.

Figure 12: LCT-IV coefficients of one frame from the /ng/ syllable. From top to bottom: 1. LCT-IV 40ms window. 2. LCT-IV 22.5 ms window.

Figure 13: From top down: The syllable /you/ waveform of the original signal, the reconstructed signal using codec no. 1 (MLT based codec with 40 ms delay), the reconstructed signal using codec no. 2 (LCT-IV based codec with 40 ms delay), the reconstructed signal using codec no. 3 (LCT-IV based codec with 22.5 ms delay).

This syllable is composed of two phonemes /y/ which is a vowel and /ou/ which is a diphthong. These phonemes are voiced. We see in Fig. 13 that the differences between the long and short bell in the LCT-IV based codec in the reconstructed signal fidelity are negligible. The differences between the LCT 40 ms window and MLT 40 ms window codecs are also minor. Figure 14 shows the LCT coefficients for 40 ms and 22.5 ms windows. We notice that in both cases there are many
LCT coefficients to encode thus the advantage of having long window is substantially reduced. Unlike the /ng/ syllable, here the energy of the signal is not concentrated only in a few coefficients. In the unvoiced phonemes, which have a wide band spectrum, all the codecs perform much worse because the transform coefficients are spread over all the regions yielding a higher quantization error. Figure 15 illustrates the reconstructed signal for the syllable /ch/ from the first part of the word “children”, for all the three codecs. In all the three codecs the reconstruction errors were big but they were not so noticeable to the ear. On the other hand, they were noticeable to the eye in the waveform illustration.

We conclude that differences between the MLT 40 ms window based codec and the LCT 40 ms window based codec reconstruction signal quality are negligible for both unvoiced and voiced phonemes as we saw in figures 11,13,15 (always the two signals in the top of these figures). Figure 16 shows the measured Segmental SNR (SSNR) for the reconstructed signal for all the three syllables. We see that the MLT 40 ms and the LCT-IV 40 ms codecs have almost the same SSNR results. The differences between the LCT-IV 40 ms and the LCT-IV 22.5 ms are significant in the /ng/ syllable and negligible in the /ch/ syllable. In the /you/ syllable the differences were noticeable but they were small.

Figure 14: LCT-IV coefficients of one frame from the /you/ syllable. From top to bottom: 1. LCT-IV 40ms window. 2. LCT-IV 22.5 ms window.

The short window has in some cases an advantage over the long window when it is used with transient signals. In these signals the error is spread over a smaller frame when short window is used. In Fig. 16 we see that the LCT-IV 22.5ms based codec achieves better SSNR results for transient frames with the voiced phoneme /ng/ than the unvoiced phoneme /ch/. This advantage is less noticeable with comparison to the mentioned disadvantages of the short window, because its duration is usually small. With rich harmonics signals like music the differences between short and long window transforms are also minor. Music signals have rich spectrum yielding at the LCT-IV output many coefficients with high values. In this case encoding of either the 22.5 ms transform coefficients or the 40 ms transform coefficients yields almost the same quantization error.
Figure 15: From top to down: the syllable /ch/ of the original waveform, the reconstructed signal using codec no. 1 (MLT based codec with 40 ms delay), the reconstructed signal using codec no. 2 (LCT-IV based codec with 40 ms delay), the reconstructed signal using codec no. 3 (LCT-IV based codec with 22.5 ms delay).

Figure 16: Segmental SNR for the words “young chi” from the sentence “young children” partitioned into the three syllables

7 Experimental Results

7.1 Measurements of the results

This section presents a comprehensive comparison among various codecs that are based on the proposed algorithm. The following measures were used to test the quality of the codecs:

Segmental SNR (SSNR) as an objective measurement of the fidelity of the reconstructed signal.

Subjective test as a measurement of the perceptual quality for the listeners (listener’s effort).

Subjective test as a measurement of the intelligibility of the codec in a very noisy scenarios for testing the robustness of the codec.

Four different codecs were chosen and one reference target codec was added for the tests. In each
test different combinations of codecs were used. G729 was chosen as a reference due to its popularity in telecom applications and its high performance. Three codecs that were tested were based on the new algorithm described in the section 6.4. They were based on LCT-IV transform each with different algorithmic delay. One codec was based on the MLT transform. One of the goals in the tests was to determine what are the tradeoffs in terms of the codec performance while the algorithmic delay is being reduced.

7.2 Objective Tests

The following test is an objective measurement for comparing the different codecs. We used the segmental SNR to measure the performance of the four different codecs:

1. 8kbs with MLT and algorithmic delay of 40 ms using a 40 ms window with a 20 ms overlap (10ms in each side).

2. 8kbs with LCT-IV and algorithmic delay of 40 ms using a 40 ms window (bell) with a 20 ms overlap (10ms in each side).

3. 8kbs with LCT-IV and algorithmic delay of 30 ms using a 30 ms window (bell) with a 10 ms overlap (5ms in each side).

4. 8kbs with LCT-IV and algorithmic delay of 22.5 ms using a 22.5 ms window (bell) with a 2.5 ms overlap (1.25ms in each side).

The SSNR measurement is useful for comparing how well the source signal is reconstructed and for analyzing the reconstructed signal fidelity. This measurement is misleading when it is used for vocoders or hybrid parametric codecs. For this reason in this test we did not include the G.729 codec as a reference.

We took eight different speech segments each with completely different characteristics and tested them all with different codecs. The different speech sections that were used and the performance (in dB) of each codec is given in table 1. The diversity of these speech signals was aimed to test the codec at different auditory scenarios. The Segmental SNR was computed on 10ms sub-frames where a sub-frame is a segment that contains 80 audio samples. The SSNR formula for computing a 10ms sub-frame (80 samples) is given by:

$$SSNR = 10 \times \log_{10} \frac{\sum_{i=0}^{79} (Inpu[i])^2}{\sum_{i=0}^{79} (Input[i] - Output[i] \times ScalingFactor)^2}$$

(7.1)
## Table 1: SSNR results for different audio segments vs. different codecs.

<table>
<thead>
<tr>
<th>Tested segment</th>
<th>40 ms MLT in dB</th>
<th>40 ms LCT in dB</th>
<th>30 ms LCT in dB</th>
<th>22.5 ms LCT in dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Conversation between two speakers in a very noisy environment. This was</td>
<td>5.733</td>
<td>5.644</td>
<td>5.443</td>
<td>5.231</td>
</tr>
<tr>
<td>recorded in a moving car with open windows and the radio turned on</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Two speakers (male and female) speaking together</td>
<td>10.935</td>
<td>10.911</td>
<td>10.686</td>
<td>10.385</td>
</tr>
<tr>
<td>3. Male reads separated words with disco music in the background</td>
<td>10.924</td>
<td>11.128</td>
<td>10.792</td>
<td>10.293</td>
</tr>
<tr>
<td>5. Single male speaker with clarinet music in the background</td>
<td>9.357</td>
<td>9.094</td>
<td>8.537</td>
<td>7.735</td>
</tr>
</tbody>
</table>

where the scaling factor between the output and the input samples for each frame is computed according to the maximum input sample in the frame. If the input power is low (less than $9,000,000$ for the sum of the squares of the samples in a sub-frame) the frame is ignored in the computation. This is because near silent signals are encoded with a comfort noise and thus SNR computation can not represent their reconstruction fidelity in these codecs.

The results are summarized in table 1. We must emphasize again that SSNR measurement does not represent the perceptual quality but it is an efficient tool for comparing among four different waveform codecs.

In a noisy background that does not have audible characteristics (segments 1 and 6) the SNR is low. We should take into account the fact that the encoder is tailored for audio sounds and even with these background noises the reconstructed speech is clear (shown in the subjective test in section 7.3).
Figure 17: SSNR values for all audio segments. Comparison between MLT and LCT-IV based codecs with 40 ms windows.

The MLT and the LCT-IV with 40 ms windows produced very similar SSNR results and the differences are negligible (see in Fig. 17 that average difference is 0.16 dB in favor of the MLT). These differences were in the computational error range (about 1.5% for the SSNR measurements). We see that the differences between MLT and LCT with the same window size are negligible.

Figure 18: SSNR comparison between different LCT-IV based codecs, which have different window sizes, for all the audio segments.

We see that when the window size decreases the SSNR decreases too (see Fig. 18). The SSNR degradation when reducing the window size of the LCT is relatively small. Figure 19 compares the relative differences between three different LCT-IV based codecs. The average differences in the SSNR results were:

- 4.98% between the LCT-IV with 30ms window size and the LCT-IV with 22.5ms window size.
- 3.69% between the LCT-IV with 40ms window size and the LCT-IV with 30ms window size.
- 8.45% between the LCT-IV with 40ms window size and the LCT-IV with 22.5ms window size.

We conclude that the SSNR differences between MLT and LCT-IV are negligible and are relatively small when the window size of the LCT-IV is reduced.
Figure 19: Differences in SSNR (%) between different window size LCT-IV based codecs for all the audio segments.

7.3 Subjective Tests

7.3.1 Perceptual Quality

This test was aimed to measure the perceptual quality of the new algorithm. Because the subjective tests were based on a subjective opinion of listeners and on a comparison between different new codecs, the ITU-T G729 codec reference codec was added. 1. New algorithm: 8kbps LCT-IV based codec with algorithmic delay of 40 ms. Using a 40 ms window (bell) with a 20 ms overlap (10 ms on each side) for the LCT-IV. 2. New algorithm: 8kbps LCT-IV based codec with algorithmic delay of 22.5 ms. Using a 22.5 ms window (bell) with a 2.5 ms overlap (1.25 ms on each side) for the LCT-IV. 3. G729 codec at 8kbps bit rate.

Three different audio segments were chosen for the test each with completely different audible characteristics. The segments were: 1 (denoted by 1) and 3 (denoted by 2) both from table 1 and the sentence of clean male speech followed by a clean female speech (denoted by 3).

15 different listeners from different ages and different genders participated in this test. Each listener was asked to grade 5 to 1 the section three times (for each codec), where 5 is the highest quality and 1 is the poorest quality. In the beginning of each test the source signal was played (as reference to grade 5) and afterwards the decompressed signals were played. High perceptual mark was given for the most comfortable experience for the listener’s ear and with the least required effort by him to understand it. The listeners were directed to refer to the speech reconstruction and not to the background sounds. For each listener the order of the compressed sections was different in order to reduce comparable effects in adjacent sections.

We averaged many kinds of listeners and by changing the order of the compressed audio sections that were played. Therefore, we should refer to these tests as a comparable relative grades between three different codecs and not as absolute grades. The results for all the three audio sections are
summarized in table 2.

<table>
<thead>
<tr>
<th>Tested Audio section</th>
<th>Average mark for LCT-IV 8kbps codec with 40 ms delay</th>
<th>Average mark for LCT-IV 8kbps codec with 22.5 ms delay</th>
<th>Average mark for G729 codec 8kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Conversation in a car</td>
<td>4</td>
<td>3.5</td>
<td>3.8</td>
</tr>
<tr>
<td>2. Words with disco music</td>
<td>4.3</td>
<td>3.5</td>
<td>3</td>
</tr>
<tr>
<td>3. Clean speech</td>
<td>3.7</td>
<td>3.3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2: Summary of perceptual quality tests results for the three audio segments

Even though the source signal was very noisy and it was very difficult to identify the speech in it all the three codecs did not crashed and preserved the reconstructed speech above the noise level. The new codec with the long delay performed better handling better the segments with multiple speakers. All three codecs managed to track the speech through the noisy background of the moving car most of the time. The new codec with the short delay was graded slightly below the other codecs, even though yielding good grades in comparison with the source signal. The advantage of the new codec was due to its ability to handle better multiple speakers. The new codec does not use a specific speech generation model as the G729 did, thus it is not speaker dependent. For the clean single speaker speech the G729 performed slightly better than the other codecs. The fact that G729 codec is based on a speech generation model causes it to be optimal in imitating speech generation of a single speaker speech. The other codecs were graded relatively high yielding acceptable reconstructed speech quality. The short delay codec was graded as the last one and the main difference was that it had a bigger distortion in some voiced phonemes (see section 6.5 for detailed explanation) especially in the male speech.

In the segment with the disco music the new codecs were graded better than the G729. The new codecs handled better the complex signal of music sounds and speech all together while the G729 sometimes had larger distortion of the signal.

### 7.3.2 Intelligibility

The goal of this test was to compare between the intelligibility of the LCT-IV based codec with long delay (40 ms) at 8kbps and the G729 codec at 8kbps in a very noisy environment. The goal was to test how well the reconstructed speech signal can be identified in a very low signal to noise ratio. This is a subjective test, but unlike the perceptual test, it can be quantified and it does not depend on the listener mood or grading strategy. This test depends only on the person’s earring capabilities. This test also isolates very well the listener’s reference to the background sounds, forcing the listener to concentrate on the reconstructed speech.
The test was performed on an audio segment that was composed of a conversation between two persons and a noise that was artificially added. The noise that was added included wide band noise composed of many audible components like radio, traffic noise and others all distributed in different points of the segment. The noise was adjusted to signal to noise ratio that makes the segment’s intelligibility very limited. This segment was fed to two codecs: G729 and LCT-IV with long delay (40ms), yielding output signals with very limited intelligibly, yet it was possible to identify some of the words in the conversation. Each listener was asked to write the words that he recognized in each decompressed section. The grades for each listener were the number of words that he or she recognized right. 15 listeners from different ages and genders participated in this test.

When we summarize the intelligibility tests then with the LCT-IV based codec an average of 22 words were correctly recognized. With the G729 codec an average of 7 words were extracted right.

In both cases, words that were recognized wrong were omitted from the count. In both cases, the words that were recognized were not part of a combined text. Therefore, it eliminates the possibility of extracting the rest of the words by deducing them from the context. The new codec obviously performed better. All the listeners found the LCT-IV based codec more comfortable for listening. The listeners missed many words with both codecs due to the fact that the words were masked very deeply by the background noise.

8 Conclusions

The LCT-IV based codec was tested with diverse audio scenarios. Complex noises were artificially added to different speech segments to create difficult background noises. The performance of the LCT-IV based codec was tested against the popular ITU-T G729 recommendation. In most of the cases the LCT-IV based codec outperformed the G729 standard in preserving better speech perceptual quality and intelligibility under these bad audible conditions. Differences in the performance were noticed with music signals and multiple speakers. The G729 codec performed better with clean speech. The G729 and the LCT-IV based codecs have the same order of complexity. Both can be implemented on a DSP with less than 15 MIPS.

Different delays from different overlapping sizes and different window sizes in the LCT-IV based codecs created different delays for the codec. The LCT-IV based codec with the longest delay of 40 ms had the best performance. The 22.5 ms LCT-IV based codec had minor degradations in comparison to the performance of the 40 ms LCT-IV based codec. The degradation in the speech reconstruction quality due to the delay reduction were acceptable. The main differences in the performance were with clean speech in voiced phonemes with only few basic harmonics. The new 8kbps codec was also examined using the MLT transform with 40 ms window and 50% overlapping between adjacent frames.
No noticeable differences were found with respect to the 40 ms LCT-IV based 8kbps codec. The results were very similar in all kinds of audio scenarios and in all kind of speech components (voiced phonemes and unvoiced phonemes). This result is interesting if we take into account the fact that the MLT was optimized from coding gain point of view (what makes it so popular in many transform codecs nowadays). At low bit rates the coding gain advantage of the MLT is not noticeable due to the smaller correlation between samples at these rates.

There are couple of interesting directions for further research such as: Modification of the quantization and the entropy coding phases of the encoder to fit better the LCT-IV properties, devising other bells that can be better optimized, improvement of the bit allocation procedure (categorization), incorporation of pitch parameter, using multiscale version of the LCT with the best basis methodology.

Audioable results that demonstrate the performance of the proposed codec will be supplied by the first author.

References


