REAL-TIME SPEECH COMPRESSION BY USING CODE EXCITED LINEAR PREDICTION ALGORITHM

S.Prabu[1], S.Nandakumar[2]

[1]School of Computing Sciences, VIT University, Vellore, Tamilnadu
[2] School of Electrical Sciences, VIT University, Vellore, Tamilnadu
E- Mail: [1]sprabu@vit.ac.in, [2] snandakumar@vit.ac.in

Abstract

A lot of effort has been spent over the last few years in the development of digital speech coding methods and their subsequent standardization. Algorithms have evolved which provide good quality speech at sub 8 kbps bit rates although at a much computational expense. Speech compression is proposed based on code excited linear prediction algorithm and implementation in DSP algorithm. Algorithm based on three-stage technique which involves simulate, evaluate, debug and implementation in G.723 low delay code excited linear prediction (LD-CELP)[4] algorithm. First stage, the algorithm is evaluated via simulation to determine whether it meets the design criterion. Then, it is implemented in real-time based an object oriented approach. After the algorithm is thoroughly tested, it is further refined to obtain tighter and faster coding. This technique can be applied to other real-time DSP algorithms. A simulation result shows that better speech quality is obtained. The techniques described in this paper are applicable to any other speech codec.

Key words: *:* Low delay Code excited Linear Prediction (LD-CELP) algorithm, DSP, Speech Compression, G.723.1 Standard

I. INTRODUCTION

A Selection of a Low Bit Rate Vocoder.

The three most important approaches are waveform coding, transform coding, and parametric coding. Waveform Coding basically does compression of sample at the transmitting end and de-compression of sample at the receiving end. Parametric Coding relies on speech characteristics. Transform Coding as the name indicates compresses speech by employing a transformation technique. Nowadays, most successful Vocoder and hybrid coders make use of linear prediction in order to estimate the needed parameters. Indeed, LPC in[5] (Linear Prediction Coding) is the most successful method for encoding at low bit rate and is used in many applications.

The comparison chart Table 1 summarizes the performance of various algorithms. As can be seen G.723.1[1] tops out as one of the better algorithms offering communication quality speech at a relatively low bit rate. Mean Opinion Score (MOS) is a subjective measure of the performance of the algorithm. An introduction to LPC and CELP [4] is given as these are assumed for G.723.1 (ACELP/MP-MLQ)

Table 1. Comparison of Speech Coding AlgorithmB. CELP

Standard	Coding Type	Bit Rate(kbps)	MOS	Algor. Delay(ms)
ITU-G.711	PCM	64	4.3	0.125
ITU-G.721	ADPCM	32	4.0	0.125
ITU-G.723.1	ACELP/ MP-MLQ	6.3,5.3	3.8	37.5
ITU-G.726	VBR-ADPCM	16,24,32,40	2.0,3.2,4, 4.2	0.125
ITU-G.728	LD-CELP	16	4.0	0.625

Standard	Coding Type	Bit Rate(kbps)	MOS	Algor. Delay(ms)
ITU-G.711	PCM	64	4.3	0.125
ITU-G.721	ADPCM	32	4.0	0.125
ITU-G.723.1	ACELP/ MP-MLQ	6.3,5.3	3.8	37.5
ITU-G.726	VBR-ADPCM	16,24,32,40	2.0,3.2,4, 4.2	0.125
ITU-G.728	LD-CELP	16	4.0	0.625

CELP stands for Code Excited Linear Prediction and starts from basic LPC coding. It is the most commonly used coder in telephony. There are many ways in which the residual and the LP coefficients are used: CELP [9] is the extreme in terms of complexity. The LP parameters are estimated as before, and used to form the synthesis filter. However, the synthetic speech is obtained from a synthetic residual that differs from LPC: it is obtained from a codebook. The residual is therefore vector quantized and chosen from a pre computed set of excitations. Only the index of the excitation used from the codebook has to be sent to the receiver. Vector quantization exploits the high correlation between the parameters and is applied to both the coefficients and the excitation. The system looks like Fig 1.

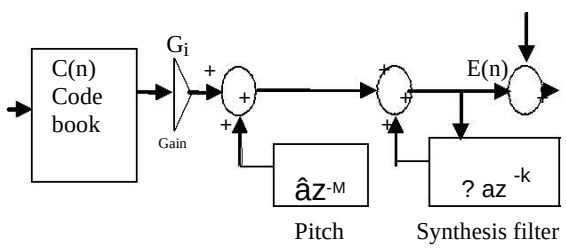

Fig. 1. Block Diagram of CELP

In CELP, the pitch is not encoded in the codebook: a long-term predictor is used instead to reduce the complexity of the codebook. Finding the appropriate LP coefficients and excitation requires a search through the codebooks. Each entry is evaluated by calculating a perceptual error: the best one is chosen. The increase in efficiency and quality is big, but the complexity is high: searching the codebook can be long, and requires a lot of computations.

II. OBJECTIVE

The objective of this paper is to design architecture for a low bit rate codec G.723.1 recommendation (ACELP/MP-MLQ) [2]. The description of the speech coding algorithm is made in terms of bit-exact, fixed point mathematical operations.

A. Simulation of G.723.1 Encoder

This coder has two bit rates associated with it. These are 5.3 and 6.3 kbit/s. The higher bit rate has greater quality. The lower bit rate gives good quality and provides system designers with additional flexibility. Both rates are a mandatory part of the encoder and decoder.This coder was optimized to represent speech with a high quality at the above rates using a limited amount of complexity. Music and other audio signals are not represented as faithfully as speech, but can be compressed and decompressed using this coder. This coder encodes speech or other audio signals in 30 msec frames. In addition, there is a look ahead of 7.5 msec, resulting in a total algorithmic delay of 37.5 msec. All additional delays in the implementation and operation of this coder are due to:

I) Actual time spent processing the data in the encoder and decoder

ii) Transmission time on the communication link

iii) Additional buffering delay for the multiplexing protocol.

The description of the speech-coding algorithm of this Codec is made in terms of bit-exact, fixed-point mathematical operations

B Encoder principle

The coder is based on the principles of linear prediction analysis-by-synthesis coding and attempts to minimize a perceptually weighted error signal. The encoder operates on blocks (frames) of 240 samples each. That is equal to 30 msec at an 8 kHz sampling rate. Each block is first high pass filtered to remove the DC component and then divided into four sub frames of 60 samples each. For every subframe, a 10th order Linear Prediction Coder (LPC) [5,7] filter is computed using the unprocessed input signal. The LPC filter for the last subframe is quantized using a Predictive Split Vector Quantizer (PSVQ). The unquantized LPC coefficients are used to construct the short-term perceptual weighting filter, which is used to filter the entire frame and to obtain the perceptually weighted speech signal.

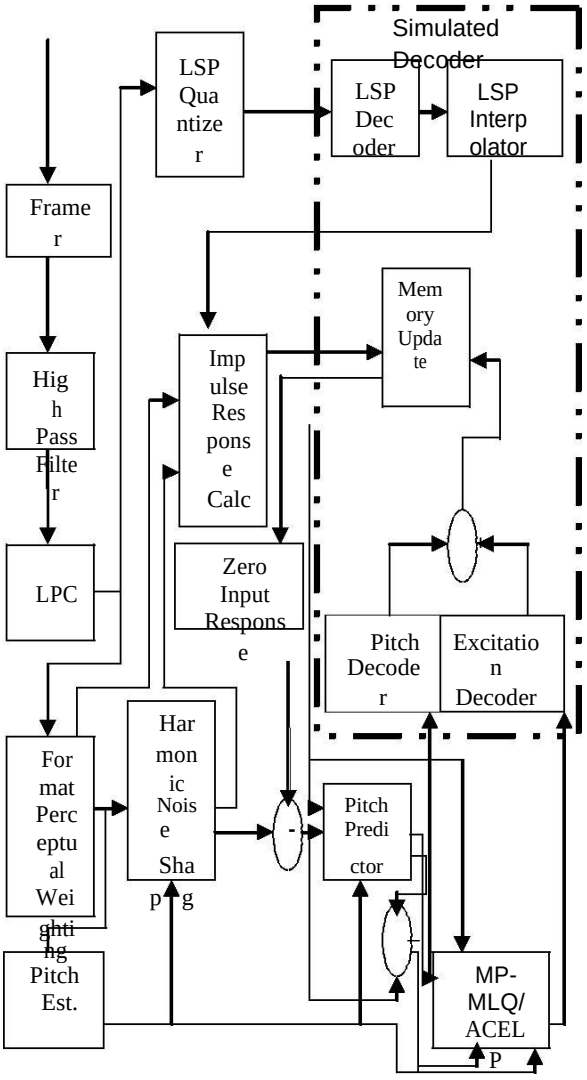

Fig. 2. Block diagram of the Speech Encoder

For every two subframes (120 samples), the open loop pitch period, LOL, is computed using the weighted speech signal. This pitch estimation is performed on blocks of 120 samples. The pitch period is searched in the range from 18 to 142 samples. From this point the speech is processed on a 60 samples per subframe basis. Using the estimated pitch period computed previously, a harmonic noise shaping filter is constructed. The combination of the LPC synthesis filter, the formant perceptual weighting filter, and the harmonic noise shaping filter is used to create an impulse response. The impulse response is then used for further computations.

Using the pitch period estimation, LOL, and the impulse response, a closed loop pitch predictor is computed. A fifth order pitch predictor is used. The pitch period is computed

as a small differential value around the open loop pitch estimate. The contribution of the pitch predictor is then subtracted from the initial target vector. Both the pitch period and the differential value are transmitted to the decoder. Finally the non-periodic component of the excitation is approximated. For the high bit rate, Multi-pulse Maximum Likelihood Quantization (MP-MLQ) excitation is used, and for the low bit rate, an algebraic-code-excitation (ACELP) is used.

III. DECODER

The G.723.1 speech decoder is divided into the following modules:

- Decoder, which includes the initialization routines as well as the decoder
- Line Spectrum Analysis, which includes LSP decoder and interpolation
- Linear Predictive Analysis, which includes LPC synthesis and Formant post filter
- Adaptive and Fixed Excitation, which includes decoding of pitch information, Excitation decoder, pitch post filter, and frame interpolation
- Miscellaneous utility functions, which include gain scaling.

A. Decoder principles

The decoder operation is also performed on a frame-by-frame basis. First the quantized LPC indices are decoded, then the decoder constructs the LPC synthesis filter. For every subframe, both the adaptive codebook excitation and fixed codebook excitation are decoded and input to the synthesis filter. The adaptive post filter consists of a formant and a forward-backward pitch post filter. The excitation signal is input to the pitch post filter, which in turn is input to the synthesis filter whose output is input to the formant post filter. A gain scaling unit maintains the energy at the input level of the formant post filter.

B. Vector Quantization

This section is to introduce vector quantization employed in gain and LSP quantization. Before introducing vector quantization, lets look at what scalar quantization is. In Scalar Quantization one represents the values by fixed subset of representative values. For example, if we have 16 bit values and we send only 8 most signifcant bits, we get an approximation of the original data at the expense of precision. In this case the fixed subset is all the 16-bit numbers divisible by 256, i.e 0, 256, 512,...Vector Quantization (VQ) is a generalization of scalar quantization to higher dimensions. This generalization opens up a wide range of possibilities and techniques not present in the scalar case. Unlike scalar quantization, VQ is usually applied to signals that have already been digitized. It is primarily used for data compression and pattern recognition. In VQ, an input pattern or word is matched to a set of stored patterns or words, and the best match is chosen. The index of the 'best match' can then be transmitted, thereby reducing the amount of data that needs to be transmitted.

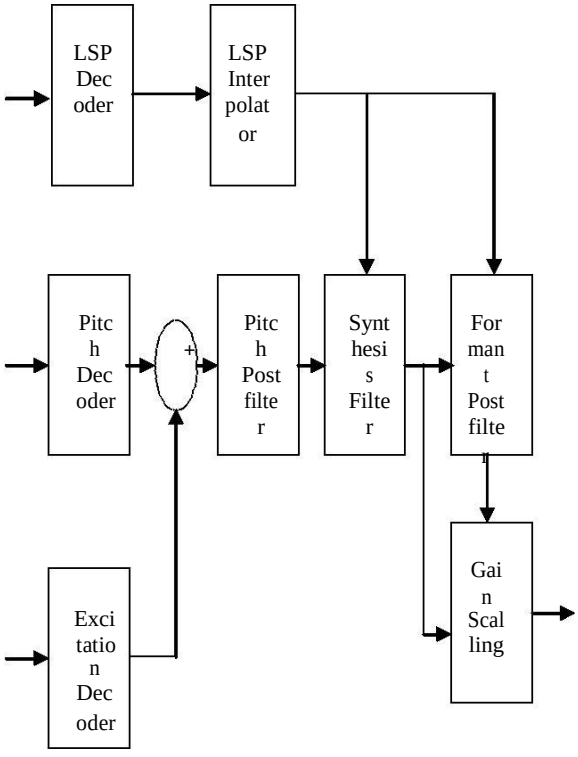

Fig. 3. Block diagram g.723.1 speech decoder

Mathematically, a vector quantizer, Q of dimension k and size N is defined as a mapping from a vector in k-dimensional Euclidean space, Rk, into a finite set of vectors, or codeword. That is:

$$Q : R^k \rightarrow C \quad [1]$$

Where C = (y1,y2,...,yN) is the codebook with codeword yi ??Rk for i ??{1, 2, ... ,N}.

Vector Quantization requires both an encoder and decoder. The encoder E maps the input Rk into the index set, I:

$$E : R^k \rightarrow I \quad [2]$$

The encoder thus outputs an index to the codeword that offers lowest distortion. In this case the lowest distortion is found by evaluating the Euclidean distance between the input vector and each codeword in the codebook. Once the

closest codeword is found, the index of that codeword is sent through a channel. The decoder D simply maps the index set I into the reproduction set C:

D : I C [3]

The decoder is a simple table lookup.It does not need to know anything about the partition cells of Rk. A Voronoi or nearest neighbor vector quantizer is a special class of vector quantizer in which the partition is completely determined by the codebook and a distortion measure. The nearest neighbor vector quantizer is, in fact, the most common type of vector quantizer in practice.

$$d(x,y)=\sum_{i=1}^{k} (xi - yi)2 \quad [4]$$

The most common distortion measure used in nearest neighbor vector quantizers is mean square error, which is defined by the Euclidean distance between vectors.

When large data sets are involved, especially when consecutive points are correlated in some way then this method finds its use. Speech compression schemes like CELP use this scheme to quantize the excitation vectors.

IV. SIMULATION OF G.723.1 DECODER

The G.723.1 does no searching, so is computationally much less complexity than the coder. The decoding process basically deals with extraction of the parameters using the codebook indices and reconstruction of the speech signal. This chapter explains the complete process of G.723.1 decoder and its simulation using MATLAB .

A. Decoder overview

First the frame parameters LP coefficients, adaptive-codebook vector, fixed-codebook vector and gain are decoded. Then these decoded parameters are used to reconstruct the speech signal. This reconstructed speech signal is enhanced by a post processing process.

The LP decoding process is done per each frame whereas the following steps are repeated for each sub frame decoding of the adaptive-codebook vector decoding of the fixed-codebook vector decoding of the adaptive and fixed-codebook gains Computation of the reconstructed speech.

B. Decoding of LP coefficients

The received indices L0, L1, L2, L3 and the code books lspcb1, lspcb2 are used to reconstruct the quantized LSP coefficients. The sum of the value of lspcb1 codebook at index L1 and the value of lspcb2 codebook at the index L2 for the first five coefficients and at L3 for the next five coefficients are obtained. These coefficients are rearranged for a minimum distance of J. The rearrangement process is done twice. First with a value of J = 0.0012, then with a value of J = 0.0006. After this rearrangement process, the quantized LSF coefficients for the current frame are obtained using the 4th order MA prediction filter as explained in the encoder. The selection of MA prediction filter of the available two filters is done based on the value of L0. After computing the LSF coefficients (ùi), the corresponding filter is checked for stability based on the following conditions

1. Arrange the coefficient ùi in increasing value.
2. if ùi < 0.005 then ùi = 0.005.
3. if ùi+1 - ùi – 0.0391 then ùi+1 = ùi + 0.0391, for I = 1,…,9
4. if ù10 > 3.135 then ù10 = 3.135.

The LSP coefficients are the cosine of the LSF coefficients.

C. Decoding of adaptive and fixed codebook gains

The received gain-codebook indices GA and GB along with the codebooks are used to decode the gains. Then the estimated fixed codebook gain is obtained using a 4th order MA prediction filter. The excitation u(n) is computed using v(n), c(n), gp and gc using the following equation

U(n) = gp * v(n) + gc * c(n) [5]

D. Computing the reconstructed speech

The excitation u(n) is passed through a 10th order synthesis filter whose coefficients are given by the LP coefficients a. The output from this LP filter is the reconstructed speech.

E. Post processing

Post-processing consists of three functions namely adaptive post-filtering, high-pass filtering and signal upscaling. The adaptive post-filter is the cascade of three filters: a long-term postfilter Hp(z), a short-term postfilter Hf (z) and a tilt compensation filter Ht(z), followed by an adaptive gain control procedure. The postfilter coefficients are updated every 5 ms subframe. The postfiltering process is organized as follows. First, the reconstructed speech is inverse filtered to produce the residual signal r^(n). This signal is used to compute the delay T and gain gt of the long-term postfilter Hp (z). The signal r^(n) is then filtered through the long-term postfilter Hp (z) and the synthesis filter1/[gf Â(z /gd)]. Finally, the output signal of the synthesis filter 1/[gf Â(z /gd)] is passed through the tilt compensation filter Ht (z) to generate the post-filtered reconstructed speech signal sf(n). Adaptive gain control is then applied to sf(n) to match the energy of s^(n).

F. High-pass filtering and up scaling

The output from the adaptive gain control unit sf'(n) is applied to a high-pass filter with a cut-off frequency of 100 Hz. This is followed by upscaling by a factor of two to compensate for the down scaling performed in the encoder to prevent fixed point overflow. Thus the completely processed speech signal is obtained. From the simulation results we found that the output from the decoder is as similar as the original speech and normal hearing cannot pickup any disturbances.

V. RESULTS AND PERFORMANCE ANALYSIS

ENCODER PERFORMANCE

The results show excitation, residual and the codebook vectors that form the excitation. As we already know, an approximation of residual that is used to reconstruct the speech is excitation. This excitation is composed of two parts adaptive codebook vector and fixed codebook vector. The above diagram shows that, adaptive codebook vector tries to follow the residual in a slow manner whereas the fixed codebook vector has the ability to spot sudden changes. In essence, the waveform shown above combined with LP parameters already coded signifies the operation of G.723.1 encoder.

```
name =MALE.WAV
Warning: Function call  modenc invokes inexact match C:\MATLAB7\work\mail send with encode and dec\MODENC.M.
Matlab simulation of speech coder
Start the encoding process
enter the input file--->male.wav
enter the output file--->>> x.dat
> In path at 115
  In addpath at 95
  In G7231Coder at 9
> In SetCoderPar at 132
  In G7231Coder at 25
  In MODENC at 15

HPFilt = b: [1 -1]
     a: [1 -0.9922]
   Mem: []

LPpar =  Rnn: [11x1 double]
     Win: [180x1 double]
  LagWin: [11x1 double]
   ECWin: [11x1 double]
  FStart: 60
  WStart: [-60 0 60 120]
   SFRef: 4
    LMem: 120
LSFpar =  ECWin: [11x1 double]
    Mean: [10x1 double]
    Pcof: 0.3750
      VQ: {[3x256 double]  [3x256 double]  [4x256 double]}
     Fix: [1x1 struct]
    IntC: [0.2500 0.5000 0.7500 1]
    lsfQ: [10x1 double]

TVpar = LSubframe: [60 60 60 60]
        PWpar: [1x1 struct]
  POLSubframe: [120 120]
   PitchOLpar: [1x1 struct]
       HNWpar: [1x1 struct]

SineDetpar =rc: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
    NThr: 14
  rcThr: 0.9500
   Pitchpar =  PMode: [4x1 double]
     PMin: [18 18]
     PMax: [141 142]
    LOffs: {[-1 0 1]  [-1 0 1 2]}
  CBookThr: 58
       b: {[5x85 double]  [5x170 double]}
       CL: [24 2048 4096]
   Tamepar: [1x1 struct]
    POffs: [-2 2]
     eMem: [146x1 double]
```

MPpar = g: [24x1 double]
Grid: {{1x2 cell} {1x2 cell} {1x2 cell} {1x2 cell}} Np: [6 5 6 5]
gIOffs: [-2 1]
LThr: 58
ModV: [65536 16384 65536 16384]
CL: [8100 810 90 9]
nCk: [31x7 double]

Clippar = MinThr: -1.0000
MinVal: -1
MaxThr: 1.0000
MaxVal: 1.0000

WAVE file: C:\MATLAB7\work\mail send with encode and dec\MALE.WAV
Number of samples : 21998 (2.746 s)
Sampling frequency: 8012
Number of channels: 1 (16-bit integer)
Frame: 1
Frame: 2
Frame: 3
Frame: 4
Frame: 5
Frame: 6
Frame: 7
Frame: 8
Frame: 9
Frame: 10
Frame: 11
Frame: 12
Frame: 13
Frame: 14
Frame: 15
Frame: 16
Frame: 17
Frame: 18
Frame: 19
Frame: 20
Frame: 21
Frame: 22
Frame: 23
Frame: 24
Frame: 25
Frame: 26
G723.1 data file: C:\MATLAB7\work\mail send with encode and dec\x.dat
Elapsed time is 7.641000 seconds.

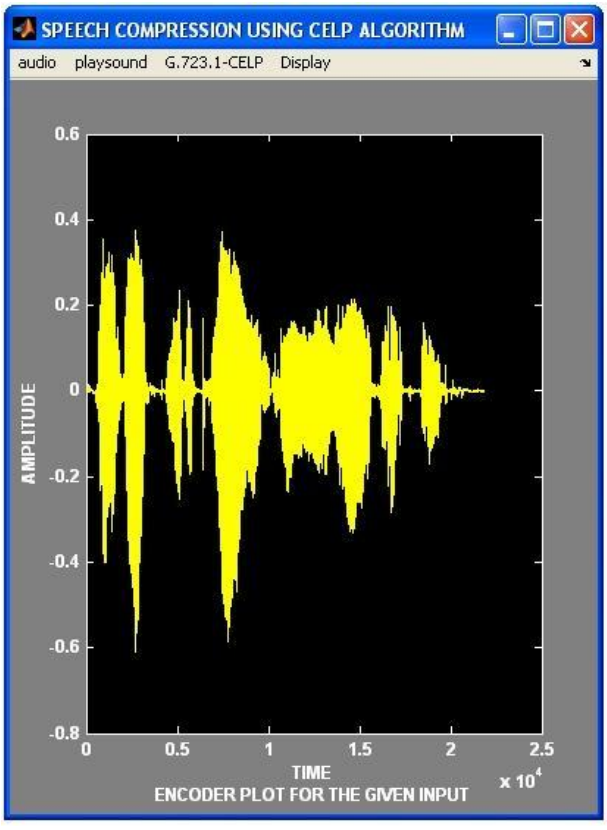

Fig. 4. 1Encoder Performance

A. Decoder Performance

The decoder part reconstructs the same excitation from the adaptive codebook vector and fixed codebook vector. This is first synthesized to form a reconstructed speech. This is used to get the residual. Post filter operates upon this residual. As can be seen there is not much difference in the

residuals of both encoder and decoder stage.

Mat lab simulation of speech coder

Start the decoding process

enter the input file--->Warning: Function call moddec invokes inexact match C:\MATLAB7\work\encode and dec\MODDEC.M.

inputfile=.dat,outputfile=.wav

enter the input file--->>> x.dat

enter the output file--->m1.wav

> In path at 115

In addpath at 95

In G7231Decoder at 8

In MODDEC at 6

In MODENC at 12

LSFpar = ECWin: [11x1 double]

Mean: [10x1 double]

Pcof: 0.3750

VQ: {[3x256 double] [3x256 double] [4x256 double]}

Fix: [1x1 struct]

IntC: [0.2500 0.5000 0.7500 1]

lsfQ: [10x1 double]

Pitch par = PMode: [4x1 double]

PMin: [18 18]

PMax: [141 145]

LOffs: {[-1 0 1] [-1 0 1 2]}

CBookThr: 58

b: {[5x85 double] [5x170 double]}

CL: [24 2048 4096]

POffs: [-2 2]

eMem: [149x1 double]

MPpar = g: [24x1 double]

Grid: {{1x2 cell} {1x2 cell} {1x2 cell} {1x2 cell}}

Np: [6 5 6 5]

gIOffs: [-2 1]

LThr: 58

ModV: [65536 16384 65536 16384]

CL: [8100 810 90 9]

nCk: [31x7 double]

Clippar =

MinThr: -1.0000

MinVal: -1

MaxThr: 1.0000

MaxVal: 1.0000

Frame: 1

Frame: 2

Frame: 3

Frame: 4

Frame: 5

Frame: 6

Frame: 7

Frame: 8

Frame: 9

Frame: 10

Frame: 11

Frame: 12

Frame: 13

Frame: 14

Frame: 15

Frame: 16

Frame: 17

Frame: 18

Frame: 19

Frame: 20

Frame: 21

Frame: 22

Frame: 23

Frame: 24

Frame: 25

Elapsed time is 0.750000 seconds.

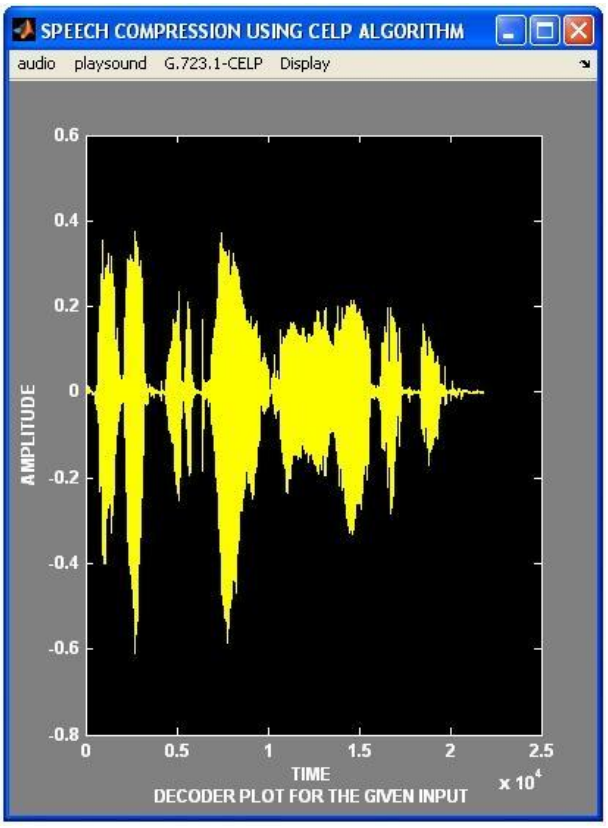

Fig. 5. Decoder Performance

VI. CONCLUSION

Architecture for G.723.1 encoder and decoder has been designed and it's performance for various input speech signals are verified using MATLAB simulation. The verification was done using 16-bit linear PCM samples from a wav file and the output written to a wav file. The input speech signal is compressed efficiently by the encoder and it is reconstructed to the form of original speech by the decoder. The reconstructed speech resembles the original input speech signal. The results obtained indicate the best performance of G.723.1 architecture designed for speech compression.

REFERENCES

[1] J.Cambell,T.Termain and V.Welch , The DoD 4.8 Kbps Standard (proposed Federal Standard 1016)", in Advances in Speech Coding ,ed.B.Atal, V.Cuperman and A.Gersho , Kluwer Academics Publishers , 1990

[2] ITU-T Recommendation G.723.1, "Dual Rate Speech Coder for multimedia Communications Transmitting at 5.3and 6.3 kbitls," Mar. 1996.

[3] Implementation of G.723.1 on the TMS320C54x – Application report from TI Spra656 – March 2000.

[4] Implementation of G.723.1 on the TMS320C54x – Application report from TI Spra656 – March 2000

[5]. J. Wang and J. D. Gibson, "Performance comparison of intraframe and interframe LSF quantization in packet networks, *Proc. 2000 IEEE Workshop on Speech Coding,*Delavan, WI, USA, September 2000

[6]. Motivation from a Full-Rate Specific Design to a DSP Core Approach for GSM Vocoders - Shervin Sheidaei ,Hamid Noori,Ahmad Akbari,Hosein Pedram Voiceage.com – Free library provider for G.723.1.

[7]. J.Cambell,T.Termain and V.Welch,1990 , The DoD 4.8 Kbps Standard (proposed Federal Standard 1016)", in Advances in Speech Coding ,ed.B.Atal, V.Cuperman and A.Gersho , Kluwer Academics Publishers.

[8] Real time imlpementation and evaluation of variable rate celp coders ETSI Telecommunication –U .Vigo Apartado ,62,36280 Vigo SPAIN . Pedram Voiceage.com – Free library provider for G.723.1.

S.Prabu received the B.E degree in Computer Science and engineering from Sona College of Technology, Salem, India in 2002 and the M.Tech degree in Remote Sensing and Geographical Information Systems (GIS) from College of Engineering Guindy, Anna University, Chennai, India, in 2004. Presently, he is working with School of Computing Sciences, VIT University, Vellore, India as a Assistant Professor (Sr.Grade).