Simulation of Codec for Adaptive Linear Prediction

Rumen P. Mironov[1]

Abstract – **The software model of Adaptive Linear Prediction Codec for one dimensional signals is developed. The presented codec is simulated on Simulink for Matlab 6.5 environment and the obtained results for real sound signals are evaluated by the calculation of MSE and SNR for the decoded signals.**

Keywords – **Digital Signal Processing, Linear Prediction, Adaptive Data Compression, Matlab Simulation.**

I. INTRODUCTION

The basic methods for digital signal compression are lossless and lossy compression. Due to the low coefficient of information reduction, the first group of methods is used most frequently in archiving systems and didn't find widespread use in the information transmission systems. The methods for lossy compression are divided into three groups: statistical, psychoacoustic and transforms methods ([1], [2], [3] and [4]). The psychoacoustic compression methods are based on the shortcomings of the human hearing system and are suitable only when the recipient of the record signals is a man. The transformation methods provide a high degree of information reduction and work mainly on the basis of unitary transformations ([3], [4]), for which after decompression the specific block distortions are received. Statistical compression methods are based on the reduction of the information redundancy of the transmitted signals and proceeds in two stages: decorrelation of communication signals and reduction of binary digits, necessary for transmision of signals ([1], [2]). One of the simplest and most convenient for practical implementation methods for decorrelation is the method for linear prediction ([2], [5]).

An adaptive method for coding of one dimensional digital signals, based on the linear prediction and Least Mean Square (LMS) weight coefficients adaptation [6] of the prediction filter is presented. From the developed mathematical equations a general block scheme of Line Prediction Codec (LPC) is synthesized and experimental results from the simulation by Simulink for Matlab 6.5 environment for test signals are given.

II. MATHEMATICAL DESCRIPTION

We will assume that the correlation covers n neighborhood elements of the input digital signal, represented by the stationary series $\{x(i)\}$ with $N > n$ values, which have zero

---
[1]Rumen P. Mironov is with the Faculty of Telecommunications, Technical University of Sofia, Boul. Kl. Ohridsky 8, Sofia 1000, Bulgaria, E-mail: rmironov@tu-sofia.bg

average component and correlation function $R_x(r)$ for: $r = \overline{0, n-1}$. Then the basic equation of the linear prediction will be presented by the following way:

$$\hat{x}(i) = a_1 x(i-1) + \ldots\ldots\ldots + a_{n-1} x(i-n+1) = \sum_{k=1}^{n-1} a_k x(i-k), \quad (1)$$

where $\hat{x}(i)$ is the value of the predicted element from the input signal $x(i)$. The prediction error is described by the equation:

$$e(i) = x(i) - \hat{x}(i), \quad (2)$$

and the quantization error - by the equation:

$$e_q(i) = Q[e(i)]. \quad (3)$$

According to the LMS algorithm of Widrow [6], the weight coefficients a_k of the prediction filter can be determined recursively:

$$a_k(i) = a_k(i-1) - \mu.\nabla_k = a_k(i-1) - \mu \frac{\partial e^2(i)}{\partial a_k}, \quad (4)$$

where: ∇_k is the gradient of the squared error by the prediction and μ is the adaptation step for the i^{-th} step of adaptation.

The derivative of the prediction error is determined by the Eqs. (1), (2), (3), and (4). The sequence for the components of weights is given by:

$$a_k(i) = a_k(i-1) + 2\mu e(i) x(i-k). \quad (5)$$

The beginning values of the weighted coefficients can be calculated after the minimization of mean square error of the prediction. From Eqs. (1) and (2) follows:

$$\overline{e^2(i)} = \frac{1}{N} \sum_{i=0}^{N-1} [x(i) - \hat{x}(i)] = E\left\{ \left[x(i) - \sum_{k=1}^{n-1} a_k x(i-k) \right]^2 \right\}, \quad (6)$$

where E is the averaging operator.

The partial derivative of $\overline{e^2(i)}$ with respect to any weight coefficient a_l can be expressed as:

$$\frac{\partial \overline{e^2(i)}}{\partial a_k} = \frac{\partial E\left\{ [x(i) - (a_1 x(i-1) + \ldots.. a_{n-1} x(i-n+1))]^2 \right\}}{\partial a_l}.$$
$$= -2E\left\{ [x(i) - (a_1 x(i-1) + \ldots.. a_{n-1} x(i-n+1))] x(i-l) \right\}$$

Once equated to zero and transformed, from the upper expression is obtained:

$$E\{x(i)x(i-l)\} = \sum_{k=1}^{n-1} a_k E\{x(i-k)x(i-l)\}. \quad (7)$$

The autocorrelation function of the digital signal, presented with the series $\{x(i)\}$ is:

$$R_x(r) = \frac{1}{N}\sum_{i=0}^{N-1} x(i)x(i-r) = E\{x(i)x(i-r)\}. \quad (8)$$

From the Eqs. (7) and (8) the following expression is obtained:

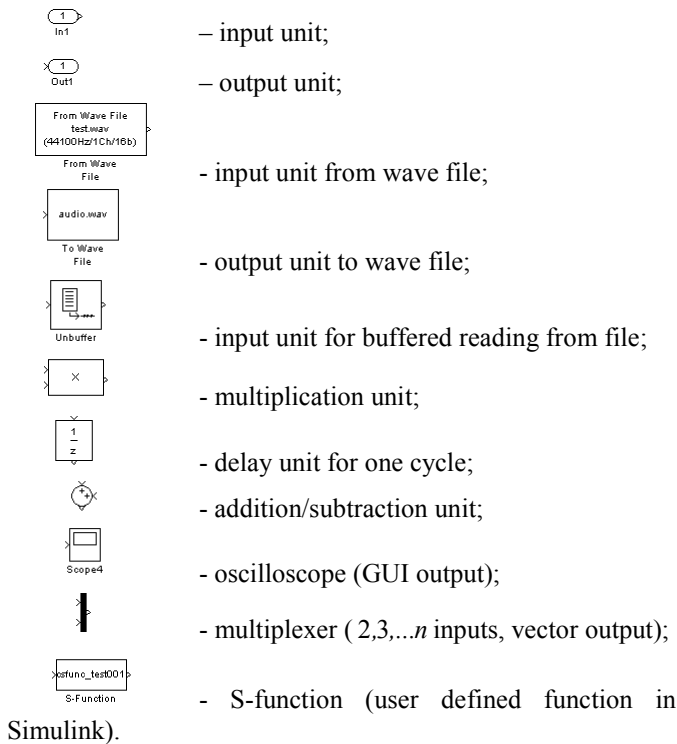$$R_x(l) = \sum_{k=1}^{n-1} a_k R_x(k-l). \quad (9)$$

For $l = \overline{1, n-1}$ equation (9) represent a linear system with $n-1$ unknowns.

The following equations are used for decoding of signals:

$$\mathbf{x}'(i) = \mathbf{e}'(i) + \hat{\mathbf{x}}'(i), \quad (10)$$

$$\mathbf{e}'(i) = Q^{-1}[e_q(i)]. \quad (11)$$

The synthesized by the Eqs. from (1) to (11) adaptive line prediction codec (ADPCM) from 2nd order ($k = 2$ and provided that there is no quantization unit) is shown on Fig. 1. On Fig. 2 and Fig.3 the synthesized coder and decoder blocks for ADPCM are shown respectively. The presented schemes are developed through the Simulink package for Matlab 6.5 environment and included the following units:

– input unit;

– output unit;

- input unit from wave file;

- output unit to wave file;

- input unit for buffered reading from file;

- multiplication unit;

- delay unit for one cycle;

- addition/subtraction unit;

- oscilloscope (GUI output);

- multiplexer ($2,3,...n$ inputs, vector output);

- S-function (user defined function in Simulink).

III. Experimental Results

The developed ADPCM codec is used for simulation on Matlab 6.5 environment of real audio signals (WAV file format, 1 channel (mono), 16 bits, 44.1 KHz sampling rate). The starting values of the weights are obtained from the correlation matrix:

$$[R_x] = \begin{bmatrix} R_x(0) & R_x(1) \\ R_x(1) & R_x(0) \end{bmatrix}$$

for autocorrelation functions $R_x(0), R_x(1), R_x(2)$, calculated from the Eq.(8). The starting weight vector $\vec{A}(0) = [a_1(0), a_2(0)]^t$ is calculated from the equation:

$$[R_x]A(0) = \begin{bmatrix} R_x(1) \\ R_x(2) \end{bmatrix},$$

by the Kramer's formulas: $a_1(0) = \frac{\Delta_1}{\Delta}$, $a_2(0) = \frac{\Delta_2}{\Delta}$.

Using Eq.(5) the values of the weight vector for the next iterations are calculated by the following:

$$\begin{bmatrix} a_1(i) \\ a_2(i) \end{bmatrix} = \begin{bmatrix} a_1(i-1) \\ a_2(i-1) \end{bmatrix} + \mu.e(i).\begin{bmatrix} x(i-1) \\ x(i-2) \end{bmatrix},$$

where the errors are calculated from the Eq.(2) and the predicted values are calculated from:

$$\hat{x}(i) = a_1(i-1).x(i-1) + a_2(i-1).x(i-2).$$

The adaptation step depends on the inequality: $0 < \mu < \frac{1}{R_x(0)}$ and the value μ=0.01 is chosen.

The mean squared error is calculated by the equation:

$$\overline{\varepsilon^2} = \frac{1}{N}.\sum_{i=1}^{N} e^2(i).$$
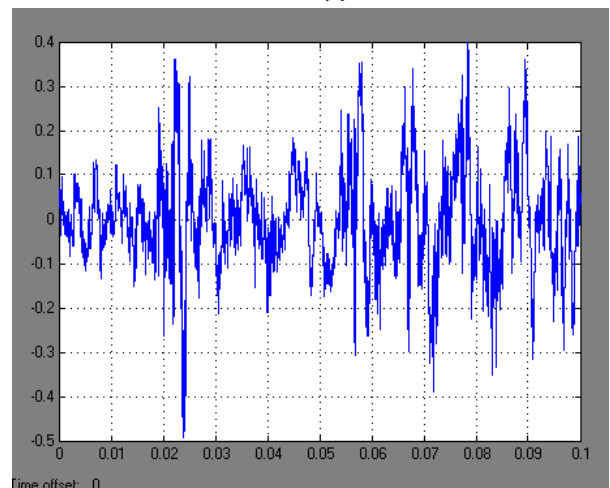

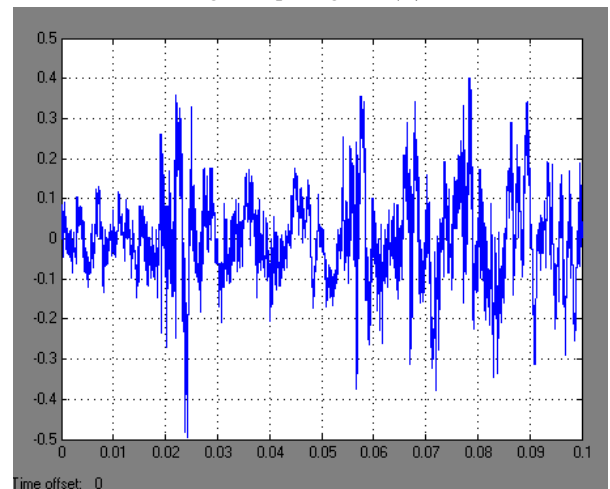Fig. 4. Input signal $x(i)$

Fig. 5. Output predicted signal $\hat{x}(i)$.

The input test signal and output predicted signal are shown on Fig.4 and Fig.5 respectively. On Fig.6 the input and output signals are shown together (visualized from the oscilloscope Scope 4). On Fig.7 the same signals are visualized by the zooming in horizontal direction (from 0.031s to 0.0324s).
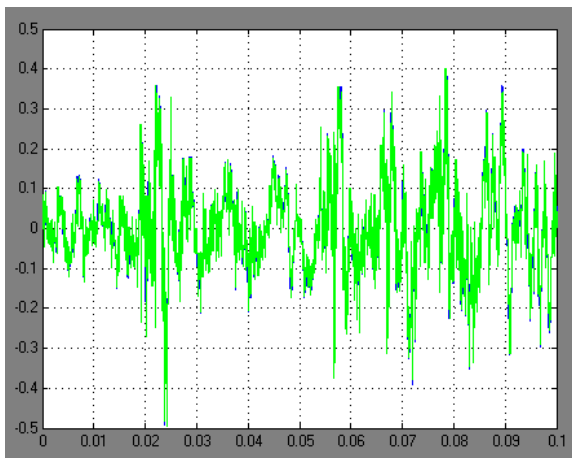
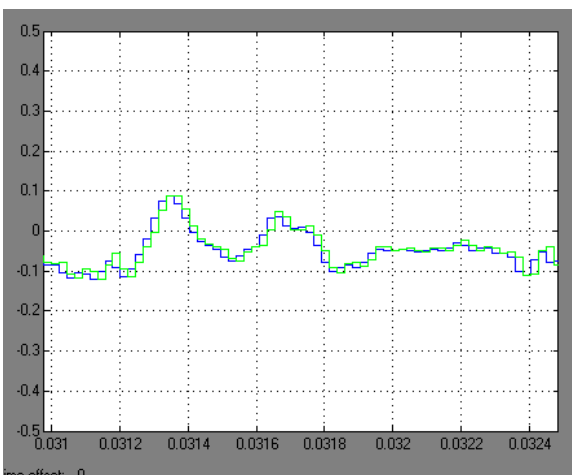Fig. 6. Input and output predicted signal MUX{ $x(i)$, $\hat{x}(i)$ }.

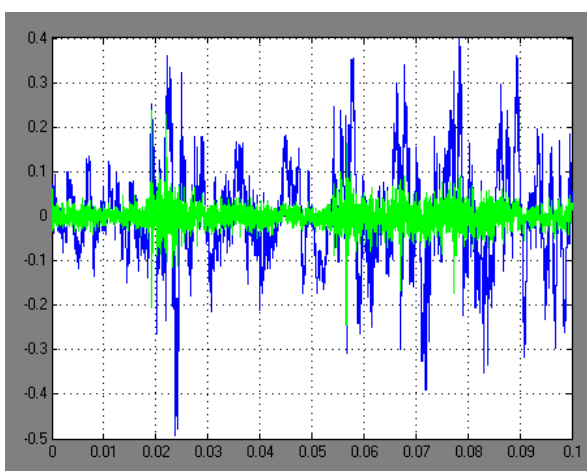Fig. 7. Input and output signal MUX{ $x(i)$, $\hat{x}(i)$ } (zoomed).

Fig. 8. Input and error signal MUX{ $x(i)$, $e(i)$ }.

On Fig.8 the input signal and the error signal are visualized together.

IV. CONCLUSION

An adaptive method for coding of one-dimensional digital signals, based on the linear prediction and Least Mean Square (LMS) weight coefficients adaptation of the prediction filter is presented. From the developed mathematical equations an algorithm and a general block scheme of adaptive linear prediction codec is synthesized and experimental results from the simulation by Simulink for Matlab 6.5 environment for test signals in WAV format are given

The developed ADPCM codec provides minimum processing error and lied to increase of PSNR with about 0.3 dB in comparison with 2 coefficients non-adaptive prediction codec.

The presented simulation model can be used in digital signal processing for spectral analysis, coding and transmission of one-dimensional signals and in distance learning by the using a Matlab Web Server.

The developed image codec is used in laboratory work on the disciplines: "Image and Signal Processing" and "Audio and Video Communication on Internet" and in the experimental work in laboratory "Electronic System for Visual Information" in Technical University of Sofia.

V. ACKNOWLEDGEMENT

VI. REFERENCES

[1] D. Stranneby, W. Walker, *Digital signal processing and applications*, Elsevier, 2004.
[2] S. Mitra, *Digital signal processing. A computer-based approach*, Mc Graw Hill, 2006.
[3] V. Madisetti, D. Williams, Eds. *The Digital Signal Processing Handbook*, CRC Press, 1998.
[4] A. D. Poularikas. *The Transforms and Applications Handbook*, Second Ed., CRC Press, 2000.
[5] G. Blanchet, M. Charbit, *Digital Signal and Image Processing Using MATLAB*, ISTE, 2006.
[6] B. Widrow, S. D. Stearns. *Adaptive Signal Processing*. Engelewood Cliffs, New York, Prentice-Hall, Inc., 1985.
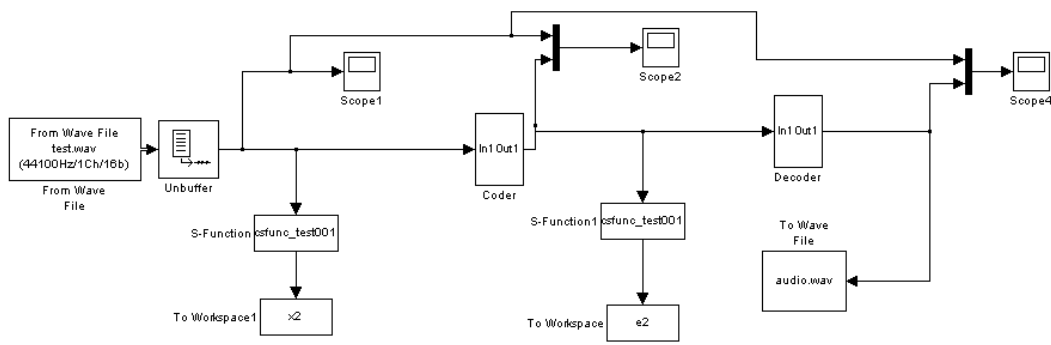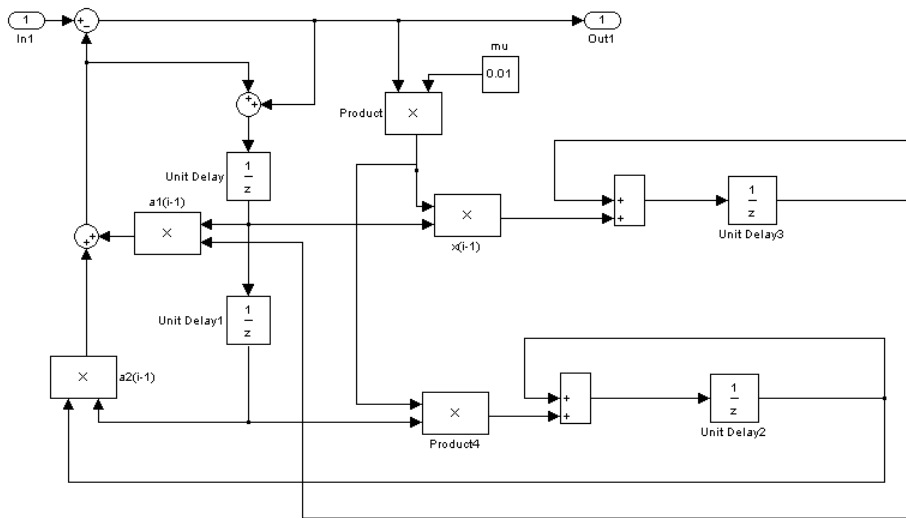
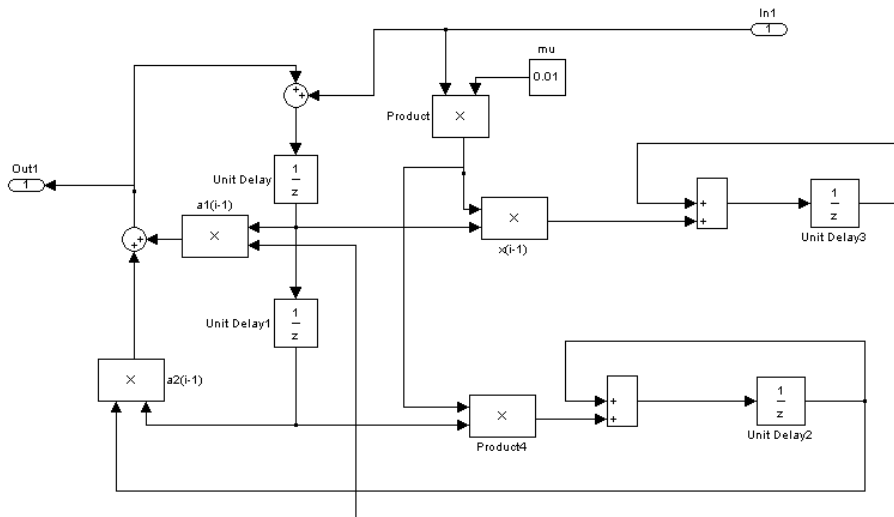Fig. 1. ADPCM Codec.

Fig. 2. ADPCM Coder.

Fig. 3. ADPCM Decoder.