World Applied Sciences Journal 7 (Special Issue of Computer & IT): 59-68, 2009
ISSN 1818-4952

Codebook Search in LD-CELP Speech Coding Algorithm Based on Multi-SOM Structure

[1]Mansour Sheikhan, [2]Vahid Tabataba Vakili and [1]Sahar Garoucy

[1]Department of Electrical Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran
Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran

Abstract: In the family of CELP coders, codebook search has high computational complexity. In this paper, the codebook search in low delay-code excited linear prediction (LD-CELP) G.728 coder is performed by a multi-self organizing map (SOM) neural model. A modified-supervised SOM training algorithm is also used in this work. In this algorithm, the codebook vectors are assigned to a class during training and a rejection term for codebook entries is used. The proposed neural search codebook module consists of 48 SOMs, which determine optimum index values of shape codebook. Empirical results show that the proposed model, which has an average classification rate of 97.8%, leads to 28% reduction in execution time as compared to a traditional implementation of G.728 encoder. However, the degradations in mean opinion score (MOS) and segmental signal to noise ratio (SNR_{seg}) are 0.16 and 0.17 dB, respectively.

Key words: Codebook search . LD-CELP speech encoder . self-organizing map

INTRODUCTION

In June 1988, International Telecommunication Union-Telecom sector (ITU-T, formerly CCITT) decided to investigate the possibility of standardizing a low delay 16 kbps speech coding for universal application. In this way, in 1989 Chen presented an algorithm based on low delay-code excited linear prediction (LD-CELP) [1]. The presented algorithm after some tests and enhancements [2, 3], became standardized by CCITT in 1992 as G.728 [4].

This coder has a one-way coding delay less than 2 msec, so it can be used in video-phone, cordless telephone, digital satellite system and applications like that in which having low delay is a critical parameter. LD-CELP is basically a backward-adaptive version of the CELP coder in which the predictor and excitation gain are updated backward adaptively by analyzing the former quantized speech and excitation, respectively. Many researches have been performed to improve LD-CELP speech coding algorithm [5-10].

On the other hand, artificial neural networks (ANNs) emerged in the recent decades as powerful and adaptive data processing models for pattern classification and feature extraction. Neural networks have been used extensively and successfully for a variety of applications in speech coding algorithms, as well. The researches on using ANNs in speech coding can be classified into two main domains: neural predictors which improve the quality of coder [11-19] and reduction the computational complexity [20-25].

In the family of CELP coders, codebook search has high computational complexity. ANNs can be used to reduce this complexity. For example, a modified Hopfield neural net is used to search in the codebook of a CELP coder [20]. Stochastic codebook (SCB) search for CELP coding is performed by the counter propagation neural network model and less computational complexity is achieved, too [21]. An efficient procedure for exploiting self organizing maps (SOMs) for a fast search quantization procedure is also presented in [22] that greatly reduce the complexity for vector quantization (VQ) of the spectral envelope. A codebook design algorithm, based on modified self-organizing feature map (SOFM) neural network, is introduced for LD-CELP in [24], as well. Huong *et al*. employed a new line spectral pairs (LSPs) codebook by using a centroid neural network (CNN) to enhance the compression rate of an adaptive multi-rate (AMR) coder [25].

LD-CELP is an analysis-by-synthesis (AbS) codebook driven method for linear predictive speech coding. The basic structure of the encoder is shown in Fig. 1. In this coder, which is an encoding method based on a source filter model, speech is reproduced by using excitation codevectors that are time-series signals and are stored in an excitation codebook, to drive a linear predictive synthesis filter that represents the spectral envelope of the input speech [4, 6].

Corresponding Author: Dr. Mansour Sheikhan, P.O. Box 11365/4435,
Post-Graduate Center, South Tehran Branch, Islamic Azad University, Iran

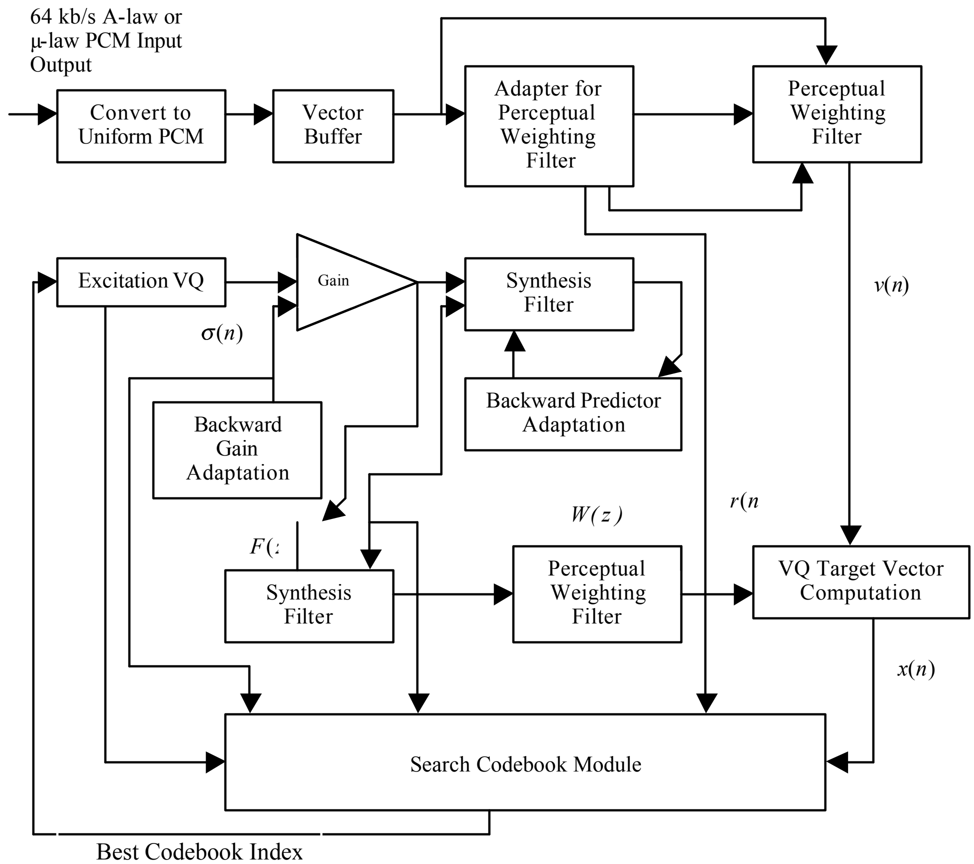

Fig. 1: Block diagram of LD-CELP encoder

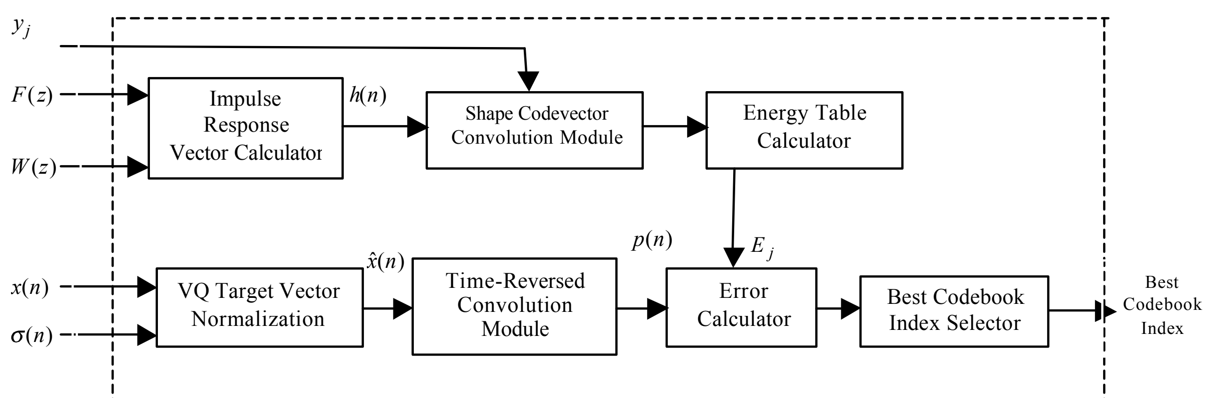

Fig. 2: Block diagram of search codebook module in LD-CELP [4]

The optimal excitation codevector is selected from the excitation codebook by using a closed-loop search according to the AbS method to find the one having the minimum perceptually-weighted waveform distortion of the synthetic speech signal to the related input speech signal.

In this study, codebook search is performed by a multi-SOM structure with modified-supervised training algorithm, which reduces the complexity of LD-CELP algorithm. This paper is organized as follows. In Section 2, the codebook search in LD-CELP encoder is described. Section 3 gives the details of modified-supervised training algorithm of SOM. The details of codebook search using multi-SOM are discussed in Section 4. The empirical results are reported in Section 5 and conclusions are drawn in Section 6.

CODEBOOK SEARCH IN LD-CELP

The LD-CELP algorithm with low rate and low complexity has very important meaning in the field of communication. On the other hand, the computations in codebook design increase rapidly with the growth of the codeword dimension k and the codebook size m. For the quantizer that needs higher dimension and higher size codebook, search is very time-consuming and difficult. In actual design of LD-CELP coder, the complexity must be reduced.

As shown in Fig. 1, in LD-CELP, zero-input response vector $r(n)$ subtracts from the VQ weighted speech vector $v(n)$ to obtain the VQ codebook search target vector $x(n)$. The excitation gain $s(n)$ is obtained with closed loop method, too. Each of the 1024 candidate codevectors is scaled by the current excitation gain and then is passed through a cascaded filter consisting of the synthesis filter $F(z)$ and the perceptual weighting filter $W(z)$ [4]. The transfer function of the cascaded filter is obtained as:

$$H(z) = W(z).F(z) \tag{1}$$

The detailed block diagram of LD-CELP search codebook module is shown in Fig. 2. The following formula is used for codebook search:

$$D = \sigma^2(n)\left\|\hat{x}(n) - g_i H(n) y_i\right\|^2 \tag{2}$$

in which, $s(n)$ is the predicted value of excitation gain and $\hat{x}(n)$ is the target vector adjusted by $s(n)$. $H(n)$ is the unit impulse response matrix of the short–term predictor, g_i is the i^{th} level in the 3-bit gain codebook and y_j is the j^{th} codevector in the 7-bit shape codebook [4].

According to G.728 recommendation, minimizing D is equivalent to maximizing D_{max}:

$$D_{max} = 2 g_i p^T(n) y_j - g_i^2 E_j \tag{3}$$

in which, $p(n) = H^T \hat{x}(n)$ and $E_j = \left\|Hy_j\right\|^2$.

The inner product term, $p_j = H^T \hat{x}(n) y_j$, which solely depends on j, takes the most of the computation in determining D_{max}. The number of multiplications and additions in computing E_j and p_j is reported in Table 1. Once the best indices for i and j are identified, they are concatenated to form the output of the codebook search module (a single 10-bit index). The 10-bit codebook index consists of two portions: 3 bits for gain codebook (b_0-b_2: 8 scalar values) and 7 bits for shape codebook (b_3-b_9: 128 codevectors).

Table 1: Number of multiplications and additions for computing E_j and p_j in search codebook module [10]

Equation	Number of multiplications	Number of additions
$E_j = \left\|Hy_j\right\|^2$	2560	2304
$p_j = H^T \hat{x}(n) y_j$	10240	9216

The values of gain are symmetric with respect to zero. The occurrence frequency characteristics of codevectors in shape codebook have not the uniform distribution [26]. The experiments show that the occurrence frequency characteristics of shape codevectors have not so significant dependency on the gain scalar values, as well. According to the results that are reported in [27], occurrence probability of shape codebook index values in the range of 65 to 128 is much higher than index values in the range of 1 to 64. In this way, Fig. 3 shows the histogram of the shape codebook index values for 11,000 sample speech frames of speech in this study.

MODIFIED SUPERVISED SOM ALGORITHM

The SOM, introduced by Kohonen [28], is a well known neural model and is popular in areas that require visualization and dimension reduction of large, high dimensional data sets. The SOM is a vector quantization method which can preserve the topological relationships between input vectors when projected to a lower dimensional display space.

The basic idea of SOM is simple. Every neuron i of the map is associated with an n-dimensional codebook vector $m_i = (m_{i1},\ldots,m_{in})^T$. The neurons of the map are connected to adjacent neurons by a neighborhood relation, which defines the topology or the structure of the map. Adjacent neurons belong to the neighborhood N_i of the neuron i. Neurons belonging to N_i are updated according to a neighborhood function $f(.)$. Most often, $f(.)$ is a Gaussian-bell function.

SOMs are typically trained in an unsupervised fashion, even if a teacher signal is available. Some attempts were made to utilize teacher signals, if available, with the goal to improve the mapping precision. Supervised training of SOMs is attempted by Kohonen in [28]. There, supervision is achieved by attaching information about class membership to the input vector while during the recognition phase, the class label is omitted.

However, the approach produces good results only for some artificial learning tasks where the number of classes is small [29]. A method for supervised training of SOM is described in this

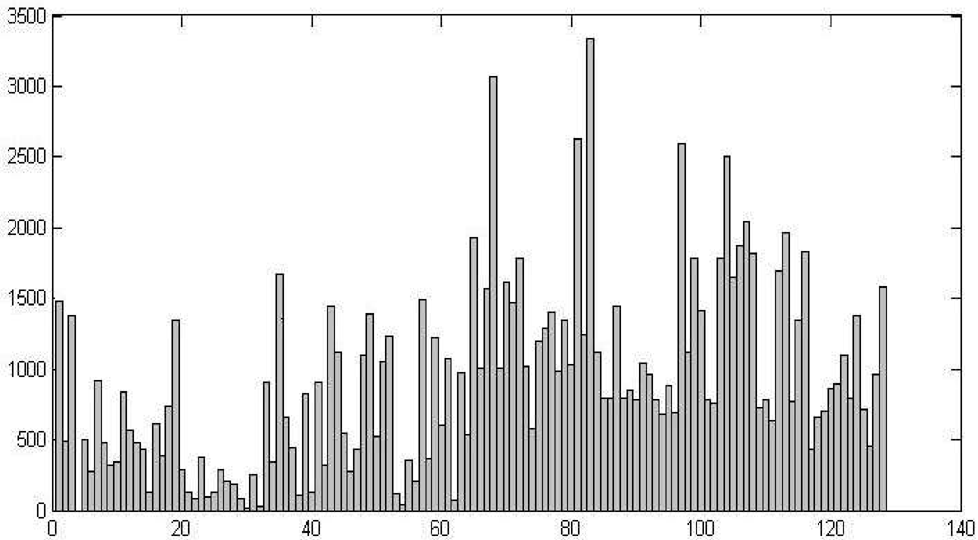

Fig. 3: Histogram of the shape codebook index values in LD-CELP for 11,000 sample speech frames

section, which has been developed with considerably more success.

In detail the method works as follows: Given a self organizing map M with k neurons. Each neuron is associated with a codebook entry $m \in R^n$. The best matching neuron m_r for an input x is obtained e.g., by using the Euclidean distance:

$$r = \arg\min_i \|(x - m_i)\Lambda\| \quad (4)$$

where Λ is a $n \times n$ dimensional diagonal matrix, which its diagonal elements $\lambda_{11} \ldots \lambda_{pp}$ are assigned to be μ_1, all remaining diagonal elements are set to μ_2. The values μ_1 and μ_2 weight the influence of the components l and c in x. Then the i^{th} element of the j^{th} codebook vector m_j is updated as follows:

$$\Delta m_{ij} = \begin{cases} -\beta\alpha(t) f(\Delta_{jr}) h(x_i, m_{ij}) & \text{; if x and } m_r \text{ are in different classes} \\ \alpha(t) f(\Delta_{jr})(x_i - m_{ij}) & \text{; else} \end{cases} \quad (5)$$

$f(\Delta_{jr})$ is a neighborhood function which will be explained later, α is the learning rate which decreases linearly to zero in time, β is a rejection rate which weights the influence of the rejection term $h(.)$. The purpose of the rejection term is to move m_r and its neighbors away from x. The effect is a reduction of the likelihood that an input node activates a codebook vector which is assigned to a foreign class in subsequent iterations. The rejection term is defined as follows:

$$h(x_i, m_{ij}) = \text{sgn}(x_i - m_{ij})(\rho_i - |x_i - m_{ij}|) \quad (6)$$

where sgn(.) is the signum function returning the sign of its argument and ρ_i is the standard deviation defined as follows:

$$\rho_i = \sqrt{\frac{\sum_{l=1}^{N}(x_{li} - \bar{x}_i)^2}{N}} \; ; \; \bar{x}_i = \frac{\sum_{l=1}^{N} x_{li}.\rho_i}{N} \quad (7)$$

where N is the number of nodes in the training set. ρ_i can be approximated by a constant when assuming that the mapping of nodes is random. This approximation significantly reduces the computational cost. The rejection term dictates stronger actions if a codebook entry is very similar to the input node. Note that $h(.)$ returns values within the range [-1; 1] eliminating the harmful influence of the magnitude of the vector elements.

The neighborhood function $f(.)$ controls the amount by which the weights of the neighboring neurons are updated. The neighborhood function $f(.)$ can take the form of a Gaussian function:

$$f(\Delta_{ir}) = \exp\left(-\frac{\|l_i - l_r\|^2}{2\sigma(t)^2}\right) \quad (8)$$

where $\sigma(t)$ is the spread decreasing with the number of iterations and l_r is the location of the winning neuron and l_i is the location of the i^{th} neuron in the lattice. Note that because of Eq. (5), the weights will not be distributed as a linear function of the input density. Training a supervised SOM network is an extension of the training algorithm

Table 2: Range of shape codebook index values in each part of multi-SOM structure

	Range of index values		
Part	First segment	Second segment	Third segment
1	$j\in[81,100]$	$j\in[65,80]$, $j\in[101,106]$	$j\in[107,128]$
2	$j\in[33,52]$	$j\in[17,26]$, $j\in[53,64]$	$j\in[1,16]$, $j\in[27,32]$

Table 3: Occurrence frequency of gain index values in 12,700 sample speech frames

Index value (i)	Occurrence frequency
1	3502
2	1802
3	515
4	121
5	3793
6	2375
7	512
8	80

Table 4: Index specifications of SOMs in the first part of proposed model

SOM	Gain codebook index (i)	Shape codebook index (j)
SOM_1	5	$(81\le j\le 100)$
SOM_2	5	$(65\le j\le 80)$ & $(101\le j\le 106)$
SOM_3	5	$(107\le j\le 128)$
SOM_4	1	$(81\le j\le 100)$
SOM_5	1	$(65\le j\le 80)$ & $(101\le j\le 106)$
SOM_6	1	$(107\le j\le 128)$
SOM_7	6	$(81\le j\le 100)$
SOM_8	6	$(65\le j\le 80)$ & $(101\le j\le 106)$
SOM_9	6	$(107\le j\le 128)$
SOM_{10}	2	$(81\le j\le 100)$
SOM_{11}	2	$(65\le j\le 80)$ & $(101\le j\le 106)$
SOM_{12}	2	$(107\le j\le 128)$
SOM_{13}	3	$(81\le j\le 100)$
SOM_{14}	3	$(65\le j\le 80)$ & $(101\le j\le 106)$
SOM_{15}	3	$(107\le j\le 128)$
SOM_{16}	7	$(81\le j\le 100)$
SOM_{17}	7	$(65\le j\le 80)$ & $(101\le j\le 106)$
SOM_{18}	7	$(107\le j\le 128)$
SOM_{19}	4	$(81\le j\le 100)$
SOM_{20}	4	$(65\le j\le 80)$ & $(101\le j\le 106)$
SOM_{21}	4	$(107\le j\le 128)$
SOM_{22}	8	$(81\le j\le 100)$
SOM_{23}	8	$(65\le j\le 80)$ & $(101\le j\le 106)$
SOM_{24}	8	$(107\le j\le 128)$

used by an unsupervised SOM network. The difference between the two approaches is that codebook vectors are assigned to a class during training and also a rejection term is used for codebook entries that do not belong to the same class as the input vector.

The weight values μ_1 and μ_2 can be computed as they depend on the dimension and magnitude of the elements in l and c. Given that the Euclidean distance in Eq. (4) is computed as follows:

$$d = \sqrt{\mu_1 \sum_{i=1}^{p} (l_i - m_i)^2 + \mu_2 \sum_{j=1}^{p} (c_j - m_{n+j})^2} \tag{9}$$

Thus, μ_1 and μ_2 balance the influence of l and c to the distance measure. Ideally, the influence of the data label and the coordinate vector on the final result is equal. A way to obtain the pair of weight values is suggested in [30]:

$$\frac{\mu_1}{\mu_2} = \frac{n}{m} \frac{\sum_{j=1}^{m} (\phi(|l_j|) - \sigma(|l_j|))}{\sum_{i=1}^{n} (\phi(c_i) - \sigma(c_i))} \tag{10}$$

where $\phi(|l_i|)$ is the average absolute value of the i^{th} element of all data labels in the data set. Similarly, $\phi(c_i)$ is the average of the i^{th} element of all coordinates. To obtain unique value pairs we make the assumption that $\mu_1 + \mu_2 = 1$.

CODEBOOK SEARCH BASED ON MULTI-SOM STRUCTURE

In this paper, a multi-SOM structure is used instead of search codebook module. This structure consists of two parts and 24 SOMs are used in each part. The first part determines optimum index values of shape codebook in the range of 65 to 128. The second part has the same function for indices in the range of 1 to 64 (Fig. 4).

In our experiments, the range of shape codebook index values in each part is segmented based on the occurrence frequency characteristics of codevectors (Table 2). The occurrence frequency of the gain codebook indices has not uniform distribution, too (Table 3).

Training dataset for each of the SOMs is formed based on the index values of gain codebook (i) and shape codebook (j), listed in Table 4 and 5. The proposed arrangement for indices is achieved by several experiments to have acceptable levels of the segmental signal to noise ratio (SNR_{seg}) and mean opinion score (MOS).

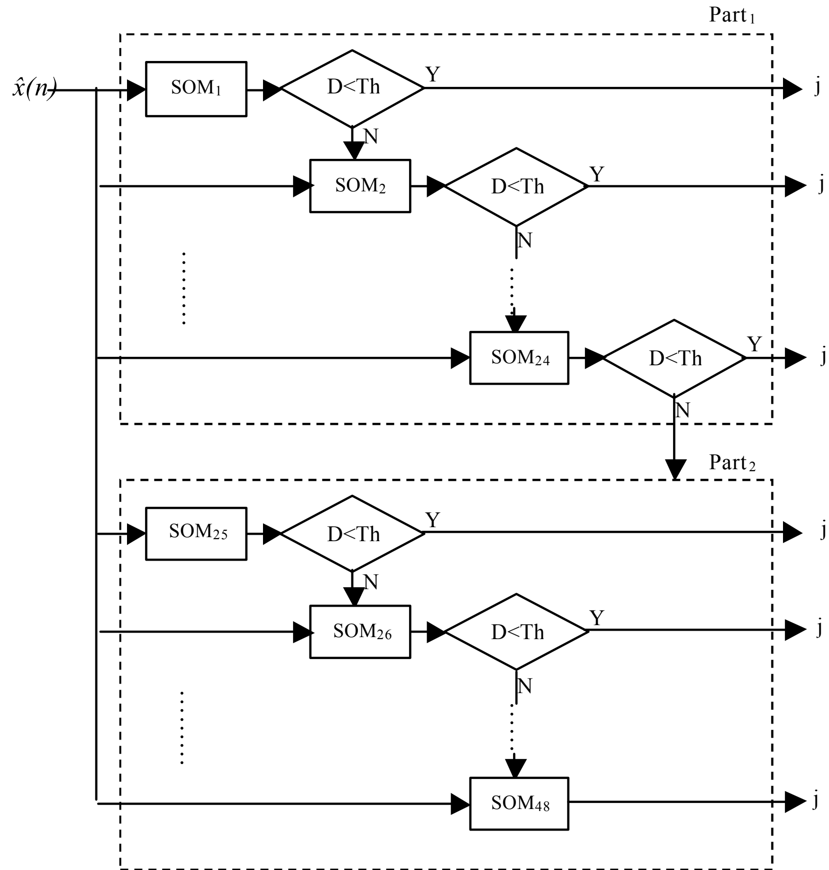

Fig. 4: The proposed multi-SOM model for search codebook module in LD-CELP

Table 5: Index specifications of SOMs in the second part of proposed model

SOM	Gain codebook index (i)	Shape codebook index (j)
SOM_{25}	5	(33≤j≤52)
SOM_{26}	5	(17≤j≤26) & (53≤j≤64)
SOM_{27}	5	(1≤j≤16) & (27≤j≤32)
SOM_{28}	1	(33≤j≤52)
SOM_{29}	1	(17≤j≤26) & (53≤j≤64)
SOM_{30}	1	(1≤j≤16) & (27≤j≤32)
SOM_{31}	6	(33≤j≤52)
SOM_{32}	6	(17≤j≤26) & (53≤j≤64)
SOM_{33}	6	(1≤j≤16) & (27≤j≤32)
SOM_{34}	2	(33≤j≤52)
SOM_{35}	2	(17≤j≤26) & (53≤j≤64)
SOM_{36}	2	(1≤j≤16) & (27≤j≤32)
SOM_{37}	3	(33≤j≤52)
SOM_{38}	3	(17≤j≤26) & (53≤j≤64)
SOM_{39}	3	(1≤j≤16) & (27≤j≤32)
SOM_{40}	7	(33≤j≤52)
SOM_{41}	7	(17≤j≤26) & (53≤j≤64)
SOM_{42}	7	(1≤j≤16) & (27≤j≤32)
SOM_{43}	4	(33≤j≤52)
SOM_{44}	4	(17≤j≤26) & (53≤j≤64)
SOM_{45}	4	(1≤j≤16) & (27≤j≤32)
SOM_{46}	8	(33≤j≤52)
SOM_{47}	8	(17≤j≤26) & (53≤j≤64)
SOM_{48}	8	(1≤j≤16) & (27≤j≤32)

It is noted that MOS provides a numerical indication of the perceived quality of received media after compression and/or transmission. The MOS is expressed as a single number in the range of 1 to 5, where 1 is the lowest perceived audio quality and 5 is the highest perceived audio quality measurement.

SNR_{seg} is an important factor in determining the quality of audio data, too. This is particularly important in speech recognition technology, since it is well known that recognition performance is strongly influenced by the SNR [31]:

$$SNR = 10\log\left(\frac{\sum_n x^2(n)}{\sum_n (x^2(n) - y^2(n))^2}\right) \tag{11}$$

where $x(n)$ is the input signal to encoder and $y(n)$ is the output signal from decoder. SNR_{seg} is defined as the average of SNR measurements:

$$SNR_{seg} = \frac{1}{N_f}\sum_{m=1}^{N} SNR_m \tag{12}$$

in which, N_f is the number of frames.

Table 6: Map size, training and test specifications of SOMs in the first part of model

SOM	Number of training samples	Number of test samples	Map size	Training time (sec)	
				T_1	T_2
SOM_1	4400	400	12×20	3.401	5.987
SOM_2	4400	400	12×20	3.764	5.456
SOM_3	4400	400	12×20	3.897	5.453
SOM_4	4400	400	12×20	3.376	5.567
SOM_5	4400	400	12×20	3.399	5.222
SOM_6	4400	400	12×20	3.123	5.569
SOM_7	4400	400	12×20	3.434	5.454
SOM_8	4400	400	12×20	3.608	5.861
SOM_9	4400	400	12×20	3.564	5.302
SOM_{10}	4400	400	12×20	3.666	5.875
SOM_{11}	4400	400	12×20	3.376	5.110
SOM_{12}	4400	400	12×20	3.354	5.132
SOM_{13}	4400	400	10×20	2.567	4.013
SOM_{14}	4400	400	10×14	2.235	4.077
SOM_{15}	4400	400	10×14	2.345	4.076
SOM_{16}	4400	400	10×14	2.567	4.179
SOM_{17}	2200	200	10×14	2.567	4.106
SOM_{18}	2200	200	10×14	2.233	4.003
SOM_{19}	1000	100	9×10	1.999	4.100
SOM_{20}	1000	100	9×10	2.001	3.986
SOM_{21}	1000	100	9×10	2.010	3.924
SOM_{22}	1000	100	9×10	1.997	3.634
SOM_{23}	1000	100	9×10	1.921	3.234
SOM_{24}	1000	100	9×10	1.987	3.244

Table 7: Map size, training and test specifications of SOMs in the second part of model

SOM	Number of training samples	Number of test samples	Map size	Training time (sec)	
				T_1	T_2
SOM_{25}	1100	100	12×20	3.321	5.384
SOM_{26}	1100	100	12×20	3.363	5.453
SOM_{27}	1100	100	12×20	3. 164	5.333
SOM_{28}	1100	100	12×20	3.273	5.563
SOM_{29}	1100	100	12×20	3.311	5.109
SOM_{30}	1100	100	12×20	3.144	5.587
SOM_{31}	1100	100	12×20	3.431	5.234
SOM_{32}	1100	100	12×20	3.345	5.654
SOM_{33}	1100	100	12×20	3.552	5.352
SOM_{34}	1100	100	12×20	3.361	5.055
SOM_{35}	1100	100	12×20	3.386	5.188
SOM_{36}	1100	100	12×20	3.308	5.431
SOM_{37}	1100	100	10×20	2.201	4.000
SOM_{38}	1100	100	10×14	2.001	4.001
SOM_{39}	1100	100	10×14	2.145	4.051
SOM_{40}	1100	100	10×14	2.137	4.017
SOM_{41}	600	80	10×14	2.117	4.106
SOM_{42}	600	80	10×14	2.123	4.009
SOM_{43}	600	80	9×10	1.799	4.001
SOM_{44}	600	80	9×10	1.900	3.701
SOM_{45}	600	80	9×10	1.986	3.333
SOM_{46}	600	80	9×10	1.891	3.330
SOM_{47}	600	80	9×10	1.900	3.209
SOM_{48}	600	80	9×10	1.960	3.204

Table 8: Number of SOMs' selection in the multi-SOM model-2700 input frames

SOM	Number of selection	SOM	Number of selection
SOM_1	251	SOM_{25}	64
SOM_2	210	SOM_{26}	61
SOM_3	171	SOM_{27}	49
SOM_4	160	SOM_{28}	41
SOM_5	148	SOM_{29}	38
SOM_6	141	SOM_{30}	39
SOM_7	138	SOM_{31}	29
SOM_8	130	SOM_{32}	21
SOM_9	131	SOM_{33}	23
SOM_{10}	120	SOM_{34}	25
SOM_{11}	117	SOM_{35}	19
SOM_{12}	116	SOM_{36}	17
SOM_{13}	80	SOM_{37}	14
SOM_{14}	61	SOM_{38}	16
SOM_{15}	56	SOM_{39}	15
SOM_{16}	58	SOM_{40}	13
SOM_{17}	49	SOM_{41}	10
SOM_{18}	41	SOM_{42}	12
SOM_{19}	4	SOM_{43}	1
SOM_{20}	2	SOM_{44}	5
SOM_{21}	1	SOM_{45}	3
SOM_{22}	1	SOM_{46}	0
SOM_{23}	0	SOM_{47}	2
SOM_{24}	0	SOM_{48}	0

Based on the index values of i and j, which are determined in the test phase of model, Eq. (2) is used for calculating the value of D. If the value of D is lower than a threshold, which is explained in the next section, the best index is found for j. Otherwise, the next SOM is employed to find the best index.

EMPIRICAL RESULTS

In this work, a 16 kbps LD-CELP vocoder is implemented based on the ITU-T G.728 recommendation [4]. The simulation of encoder and decoder is performed by MATLAB v.7.2 software. Training dataset includes 103,200 frames of speech from 25 male and 30 female speakers. The sampling frequency is 8 kHz and the frame size is 20 samples.

We used a learning rate $\alpha(0)=0.5$, a rejection rate $\beta(0)=0.05$, a neighborhood spread $\sigma=40$, $\mu_1=0.43$ and $\mu_2=0.57$ in simulations. The fine tuning of SOMs is also performed, in which learning rate is set 0.05. The specifications of SOMs in the first and second parts of proposed model in terms of "number of training and test samples", "map size", "training time in self-organization phase (T_1)" and "training time in convergence or fine tuning phase (T_2)" are reported in Table 6 and 7, respectively.

Table 9: AQE and ATE for different map sizes

Quality index	Map size 9×10	10×14	12×20
AQE	0.159	0.099	0.073
ATE	0.122	0.141	0.137

Table 10: Performance comparison of proposed model with a traditional G.728 implementation

System	Execution time (sec)	SNR_{seg} (dB)	MOS
Traditional G.728 [3, 35]	4.04	18.45	3.91
Proposed model	2.91	18.28	3.75

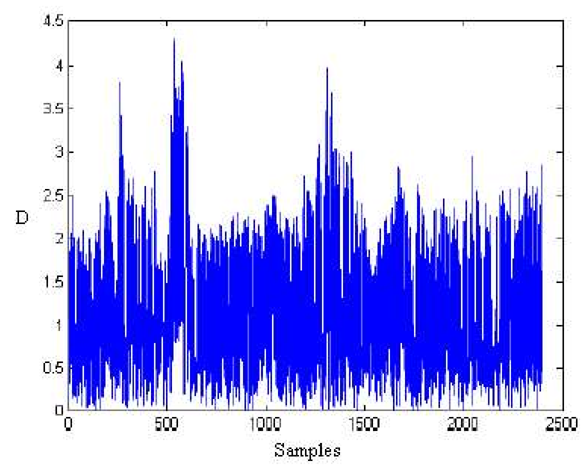

Fig. 5: Values of D for 2400 frames of speech

The optimum value of threshold is selected so as to maximize the SNR. The similar problem was studied by Max [32] and later by Paez and Glisson [33]. This iterative procedure is used and an optimum value of 2.83 is achieved for 2400 frames of speech. The value of D for these frames is depicted in Fig. 5.

To show the effectiveness of the proposed model, which has the average classification rate of 97.8% over 9640 test samples, 2700 frames of speech are selected and applied to the trained model. The number of SOMs' selection is reported in Table 8.

In general in constructing SOM, two quality indices are considered, i.e. average quantization error (AQE) and average topographic error (ATE). AQE is the average distance between each input vector and its best matching unit (BMU) and is used to measure map resolution. ATE represents the accuracy of the map in preserving topology and the error value is calculated from the proportion of all data vectors for which first and second BMUs are not adjacent for measuring topology preservation [34]. These two indices serve as a criterion in our research to choose a suitable map. The values of AQE and ATE for different map sizes of SOM are reported in Table 9.

CONCLUSIONS

In this paper, codebook search module in the structure of LD-CELP encoder was replaced by a multi-SOM neural model. Although, SOM is typically trained in an unsupervised fashion, in this study a modified-supervised SOM model was used. In this way, the codebook vectors were assigned to a class during training and a rejection term for codebook entries, that did not belong to the same class as the input vector, was used.

The proposed model had an average classification rate of 97.8% and led to 28% reduction in execution time as compared to a traditional implementation of ITU-T G.728 encoder (Table 10). However, MOS and SNR_{seg} were reduced 0.16 and 0.17 dB, respectively. Therefore, the proposed model led to noticeable reduction in computational complexity, without significant degradation in MOS and SNR_{seg}.

REFERENCES

1. Chen, J.H., 1989. A Robust Low-Delay CELP Speech Coder at 16 kbit/s. In the Proceedings of the IEEE GLOBCOM, 2: 1237-1241.
2. Chen, J.H. and R.V. Cox, 1990. LD-CELP: A High Quality 16 kbit/s Speech Coder with Low Delay. In the Proceedings of the IEEE GLOBCOM, 1: 528-532.
3. Chen, J.H., R.V. Cox, Y.C. Lin and N. Jayant, 1992. A Low Delay CELP Coder for CCITT 16 kb/s Speech Coding Standard. IEEE Journal on Selected Areas on Commmunication, 10: 830-847.
4. ITU-T G. 728 Recommendation, 1992. Coding of Speech at 16 kb/s Using Low-Delay Code Excited Linear Prediction.
5. Chen, J.H. and R.V. Cox, 1993. The Creation and Evolution of LD-CELP: From Concept to Standard. Speech Communication, 12: 103-112.
6. Zahang, G., K.M. Xie and L.Y. Huangfu, 2003. Optimization Gain Codebook of LD-CELP. In the Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2: 149-152.
7. Zahang, G., K.M. Xie, X.Y. Zhang and L.Y. Huangfu, 2003. Improvement on G.728 by Normalized Shape Codebook with Exact Gain. Journal of China Institute of Communications, 24: 87-92.
8. Zahang, G., K.M. Xie and L.Y. Huangfu, 2004. Improving G.728's Hybrid Window and Excitation Gain. In the Proceeding of the Asia-Pacific IEEE Conference on Circuits and System, 1: 185-188.
9. Akbari, M., B. Vosoughi Vahdat and K. Nayebi, 2005. Real-Time Implementation and Optimization of ITU's G.728 on TMS320C64X DSP. In the Proceeding of the IEEE International Symposium on Signal Processing and Information Technology, pp: 896-900.
10. Xueying, Z., Z.Q. Qun and M.Z. Yanys, 2008. Reducing the Complexity of LD-CELP Speech Coding Algorithm Using Direct Vector Quantization. In the Proceedings of the International Conference on Communication, Circuits and Systems, 10: 811-815.
11. Thyssen, J., H. Nielsen and S.D. Hansen, 1994. Nonlinear Short-Term Prediction in Speech Coding. In the Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 1: 185-188.
12. Le, T.T. and J.S. Mason, 1994. Nonlinear Predictor for Speech Enhancement. In the Proceedings of the IEE Colloquium on Techniques for Speech Processing and their Applications, 11: 4-11.
13. Birgmeier, M., 1996. Nonlinear Prediction of Speech Signals Using Radial Basis Function Networks. In the Proceedings of the European Signal Processing Conference, 1: 459-462.
14. Kumar, A. and A. Gersho, 1997. LD-CELP Speech Coding with Nonlinear Prediction. IEEE Signal Processing Letters, 4: 89-91.
15. Ma, N. and G. Wei, 1998. Speech Coding with Nonlinear Local Prediction Model. In the Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2: 1101-1104.
16. Faundez, M., 1999. Adaptive Hybrid Speech Coding with a MLP/LPC Structure. In the Proceedings of the International Work-Conference on Artificial and Natural Neural Networks, 11: 814-823.
17. Sassi, S.B., R. Braham and A. Belghith, 2001. Neural Speech Synthesis System for Arabic Language Using CELP Algorithm. In the Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications, pp: 119-121.

18. Faúndez-Zanuy, M., S. McLaughlin, A. Esposito, A. Hussain, J. Schoentgen, G Kubin, W.B. Kleijn and P. Maragos, 2002. Nonlinear Speech Processing: Overview and Applications. Control and Intelligent Systems, 30: 1-10.
19. Faúndez-Zanuy, M., 2003. Nonlinear Speech Coding with MLP, RBF and Elman Based Prediction. Lecture Notes in Computer Science, 2687: 671-678.
20. Easton, M.G. and C.C. Goodyear, 1991. A CELP Codebook and Search Technique Using a Hopfield Net. In the Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pp: 685-688.
21. Indrayanto, A., A. Langi and W. Kinsner, 1991. A Neural Network Mapper for Stochastic Codebook Parameter Encoding in Code Excited Linear Predictive Speech Processing. In the Proceedings of the IEEE Western Canada Conference on Computer, Power and Communications Systems, pp: 221-224.
22. Hernandez-Gomez, L.A. and E. Lopez-Gonzalo, 1993. Phonetically-Driven CELP Coding Using Self-Organizing Maps. In the Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2: 628-631.
23. Wu, L., M. Nranjan and F. Fallside, 1994. Fully Vector-Quantized Neural Network-Based Code Excited Nonlinear Predictive Speech Coding. IEEE Transactions on Speech and Audio Processing, 2: 482-489.
24. Wu, S., G. Zhang, X. Zhang and Q. Zhao, 2008. A LD-CELP Speech Coding Algorithm Based on Modified SOFM Vector Quantizer. In the Proceedings of the International Symposium on Intelligent Information Technology Applications, pp: 408-411.
25. Huong, V., B.J. Min, D.C. Park and D.M. Woo, 2008. A New Vocoder Based on AMR 7.4 kbit/s Mode in Speaker Dependent Coding System. In the Proceedings of the ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp: 163-167.
26. Kawano, N., H. Yajima, A. Hotta and Y. Naito, 1995. A Variable Bit-Rate LD-CELP Speech Coder at 16, 12.8 and 9.6 kbit/s. In the Proceedings of the IEEE Workshop on Speech Coding for Telecommunications, pp: 95-96.
27. CCITT, 1992. A Statistical Characteristic of 16 kbit/s LD-CELP Shape Codebook Used in the Draft Recommenation G.728.COM-XV-175-E.
28. Kohonen, T., 1995. Self-Organizing Maps. Springer Series in Information Sciences.
29. Hagenbuchner, M., A. Tsoi and A. Sperduti, 2001. A Supervised Self-Organizing Map for Structured Data. In Advances in Self-Organizing Maps, pp: 21-28.
30. Hagenbuchner, M., A. Sperduti and A. Tsoi, 2003. A Self-Organizing Map for Adaptive Processing of Structured Data. IEEE Transactions on Neural Networks, 14: 491-505.
31. Deller, J.R., J.H.L. Hansen and J.G. Proakis, 2000. Discrete-Time Processing of Speech Signals. IEEE Press.
32. Max, J., 1960. Quantizing for Minimum Distortion. IRE Transactions on Information Theory, 6: 7-12.
33. Paez, M.D. and T.H. Glisson, 1972. Minimum Mean Squared-Error Quantization in Speech. IEEE Transactions on Communication, 20: 225-230.
34. Uriarte, E.A. and F.D. Martin, 2006. Topology Preservation in SOM. World Academy of Sciences, Engineering and Technology, 21: 52-55.
35. Zahang, G., K.M. Xie, X.Y. Zhang and L.Y. Huangfu, 2006. How to Optimize the Gain Filter of LD-CELP. In the Proceedings of the Information and Communication Technologies Conference, 1: 1206-1211.