

# Design and Comparison of Vector Quantization Codebooks for Narrowband Speech Coding

Hiba Faraj\*, Selma Ozaydin

Department of Electronic and Communication Engineering, Cankaya University, Turkey

Copyright©2019 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

**Abstract** Vector quantization codebook algorithms are used for coding of narrow band speech signals. Multi-stage vector quantization and split vector quantization methods are two important techniques used for coding of narrowband speech signals and these methods are very popular due to the high bit rate minimization during coding of the signals. This paper presents performance measurements of multistage vector quantization and split vector quantization methods. We used line spectral frequencies for coding of the speech signals in codebook tables so as to ensure filter stability after quantization. The codebooks were generated by using the Linde-Buzo-Gray (LBG) algorithm. The tests were performed by selecting large amount of input data in training and test stages and to evaluate noise robustness of the methods, both noisy and clean speech signals were used. As a result, different codebooks were designed and tested in many stages and different bit rates to measure quantization performance. It is measured in terms of spectral distortion evaluation. We obtained the best result in 24bit multistage vector quantization codebook with a spectral distortion less than 1 dB for clean speech training data input. When we compared multistage and split vector quantization codebook spectral distortion results, multistage codebooks gave better performance in each option.

**Keywords** Multistage Vector Quantization, Split Vector Quantization, Linear Predictive Coding, Line Spectral Frequencies, Codebook Design

---

## 1. Introduction

Coding algorithms minimize the bit rate of an input signal without any harmful loss in the signal quality. Digital coding of speech signals in narrowband coding is the process of obtaining a compressed representation of the input signals with a perceptual quality and for efficient transmission through a channel. While an uncompressed

speech signal transmission occurs about 64kbps, an encoder algorithm can reduce the bit rate of an input signal below 64kbps. In a low bit rate speech coding algorithm, the bit rate can be reduced as low as below 1kbps, i.e. a reduction in bit rate by 64 times. Linear predictive coding (LPC) is a major part of the speech compression algorithm. Linear Predictive Coding is one possible technique of analyzing and synthesizing human speech. LPC makes coding at low bit rates possible. For LPC-10, the bit rate is about 2.4 kbps. Even though this method results in an artificial sounding speech, it is intelligible. This method has found extensive use in military applications, where a high quality speech is not as important as a low bit rate to allow for heavy encryptions of secret data. However, since a high quality sounding speech is required in the commercial market, engineers are faced with using other techniques that normally use higher bit rates and result in higher quality output. Vector quantization (VQ) of spectral parameters has an essential role in reduction of bit rate for linear predictive (LP) based speech coders. Efficient quantization of linear predictive coefficient (LPC) parameters in very low bit rates directly effects the bit rate reduction of a speech coder. Sequential VQ applications have a limited performance due to its computational complexity and large codebooks. Two popular methods for the solution of computational complexity in sequential VQ are multistage vector quantization (MSVQ) and split vector quantization (SVQ) methods. They are very commonly used to encode narrowband speech signals. The parameters used for coding the speech signals are usually the line spectral frequencies (LSF) to ensure the stability after quantization. There are some studies in the literature comparing MSVQ and SVQ methods and in these studies MSVQ shows better SD performance. However, these studies in the literature only compare the performances of MSVQ and SVQ codebooks for clean input speech data. In this study, we performed the tests for both clean and noisy speech test data to evaluate their robustness to noisy conditions. Original contribution of this study lies in the evaluation both clean noisy input data and also comparison of the codebook performances according to changing

stages in each bit rate. In this framework, we designed many MSVQ and SVQ codebooks and we compared their performances by measuring the SD results.

## 2. LBG Codebook Design Algorithm

The goal in designing an optimal vector quantizer is to obtain N-element reproduction vectors representing any input vector with minimum spectral distortion. The codebook is then designed and trained by the training input vectors. In training stage of vector quantization, speech signals are used for generation of codebooks. During training stage, LSF vectors are obtained from input speech signals. There is a well-known iterative algorithm, namely LBG algorithm [1], The LBG VQ design technique is an iterative K-means clustering algorithm. In LBG algorithm, L training vectors are clustered into M codebook vectors.

The LBG algorithm tends to require the initial codebook. At the beginning of the algorithm, there is an initial code vector which is set as an average for the entire training sequences. Then, the code vector is then split into two. Afterwards, these two coded vectors are split into four. The splitting process continues until the required amount of code vectors is attained. In encoding the real process, there is an input vector which searches for the best matching codebook from the obtained codebook. The function of the decoder is to work like a table look-up which receives the transmitted index and look at the codebook for the vector which corresponds to and then matched vector from the codebook is then used to represent the input vector.

The input to the LBG algorithm is a training speech sequence obtained from people of different groups and of different ages. The speech signals used in training stage must be free of background noise. The algorithm is formally implemented in Figure 1 and by the following recursive procedure:

**Step 1:** Design a 1-vector codebook, set  $m=1$ , Calculate centroid

$$C_i = \frac{1}{T} \sum_{j=1}^T X_j \quad (1)$$

$$C_{2i-1} = C_i X (1 + \delta) \quad (2)$$

Where  $T$  is the total number of data vectors.

**Step 2:** Double the size of the codebook by splitting (divide each centroid  $C_i$  into two close vectors)

$$C_{2i} = C_i X (1 - \delta) \quad (3)$$

$1 \leq i \leq m$ , here  $\delta$  is a small fixed perturbation scalar (let  $m = 2m$ ) Set  $n = 0$  here  $n$  is iterative time

**Step 3:** Nearest-Neighbor Search

Find the nearest neighbor to each data vector. Put  $X_j$  in the partitioned set  $P_i$  if  $C_i$  is the nearest neighbor to  $X_j$

**Step 4:** Find Average Distortion.

After obtaining the partitioned sets  $P = (P_i, 1 \leq i \leq m)$  set

$n = n + 1$ , Calculate the overall average distortion

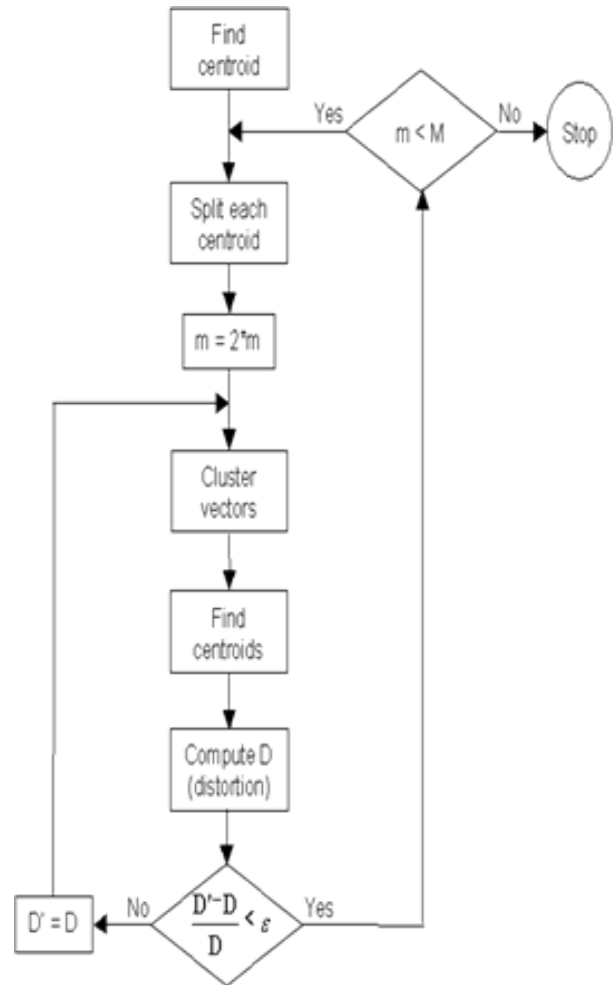


Figure1. Flow diagram of the LBG algorithm

$$D_n = \frac{1}{T} \sum_{i=1}^m \sum_{j=1}^{T_i} (D_j^{(i)}, C_i) \quad (4)$$

Where  $P_i = \{X_1^{(i)}, X_2^{(i)}, \dots, X_{T_i}^{(i)}\}$

**Step 5:** Centroid Update.

Find centroids of all disjoint partitioned sets  $P_i$  by

$$C_i = \frac{1}{T} \sum_{j=1}^{T_i} X_j^{(i)} \quad (5)$$

**Step 6:** Iteration 1.

If  $(D_{n-1} - D_n) / D_n > \delta$  (6) go to step 3 otherwise go to step 7

**Step 7:** Iteration 2.

If  $m = N$  then take the codebook  $C_i$  as the final codebook; otherwise, go to step 2.  $N$  is the codebook size.

## 3. Split Vector Quantization

Split vector quantization aims to reduce the complexity of searching and storage by splitting the input vectors in subvectors. In SVQ, the input LSF vectors are usually split into two or three parts depending on the type of vectors in

consideration, and then they are separately quantized. Then, the quantization of the subvectors in the codebooks is performed in each stage. Training the codebooks for SVQ is straightforward because each codebooks in the stages can be treated like a separate conventional vector quantizer, and can thus be trained separately using the K-mean algorithm.

When designing SVQ codebooks in our algorithm, in order to find the optimal partitioning for the LSF vectors by dividing the vectors into sub stages. There are many techniques when designing the codebooks in the MSVQ. The simplest method is to train the codebooks sequentially. Here, the codebook for the first stage is computed by using K-mean algorithm and the training data is quantized with the obtained one-stage vector quantizer. The resulting quantization error vectors are used for the second stage. This is repeated for all stages, with each new codebook trained using the error between the original and the reconstructed vectors including all the previous stages. In a 3-stage vector quantization, the LPC parameter vector (in some suitable representation such as the LSF

representation) is quantized by the first-stage vector quantizer and the error vector  $\mathbf{e}$  (which is the difference between the input and output vectors of the first stage) is quantized by the second-stage vector quantizer [6]. The final quantized version of the LPC vector is obtained by summing the outputs of the three stages.

To minimize the complexity of the 3-stage vector quantizer, selection of a proper distortion measure is the most important issue in the design and operation of a vector quantizer. Since the spectral distortion is used here for evaluating LPC quantization performance, ideally it should be used to design the vector quantizer. However, it is very difficult to design a vector quantizer using this distortion measure. Therefore, simpler distance measures (such as the Euclidean and the weighted Euclidean distance measures) between the original and quantized LPC parameter vectors.

The intra-frame correlation was calculated over the whole training database as a correlation between  $LSF_i$  and  $LSF_j$  in the same frame,  $i, j = 1, 2, \dots, 10$ . The intra-frame correlation coefficients are given in Table 1. The correlation between successive LSFs is significant [4].

**Table 1.** The intra-frame correlation coefficients

		I									
J	1	2	3	4	5	6	7	8	9	10	
1	1.0000	0.3149	0.3069	0.2388	0.2148	0.2084	0.1630	0.1815	0.1576	0.0808	
2	0.3149	1.0000	0.4968	0.4349	0.3882	0.3687	0.3091	0.2959	0.2405	0.1648	
3	0.3069	0.4968	1.0000	0.5822	0.5301	0.4928	0.4168	0.3952	0.3269	0.2013	
4	0.2388	0.4349	0.5822	1.0000	0.5609	0.5136	0.4315	0.4002	0.3247	0.2015	
5	0.2148	0.3882	0.5301	0.5609	1.0000	0.5470	0.4472	0.4090	0.3218	0.2096	
6	0.2084	0.3687	0.4928	0.5136	0.5470	1.0000	0.5096	0.4336	0.3444	0.2118	
7	0.1630	0.3091	0.4168	0.4315	0.4472	0.5096	1.0000	0.4610	0.3670	0.2243	
8	0.1815	0.2959	0.3952	0.4002	0.4090	0.4336	0.4610	1.0000	0.3938	0.2472	
9	0.1576	0.2405	0.3269	0.3247	0.3218	0.3444	0.3670	0.3938	1.0000	0.2975	
10	0.0808	0.1648	.2013	0.2015	0.2096	0.2118	0.2243	0.2472	0.2975	1.0000	

### 4. Multistage Vector Quantization

The MSVQ reduces the complexity in a vector quantizer (in some suitable representation such as the LSF representation) are used to design the LPC vector quantizer.

To find the best LPC parametric representation for the Euclidean distance measure, the study of the 3-stage vector quantizer with the distance measure in the following three domains: the LSF domain, the arcsine reflection coefficient domain and the log-area ratio domain is done [7]. The 3-stage vector quantizer performs better with the LSF representation than with the other two representations. The Euclidean distance measure used for vector quantization in the preceding section provides equal weights to individual components of the LSF vector, which obviously are not proportional to their spectral sensitivities. Weighted Euclidean distance was proposed to measure the LSF domain, which tries to assign weights to individual LSFs according to their spectral sensitivities. The weighted Euclidean distance measure between the test LSF vector  $f$  and the reference LSF vector  $f^{\wedge}$  is given by:

$$d(f, \hat{f}) = \sum_{i=1}^{10} [C_i W_i (f_i - \hat{f}_i)]^2 \tag{6}$$

Where  $f_i$  and  $\hat{f}_i$  are the  $i$ -th LSFs in the test and reference vector, respectively, and  $C_i$  and  $W_i$  are the weights assigned to the  $i$ -th LSF. These are given by

$$W_i = [P(f_i)]^r \tag{7}$$

Where  $P(f)$  is the LPC power spectrum associated with the test vector as a function of frequency  $f$  and  $r$  is an empirical constant that controls the relative weights given to different LSFs and is determined experimentally. The three multi-stage vector quantizer block diagram is shown in Figure 2. In Multistage Vector Quantization the input vector  $s$  to be quantized is passed through the first stage of the vector quantizer so as to obtain the quantized version of the input vector  $\hat{s}_1$ . The quantization error or residual error  $e_1 = s - \hat{s}_1$  at the first stage is the difference of the input vector and the quantized version of the input vector. The quantization error at the first stage is given as an input to the second stage vector quantizer to obtain the quantized version of the error vector  $\hat{e}_1$  at the first stage. The quantization error at the second stage  $e_2 = e_1 - \hat{e}_1$  is given as input to the third stage vector quantizer to obtain the quantized version of the error vector at the second stage  $\hat{e}_2$  and this process continues for the required number of stages. Finally the decoder takes the indices  $I_i$  from each quantizer stage and adds the corresponding codewords to obtain the quantized version of the input vector  $\hat{s} = \hat{s}_1 + \hat{e}_1 + \hat{e}_2$

$$e_1 = s - \hat{s}_1 \tag{8}$$

$$e_2 = e_1 - \hat{e}_1 \tag{9}$$

$$\hat{s} = \hat{s}_1 + \hat{e}_1 + \hat{e}_2 \tag{10}$$

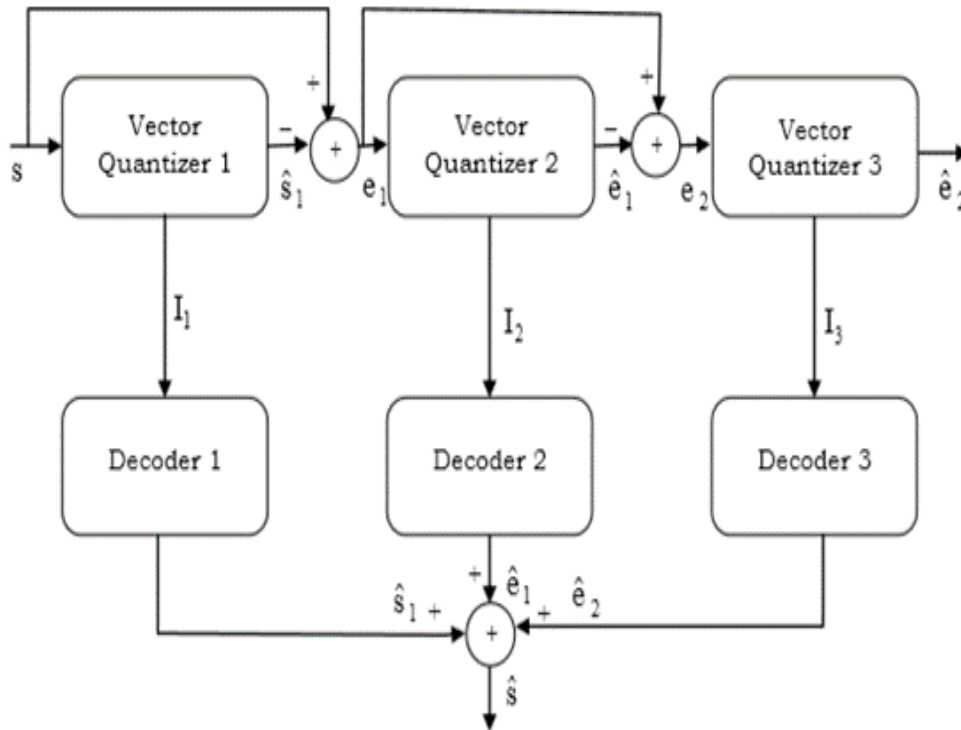


Figure2. Codebook generation for different stages of MSVQ

$$A(z) = 1 + \sum_{k=1}^p a_k z^{-k}, \quad \hat{A}(z) = 1 + \sum_{k=1}^p \hat{a}_k z^{-k} \quad (11)$$

$$d_{SD}(A(z), \hat{A}(z)) \approx \sqrt{\frac{1}{n_1 - n_0} \sum_{n=n_0}^{n_1-1} \left[ 10 \log_{10} \left( \frac{|\hat{A}(e^{j2\pi n/N})|^2}{|A(e^{j2\pi n/N})|^2} \right) \right]^2} \quad (12)$$

$$SD_{avg} = \sqrt{\frac{1}{T} \sum_{i=1}^T d_{SD}^2(e^{j2\pi i/N})} \text{ dB}^2 \quad (13)$$

## 5. Spectral Distortion Calculation

To evaluate the performance of the MSVQ and SVQ codebooks, the spectral distortion (SD) was measured. Spectral distortion is one of the most frequently used objective measure technique for evaluating the performance of LSF quantizers. The quantization performance of LPC parameters can be evaluated by the average spectral distortion of all the frames. The spectral distortion of each frame is defined as the root mean squared error between the power spectral density estimate of the original and the reconstructed LPC parameter vector. The spectral distortion can be calculated by the following formula and defined in dB;

Where  $A(z)$  and  $\hat{A}(z)$  denote the original and quantized  $p$ th order LPC polynomials, while  $n_1$  and  $n_0$  and correspond to lower (100Hz) and upper (3800Hz) frequencies respectively in our algorithm.  $A(z)$  is the optimal  $p$ th order linear predictor and  $\hat{A}(z)$  is the predictor with quantized coefficients.  $N = 256$  point FFT is used. The SD value is calculated over the bandwidth of 0-4kHz due to the 8kHz sampled input speech data. The spectral distortion is evaluated for all frames in the test data and its average value is computed. This average value represents the distortion associated with a particular quantizer. As mentioned before, the average spectral distortion has been used extensively in the past to measure the performance of LPC parameter quantizers. Transparent coding means that the coded speech is indistinguishable from the original speech through listening tests. For a transparent coding of an LPC based quantization of speech, the average SD should be about approximately 1dB.

## 6. Design of SVQ and MSVQ Codebooks and Performance

In this study, different codebooks were designed using K-mean algorithm. The multistage vector quantization and split vector quantization methods were used to generate the codebooks of bit rates from 14 to 24 bits/frames. Then, the

SD is calculated using clean and noisy speech data. During the experiments, standard TIMIT speech database containing American and British English speeches are used. TIMIT is originally sampled at 8kHz. For designing of SVQ or MSVQ codebooks, the LSF vectors were calculated by using TIMIT database as an input speech signal. The frame length is about 30ms with hamming window and overlap 10ms. 10th order LPC analysis based on an autocorrelation method was performed for every 20ms frame. The resulting coefficients of the 10th order LP polynomial  $A(z)$  were converted into the LSF domain. The input speech data used in the experiments divided into two separate databases as training of length 30 minutes and test database of length 15 minutes. The test database is divided into clean and noisy speeches. After design of codebooks, the spectral distortion (SD) values were calculated over the frequency band of 100-3800Hz for 8 kHz sampled speech.

For training and testing MSVQ codebooks, we have used 3 and 4 stages MSVQ codebooks for clean and noisy input speeches and compared SD result of each codebooks. The MSVQ codebooks in this study are designed using the M-L search, in which the M-best vector combinations are searched at each of the L stages, achieves performance close to that of the optimal search with a relatively low complexity. We used  $M=8$  search depth for training and testing. In our algorithms, a large number of three and four stage codebooks with different numbers of bits and different bit allocations were trained and evaluated [8, 9]. The largest size of an individual codebook was selected as 8 bits (256 code vectors). The following observations were made based on our simulation results. When we compared 3-stage and 4-stage codebook SD results, it is seen that spectral distortion increases when stage numbers are increased for each bit/frame numbers. From these results, it can be said that, the number of stages should be kept as low as possible to minimize the spectral distortion value. As one can expect, the spectral distortion increased when we used the noisy input speech data for testing the codebooks. The split that has been used in our study are (3,3,4), (4,4,2) and (4,3,3) 3-split codebooks. The most common splitting

scheme is (3, 3, 4) in which the first split contains the first three components of the 10 dimensional LSF vector, LSF1–LSF3, the second split consist of LSF4–LSF6 while LSF7–LSF10 constitute the third split. For evaluating the performance of the 2-split VQ, the LSF vectors were split as (4, 6), (6, 4) and (5, 5). The quantizers were trained using K-mean algorithm and the maximum size of an individual codebook was restricted to 1024 vectors (10 bits) For 2-split quantizers. The spectral distortion results of 2-split & 3-split SVQ codebooks for clean speech input data can be seen in Table 2, 3 and 4, respectively. The spectral distortion results for noisy speech input data can be seen in Table 5 and Table 6.

In split vector quantization we used two and three splitting's. The results in Table 4 indicate that the best performance was obtained in (4,4,2) as shown in figure 3 split codebooks when we compared to (3,3,4) and (4,3,3). According to these results, we can say that when we use large codebooks for the first and second split, quantization accuracy increases according to other equal size codebooks. We can also say that, when we compare SD values of SVQ and MSVQ for the same bitrate codebooks, MSVQ gives better performance. Within 3-stage codebooks, we obtained best performance SD value in 24bit/frame.

Evaluation of the 3-split quantizers, relatively large codebooks can be used for the first and the second split. These codebooks should preferably be of equal sizes or alternatively one extra bit may be allocated for the second split. However, when the bit rate increases the largest allowed codebook size becomes as limiting factor and a few bits are wasted as they are forced to be allocated for the least significant third split. In our simulations, this limit was reached at the bit rate of 24, as the codebooks of the first two splits could not be enlarged and thus more than three bits had to be allocated for the last split. At higher bit rates, the best performance was obtained with the (4,4,2) splitting, where the largest codebook is required for the middle split and the codebooks for the first as well as for the third split can be of equal sizes or one extra bit can be allocated for the first split. Among the 2-split quantizers included, the best performance was obtained using the quantizer with (6, 4) splitting as long as the difference between codebook sizes could be maintained large enough. In multistage vector quantization a training speech sequence is 1st used to generate the codebook. Speech signal is segmented (windowed) into successive short frames and each frame of speech is represented by a vector. Tables 4 and Table 6 show the spectral distortion (dB), at various bit rates for a 3-stage multistage vector quantizer and 4-stage multistage vector quantizer, as we can see from the tables that with 3-stage gives better performance than in 4-stages in terms of SD. As shown in Figure 4 we compared 4-stage multistage vector quantization with the best 3-split in split vector quantization (4,4,2) as we can see that 4-stage MSVQ gives very good performance in SD.

The bit allocations for SVQ and MSVQ that yielded the best performance at different bit rates are given in Tables below.

**Table 2.** 2-split (SVQ) using clean speech

Bits/frame	SD(dB) for (6,4)	SD(dB) for (5,5)	SD(dB) for (4,6)
14	2.81	2.75	2.90
16	2.49	2.51	2.53
18	2.22	2.20	2.24
20	2.11	2.04	2.03

**Table 3.** 3-split (SVQ) using clean speech

Bits/frame	SD(dB) for (3,3,4)	SD(dB) for (4,4,2)	SD(dB) for (4,3,3)
14	2.95	2.82	3.13
16	2.78	2.70	3.03
18	2.50	2.55	2.59
20	1.28	2.36	2.28
22	2.01	1.95	2.07
24	1.86	1.80	1.84

**Table 4.** Result for 3 and 4 stages (MSVQ) using clean speech

Number of bits	SD for 3-stage	Number of bits	SD for 4-stage
14	1.84	14	2.09
16	1.70	16	1.95
18	1.51	18	1.82
20	1.33	20	1.62
22	1.19	22	1.44
24	1.05	24	1.26

**Table 5.** Result for 3-split (SVQ) using noisy speech

Bits/frame	SD(dB)for (3,3,4)	SD(dB)for (4,3,3)	SD(dB)for (4,4,2)
14	4.22	4.64	3.92
16	3.94	4.55	3.81
18	3.76	3.91	3.53
20	3.38	3.22	3.30
22	3.06	2.98	2.88
24	2.91	2.73	2.68

**Table 6.** Result for 3 and 4 stages (MSVQ) using noisy speech

Number of bits	SD for 3-stage	Number of bits	SD for 4-stage
14	3.87	14	4.09
16	3.70	16	3.86
18	3.45	18	3.65
20	3.28	20	3.43
22	3.08	22	3.29
24	2.91	24	3.06

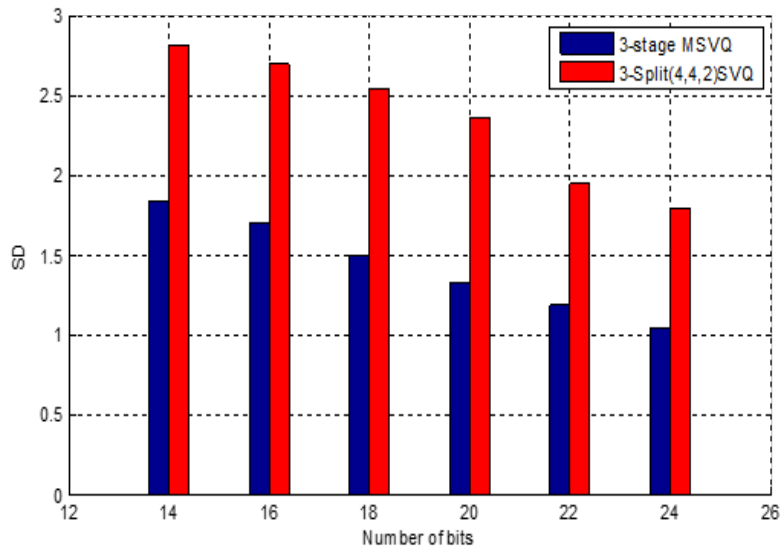


Figure 3. 3-stage MSVQ and 3-split SVQ for clean speech

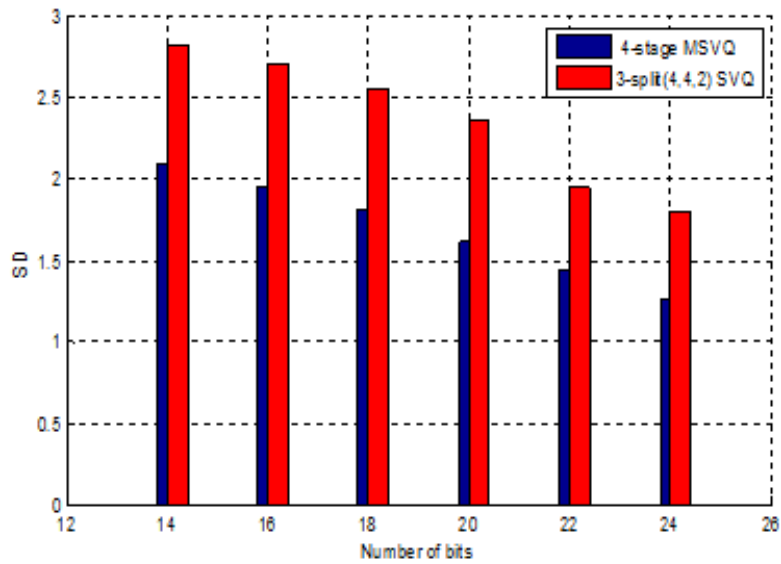


Figure 4. 4-stage MSVQ and 4-split SVQ for clean speech

## 7. Conclusions

The performance of two popular quantization methods, SVQ and MSVQ, was evaluated in LSF quantization. To achieve the best performance with MSVQ, it was found that the number of stages should generally be kept at minimum provided that maximum size codebooks are used for each stage. The evaluation of SVQ with 22 different splitting schemes indicated that none of the splitting schemes outperformed the others at all bit rates. In MSVQ two types of multistage were designed which are 3 and 4 MSVQ and calculating the spectral distortion for 33 different bit allocation. The bit allocation was more complicated with SVQ than with MSVQ because the optimal allocation depends on both the sub-vector lengths

and the frequency bands that the splits cover. The comparison between the quantization methods indicated that MSVQ clearly outperforms SVQ. By using MSVQ instead of SVQ, at least 2–3 bits can be saved without any quality degradations. In terms of SD as can be seen from tables and figures MSVQ gave better performance than SVQ.

## REFERENCES

[1] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. COM-28, no.1, pp.84–95, Jan.1980.

- [2] M.Suman, K.B.N.Prasanna ,Kumar, K.Kavindra Kumar , G. Phrudhi Teja “A new approach on Compression of Speech Signals using MSVQ and its Enhancement Using Spectral Subtraction under Noise free and Noisy Environment”, *Advances in Digital Multimedia* 52 Vol. 1, No. 1, March 2012
- [3] P. Kroon and W. B. Kleijn. *Linear Predictive Analysis by Synthesis Coding*, in *Modern Methods of Speech Processing*, Chapter 3. (Kluwer Academic Publishers, 1995)
- [4] Ulpu Sinervo, Jani Nurminen, Ari Heikkinen, and Jukka Saarinen, “Evaluation of split and multistage techniques in 22 LSF quantization”, 2003.
- [5] V. Krishnan, D.V. Anderson and K.K. Truong, “Optimal multistage vector quantization of LPC parameters over noisy channels”, *IEEE Trans. Speech Audio Processing*, Vol. 12, No. 1, pp. 1–8, Jan. 2004.
- [6] Kanawade. P. R, Gundal .S .S, “Tree structured vector quantization based technique for speech compression,” 237 *International Conference on Data Management, Analytics and Innovation (ICDMAI)*, 2017.
- [7] Satya Sai Ram .M , Siddaiah .P, and M.Madhavi Latha, “Multi Switched Split vector quantizer,” *International Journal of Computer, Information, and Systems Science, and Engineering*, pp. 1-6, Winter 2008.
- [8] Kleijn. W. B, Paliwal .K.K, “An introduction to Speech coding,” *Speech coding and synthesis*, Elsevier science, pp.13 1-47, 1995.
- [9] Chin-Hui Lee, “Robust linear prediction for speech analysis,” in: *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing*, Vol. 12, pp. 289–292, Apr 1987.