Codebook Design Using Simulated Annealing Algorithm for Vector Quantization of Line Spectrum Pairs

Fatiha Merazka

Telecommunications Department
University of science & technology Houari Boumediene
P.O.Box 32 El Alia 16111 Bab Ezzouar, Algiers
Algeria

fmerazka@usthb.dz

Abstract. An important issue in vector quantization (VQ) is the design of the codebook. The standard method for codebook design has been the generalized Lloyd algorithm (GLA) and Lind, Buzo and Gray (LBG) algorithm. Theses algorithms can get stuck in suboptimal codebooks due to the presence of several locally minimum distortion values. Simulated annealing (SA) is an optimization procedure that uses randomness to escape local minima in its search for a globally minimum state. In this paper, we propose a method of applying simulated annealing to VQ codebook design problem. The results presented for speech samples represented by line spectrum Pairs (LSP) indicate that the resulting design with simulated annealing are better compared to GLA and LBG algorithms.

Keywords: LSP, Simulated Annealing, GLA, LBG, MSE, SNR

1 Introduction

VQ has become a powerful tool and its application has been frequently reported in the speech and image coding literature [1-3]. The basic definition of a vector quantizer Q of dimension n and size K is a mapping of a vector from n-dimensional Euclidean space, R^n to a finite set, C, containing K reproduction code-vectors [1]:

$$Q : R^n \rightarrow C, \tag{1}$$

where $C = \{y_i : i \in I\}$ and $y_i \in R^n$ [1]. Associated with each reproduction code-vector is a partition of R^n, called a region or cell, $S = \{S_i : i \in I\}$ [4]. The most popular form of VQ is the nearest neighbor VQ, where for each input source vector, x, a

9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)
19-22 June 2012, Queen Mary University of London

search is done through the entire codebook to find the nearest code-vector, y_i, which has the minimum distance [5]:

$$y_i = Q(x) \tag{2}$$

$$\text{if}\, d(x, y_i) < d(x, y_j) \quad for \quad i \neq j \tag{3}$$

where $d(x, y)$ is a distance measure between the vectors, x and y. The mean squared error (*mse*) is used as the distance measure. Depending on the coding application, other more meaningful distance measures may be used such as the the Mahalanobis distance [6], Itakura–Saito distance [7], or other perceptually-weighted distance measures [8].

If the dimension of the vectors is n and a codebook of K code-vectors is used, each vector will be represented as a binary code of length $\lceil \log_2 K \rceil$ bits. Hence the bitrate of the vector quantizer is given by $\frac{1}{n} \lceil \log_2 K \rceil$ bits/sample [5].

Codebook design is the key problem of VQ and the generated codebook has more effect on the compression performance. The most widely used technique to create codebooks is a generalized Lloyd algorithm (GLA)[9], which is an iterative descent technique where an initial codebook is continually refined so that each iteration reduces the distortion involved in coding a given training set. The GLA algorithm provides no guarantee of optimality; a locally optimal solution may be obtained. The Linde–Buzo–Gray (LBG) algorithm [10] is an extension of the iterative Lloyd method [9], for use in VQ design. Because the LBG algorithm is not a variational technique, it can be used for cases where: the probability distribution of the data is not known a priori; we are only given a large set of training vectors; and the source is assumed to be ergodic [10]. The LBG algorithm involves refining an initial set of reproduction code-vectors using the Lloyd conditions [11], based on the given training vectors. The iterative procedure is stopped after the change in distortion becomes negligible.

Research efforts in codebook design have been concentrated in two directions: to generate a better codebook that approaches the global optimal solution, and to reduce the computational complexity.

All of the above algorithms have a local minimum problem. That is, the codebook guarantees local minimum distortion, but not global minimum distortion. To solve this problem, simulated annealing algorithms applied to image coding [12-14] have been proposed. Also, the method of using different initial points to find different codebooks, and then selecting the least distortion codebook as the final codebook, has been investigated. These last two methods can improve the codebook, but they increase the complexity significantly, and they cannot guarantee global optimality.

Competitive learning has also been applied to codebook design [15-20]. Codebook design algorithms based on evolutionary computation are new methods. In the design of the VQ codebook, the genetic algorithm (GA) is a random optimization algorithm based on the process of biological evolution by natural selection and genetic variation. GA has strong global search ability, but a weak local optimum capacity and slow convergence rate. It has advantages of easy use, universality and wide range of application [21-24].

Research on codebook design for VQ using simulated annealing has spanned over twenty years. Most work was focused on codebooks design for image coding. Less attention was paid to codebook design for speech signals.

This paper explores the application of SA algorithm to design codebooks for split VQ (SVQ) of line spectrum pairs (LSP) parameters of speech signals and compare them with GLA and LBG, since these two algorithms remain used most often for developing codebooks [25,26].

Our motivation for the use of LSP coefficient, to represent speech, is due to the fact that in many speech coders, the parameters of the all-zero predictor filter or the corresponding all-pole synthesis filter are coded and sent as part of the information stream [27]-[30].. Recently, there has been a growing interest in the use of (LSP's) to code the filter parameters for linear predictive coding (LPC) of speech

LSP's are an alternative to the direct form predictor coefficients or the lattice form reflection coefficients for representing the filter response. The direct form coefficient representation of the LPC filters is not conducive to efficient quantization. Instead, nonlinear functions of the reflection coefficients (e.g., log-area ratio or inverse sine of the reflection coefficient) are often used as transmission parameters [31]. These parameters are preferable because they have a relatively low spectral sensitivity.

LSP's are an alternate parameterization of the filter with a one-to-one correspondence with the direct form predictor coefficients. The concept of an LSP was introduced by Itakura [32]. LSP's encode speech spectral information more efficiently than other transmission parameters [28,33].

We have opted for the quantization of the LSPs by SVQ. Our choice is justify by the fact that VQ provides greater quantization efficiency than the scalar quantization due to the high correlation between neighboring spectral lines and the intuitive spectral interpolation [1]. Moreover, and in order to make VQ practical for large dimension and high bitrates, a structure can be imposed on the codebook to decrease the search complexity and/or storage requirements. One way of achieving this is to use decompose the codebook into a Cartesian product of smaller codebooks [1,34].

We will apply a 3-3-4 SVQ at 24 bits/frame to test our codebooks design. SVQ was first introduced by Paliwal and Atal [28,35] for quantization of line spectrum frequencies (LSF) in narrowband CELP speech coders and is used in the adaptive multirate narrowband (AMR-NB) codec [36]. SVQ is also used for quantizing Mel frequency-warped cepstral coefficients (MFCCs) in the ETSI distributed speech recognition (DSR) standard [37].

In all cases, the codebook is designed to minimize the mean squared error *mse*, defined by:

$$mse = \frac{1}{k}\sum_{i=0}^{k-1}\frac{1}{n}\left\|\left(x_i - y_i\right)\right\|^2 \tag{4}$$

where x_i is the *i*th input sample and y_i is the *i*th output codeword, n is the dimension of the vectors and k is the size of training sequence.

The signal to noise (*SNR*) ratio to be maximized is given by:

$$SNR = 10\log_{10}\frac{source \quad power}{quantization \quad error} = 10\log_{10}\frac{\sigma_{source}^2}{mse} \tag{5}$$

The rest of this paper is organized as follows. In section 2, definitions and properties of LSP parameters are presented. In section 3, the SVQ method used for the quantization of LSP coefficients is detailed. The SA algorithm is presented in section 4. Simulation results and discussions are given in section 5. Section 5 is dedicated to the conclusion.

2 LSP Properties

The linear predictive coding (LPC) method [38] is one of the most popular approaches for describing the time varying short-term spectrum of the speech signal. In many speech coding systems, LPC coefficients are transformed to the Line Spectrum Pairs (LSP) parameters [32] which are very effective representation for quantization of the LPC information. These parameters are preferable because they have a relatively low spectral sensitivity. This can be attributed to the intimate relationship between the LSP's and the formant frequencies. Accordingly, LSP's can be quantized taking into account spectral features known to be important in perceiving speech signals. In addition, LSP's lend themselves to frame-to-frame interpolation with smooth spectral changes because of their frequency domain interpretation. The LSP are related to the poles of the LPC filter (or the zeros of the inverse filter) in the Z-plane. For a p th-order LPC analysis, the Z-transform of the LPC inverse filter is denoted by:

$$A_p(z) = 1 + a_1 z^{-1} + \ldots + a_p z^{-p} \tag{6}$$

The parameters $\{a_i\}$ $i = 1,2,\ldots,p$, are commonly referred to as the LPC coefficients [38],

From (1) two new polynomials are defined:

$$\left.\begin{matrix} P(z) \\ Q(z) \end{matrix}\right\} = A_p(z) \pm z^{-(p+1)} A_p\left(z^{-1}\right) \tag{7}$$

The roots of these polynomials are usually called the Line Spectrum Pairs (LSP). These polynomials have the following properties:

All zeros of LSP polynomials are on the unit circle.

Zeros of P(z) and Q(z) are interlaced with each other on the unit circle.

The minimum phase property of $A_p(z)$ can be easily preserved if the first two properties are intact after quantization.

Some important properties are described in detail in [32].

The *10*th-order linear prediction corresponds to the frequency range of narrowband speech coders [39, 40].

3 Split vector quantization of LSP parameters

In this section we will present the SVQ definitions used for LSP coefficients quantization.

An m part, n-dimensional SVQ [2] operating at b bits/vector, divides the vector space, R^n, into m lower dimensional subspaces, $\left\{R_i^{n_i}\right\}_{i=1}^m$, where $n = \sum_{i=1}^m n_i$. Independent codebooks, $\{C_i\}_{i=1}^m$, operating at $\{b_i\}_{i=1}^m$ bits/vector, where $b = \sum_{i=1}^m b_i$, are then designed for each subspace. In order to quantize a vector of dimension n, the vector is split into subvectors of smaller dimension. Each of these subvectors is then encoded using their respective codebooks. The memory and computational requirements of the SVQ codebook are smaller than that of an unstructured VQ codebook. In terms of the number of floating point values for representing the SVQ codebooks as opposed to that of unstructured VQ:

$$\sum_{i=1}^{m} n_i 2^{b_i} \le n 2^b \tag{8}$$

while the effective number of code-vectors of the resulting product codebook is the same as that of unstructured VQ at the same bitrate:

$$\prod_{i=1}^{m} 2^{b_i} = 2^{b} \tag{9}$$

Therefore, the computational complexity and memory requirements for SVQ can be reduced considerably by splitting vectors into more parts.

In our study, the LSP parameters vector of dimension 10 is split into three sub-vectors, with the first sub-vector containing the three lowest LSP's, the second sub-vector containing the three middle LSP's and the final sub-vector containing the four highest LSP's [28].

4 Simulated Annealing

This section introduces the principle of SA algorithm suitable for solving the problem of VQ codebook design..

Simulated annealing is the computer modeling of the annealing process. By appropriately defining an effective temperature for the multivariable system, simulated annealing can solve a wide collection of optimization problems. Kirkpatrick et al. [12] were the first to use simulated annealing to solve such optimization problems. Starting from an initial state and with an initial temperature T_0, the simulated annealing proceeds as follows: Alter the state by a random perturbation, and compute the resulting change in the cost function, ΔE. If $\Delta E \leq 0$, then the perturbed state is accepted as the new state. If $\Delta E > 0$, then the perturbation is accepted with probability $p(\Delta E) = \exp(-\Delta E / T)$. The state of the system is repeatedly perturbed until either a fixed number of attempts are made or a minimum number of attempts are accepted. The temperature T is then reduced to the next lower temperature, and perturbations are again carried out. The number of perturbations attempted at each temperature and the sequence of temperatures is called the annealing schedule. Kirkpatrick et al. [41] recommended that the annealing schedule be developed by trial and error for a given problem, and chose $T_n = (0.9)^n T_0$, for the VLSI partitioning problem. Hajek [41] recommended the schedule $T_n = C/\log(n+1)$ since this helps guarantee the global minimum. The primary advantage of simulated annealing is its ability to avoid local minima in its search for the state with globally minimum energy. Changes that both decrease and increase the cost function are accepted, making escape from local minima possible. In the next section codebooks design based SA will be tested.

5 Simulation results and discussions

A key issue in VQ is the design of the codebook. Usually, VQ codebooks designed using GLA [9] and LBG [10] algorithms, as stated in the introduction, can get trapped in local minima. Here we will investigate the use of SA method to optimize codebooks for SVQ (3-3-4) with LSP parameters and compare its performance with GLA and LBG in terms of *mse*, *SNR*, number of iterations and time execution.

The LSP coefficients where generated from the ITU-T G.729 standard, which operates at 8 kbits/s[41]. The speech used is extracted from the TIMIT database [42]. The total number of vectors used for the training sequence is 229829.

Here, vectors of dimension 10 representing speech LSP parameters are splitted into three subvectors of dimensions 3, 3 and 4 respectively. The bit allocation for each subvector is 8 bits with a total number of 24 bits/vector.

The annealing schedule used is the common $T_n = (0.9)^n T_0$, with T_0 selected to achieve 99% acceptance. Twenty five acceptances or rejections were required before decreasing the temperature.

Table 1 summarizes the results obtained with the GLA algorithm. It shows the $mse_{initial}$ and mse_{final} when the mse (eq. 4) is used as a cost function. Four cases are considered for each sub-vector of dimension n. Results obtained, when SNR (eq. 5) is considered as cost function to be maximized, are also reported in Table 1. The $SNR_{initial}$ and SNR_{final} are given considering four cases as mse is used cost function.

The total number of iteration and time execution is also given for each case.

The initial codebooks, for the three sub-vectors considered here, are generated randomly from the training sequence. Codebook index corresponds to the initial codebook and can be any integer value.

Tests for the LBG algorithm, reported in Table 2, are given by the statistical properties corresponding to each subvector represented by means and variances and are. Table 2 shows also the optimal *mse* and *SNR* and the corresponding number of iterations and time execution for each subvector of dimension n and size K=256.

The results obtained for SA algorithm are summarized in Table 3. The initial temperature T_i and its corresponding mse_{initia} are given for each subvector of dimension n when the mse is used as a cost function. The same thing for the final T_f and its corresponding mse_{final} when the SNR is used as a cost function. It is also given in the same table the corresponding number of iterations and time execution for each vector of dimension n.

Table 1. Results obtained with GLA for codebooks of dimension n and size K=256 of speech LSPs

Code-book	$mse_{initial}$	mse_{final}	$SNR_{initial}$	SNR_{final}	Codebook Index.	iterations	Time
	0.001706	0.000157	12.734338	23.097418	0	14	1.152 s
$n=3$	0.000130	0.000073	23.921389	26.416121	50	12	0.981 s
K=256	0.000214	0.000062	21.748373	27.116219	100	9	0.741 s
	0.011953	0.000829	4.280092	15.866799	400	23	1.883 s
	0.002855	0.000418	16.250179	24.590706	0	24	1.983 s
$n=3$	0.000590	0.000299	23.095110	26.054218	50	9	0.742 s
K=256	0.000810	0.000232	21.722563	27.145992	100	9	0.741 s
	0.010768	0.001036	10.484547	20.650553	400	13	1.072 s
	0.002654	0.000377	17.164139	25.634670	0	27	2.213 s
$n=4$	0.000799	0.000355	22.378906	25.897436	50	15	1.241 s
K=256	0.000953	0.000448	21.614010	24.888277	200	7	0.58 s
	0.005073	0.000689	14.350595	23.024261	300	15	1.231 s

Table 2. Results obtained with LBG for codebooks of dimension n and size K=256 of speech LSPs

Codebook	Mean	Variance	mse_{final}	SNR_{final}	iterations	Time
$n=3$ K=256	0.509195	0.032024	0.000047	28.294531	92	2.233 s
$n=3$ K=256	1.288222	0.120389	0.000194	27.930449	90	1.922 s
$n=4$ K=256	2.304135	0.138155	0.000222	27.936451	86	1.672 s

Table 3. Results obtained with SA for codebooks of dimension n and size K=256 of speech LSPs

Code-book	T_i	T_f	$mse_{initial}$	mse_{final}	$SNR_{initial}$	SNR_{final}	iterations	Time
$n=3$ K=256	1	$9.89\ 10^{-36}$	0.011953	0.000034	4.280031	29.720501	19819	14min38s
$n=3$ K=256	10	$9.74\ 10^{-36}$	0.010768	0.000136	10.484393	29.469749	20299	15min26s
$n=4$ K=256	500	$9.88\ 10^{-36}$	0.005534	0.000186	13.972967	28.714369	21238	14min55s

Comparing Tables 2 and 3, we can see that SA algorithm gives better results than LBG in terms of mse_{final} for the three codebooks considered. We can also notice that the SA algorithm is more CPU time consuming compared to the LBG algorithm.
Comparing Tables 1, 2 and 3, the results obtained with GLA in terms of mse_{final}, SNR_{final} are worse than the other two methods, they are highly dependent on the initial codebook chosen.

Comparing Tables 2 and 3, we notice that he LBG algorithm is faster but less efficient than SA. This was expected because a descent method (LBG in our case) is theoretically less time consuming than Sa but less efficient. A descent method is often trapped in a local minimum especially when the objective function to minimize in our case, the *mse*, has several minima; this is due to the search criterion of a descent method. It evolves in its search for the solution (quantization codebook in our case) through the optimal set of solutions by not accepting a lower cost solution than the current solution from one step to another; it stops the search if a minimum is met but not necessarily the global minimum.

The performance of a descent method is directly related to the quality of the initial solution from which begins the search procedure for the optimal solution, that's what we found for GLA. GLA is a descent method, which is identical to LBG (same optimality conditions), but the major problem in the GLA algorithm is the choice of the initial codebook. We chose to create the initial codebook of GLA randomly and we noticed that some GLA codebooks approached the initial results obtained by LBG and SA, but more often in practice it is not easy to find an initial codebook that ensure the convergence of GLA to an optimal codebook, it may be more difficult than the original problem and therefore a waste of time and more without reaching suitable results.

SA performs better than LBG and GLA; this is due to the global search of SA. It accepts solutions that improve the cost of the objective function (in our case *mse* or

SNR) and also in a controlled manner (probabilistic) solution which degrade it. The performance of SA is directly related to the cooling scheme selected and the number of iterations per temperature. For a given time, the simulated annealing will approach as possible the optimal solution. In some problems the time required for simulated annealing performance could be seen as a disadvantage, but for the quantization problem this is not a waste of time because the quantization codebook is designed eternally.

6 Conclusion

Simulated annealing is a powerful optimization procedure that achieves near globally-minimum-cost solutions to many optimization problems. In this paper, we attempted to apply SA to improve the quality of codebooks SVQ for the quantization of spectral parameters represented by LSPs. SA provided the best SNR and mse results, avoiding the initial codebook dependence found when using the GLA.

Simulated annealing, while itself, too time consuming, does serve to obtain a near globally-optimum solution for codebook design. Future research will focus on more sophisticated algorithms based genetic algorithms and Tabu search.

References

1. Gersho, A., Gray, R. M.: Vector Quantization and Signal Compression. Kluwer Academic Publishers (1992)
2. Gray, R. M.: Vector quantization. Mag. IEEE Acou. Spee. Sig. Pro. 1, 4-29 (1984)
3. Nasrabadi, N. M., King, R. A.: Image coding using vector quantization:A review. IEEE. Tran. Com. 36, 957-971(1988)
4. R.M. Gray, D.L. Neuhoff, Quantization, IEEE Trans. Inform. Theory 44 (6) (1998) 2325–2383.
5. K. Sayood, Introduction to Data Compression, Morgan Kaufmann Publishers, San Francisco, 1996.
6. Mahalanobis, P.C. :On the generalized distance in statistics, in: Proc. Indian Nat. Inst. Sci. Calcutta, 2, 49–55 (1936).
7. Itakura, F. :Minimum prediction residual principle applied to speech recognition, IEEE Trans. Acoust. Speech Signal Process. ASSP-23 (1) (1975) 67–72.
8. Makhoul, J. Roucos, S. Gish, : Vector quantization in speech coding, Proc. IEEE 73 (1985) 1551–1588.
9. Lloyd, S.P.: Least square quantization in PCM. IEEE Trans. Inform. Theo.28, 129–137. (1982)
10. Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. IEEE Trans. Commun. COM. 28, 84–95 (1980)
11. Lloyd, S.P. :Least square quantization in PCM, IEEE Trans. Inform. Theory IT-28 (2) 129–137 (1982)
12. S. Kirkpatrick, C. D. Gellatt, Jr., and M. P. Vecchi. "Optimization by simulated annealing," Science. vol. 220. 671 -680 (1983).
13. Vaisey J. and Gersho, A. :Simulated annealing and codebook design," in Proc. IEEE ICASSP .,1176-1179 (1988)

14. Flanagan, J. K.. Morrell, D.R. Frost. R. L. Read, C. J and Nelson, B. E. V:ector quantization codebook generation using simulated annealing, in Proc. IEEE ICASSP, 3, 1759- 1762 (1989)
15. Nasrabadi, N. M., Feng, Y. "Vector Quantization of Images Based Upon the Kohonen Self-organizing Feature Maps." Proceedings IEEE International Conference on Neural Networks, 1, 101-108 (1988)
16. Wu, F. H., Ganesan, K. "Comparative Study of Algorithms for VQ Design Using Conventional and Neural-net Based Approaches." Proceedings International Conference on Acoustics, Speech, and Signal Processing (ICASSP'89), 2, 751-754(1989)
17. Madeiro, F., Vajapeyam, M. S., Morais, M. R., Aguiar Neto, B. G., and Alencar, M. S. "Multiresolution Codebook Design for Wavelet/VQ Image Coding". Proceedings of the 15th International Conference on Pattern Recognition (ICPR'20000 Barcelona, 3, 79-82, (2000).
18. Chang, C.-H., Xu, P., Xiao, R., Srikanthan, T. New Adaptive Color Quantization Method Based on Self-Organizing Maps. IEEE Transactions on Neural Networks, 16 (1), 237-249, (2005).
19. D.E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley, Reading, 1989.
20. V. Delport, N. Koschorreck, "Genetic Algorithm for Codebook Design in Vector Quantization", Electronics Letters, 31(2), 84-85 (1995)
21. Pan, J. S. Mcinnes, F. R.. Jack, M. A :VQ Codebook Design Using Genetic Algorithms", Electronics Letters, 31(17), 1418-1419 (1995)
22. Hsiang-Cheh Huang, Jeng-Shyang Pan, Zhe-Ming Lu, Sheng-He Sun, Hsueh-Ming Hang. Vector quantization based on genetic simulated annealing. Signal Processing, 81, 1513-1523 (2001)
23. Yuan, Y J Zhou, Q Zhao. P H :Vector quantization codebook design method for speech recognition based on genetic algorithm, .Proceedings of the 2010 2nd International Conference on Information Engineering and Computer Science. Wuhan, 1-4 (2010)
24. Santo, P.H.E.; Albuquerque, R.C.; Cunha, D.C.; Madeiro, F.;, On Frequency Sensitive Competitive Learning for VQ Codebook Design, Neural Networks, 2008. SBRN '08. 10th Brazilian Symposium on , 135-140, 26-30 (2008)
25. Kang, S.. Shin, Y and Fischer, T. R. Low-complexity predictive trellis coded quantization of speech line spectral frequencies, IEEE Trans. Signal Processing, 52, 2070-2079 (2004)
26. So S.and Paliwal, K.K.: A comparative study of LPC parameter representations and quantization schemes for wideband speech coding", Digital Signal Processing, 17(1), 114-137(2007)
27. Wakita, H. :Linear prediction voice synthesizers: Line spectrum pairs (LSP) is the newest of several techniques, Speech Technol., (1981).
28. Paliwal, K.K. Atal, B.S. :Efficient vector quantization of LPC parameters at 24 bits/frame, IEEE Trans. Speech Audio Process. 1 (1) 3–14 (1993)
29. A.M. Smith, J.P. Ashley, M.A. Jasiuk, W. Peng, Normalization and polygon error detection for split VQ of line spectral frequencies, in: IEEE Speech Coding Workshop, 125–27 (2000)
30. Nordén, F. Eriksson, T: .On split quantization of LSF parameters, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Montreal, I-157–I-160 (2004)
31. Markel, J. D. and Gray, A. H. Jr.: Linear Prediction of speech. New York: Springer-Verlag, 1976
32. Itakura, F. :Line spectrum representation of linear predictor coefficients of speech signal's," J. Acoust. SOC. Amer . , 57, S35(A), (1975)
33. Kang G. S . and. Fransen, L. J :Low bit rate speech encoders based on line spectrum frequencies (LSFs)," Naval Res. Lab., Rep. 8857, (1984)
34. Gray R. and Neuhoff, D. :Quantization, IEEE Trans. Inform. Theory, 44, 2325–2383(1998)

35. Paliwal, K.K. Atal, B.S. :Efficient vector quantization of LPC parameters at 24 bits/frame, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, 1991, pp. 661–664.
36. 3rd generation partnership project; Technical specification group services and system aspects; Mandatory speech codec speech processing functions; Adaptive multi-rate (AMR) speech codec; Transcoding functions (Release 5), Technical Specification TS 26.090, 3rd Generation Partnership Project (3GPP), June 2002.
37. Speech processing, transmission and quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms, Tech. Rep. Standard ES 201 108, European Telecommunications Standards Institute (ETSI), April 2000.
38. Makhoul, J.: Linear prediction: A tutorial review speech." Proc. IEEE. 63,124-143(1975)
39. ITU, ITU-T G.723.1: Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s, ITU 1996.
40. ITU, ITU-T G.729: CS-ACELP Speech Coding at 8 kbit/s, ITU 1998
41. Hajek, B.: A tutorial survey of theory and applications of simulated annealing," in Proc. 24th Conf. on Decision and Control, 755-760 (1985)
42. NIST,Timit Speech Corpus, NIST (1990)