The α_i coefficients are the direct form of LPC. The filter $H(z)$ is stable if it is minimum phase, i.e. all the roots of the equation (5.1) are within the unit circle. If α_i were quantized directly, the stability of the filter $H(z)$ is not easily guaranteed as the roots of equation (5.1) are not usually computed to check for stability. Thus a more useful parameter, the PARCOR (partial correlation) coefficients, k_i, are usually used for quantization. The distribution plots of PARCOR parameters for a 10^{th}-order LPC filter are shown in Figure 5.1. The forward and backward transformation are given below [3].

LPC to PARCOR:

$$a_j^p = \alpha_j \quad 1 \le j \le p$$

$$\begin{aligned} &\text{For } i = p, p-1, \ldots, 1 \\ &a_j^{i-1} = (a_j^i + a_i^i a_{i-j}^i)/(1 - k_i^2), \quad 1 \le j \le i-1 \\ &k_{i-1} = a_{i-1}^{i-1} \end{aligned} \tag{5.2}$$

PARCOR to LPC:

$$\begin{aligned} &\text{For } i = 1, 2, \ldots, p \\ &a_i^i = k_i \\ &a_j^i = a_j^{i-1} - k_i a_{i-j}^{i-1}, \quad 1 \le j \le i-1 \\ &\alpha_j = a_j^p, \quad 1 \le j \le p \end{aligned} \tag{5.3}$$

The LPC filter is stable if $|k_i| \le 1.0$. Although k_i can easily be checked for stability, they are not suitable for quantization because they possess a nonflat spectral sensitivity, i.e. values of k_i near unity require more quantization accuracy than those away from unity. Thus, nonlinear functions of k_i are required, with the Log-Area Ratio (LAR) and inverse sine (IS) functions being the most widely used [4]. For LAR and IS, the forward and backward transformation are given below:

PARCOR to LAR:

$$g_i = \log\left(\frac{1 - k_i}{1 + k_i}\right), \quad 1 \le i \le p \tag{5.4}$$

LAR to PARCOR:

$$k_i = \left(\frac{1 - 10^{g_i}}{1 + 10^{g_i}}\right), \quad 1 \le i \le p \tag{5.5}$$

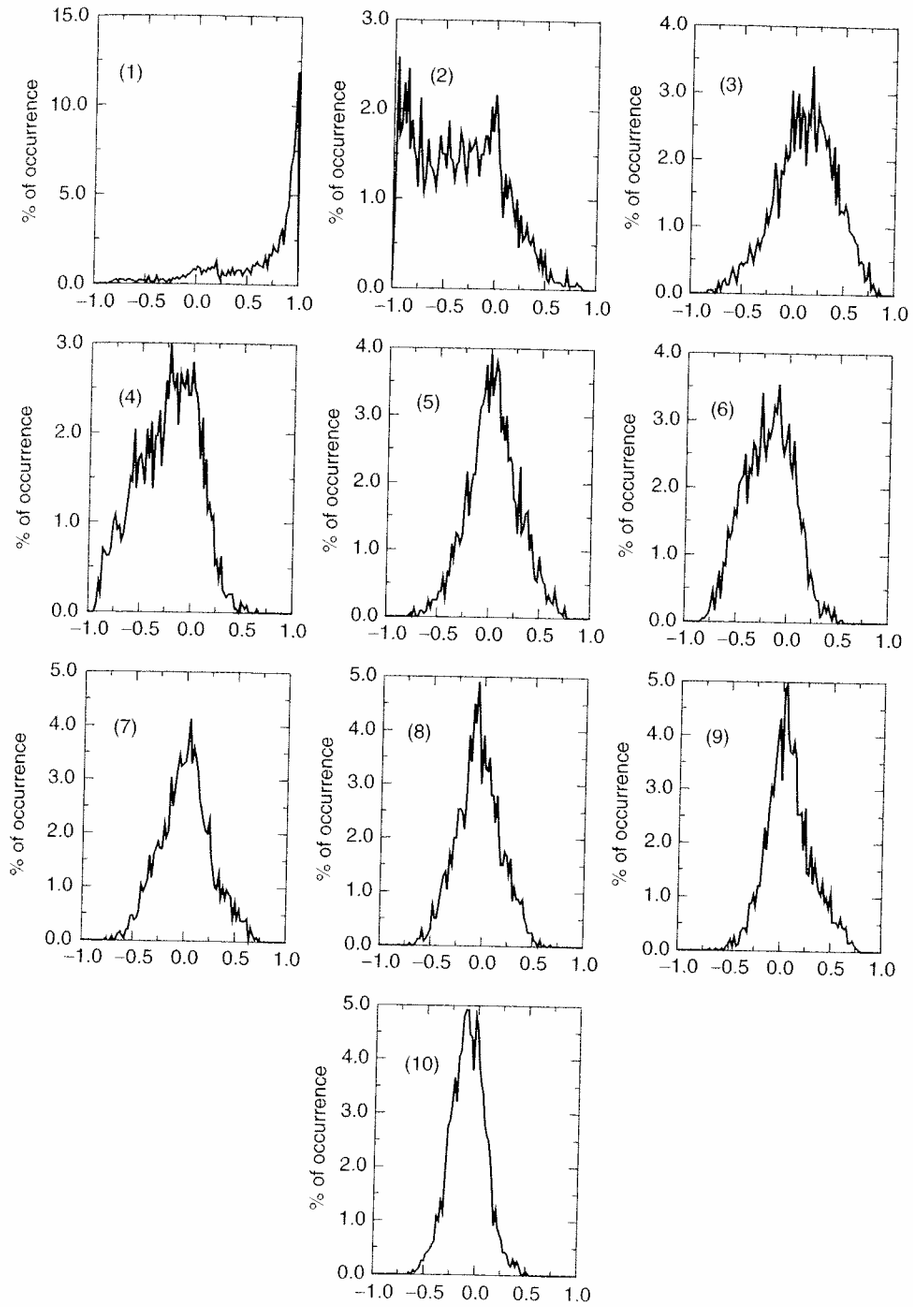

Figure 5.1 The distribution plots of PARCOR LPC

PARCOR to IS:

$$s_i = \sin^{-1}(k_i), \quad 1 \le i \le p \tag{5.6}$$

IS to PARCOR:

$$k_i = \sin(s_i), \quad 1 \le i \le p \tag{5.7}$$

The distribution plots of LAR and IS parameters for a 10^{th} order LPC filter are shown in Figures 5.2 and 5.3 respectively.

Although it is possible to design good performance quantizers using the LAR and IS representations, the frame-to-frame correlation of LPC (which evidently exists for slowly-varying parts of speech) is not highlighted in either LAR or IS representations, i.e. it is difficult to predict frame-to-frame parameter values. Thus, not all the redundancies are fully exploitable.

In view of the shortcomings of LAR and IS representation, the line spectral pairs (LSP) or frequencies (LSF) representations of LPC have been investigated [2]. The concept of LSF was introduced by Itakura, but it remained almost dormant until its usefulness was re-examined in the latest speech coding standards. LSFs encode speech spectral information in the frequency domain and have been found to be capable of improving the coding efficiency by more than other transformation techniques, especially when incorporated into predictive quantization schemes. For use in conventional scalar quantization, it has been shown by Cox [4] and others that LSF is not significantly better than LAR or IS, but it does have other properties which are desirable, as will be discussed in later sections. The fact that LSF representation is in the frequency domain means that quantization can easily incorporate spectral features known to be important in perceiving speech signals. In addition, LSFs lend themselves to frame-to-frame interpolation with smooth spectral changes because of their intimate relationship with format frequencies.

5.3 LPC to LSF Transformation

An all-pole digital filter for speech synthesis, $H(z)$, can be derived from linear predictive analysis and is given by

$$H(z) = 1/A_p(z) \tag{5.8}$$

where,

$$A_p(z) = 1 + \sum_{k=1}^{p} \alpha_k z^{-k} \tag{5.9}$$

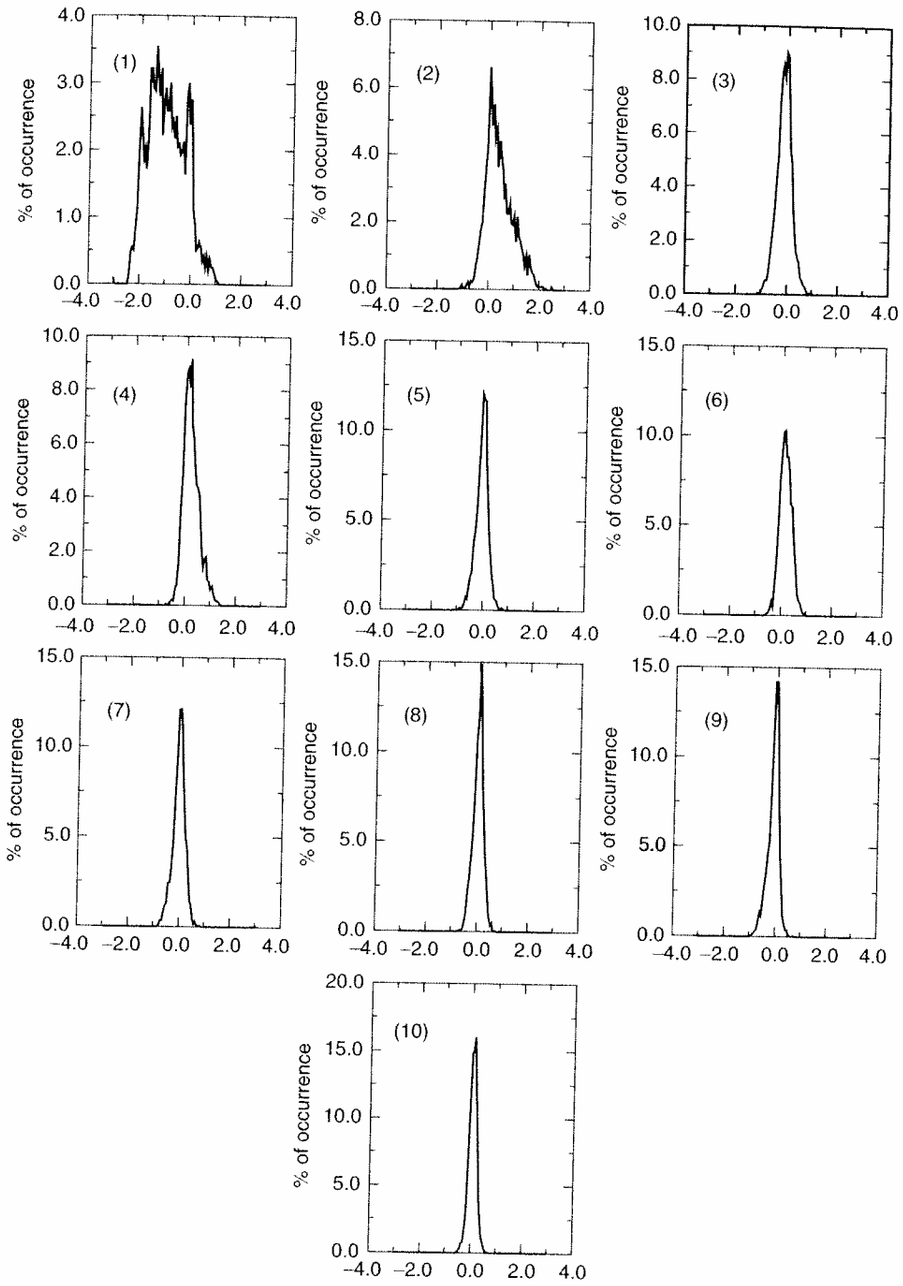

Figure 5.2 The distribution plots of LAR parameters

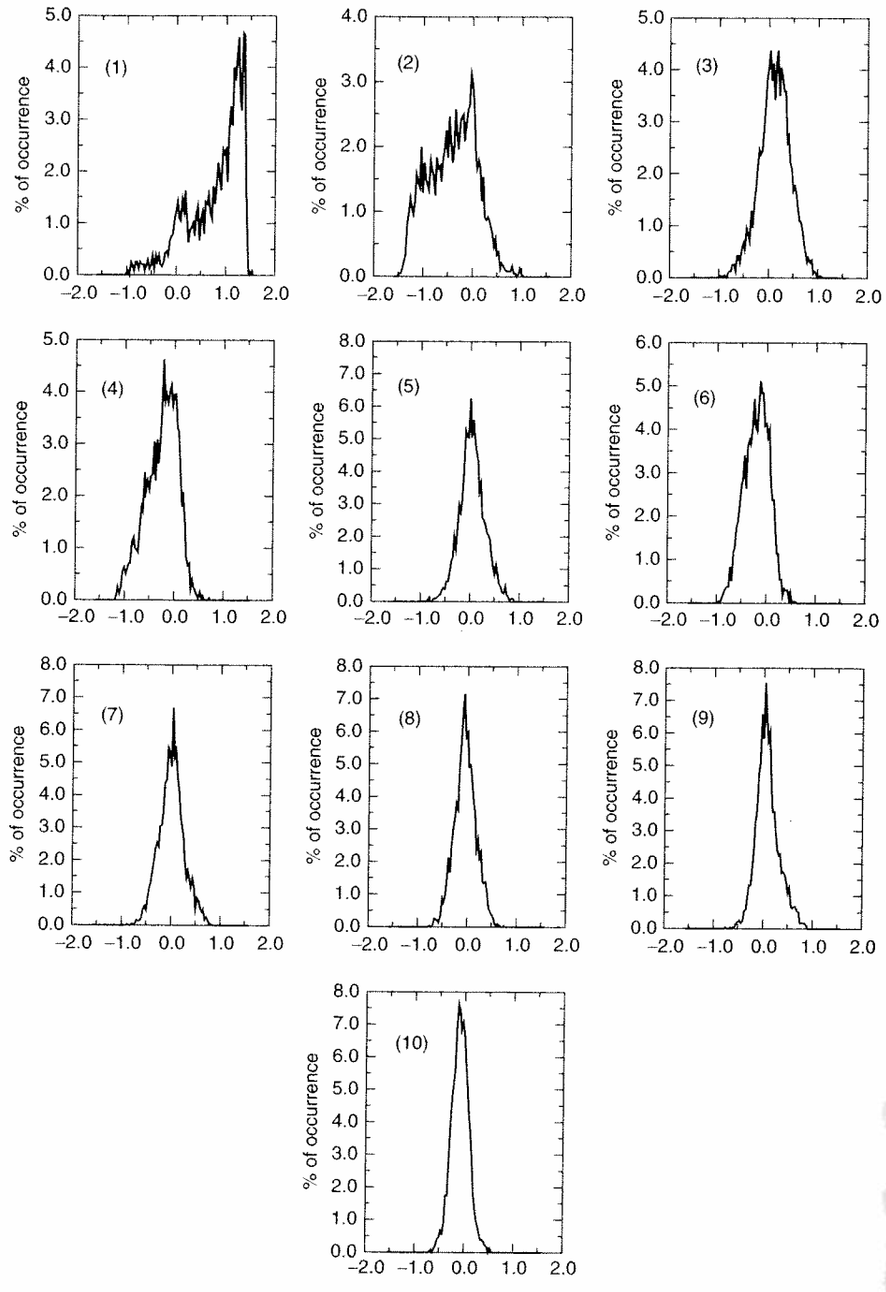

Figure 5.3 The distribution plots of inverse sine parameters (horizontal axis is in radians)

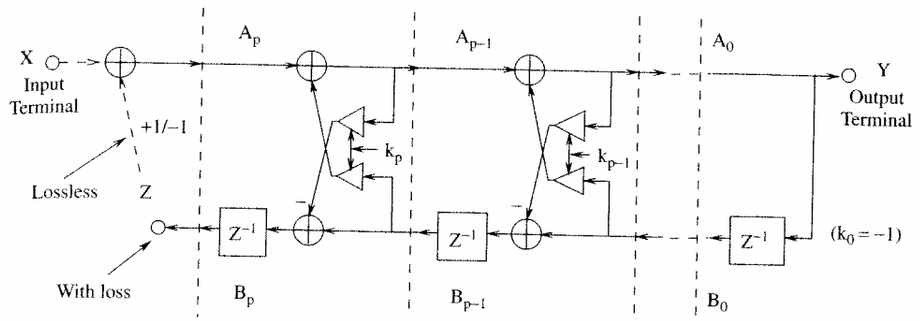

Figure 5.4 PARCOR structure of LPC synthesis

The PARCOR representation is an equivalent version and its digital form is as shown in Figure 5.4, where,

$$A_{p-1}(z) = A_p(z) + k_p B_{p-1}(z) \tag{5.10}$$

$$B_p(z) = z^{-1}[B_{p-1}(z) - k_p A_{p-1}(z)] \tag{5.11}$$

where $A_0(z) = 1$ and $B_0(z) = z^{-1}$, and

$$B_p(z) = z^{-(p+1)} A_p(z^{-1}) \tag{5.12}$$

The PARCOR representation as shown in Figure 5.4 is stable for $|k_i| < 1$ for all i. In Figure 5.4, the transfer function, TF, from X to Y is $H_p(z)$, and from Y to Z is $B_p(z)$, therefore the TF from X to Z is given by equation (5.13) where $R_p(z)$ is the ratio filter,

$$R_p = B_p(z)/A_p(z) \tag{5.13}$$

The PARCOR synthesis process can be viewed as sound wave propagation through a lossless acoustic tube, consisting of p sections of equal length but nonuniform cross sections. The acoustic tube is open at the terminal corresponding to the lips and each section is numbered from the lips. Mismatching between the adjacent sections p and $(p + 1)$ causes wave propagation reflection. The reflection coefficients are equal to the p^{th} PARCOR coefficient k_p. Section $p + 1$, which corresponds to the glottis, is terminated by a matched impedance. The excitation signal applied to the glottis drives the acoustic tube.

In PARCOR analysis, the boundary condition at the glottis is impedance-matched. Now consider a pair of artificial boundary conditions where the acoustic tube is completely closed or open at the glottis. These conditions correspond to $k_{p+1} = 1$ and $k_{p+1} = -1$, a pair of extreme values for the

artificially-extended PARCOR coefficients which correspond to perfectly lossless tubes. The value Q of each resonance becomes infinite and the spectrum of distributed energy is concentrated in several line spectra. The feedback conditions for $k_{p+1} = -1$ correspond to a perfect closure at the input (glottis) and for $k_{p+1} = 1$ correspond to an opening to infinite free space. To derive the line spectra or line spectrum frequencies (LSF), we proceed as follows (it is assumed that the PARCOR filter is stable and the order is even). $A_p(z)$ may be decomposed to a set of two transfer functions, one having an even symmetry and the other having an odd symmetry. This can be accomplished by taking a difference and sum between $A_p(z)$ and its conjugate functions. Hence the transfer functions with $k_{p+1} = \pm 1$ are denoted by $P_{p+1}(z)$ and $Q_{p+1}(z)$.

$$\begin{aligned} &\text{For } k_{p+1} = 1, \quad P_{p+1}(z) = A_p(z) - B_p(z) \ \text{ (Difference filter)} \\ &\text{For } k_{p+1} = -1, \quad Q_{p+1}(z) = A_p(z) + B_p(z) \ \text{ (Sum filter)} \end{aligned} \tag{5.14}$$

$$\Rightarrow A_p(z) = \frac{1}{2}[P_{p+1}(z) + Q_{p+1}(z)] \tag{5.15}$$

Substituting equation (5.12) into (5.14),

$$\begin{aligned} P_{p+1}(z) &= A_p(z) - z^{-(p+1)}A_p(z^{-1}) \\ &= 1 + (\alpha_1 - \alpha_p)z^{-1} + \ldots + (\alpha_p - \alpha_1)z^{-p} - z^{-(p+1)} \\ &= z^{-(p+1)} \prod_{i=0}^{p+1} (z + a_i) \end{aligned} \tag{5.16}$$

where a_i is generally complex. Similarly,

$$Q_{p+1}(z) = z^{-(p+1)} \prod_{i=0}^{p+1} (z + b_i) \tag{5.17}$$

As we know that two roots exist ($k_{p+1} = \pm 1$), the order of $P_{p+1}(z)$ and $Q_{p+1}(z)$ can be reduced, i.e.

$$\begin{aligned} P'(z) &= \frac{P_{p+1}(z)}{(1 - z)} \\ &= A_0 z^p + A_1 z^{(p-1)} + \ldots + A_p \end{aligned} \tag{5.18}$$

and,

$$\begin{aligned} Q'(z) &= \frac{Q_{p+1}(z)}{(1 + z)} \\ &= B_0 z^p + B_1 z^{(p-1)} + \ldots + B_p \end{aligned} \tag{5.19}$$

where,

$$A_0 = 1 \tag{5.20}$$

$$B_0 = 1 \tag{5.21}$$

$$A_k = (\alpha_k - \alpha_{p+1-k}) + A_{k-1} \tag{5.22}$$

$$B_k = (\alpha_k + \alpha_{p+1-k}) - B_{k-1} \tag{5.23}$$

$$\text{for } k = 1, \ldots, p$$

The LSFs are the angular positions of the roots of $P'(z)$ and $Q'(z)$ with $0 \leq \omega_i \leq \pi$. The roots occur in complex conjugate pairs and have the following properties:

1. All roots of $P'(z)$ and $Q'(z)$ lie on the unit circle.
2. The roots of $Q'(z)$ and $P'(z)$ alternate with each other on the unit circle, i.e. the following is always satisfied, $0 \leq \omega_{q,0} < \omega_{p,0} < \omega_{q,1} < \omega_{p,1} \ldots, \leq \pi$.

5.3.1 Complex Root Method

The roots of equation (5.18) can be solved using complex arithmetic. This will give complex conjugate roots on the unit circle and the frequencies are then given by the inverse tangent of the roots. This method is obviously very complex as it involves solving two polynomials of p^{th} order using complex arithmetic. Also, as it uses an iteration procedure for determining the roots, the time required for this method is not deterministic which is undesirable for real-time implementations.

5.3.2 Real Root Method

As the coefficients of $P'(z)$ and $Q'(z)$ are symmetrical the order of equation (5.18) can be reduced to $p/2$.

$$\begin{aligned} P'(z) &= A_0 z^p + A_1^{p-1} + \ldots + A_1 z^1 + A_0 \\ &= z^{p/2}[A_0(z^{p/2} + z^{-p/2}) + A_1(z^{(p/2-1)} + z^{-(p/2-1)}) + \ldots + A_{p/2}] \end{aligned} \tag{5.24}$$

Similarly,

$$Q'(z) = B_0 z^p + B_1^{p-1} + \ldots + B_1 z^1 + B_0 \tag{5.25}$$
$$= z^{p/2}[B_0(z^{p/2} + z^{-p/2}) + B_1(z^{(p/2-1)} + z^{-(p/2-1)}) + \ldots + B_{p/2}]$$

As all roots are on the unit circle, we can evaluate equation (5.24) on the unit circle only.

$$\text{Let } z = e^{j\omega} \text{ then } z^1 + z^{-1} = 2\cos(\omega) \tag{5.26}$$

$$P'(z) = 2e^{jp\omega/2}\left[A_0 \cos\left(\frac{p}{2}\omega\right) + A_1 \cos\left(\frac{p-2}{2}\omega\right) + \ldots + \frac{1}{2}A_{p/2}\right] \tag{5.27}$$

$$Q'(z) = 2e^{jp\omega/2}\left[B_0 \cos\left(\frac{p}{2}\omega\right) + B_1 \cos\left(\frac{p-2}{2}\omega\right) + \ldots + \frac{1}{2}B_{p/2}\right] \tag{5.28}$$

By making the substitution $x = \cos(\omega)$, equations (5.27) and (5.28) can be solved for x. For example, with $p = 10$, the following is obtained:

$$P'_{10}(x) = 16A_0x^5 + 8A_1x^4 + (4A_2 - 20A_0)x^3 + (2A_3 - 8A_1)x^2$$
$$+(5A_0 - 3A_2 + A_4)x + (A_1 - A_3 + 0.5A_5) \tag{5.29}$$

and similarly,

$$Q'_{10}(x) = 16B_0x^5 + 8B_1x^4 + (4B_2 - 20B_0)x^3 + (2B_3 - 8B_1)x^2$$
$$+(5B_0 - 3B_2 + B_4)x + (B_1 - B_3 + 0.5B_5) \tag{5.30}$$

The LSFs are then given by:

$$LSF(i) = \frac{\cos^{-1}(x_i)}{2\pi T}, \quad \text{for } 1 \le i \le p \tag{5.31}$$

The distribution plots of LSFs for a 10^{th} order LPC filter are shown in Figure 5.5 and a typical LSF plot is shown in Figure 5.6, where the first half is active speech and the second half is silence. Notice that during silent regions the frequencies are evenly spread between 0 and $f_s/2$ where f_s is the sampling frequency. This method is obviously considerably simpler than the complex root method, but it still suffers from indeterministic computation time. However, a faster root search can be accomplished by noting that the change from one LSF vector to the next is not too drastic in most cases. Thus by using the previous values as the starting estimates of the roots, the number of iterations required per root is considerably reduced, e.g. typically from 5 to 10 iterations.

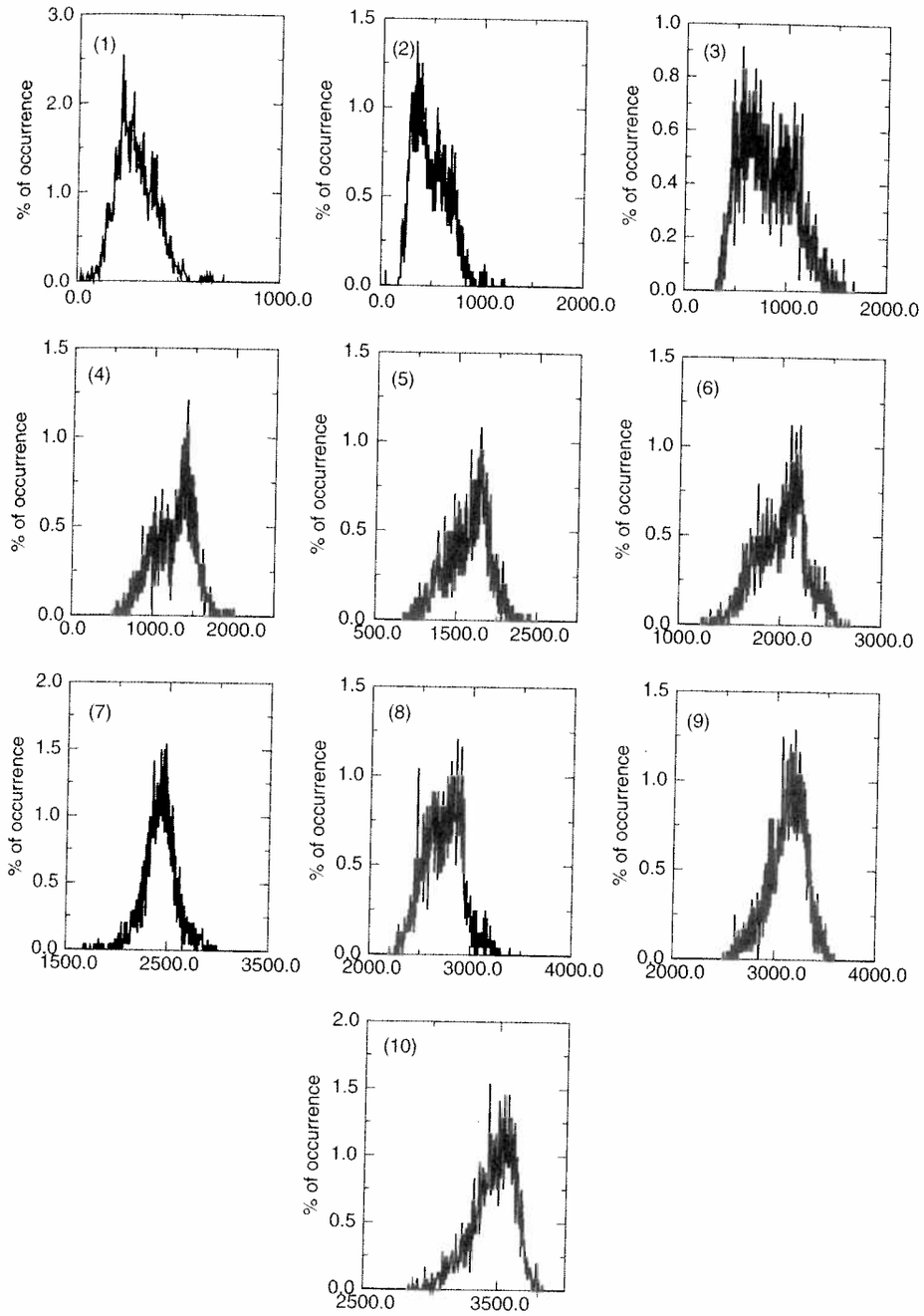

Figure 5.5 The distribution plots of LSF parameters (horizontal axis is in Hz)

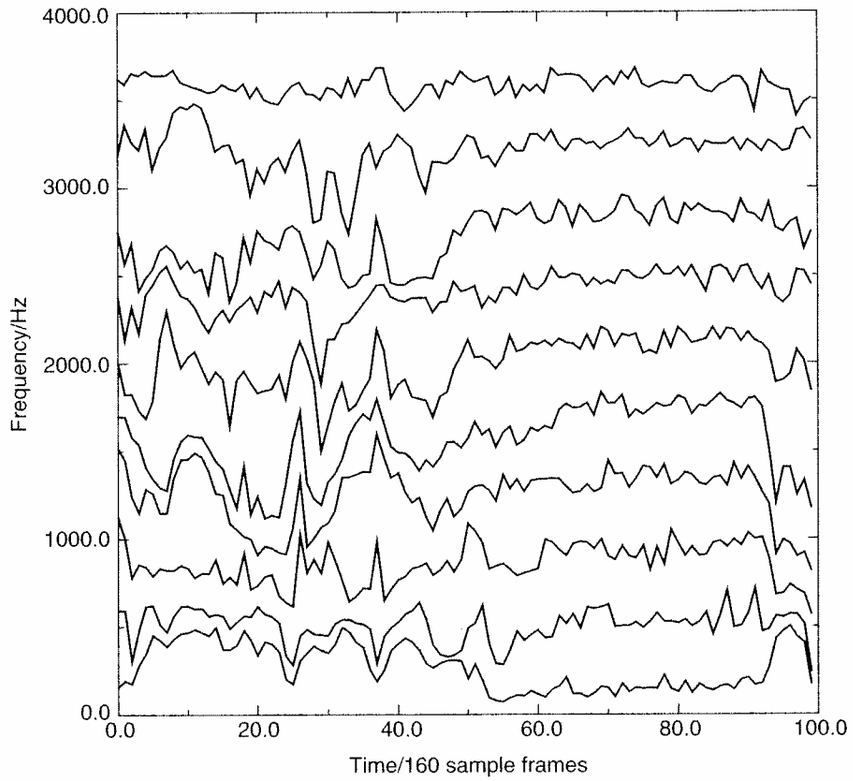

Figure 5.6 Typical LSF trajectories for voiced and unvoiced speech

5.3.3 Ratio Filter Method

The expression for the ratio filter is given by equation (5.32). The phase response, $\phi(kf_s)$, of the ratio filter is given by equation (5.34). The frequency corresponding to a multiple of $-\pi$ and -2π radians are the lower and upper line spectra of the LSF [5].

$$R_p(z) = \frac{z^{-(n+1)}A_p(z^{-1})}{A_p(z)} \tag{5.32}$$

where,

$$A_p(z) = 1 - \sum_{i=1}^{n} \beta_i z^{-i} \tag{5.33}$$

and $\beta_i = -\alpha_i$ where α_i are the LPC.

$$\phi(kf_s) = -(n+1)(2\pi Tkf_s) - 2\tan^{-1}\left\{\frac{\sum_{i=1}^{n}\beta_i \sin(2\pi iTkf_s)}{1 - \sum_{i=1}^{n}\beta_i \cos(2\pi iTkf_s)}\right\} \tag{5.34}$$

where T is the sampling period, f_s is the frequency step, and $k = 1, 2, 3, \ldots, K_{max}$.

By performing a Discrete Fourier Transform (DFT) on the coefficient sequence, A_k and B_k, ω_i can be solved as the zero-valued frequencies of a power spectrum. A typical plot, showing the partial minima of the spectrum, is shown in Figure 5.7.

If the spectrum were to be obtained directly, it would involve an enormous number of computations. Fortunately, a number of computation reductions can be made. The aim is to find the partial minima of the response, thus

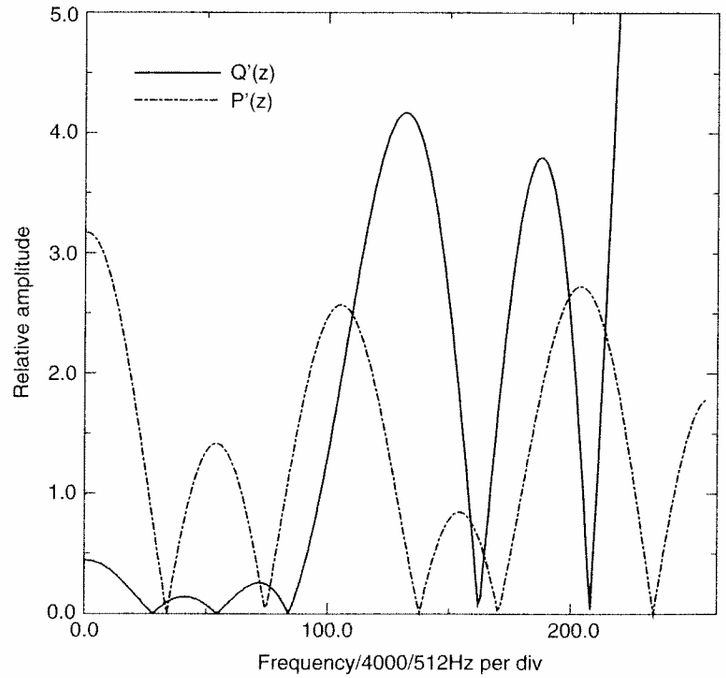

Figure 5.7 Zero frequency plot for one frame of the DFT–LSF method

the absolute values of the response are not critical; only the locations of the minima are vital. The spectrum is given by equation (5.35) where **P** is the spectrum, **W** is the $L \times L$ DFT kernel, and **S** is the input sequence. L is the size of the transform.

$$\begin{bmatrix} - \\ \mathbf{P} \\ - \end{bmatrix} = \begin{bmatrix} - & - & - \\ - & \mathbf{W} & - \\ - & - & - \end{bmatrix} \begin{bmatrix} - \\ \mathbf{S} \\ - \end{bmatrix} \tag{5.35}$$

As the input sequences A_k and B_k are real, we can move them from the start to the middle of **S** with zeros elsewhere. This will produce an even spectrum which means that only $f_s/2$ terms need to be computed. Also, the spectrum will be real, thus only the cosine-terms in the kernel require computing. Since the sequences A_k and B_k are even, only half of the values need to be computed, i.e. A_0 to $A_{p/2-1}$ and $1/2A_{p/2}$, and similarly for B_k. With these savings the number of multiply–adds is reduced to $p/2 + 1$ per spectrum point. The cosine terms are fixed for a particular transform size, therefore they can be pre-computed and stored in a lookup table.

Once the spectrum is found the partial minima need to be located and this involves computationally expensive comparisons. As the LSF are naturally ordered, i.e. the frequencies alternate between $Q(z)$ and $P(z)$, they can be located in an efficient manner. The first $Q(z)$ LSF starts at the origin, then the first $P(z)$ LSF starts from the previous $Q(z)$ LSF location. Once the first $P(z)$ LSF is found the second $Q(z)$ LSF is located, starting from the previous $P(z)$ location. This alternation is repeated until all LSFs are found. Thus in total only one pass of the frequency range is made instead of two.

5.3.4 Chebyshev Series Method

Another step-wise method which requires no prior storage or calculation of trigonometric functions is the Chebyshev Series Method [6]. By expanding equation (5.24) with the Chebyshev polynomial set, the mapping $x = \cos(\omega)$ maps the upper semicircle in the z-plane to the real interval $[+1, -1]$. Therefore, all the roots x_i lie between -1 and $+1$, with the root corresponding to the lowest frequency LSF being the one nearest to $+1$. Thus the basic task is similar to the DFT method, i.e. we isolate the roots of $P'(z)$ and $Q'(z)$ by searching incrementally for intervals in which the sign changes which is refined by successive bisections of the root interval.

5.3.5 Adaptive Sequential LMS Method

All of the previously described methods for deriving the LSF parameters required the intermediate step of calculating the LPC before proceeding to the computation of the LSF parameters. However, using a Least Mean Squares adaptive method [7] the LSF parameters can be computed directly from the speech samples themselves. The LMS algorithm aims to minimize the mean-square value of the PARCOR lattice filter output, and thus flatten its frequency spectrum by a 'noisy steepest-descent' procedure which uses the squared value of a single output sample to approximate the mean-square value. Thus the algorithm begins the sequential estimation using evenly-distributed estimated LSFs and, as each sample of speech is processed, a new LSF vector estimate is obtained. Depending on the adaptation rate required, the algorithm converges to the correct value after around 100 samples of input.

The LMS method is very attractive because it requires no LPC analysis. However, as it is a 'learning' type algorithm, it is susceptible to 'out-lier' input samples, i.e. samples which are different in character to the majority of speech samples. The effect of these unusual inputs is to throw the algorithm off its convergence curve; if this occurs at the end of a frame there will be no time for correction before the final values are used.

5.4 LSF to LPC Transformation

There are two methods for the inverse transformation, neither of which is as computationally intensive as the forward transformation. The two methods are equivalent but the LPC synthesis method is perhaps more easily visualized.

5.4.1 Direct Expansion Method

In all of the LPC to LSF methods above the aim is to find the roots of equation (5.16), i.e. a_i and b_i. Having found these roots using any of the methods, the LPC, α_i, can be simply found by multiplying out the product terms of equation (5.16), i.e.

$$\begin{aligned} P_{p+1}(z) &= z^{-(p+1)}[P'(z)(1-z)] \qquad (5.36) \\ &= z^{-(p+1)}[(1-z)(z-r_0)(z-r_0^*)\ldots(z-r_{p/2})(z-r_{p/2}^*] \\ &= z^{-(p+1)}[(1-z)(z^2-2u_0z+t_0)\ldots(z^2-2u_{p/2}z+t_{p/2})] \\ &= S_0 + S_1z^{-1} + \ldots + S_pz^{-p} + S_{p+1}z^{-(p+1)} \qquad (5.37) \end{aligned}$$

Similarly,

$$Q_{p+1}(z) = T_0 + T_1z^{-1} + \ldots + T_pz^{-p} + T_{p+1}z^{-(p+1)} \tag{5.38}$$

where,

$$\begin{aligned} & r_i = u_i + jv_i \quad \text{and} \quad r_i^* = u_i - jv_i \\ \Rightarrow \quad & r_i + r_i^* = 2u_i \quad \text{and} \quad r_i \times r_i^* = u_i^2 + v_i^2 = t_i \end{aligned} \tag{5.39}$$

Equating the terms of equations (5.37) and (5.16),

$$S_0 = 1 \tag{5.40}$$

$$T_0 = 1 \tag{5.41}$$

$$S_{p+1} = -1 \tag{5.42}$$

$$T_{p+1} = 1 \tag{5.43}$$

$$\alpha_i = \frac{1}{2}(T_i + S_i) \tag{5.44}$$

$$\alpha_{p+1-i} = \frac{1}{2}(T_i - S_i) \tag{5.45}$$

$$\text{for } i = 1, \ldots, P/2$$

5.4.2 LPC Synthesis Filter Method

An LPC synthesis can be constructed directly using the LSF coefficients. The filter is derived from the following,

$$H(z) = 1/A_p(z) = 1/[1 + (A_p(z) - 1)] \tag{5.46}$$

$$= \frac{1}{1 + 1/2[(P_{p+1}(z) - 1) + (Q_{p+1}(z) - 1)]}$$

i.e.

$$A_p(z) - 1 = 1/2[(P_{p+1}(z) - 1) + (Q_{p+1}(z) - 1)] \tag{5.47}$$

$$= 1/2 \left\{ (1 - z) \prod_{i=1}^{p/2} (1 - 2\cos\omega_i z + z^2) - 1 \right.$$

$$\left. + (1 + z) \prod_{i=1}^{p/2} (1 - 2\cos\theta_i z + z^2) - 1 \right\} \tag{5.48}$$

Let $u_i = -2\cos\omega_i$, $v_i = -2\cos\theta_i$

where w_i and θ_i are the even and odd number LSFs given by $LSF(i)2\pi T$.

$$A_p(z) - 1 = 1/2 \left\{ \prod_{i=1}^{p/2} (1 + u_i z + z^2) \right. \tag{5.49}$$

$$\left. - z \prod_{i=1}^{p/2} (1 + u_i z + z^2) - 1 \right\}$$

$$+ 1/2 \left\{ \prod_{i=1}^{p/2} (1 + v_i z + z^2) \right.$$

$$\left. - z \prod_{i=1}^{p/2} (1 + v_i z + z^2) - 1 \right\} \tag{5.50}$$

$$= z/2 \left\{ (u_1 + z) - \prod_{j=1}^{p/2} (1 + u_j z + z^2) \right.$$

$$\left. + \sum_{i=1}^{p/2-1} (u_{i+1} + z) \prod_{j=1}^{i} (1 + u_j z + z^2) \right\}$$

$$+ z/2 \left\{ (v_1 + z) - \prod_{j=1}^{p/2} (1 + v_j z + z^2) \right.$$

$$\left. + \sum_{i=1}^{p/2-1} (v_{i+1} + z) \prod_{j=1}^{i} (1 + v_j z + z^2) \right\} \tag{5.51}$$

An 8^{th} order inverse filter is shown in Figure 5.8. The LPC are simply the impulse response of the filter.

5.5 Properties of LSFs

A very important LSF property, as mentioned earlier, is the natural ordering of its parameters. This ordering property was already used to good effect in speeding up the LPC to LSF transformation procedure. The ordering property indicates that the LSFs within a frame, and from frame to frame, are correlated. In order to illustrate the intra-frame correlation property of the

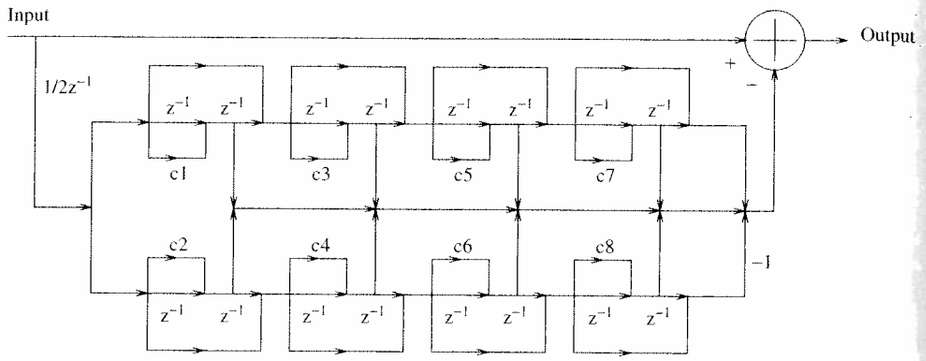

Figure 5.8 Practical scheme of LSF inverse filter ($c_i = -2\cos\omega_i$, for even i, and $c_i = -2\cos\theta_i$, for odd i)

Table 5.1 Experimental conditions for estimating Ω and Ψ

Sampling Frequency	8 kHz
Frame Update	10 ms
Window	20 ms Hamming
Analysis order	10
Number of Frames	6000

LSF vector, ω, Table 5.2 presents the matrix $\mathbf{\Omega} = \{\phi_{i,j}\}$ where,

$$\phi_{i,j} = \omega_{n,i} \times \omega_{n,j}, \quad i = 1, 2, \ldots, p, \; j = 1, 2, \ldots, p \tag{5.52}$$

for the experimental conditions according to Table 5.1. The relatively high correlation between neighbouring LSFs is clear. Similarly, to illustrate the inter-frame correlation of the LSF parameters, Table 5.3 presents the matrix $\mathbf{\Psi} = \{\phi_{i,k}\}$ where,

$$\phi_{i,k} = \omega_{n,i} \times \omega_{n-k,i}, \quad i = 1, 2, \ldots, p, \; k = 1, 2, \ldots, p \tag{5.53}$$

From Tables 5.2 and 5.3, it is clear that there is a strong correlation between the LSFs of adjacent frames as well as neighbouring parameters in the same frame. Therefore, any compression algorithm that effectively makes use of these correlations can result in improved performance over those that do not incorporate this correlation property.

Table 5.2 Intra-frame correlation coefficients Ω

	j									
i	1	2	3	4	5	6	7	8	9	10
1	1.00	0.65	−0.30	−0.35	−0.41	−0.49	−0.39	−0.40	−0.36	−0.20
2	0.65	1.00	0.28	0.11	−0.07	−0.13	−0.07	−0.05	−0.06	−0.07
3	−0.30	0.28	1.00	0.72	0.50	0.53	0.46	0.54	0.39	0.28
4	−0.35	0.11	0.72	1.00	0.72	0.62	0.46	0.42	0.45	0.21
5	−0.41	−0.07	0.50	0.72	1.00	0.79	0.52	0.47	0.34	0.26
6	−0.49	−0.13	0.53	0.62	0.79	1.00	0.71	0.61	0.49	0.28
7	−0.39	−0.07	0.46	0.46	0.52	0.71	1.00	0.73	0.58	0.41
8	−0.40	−0.05	0.54	0.42	0.47	0.61	0.73	1.00	0.58	0.46
9	−0.36	−0.06	0.39	0.45	0.34	0.49	0.58	0.58	1.00	0.41
10	−0.20	−0.07	0.28	0.21	0.26	0.28	0.41	0.46	0.41	1.00

Table 5.3 Inter-frame correlation coefficients Ψ

	k									
i	1	2	3	4	5	6	7	8	9	10
1	0.93	0.84	0.76	0.68	0.61	0.55	0.50	0.45	0.41	0.36
2	0.89	0.75	0.63	0.54	0.46	0.38	0.32	0.27	0.22	0.18
3	0.92	0.80	0.70	0.60	0.51	0.43	0.36	0.30	0.24	0.20
4	0.92	0.82	0.73	0.64	0.56	0.49	0.43	0.37	0.32	0.27
5	0.95	0.88	0.81	0.74	0.67	0.61	0.54	0.48	0.43	0.37
6	0.94	0.85	0.77	0.69	0.62	0.56	0.49	0.44	0.38	0.33
7	0.93	0.83	0.75	0.66	0.58	0.50	0.43	0.37	0.31	0.26
8	0.91	0.81	0.72	0.64	0.56	0.49	0.43	0.37	0.32	0.28
9	0.87	0.73	0.64	0.55	0.48	0.42	0.37	0.33	0.29	0.25
10	0.82	0.66	0.57	0.50	0.44	0.38	0.34	0.30	0.27	0.24

5.6 LSF Quantization

Most modern speech coders make use of LPC modelling during speech processing. Although some coders use a backward-adaptive LPC filter [8], most speech coders extract the LPC parameters from the input speech at regular intervals, transform them into the LSF domain, and quantize them for transmission to the decoder.

Low distortion LSF quantization is essential for the overall quality of decoded speech, and the number of bits allocated to LSFs usually takes a significant proportion of the overall bit rate, up to over 50% for very low

bit-rate speech coders. Therefore the overall success of a given speech coding scheme depends greatly on the quality of the LSF quantizer used.

Scalar schemes can be used, as they present very low complexity and storage requirements. However they cannot make use of the high intra-frame correlation exhibited by LSF vectors and, hence, they are very rarely used due to their poor performance. Vector quantization (VQ) schemes can be used to exploit intra-frame correlations. VQ exploits the redundancies in the LSF vector well and can provide high quality quantization for a relatively limited number of bits per frame of speech. As a result, they are widely used in modern speech coders. The following sections investigate the use of VQ for LSF quantization and ways of maximizing the performance of such schemes in several coder configurations.

5.6.1 Distortion Measures

In order to achieve good performance quantization of LSF parameters, it is necessary to have a way of linking the quantization error to the distortion in perceptual quality. Due to the complex relationship that exists between a set of LSF coefficients and the frequency response of the corresponding LPC filter, using a Mean-Square Error (MSE) measurement may not lead to an optimal performance of the quantizer.

A widely-used technique for computing the distortion that exists between the original set of LSFs and their quantized version is the Log Spectral Distortion measure. However a Weighted Mean-Square Error (WMSE) measurement may also lead to good results if an appropriate weighting function is used.

5.6.2 Spectral Distortion

The mean square log spectral distortion, which will be referred to simply as spectral distortion (SD), is defined as:

$$sd = \frac{1}{\pi}\int_0^{\pi} [10\ log_{10}S(w) - 10\ log_{10}S'(w)]^2 \tag{5.54}$$

where $S(w)$ and $S'(w)$ are the frequency responses of the LPC filter derived from the original and quantized LSFs, respectively. $S(w)$ can therefore be defined as:

$$S(w) = 1/\mid A(w) \mid^2 \tag{5.55}$$

which leads to,

$$S(w) = 1/\mid 1 - \sum_{k=1}^{p} a_k e^{-jwk} \mid^2 \tag{5.56}$$

where a_k are the LPC coefficients. This can be evaluated using an N-point Fourier Transform, giving the following expression:

$$SD = \frac{1}{N/2}\sum_{k=0}^{N/2-1} \left[10\ log_{10}\mid A'(k)\mid^2 - 10\ log_{10}\mid A(k)\mid^2\right]^2 \tag{5.57}$$

Moreover, it is common practice to restrict the computation of the distortion to a limited portion of the spectrum, typically the 125–3100 Hz band. The reason is that the portions of the spectrum below 125 Hz and above 3100 Hz usually have perceptually little impact but may significantly affect the computed spectral distortion, due to the use of the *log* function.

5.6.3 Average Spectral Distortion and Outliers

The spectral distortion (SD) measure gives a good indication of the perceptual difference between two sets of LSFs. The overall distortion caused by a quantization scheme can be computed by simply averaging the SD obtained over a large sequence of LSF vectors. It is commonly accepted that an average SD below 1 dB is necessary for an LSF quantizer to be transparent, i.e. not to add any audible distortion to synthesized speech. However, the average SD (*aveSD*) is not sufficient to determine the performance of a quantizer. The human ear is very sensitive to occasional large quantization errors. Therefore it is also important that the number of times the quantizer gives a large distortion is kept to a minimum. It is customary to use the percentage of input vectors giving spectral distortions above 2 dB and 4 dB as a quality measure. These measures are referred to as outliers at 2 dB and 4 dB, respectively.

The set of requirements usually considered necessary to achieve good quality speech is [9]:

- Average spectral distortion less than 1 dB
- Fewer than 2 % outliers at 2 dB
- No outliers at 4 dB

These three parameters need to be considered when evaluating the performance of an LSF quantizer. However an optimization has to be carried out to achieve the best overall performance for a given bit rate, i.e. accepting a larger average spectral distortion in return for fewer outliers.

5.6.4 MSE Weighting Techniques

Although spectral distortion is a fairly accurate representation of how quantization noise in the LSF is perceived, its high computational complexity limits its use. In order to compare two sets of LSFs, two fairly large fast Fourier Transforms (FFT) need to be computed and a logarithm must then be

Table 5.7 MSVQ and SVQ structures for Figure 5.11

Stages	MSVQ bit allocation	SVQ	
		Bit allocation	Vector split
2	12,12	12,12	5,5
3	8,8,8	8,8,8	3,3,4
4	6,6,6,6	6,6,6,6	3,2,2,3
5	5,5,5,5,4	5,5,5,5,4	2,2,2,2,2

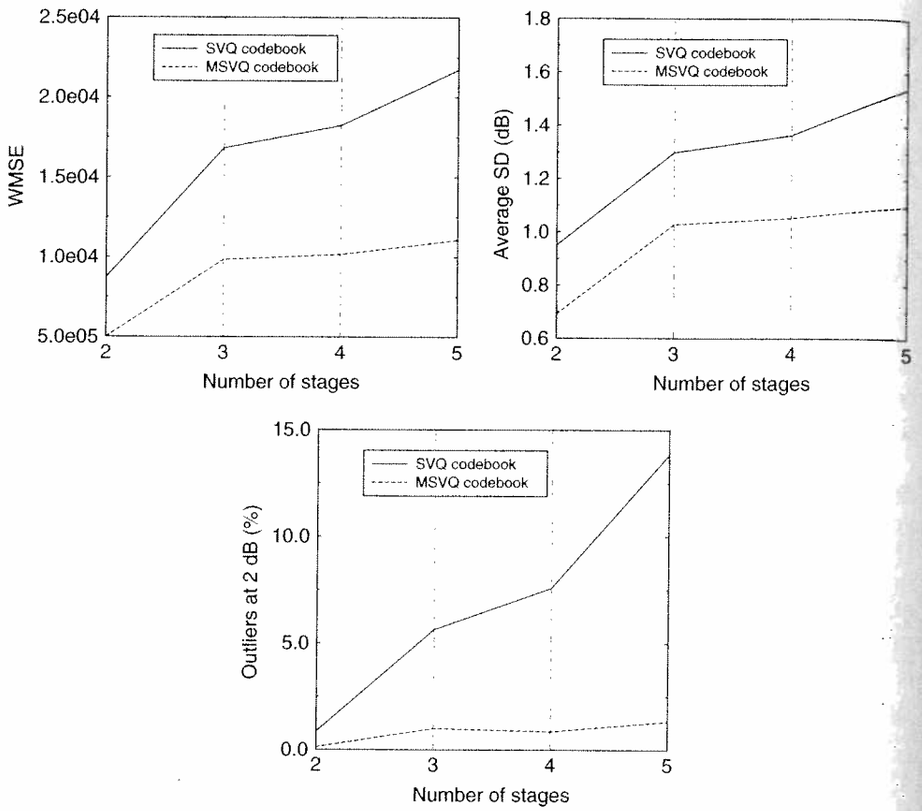

Figure 5.11 Performance comparison of various SVQ and MSVQ codebook structures

in Figure 5.12. Outliers at 4 dB have not been plotted, as they are zero for all cases. The advantage of a TS over both SS and FS is evident in these graphs. For M greater than or equal to eight, the performance of the TS is very close to that of the FS, at a much reduced complexity. It is also significantly better than that of SS, for a relatively small increase in complexity. The complexity

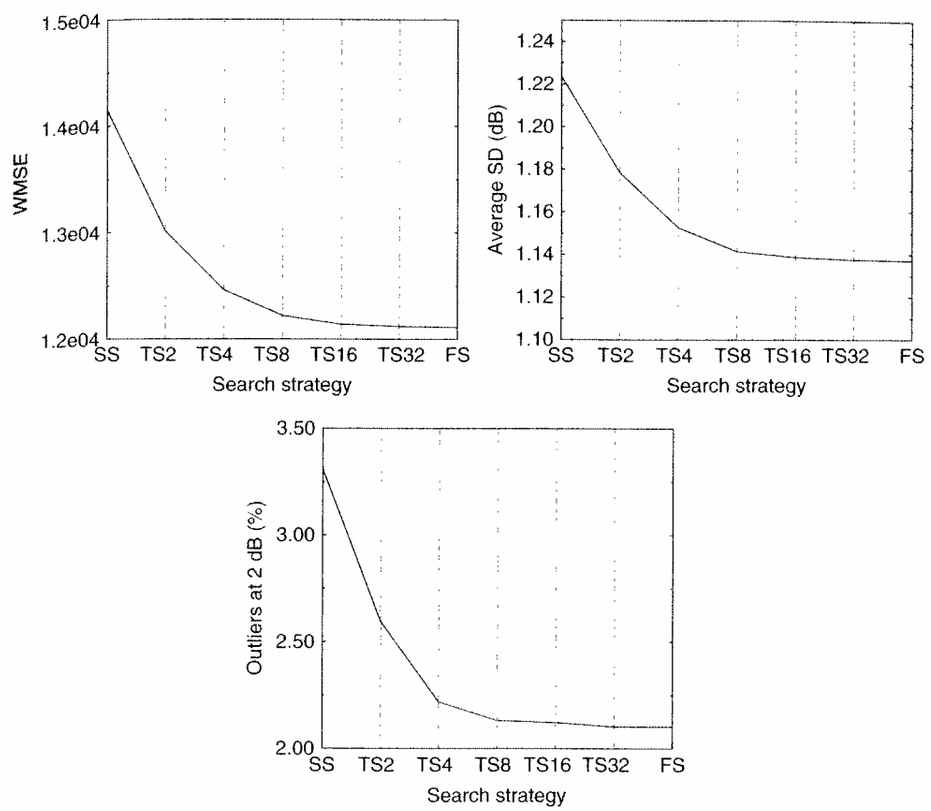

Figure 5.12 Performance comparison of various search techniques

in multiply–adds per input vector is given in Table 5.8. It is to be noted that, in the test, codebooks have been trained using the SS algorithm. Therefore, they are only optimal for an SS search. Better performance for the TS and FS cases can be obtained by using the same search in the training as the one used during the operation of the quantizer. This is illustrated in Figure 5.13, where WMSE, average SD and outliers at 2 dB are plotted for the original codebook and the retrained codebooks, for SS and TS with values of M ranging from 2 to 32. Due to the very high complexity of the FS, it was not possible to fully retrain the codebook using FS, although the results are expected to be similar to that of TS with $M = 32$.

5.8.3 Perceptual Weighting Techniques

Several weighting techniques were described in Section 5.6.4. A good weighting technique should give a distortion measure which is well correlated with the spectral distortion measure, which is our reference here. For testing, we