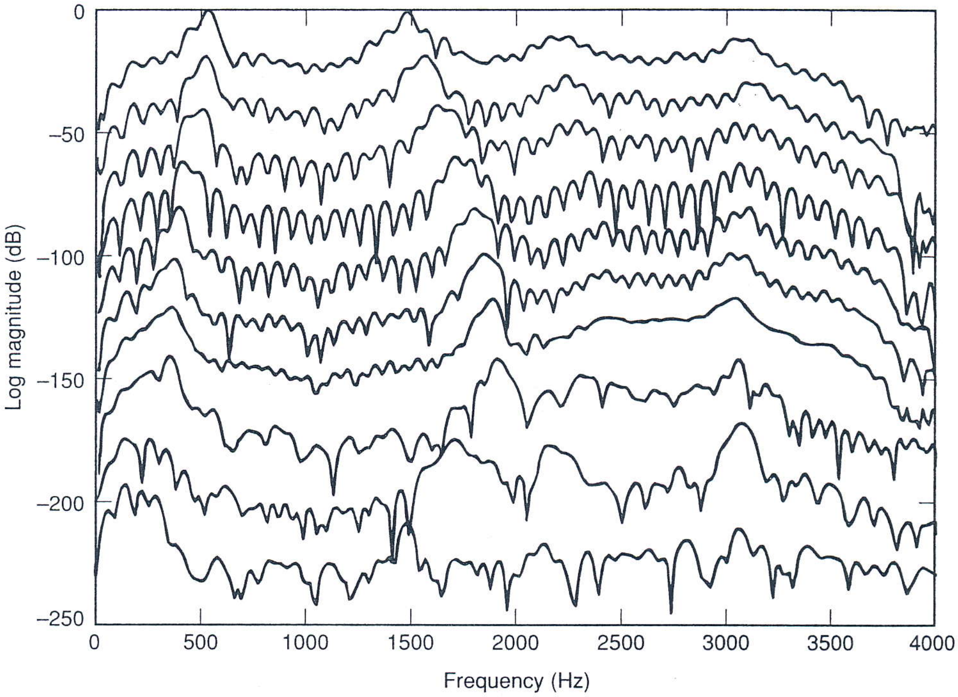
length. You may wish to use the M-file of Exercise 3.1 as the basis for your program. Only a few simple modifications should be necessary. Figure 10.4 shows an example of how your output should look.

Test·your program for the three cases `nstart = 3750, 16100, 17200` as in Exercise 3.1. Use values of `nsect = 10`, `ninc = 200`, `nwin = 401`, and `nfft = 512`. Can you see how the formant frequencies vary with time for the voiced segments?

Also try your program on the preemphasized speech and note again the effect of the preemphasis filter.

Figure 10.4
Short-time spectrum: 201-point window, 200 samples between segments.



SPEECH MODELING

The basis for most digital speech processing algorithms is a discrete-time system model for the production of samples of the speech waveform. Many useful models have been used as the basis for speech synthesis, speech coding, and speech recognition algorithms. The purpose of this set of projects is examine some of the details of the model depicted in Fig. 10.5.

PROJECT 1: GLOTTAL PULSE MODELS

The model of Fig. 10.5 is the basis for thinking about the speech waveform, and in some cases such a system is used explicitly as an speech synthesizer. In speech production, the excitation for voiced speech is a result of the quasi-periodic opening and closing of the opening between the vocal cords (the glottis). This is modeled in Fig. 10.5 by a combination of the impulse train generator and the glottal pulse model filter. The shape of the pulse affects the magnitude and phase of the spectrum of the synthetic speech output of the model. In this project we study the part labeled "Glottal Pulse Model $G(z)$" in Fig. 10.5.

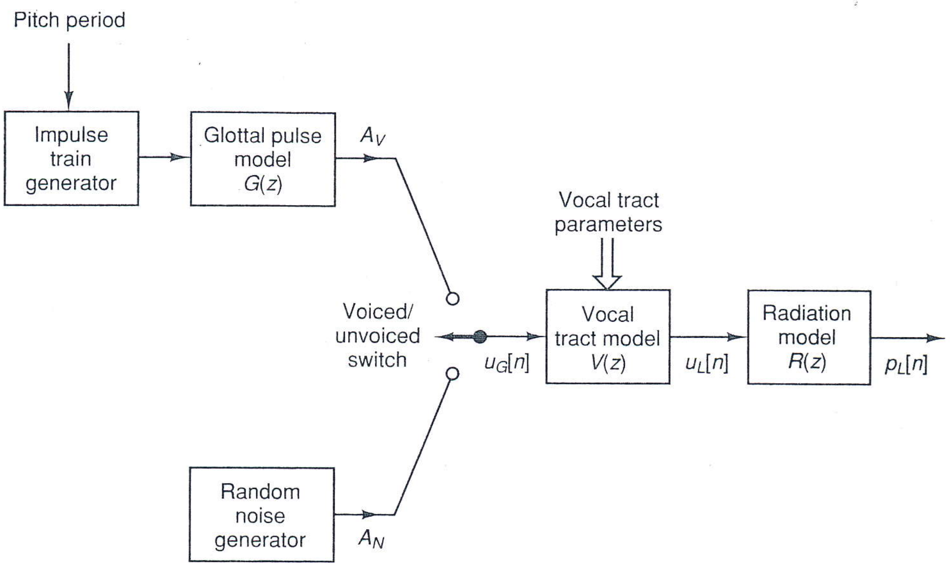


Figure 10.5
Discrete-time system model for speech production.

EXERCISE 1.1

Exponential Model

A simple model that we will call the *exponential model* is represented by

$$G(z) = \frac{-ae\ln(a)\,z^{-1}}{(1-az^{-1})^2} \tag{1-1}$$

where $e = 2.71828\ldots$ is the natural log base. Determine an analytical expression for $g[n]$, the inverse z-transform of $G(z)$. [The numerator of (1-1) is chosen so that $g[n]$ has maximum value of approximately 1.) Write an M-file to generate `Npts` samples of the corresponding glottal pulse waveform $g[n]$ and compute the frequency response of the glottal pulse model. The calling sequence for this function should be

```
[gE,GE,W]=glottalE(a,Npts,Nfreq)
```

where `gE` is the exponential glottal waveform vector of length `Npts`, and `GE` is the frequency response of the exponential glottal model at the `Nfreq` frequencies `W` between 0 and π radians. You will use this function later.

EXERCISE 1.2

Rosenberg Model

Rosenberg [9] used inverse filtering to extract the glottal waveform from speech. Based on his experimental results, he devised a model for use in speech synthesis, which is given by the equation

$$g_R[n] = \begin{cases} \frac{1}{2}[1-\cos(\pi n/N_1)] & 0 \le n \le N_1 \\ \cos[\pi(n-N_1)/(2N_2)] & N_1 \le n \le N_1+N_2 \\ 0 & \text{otherwise} \end{cases} \tag{1-2}$$

This model incorporates most of the important features of the time waveform of glottal waves estimated by inverse filtering and by high-speed motion pictures [3, 9].

Write an M-file to compute all N_1+N_2+1 samples of a Rosenberg glottal pulse with parameters N_1 and N_2 and to compute the frequency response of the Rosenberg glottal pulse model. The calling sequence for this function should be

```
[gR,GR,W]=glottalR(N1,N2,Nfreq)
```

where `gR` is the Rosenberg glottal waveform vector of length `N1+N2+1`, and `GR` is the frequency response of the glottal model at the `Nfreq` frequencies `W` between 0 and π radians.

EXERCISE 1.3

Comparison of Glottal Pulse Models

In this exercise you will compare three glottal pulse models.

a. First, use the M-files from Exercises 1.1 and 1.2 to compute `Npts=51` samples of the exponential glottal pulse `gE` for `a=0.91` and compute the Rosenberg pulse `gR` for the parameters `N1=40` and `N2=10`.

b. Also compute a new pulse `gRflip` by time-reversing `gR` using the MATLAB function `fliplr( )` for row vectors or `flipud( )` for column vectors. This has the effect of creating a new causal pulse of the form

$$g_{R\text{flip}}[n] = g_R[-(n - N_1 - N_2)] \qquad (1\text{-}3)$$

Determine the analytical relationship between $G_{R\text{flip}}(e^{j\omega})$, the Fourier transform of $g_{R\text{flip}}[n]$, and $G_R(e^{j\omega})$, the Fourier transform of $g_R[n]$.

c. Now plot all three of these 51-point vectors on the same graph using `plot( )`. Also plot the frequency response magnitude in dB for all three pulses on the same graph. Experiment with the parameters of the models to see how the time-domain wave shapes affect the frequency response.

d. Write an M-file to plot Rosenberg pulses for the three cases $N_2 = 10, 15, 25$ with $N_1 + N_2 = 50$ all on the same graph. Similarly, plot the Fourier transforms of these pulses together on another graph. What effect does the parameter N_2 have on the Fourier transform?

e. The exponential model has a zero at $z = 0$ and a double pole at $z = a$. For the parameters `N1=40` and `N2=10`, use the MATLAB function `roots( )` to find the zeros of the z-transform of the Rosenberg model and also the zeros of the flipped Rosenberg model. Plot them using the M-file `zplane( )`. Note that the Rosenberg model has all its zeros outside the unit circle (except one at $z = 0$). Such a system is called a *maximum-phase* system. The flipped Rosenberg model, however, should be found to have all its zeros inside the unit circle, and thus it is a *minimum-phase* system. Show that, in general, if a signal is maximum-phase, then flipping it as in (1-3) produces a minimum-phase signal, and vice versa.

PROJECT 2: LOSSLESS TUBE VOCAL TRACT MODELS

One approach to modeling sound transmission in the vocal tract is through the use of concatenated lossless acoustic tubes as depicted in Fig. 10.6.

Using the acoustic theory of speech production [3, 4, 10], it can be shown that the lossless assumption and the regular structure lead to simple wave equations and simple boundary conditions at the tube junctions, so that a solution for the transmission properties of the model is relatively straightforward and can be interpreted as in Fig. 10.7a, where $\tau = \Delta x/c$ is the one-way propagation delay of the sections. For sampled signals with

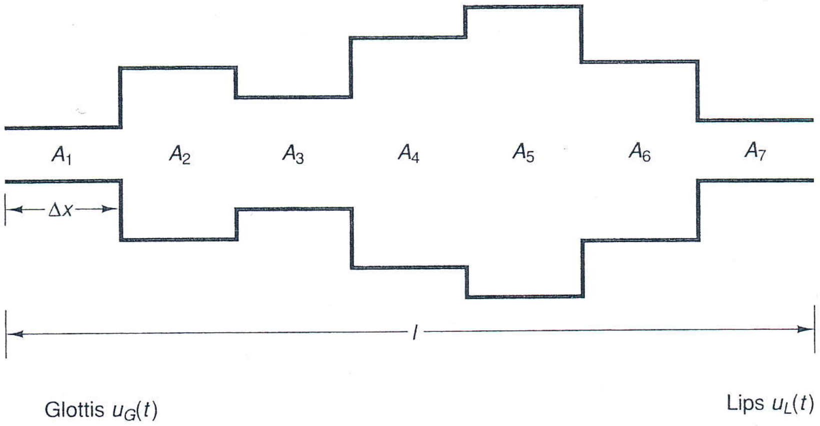


Figure 10.6 Concatenation of ($N = 7$) lossless acoustic tubes of equal length as a model of sound transmission in the vocal tract.

sampling period $T = 2\tau$, the structure of Fig. 10.7a (or equivalently Fig. 10.6) implies a corresponding discrete-time lattice filter [4] as shown in Fig. 10.7b or c.

Lossless tube models are useful for gaining insight into the acoustic theory of speech production, and they are also useful for implementing speech synthesis systems. It is shown in [4] that if $r_G = 1$, the discrete-time vocal tract model consisting of a concatenation of N lossless tubes of equal length has system function

$$V(z) = \frac{\prod_{k=1}^{N}(1 + r_k)z^{-N/2}}{D(z)} \tag{2-1}$$

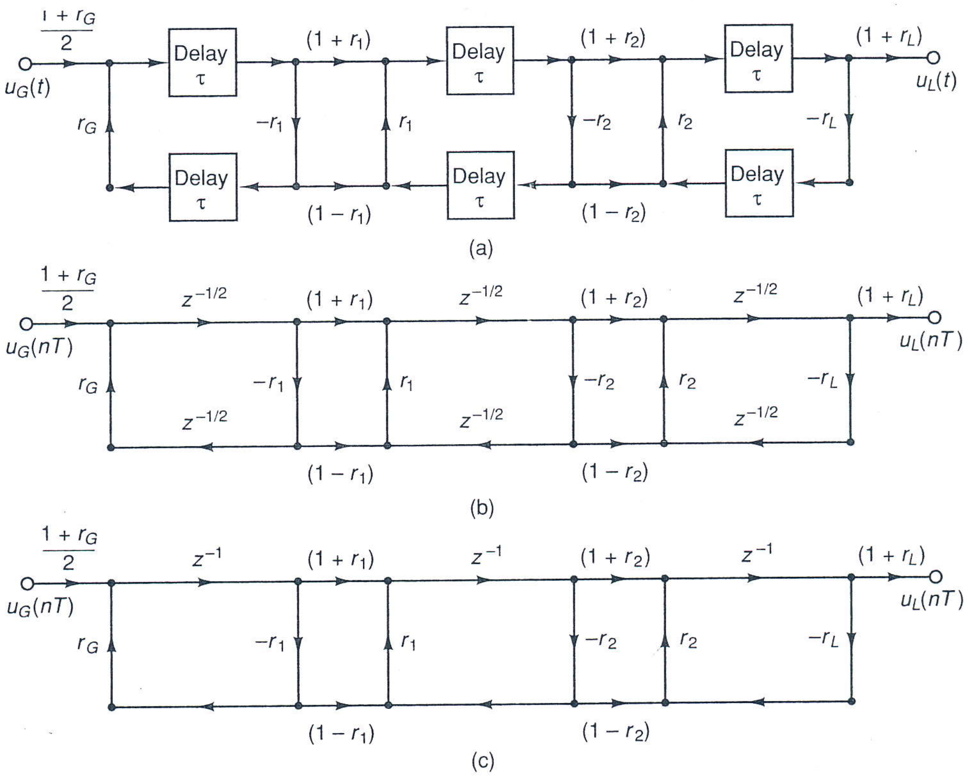


Figure 10.7 (a) Signal flow graph for lossless tube model ($N = 3$) of the vocal tract; (b) equivalent discrete-time system; (c) equivalent discrete-time system using only whole-sample delays in ladder part.

The denominator polynomial $D(z)$ in (2-1) satisfies the polynomial recursion [4]

$$\begin{aligned} D_0(z) &= 1 \\ D_k(z) &= D_{k-1}(z) + r_k z^{-k} D_{k-1}(z^{-1}) \qquad k = 1, 2, \dots, N \\ D(z) &= D_N(z) \end{aligned} \tag{2-2}$$

where the r_k's in (2-2) are the reflection coefficients at the tube junctions,

$$r_k = \frac{A_{k+1} - A_k}{A_{k+1} + A_k} \tag{2-3}$$

In deriving the recursion in (2-2), it was assumed that there were no losses at the glottal end ($r_G = 1$) and that all the losses are introduced at the lip end through the reflection coefficient

$$r_N = r_L = \frac{A_{N+1} - A_N}{A_{N+1} + A_N} \tag{2-4}$$

where A_{N+1} is the area of an impedance-matched (no reflections at its end) tube that can be chosen to introduce a loss in the system [4].

Suppose that we have a set of areas for a lossless tube model, and we wish to obtain the system function for the system so that we can use the MATLAB `filter( )` function to implement the model; that is, we want to obtain the system function of (2-1) in the form

$$V(z) = \frac{G}{D(z)} = \frac{G}{1 - \sum_{k=1}^{N} \alpha_k z^{-k}} \tag{2-5}$$

[Note that in (2-5) we have dropped the delay of $N/2$ samples, which is inconsequential for use in synthesis.] The following MATLAB M-file called `AtoV.m` implements (2-2) and (2-3); that is, it takes an array of tube areas and a reflection coefficient at the lip end and finds the parameters of (2-5) along with the reflection coefficients.

As test data for this project, the area functions shown in Table 10.4 were obtained by interpolating and resampling area function data for Russian vowels as given by Fant [10].

TABLE 10.4

Vocal Tract Area Data for Two Russian Vowels.

Section	1	2	3	4	5	6	7	8	9	10
vowel AA	1.6	2.6	0.65	1.6	2.6	4	6.5	8	7	5
vowel IY	2.6	8	10.5	10.5	8	4	0.65	0.65	1.3	3.2

```
function        [r,D,G]=AtoV(A,rN)
%       function to find reflection coefficients
%       and system function for
%       lossless tube models.
%               [r,D,G]=AtoV(A,rN)
%               rN = reflection coefficient at lips (abs value < 1)
%               A = array of areas
%               D = array of denominator coefficients
%               G = numerator of system function
%               r = corresponding reflection coefficients
%       assumes no losses at the glottis end (rG=1).
[M,N] = size(A);
if(M~=1)  A = A';  end      %-- make row vector
```

```
N = length(A);
r = [];
for m=1:N-1
     r = [r (A(m+1)-A(m))/(A(m+1)+A(m))];
end
r = [r rN];
D = [1];
G = 1;
for m=1:N
     G = G*(1+r(m));
     D = [D 0] + r(m).*[0 fliplr(D)];
end
```

EXERCISE 2.1

Frequency Response and Pole–Zero Plot

a. Use the M-file `AtoV( )` to obtain the denominator $D(z)$ of the vocal tract system function, and make plots of the frequency response for each area function for `rN=0.71` and also for the totally lossless case $rN = 1$. Plot the two frequency responses for a given vowel on the same plot.

b. Factor the polynomials $D(z)$ and plot the poles in the z-plane using `zplane( )`. Plot the roots of the lossy case as `o`'s and the roots of the lossless case as `x`'s. (See `help zplane` from Appendix A.) Where do the roots lie for the lossless case? How do the roots of $D(z)$ shift as `rN` decreases away from unity? Convert the angles of the roots to analog frequencies corresponding to a sampling rate of $1/T = 10,000$ samples/s, and compare to the formant frequencies expected for these vowels [3, 4, 10]. For this sampling rate, what is the effective length of the vocal tract, in centimeters?

EXERCISE 2.2

Finding the Model from the System Function

The inverse problem arises when we want to obtain the areas and reflection coefficients for a lossless tube model given the system function in the form of (2-5). We know that the denominator of the system function, $D(z)$, satisfies (2-2). In this part we use (2-2) to develop an algorithm for finding the reflection coefficients and the areas of a lossless tube model having a given system function denominator.

a. Show that r_N is equal to the coefficient of z^{-N} in the denominator of $V(z)$ (i.e., $r_N = -\alpha_N$).

b. Use (2-2) to show that

$$D_{k-1}(z) = \frac{D_k(z) - r_k z^{-k} D_k(z^{-1})}{1 - r_k^2} \qquad k = N, N-1, \ldots, 2$$

c. How would you use the results of parts (a) and (b) to find r_{N-1} from $D_N(z) = D(z)$?

d. Using the results of parts (a), (b), and (c), state an algorithm for finding all of the reflection coefficients r_k, $k = 1, 2, \ldots, N$ and all of the tube areas A_k, $k = 1, 2, \ldots, N$. Are the A_k's unique? Write a MATLAB function to implement your algorithm for converting from $D(z)$ to reflection coefficients and areas. This M-file should adhere to the following definition:

```
function        [r,A]=VtoA(D,A1)
%         function to find reflection coefficients
%         and tube areas for lossless tube models.
%          [r,A]=VtoA(D,A1)
%                 A1 = arbitrary area of first section
%                   D = array of denominator coefficients
%                   A = array of areas for lossless tube model
%                   r = corresponding reflection coefficients
%          assumes no losses at the glottis end (rG=1).
```

[This new M-file can be similar in structure to `AtoV( )`.] For the vowel /a/, the denominator of the 10th-order model should be (to four-digit accuracy)

$$\begin{aligned} D(z) &= 1 - 0.0460z^{-1} - 0.6232z^{-2} + 0.3814z^{-3} + 0.2443z^{-4} + 0.1973z^{-5} \\ &\quad + 0.2873z^{-6} + 0.3655z^{-7} - 0.4806z^{-8} - 0.1153z^{-9} + 0.7100z^{-10} \end{aligned}$$

Use your MATLAB program to find the corresponding reflection coefficients and tube areas and compare to the data for the vowel /a/ in Table 10.4. If your program is working, there may still be small differences between its output and the data of Table 10.4. Why?

PROJECT 3: VOWEL SYNTHESIS

For voiced speech, the speech model of Fig. 10.5 can be simplified to the system of Fig. 10.8. The excitation signal $e[n]$ is a quasi-periodic impulse train and the glottal pulse model could be either the exponential or the Rosenberg pulse. The vocal tract model could be a lattice filter of the form of Fig. 10.7c, or it could be an equivalent direct-form difference equation as implemented by MATLAB.

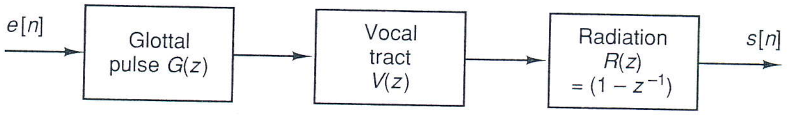


Figure 10.8 Simplified model for synthesizing voiced speech.

Hints

In this project we use the M-files written in Projects 1 and 2, together with the `filter( )` and `conv( )` functions to implement parts of the system of Fig. 10.8 and thereby synthesize periodic vowel sounds. A periodic pulse train can be synthesized by using the M-file `zerofill( )` from Appendix A, together with the MATLAB function `ones( )`.

EXERCISE 3.1

Periodic Vowel Synthesis

Assume a sampling rate of 10000 samples/s. Create a periodic impulse train vector `e` of length 1000 samples, with period corresponding to a fundamental frequency of 100 Hz. Then use combinations of `filter( )` and `conv( )` to implement the system of Fig. 10.8.

Use the excitation `e` and radiation system $R(z) = (1 - z^{-1})$ to synthesize speech for both area functions given above and for all three glottal pulses studied in Project 2. Use `subplot( )` and `plot( )` to make a plot comparing 1000 samples of the synthetic speech outputs for the exponential glottal pulse and the Rosenberg minimum-phase pulse. Make another plot comparing the outputs for the two Rosenberg pulses.

EXERCISE 3.2

Frequency Response of Vowel Synthesizer

Plot the frequency response (log magnitude in dB) of the overall system with system function $H(z) = G(z)V(z)R(z)$ for the case of the Rosenberg glottal pulse, $R(z) = (1 - z^{-1})$, and vocal tract response for the vowel /a/. Save your result for use in Exercise 3.3.

EXERCISE 3.3

Short-Time Fourier Transform of Synthetic Vowel

Compute the DFT of a Hamming-windowed segment (401 points) of the synthetic vowel and plot the log magnitude on the same graph as the frequency response of the synthesizer.

EXERCISE 3.4

Noise Excitation (Whispered Speech)

In producing whispered speech, the vocal tract is excited by turbulent airflow produced at the glottis. This can be modeled by exciting only the cascaded vocal tract and radiation filters with random noise. Using the function `randn( )`, excite the cascaded vocal tract/radiation filters for the vowel AA with a zero-mean Gaussian noise input. Plot the waveform and repeat Exercises 3.2 and 3.3 for the "whispered" vowel.

EXERCISE 3.5

Listening to the Output (Optional)

If D to A facilities are available on your computer, create files of synthetic voiced and whispered vowels of length corresponding to 0.5 s duration in the proper binary format, and play them out through the D to A system. For a 16-bit D to A converter you should scale the samples appropriately and use `round( )` to convert them to integers (of magnitude ≤ 32767) before writing the file. Does the synthetic speech sound like the desired vowels?

SPEECH QUANTIZATION

OVERVIEW

Sampling and quantization (or A-to-D conversion) of speech waveforms is important in digital speech processing because it is the first step in any digital speech processing system, and because one of the basic problems of speech processing is digital coding of the speech signal for digital transmission and/or storage. Sampling and quantization of signals is generally implemented by a system of the form of Fig. 10.9. In a hardware realization, the sample-and-hold circuit samples the input continuous-time signal and holds the value constant during the sampling period T. This gives a constant signal at the input of the A-to-D converter, whose purpose is to decide which of its quantization levels is closest to the input sample value. Every T seconds, the A-to-D converter emits a digital code corresponding to that level. Normally, the digital code is assigned according to a convenient binary number system such as two's-complement so that the binary numbers can be taken as numerical representations of the sample values.

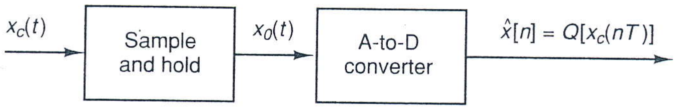


Figure 10.9 Representation of hardware for sampling and quantization of speech signals.

An equivalent representation of sampling and quantization is depicted in Fig. 10.10. This representation is convenient because it separates the sampling and quantization into two independent operations. The operation of the ideal sampler is well understood. The sampling theorem states that a bandlimited signal can be reconstructed precisely from samples taken at the rate of twice the highest frequency in the spectrum of the signal. In these projects it will be assumed that the speech signal has been low-pass filtered and