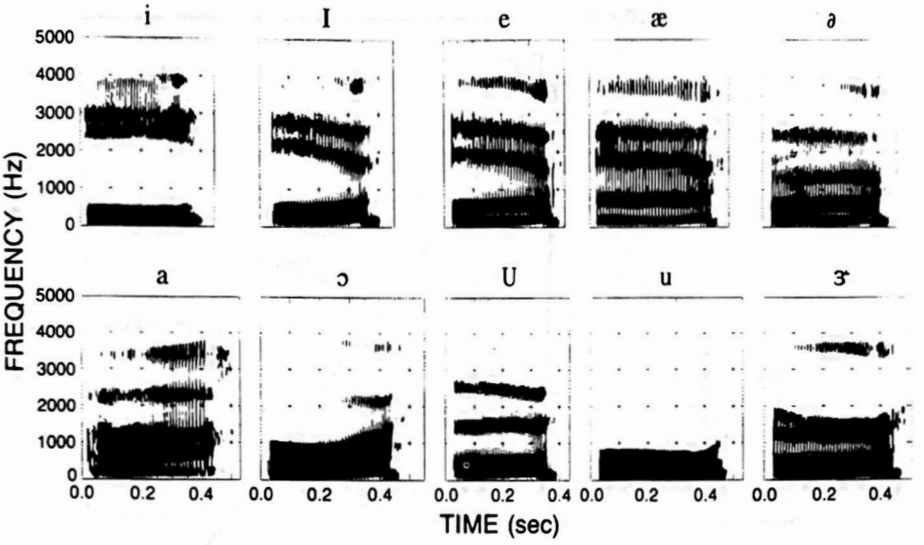


Figure 2.15 Spectrograms of the vowel sounds.

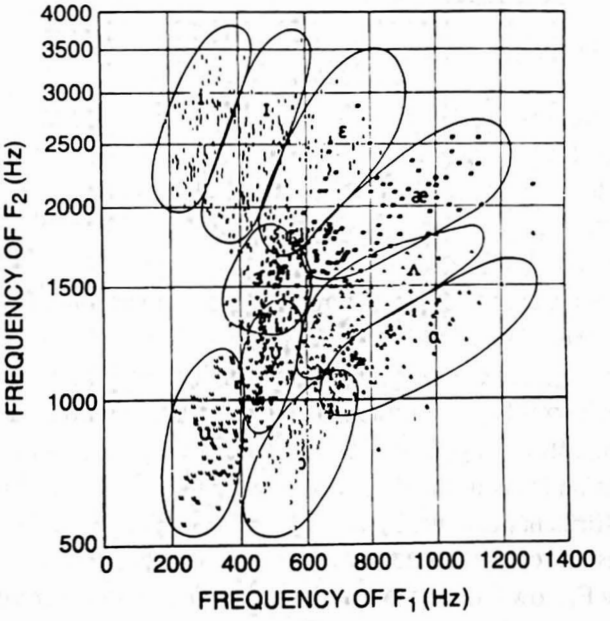


Figure 2.16 Measured frequencies of first and second formants for a wide range of talkers for several vowels (after Peterson & Barney [7]).

overlap between the formant frequencies for *different* vowel sounds by different talkers. The ellipses drawn in this figure represent gross characterizations of the regions in which most of the tokens of the different vowels lie. The message of Figure 2.16, for speech recognition by machine, is fairly clear; that is, it is not just a simple matter of measuring formant frequencies or spectral peaks accurately to accurately classify vowel sounds; one

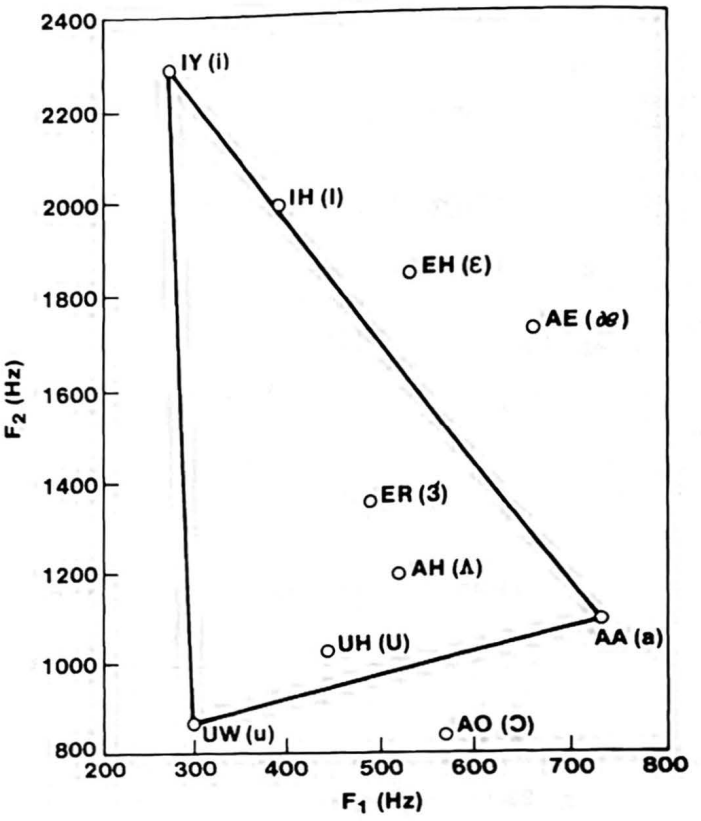


Figure 2.17 The vowel triangle with centroid positions of the common vowels.

must do some type of talker (accent) normalization to account for the variability in formants and overlap between vowels.

A common way of exploiting the information embodied in Figures 2.15 and 2.16 is to represent each vowel by a centroid in the formant space with the realization that the centroid, at best, represents average behavior and does not represent variability across talkers. Such a representation leads to the classic vowel triangle shown in Figure 2.17 and represented in terms of formant positions by the data given in Table 2.2. The vowel triangle represents the extremes of formant locations in the F_1-F_2 plane, as represented by /i/ (low F_1, high F_2), /u/ (low F_1, low F_2), and /a/ (high F_1, low F_2), with other vowels appropriately placed with respect to the triangle vertices. The utility of the formant frequencies of Table 2.2 has been demonstrated in text-to-speech synthesis in which high-quality vowel sounds have been synthesized using these positions for the resonances [8].

2.4.2 Diphthongs

Although there is some ambiguity and disagreement as to what is and what is not a diphthong, a reasonable definition is that a diphthong is a gliding monosyllabic speech

TABLE 2.2. Formant frequencies for typical vowels.

ARPABET Symbol for Vowel	IPA Symbol	Typical Word	F_1	F_2	F_3
IY	/i/	beet	270	2290	3010
IH	/ɪ/	bit	390	1990	2550
EH	/ɛ/	bet	530	1840	2480
AE	/æ/	bat	660	1720	2410
AH	/ʌ/	but	520	1190	2390
AA	/a/	hot	730	1090	2440
AO	/ɔ/	bought	570	840	2410
UH	/ʊ/	foot	440	1020	2240
UW	/u/	boot	300	870	2240
ER	/ɝ/	bird	490	1350	1690

sound that starts at or near the articulatory position for one vowel and moves to or toward the position for another. According to this definition, there are six diphthongs in American English, namely /a^y/ (as in buy), /a^w/ (as in down), /e^y/ (as in bait), and /$ɔ^y$/ (as in boy), /o/ (as in boat), and /ju/ (as in you).

The diphthongs are produced by varying the vocal tract smoothly between vowel configurations appropriate to the diphthong. Figure 2.18 shows spectrogram plots of four of the diphthongs spoken by a male talker. The gliding motions of the formants are especially prominent for the sounds /a^y/, /a^w/ and /$ɔ^y$/ and are somewhat weaker for /e^y/ because of the closeness (in vowel space) of the two vowel sounds comprising this diphthong.

An alternative way of displaying the time-varying spectral characteristics of diphthongs is via a plot of the values of the second formant versus the first formant (implicitly as a function of time) as shown in Figure 2.19 [9]. The arrows in this figure indicate the direction of motion of the formants (in the ($F_1 - F_2$) plane) as time increases. The dashed circles in this figure indicate average positions of the vowels. Based on these data, and other measurements, the diphthongs can be characterized by a time-varying vocal tract area function that varies between two vowel configurations.

2.4.3 Semivowels

The group of sounds consisting of /w/, /l/, /r/, and /y/ is quite difficult to characterize. These sounds are called semivowels because of their vowel-like nature. They are generally characterized by a gliding transition in vocal tract area function between adjacent phonemes. Thus the acoustic characteristics of these sounds are strongly influenced by the context in which they occur. For our purposes, they are best described as transitional, vowel-like sounds, and hence are similar in nature to the vowels and diphthongs.

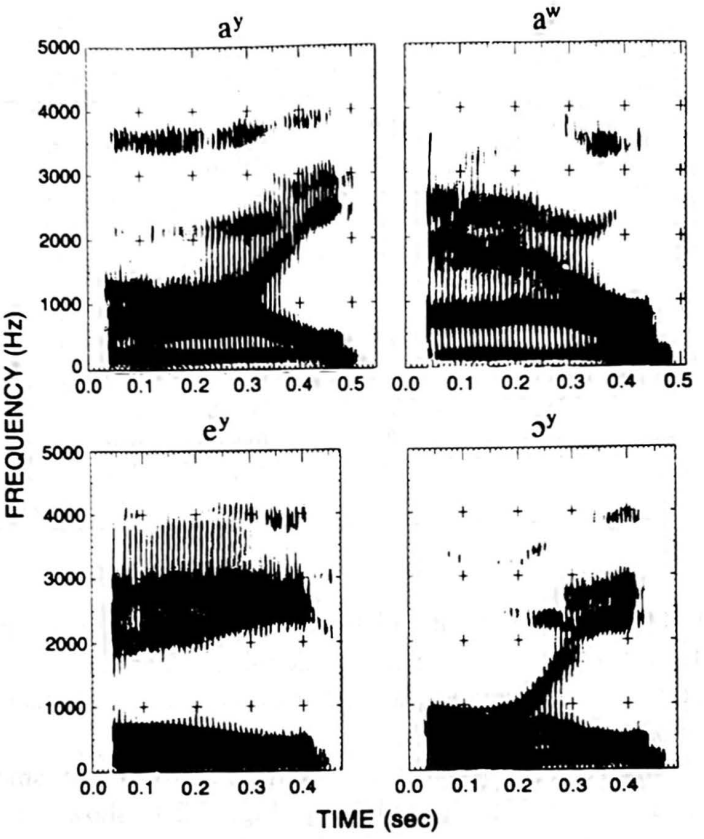


Figure 2.18 Spectrogram plots of four diphthongs.

2.4.4 Nasal Consonants

The nasal consonants /m/, /n/, and /η/ are produced with glottal excitation and the vocal tract totally constricted at some point along the oral passageway. The velum is lowered so that air flows through the nasal tract, with sound being radiated at the nostrils. The oral cavity, although constricted toward the front, is still acoustically coupled to the pharynx. Thus, the mouth serves as a resonant cavity that traps acoustic energy at certain natural frequencies. As far as the radiated sound is concerned, these resonant frequencies of the oral cavity appear as antiresonances, or zeros of the transfer function of sound transmission. Furthermore, nasal consonants and nasalized vowels (i.e., some vowels preceding or following nasal consonants) are characterized by resonances that are spectrally broader, or more highly damped, than those for vowels.

The three nasal consonants are distinguished by the place along the oral tract at which a total constriction is made. For /m/ the constriction is at the lips; for /n/ the constriction is just behind the teeth; and for /η/ the constriction is just forward of the velum itself. Figure 2.20 shows typical speech waveforms and Figure 2.21 spectrograms for two nasal consonants in the context vowel-nasal-vowel. The waveforms of /m/ and /n/ look very similar. The spectrograms show a concentration of low-frequency energy with a midrange of frequencies that contain no prominent peaks. This is because of the particular

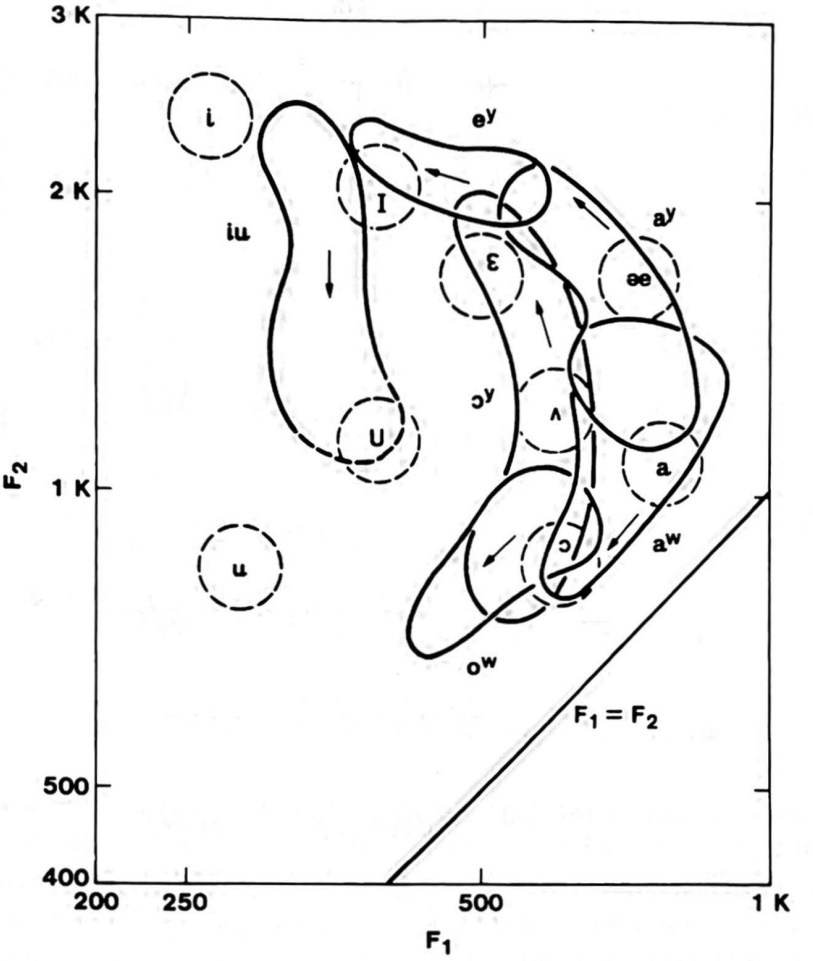


Figure 2.19 Time variation of the first two formants for the diphthongs (after Holbrook and Fairbanks [9]).

combination of resonances and antiresonances that result from the coupling of the nasal and oral tracts.

2.4.5 Unvoiced Fricatives

The unvoiced fricatives /f/, /θ/, /s/, and /sh/ are produced by exciting the vocal tract by a steady air flow, which becomes turbulent in the region of a constriction in the vocal tract. The location of the constriction serves to determine which fricative sound is produced. For the fricative /f/ the constriction is near the lips; for /θ/ it is near the teeth; for /s/ it is near the middle of the oral tract; and for /sh/ it is near the back of the oral tract. Thus the system for producing unvoiced fricatives consists of a source of noise at a constriction, which separates the vocal tract into two cavities. Sound is radiated from the lips—that is, from the front cavity. The back cavity serves, as in the case of nasals, to trap energy and thereby introduce antiresonances into the vocal output. Figure 2.22 shows the waveforms and Figure 2.23 the spectrograms of the fricatives /f/, /s/ and /sh/. The nonperiodic nature

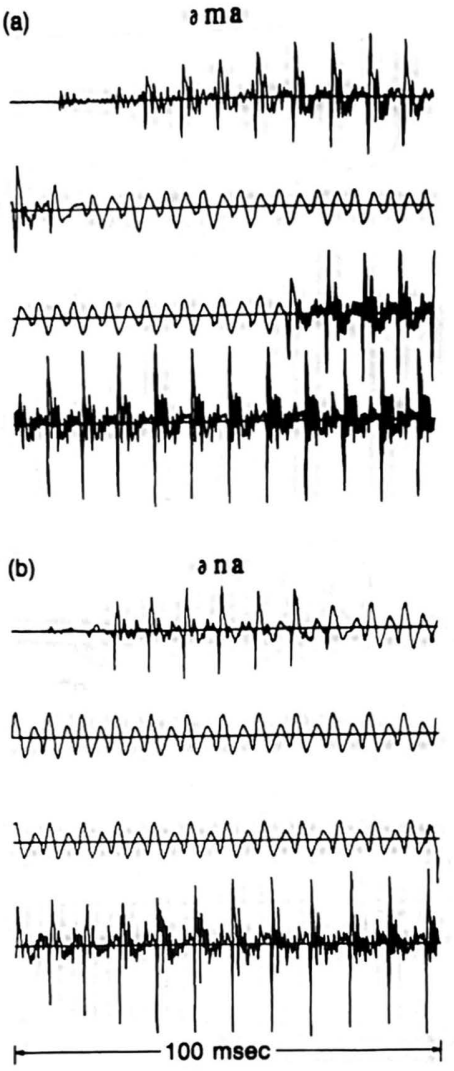


Figure 2.20 Waveforms for the sequences /ə-m-a/ and /ə-n-a/.

of fricative excitation is obvious in the waveform plots. The spectral differences among the fricatives are readily seen by comparing the three spectrograms.

2.4.6 Voiced Fricatives

The voiced fricatives /v/, /th/, /z/ and /zh/ are the counterparts of the unvoiced fricatives /f/, /θ/, /s/, and /sh/, respectively, in that the place of constriction for each of the corresponding phonemes is essentially identical. However, the voiced fricatives differ markedly from their unvoiced counterparts in that two excitation sources are involved in their production. For voiced fricatives the vocal cords are vibrating, and thus one excitation source is at the glottis. However, since the vocal tract is constricted at some point forward of the glottis, the air flow becomes turbulent in the neighborhood of the constriction. Thus the

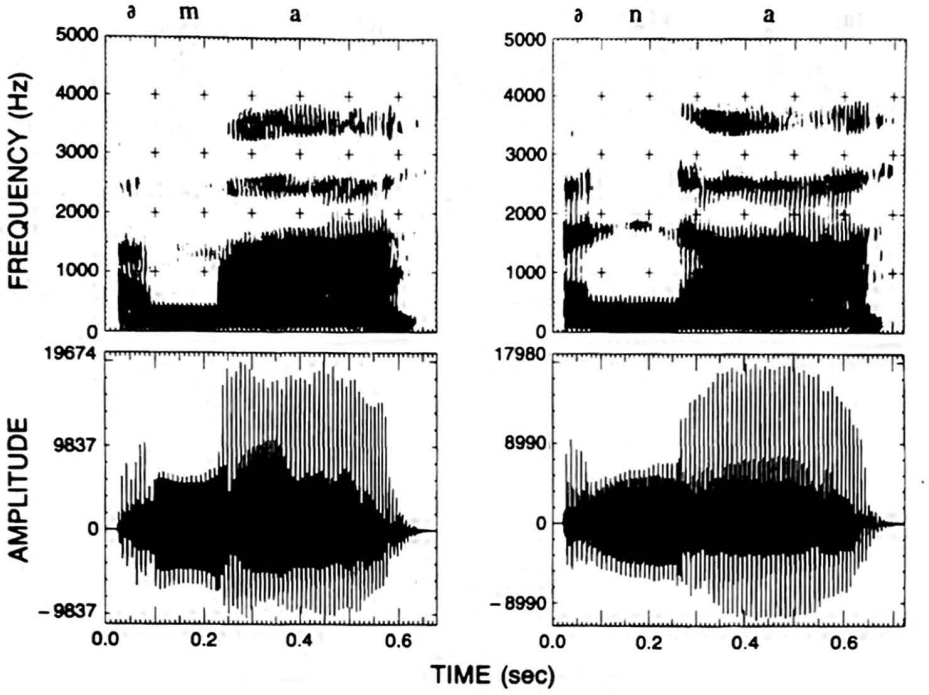


Figure 2.21 Spectrograms of the sequences /ə-m-a/ and ə-n-a/.

spectra of voiced fricatives can be expected to display two distinct components. These excitation features are readily observable in Figure 2.24, which shows typical waveforms, and in Figure 2.25, which shows spectra for two voiced fricatives. The similarity of the unvoiced fricative /f/ to the voiced fricative /v/ is easily shown in a comparison between corresponding spectrograms in Figures 2.23 and 2.25. Likewise, it is instructive to compare the spectrograms of /sh/ and /zh/.

2.4.7 Voiced and Unvoiced Stops

The voiced stop consonants /b/, /d/, and /g/, are transient, noncontinuant sounds produced by building up pressure behind a total constriction somewhere in the oral tract and then suddenly releasing the pressure. For /b/ the constriction is at the lips; for /d/ the constriction is at the back of the teeth; and for /g/ it is near the velum. During the period when there is total constriction in the tract, no sound is radiated from the lips. However, there is often a small amount of low-frequency energy radiated through the walls of the throat (sometimes called a voice bar). This occurs when the vocal cords are able to vibrate even though the vocal tract is closed at some point.

Since the stop sounds are dynamical in nature, their properties are highly influenced by the vowel that follows the stop consonant. As such, the waveforms for stop consonants give little information about the particular stop consonant. Figure 2.26 shows the waveform of the syllable /ə-b-a/. The waveform of /b/ shows few distinguishing features except for the voiced excitation and lack of high-frequency energy.

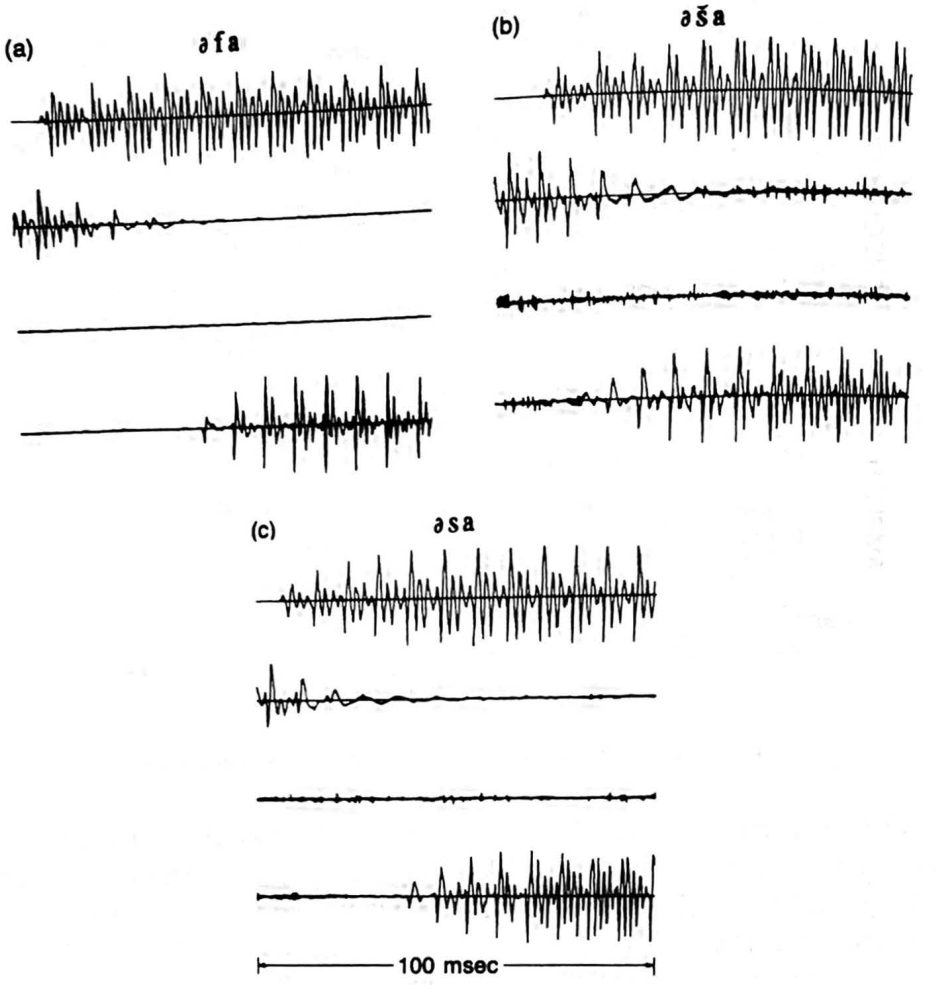


Figure 2.22 Waveforms for the sounds /f/, /s/ and /sh/ in the context /ə-x-a/ where /x/ is the unvoiced fricative.

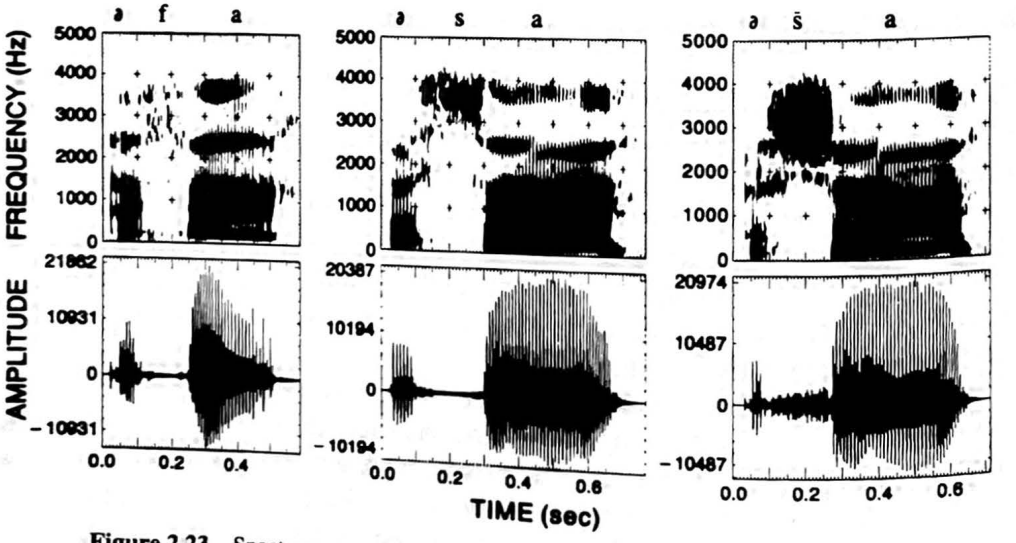


Figure 2.23 Spectrogram comparisons of the sounds /ə-f-a/, /ə-s-a/ and /ə-sh-a/.

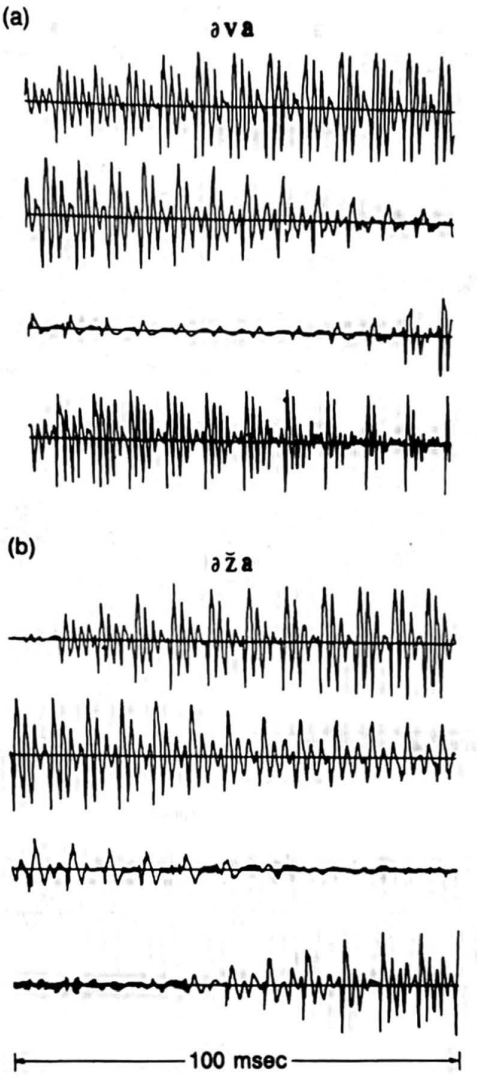


Figure 2.24 Waveforms for the sequences /ə-v-a/ and /ə-zh-a/.

The unvoiced stop consonants /p/, /t/, and /k/ are similar to their voiced counterparts /b/, /d/, and /g/, with one major exception. During the period of total closure of the tract, as the pressure builds up, the vocal cords do not vibrate. Then, following the period of closure, as the air pressure is released, there is a brief interval of friction (due to sudden turbulence of the escaping air) followed by a period of aspiration (steady air flow from the glottis exciting the resonances of the vocal tract) before voiced excitation begins.

Figure 2.27 shows waveforms and Figure 2.28 shows spectrograms of the voiced stop /b/ and the voiceless stop consonants /p/ and /t/. The "stop gap," or time interval, during which the pressure is built up is clearly in evidence. Also, it can be readily seen that the duration and frequency content of the frication noise and aspiration vary greatly with the stop consonant.

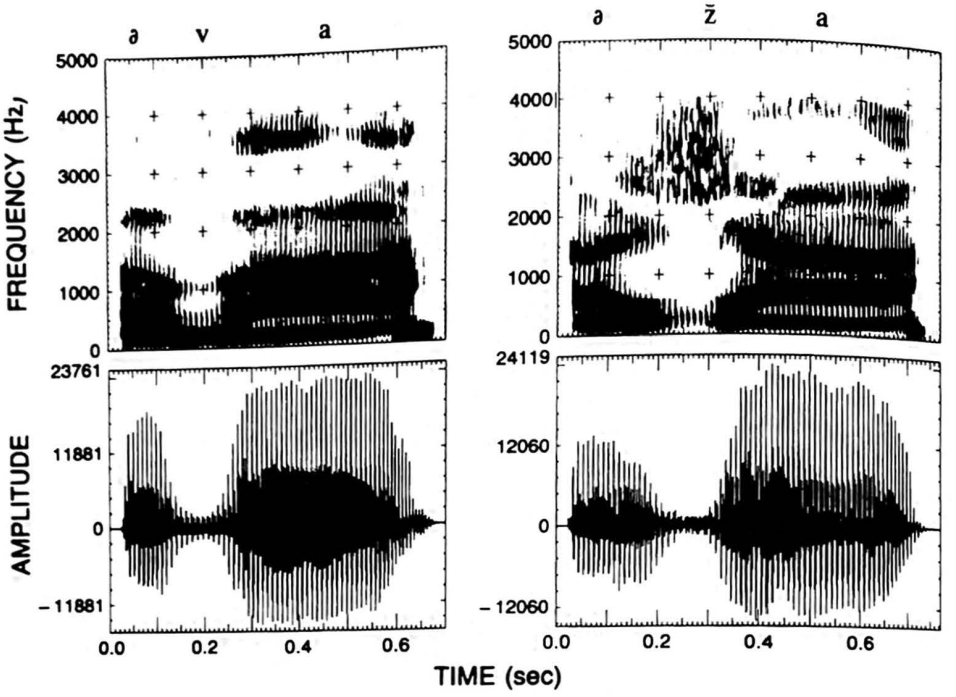


Figure 2.25 Spectrograms for the sequences /ə-v-a/ and /ə-zh-a/.

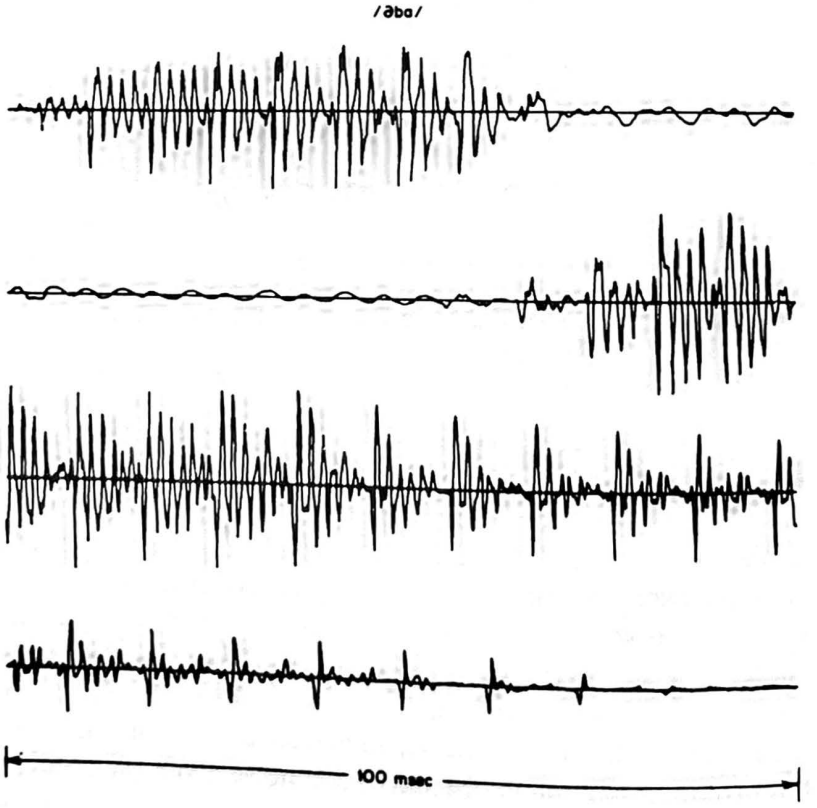


Figure 2.26 Waveform for the sequence /ə-b-a/.

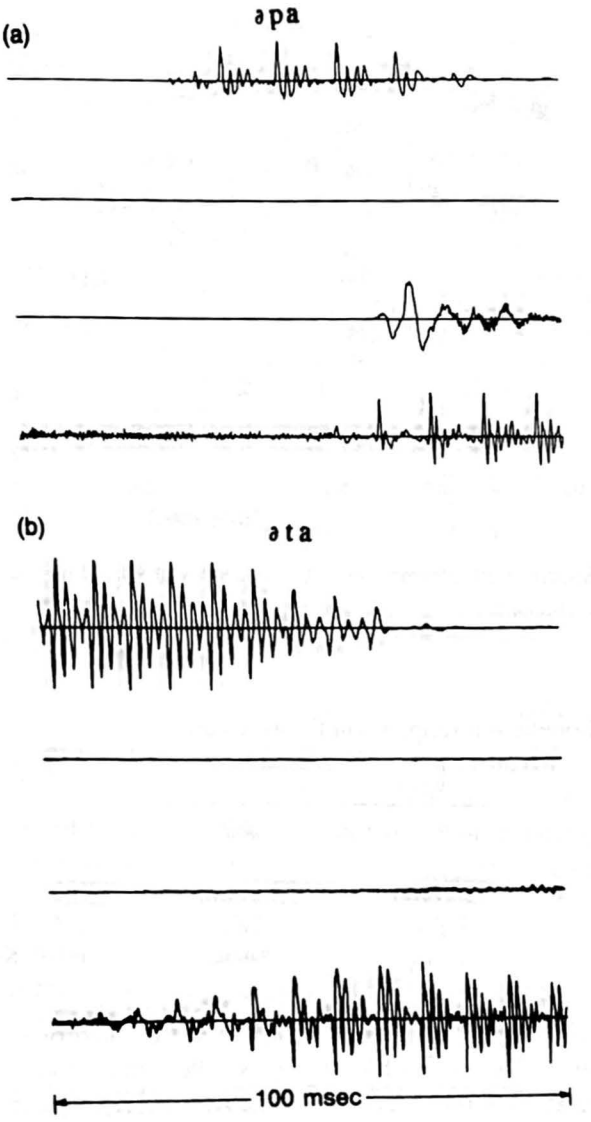


Figure 2.27 Waveforms for the sequences /ə-p-a/ and /ə-t-a/.

2.4.8 Review Exercises

As a self-check on the reader's understanding of the material on speech sounds and their acoustic manifestations, we now digress and present some simple exercises along with the solutions. For maximum effectiveness, the reader is encouraged to think through each exercise before looking at the solution.

Exercise 2.1

1. Write out the phonetic transcription for the following words:

   he, eats, several, light, tacos

2. What effect occurs when these five words are spoken in sequence as a sentence? What does this imply about automatic speech recognition?

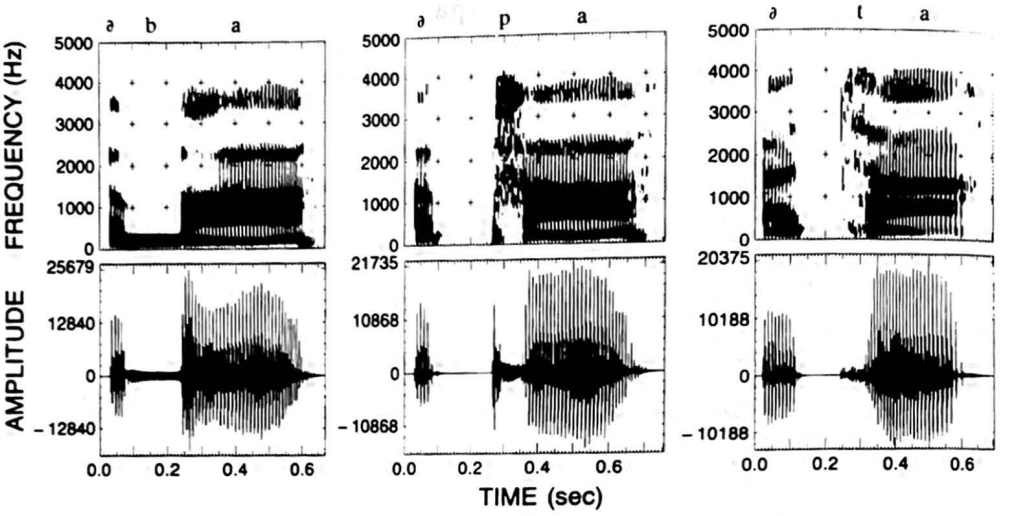


Figure 2.28 Spectrogram comparisons of the sequences of voiced (/ə-b-a/) and voiceless (/ə-p-a/ and /ə-t-a/) stop consonants.

Solution 2.1

1. The phonetic transcriptions of the words are

Word	Phoneme Sequence	ARPABET
he	/hi/	HH–IY
eats	/its/	IY–TS
several	/sɛvrʌl/	S–EH–V–R–AH–L
light	/laʸt/	L–AY–T
tacos	/takoz/	T–AA–K–OW–Z

2. When the words are spoken together, the last sound of each word merges with the first sound of the succeeding word (since they are the same sound), resulting in strong coarticulation of boundary sounds. The ARPABET transcription for the sentence is:

   HH–IY–T–S–EH–V–R–AH–L–AY–T–AA–K–OW–Z

   All information about word boundaries is totally lost; furthermore, the durations of the common sounds at the boundaries of words are much shorter than what would be predicted from the individual words.

Exercise 2.2

Some of the difficulties in large vocabulary speech recognition are related to the irregularities in the way basic speech sounds are combined to produce words. Exercise 2.2 highlights a couple of these difficulties.

1. In word initial position of American English, which phoneme or phonemes can never occur? Which hardly ever occur?
2. There are many word initial consonant clusters of length two, such as *sp*eak, *dr*ank, *pl*ead, and *pr*ess. How many word initial consonant clusters of length three are there in American English? What general rule can you give about the sounds in each of the three positions?

3. A nasal consonant can be combined with a stop consonant (e.g., ca*mp*, te*nd*) in a limited number of ways. What general rule do such combinations obey? There are several notable exceptions to this general rule. Can you give a couple of exceptions? What kind of speaking irregularity often results from these exceptions?

Solution 2.2

1. The only phoneme that never occurs in initial word position in English is the /ng/ sound (e.g., si*ng*). The only other sound that almost never occurs naturally in English, in initial word position, is /zh/ except some foreign words imported into English, such as gendarme, which does have an initial /zh/.
2. The word initial consonant clusters of length three in English include

/spl/	—	split
/spr/	—	spring
/skw/	—	squirt
/skr/	—	script
/str/	—	string

The general rule for such clusters is

/sound s/unvoiced stop/semivowel/

3. The general rule for a nasal-stop combination is that the nasal and stop have the same place of articulation, e.g., front/lips (/mp/), mid/dental (/nt/), back/velar (/ng k/). Exceptions occur in words like summed (/md/) or hanged (/ng d/) or dreamt (/mt/). There is often a tendency to insert an extra stop in such situations (e.g., dreamt → /drempt/).

Exercise 2.3

An important speech task is accurate digit recognition. This exercise seeks to exploit knowledge of acoustic phonetics to recognize first isolated digits, and next some simple connected digit strings. We first need a sound lexicon (a dictionary) for the digits. The sound lexicon describes the pronunciations of digits in terms of the basic sounds of English. Such a sound lexicon is given in Table 2.3. A single male adult talker (LRR) spoke each of the 11 digits in random sequence and in isolation, and spectrograms of these spoken utterances are shown in Figure 2.29. Figure 2.30 shows spectrograms of two connected digit sequences spoken by the same talker.

1. Identify each of the 11 digits based on the acoustic properties of the sounds within the digit (as expressed in the sound lexicon). Remember that each digit was spoken exactly once.
2. Try to identify the spoken digits in each of the connected digit strings.

Solution 2.3

1. The digits of the top row are 3 and 7:
   a. The digit 3 is cued by the distinctive brief initial fricative (/θ/), followed by the semivowel /r/ where the second and third formants both get very low in frequency, followed by the /i/ where F_2 and F_3 both become very high in frequency.
   b. The digit 7 is cued by the strong /s/ frication at the beginning, the distinctive /ɛ/, followed by the voiced fricative /v/, a short vowel /ə/ and ending in the strong

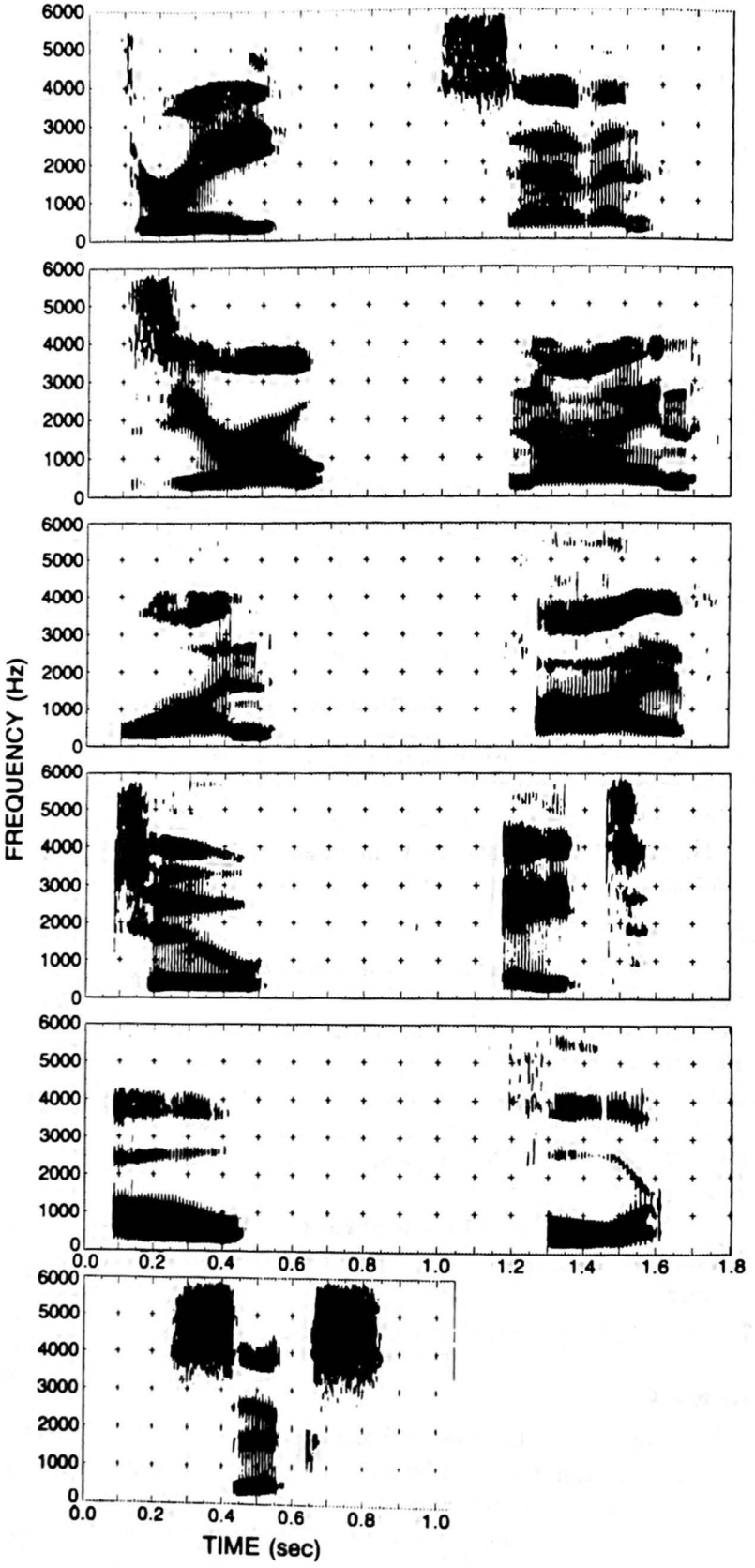


Figure 2.29 Spectrograms of the 11 isolated digits, 0 through 9 plus oh, in random sequence.

TABLE 2.3. Sound Lexicon of Digits

Word	Sounds	ARPABET
Zero	/z I r o/	Z–IH–R–OW
One	/w ʌ n/	W–AH–N
Two	/t u/	T–UW
Three	/θ r i/	TH–R–IY
Four	/f o r/	F–OW–R
Five	/f a^y v/	F–AY–V
Six	/s I k s/	S–IH–K–S
Seven	/s ɛ v ə n/	S–EH–V–AX–N
Eight	/e^y t/	EY–T
Nine	/n a^y n/	N–AY–N
Oh	/o/	OW

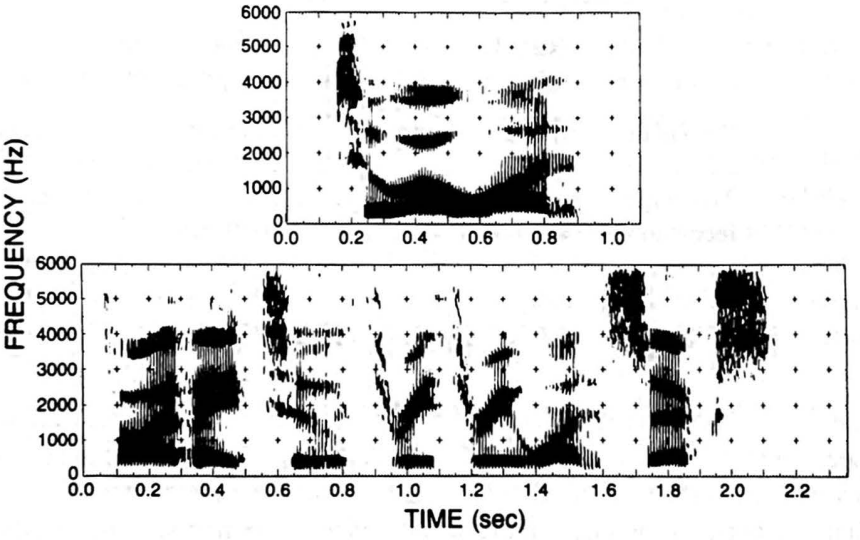


Figure 2.30 Spectrograms of two connected digit sequences.

nasal /n/.

The digits in the second row are 0 and 9:

a. The initial /z/ is cued by the strong frication with the presence of voicing at low frequencies; the following /I/ is seen by the high F_2 and F_3, the /r/ is signaled by the low F_2 and F_3, and the diphthong /o/ is signaled by the gliding motion of F_2 and F_3 toward an /u/-like sound.

b. The digit 9 is cued by the distinct initial and final nasals /n/ and by the /a^y/ glide between the nasals.

The digits in the third row are 1 and 5:

a. The digit 1 is cued by the strong initial semivowel /w/ with very low F_2 and by the strong final nasal /n/.

b. The digit 5 is cued by the weak initial frication of /f/, followed by the strong diphthong /a^y/ and ending in the very weak fricative /v/.

The digits in the fourth row are 2 and 8:

a. The digit 2 is cued by the strong /t/ burst and release followed by the glide to the /u/ sound.

b. The digit 8 is cued by the initial weak diphthong /e^y/ followed by a clear stop gap of the /t/ and then the /t/ release.

The digits in the fifth row are "oh" and 4:

a. The digit "oh" is virtually a steady sound with a slight gliding tendency toward /u/ at the end.

b. The digit 4 is cued by the weak initial fricative /f/, followed by the strong /o/ vowel and ending with a classic /r/ where F_2 and F_3 merge together.

The digit in the last row is 6:

a. The digit 6 is cued by the strong /s/ frication at the beginning and end, and by the steady vowel /I/ followed by the stop gap and release of the /k/.

2. By examining the isolated digit sequences, one can eventually (with a lot of work and some good luck) conclude that the two sequences are

Row 1:	2-oh-1	(telephone area code)
Row 2:	5-8-2-3-3-1-6	(7-digit telephone number)

We will defer any explanation of how any reasonable person, or machine, could perform this task until later in this book when we discuss connected word-recognition techniques. The purpose of this exercise is to convince the reader how difficult a relatively simple recognition task can be.

2.5 APPROACHES TO AUTOMATIC SPEECH RECOGNITION BY MACHINE

The material presented in the previous sections leads to a straightforward way of performing speech recognition by machine whereby the machine attempts to decode the speech signal in a sequential manner based on the observed acoustic features of the signal and the known relations between acoustic features and phonetic symbols. This method, appropriately called the acoustic-phonetic approach, is indeed viable and has been studied in great depth for more than 40 years. However, for a variety of reasons, the acoustic-phonetic approach has not achieved the same success in practical systems as have alternative methods. Hence, in this section, we provide an overview of several proposed approaches to automatic speech recognition by machine with the goal of providing some understanding as to the essentials of each proposed method, and the basic strengths and weaknesses of each approach.

Broadly speaking, there are three approaches to speech recognition, namely:

1. the acoustic-phonetic approach
2. the pattern recognition approach
3. the artificial intelligence approach

The acoustic-phonetic approach is based on the theory of acoustic phonetics that postulates that there exist finite, distinctive phonetic units in spoken language and that the phonetic

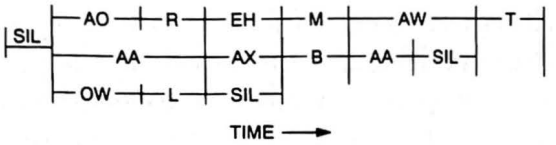


Figure 2.31 Phoneme lattice for word string.

units are broadly characterized by a set of properties that are manifest in the speech signal, or its spectrum, over time. Even though the acoustic properties of phonetic units are highly variable, both with speakers and with neighboring phonetic units (the so-called coarticulation of sounds), it is assumed that the rules governing the variability are straightforward and can readily be learned and applied in practical situations. Hence the first step in the acoustic-phonetic approach to speech recognition is called a segmentation and labeling phase because it involves segmenting the speech signal into discrete (in time) regions where the acoustic properties of the signal are representative of one (or possibly several) phonetic units (or classes), and then attaching one or more phonetic labels to each segmented region according to the acoustic properties. To actually do speech recognition, a second step is required. This second step attempts to determine a valid word (or string of words) from the sequence of phonetic labels produced in the first step, which is consistent with the constraints of the speech-recognition task (i.e., the words are drawn from a given vocabulary, the word sequence makes syntactic sense and has semantic meaning, etc.).

To illustrate the steps involved in the acoustic-phonetic approach to speech recognition, consider the phoneme lattice shown in Figure 2.31. (A phoneme lattice is the result of the segmentation and labeling step of the recognition process and represents a sequential set of phonemes that are likely matches to the spoken input speech.) The problem is to decode the phoneme lattice into a word string (one or more words) such that every instant of time is included in one of the phonemes in the lattice, and such that the word (or word sequence) is valid according to rules of English syntax. (The symbol SIL stands for silence or a pause between sounds or words; the vertical position in the lattice, at any time, is a measure of the goodness of the acoustic match to the phonetic unit, with the highest unit having the best match.) With a modest amount of searching, one can derive the appropriate phonetic string SIL–AO–L–AX–B–AW–T corresponding to the word string "all about," with the phonemes L, AX, and B having been second or third choices in the lattice and all other phonemes having been first choices. This simple example illustrates well the difficulty in decoding phonetic units into word strings. This is the so-called lexical access problem. Interestingly, as we will see in the next section, the real problem with the acoustic-phonetic approach to speech recognition is the difficulty in getting a reliable phoneme lattice for the lexical access stage.

The pattern-recognition approach to speech recognition is basically one in which the speech patterns are used directly without explicit feature determination (in the acoustic-phonetic sense) and segmentation. As in most pattern-recognition approaches, the method has two steps—namely, training of speech patterns, and recognition of patterns via pattern comparison. Speech "knowledge" is brought into the system via the training procedure. The concept is that if enough versions of a pattern to be recognized (be it a sound, a word, a phrase, etc.) are included in a training set provided to the algorithm, the training procedure

should be able to adequately characterize the acoustic properties of the pattern (with no regard for or knowledge of any other pattern presented to the training procedure). This type of characterization of speech via training is called pattern classification because the machine learns which acoustic properties of the speech class are reliable and repeatable across all training tokens of the pattern. The utility of the method is the pattern-comparison stage, which does a direct comparison of the unknown speech (the speech to be recognized), with each possible pattern learned in the training phase and classifies the unknown speech according to the goodness of match of the patterns.

The pattern-recognition approach to speech recognition is the basis for the remainder of this book. Hence there will be a great deal of discussion and explanation of virtually every aspect of the procedure. However, at this point, suffice it to say that the pattern-recognition approach is the method of choice for speech recognition for three reasons:

1. Simplicity of use. The method is easy to understand, it is rich in mathematical and communication theory justification for individual procedures used in training and decoding, and it is widely used and understood.
2. Robustness and invariance to different speech vocabularies, users, feature sets, pattern comparison algorithms and decision rules. This property makes the algorithm appropriate for a wide range of speech units (ranging from phonemelike units all the way through words, phrases, and sentences), word vocabularies, talker populations, background environments, transmission conditions, etc.
3. Proven high performance. It will be shown that the pattern-recognition approach to speech recognition consistently provides high performance on any task that is reasonable for the technology and provides a clear path for extending the technology in a wide range of directions such that the performance degrades gracefully as the problem becomes more and more difficult.

The so-called artificial intelligence approach to speech recognition is a hybrid of the acoustic-phonetic approach and the pattern-recognition approach in that it exploits ideas and concepts of both methods. The artificial intelligence approach attempts to mechanize the recognition procedure according to the way a person applies its intelligence in visualizing, analyzing, and finally making a decision on the measured acoustic features. In particular, among the techniques used within this class of methods are the use of an expert system for segmentation and labeling so that this crucial and most difficult step can be performed with more than just the acoustic information used by pure acoustic-phonetic methods (in particular, methods that integrate phonemic, lexical, syntactic, semantic, and even pragmatic knowledge into the expert system have been proposed and studied); learning and adapting over time (i.e., the concept that knowledge is often both static and dynamic and that models must adapt to the dynamic component of the data); the use of neural networks for learning the relationships between phonetic events and all known inputs (including acoustic, lexical, syntactic, semantic, etc.) as well as for discrimination between similar sound classes.

The use of neural networks could represent a separate structural approach to speech recognition or be regarded as an implementational architecture that may be incorporated

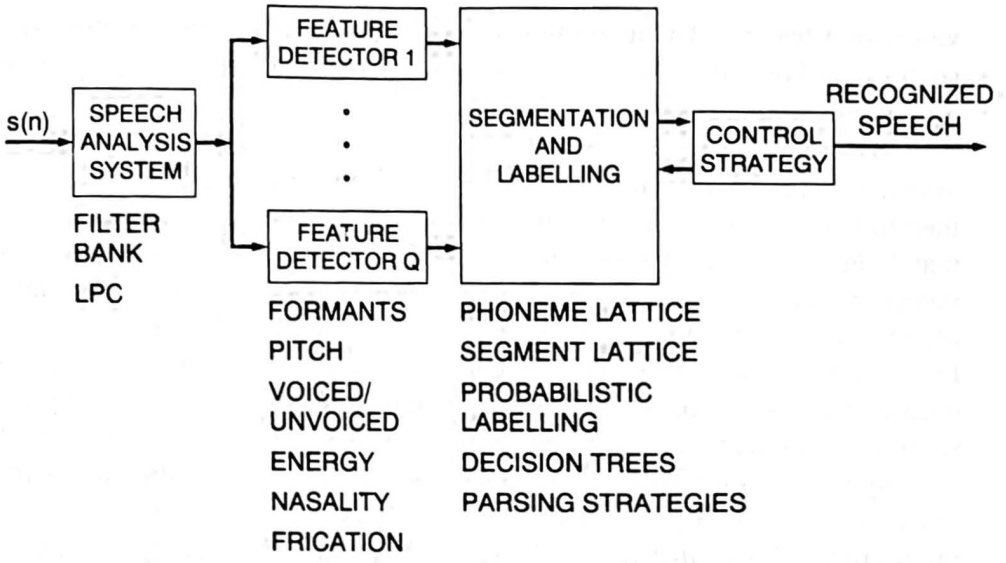


Figure 2.32 Block diagram of acoustic-phonetic speech-recognition system.

in any of the above three classical approaches. The concepts and ideas of applying neural networks to speech-recognition problems are relatively new; hence we will devote a fair amount of discussion within this chapter to outline the basic ways in which neural networks are used in general, and applied to problems in speech recognition, in particular. In the next several sections we expand on the ideas of these three general approaches to speech recognition by machine.

2.5.1 Acoustic-Phonetic Approach to Speech Recognition

Figure 2.32 shows a block diagram of the acoustic-phonetic approach to speech recognition. The first step in the processing (a step common to all approaches to speech recognition) is the speech analysis system (the so-called feature measurement method), which provides an appropriate (spectral) representation of the characteristics of the time-varying speech signal. The most common techniques of spectral analysis are the class of filter bank methods and the class of linear predictive coding (LPC) methods. The properties of these methods will be discussed in great detail in Chapter 3. Broadly speaking, both of these methods provide spectral descriptions of the speech over time.

The next step in the processing is the feature-detection stage. The idea here is to convert the spectral measurements to a set of features that describe the broad acoustic properties of the different phonetic units. Among the features proposed for recognition are nasality (presence or absence of nasal resonance), frication (presence or absence of random excitation in the speech), formant locations (frequencies of the first three resonances), voiced-unvoiced classification (periodic or aperiodic excitation), and ratios of high- and low-frequency energy. Many proposed features are inherently binary (e.g., nasality, frication, voiced-unvoiced); others are continuous (e.g., formant locations, energy ratios). The feature-detection stage usually consists of a set of detectors that operate in parallel and use appropriate processing and logic to make the decision as to presence or absence, or

value, of a feature. The algorithms used for individual feature detectors are sometimes sophisticated ones that do a lot of signal processing, and sometimes they are rather trivial estimation procedures.

The third step in the procedure is the segmentation and labeling phase whereby the system tries to find stable regions (where the features change very little over the region) and then to label the segmented region according to how well the features within that region match those of individual phonetic units. This stage is the heart of the acoustic-phonetic recognizer and is the most difficult one to carry out reliably; hence various control strategies are used to limit the range of segmentation points and label possibilities. For example, for individual word recognition, the constraint that a word contains at least two phonetic units and no more than six phonetic units means that the control strategy need consider solutions with between 1 and 5 internal segmentation points. Furthermore, the labeling strategy can exploit lexical constraints on words to consider only words with n phonetic units whenever the segmentation gives $n - 1$ segmentation points. These constraints are often powerful ones that reduce the search space and significantly increase performance (accuracy of segmentation and labeling) of the system.

The result of the segmentation and labeling step is usually a phoneme lattice (of the type shown in Figure 2.31) from which a lexical access procedure determines the best matching word or sequence of words. Other types of lattices (e.g., syllable, word) can also be derived by integrating vocabulary and syntax constraints into the control strategy as discussed above. The quality of the matching of the features, within a segment, to phonetic units can be used to assign probabilities to the labels, which then can be used in a probabilistic lexical access procedure. The final output of the recognizer is the word or word sequence that best matches, in some well-defined sense, the sequence of phonetic units in the phoneme lattice.

2.5.1.1 Acoustic Phonetic Vowel Classifier

To illustrate the labeling procedure on a segment classified as a vowel, consider the flow chart of Figure 2.33. We assume that three features have been detected over the segment—namely, first formant, F_1, second formant, F_2, and duration of the segment, D. Consider just the set of steady vowels (i.e., we exclude the diphthongs). To classify a vowel segment as one of the 10 steady vowels, several tests can be made to separate groups of vowels. As shown in Figure 2.33 the first test separates vowels with low F_1 (called diffuse vowels and including /i/, /ɪ/, /ə/, /ʊ/, /u/) from vowels with high F_1 (called compact vowels and including /ɛ/, /æ/, /a/, /ʌ/, /ɔ/). Each of these subsets can be split further on the basis of F_2 measurements, with acute vowels having high F_2 and grave vowels having low F_2. The third test is one based on segment duration, which separates tense vowels (large values of D) from lax vowels (small values of D). Finally, a finer test on formant values separates the remaining unresolved vowels, resolving the vowels into flat vowels (where $F_1 + F_2$ exceeds a threshold T) and plain vowels (where $F_1 + F_2$ falls below the threshold T).

It should be clear that there are several thresholds embedded within the vowel classifier. Such thresholds are often determined experimentally so as to maximize classification accuracy on a given corpus of speech.

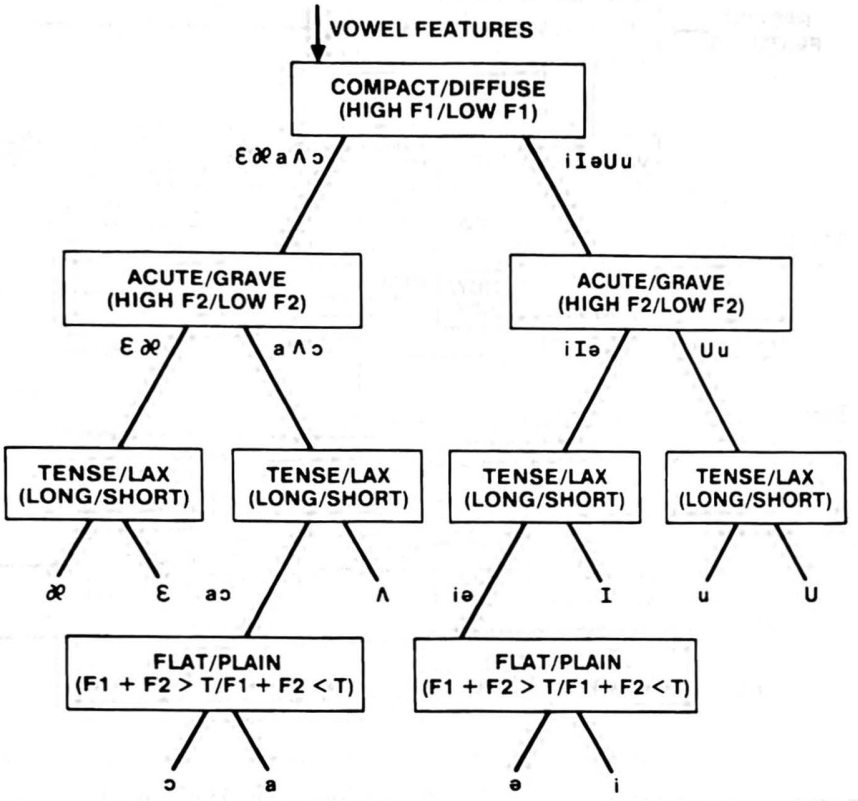


Figure 2.33 Acoustic-phonetic vowel classifier.

2.5.1.2 Speech Sound Classifier

Vowel classification is just a small part of the phonetic labeling procedure of an acoustic-phonetic recognizer. In theory, one needs a method of classifying an arbitrary segment into one (or more) of the 40 plus phonetic units discussed earlier in this chapter. Rather than discussing how to solve this very difficult problem, consider the somewhat simpler problem of classifying a speech segment into one of several broad speech classes—e.g., unvoiced stop, voiced stop, unvoiced fricative. Again there is no simple or generally well-accepted procedure for accomplishing this task; however, we show in Figure 2.34 one simple and straightforward way to accomplish such a classification.

The method uses a binary tree to make decisions as to various broad sound classes. The first decision is a *sound/silence* split in which the speech features (primarily energy in this case) are compared to selected thresholds, and *silence* is split off if the test is negative for speech sounds. The second decision is a *voiced/unvoiced* decision (primarily based on the presence of periodicity within the segment) in which unvoiced sounds are split apart from voiced sounds. A test for unvoiced stop consonants is made (seeing if a stop gap of silence preceded the segment), and this separates the unvoiced stops (/t/, /p/, /k/, /ĉ/) from the unvoiced fricatives (/f/, /θ/, /s/, /ŝ/). A *high-frequency/low-frequency* (energy) test separates voiced fricatives (/v/, /ð/, /z/, /ẑ/) from other voiced sounds. Voiced stops are separated out by checking to see whether the preceding sound is silence (or silencelike). Finally, a *vowel/sonorant* spectral test (searching for spectral gaps) separates vowels from

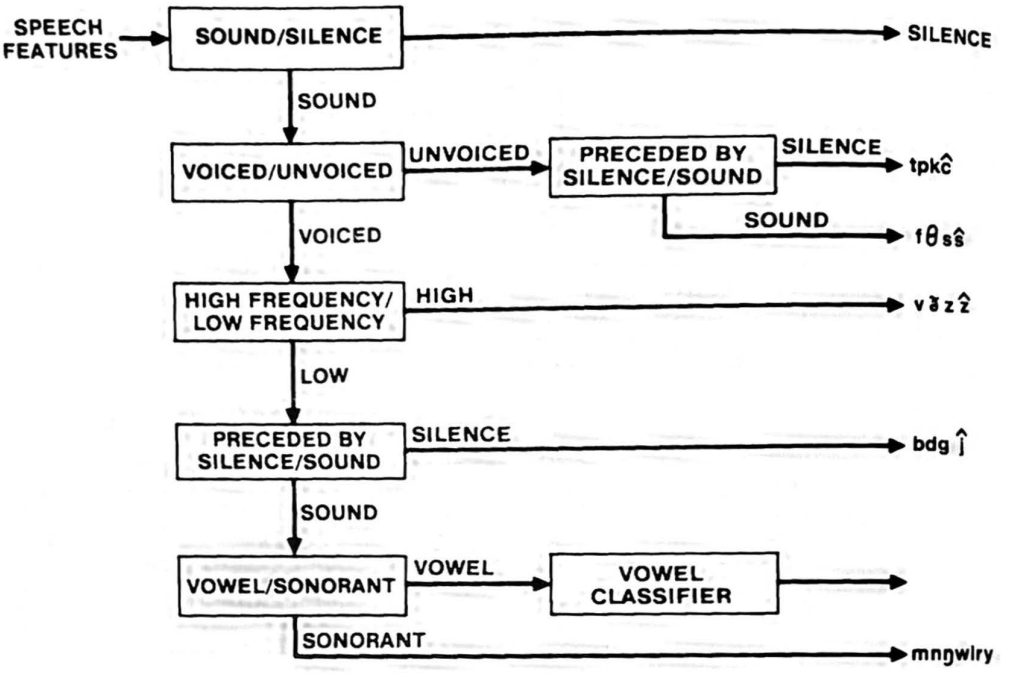


Figure 2.34 Binary tree speech sound classifier.

sonorants (nasal consonants and /w/, /l/, /r/, and /y/). The vowel classifier of Figure 2.33 can then be used for finer vowel distinctions.

The tests shown in Figure 2.34 are rather crude and are therefore highly prone to error. For example, some voiced stop consonants are *not* preceded by silence or by a silencelike sound. Another problem is that no way of distinguishing diphthongs from vowels is provided. Virtually every decision in the binary tree is subject to scrutiny as to its utility in any practical system.

2.5.1.3 Examples of Acoustic Phonetic Labeling

To illustrate some of the difficulties faced by the acoustic-phonetic approach to speech recognition, consider the following example. (Shown in the example is the phonetic labeling of a sentence [only the top-choice phonetic candidate is shown for each segment], along with its decoding into the proper word sequence.) In this example (taken from an actual acoustic-phonetic recognizer) we see that there are inserted phonetic units (Y in "MAY," AX in "BY"), deleted phonetic units (N in "EARN," N in "MONEY"), and phonetic substitutions (J for K in "WORKING," N for NG in "WORKING"). The difficulty of proper decoding of phonetic units into words and sentences grows dramatically with increases in the rates of phoneme insertion, deletion, and substitution.

phonemes:	/sil/	–/j/–/e/–/n/	/m/–/e/–/y/	/ɝ/–/m/–/ɔ/ – /r/	/m/–/ʌ/–/sil/–/e/	
ARPABET:	SIL	–JH–EY– N + M –EY– Y +	ER– M –AO– R	+ M –AH–SIL–EY		
words:		JANE	MAY	EARN MORE	MONEY	

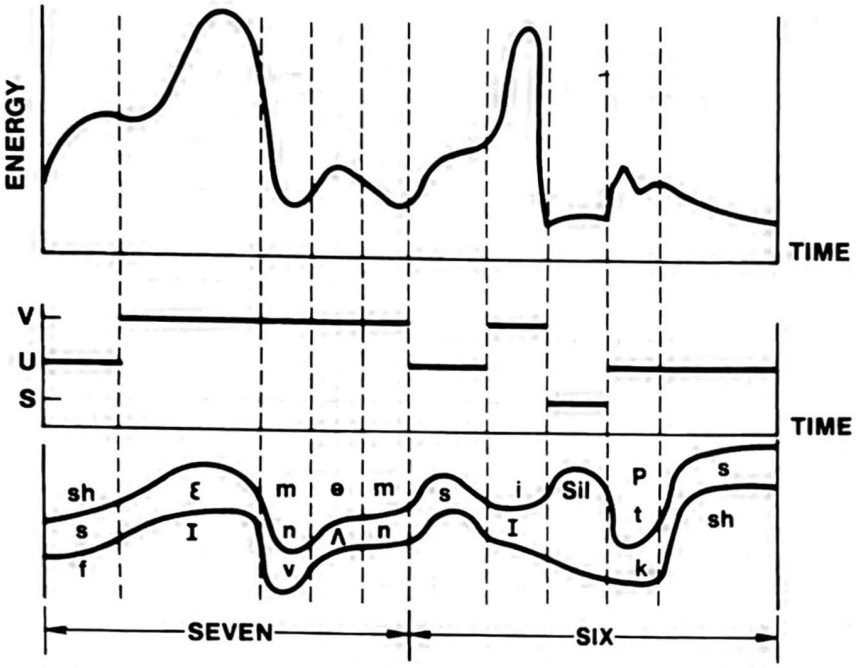


Figure 2.35 Segmentation and labeling for word sequence "seven–six."

phonemes:	/b/–/aʸ/–/ə/ /w/–/ɝ/–/sil/–/j/–/l/–/n/ /h/–/a/–/r/–/sil/–/d/
ARPABET:	B–AY–AX+ W–ER–SIL–J–IH–N +HH–AA–R–SIL–D
words:	BY WORKING HARD

Two other examples of acoustic-phonetic segmentation and labeling are given in Figures 2.35 and 2.36. Shown in these figures are the energy contour of the speech signal, the voiced-unvoiced-silence classification over time, the segmentation points, and the lattice of phonetic units. The "proper" decoding of the lattice corresponding to the spoken word is shown as the phonetic units enclosed within the solid heavy lines. For the example of Figure 2.35 (the digit sequence "seven-six"), we see that although most top phoneme candidate errors are within the same sound class (e.g., /sh/ instead of /s/), some errors are between classes (e.g., /m/ instead of /v/). For decoding into digits, such cross-class errors are usually of little significance.

For the example of Figure 2.36 (the word sequence "did you"), the decoding into phonetic units is only the first step in a difficult decoding problem, because the basic speech sounds of the words "did" and "you" are phonologically changed in context from D-IH-D-Y-UW to D-IH-J-UH. This phonological effect exacerbates the problem of acoustic phonetic decoding even further than the insertion/deletion/substitution problems mentioned earlier.

2.5.1.4 Issues in Acoustic Phonetic Approach

Many problems are associated with the acoustic-phonetic approach to speech recognition. These problems, in many ways, account for the lack of success in practical speech-recognition systems. Among these are the following:

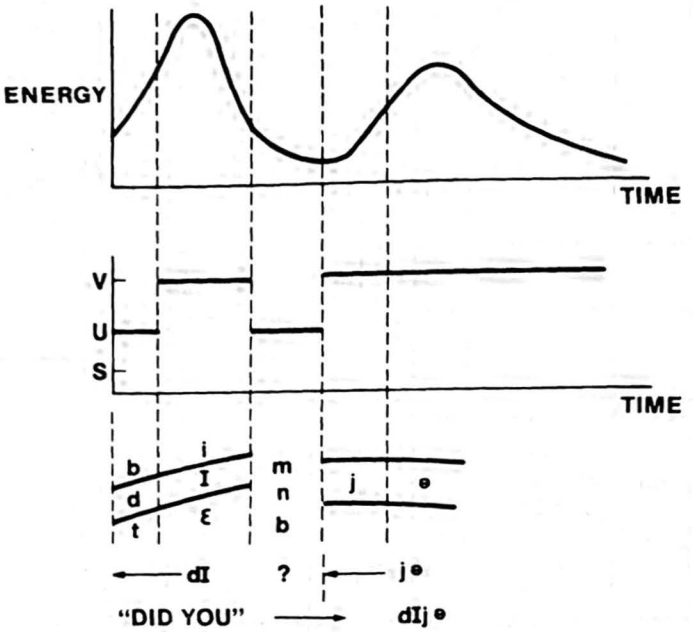


Figure 2.36 Segmentation and labeling for word sequence "did you."

1. The method requires extensive knowledge of the acoustic properties of phonetic units. (Recall that the existence of phonetic units is assumed a priori in the acoustic-phonetic approach. Knowledge of acoustic properties of these phonetic units often is established in an a posteriori manner.) This knowledge is, at best incomplete, and at worst totally unavailable for all but the simplest of situations (e.g., steady vowels).
2. The choice of features is made mostly based on ad hoc considerations. For most systems the choice of features is based on intuition and is not optimal in a well-defined and meaningful sense.
3. The design of sound classifiers is also not optimal. Ad hoc methods are generally used to construct binary decision trees. More recently classification and regression tree (CART) methods have been used to make the decision trees more robust [10]. However, since the choice of features is most likely to be suboptimal, optimal implementation of CART is rarely achieved.
4. No well-defined, automatic procedure exists for tuning the method (i.e., adjusting decision thresholds, etc.) on real, labeled speech. In fact, there is not even an ideal way of labeling the training speech in a manner consistent and agreed on uniformly by a wide class of linguistic experts.

Because of all these problems, the acoustic-phonetic method of speech recognition remains an interesting idea but one that needs much more research and understanding before it can be used successfully in actual speech-recognition problems.

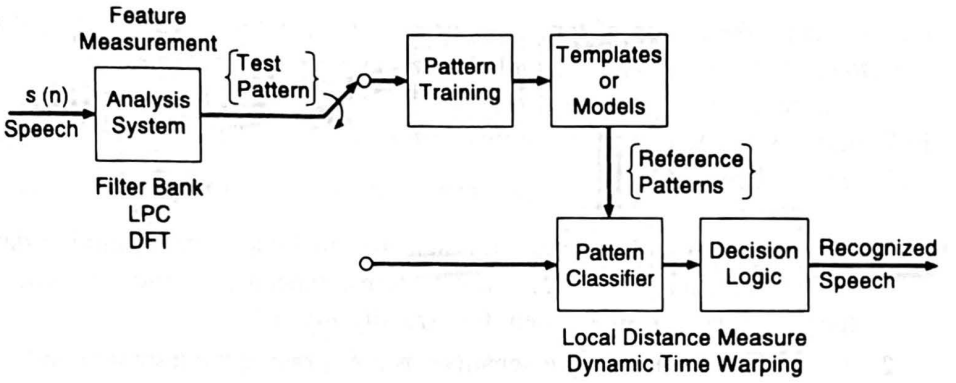


Figure 2.37 Block diagram of pattern-recognition speech recognizer.

2.5.2 Statistical Pattern-Recognition Approach to Speech Recognition

A block diagram of a canonic pattern-recognition approach to speech recognition is shown in Figure 2.37. The pattern-recognition paradigm has four steps, namely:

1. Feature measurement, in which a sequence of measurements is made on the input signal to define the "test pattern." For speech signals the feature measurements are usually the output of some type of spectral analysis technique, such as a filter bank analyzer, a linear predictive coding analysis, or a discrete Fourier transform (DFT) analysis.
2. Pattern training, in which one or more test patterns corresponding to speech sounds of the same class are used to create a pattern representative of the features of that class. The resulting pattern, generally called a reference pattern, can be an exemplar or template, derived from some type of averaging technique, or it can be a model that characterizes the statistics of the features of the reference pattern.
3. Pattern classification, in which the unknown test pattern is compared with each (sound) class reference pattern and a measure of similarity (distance) between the test pattern and each reference pattern is computed. To compare speech patterns (which consist of a sequence of spectral vectors), we require both a local distance measure, in which local distance is defined as the spectral "distance" between two well-defined spectral vectors, and a global time alignment procedure (often called a dynamic time warping algorithm), which compensates for different rates of speaking (time scales) of the two patterns.
4. Decision logic, in which the reference pattern similarity scores are used to decide which reference pattern (or possibly which sequence of reference patterns) best matches the unknown test pattern.

The factors that distinguish different pattern-recognition approaches are the types of feature

measurement, the choice of templates or models for reference patterns, and the method used to create reference patterns and classify unknown test patterns.

The remaining chapters of this book will discuss all aspects of the model shown in Figure 2.37. The general strengths and weaknesses of the pattern recognition model include the following:

1. The performance of the system is sensitive to the amount of training data available for creating sound class reference patterns; generally the more training, the higher the performance of the system for virtually any task.
2. The reference patterns are sensitive to the speaking environment and transmission characteristics of the medium used to create the speech; this is because the speech spectral characteristics are affected by transmission and background noise.
3. No speech-specific knowledge is used explicitly in the system; hence, the method is relatively insensitive to choice of vocabulary words, task, syntax, and task semantics.
4. The computational load for both pattern training and pattern classification is generally linearly proportional to the number of patterns being trained or recognized; hence, computation for a large number of sound classes could and often does become prohibitive.
5. Because the system is insensitive to sound class, the basic techniques are applicable to a wide range of speech sounds, including phrases, whole words, and subword units. Hence we will see how a basic set of techniques developed for one sound class (e.g., words) can generally be directly applied to different sound classes (e.g., subword units) with little or no modifications to the algorithms.
6. It is relatively straightforward to incorporate syntactic (and even semantic) constraints directly into the pattern-recognition structure, thereby improving recognition accuracy and reducing computation.

2.5.3 Artificial Intelligence (AI) Approaches to Speech Recognition

The basic idea of the artificial intelligence approach to speech recognition is to compile and incorporate knowledge from a variety of knowledge sources and to bring it to bear on the problem at hand. Thus, for example, the AI approach to segmentation and labeling would be to augment the generally used acoustic knowledge with phonemic knowledge, lexical knowledge, syntactic knowledge, semantic knowledge, and even pragmatic knowledge. To be more specific, we first define these different knowledge sources:

- acoustic knowledge—evidence of which sounds (predefined phonetic units) are spoken on the basis of spectral measurements and presence or absence of features
- lexical knowledge—the combination of acoustic evidence so as to postulate words as specified by a lexicon that maps sounds into words (or equivalently decomposes words into sounds)
- syntactic knowledge—the combination of words to form grammatically correct strings (according to a language model) such as sentences or phrases

- semantic knowledge—understanding of the task domain so as to be able to validate sentences (or phrases) that are consistent with the task being performed, or which are consistent with previously decoded sentences
- pragmatic knowledge—inference ability necessary in resolving ambiguity of meaning based on ways in which words are generally used.

To illustrate the correcting and constraining power of these knowledge sources, consider the following sentences:

1. Go to the refrigerator and get me a book.
2. The bears killed the rams.
3. Power plants colorless happily old.
4. Good ideas often run when least expected.

The first sentence is syntactically meaningful but semantically inconsistent. The second sentence can be interpreted in at least two pragmatically different ways, depending on whether the context is an event in a jungle or the description of a football game between two teams called the "bears" and the "rams." The third sentence is syntactically unacceptable and semantically meaningless. The fourth sentence is semantically inconsistent and can trivially be corrected by changing the word *run* to *come*, a slight phonetic difference.

The word-correcting capability of higher-level knowledge sources is illustrated in Figure 2.38, which shows the word error probability of a recognizer both with and without syntactic constraints, as a function of a "deviation" parameter sigma. As the deviation parameter gets larger, the word error probability increases for both cases; however, without syntax the word error probability rapidly leads to 1.0, but with syntax it increases gradually with increases in the noise parameter.

There are several ways to integrate knowledge sources within a speech recognizer. Perhaps the most standard approach is the "bottom-up" processor (Figure 2.39), in which the lowest-level processes (e.g., feature detection, phonetic decoding) precede higher-level processes (lexical decoding, language model) in a sequential manner so as to constrain each stage of the processing as little as possible. An alternative is the so-called "top-down" processor, in which the language model generates word hypotheses that are matched against the speech signal, and syntactically and semantically meaningful sentences are built up on the basis of the word match scores. Figure 2.40 shows a system that is often implemented in the top-down mode by integrating the unit matching, lexical decoding, and syntactic analyses modules into a consistent framework. (This system will be discussed extensively in the chapter on large-vocabulary continuous-speech recognition.)

A third alternative is the so-called blackboard approach, as illustrated in Figure 2.41. In this approach, all knowledge sources (KS) are considered independent; a hypothesis-and-test paradigm serves as the basic medium of communication among KSs; each KS is data driven, based on the occurrence of patterns on the blackboard that match the templates specified by the KS; the system activity operates asynchronously; assigned cost and utility considerations and an overall ratings policy to combine and propagate ratings across all levels. The blackboard approach was extensively studied at CMU in the 1970s [11].

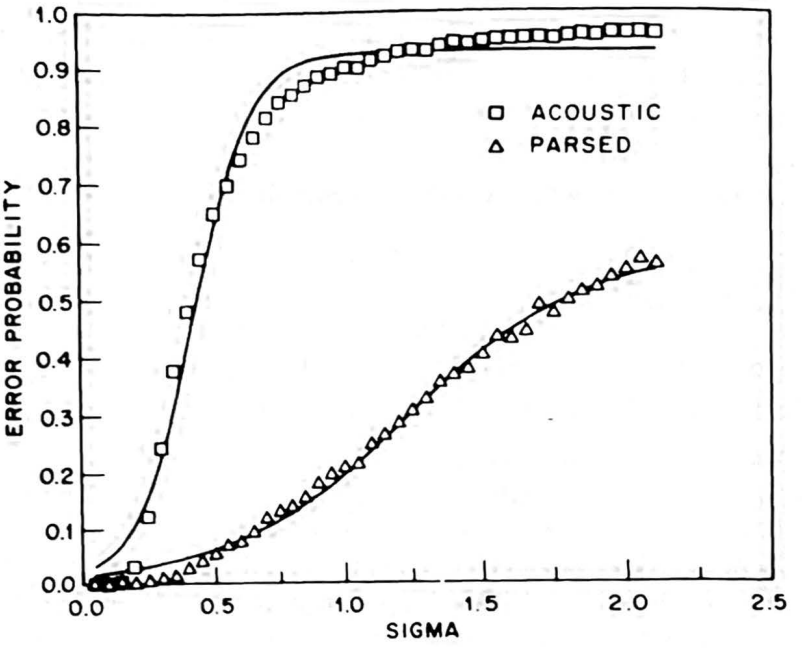


Figure 2.38 Illustration of the word correction capability of syntax in speech recognition (after Rabiner and Levinson [1]).

2.5.4 Neural Networks and Their Application to Speech Recognition

A variety of knowledge sources need to be established in the AI approach to speech recognition. Therefore, two key concepts of artificial intelligence are automatic knowledge acquisition (learning) and adaptation. One way in which these concepts have been implemented is via the neural network approach. In this section, we discuss the motivation for why people have studied neural networks and how they have been applied to speech-recognition systems.

Figure 2.42 shows a conceptual block diagram of a speech understanding system loosely based on a model of speech perception in human beings. The acoustic input signal is analyzed by an "ear model" that provides spectral information (over time) about the signal and stores it in a sensory information store. Other sensory information (e.g., from vision or touch) is available in the sensory information store and is used to provide several "feature-level" descriptions of the speech. Both long-term (static) and short-term (dynamic) memory are available to the various feature detectors. Finally, after several stages of refined feature detection, the final output of the system is an interpretation of the information in the acoustic input.

The system of Figure 2.42 is meant to model the human speech understanding system. The auditory analysis is based loosely on our understanding of the acoustic processing in the ear. The various feature analyses represent processing at various levels in the neural pathways to the brain. The short- and long-term memory provide external control of the neural processes in ways that are not well understood. The overall form of the model is that of a feed forward connectionist network—that is, a neural net. To better explain the strengths and limitations of neural networks, we now give a brief introduction to the issues

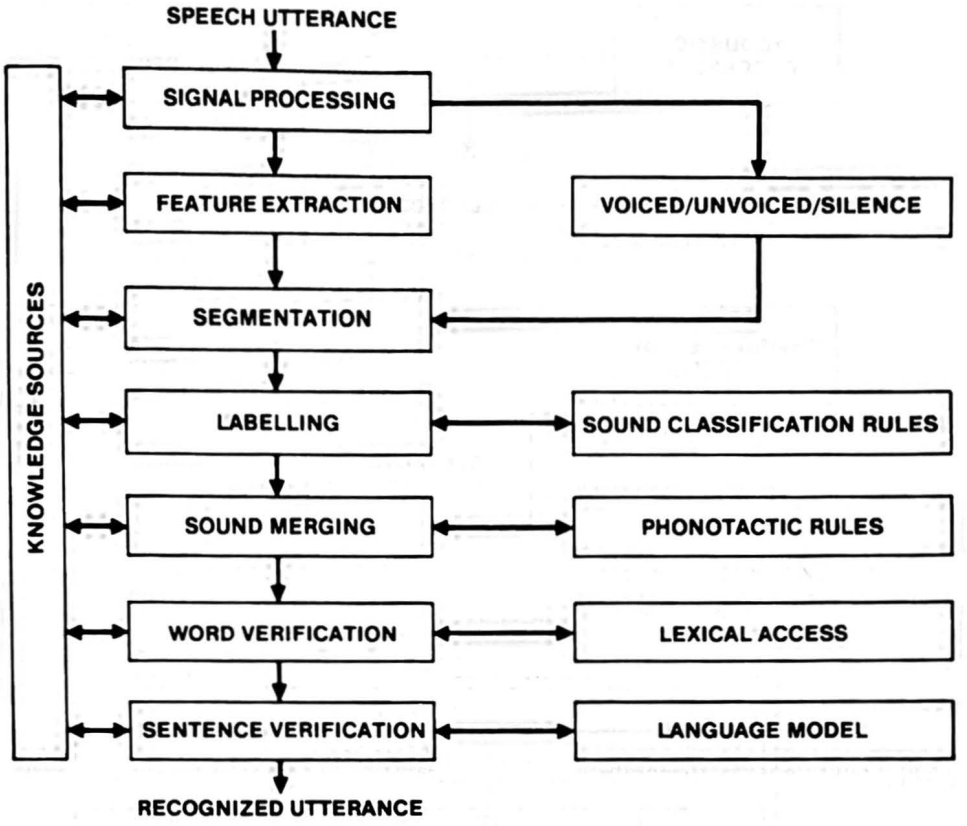


Figure 2.39 A bottom-up approach to knowledge integration for speech recognition.

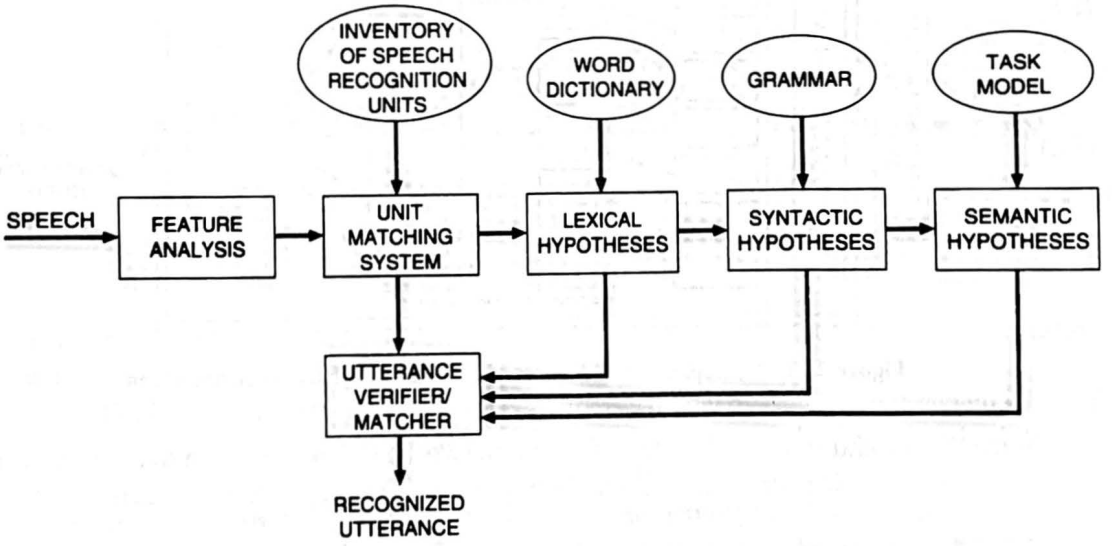


Figure 2.40 A top-down approach to knowledge integration for speech recognition.

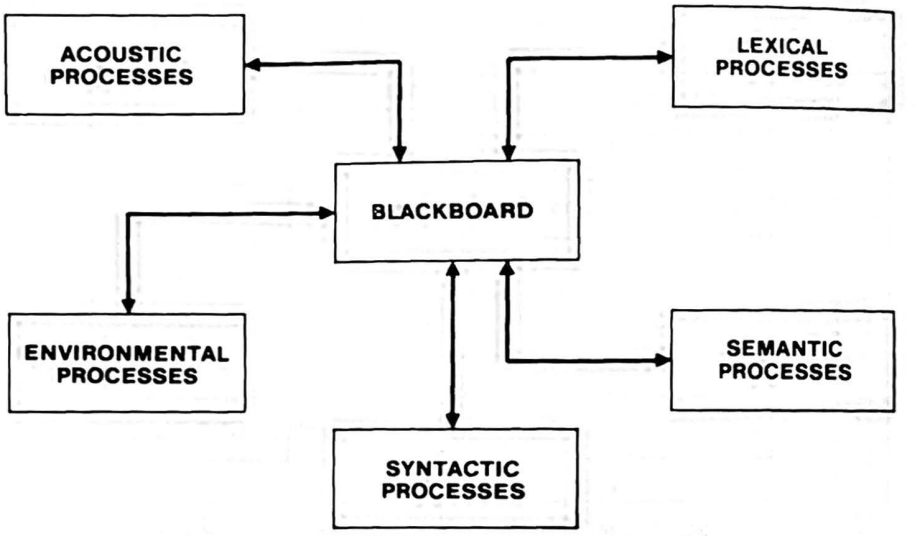


Figure 2.41 A blackboard approach to knowledge integration for speech recognition (after Lesser et al. [11]).

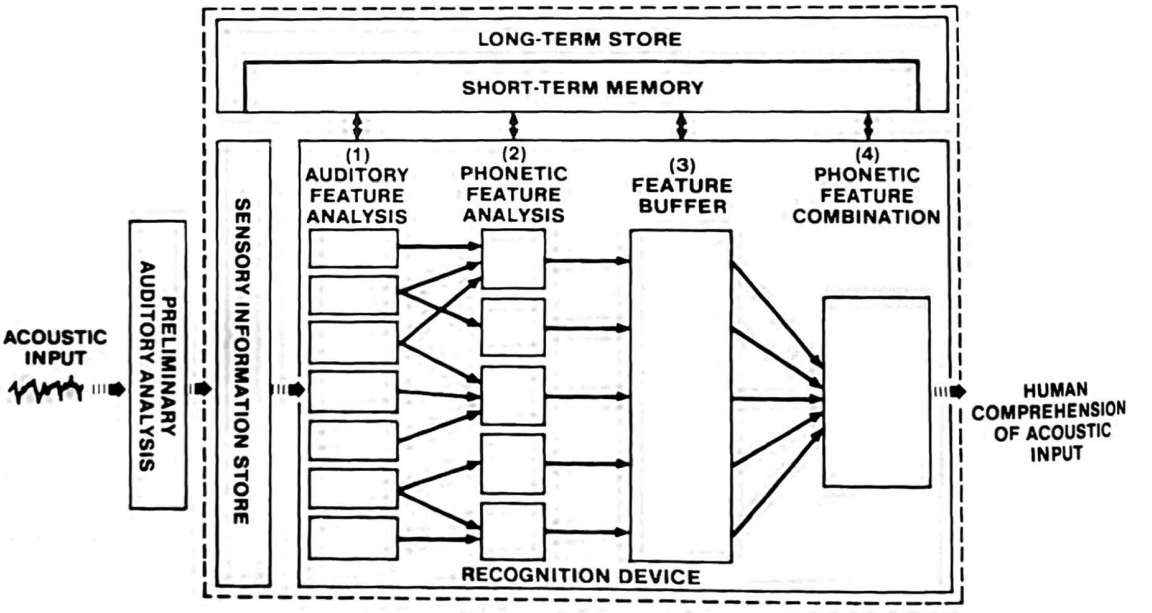


Figure 2.42 Conceptual block diagram of a human speech understanding system.

in the theory and implementations of neural networks. Then we return to some practical proposals for how neural networks could implement actual speech recognizers.

2.5.4.1 Basics of Neural Networks

A neural network, which is also called a connectionist model, a neural net, or a parallel distributed processing (PDP) model, is basically a dense interconnection of simple, non-linear, computation elements of the type shown in Figure 2.43. It is assumed that there

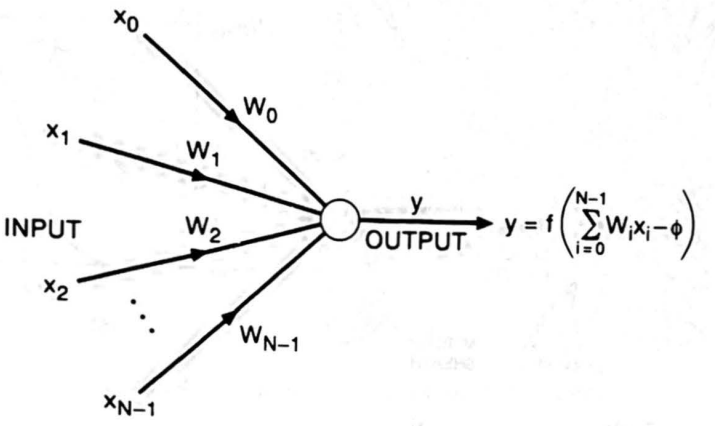


Figure 2.43 Simple computation element of a neural network.

are N inputs, labeled $x_1, x_2, \ldots, x_N$, which are summed with weights $w_1, w_2, \ldots, w_N$, thresholded, and then nonlinearly compressed to give the output y, defined as

$$y = f\left(\sum_{i=1}^{N} w_i x_i - \phi\right), \tag{2.1}$$

where ϕ is an internal threshold or offset, and f is a nonlinearity of one of the types given below:

1. hard limiter

$$f(x) = \begin{cases} +1, & x \leq 0 \\ -1, & x < 0 \end{cases} \tag{2.2}$$

or

2. sigmoid functions

$$f(x) = \tanh(\beta x), \quad \beta > 0 \tag{2.3}$$

or

$$f(x) = \frac{1}{1 + e^{-\beta x}}, \quad \beta > 0. \tag{2.4}$$

The sigmoid nonlinearities are used most often because they are continuous and differentiable.

The biological basis of the neural network is a model by McCullough and Pitts [12] of neurons in the human nervous system, as illustrated in Figure 2.44. This model exhibits all the properties of the neural element of Figure 2.43, including excitation potential thresholds for neuron firing (below which there is little or no activity) and nonlinear amplification, which compresses strong input signals.

2.5.4.2 Neural Network Topologies

There are several issues in the design of so-called artificial neural networks (ANNs), which model various physical phenomena, where we define an ANN as an arbitrary connection of

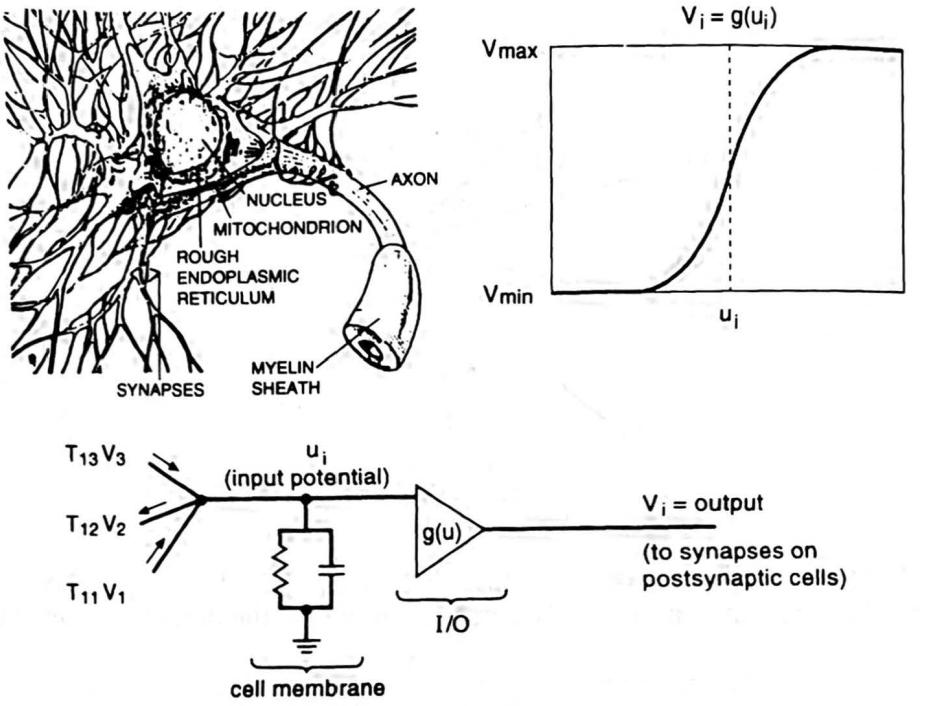


Figure 2.44 McCullough-Pitts model of neurons (after McCullough and Pitts [12]).

simple computational elements of the type shown in Figure 2.43. One key issue is *network topology*—that is, how the simple computational elements are interconnected. There are three standard and well known topologies:

- single/multilayer perceptrons
- Hopfield or recurrent networks
- Kohonen or self-organizing networks

In the single/multilayer perceptron, the outputs of one or more simple computational elements at one layer form the inputs to a new set of simple computational elements of the next layer. Figure 2.45 shows a single-layer perceptron and a three-layer perceptron. The single-layer perceptron has N inputs connected to M outputs in the output layer. The three-layer perceptron has two hidden layers between the input and output layers. What distinguishes the layers of the multilayer perceptron is the nonlinearity at each layer that enables the mapping between the input and output variables to possess certain particular classification/discrimination properties. For example, it can be proven that a single-layer perceptron, of the type shown in Figure 2.45a, can separate static patterns into classes with class boundaries characterized by hyperplanes in the $(x_1, x_2, \ldots, x_N)$ space. Similarly, a multilayer perceptron, with at least one hidden layer, can realize an arbitrary set of decision regions in the $(x_1, \ldots, x_N)$ space. Thus, for example, if the inputs to a multilayer perceptron are the first two speech resonances (F_1 and F_2), the network can implement a set of decision regions that partition the ($F_1 - F_2$) space into the 10 steady state vowels, as shown in Figure 2.46 [13].

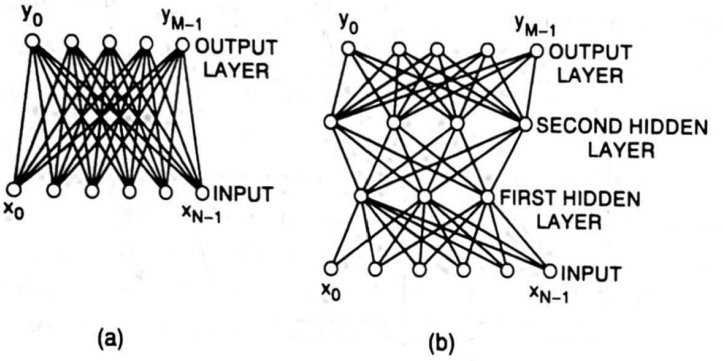


Figure 2.45 Single-layer and three-layer perceptrons.

- **Perceptrons:**

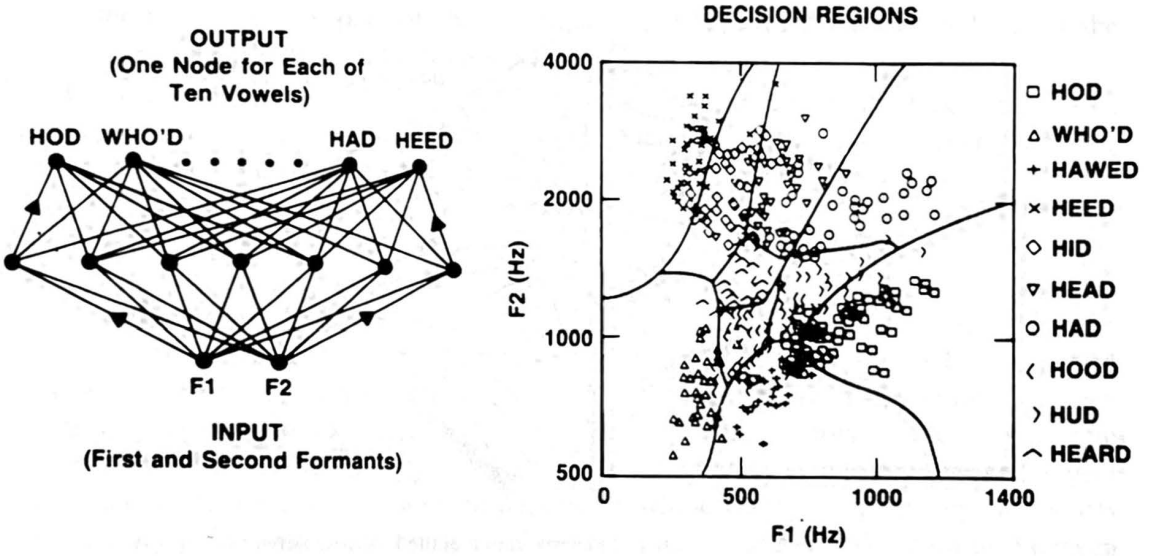


Figure 2.46 A multilayer perceptron for classifying steady vowels based on F_1, F_2 measurements (after Lippmann [13]).

The Hopfield network is a *recurrent network* in which the input to each computational element includes both inputs as well as outputs. Thus with the input and output indexed by time, $x_i(t)$ and $y_i(t)$, and the weight connecting the i^{th} node and the j^{th} node denoted by w_{ij}, the basic equation for the i^{th} recurrent computational element is

$$y_i(t) = f\left[x_i(t) + \sum_j w_{ij} y_j(t-1) - \phi\right] \tag{2.5}$$

and a recurrent network with N inputs and N outputs would have the form shown in Figure 2.47. The most important property of the Hopfield network is that when $w_{ij} = w_{ji}$ and when the recurrent computation (Eq. (2.5)) is performed asynchronously, for an arbitrary constant input, the network will eventually settle to a fixed point where $y_i(t) = y_i(t-1)$ for all i. These fixed relaxation points represent stable configurations of the network and can be used in applications that have a fixed set of patterns to be matched (e.g., printed letters)

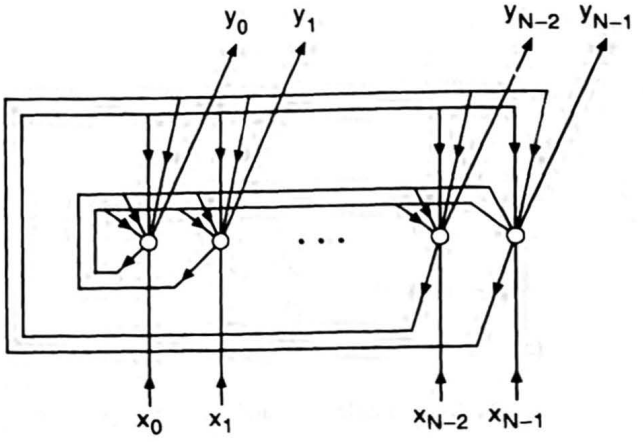


Figure 2.47 Model of a recurrent neural network.

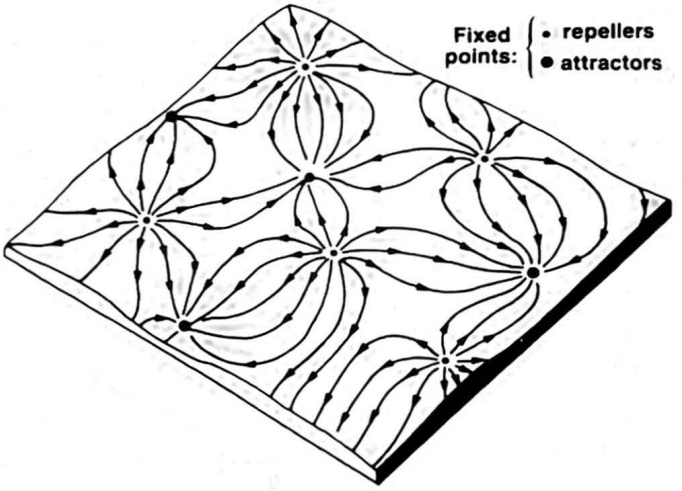


Figure 2.48 A fixed point interpretation of the Hopfield network.

in the form of a content addressable or associative memory. A simple interpretation of the Hopfield network is shown in Figure 2.48, which shows that the recurrent network has a stable set of attractors and repellers, each forming a fixed point in the input space. Every input vector, **x**, is either "attracted" to one of the fixed points or "repelled" from another of the fixed points. The strength of this type of network is its ability to correctly classify "noisy" versions of the patterns that form the stable fixed points.

The third popular type of neural network topology is the Kohonen, self-organizing feature map, which is a clustering procedure for providing a codebook of stable patterns in the input space that characterize an arbitrary input vector, by a small number of representative clusters. We defer a discussion of this type of network to the next chapter, where the ideas of vector quantization are presented in detail.

2.5.4.3 Network Characteristics

Four model characteristics must be specified to implement an arbitrary neural network:

1. number and type of inputs—The issues involved in the choice of inputs to a neural network are similar to those involved in the choice of features for any pattern-classification system. They must provide the information required to make the decision required of the network.
2. connectivity of the network—This issue involves the size of the network—that is, the number of hidden layers and the number of nodes in each layer between input and output. There is no good rule of thumb as to how large (or small) such hidden layers must be. Intuition says that if the hidden layers are large, then it will be difficult to train the network (i.e., there will be too many parameters to estimate). Similarly, if the hidden layers are too small, the network may not be able to accurately classify all the desired input patterns. Clearly practical systems must balance these two competing effects.
3. choice of offset—The choice of the threshold, ϕ, for each computational element must be made as part of the training procedure, which chooses values for the interconnection weights (w_{ij}) and the offset, ϕ.
4. choice of nonlinearity—Experience indicates that the exact choice of the nonlinearity, f, is not very important in terms of the network performance. However, f must be continuous and differentiable for the training algorithm to be applicable.

2.5.4.4 Training of Neural Network Parameters

To completely specify a neural network, values for the weighting coefficients and the offset threshold for each computation element must be determined, based on a labeled set of training data. By a labeled training set of data, we mean an association between a set of Q input vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_Q$ and a set of Q output vectors $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_Q$ where $\mathbf{x}_1 \Rightarrow \mathbf{y}_1, \mathbf{x}_2 \Rightarrow \mathbf{y}_2, \ldots, \mathbf{x}_Q \Rightarrow \mathbf{y}_Q$. For multilayer perceptrons a simple iterative, convergent procedure exists for choosing a set of parameters whose value asymptotically approaches a stationary point with a certain optimality property (e.g., a local minimum of the mean squared error, etc.). This procedure, called back propagation learning, is a simple stochastic gradient technique. For a simple, single-layer network, the training algorithm can be realized via the following convergence steps:

Perceptron Convergence Procedure

1. **Initialization**: At time $t = 0$, set $w_{ij}(0)$, ϕ_j to small random values (where w_{ij} are the weighting coefficients connecting i^{th} input node and j^{th} output node and ϕ_j is the offset to a particular computational element, and the w_{ij} are functions of time).
2. **Acquire Input**: At time t, obtain a *new* input $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$ with the desired output, $\mathbf{y}^x = \{y_1^x, y_2^x, \ldots, y_M^x\}$.
3. **Calculate Output**:

$$y_j = f\left(\sum_{i=1}^{N} w_{ij}(t)x_i - \phi_j\right).$$

4. **Adapt Weights**: Update the weights as

$$w_{ij}(t+1) = w_{ij}(t) + \mathcal{T}(t)\,[y_j^x - y_j] \cdot x_i$$

where the "step size" $\mathcal{T}(t)$ satisfies the constraints:

a. $$\lim_{T\to\infty} \sum_{t=1}^{T} \mathcal{T}(t) = \infty$$

b. $$\lim_{T\to\infty} \sum_{t=1}^{T} \mathcal{T}^2(t) < \infty$$

That is, compute the gradient of the error $\sum_{j=1}^{M} (y_j^x - y_j)^2$ in the direction of the weight $w_{ij}(t)$. (A conventional choice of $\mathcal{T}(t)$ is $1/t$.)

5. **Iteration**: Iterate steps 2–4 until:

$$w_{ij}(t+1) = w_{ij}(t), \qquad \forall i,\ t,\ j.$$

The perceptron convergence procedure is a slow, methodical procedure for estimating the coefficients of a system (a classifier as well as a neural network) based on a mean squared error criterion and has been extensively studied for several decades. The algorithm is simple and is guaranteed to converge, in probability, under a restricted set of conditions on $\mathcal{T}(t)$. However, its speed of convergence in many cases is not sufficiently fast. Alternative procedures for estimating neural network coefficients have been used with varying degrees of success.

2.5.4.5 Advantages of Neural Networks

Neural networks have been given serious consideration for a wide range of problems (including speech recognition) for several reasons. These include the following:

1. They can readily implement a massive degree of parallel computation. Because a neural net is a highly parallel structure of simple, identical, computational elements, it should be clear that they are prime candidates for massively parallel (analog or digital) computation.
2. They intrinsically possess a great deal of robustness or fault tolerance. Since the "information" embedded in the neural network is "spread" to every computational element within the network, this structure is inherently among the least sensitive of networks to noise or defects within the structure.
3. The connection weights of the network need not be constrained to be fixed; they can be adapted in real time to improve performance. This is the basis of the concept of adaptive learning, which is inherent in the neural network structure.
4. Because of the nonlinearity within each computational element, a sufficiently large neural network can approximate (arbitrarily closely) any nonlinearity or nonlinear dynamical system. Hence neural networks provide a convenient way of implementing nonlinear transformations between arbitrary inputs and outputs and are often more efficient than alternative physical implementations of the nonlinearity.

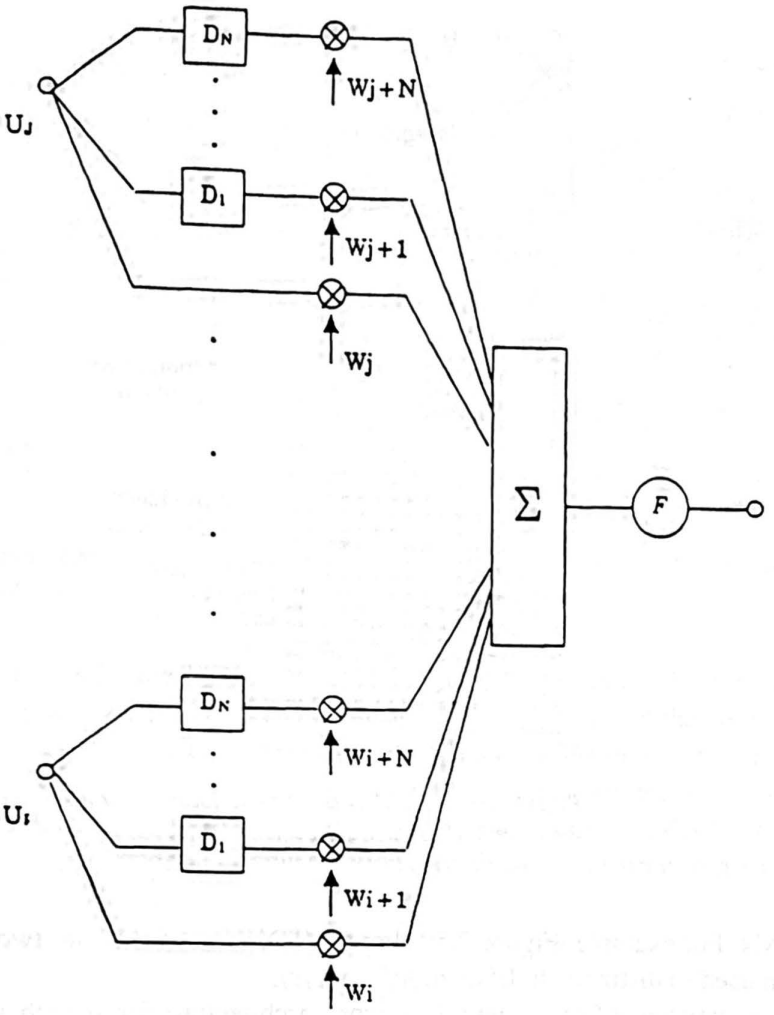


Figure 2.49 The time delay neural network computational element (after Waibel et al. [14]).

2.5.4.6 Neural Network Structures for Speech Recognition

Conventional artificial neural networks are structured to deal with static patterns. As discussed throughout this chapter, speech is inherently dynamic in nature. Hence, some modifications to the simple structures discussed in the previous sections are required for all but the simplest of problems. There is no known correct or proper way of handling speech dynamics within the framework already discussed; however, several reasonable structures have been proposed and studied and we will point out a few such structures in this section.

Perhaps the simplest neural network structure that incorporates speech pattern dynamics is the time delay neural network (TDNN) computation element shown in Figure 2.49 [14]. This structure extends the input to each computational element to include N speech frames (i.e., spectral vectors that cover a duration of $N\Delta$ seconds, where Δ is the time separation between adjacent speech spectra). By expanding the input to N frames (where N is on the order of 15), various types of acoustic-phonetic detectors become practical via

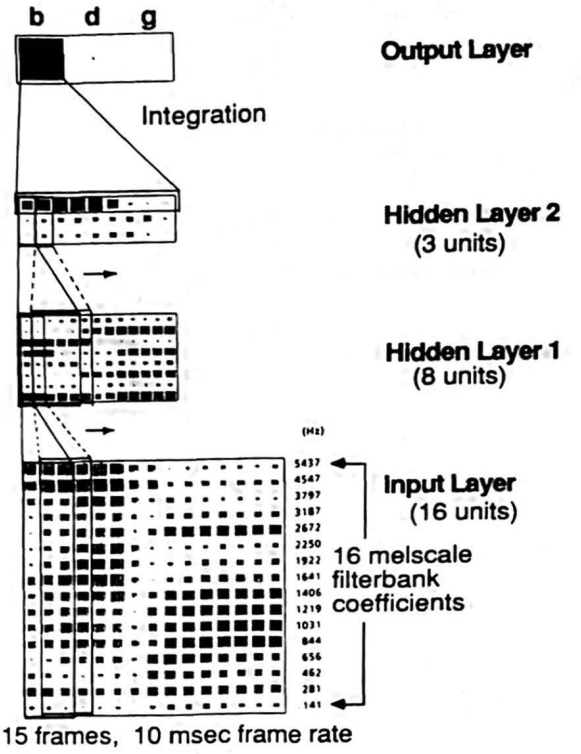


Figure 2.50 A TDNN architecture for recognizing /b/, /d/ and /g/ (after Waibel et al. [14]).

the TDNN. For example, Figure 2.50 shows a TDNN network with two hidden layers that has been used to distinguish /b/ from /d/ from /g/.

A somewhat different neural network architecture for speech recognition, which combines the concept of a matched filter with a conventional neural network to account for the dynamic within speech, is shown in Figure 2.51 [15]. The "acoustic features" of the speech are estimated via conventional neural network architectures; the pattern classifier takes the detected acoustic feature vectors (delayed appropriately) and convolves them with filters "matched" to the acoustic features and sums up the results over time. At the appropriate time (corresponding to the end of some speech unit to be detected or recognized), the output units indicate the presence of the speech.

To illustrate how the network of Figure 2.51 could be used for speech recognition, consider, as shown in Figure 2.52, a "sound" to be recognized that is characterized (in some type of sound lexicon) as the sequence of acoustic features $(\alpha, \epsilon, \delta, \beta, \gamma)$. Assume that this sound is the input to an appropriately designed network of the type shown in Figure 2.51, and the input is as shown in the first line of Figure 2.52. When the acoustic feature α is detected (as indicated by the line labeled $D_\alpha(t)$), it is delayed and then convolved with a matched filter with a long time spreading function, yielding the signal $D_\alpha(t-\tau) \star P_\alpha(\tau)$ as shown in the next line of the figure. Similarly acoustic features ε, δ, β, and γ are detected, delayed appropriately, and convolved with the appropriate matched filter, as shown in the

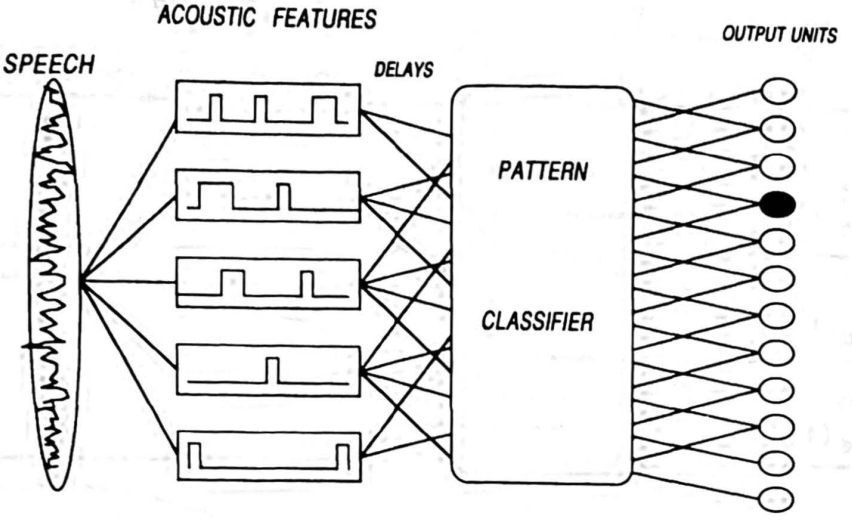


Figure 2.51 A combination neural network and matched filter for speech recognition (after Tank & Hopfield [15]).

succeeding lines in Figure 2.52. Finally, at the end of the sequence, the convolved outputs are summed up and yield a large value, indicating the recognition of the appropriate sound.

Finally, yet a third way of integrating temporal information into a neural network is shown in Figure 2.53. This network is called a hidden control neural network (HCNN) [16] and uses the time varying control, **c**, as a supplement to the standard input, **x**, to allow the network properties (input-output relations) to change over time in a well-prescribed manner.

2.6 SUMMARY

In this chapter we have presented a brief discussion of the basic speech-production/perception mechanism in human beings, and we have illustrated how we can exploit the so-called acoustic-phonetic properties of speech to identify basic sounds. Acoustic-phonetics is the broad underpinning of all speech-recognition work. Differences in approach lie in the degree of reliance on how much acoustic-phonetics can be used in the recognition process. At one extreme is the class of acoustic-phonetic recognition methods that places total reliance on the acoustic-phonetic mapping; at the other extreme is the class of pattern-recognition approaches that do not make a priori assumptions on the phonetic characteristics and instead choose to "relearn" the appropriate acoustic-phonetic mapping for specific word vocabularies and tasks via an appropriately designed training set. Finally, there is the hybrid class of artificial intelligence approaches that exploit, in various degrees, aspects of both extreme views of the speech-recognition process. We also discussed the fundamentals of neural networks, which can be considered a separate structural approach, as well as a new pattern classifier design, with potential to benefit or advance all three classical approaches described in this chapter.

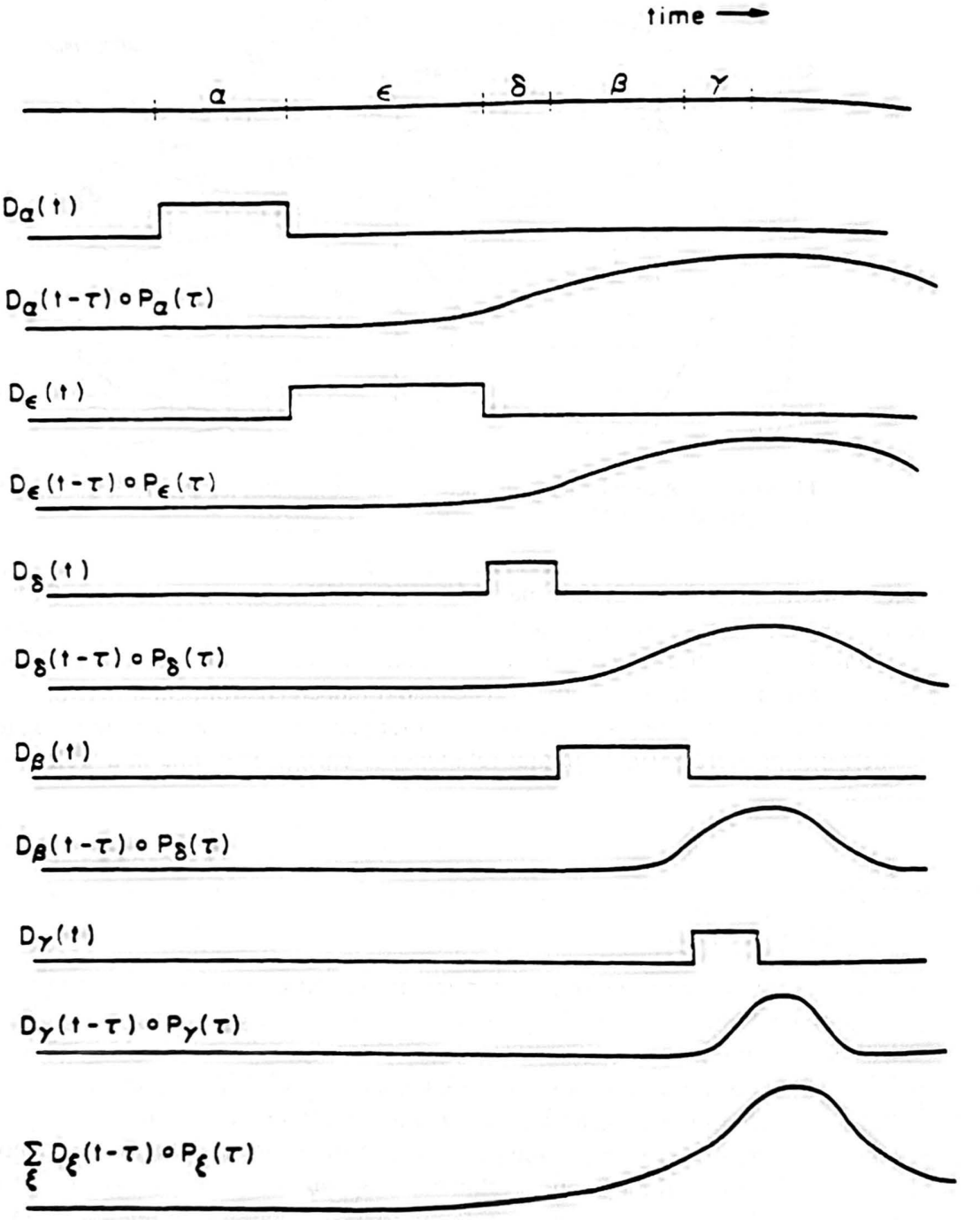


Figure 2.52 Example illustrating the combination of a neural network and a set of matched filters (after Tank & Hopfield [15]).

In the remainder of this book we will primarily discuss aspects of the pattern-recognition approach to speech recognition. However, the alternative methods will always be lurking just below the surface of our discussion.

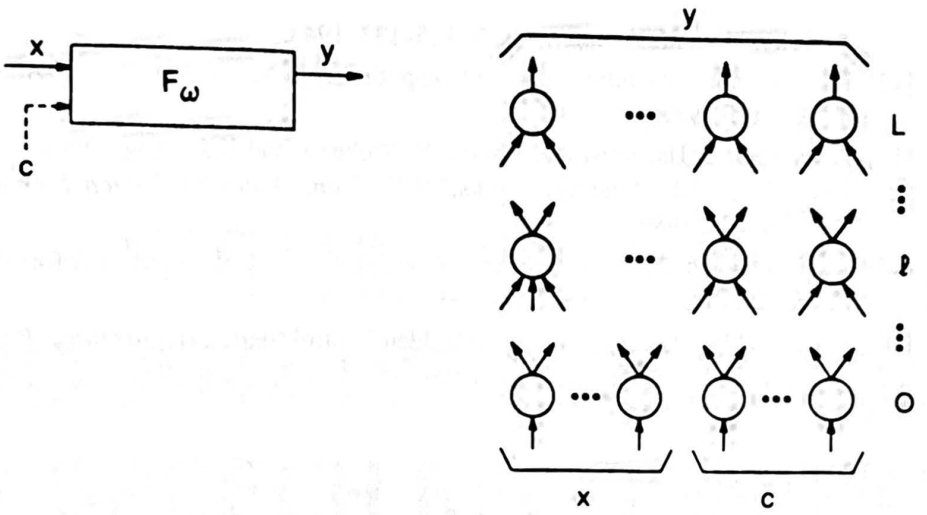


Figure 2.53 The hidden control neural network (after Levin [16]).

REFERENCES

[1] L.R. Rabiner and S.E. Levinson, "Isolated and Connected Word Recognition—Theory and Selected Applications," *IEEE Trans. Communications*, COM-29 (5): 621–659, May 1981.

[2] J.L. Flanagan, C.H. Coker, L.R. Rabiner, R.W. Schafer, and N. Umeda, "Synthetic Voices for Computers," *IEEE Spectrum*, 7 (10): 22–45, October 1970.

[3] J.L. Flanagan, *Speech Analysis, Synthesis, and Perception*, 2nd ed., Springer-Verlag, New York, 1972.

[4] K. Ishizaka and J.L. Flanagan, "Synthesis of Voiced Sounds from a Two-Mass Model of the Vocal Cords," *Bell System Tech. J.*, 50 (6): 1233–1268, July–Aug., 1972.

[5] B.S. Atal and S.L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *J. Acoust. Soc. Am.*, 50 (2): 637-655, August 1971.

[6] J.E. Shoup, "Phonological Aspects of Speech Recognition," 125–138, Ch. 6 in *Trends in Speech Recognition*, W. A. Lea, Ed., Prentice-Hall, Englewood Cliffs, NJ, 1980.

[7] G.E. Peterson and H.L. Barney, "Control Methods Used in a Study of the Vowels," *J. Acoust. Soc. Am.*, 24 (2): 175–194, March 1952.

[8] L.R. Rabiner, "Speech Synthesis by Rule—An Acoustic Domain Approach," PhD. thesis, MIT, Cambridge, MA, June 1967.

[9] A. Holbrook and G. Fairbanks, "Diphthong Formants and Their Movements," *J. Speech Hearing Res.*, 5 (1): 38–58, March 1962.

[10] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth & Brooks, Monterey, CA, 1984.

[11] V.R. Lesser, R.D. Fennell, L.D. Erman, and D.R. Reddy, "Organization of the Hearsay–II Speech Understanding System," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-23 (1): 11–23, 1975.

[12] W.S. McCullough and W.H. Pitts, "A Logical Calculus of Ideas Immanent in Nervous

Activity," *Bull Math Biophysics*, 5: 115–133, 1943.

[13] R. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Mag.*, 4 (2): 4–22, April 1987.

[14] A. Weibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme Recognition Using Time Delay Neural Networks," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-37: 328–339, 1989.

[15] D.W. Tank and J.J. Hopfield, "Neural Computation by Concentrating Information in Time," *Proc. Nat. Academy Sciences*, 84: 1896–1900, April 1987.

[16] E. Levin, "Word Recognition Using Hidden Control Neural Architecture," *Proc. ICASSP 90*, 433–436, Albuquerque, NM, April 1990.

Chapter 3

SIGNAL PROCESSING AND ANALYSIS METHODS FOR SPEECH RECOGNITION

3.1 INTRODUCTION

As discussed in Chapter 1, a speech-recognition system, at its most elementary level, comprises a collection of algorithms drawn from a wide variety of disciplines, including statistical pattern recognition, communication theory, signal processing, combinatorial mathematics, and linguistics, among others. Although each of these areas is relied on to varying degrees in different recognizers, perhaps the greatest common denominator of all recognition systems is the signal-processing front end, which converts the speech waveform to some type of parametric representation (generally at a considerably lower information rate) for further analysis and processing. Because of the singular importance of signal-processing techniques to the understanding of how speech recognizers are designed and how they function, we devote this chapter to a discussion of the most commonly used techniques in this area.

A wide range of possibilities exists for parametrically representing the speech signal. These include the short time energy, zero crossing rates, level crossing rates, and other related parameters. Probably the most important parametric representation of speech is the short time spectral envelope, as discussed in Chapter 2. Spectral analysis methods are therefore generally considered as the core of the signal-processing front end in a speech-recognition system. In this chapter we discuss two dominant methods of spectral analysis—namely, the filter-bank spectrum analysis model, and the linear predictive coding (LPC) spectral analysis model. Also discussed in this chapter is the technique called vector

quantization, which is a procedure for encoding a continuous spectral representation by a "typical" spectral shape in a finite codebook (collection) of spectral shapes, thereby reducing the information rate of the signal processing even further. The technique of vector quantization can be applied to any spectral representation, including both the filter bank and LPC models.

Finally, we close with a brief discussion of an auditory signal-processing model that has been proposed as an alternative to both filter banks and LPC models for speech spectral analysis. The argument for such a model is that, because it is based on known properties of the human auditory system (i.e., a model of cochlea mechanics), it is inherently a better representation of the relevant spectral information than either a filter-bank or an LPC model, and furthermore it should be quite robust to noise and reverberation.

3.1.1 Spectral Analysis Models

To motivate our discussion and see how the signal-processing techniques fit into our canonic recognition system models, let us review the pattern-recognition model of Figure 3.1a and the acoustic-phonetic model of Figure 3.1b. The three basic steps in the pattern-recognition model are (1) parameter measurement (in which a test pattern is created), (2) pattern comparison, and (3) decision making. The function of the parameter measurement block is to represent the relevant acoustic events in the speech signal in terms of a compact, efficient set of speech parameters. Although the choice of which parameters to use is dictated by other considerations (e.g., computational efficiency, type of implementation, available memory), the way in which the chosen representation is computed is based strictly on signal-processing considerations. In a similar manner, in the acoustic-phonetic model of recognition, the first step in the processing is essentially identical to that used in the pattern-recognition approach—namely, parameter measurement—although the steps that follow are markedly different. Hence, it is clear that a good fundamental understanding of the way in which we use signal-processing techniques to implement the parameter-measurement phase of the recognizer is mandatory for understanding the strengths and shortcomings of the various approaches to speech recognition that have been proposed and studied in the literature.

As mentioned previously, the two most common choices of a signal-processing front end for speech recognition are a bank-of-filters model and an LPC model. The overall structure of the bank-of-filters model is shown in Figure 3.2. The speech signal, $s(n)$, (assumed to be in digital form throughout this book), is passed through a bank of Q bandpass filters whose coverage spans the frequency range of interest in the signal (e.g., 100–3000 Hz for telephone-quality signals, 100–8000 Hz for broadband signals). The individual filters can and generally do overlap in frequency, as shown at the bottom of Figure 3.2. The output of the i^{th} bandpass filter, $X_n(e^{j\omega_i})$ (where ω_i is the normalized frequency $2\pi f_i/F_s$, with F_s the sampling frequency) is the short-time spectral representation of the signal $s(n)$, at time n, as seen through the i^{th} bandpass filter with center frequency ω_i. It can readily be seen that in the bank-of-filters model each bandpass filter processes the speech signal independently to produce the spectral representation X_n. The LPC analysis approach, as

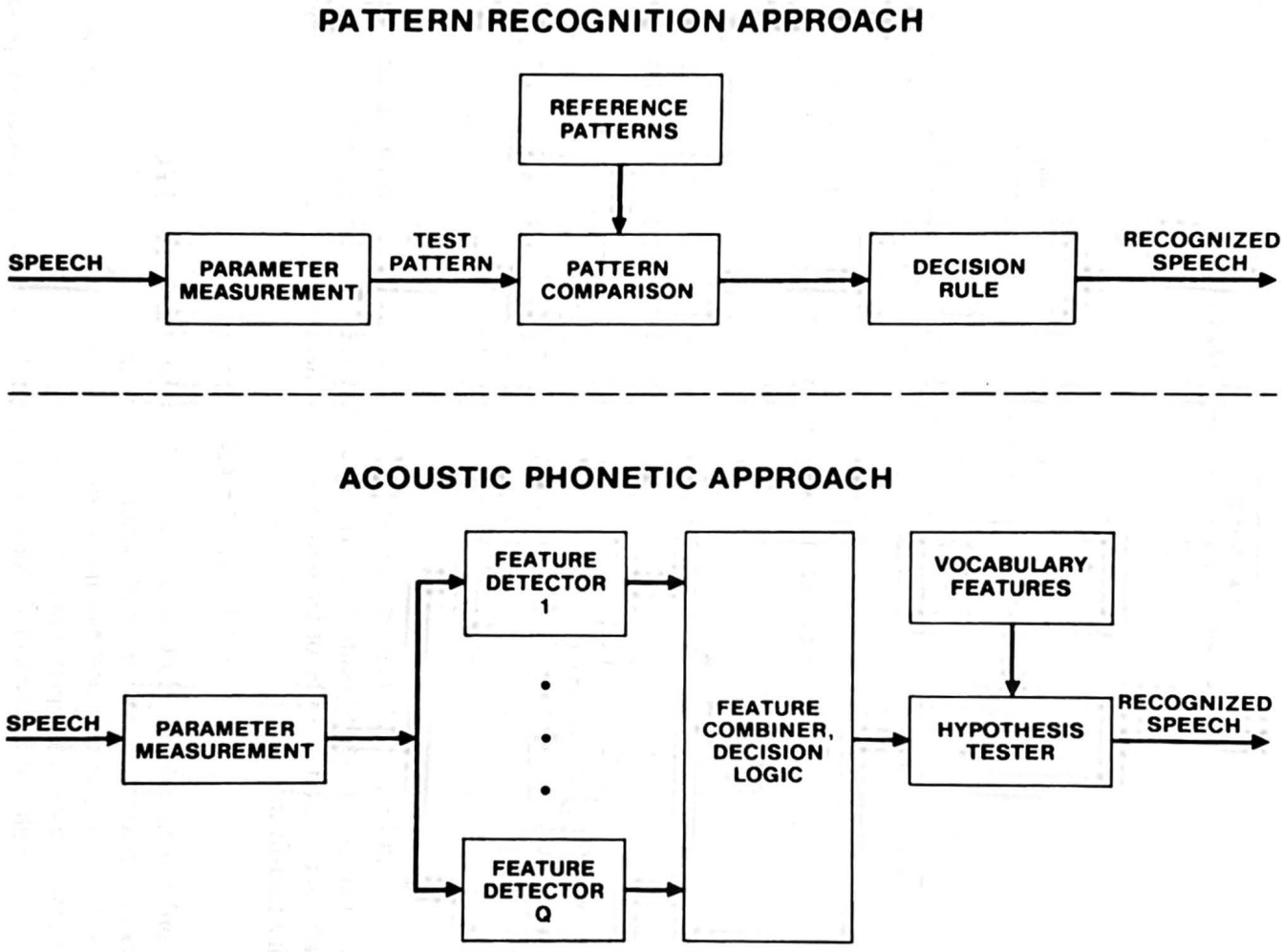
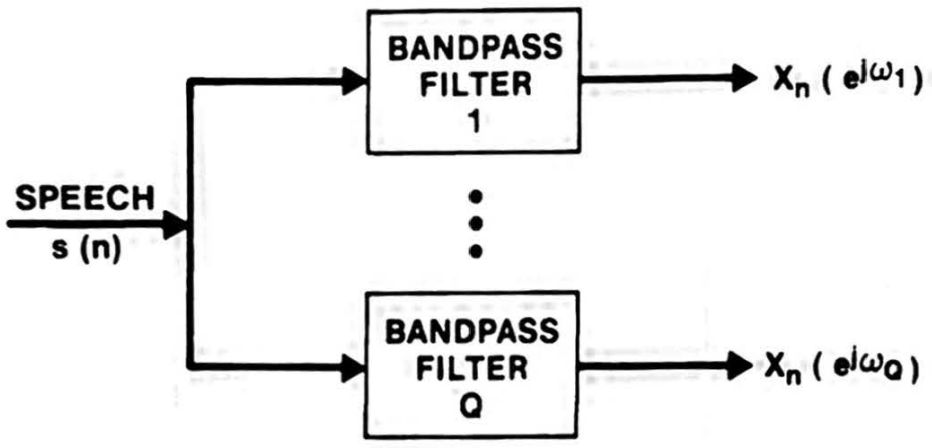


Figure 3.1 (a) Pattern recognition and (b) acoustic phonetic approaches to speech recognition.



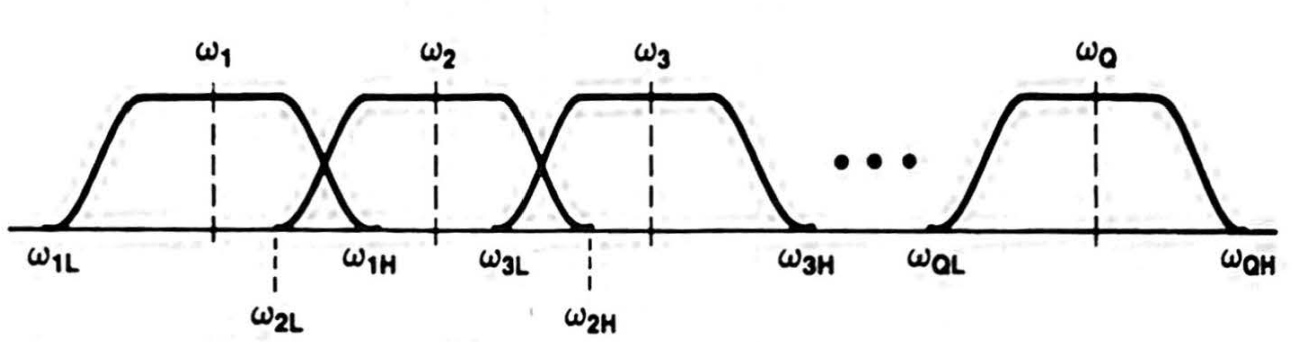


Figure 3.2 Bank-of-filters analysis model.

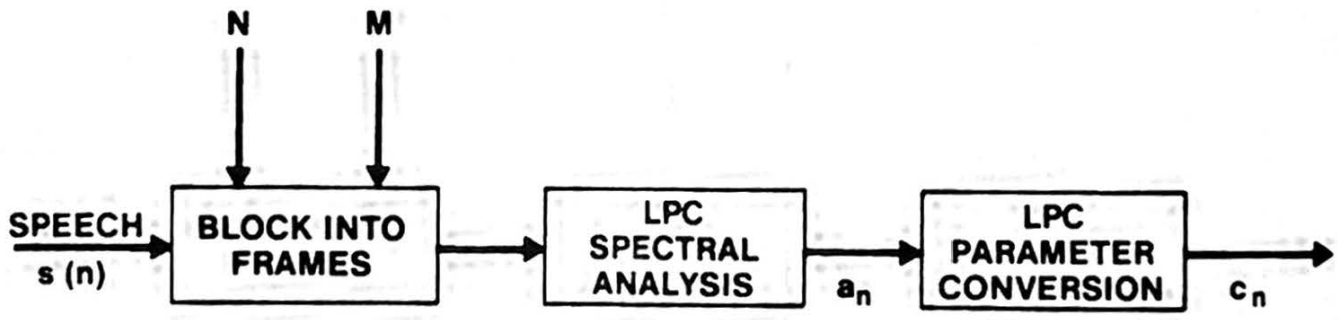


Figure 3.3 LPC analysis model.

illustrated in Figure 3.3, performs spectral analysis on blocks of speech (speech frames) with an all-pole modeling constraint. This means that the resulting spectral representation $X_n(e^{j\omega})$ is constrained to be of the form $\sigma/A(e^{j\omega})$, where $A(e^{j\omega})$ is a p^{th} order polynomial with z-transform

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}.$$

The order, p, is called the LPC analysis order. Thus the output of the LPC spectral analysis block is a vector of coefficients (LPC parameters) that specify (parametrically) the spectrum of an all-pole model that best matches the signal spectrum over the period of time in which the frame of speech samples was accumulated.

Although alternative signal-processing front-end processors have been proposed for speech-recognition systems, the filter-bank and LPC models have proven themselves to give the highest performance in practical speech-recognition systems. Thus, in this chapter, we will discuss these two analysis approaches in greater detail and show how they fit into the framework of the pattern-recognition and acoustic-phonetic approaches to recognition.

3.2 THE BANK-OF-FILTERS FRONT-END PROCESSOR

A block diagram of the canonic structure of a complete filter-bank front-end analyzer is given in Figure 3.4. The sampled speech signal, $s(n)$, is passed through a bank of Q bandpass filters, giving the signals

$$s_i(n) = s(n) * h_i(n), \qquad 1 \le i \le Q \tag{3.1a}$$

$$= \sum_{m=0}^{M_i-1} h_i(m)s(n-m), \tag{3.1b}$$

where we have assumed that the impulse response of the i^{th} bandpass filter is $h_i(m)$ with a duration of M_i samples; hence, we use the convolution representation of the filtering operation to give an explicit expression for $s_i(n)$, the bandpass-filtered speech signal. Since the purpose of the filter-bank analyzer is to give a measurement of the energy of the speech signal in a given frequency band, each of the bandpass signals, $s_i(n)$, is passed through a nonlinearity, such as a full-wave or half-wave rectifier. The nonlinearity shifts the bandpass signal spectrum to the low-frequency band as well as creates high-frequency images. A lowpass filter is used to eliminate the high-frequency images, giving a set of signals, $u_i(n)$, $1 \le i \le Q$, which represent an estimate of the speech signal energy in each of the Q frequency bands.

To more fully understand the effects of the nonlinearity and the lowpass filter, let us assume that the output of the i^{th} bandpass filter is a pure sinusoid at frequency ω_i, i.e.

$$s_i(n) = \alpha_i \sin(\omega_i n). \tag{3.2}$$

This assumption is valid for speech in the case of steady state voiced sounds when the bandwidth of the filter is sufficiently narrow so that only a single speech harmonic is passed by the bandpass filter. If we use a full-wave rectifier as the nonlinearity, that is,

$$\begin{aligned} f(s_i(n)) &= s_i(n) \quad \text{for } s_i(n) \ge 0 \\ &= -s_i(n) \quad \text{for } s_i(n) < 0. \end{aligned} \tag{3.3}$$

Then we can represent the nonlinearity output as

$$v_i(n) = f(s_i(n)) = s_i(n) \cdot w(n), \tag{3.4}$$

where

$$w(n) = \begin{cases} +1 & \text{if } s_i(n) \ge 0 \\ -1 & \text{if } s_i(n) < 0 \end{cases} \tag{3.5}$$

as illustrated in Figure 3.5(a)–(c). Since the nonlinearity output can be viewed as a modulation in time, as shown in Eq. (3.4), then in the frequency domain we get the result that

$$V_i(e^{j\omega}) = S_i(e^{j\omega}) \circledast W(e^{j\omega}), \tag{3.6}$$

where $V_i(e^{j\omega})$, $S_i(e^{j\omega})$, and $W(e^{j\omega})$ are the Fourier transforms of the signals $v_i(n)$, $s_i(n)$ and $w(n)$, respectively, and $\circledast$ is circular convolution. The spectrum $S_i(e^{j\omega})$ is a single impulse at $\omega_0 = \omega_i$, whereas the spectrum $W(e^{j\omega})$ is a set of impulses at the odd-harmonic

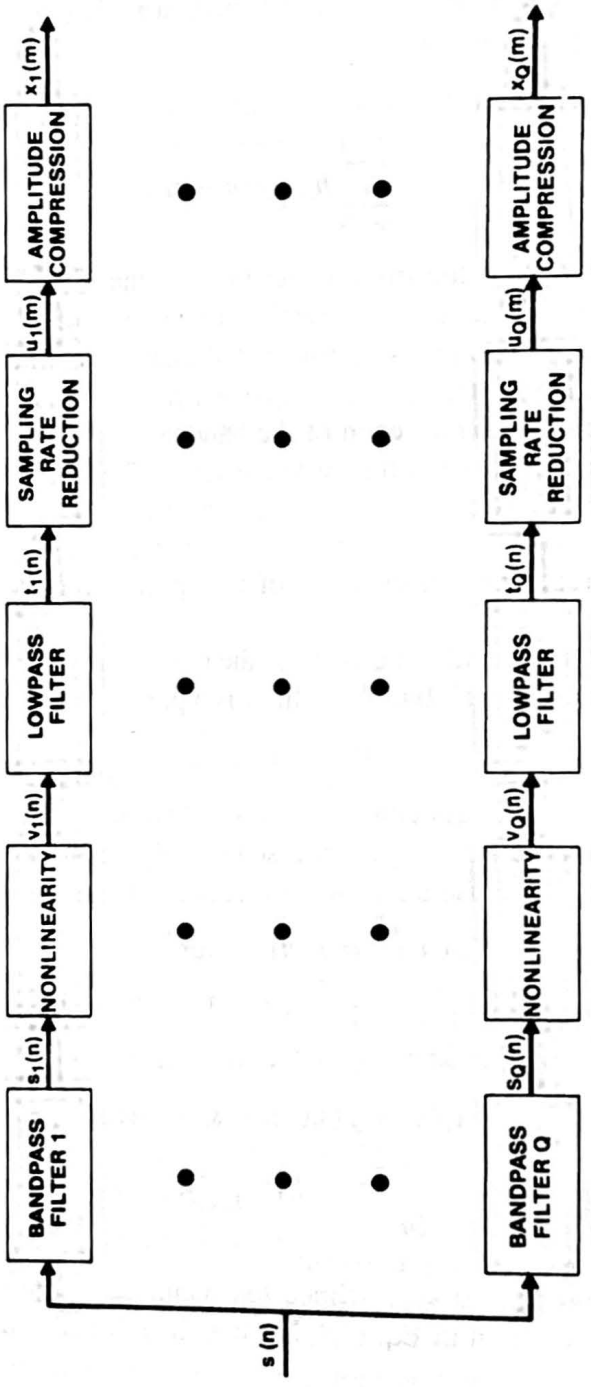


Figure 3.4 Complete bank-of-filters analysis model.

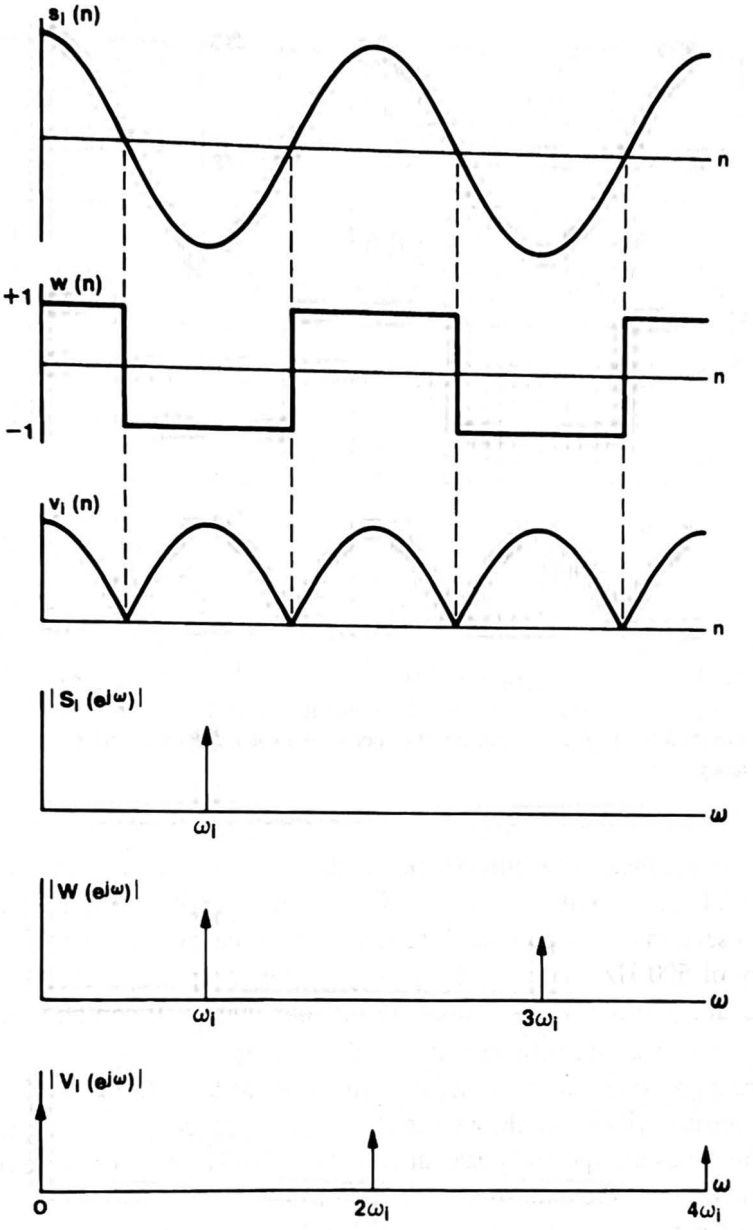


Figure 3.5 Typical waveforms and spectra for analysis of a pure sinusoid in the filter-bank model.

frequencies $\omega_q = \omega_i q$, $q = 1, 3, \ldots, q_{\max}$. Hence the spectrum of $V_i(e^{j\omega})$ is an impulse at $\omega = 0$ and a set of smaller amplitude impulses at $\omega_q = \omega_i q$, $q = 2, 4, 6, \ldots$, as shown in Figure 3.5 (d)–(f). The effect of the lowpass filter is to retain the DC component of $V_i(e^{j\omega})$ and to filter out the higher frequency components due to the nonlinearity.

The above analysis, although only strictly correct for a pure sinusoid, is a reasonably good model for voiced, quasiperiodic speech sounds so long as the bandpass filter is not so wide that it has two or more strong signal harmonics. Because of the time-varying nature of the speech signal (i.e., the quasiperiodicity), the spectrum of the lowpass signal is not a pure

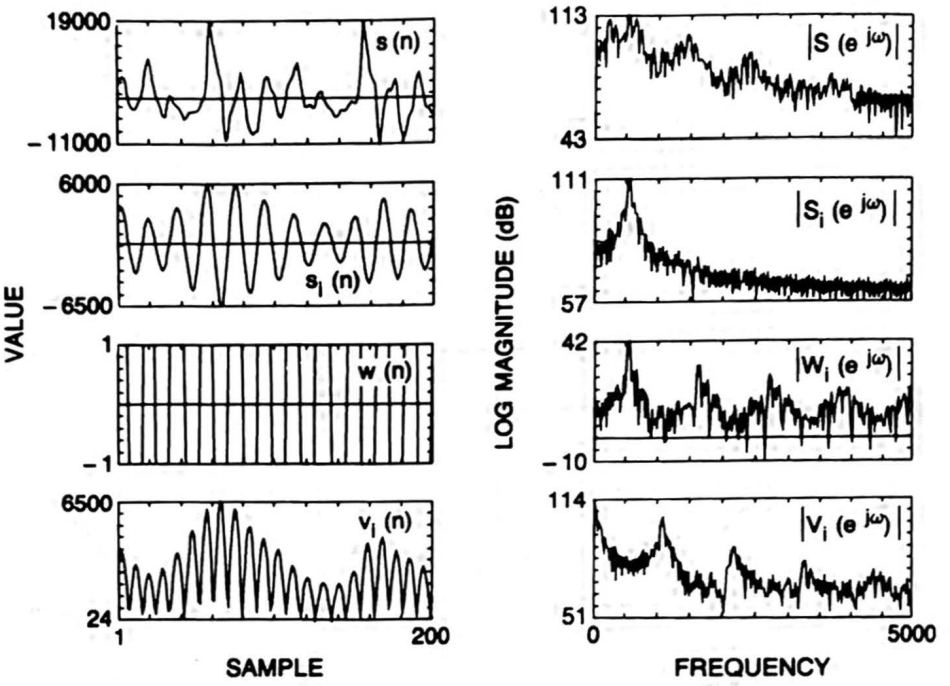


Figure 3.6 Typical waveforms and spectra of a voice speech signal in the bank-of-filters analysis model.

DC impulse, but instead the information in the signal is contained in a low-frequency band around DC. Figure 3.6 illustrates typical waveforms of $s(n)$, $s_i(n)$, $w(n)$ and $v_i(n)$ for a brief (20 msec) section of voiced speech processed by a narrow bandwidth channel with center frequency of 500 Hz (sampling frequency for this example is 10,000 Hz). Also shown are the resulting spectral magnitudes for the four signals. It can be seen that $|S_i(e^{j\omega})|$ has most of its energy around 500 Hz ($\omega = 1000\pi$), whereas $|W_i(e^{j\omega})|$ (which is quasiperiodic) approximates an odd harmonic signal with peaks at 500, 1500, 2500 Hz. The resulting signal spectrum, $|V_i(e^{j\omega})|$, shows the desired low-frequency concentration of energy as well as the undesired spectral peaks at 1000 Hz, 2000 Hz, etc. The role of the final lowpass filter is to eliminate the undesired spectral peaks.

The bandwidth of the signal, $v_i(n)$, is related to the fastest rate of motion of speech harmonics in a narrow band and is generally acknowledged to be on the order of 20–30 Hz. Hence, the final two blocks of the canonic bank-of-filters model of Figure 3.4 are a sampling rate reduction box in which the lowpass-filtered signals, $t_i(n)$, are resampled at a rate on the order of 40–60 Hz (for economy of representation), and the signal dynamic range is compressed using an amplitude compression scheme (e.g., logarithmic encoding, μ-law encoding).

Consider the design of a $Q = 16$ channel filter bank for a wideband speech signal where the highest frequency of interest is 8 kHz. Assume we use a sampling rate of $F_s = 20$ kHz on the speech data to minimize the effects of aliasing in the analog-to-digital conversion. The information (bit rate) rate of the raw speech signal is on the order of 240 kbits per second (20 k samples per second times 12 bits per sample). At the output of

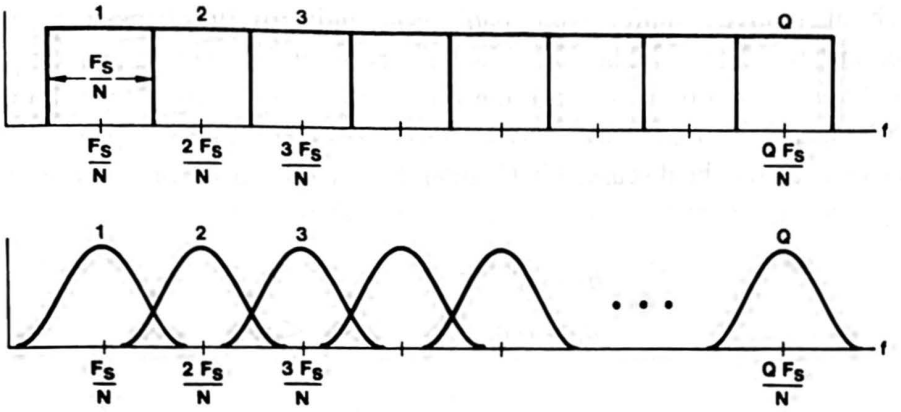


Figure 3.7 Ideal (a) and realistic (b) set of filter responses of a Q-channel filter bank covering the frequency range F_s/N to $(Q + {}^1/_2)F_s/N$.

the analyzer, if we use a sampling rate of 50 Hz and we use a 7 bit logarithmic amplitude compressor, we get an information rate of 16 channels times 50 samples per second per channel times 7 bits per sample, or 5600 bits per second. Thus, for this simple example we have achieved about a 40-to-1 reduction in bit rate, and hopefully such a data reduction would result in an improved representation of the significant information in the speech signal.

3.2.1 Types of Filter Bank Used for Speech Recognition

The most common type of filter bank used for speech recognition is the uniform filter bank for which the center frequency, f_i, of the i^{th} bandpass filter is defined as

$$f_i = \frac{F_s}{N} i, \qquad 1 \le i \le Q, \tag{3.7}$$

where F_s is the sampling rate of the speech signal, and N is the number of uniformly spaced filters required to span the frequency range of the speech. The actual number of filters used in the filter bank, Q, satisfies the relation

$$Q \le N/2 \tag{3.8}$$

with equality when the entire frequency range of the speech signal is used in the analysis. The bandwidth, b_i, of the i^{th} filter, generally satisfies the property

$$b_i \ge \frac{F_s}{N} \tag{3.9}$$

with equality meaning that there is no frequency overlap between adjacent filter channels, and with inequality meaning that adjacent filter channels overlap. (If $b_i < F_s/N$, then certain portions of the speech spectrum would be missing from the analysis and the resulting speech spectrum would not be considered very meaningful.) Figure 3.7a shows a set of Q ideal, non-overlapping, bandpass filters covering the range from $F_s/N({}^1/_2)$ to $(F_s/N)(Q + {}^1/_2)$. Similarly Figure 3.7b shows a more realistic set of Q overlapping filters covering approximately the same range.

The alternative to uniform filter banks is nonuniform filter banks designed according to some criterion for how the individual filters should be spaced in frequency. One commonly used criterion is to space the filters uniformly along a logarithmic frequency scale. (A logarithmic frequency scale is often justified from a human auditory perception point of view, as will be discussed in Chapter 4.) Thus for a set of Q bandpass filters with center frequencies, f_i, and bandwidths, b_i, $1 \leq i \leq Q$, we set

$$b_1 = C \tag{3.10a}$$

$$b_i = \alpha b_{i-1}, \qquad 2 \leq i \leq Q \tag{3.10b}$$

$$f_i = f_1 + \sum_{j=1}^{i-1} b_j + \frac{(b_i - b_1)}{2}, \tag{3.11}$$

where C and f_1 are the arbitrary bandwidth and center frequency of the first filter, and α is the logarithmic growth factor.

The most commonly used values of α are $\alpha = 2$, which gives an octave band spacing of adjacent filters, and $\alpha = 4/3$ which gives a $1/3$ octave filter spacing. Consider the design of a four band, octave-spaced, non-overlapping filter bank covering the frequency band from 200 to 3200 Hz (with a sampling rate of 6.67 kHz). Figure 3.8a shows the ideal filters for this filter bank. Values for f_1 and C of 300 Hz and 200 Hz are used, giving the following filter specifications:

Filter 1: $f_1 = 300$ Hz, $b_1 = 200$ Hz
Filter 2: $f_2 = 600$ Hz, $b_2 = 400$ Hz
Filter 3: $f_3 = 1200$ Hz, $b_3 = 800$ Hz
Filter 4: $f_4 = 2400$ Hz, $b_4 = 1600$ Hz

An example of a 12-band, 1/3-octave, ideal filter-bank specifications, covering the band from about 200 to 3200 Hz, is given in Figure 3.8b. For this example, $C = 50$ Hz, and $f_1 \simeq 225$ Hz.

An alternative criterion for designing a nonuniform filter bank is to use the critical band scale directly. The spacing of filters along the critical band is based on perceptual studies and is intended to choose bands that give equal contribution to speech articulation. The general shape of the critical band scale is given in Figure 3.9. The scale is close to linear for frequencies below about 1000 Hz (i.e., the bandwidth is essentially constant as a function f), and is close to logarithmic for frequencies above 1000 Hz (i.e., the bandwidth is essentially exponential as a function of f). Several variants on the critical band scale have been used, including the mel scale and the bark scale. The differences between these variants are small and are, for the most part, insignificant with regard to design of filter banks for speech-recognition purposes. For example, Figure 3.8c shows a 7-band critical-band filter-bank specification.

Other criteria for designing nonuniform filter banks have been proposed in the literature. For the most part, the uniform and nonuniform designs based on critical band scales have been the most widely used and studied filter-bank methods.

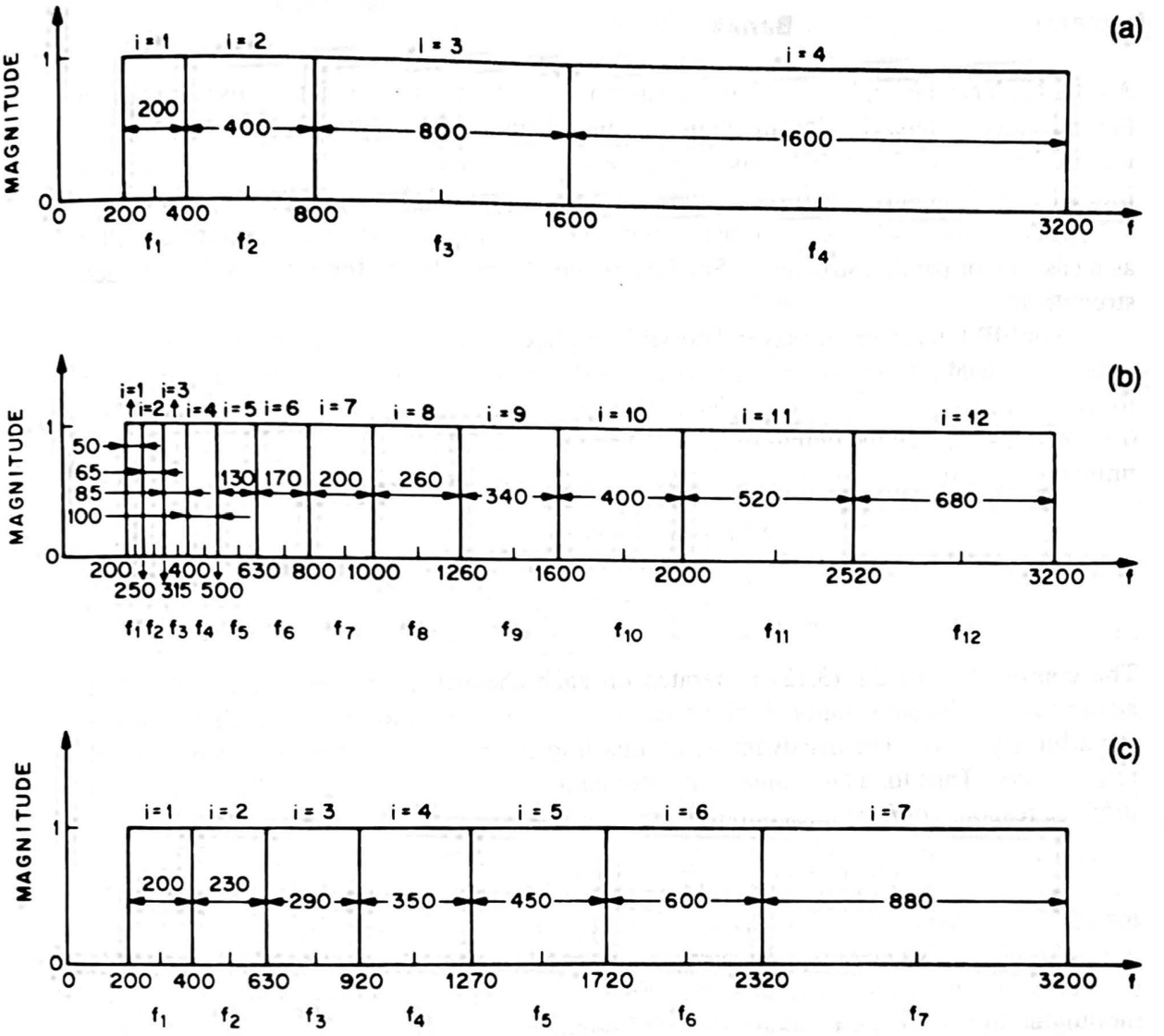


Figure 3.8 Ideal specifications of a 4-channel octave band-filter bank (a), a 12-channel third-octave band filter bank (b), and a 7-channel critical band scale filter bank (c) covering the telephone bandwidth range (200–3200 Hz).

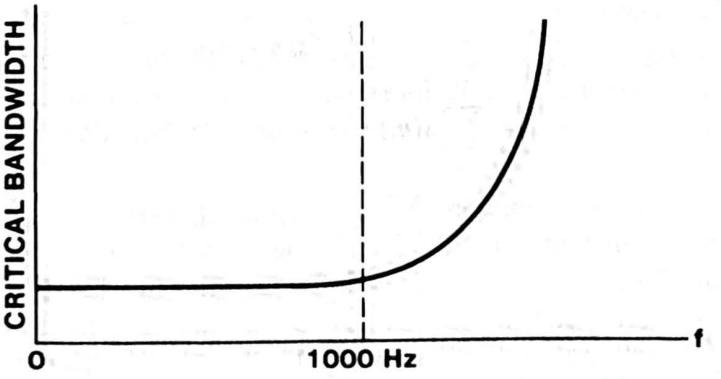


Figure 3.9 The variation of bandwidth with frequency for the perceptually based critical band scale.

3.2.2 Implementations of Filter Banks

A filter bank can be implemented in several ways, depending on the method used to design the individual filters. Design methods for digital filters fall into two broad classes: (1) infinite impulse response (IIR) and (2) finite impulse response (FIR) methods. For IIR filters (also commonly called recursive filters in the literature), the most straightforward, and generally the most efficient implementation is to realize each individual bandpass filter as a cascade or parallel structure. (See Reference [1], pp. 40–46, for a discussion of such structures.)

For FIR filters there are several possible methods of implementing the bandpass filters in the filter bank. The most straightforward and the simplest implementation is the direct form structure. In this case, if we denote the impulse response for the i^{th} channel as $h_i(n)$, $0 \leq n \leq L-1$, then the output of the i^{th} channel, $x_i(n)$, can be expressed as the discrete, finite convolution of the input signal, $s(n)$, with the impulse response, $h_i(n)$, i.e.

$$x_i(n) = s(n) * h_i(n) \tag{3.12a}$$

$$= \sum_{m=0}^{L-1} h_i(m)s(n-m). \tag{3.12b}$$

The computation of Eq. (3.12) is iterated on each channel i, for $i = 1, 2, \ldots, Q$. The advantages of the convolutional, direct form structure are its simplicity and that it works for arbitrary $h_i(n)$. The disadvantage of this implementation is the high computational requirement. Thus for a Q-channel FIR filter bank, where each bandpass FIR filter has an impulse response of L samples duration, we require

$$C_{\text{DFFIR}} = LQ \quad \cdot, + \qquad \text{(multiplication, addition)} \tag{3.13}$$

for a complete evaluation of $x_i(n)$, $i = 1, 2, \ldots, Q$, at a single value of n.

An alternative, less-expensive implementation can be derived for the case in which each bandpass filter impulse response can be represented as a fixed lowpass window, $w(n)$, modulated by the complex exponential, $e^{j\omega_i n}$—that is,

$$h_i(n) = w(n)e^{j\omega_i n}. \tag{3.14}$$

In this case Eq. (2.12b) becomes

$$x_i(n) = \sum_m w(m)e^{j\omega_i m}s(n-m)$$

$$= \sum_m s(m)\, w(n-m)e^{j\omega_i(n-m)}$$

$$= e^{j\omega_i n}\sum_m s(m)w(n-m)e^{-j\omega_i m} \tag{3.15a}$$

$$= e^{j\omega_i n}S_n(e^{j\omega_i}), \tag{3.15b}$$

where $S_n(e^{j\omega_i})$ is the short-time Fourier transform of $s(n)$ at frequency $\omega_i = 2\pi f_i$. The importance of Eq. (3.15) is that efficient procedures often exist for evaluating the short-

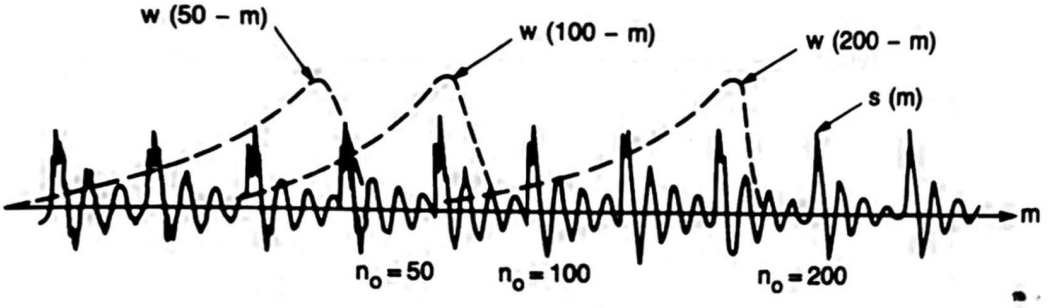


Figure 3.10 The signals $s(m)$ and $w(n-m)$ used in evaluation of the short-time Fourier transform.

time Fourier transform using FFT methods. We will discuss such procedures shortly; first, however, we briefly review the interpretations of the short-time Fourier transform (see Ref. [2] for a more complete discussion of this fascinating branch of signal processing).

3.2.2.1 Frequency Domain Interpretation of the Short-Time Fourier Transform

The short-time Fourier transform of the sequence $s(m)$ is defined as

$$S_n(e^{j\omega_i}) = \sum_m s(m)w(n-m)e^{-j\omega_i m}. \tag{3.16}$$

If we take the point of view that we are evaluating $S_n(e^{j\omega_i})$ for a fixed $n = n_0$, then we can interpret Eq. (3.16) as

$$S_{n_0}(e^{j\omega_i}) = \text{FT}[s(m)w(n_0 - m)]|_{\omega=\omega_i} \tag{3.17}$$

where FT[·] denotes the Fourier Transform. Thus $S_{n_0}(e^{j\omega_i})$ is the conventional Fourier transform of the windowed signal, $s(m)\ w(n_0 - m)$, evaluated at the frequency $\omega = \omega_i$. Figure 3.10 illustrates the signals $s(m)$ and $w(n-m)$, at times $n = n_0 = 50$, 100, and 200 to show which parts of $s(m)$ are used in the computation of the short-time Fourier transform. Since $w(m)$ is an FIR filter (i.e., of finite size), if we denote that size by L, then using the conventional Fourier transform interpretation of $S_n(e^{j\omega_i})$, we can state the following:

1. If L is large, relative to the signal periodicity (pitch), then $S_n(e^{j\omega_i})$ gives good frequency resolution. That is, we can resolve individual pitch harmonics but only roughly see the overall spectral envelope of the section of speech within the window.
2. If L is small relative to the signal periodicity, then $S_n(e^{j\omega_i})$ gives poor frequency resolution (i.e., no pitch harmonics are resolved), but a good estimate of the gross spectral shape is obtained.

To illustrate these points, Figures 3.11–3.14 show examples of windowed signals, $s(m)w(n-m)$, (part a of each figure) and the resulting log magnitude short time spectra, $20 \log_{10} |S_n(e^{j\omega})|$ (part b of each figure). Figure 3.11 shows results for an $L = 500$-point Hamming window applied to a section of voiced speech. The periodicity of the signal is clearly seen in the windowed time waveform, as well as in the short-time spectrum in which the fundamental frequency and its harmonics show up as narrow peaks at equally spaced

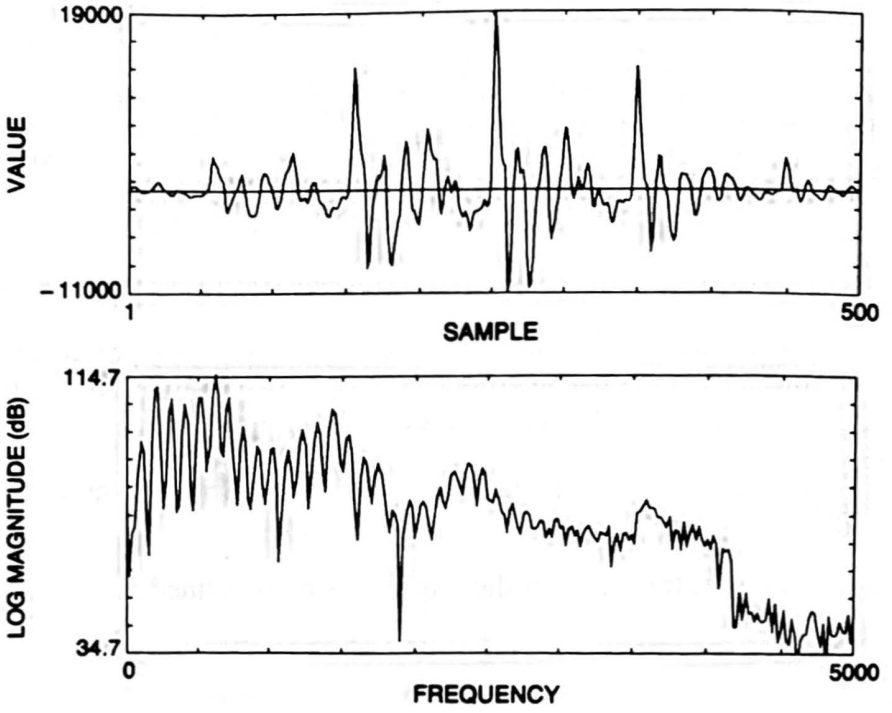


Figure 3.11 Short-time Fourier transform using a long (500 points or 50 msec) Hamming window on a section of voiced speech.

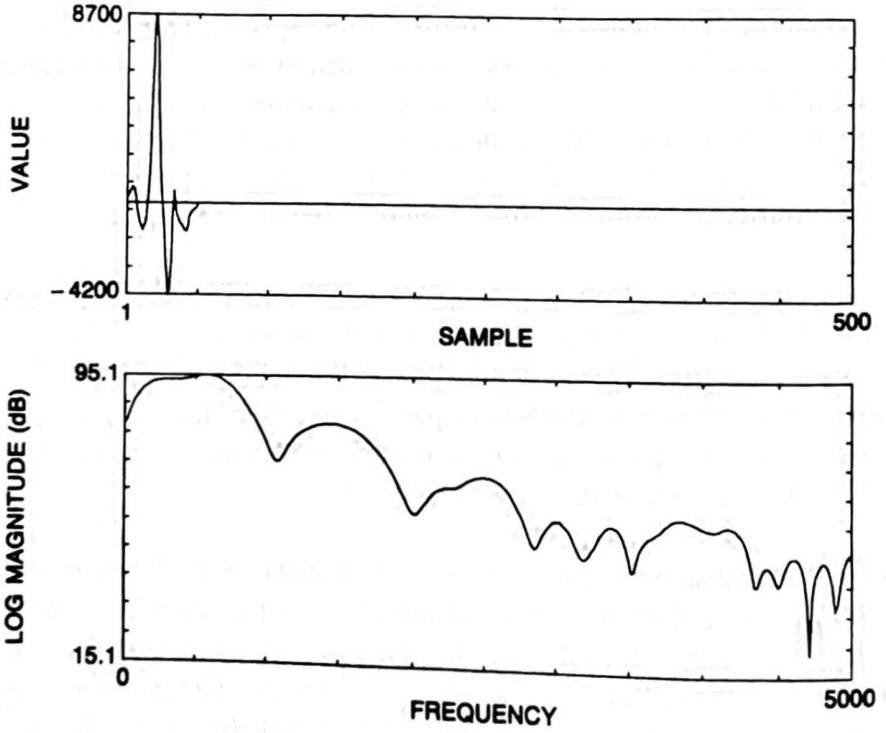


Figure 3.12 Short-time Fourier transform using a short (50 points or 5 msec) Hamming window on a section of voiced speech.

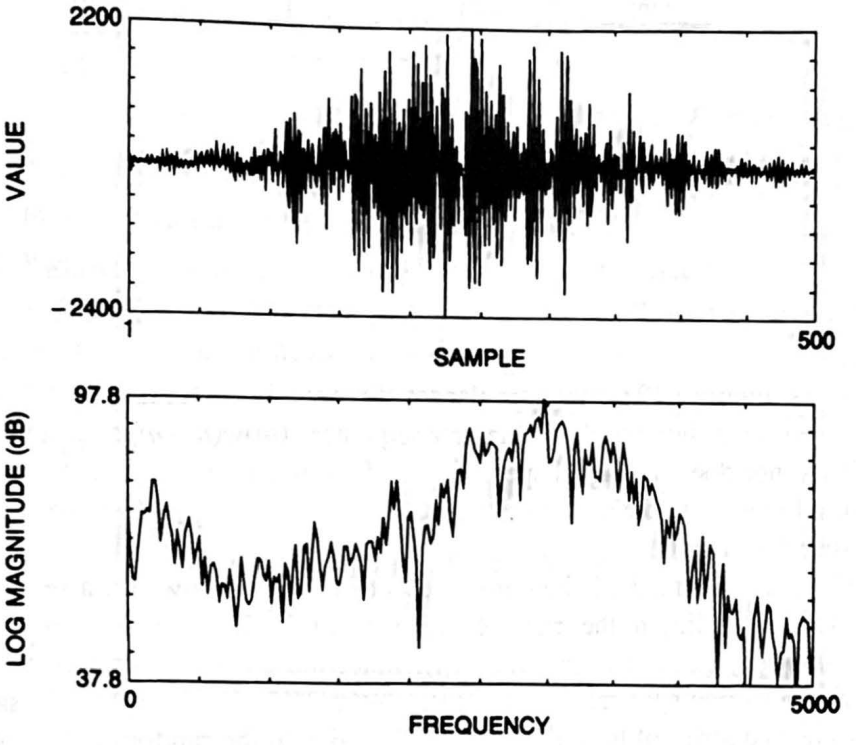


Figure 3.13 Short-time Fourier transform using a long (500 points or 50 msec) Hamming window on a section of unvoiced speech.

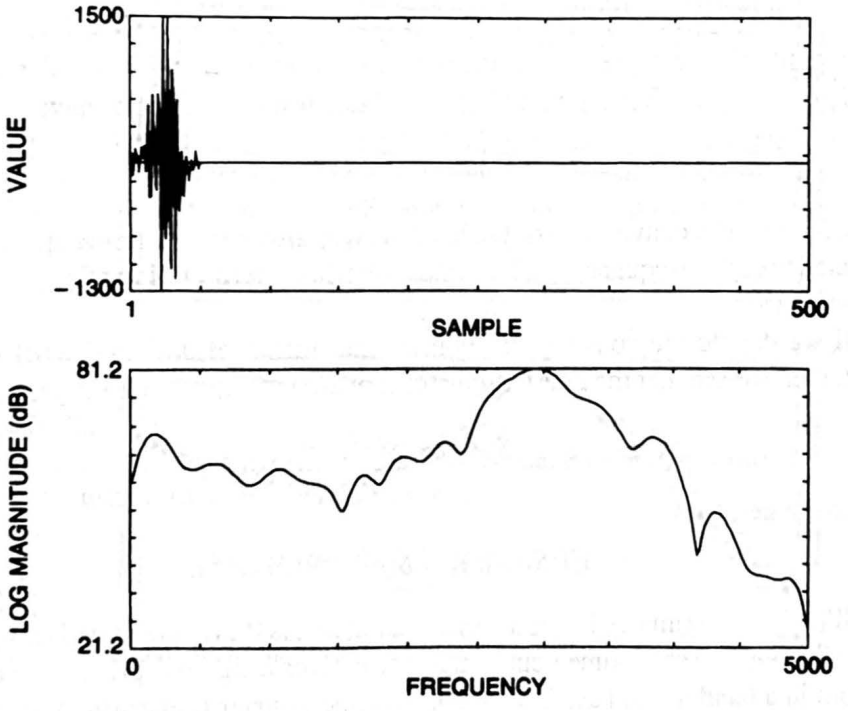


Figure 3.14 Short-time Fourier transform using a short (50 points or 5 msec) Hamming window on a section of unvoiced speech.

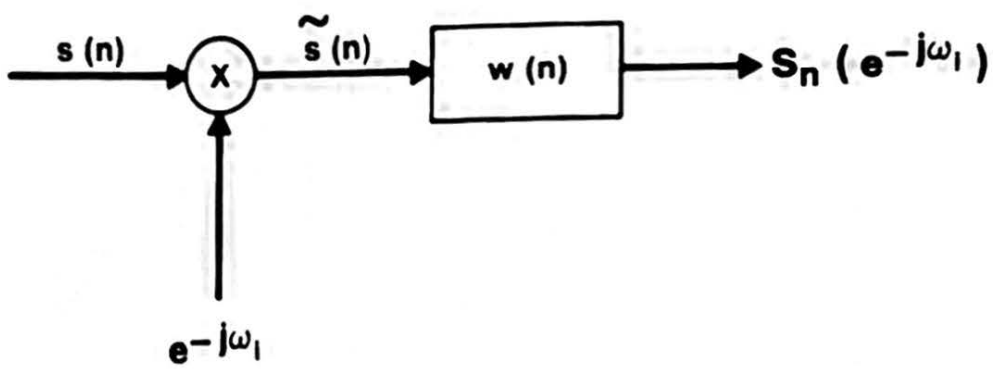


Figure 3.15 Linear filter interpretation of the short-time Fourier transform.

frequencies. Figure 3.12 shows a similar set of comparisons for an $L = 50$-point Hamming window. For such short windows, the time sequence $s(m)w(n-m)$ does not show the signal periodicity, nor does the signal spectrum. In fact, what we see in the short-time Fourier transform log magnitude is a few rather broad peaks in frequency corresponding roughly to the speech formants.

Figures 3.13 and 3.14 show the effects of using windows on a section of unvoiced speech (corresponding to the fricative /sh/) for an $L = 500$ sample window (Figure 3.13) and $L = 50$ sample window (Figure 3.14). Since there is no periodicity in the signal, the resulting short-time spectral magnitude of Figure 3.13, for the $L = 500$ sample window shows a ragged series of local peaks and valleys due to the random nature of the unvoiced speech. Using the shorter window smoothes out the random fluctuations in the short-time spectral magnitude and again shows the broad spectral envelope very well.

3.2.2.2 Linear Filtering Interpretation of the Short-Time Fourier Transform

The linear filtering interpretation of the short-time Fourier transform is derived by considering $S_n(e^{j\omega_i})$, of Eq. (3.16), for fixed values of ω_i, in which case we have

$$S_n(e^{j\omega_i}) = s(n)e^{-j\omega_i n} \circledast w(n). \tag{3.18}$$

That is, $S_n(e^{j\omega_i})$ is a convolution of the lowpass window, $w(n)$, with the speech signal, $s(n)$, modulated to center frequency ω_i. This linear filtering interpretation of $S_n(e^{j\omega_i})$ is illustrated in Figure 3.15.

If we denote the conventional Fourier transforms of $s(n)$ and $w(n)$ by $S(e^{j\omega})$ and $W(e^{j\omega})$, then we see that the Fourier transform of $\tilde{s}(n)$ of Figure 3.15 is just

$$\tilde{S}(e^{j\omega}) = S(e^{j(\omega+\omega_i)}) \tag{3.19}$$

and thus we get

$$\mathrm{FT}(S_n(e^{j\omega_i})) = S(e^{j(\omega+\omega_i)})W(e^{j\omega}). \tag{3.20}$$

Since $W(e^{j\omega})$ approximates 1 over a narrow band, and is 0 everywhere else, we see that, for fixed values, ω_i, the short-time Fourier transform gives a signal representative of the signal spectrum in a band around ω_i. Thus the short-time Fourier transform, $S_n(e^{j\omega_i})$, represents the signal spectral analysis at frequency ω_i by a filter whose bandwidth is that of $W(e^{j\omega})$.

3.2.2.3 Review Exercises

Exercise 3.1

A speech signal is sampled at a rate of 20,000 samples per second ($F_s = 20$ kHz). A 20-msec window is used for short-time spectral analysis, and the window is moved by 10 msec in consecutive analysis frames. Assume that a radix-2 FFT is used to compute DFTs.

1. How many speech samples are used in each segment?
2. What is the frame rate of the short-time spectral analysis?
3. What size DFT and FFT are required to guarantee that no time-aliasing will occur?
4. What is the resulting frequency resolution (spacing in Hz) between adjacent spectral samples?

Solution 3.1

1. Twenty msec of speech at the rate of 20,000 samples per second gives

$$20 \times 10^{-3} \text{ sec} \times 20{,}000 \text{ samples/sec} = 400 \text{ samples}.$$

Each section of speech is 400 samples in duration.

2. Since the shift between consecutive speech frames is 10 msec (i.e., 200 samples at a 20,000 samples/sec rate), the frame rate is

$$\text{frame rate} = \frac{1}{\text{frame shift}} = \frac{1}{10 \times 10^{-3} \text{ sec}} = 100/\text{sec}.$$

That is, 100 spectral analyses are performed per second of speech.

3. To avoid time aliasing in using the DFT to evaluate the short-time Fourier transform, we require the DFT size to be at least as large as the frame size of the analysis frame. Hence, from part 1, we require at least a 400-point DFT. Since we are using a radix 2 FFT, we require, in theory, a 512-point FFT (the smallest power of 2 greater than 400) to compute the DFT without time aliasing. (We would use the 400 speech samples as the first 400 points of the 512-point array; we pad 112 zero-valued samples to the end of the array to fill in and give a 512-point array.) Since the speech signal is real (as opposed to complex), we can use an FFT size of 256 by appropriate signal preprocessing and postprocessing with a complex FFT algorithm.

4. The frequency resolution of the analysis is defined as

$$\text{frequency resolution} = \frac{\text{sampling rate}}{\text{DFT size}} = \frac{20{,}000 \text{ Hz}}{512} \cong 39 \text{ Hz}.$$

Exercise 3.2

If the sequences $s(n)$ and $w(n)$ have normal (long-time) Fourier transforms $S(e^{j\omega})$ and $W(e^{j\omega})$, then show that the short-time Fourier transform

$$S_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} s(m)w(n-m)e^{-j\omega m}$$

can be expressed in the form

$$S_n(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(e^{j\theta})e^{j\theta n}S(e^{j(\omega+\theta)})d\theta.$$

That is, $S_n(e^{j\omega})$ is a smoothed (by the window spectrum) spectral estimate of $S(e^{j\omega})$ at frequency ω.

Solution 3.2

The long-time Fourier transforms of $s(n)$ and $w(n)$ can be expressed as

$$S(e^{j\omega}) = \sum_{n=-\infty}^{\infty} s(n)e^{-j\omega n}$$

$$W(e^{j\omega}) = \sum_{n=-\infty}^{\infty} w(n)e^{-j\omega n}.$$

The window sequence, $w(n)$, can be recovered from its long-time Fourier transform via the integration

$$w(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(e^{j\omega})e^{j\omega n}d\omega.$$

Hence, the short-time Fourier transform

$$S_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} s(m)w(n-m)e^{-j\omega m}$$

can be put in the form (by substituting for $w(n-m)$):

$$S_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} s(m)\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} W(e^{j\theta})e^{j\theta(n-m)}d\theta\right]e^{-j\omega m}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} W(e^{j\theta})e^{j\theta n}\left[\sum_{m=-\infty}^{\infty} s(m)e^{-j(\omega+\theta)m}\right]d\theta$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} W(e^{j\theta})e^{j\theta n}S(e^{j(\omega+\theta)})d\theta.$$

Exercise 3.3

If we define the short-time spectrum of a signal in terms of its short-time Fourier transform as

$$X_n(e^{j\omega}) = \left|S_n(e^{j\omega})\right|^2$$

and we define the short-time autocorrelation of the signal as

$$R_n(k) = \sum_{m=-\infty}^{\infty} w(n-m)s(m)w(n-k-m)\,s(m+k)$$

then show that for

$$S_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} s(m)w(n-m)e^{-j\omega m}$$

$R_n(k)$ and $X_n(e^{j\omega})$ are related as a normal (long-time) Fourier transform pair. In other words, show that $X_n(e^{j\omega})$ is the (long-time) Fourier transform of $R_n(k)$, and vice versa.