SPEECH SYNTHESIS AND RECOGNITION

2ND EDITION

John Holmes
and Wendy Holmes

Speech Synthesis and Recognition

Speech Synthesis and Recognition

Second Edition

John Holmes and Wendy Holmes

TAYLOR & FRANCIS
Founded 1798

London and New York

First edition by the late Dr J.N.Holmes published 1988 by Van Nostrand Reinhold
Second edition published 2001 by Taylor & Francis
11 New Fetter Lane, London EC4P 4EE

Simultaneously published in the USA and Canada
by Taylor & Francis
29 West 35th Street, New York, NY 10001

Taylor & Francis is an imprint of the Taylor & Francis Group

This edition published in the Taylor & Francis e-Library, 2003.

© 2001 Wendy J.Holmes

Publisher's Note
This book has been prepared from camera-ready copy provided by the authors.

All rights reserved. No part of this book may be reprinted or reproduced or utilised in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying and recording, or in any information storage or retrieval system, without permission in writing from the publishers.

Every effort has been made to ensure that the advice and information in this book is true and accurate at the time of going to press. However, neither the publisher nor the authors can accept any legal responsibility or liability for any errors or omissions that may be made. In the case of drug administration, any medical procedure or the use of technical equipment mentioned within this book, you are strongly advised to consult the manufacturer's guidelines.

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging in Publication Data

Holmes, J.N.
Speech synthesis and recognition/John Holmes and Wendy Holmes.—2nd ed
p.cm.
Includes bibliographical references and index.
ISBN 0-7484-0856-8 (hc.)—ISBN 0-7484-0857-6 (pbk.)
1. Speech processing systems. I.Holmes, Wendy (Wendy J.) II. Title.

TK77882.S65 H64 2002
006.4'54–dc21

2001044279

ISBN 0-203-48468-1 Master e-book ISBN

ISBN 0-203-79292-0 (Adobe eReader Format)
ISBN 0-7484-0856-8 (hbk)
ISBN 0-7484-0857-6 (pbk)

CONTENTS

PREFACE TO THE FIRST EDITION

As information technology continues to make more impact on many aspects of our daily lives, the problems of communication between human beings and information-processing machines become increasingly important. Up to now such communication has been almost entirely by means of keyboards and screens, but there are substantial disadvantages of this method for many applications. Speech, which is by far the most widely used and natural means of communication between people, is at first sight an obvious substitute. However, this deceptively simple means of exchanging information is, in fact, extremely complicated. Although the application of speech in the man-machine interface is growing rapidly, in their present forms machine capabilities for generating and interpreting speech are still a travesty of what a young child can achieve with ease. This volume sets out to explain why the problem is so difficult, how it is currently being tackled, and what is likely to happen in the future as our knowledge and technological capability improve. It does not attempt to cover the human factors aspects of using speech for the man-machine interface, as this is another specialism.

This book is intended as an introduction to and summary of the current technology of speech synthesis and recognition. It is most appropriate as a text for those graduate students or specialist undergraduates in information technology who have an electronic engineering or computer science background. Although the book should be useful for people trained in other disciplines, such as linguistics or psychology, some additional reading on signal processing mathematics and electronic technology would probably be necessary for such readers to derive the maximum benefit.

This volume should also be suitable as background material for electronic engineers in industry who need to apply their skills to engineering speech technology products, and for systems engineers who wish to use speech technology devices in complete information processing systems.

An advanced mathematical ability is not required, although it is assumed the reader has some familiarity with the basic mathematics of electronic engineering, such as Fourier analysis, convolution, continuous and discrete-time filters, etc. No specialist knowledge of phonetics or of the properties of speech signals is assumed. Chapter 8, which describes the application of hidden Markov models to speech recognition, requires some statistics knowledge—in particular elementary probability theory and the properties of the normal distribution. I believe that for those trying to understand hidden Markov models for the first time, a large part of the problem arises from the difficulty of remembering the meanings of the symbols used in the equations. For this reason the symbols adopted in Chapter 8 are different from those used almost universally in research papers in this field. Instead they have been made to have some mnemonic association with the quantities they describe. Once the form of the equations has become familiar and their significance is understood, the reader should have no difficulty in transferring to the standard notation, using *a, b, a* and *ß*.

Although this book explains some of the basic concepts in great detail, a volume of this size cannot hope to give a comprehensive coverage of speech synthesis and recognition. It should, however, provide sufficient information to enable the reader to understand many of the current research papers in this area. The subjects described owe a lot to numerous published papers by many different authors over the last 50 years. Study of these papers should not be necessary for the reader to follow the explanations given here, and to simplify the text only a few important original sources of some of the subjects have been referenced directly. However, in Chapter 11 there is a bibliography containing sufficient information to enable readers with more specialist interests to trace all the significant literature in any of the fields covered.

For much of my knowledge in the subjects covered in this book, I am greatly indebted to all of my many colleagues and other associates in the field of speech research, particularly during the long period I was in the Joint Speech Research Unit. In preparing the book I have received much useful advice and detailed help from Andy Downton, in his role as a series editor. I also wish to express my special gratitude to Norman Green, Wendy Holmes, Martin Russell and Nigel Sedgwick, who made valuable constructive comments on drafts of various chapters.

John Holmes, 1988

PREFACE TO THE SECOND EDITION

Since the first publication of *Speech Synthesis and Recognition* there has been a huge growth in the number and diversity of successful applications of speech technology. This increase in deployment of the technology has been made possible by development of the algorithms, supported by general advances in the processing power and memory provided by modern computers.

Like the original version, this new edition of the book aims to provide an easy-to-read introduction to the field of speech technology that is suitable both for students and for professional engineers. While there are now many other textbooks available, these tend to be more in-depth and more mathematically demanding, whereas here the emphasis is on explaining the principles behind the techniques used in speech synthesis and recognition. This book will hopefully provide the reader with a thorough grounding, from which it would then be easier to tackle the more advanced texts and many research papers. While the original version of this book is of course now very out of date, feedback we received suggested that it had been useful as a compact yet thorough introduction. We have therefore followed the same style and format as were used in the first edition, but have tried with this new edition to bring it up to date to reflect the many advances that have been made in the past 10 years or so. All the chapters in the original book have been updated and the new edition is longer by six chapters in order to incorporate new material.

In the area of speech synthesis, there have been significant recent advances in concatenative synthesis techniques. The material on speech synthesis has therefore now been split into three chapters, dealing separately with concatenative methods of speech generation (Chapter 5), phonetic synthesis by rule (Chapter 6) and systems for synthesis from text or from concept (Chapter 7).

In speech recognition, statistical pattern matching is still the basis for the most successful systems, and the underlying principles have not changed. Thus, as with the first edition, the material on speech recognition begins with introductions to template matching (now Chapter 8) and to hidden Markov models (HMMs) for statistical pattern matching (now Chapter 9). New chapters have been added to cover the extensive refinements and developments of the techniques that have been made in recent years. Chapter 10 provides an introduction to methods for front-end analysis, while Chapter 11 covers other methods that are used to achieve good performance in any practical recognition system. Chapter 12 deals specifically with large-vocabulary recognition and includes language modelling. In contrast to the first edition, when describing HMMs the standard notation for forward and backward probabilities has been adopted. Although the symbols used in the first edition were intended to provide a more intuitive connection with the concepts involved, the use of the standard notation should make it easier for the reader subsequently to follow other publications on this subject.

Short chapters have been added to give introductions to the use of neural networks for speech recognition (Chapter 13), and the automatic recognition of speaker attributes such as speaker identity or language (Chapter 14). A further new chapter (Chapter 15) has been added to summarize performance and applications of current speech technology. This chapter is intended to give an indication of the types of applications for which different forms of synthesis and recognition technology are appropriate, and to explain some of the issues in applying these technologies appropriately. However, it has not been possible in a book of this size to do any more than briefly mention related subjects such as the design of the user interface and dialogue management.

As with the first edition, the main text of the book concentrates on explaining concepts and only gives direct references where they were considered essential. Reference sources are presented separately in Chapter 16, with the aim being to give at least one reference for each major topic covered. There is now so much published material available that the references given are necessarily selective, but it is hoped that they can provide a suitable starting point for studying any particular topic in greater depth. A short glossary has also been added to cover some of the main terms used in the book and more generally in speech science and technology.

An influential change since the publication of the first edition of this book is the availability of the Internet as a rich source of reference material. The Web site http://www.speechbook.net has been set up for this book and the intention is to use this site to list corrections and other information relating to the book, as well as to provide up-to-date links to Web sites that are relevant to topics covered in the book.

It was around the end of 1996 when my father found out that the first edition was out of print and would not be reprinted. Although initially he was reluctant to undertake the major revision required to produce a second edition, he gradually changed his mind and developed great enthusiasm for it following encouraging comments from users of the first edition and enlisting a co-author (me!) to help in the updating. In 1998 Taylor & Francis agreed to take the book on, and my father and I spent time planning and organizing the contents and layout before starting on the writing itself. Tackling the project jointly was proving to be a great success, but sadly events did not turn out as we had hoped and my father died in 1999 when we were only part of the way through our task. Since that time I have done my best to complete the book following our joint plan and incorporating the ideas that we had discussed, but I have often had to make decisions without the benefit of input from my father, with his wealth of experience and great eye for detail of wording and presentation. While I am relieved to have finally finished the book, I am sad that he did not live to see the completion of this project to which he was so dedicated.

The material in this book has been drawn from a combination of several published sources together with personal experience. For my own knowledge I owe a lot to my father, and I have also benefited from interaction and discussion with many others working in speech research, especially my past and present colleagues of what was the Speech Research Unit and is now 20/20 Speech Ltd.

Completing the book has turned out to be a difficult and lengthy task, and I am grateful to the many people who have offered me support and encouragement. I would like especially to thank Andy Breen, John Bridle, Dave Carter, Chris Darwin, Phil Green, Roger Moore, Steve Renals, Peter Roach, Martin Russell, Nigel Sedgwick, Tim Thorpe and Mike Tomlinson for careful reading and constructive criticism of drafts of various chapters. I also express my gratitude to Tony Moore and his many colleagues at Taylor & Francis who have been involved at different stages, both for practical help and for their patience and understanding during the long time that it has taken for me to complete the book.

Wendy Holmes, 2001

LIST OF ABBREVIATIONS

A-D	Analogue-to-Digital
ADPCM	Adaptive Differential Pulse Code Modulation
ANN	Artificial Neural Network
APC	Adaptive Predictive Coding
ARPA	Advanced Research Projects Agency
ASR	Automatic Speech Recognition
ATIS	Air Travel Information System
BM	Basilar Membrane
BN	Broadcast News
CELP	Code-Excited Linear Prediction
CH	CallHome
CMN	Cepstral Mean Normalization
CMS	Cepstral Mean Subtraction
CMU	Carnegie-Mellon University
CV	Consonant-Vowel
DARPA	Defense Advanced Research Projects Agency
dB	deciBel
DCT	Discrete Cosine Transform
DP	Dynamic Programming
DPCM	Differential Pulse Code Modulation
DRT	Diagnostic Rhyme Test
DSP	Digital Signal Processor
DTMF	Dual-Tone Multi-Frequency
DTW	Dynamic Time Warping
EER	Equal Error Rate
EIH	Ensemble Interval Histogram
EM	Expectation Maximization
ERB	Equivalent Rectangular Bandwidth
F_0	Fundamental frequency
FD-PSOLA	Frequency-Domain Pitch-Synchronous OverLap-Add
GMM	Gaussian Mixture Model
GPD	Generalized Probabilistic Descent
GSD	Generalized Synchrony Detector
HMM	Hidden Markov Model
HMS	Holmes-Mattingly-Shearme
Hz	Hertz
IPA	International Phonetic Alphabet
ITU	International Telecommunications Union
IVR	Interactive Voice Response
LDA	Linear Discriminant Analysis
LP	Linear Prediction
LPC	Linear Predictive Coding
LP-PSOLA	Linear-Predictive Pitch-Synchronous OverLap-Add
LVCSR	Large-Vocabulary RecognitionContinuous Speech
MAP	Maximum *A Posteriori*
MBE	Multi-Band Excitation
MBROLA	Multi-Band Resynthesis OverLap-Add
MBR-PSOLA	Multi-Band Resynthesis Pitch-Synchronous OverLap-Add

MCE	Minimum Classification Error
MELP	Mixed Excitation Linear Prediction
MFCC	Mel-Frequency Cepstral Coefficient
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MLP	Multi-Layer Perceptron
MMI	Maximum Mutual Information
MOS	Mean Opinion Score
NAB	North American Business news
NIST	National Institute of Standards and Technology
OLA	OverLap-Add
PCA	Principal Components Analysis
PCM	Pulse Code Modulation
p.d.f.	probability density function
PDP	Parallel Distributed Processing
PLP	Perceptual Linear Prediction
PMC	Parallel Model Combination
PSOLA	Pitch-Synchronous Overlap-Add
PSQM	Perceptual Speech Quality Measure
PTC	Psychophysical Tuning Curve
RASTA	RelAtive SpecTrAl
RELP	Residual-Excited Linear Prediction
RM	Resource Management
ROC	Receiver Operating Characteristic
SAT	Speaker-Adaptive Training
SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level
SWB	SWitchBoard
TDNN	Time-Delay Neural Network
TD-PSOLA	Time-Domain Pitch-Synchronous OverLap-Add
TOBI	TOnes and Break Indices
TTS	Text-To-Speech
VC	Vowel-Consonant
VOT	Voice Onset Time
VQ	Vector Quantization
VTLN	Vocal Tract Length Normalization
WER	Word Error Rate
WSJ	*Wall Street Journal*

CHAPTER 1

Human Speech Communication

1.1 VALUE OF SPEECH FOR HUMAN-MACHINE COMMUNICATION

Advances in electronic and computer technology are causing an explosive growth in the use of machines for processing information. In most cases this information originates from a human being, and is ultimately to be used by a human being. There is thus a need for effective ways of transferring information between people and machines, in both directions. One very convenient way in many cases is in the form of speech, because speech is the communication method most widely used between humans; it is therefore extremely natural and requires no special training.

There are, of course, many circumstances where speech is not the best method for communicating with machines. For example, large amounts of text are much more easily received by reading from a screen, and positional control of features in a computer-aided design system is easier by direct manual manipulation. However, for interactive dialogue and for input of large amounts of text or numeric data speech offers great advantages. Where the machine is only accessible from a standard telephone instrument there is no practicable alternative.

1.2 IDEAS AND LANGUAGE

To appreciate how communication with machines can use speech effectively, it is important to understand the basic facts of how humans use speech to communicate with each other. The normal aim of human speech is to communicate ideas, and the words and sentences we use are not usually important as such. However, development of intellectual activity and language acquisition in human beings proceed in parallel during early childhood, and the ability of language to code ideas in a convenient form for mental processing and retrieval means that to a large extent people actually formulate the ideas themselves in words and sentences. The use of language in this way is only a convenient coding for the ideas. Obviously a speaker of a different language would code the same concepts in different words, and different individuals within one language group might have quite different shades of meaning they normally associate with the same word.

1.3 RELATIONSHIP BETWEEN WRITTEN AND SPOKEN LANGUAGE

The invention of written forms of language came long after humans had established systems of speech communication, and individuals normally learn to speak long before they learn to read and write. However, the great dependence on written language in modern civilization has produced a tendency for people to consider language primarily in its written form, and to regard speech as merely a spoken form of written text—possibly inferior because it is imprecise and often full of errors. In fact, spoken and written language are different in many ways, and speech has the ability to capture subtle shades of meaning that are quite difficult to express in text, where one's only options are in choice of words and punctuation. Both speech and text have their own characteristics as methods of transferring ideas, and it would be wrong to regard either as an inferior substitute for the other.

1.4 PHONETICS AND PHONOLOGY

The study of how human speech sounds are produced and how they are used in language is an established scientific discipline, with a well-developed theoretical background. The field is split into two branches: the actual generation and classification of speech sounds falls within the subject of **phonetics,** whereas their functions in languages are the concern of **phonology**. These two subjects need not be studied in detail by students of speech technology, but some phonetic and phonological aspects of the generation and use of speech must be appreciated in general terms. The most important ones are covered briefly in this chapter.

1.5 THE ACOUSTIC SIGNAL

The normal aim of a talker is to transfer ideas, as expressed in a particular language, but putting that language in the form of speech involves an extremely complicated extra coding process (Figure 1.1). The actual signal transmitted is predominantly acoustic, i.e. a variation of sound pressure with time. Although particular speech sounds tend to have fairly characteristic properties (better specified in spectral rather than waveform terms), there is great variability in the relationship between the acoustic signal and the linguistic units it represents. In analysing an utterance linguistically the units are generally discrete—e.g. words, phrases, sentences. In speech the acoustic signal is continuous, and it is not possible to determine a precise mapping between time intervals in a speech signal and the words they represent. Words normally join together, and in many cases there is no clear acoustic indication of where one word ends and the next one starts. For example, in "six seals" the final sound of the "six" is not significantly different from the [s] at the beginning of "seals", so the choice of word boundary position will be arbitrary. All else being equal, however, one can be fairly certain that the [s] sound in the middle of "sick seals" will be shorter, and this duration difference will probably be the only reliable distinguishing feature in the acoustic signal for resolving any possible confusion between such pairs of words. The acoustic difference between "sick seals" and "six eels" is likely to be even more subtle.

Although the individual sound components in speech are not unambiguously related to the identities of the words, there is, of course, a high degree of systematic relationship that applies most of the time. Because speech is generated by the human vocal organs (explained further in Chapter 2) the acoustic properties can be related to the positions of the articulators. With sufficient training, phoneticians can, based entirely on listening, describe speech in terms of a sequence of events related to articulatory gestures. This auditory analysis is largely independent of age or sex of the speaker. The **International Phonetic Alphabet (IPA)** is a system of

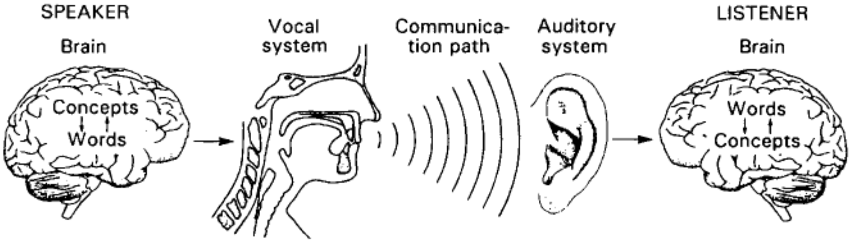


Figure 1.1 Illustration of the processes involved in communicating ideas by speech. It is not easy to separate the concepts in the brain from their representation in the form of language.

notation whereby phoneticians can describe their analysis as a sequence of discrete units. Although there will be a fair degree of unanimity between phoneticians about the transcription of a particular utterance, it has to be accepted that the parameters of speech articulation are continuously variable. Thus there will obviously be cases where different people will judge a particular stretch of sound to be on the opposite sides of a phonetic category boundary.

1.6 PHONEMES, PHONES AND ALLOPHONES

Many of the distinctions that can be made in a narrow phonetic transcription, for example between different people pronouncing the same word in slightly different ways, will have no effect on meaning. For dealing with the power of speech sounds to make distinctions of meaning it has been found useful in phonology to define the **phoneme,** which is the smallest unit in speech where substitution of one unit for another might make a distinction of meaning. For example, in English the words “do” and “to” differ in the initial phoneme, and “dole” and “doll” differ in the middle (i.e. the vowel sound). There may be many different features of the sound pattern that contribute to the phonemic distinction: in the latter example, although the tongue position during the vowel would normally be slightly different, the most salient feature in choosing between the two words would probably be vowel duration. A similar inventory of symbols is used for phonemic notation as for the more detailed phonetic transcription, although the set of phonemes is specific to the language being described. For any one language only a small subset of the IPA symbols is used to represent the phonemes, and each symbol will normally encompass a fair range of phonetic variation. This variation means that there will be many slightly different sounds which all represent manifestations of the same phoneme, and these are known as **allophones**.

Phonologists can differ in how they analyse speech into phoneme sequences, especially for vowel sounds. Some economize on symbols by representing the long vowels in English as phoneme pairs, whereas they regard short vowels as single phonemes. Others regard long and short vowels as different single phonemes, and so need more symbols. The latter analysis is useful for acknowledging the difference in phonetic quality between long vowels and their nearest short counterparts, and will be adopted throughout this book. We will use the symbol set that is most widely used by the current generation of British phoneticians, as found in Wells (2000) for example.

With this analysis there are about 44 phonemes in English. The precise number and the choice of symbols depends on the type of English being described (i.e. some types of English do not make phonetic distinctions between pairs of words that are clearly distinct in others). It is usual to write phoneme symbols between oblique lines, e.g. /t/ , but to use square brackets round the symbols when they represent a particular allophone, e.g. [t]. Sometimes the word **phone** is used as a general term to describe acoustic realizations of a phoneme when the variation between different allophones is not being considered.

Many of the IPA symbols are the same as characters of the Roman alphabet, and often their phonetic significance is similar to that commonly associated with the same letters in those languages that use this alphabet. To avoid need for details of the IPA notation in this book, use of IPA symbols will mostly be confined to characters whose phonemic meaning should be obvious to speakers of English.

There is a wide variation in the acoustic properties of allophones representing a particular phoneme. In some cases these differences are the result of the influence of neighbouring sounds on the positions of the tongue and other articulators. This effect is known as **co-articulation**. In other cases the difference might be a feature that has developed for the language over a period of time, which new users learn as they acquire the language in childhood. An example of the latter phenomenon is the vowel difference in the words “coat” and “coal” as spoken in southern England. These vowels are acoustically quite distinct, and use a slightly different tongue position. However, they are regarded as allophones of the same phoneme because they are never used as alternatives to distinguish between words that would otherwise be identical. Substituting one vowel for the other in either word would not cause the word identity to change, although it would certainly give a pronunciation that would sound odd to a native speaker.

1.7 VOWELS, CONSONANTS AND SYLLABLES

We are all familiar with the names **vowel** and **consonant** as applied to letters of the alphabet. Although there is not a very close correspondence in English between the letters in conventional spelling and their phonetic significance, the categories of vowel and consonant are for the most part similarly distinct in spoken language.

During vowels the flow of air through the mouth and throat is relatively unconstricted and the original source of sound is located at the larynx (see Chapter 2), whereas in most consonants there is a substantial constriction to air flow for some of the time. In some consonants, known as **stop consonants** or **plosives,** the air flow is completely blocked for a few tens of milliseconds. Although speech sounds that are classified as vowels can usually be distinguished from consonants by this criterion, there are some cases where the distinction is not very clear. It is probably more useful to distinguish between vowels and consonants phonologically, on the basis of how they are used in making up the words of a language. Languages show a tendency for vowels and consonants to alternate, and sequences of more than three or four vowels or consonants are comparatively rare. By considering their functions and distributions in the structure of language it is usually fairly easy to decide, for each phoneme, whether it should be classified as a vowel or a consonant.

In English there are many vowel phonemes that are formed by making a transition from one vowel quality to another, even though they are regarded as single phonemes according to the phonological system adopted in this book. Such vowels are known as **diphthongs**. The vowel sounds in “by”, “boy” and “bough” are typical examples, and no significance should be assigned to the fact that one is represented by a single letter and the others by “oy” and “ough”. Vowels which do not involve such a quality transition are known as **monophthongs**.

There are some cases where the different phonological structure will cause phonetically similar sounds to be classified as vowels in one language and consonants in another. For example the English word “pie” and the Swedish word “paj”, which both have the same meaning, also sound superficially rather similar. The main phonetic difference is that at the end of the word the tongue will be closer to the palate in the Swedish version. However, the English word has two phonemes: the initial consonant, followed by a diphthong for the vowel. In contrast the Swedish word has three phonemes: the initial consonant, followed by a monophthong vowel and a final consonant. The final consonant is very similar phonetically to the initial consonant in the English word “yet”.

All spoken languages have a syllabic structure, and all languages permit **syllables** consisting of a consonant followed by a vowel. This universal fact probably originates from the early days of language development many thousands of years ago. The natural gesture of opening the mouth and producing sound at the larynx will always produce a vowel-like sound, and the properties of the acoustic system during the opening gesture will normally generate some sort of consonant. Some languages (such as Japanese) still have a very simple syllabic structure, where most syllables consist of a single consonant followed by a vowel. In languages of this type syllable sequences are associated with alternate increases and decreases of loudness as the vowels and consonants alternate. In many other languages, however, a much wider range of syllable types has evolved, where syllables can consist of just a single vowel, or may contain one or more consonants at the beginning and the end. A syllable can never contain more than one vowel phoneme (although that one may be a diphthong), but sometimes it may not contain any. In the second syllable of many people’s pronunciation of English words such as “button”, “prism” and “little”, the final consonant sound is somewhat lengthened, but is not preceded by a vowel. The articulatory constriction needed for the consonant at the end of the first syllable is followed immediately by that for the final consonant. Other English speakers might produce a short neutral vowel between the two consonants; the end result will sound fairly similar, and will be judged by listeners as containing two syllables in both cases. When the vowel is omitted the final consonant is classified as **syllabic**.

Perception of syllables in a language like English depends on many different factors. A reduction of signal level between two higher-level regions generally implies a syllable boundary, but **pitch** change is often used for separating syllables. For example, in the abbreviations “i.e.” and “I.E.E.” there is no change of vowel quality in the second abbreviation to indicate that “E.E.” is two syllables, and there may be no obvious change of signal level. However, there is usually a noticeable drop of pitch to mark the boundary between the two letters. For any utterance, the decision as to whether there is one E or two will rely on a combination of duration, pitch change and loudness change, and pitch change tends to have most influence.

Problems with determining how many syllables there are in a word occur mainly in the case of words nominally containing sequences of vowel phonemes. Casual pronunciation may sometimes merge vowel sequences, so that there is no natural acoustic boundary between them, and sometimes no obvious change of phonetic quality during the vowel part of the word. An example is the word “tower”. In the most widely used pronunciation in southern England this word has two syllables, where the vowel of the first syllable is a diphthong and in the second syllable is a short neutral vowel; there is normally no consonant phoneme between them. Some people, however, round their lips so strongly between the two vowels that a /w/ consonant is perceived in the middle of the word. Other people go to the other extreme, and merge the vowels of the two syllables into a single long monophthong, which is not much different from a lengthened version of the vowel quality at the beginning of the usual diphthong. In a case such as this the word can only be regarded as having one syllable. But intermediate pronunciations are possible, so there are some pronunciations which are in the borderline region where it is impossible to be certain whether there is one syllable or two.

1.8 PHONEMES AND SPELLING

It is very important in the study of speech not to be confused by the conventional spelling of words, particularly for English where the relationship between spelling and pronunciation is so unpredictable. Although the vowel/consonant distinction in English orthography is not very different from that in phonetics and phonology, there are obvious anomalies. In the word “gypsy”, for example, both occurrences of the letter y function as vowels, whereas in “yet” the y is clearly a consonant. The letter x in “vex” represents a sequence of two consonants (it is transcribed phonemically as /veks/), but gh in “cough” represents a single phoneme, /f/.

There are many cases in English where the letter e after a consonant is not pronounced, but its presence modifies the phonemic identity of the vowel before the consonant (e.g. “dote” contrasts with “dot”). Combinations of vowel letters are often used to represent a single vowel phoneme (such as in “bean”) and in several varieties of English a letter r after a vowel is not pronounced as a consonant but causes the vowel letter to represent a different phoneme (for example, the change of “had” to “hard” and “cod” to “cord” only involves a change of vowel quality).

1.9 PROSODIC FEATURES

The phoneme identities are not the only carriers of linguistic information in speech. Pitch, intensity and timing are also important in human speech communication. In some languages, of which Chinese is the most obvious example, the pattern of pitch within a word is needed to supplement knowledge of the phonemes to determine the word’s identity. In Mandarin Chinese there are four different **tones** that can be used in each syllable, representing four different pitch patterns. In most European languages pitch, intensity and timing (collectively known as the **prosodic** features of the speech) do not normally affect the identities of the words, but they do provide useful additional information about what is being said.

Prosodic features can be used to indicate the mood of the speaker, and to emphasize certain words. Prosody is also the main factor responsible for determining which syllables are **stressed** in polysyllabic words. The most salient prosodic feature for indicating stress and word prominence is not, as one might expect, intensity but is in fact pitch—in particular the change of pitch on stressed syllables. Sound duration also increases for stressed syllables, but there are many other factors that affect durations of sounds, such as their positions in a sentence and the identities of the neighbouring sounds. Although prominent syllables do tend to be slightly more intense, and low-pitched sounds at the ends of phrases are often a few decibels weaker, intensity is less significant in assisting speech interpretation than are pitch and duration.

By focusing attention on the most important words, correct prosody is a great help in the interpretation of spoken English. Speech in which the prosody is appreciably different from that normally used by a native speaker can be extremely difficult to understand. Although the detailed structure of the pitch pattern may vary considerably between different local English accents (for example, between London and Liverpool), the general way in which prosody is used to mark stress is similar. The rhythmic structure is however completely different in certain other languages, such as French. In these **syllable-timed** languages the syllables seem to come in a much more uniform stream than in **stress-timed** languages such as English, where there tends to be a regular **beat** on the main stressed syllables. The implication is that in English the unstressed syllables between the syllables carrying the most prominence are shorter if there are more of them. Although this difference in type of rhythm between English and French is clearly perceived by listeners, there has been much controversy over its physical correlates. Attempts to find a systematic difference in the measured patterns of syllable durations between English and French in spontaneous conversation have not been very successful.

1.10 LANGUAGE, ACCENT AND DIALECT

Different languages often use quite different phonetic contrasts to make phonemic distinctions. This fact causes great difficulty for foreign language learners, particularly if their speech habits are already firmly established in their native language before another is encountered. It is beyond the scope of this book to give details of this effect, but a simple example will illustrate the point. In Japanese there is no phoneme corresponding to the English /l/, but there is one that is acoustically somewhat similar to the English /r/. When most Japanese hear an [l] in English it does not sound very close to any sound in their own language, but it is perceptually nearer to the sound associated with their /r/ than to any other. Speech is used for transmitting language, and there is a strong tendency to subconsciously replace one's memory of a speech sound by its linguistic label (i.e. phoneme) within a second or two of hearing it uttered. It is very common, therefore, for Japanese to be unable to distinguish, in both perception and production, between English words that differ only by an /l/-/r/ contrast (e.g. "light" and "right"). This comment is not to be interpreted as a criticism specifically of foreign speakers of English: native English speakers have similar difficulties, particularly in discerning vowel contrasts in languages with a very rich vowel system such as Swedish.

Different **accents** of the same language, although they may have just as much acoustic difference as different languages between representations of equivalent phonemes, do not normally cause much difficulty for native speakers. Because the underlying linguistic structure is almost identical, there are not many cases where the differences of phonetic quality between accents actually cause confusion in the intended word. For example, Scottish English does not distinguish between the vowels in “good” and “food”, but this does not cause confusion to southern English listeners because in this case the intended word (in their own accent) will be more similar to what they hear than any alternative. Even when there is a possible word confusion (such as in the identical southern English pronunciations of “flaw” and “floor”, which would be clearly distinct in Scottish), there is usually enough context available for only one of the word candidates to make sense.

The term **dialect** is often used to refer to clearly different varieties, spoken by a substantial group of people, of basically the same language. In addition to having appreciable variations of pronunciation, as in the examples above, dialects are often associated with the use of alternative words and sometimes with grammatical changes, which are not encountered outside the area where the dialect is spoken.

1.11 SUPPLEMENTING THE ACOUSTIC SIGNAL

It is apparent from the comments above that when humans listen to speech they do not hear an unambiguous sequence of sounds, which can be decoded one by one into phonemes and grouped into words. In many cases, even for an unambiguous sequence of phonemes, there is ambiguity about the sequence of words. (Consider the sentences: “It was a grey day.” and “It was a grade A.”) In fluent speech it will frequently be the case that the sound pattern associated with phonemes, particularly in unstressed positions, is not sufficiently distinct from the sound of alternative phonemes for the intended word to be clear. In normal conversation false starts to words, hesitation and mild stuttering are all extremely common. In the presence of background noise or a reverberant environment the speech signal might be further distorted so that distinctions that were clear at the speaker’s mouth are no longer so at the listener’s ear. Yet people can communicate by speech extremely easily.

In normal language there is so much redundancy in the linguistic message that only a small fraction of the information potentially available is necessary for the listener to deduce the speaker’s meaning (even if, in some cases, there will be uncertainty about some of the minor words). Relevant information includes what the listener knows about the speaker, and therefore what he/she is likely to talk about. If the conversation has already been in progress for some time, there will be a very strong influence from the previous context. Once the listener has become accustomed to the speaker’s voice, allowance will be made for his/her particular accent in resolving some phonemic ambiguities. But most of all, for each sentence or phrase, the listener will choose the one interpretation that seems to make most sense taking into account all available information, both acoustic and contextual. In some cases the final decision will actually involve rejecting some phonemes which accord well with the acoustic signal, in favour of others which would seem less likely based on the acoustic evidence alone. Except when the acoustic evidence is very much at variance with the norm for the chosen phoneme, the listener will not usually even be aware that the acoustic pattern was not quite right. By analogy, when people read printed text, minor typographical errors are often unnoticed, and the intended words are perceived as though they were really there.

Most people are familiar with the fact that in a crowded room, such as at a cocktail party, they can converse with the group of people in their immediate vicinity, even though there is a lot of competing speech at a high acoustic level from all the other people in the room. There is extra information in this case that is not available, for example, when listening through a telephone receiver. In the first case the availability of two ears enables some directional discrimination to be used. The human hearing system can infer direction by using the difference in intensity and time of arrival at the two ears of sounds that have otherwise similar structure.

The other important factor in face-to-face communication is the ability to see the speaker, and to correlate the acoustic signal with observed lip movements, and with other gestures which may be used to supplement the speech. Although it is usual to associate lip-reading with deaf people, even those with normal hearing generally develop a significant degree of subconscious ability to integrate visual with auditory information to assist in decoding speech. This lip-reading ability may not be sufficient on its own to resolve what has been said, but it is of great value in selecting between consonant sounds that can be confusable in background noise if relying on only the acoustic signal, yet have very distinct lip movements.

For the newcomer to this subject, the most surprising thing is perhaps that the listener is entirely unaware of this integration of visual with auditory information. The subjective impression to the listener is of actually “hearing” the acoustically ambiguous stimulus correctly, and the joke about partially deaf people putting on their glasses so that they can hear better is a reality. In fact this same phenomenon of integrating knowledge sources in one’s perception of the words ‘heard’ applies to all knowledge, including knowledge about the speaker, linguistic knowledge and knowledge of the real world’s influence on what people are likely to say.

1.12 THE COMPLEXITY OF SPEECH PROCESSING

It is clear that the human perceptual and cognitive systems must be enormously complex to be able to perform the task of linguistic processing. The very large number of neurons are, of course, working in parallel. Thus, although the actual processing speed in any one part of the central nervous system is very slow compared with the speed of modern electronic circuits, the overall perceptual decisions can be made within a few hundreds of milliseconds. Where machines are required to recognize and interpret speech, it is apparent that emulating human performance in processing normal relaxed conversation will not be possible without the machine having extensive linguistic knowledge and a very high ability to simulate human intelligence. However, if the task of the machine is simplified by placing constraints on what can be said, it is already possible to use speech for many types of human-machine interaction. Recent developments have greatly increased the range and complexity of tasks for which speech can be usefully applied, and speech technology capabilities are advancing all the time. Even so, the situation so often depicted in science fiction, where machines have no difficulty at all in understanding whatever people say to them, is still many years away.

CHAPTER 1 SUMMARY

- The use of speech offers great advantages for many types of human-machine communication, particularly by telephone. Understanding how humans use speech to communicate with each other highlights some of the issues.
- Speech is mainly used to communicate ideas, and the ideas are normally formulated in the brain in the form of language. Spoken and written language are quite different in their capabilities.
- Phonetics is the study of the production and properties of speech sounds, and phonology is the study of how they are used in language. The relationship between the acoustic properties of a speech signal and the linguistic units it represents is extremely complicated.
- The individual speech sounds (phones) are physical realizations of the smallest linguistic units (phonemes) in a speech signal. Allophones are different sounds that represent the same phoneme.
- In speech, vowels and consonants can be defined by their phonetic properties, but phonological functions should also be taken into account. Classification of vowels and consonants, and determination of the sequence of phonemes, is often only slightly related to conventional spelling, especially for English.
- Pitch, intensity and timing collectively make up the prosodic features of speech, which supplement the phonetic properties. Prosody is valuable for indicating important words, and adding emotional content to a message.
- Phonetic features are used differently in different languages, and people cannot generally detect phonetic differences not used in their native language.
- Human speech comprehension uses all available information to supplement phonetic properties of the speech. This information includes the direction of the sound source, lip movements and other gestures where these can be seen, and extensive knowledge of the language, context, and state of the world.
- The human speech perception and production processes are so complicated that their full capabilities will not be emulated by machines for many years. However, for more limited speech generation and recognition applications there are already useful systems, and capabilities are improving all the time.

CHAPTER 1 EXERCISES

E1.1 Give examples of circumstances where speech would not be the best medium for human-machine communication, and other situations where there is a great advantage in using speech, or even no practical alternative.

E1.2 What is the difference between phonetics and phonology?

E1.3 Explain, with examples, why it may be impossible to unambiguously divide a speech signal into separate words without knowing the word identities.

E1.4 Explain the relationship between phonemes, phones and allophones.

E1.5 What factors contribute to the distinction between vowels and consonants?

E1.6 Discuss the role of prosody in speech communication.

E1.7 Why is speech communication often possible even if the signal is distorted?

CHAPTER 2

Mechanisms and Models of Human Speech Production

2.1 INTRODUCTION

When developing speech synthesis and recognition systems for their many possible applications, the task is made much easier if one understands how humans generate speech, and how the various human processes can be modelled by electric circuits or in a computer. A speech generation model, in addition to aiding understanding of speech production, can itself form a useful basis for a speech synthesis system.

The main organs of the human body responsible for producing speech are the **lungs, larynx, pharynx, nose** and various parts of the **mouth,** which are illustrated by the cross-section shown in Figure 2.1. Muscular force to expel air from the lungs provides the source of energy. The air flow is modulated in various ways to produce acoustic power in the audio frequency range. The properties of the resultant sound are modified by the rest of the vocal organs to produce speech.

The process of acoustic **resonance** is of prime importance in determining the properties of speech sounds. The principal resonant structure, particularly for vowels, is known as the **vocal tract;** it starts at the larynx and extends up through the pharynx and mouth to the lips. For some sounds the nose is also coupled in to

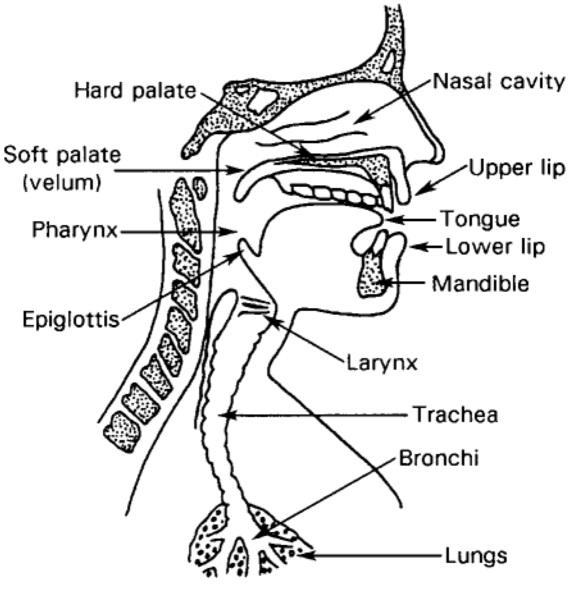


Figure 2.1 Diagrammatic cross-section of the human head, showing the vocal organs.

make a more complicated resonant system. The frequencies of the resonances and the way they move with time, and to a lesser extent their associated intensities, are crucial in determining what is being said. The main resonant modes of the vocal tract are known as **formants** and by convention they are numbered from the low-frequency end. For conciseness they are usually referred to as F_1, F_2, F_3, etc. In general F_1 and F_2 (usually in the range 250 Hz to 3 kHz) are the most significant in determining the phonetic properties of speech sounds, but some higher-frequency formants can also be important for certain phonemes. The resonant system can be viewed as a **filter** that shapes the spectrum of the sound **source** to produce speech.

2.2 SOUND SOURCES

The air stream from the lungs can produce three different types of sound source to excite the acoustic resonant system. These various sound sources are brought into operation according to what type of speech sound is being produced.

For voiced sounds, which normally include all vowels and some consonants, such as [m, n, l, w], the air flow from the lungs and up the trachea is modulated by vibrations of the **vocal folds,** located in the larynx. The vocal folds (sometimes also known as the **vocal cords**) are illustrated in Figure 2.2. They are two folds of tissue stretched across the opening in the larynx. The front ends of the folds are joined to the thyroid cartilage, and the rear ends to the **arytenoid** cartilages. The arytenoids can, under muscular control, move far apart so that there is a wide triangular opening between the vocal folds. This is the normal condition for breathing. They can also bring the folds tightly together, completely closing the top of the trachea. This condition is achieved when one holds one's breath, and it occurs automatically during swallowing, to prevent food or drink from entering the lungs. The arytenoids can also be held so that the vocal folds are almost touching. If air is forced through the slit-like opening between them (known as the **glottis**), the folds will start to vibrate, and so modulate the air flow. The result is a build-up of vocalfold oscillation whose frequency is mainly determined by the mass and tension of the folds, but is also affected by the air pressure from the lungs. The modulation of the air stream by the vibrating vocal folds is known as **phonation**. When the

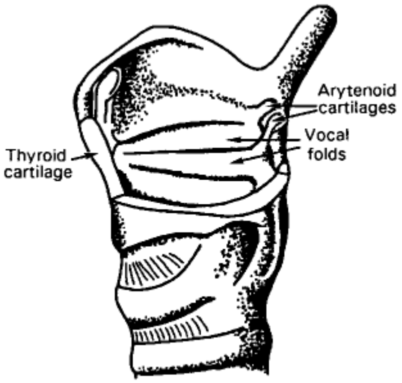


Figure 2.2 Cut-away view of the human larynx.

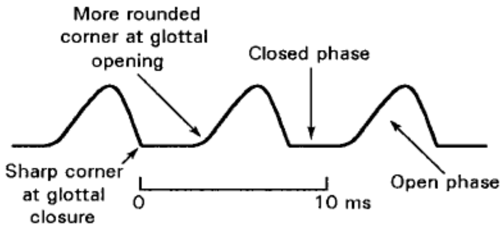

More rounded corner at glottal opening
Closed phase
Sharp corner at glottal closure
0
10 ms
Open phase

Figure 2.3 Typical air-flow waveform through the glottis during phonation.

vibration amplitude has built up sufficiently, which usually happens after one or two cycles, the extent of the movement is such that the vocal folds make contact in the closing phase, thus completely and abruptly stopping the air flow.

The variation of volume flow through the glottis is typically as shown in Figure 2.3. The presence of a sharp corner at the point of closure gives rise to a power spectrum of the airflow waveform with frequency components of significant (though small) magnitude up to several kHz. It is thus the shock to the resonant system resulting from the sudden blocking of the air flow through the glottis that causes the main excitation of the formants in every phonation cycle. The **fundamental frequency** of this signal lies typically in the range 50–200 Hz for adult male speakers, and about one octave higher for adult females. The subjective impression of voice **pitch** is very closely related to the fundamental frequency, and is only slightly affected by the formant frequencies. Although the spectrum of a single glottal pulse has a continuous distribution in frequency, periodic repetition of the pulses causes the total voiced excitation to approximate to a line spectrum.

Besides the gradual build-up of phonation described above, it is also possible for phonation to start with the vocal folds held just in contact. In this case the build-up of pressure starts the process by forcing the folds apart to allow the first glottal pulse through, but within two or three cycles the vibration will settle into a periodic pattern, similar to that which occurs when the folds are slightly apart at the start of phonation. With a closed-glottis start, the formants will even be excited by the closure in the first cycle of vibration.

The cessation of phonation can also have two distinct patterns, depending on whether the folds are relaxed and pulled apart, or are forced tightly together. In the former case the vibration dies out gradually, with the folds not touching in the last few cycles. In the latter, the pulses cease very quickly but the glottal closure remains sharp, even in the last pulse. In addition, the last two or three pulses before cessation are usually further apart in time. Firmly closing the glottis to stop phonation for a few tens of milliseconds, and then allowing phonation to re-start suddenly by relaxing the closing force, produces the so-called **glottal stops** that are a feature of many people's speech in, for example, London and Glasgow.

The overall complexity of the vocal fold vibration differs for different people, and the shape of the flow waveform varies with vocal effort and other aspects of voice quality. For example, sometimes the parting of the vocal folds is sufficiently fast for there to be significant power in the higher audio frequencies at that part of the cycle also. In addition to the actual air flow through the glottis, there are other small components of the effective volume velocity into the bottom of the pharynx that arise from surface movements of the vocal folds. During phonation the whole vocal fold structure moves up and down as well as laterally. On glottal closure there is a rippling motion of the upper vocal fold surface which causes additional air displacement just above the larynx during the closed period. The volume displacement caused by this effect is small compared with the total volume of a glottal pulse, and its influence on the low-frequency power (i.e. the lowest two or three harmonics) is negligible. However, these surface movements are fairly rapid (involving times of the order of 1 ms). At higher frequencies, where the energy associated with the sharpness of glottal closure is only a very small fraction of the total pulse energy, this additional source of volume flow can significantly modify the spectral components of glottal excitation. In some situations it might contribute to the characteristic voice qualities of different speakers, although any effect will be small compared with other speaker-specific factors affecting voice quality.

The second major source of sound in speech production is the air turbulence that is caused when air from the lungs is forced through a constriction in the vocal tract. Such constrictions can be formed in the region of the larynx, as in the case of [h] sounds, and at many other places in the tract, such as between various parts of the tongue and the roof of the mouth, between the teeth and lips, or between the lips. The air turbulence source has a broad continuous spectrum, and the spectrum of the radiated sound is affected by the acoustics of the vocal tract, as in the case of voiced sounds. Sustainable consonant sounds that are excited primarily by air turbulence, such as [s, f], are known as **fricatives,** and hence the turbulence noise is often referred to as **frication**.

The third type of sound source results from the build-up of pressure that occurs when the vocal tract is closed at some point for a stop consonant. The subsequent plosive release of this pressure produces a transient excitation of the vocal tract which causes a sudden onset of sound. If the vocal folds are not vibrating during the closure, the onset is preceded by silence. If the vocal folds are vibrating during the pressure build-up, the plosive release is preceded by low-level sound; the power of this sound is mostly at the fundamental frequency of phonation, and is radiated through the walls of the vocal tract. The plosive release approximates a step function of pressure, with a consequent -6 dB/octave spectrum shape, but its effect is very brief and the resultant excitation merges with the turbulent noise at the point of constriction, which normally follows the release.

In connected speech muscular control is used to bring all of these sound sources into play with just the right timing for them to combine, in association with the appropriate dimensions of the resonant system, to produce the complex sequence of sounds that we recognize as a linguistic message. For many sounds (such as [v, z]) voiced excitation from the vocal folds occurs simultaneously with turbulent excitation. It is also possible to have turbulence generated in the larynx during vowels to achieve a **breathy** voice quality. This quality is produced by not closing the arytenoids quite so much as in normal phonation, and by generating the vibration with a greater air flow from the lungs. There will then be sufficient random noise from air turbulence in the glottis combined with the periodic modulation of the air flow to produce a characteristic breathiness that is common for some speakers. If this effect is taken to extremes, a slightly larger glottal opening, tense vocal folds and more flow will not produce any phonation, but there will then be enough turbulence at the larynx to produce whispered speech.

2.3 THE RESONANT SYSTEM

In the discussion that follows, the concepts of acoustic resonance, coupling, damping, impedance, etc. are widely used. For electrical engineers these concepts in acoustics are not normally familiar, but they are, in fact, very closely analogous to their electrical counterparts, and it can therefore be helpful to think of them in electrical terms. In acoustic systems it is normal to regard sound pressure as analogous to voltage, and volume flow as analogous to current. Energy loss as a result of viscosity is then represented by series electrical resistance, and heat conduction losses can be associated with shunt conductance. The inertance of a mass of air corresponds to inductance, and compliance of the air to capacitance.

Using these concepts the theory of sound transmission in the vocal tract is very similar to electrical transmission line theory, and the major structural discontinuity at the larynx end can be modelled fairly well by appropriate lumped values of resistive and reactive components. There are, however, many idealizing assumptions necessary about sound propagation if the equivalent electrical circuits are to be simple enough for easy analysis.

If the **soft palate** (or **velum**) is raised and held in contact with the rear wall of the pharynx there will be no opening between the pharynx and nose; the properties of the vocal tract between larynx and lips can then be modelled fairly closely by an unbranched air-filled tube with several cylindrical sections butted together. Assuming the cross dimensions of this tube are such that there is only plane wave propagation along its length at audio frequencies, and assuming sound propagation within the tube is totally without loss, it is not too difficult to calculate the response of such a tube, i.e. the mathematical transfer function relating volume velocity inserted at the larynx end to that radiated from the lips. The mathematics becomes more practical if, instead of radiating into free space, the lip opening is represented as coupling into an infinite length tube of cross-section that is large compared with the opening, as illustrated in Figure 2.4. A full analysis of this situation is beyond the scope of this book, but is given by Rabiner and Schafer (1978), pp. 92–98.

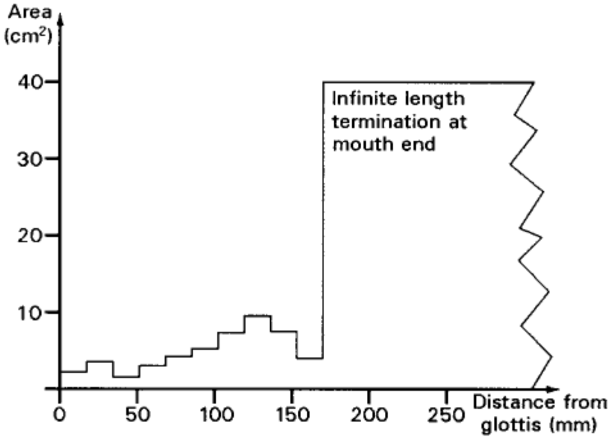


Figure 2.4 Graph of cross-section of a 10-section acoustic tube modelling a typical vowel. The mouth termination is shown coupling into an infinite tube of cross-section 40 cm^2.

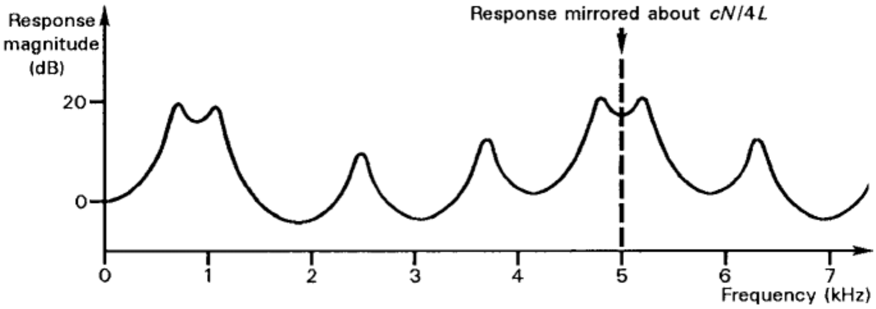


Figure 2.5 Typical response of a 10-section acoustic tube, such as is illustrated in Figure 2.4.

The results of this analysis show that the transfer function is periodic in frequency, with a repetition every *cN/*2*L,* where *c* is the velocity of sound, N is the number of elementary tubes and *L* is the total length of the model tract. As any transfer function for a real system must have a frequency response that is symmetrical about zero frequency, the periodicity implies there is also symmetry about odd multiples of *cN/*4*L.* A typical response is shown in Figure 2.5. The frequency-domain periodicity is exactly the same as that which occurs in sampled-data filters, and is evident in the *s*-plane to *z*-plane transformation used in sampled-data filter theory. The relevance of sampled-data filter theory is a consequence of the fact that a wave travelling in an abutted set of uniform tubes only has any disturbance to its propagation when it meets a change in diameter, to cause partial reflection. As these changes occur at regular distances, and therefore at regular time intervals, the response must be representable by a sampled-data system, which has a sampling rate of *cN/*2*L* (i.e. the sampling interval is equal to twice the wave propagation time through one tube section). The factor of 2 arises because the minimum time before any partial reflection can again influence the conditions at any tube junction is equal to the time taken for the reflected component to return to the previous junction, and then be reflected back to the point under consideration.

For a sound velocity of 350 m/s and a tract length of 0.175 m (typical for an adult male speaker) a total of 10 elementary tubes will specify a total of five resonances within the range 0–5 kHz, which will be the five lowest formants of the system. There will, however, be mirrored and repeated resonances at higher frequencies, which will be very unlikely to fit the real speech spectrum above 5 kHz. A greater number of tube sections would enable the resonant modes to be independently specified up to a higher frequency, and could be represented by a sampled-data filter with a higher sampling rate.

The transfer function of the acoustic tube model has an infinite number of poles, but no zeros. The magnitude of the transfer function is directly related to the frequencies of the poles, and when the dimensions of the tube are such that two resonant modes move close in frequency, the intensities associated with these resonances will increase. If the tube is uniform the resonances will be equally spaced, at *c/*4*L,* 3*c/*4*L,* 5*c/*4*L,* etc., and the magnitude of the transfer function will be the same at each resonant frequency.

The above calculations involve several idealizing assumptions, many of which do not fit the facts of human speech production very closely. However, this acoustic tube model will predict the frequencies of the three or four lowest formants in the vocal tract fairly well if the cross-sectional area is known as a function of distance along the tract. The model will not, however, describe the high frequencies well, for a number of reasons. First, the assumption of plane waves is only valid when the cross dimensions of the tube are small compared with a half-wavelength of sound. This assumption will be seriously in error at some places in the tract from about 3 kHz upwards. Second, the complexities of shape around, for example, the epiglottis, the sides of the tongue, the teeth, etc., are totally unlike the abutted cylindrical tubes of the model. Third, there are significant losses in the vocal tract from many causes. These losses will give rise to an increase in damping of the resonances, and in particular the higher frequency resonances will become very heavily damped because the reflection from the mouth opening will not be so effective for wavelengths comparable with the mouth dimensions.

For **nasal consonants,** e.g. [m, n], the velum is lowered to produce appreciable coupling between the top of the pharynx and the nose. In addition, the mouth is blocked at some point determined by the identity of the consonant. There will thus be a branched acoustic system, illustrated in Figure 2.6, with a closed side branch (the mouth). Apart from the effect of vocal tract wall vibrations, all the sound will come out of the nose. Mathematical analysis then becomes much more difficult, partly because of the unknown and variable coupling at the velar opening, but even more due to the very complicated structure of the nasal cavities. Apart from the division into two parts by the nasal septum, the inside of the nasal cavities has elaborately shaped bony structures, some acoustic coupling into the sinus cavities, and a considerable quantity of hair around the nostrils. The effect of all of these features is to increase the damping of the resonances, to increase the total number of resonant modes in a given frequency range, and to cause spectral zeros as well as poles in the transfer function as a result of the side branch.

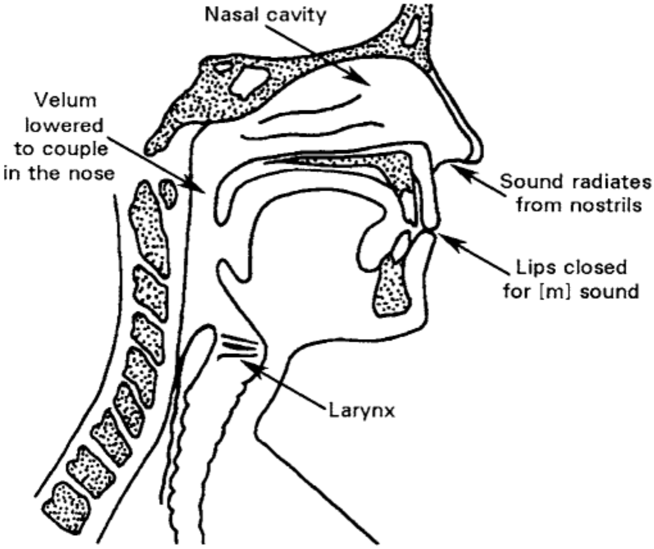

Nasal cavity
Velum lowered to couple in the nose
Sound radiates from nostrils
Lips closed for [m] sound
Larynx

Figure 2.6 Acoustic system for producing a typical nasal consonant, [m].

It is also possible to have the velum lowered during vowel sounds, to produce **nasalized vowels**. In languages such as French the nasalized vowels are distinct phonemes, and the change to the properties of the acoustic signal as a result of nasalization is very noticeable. In other languages, such as English, nasality in vowels has no linguistic significance. However, because specific muscular effort is required to keep the velum raised to decouple the nose, it is common for there to be a considerable degree of nasal coupling during English vowels, particularly when those vowels are adjacent to nasal consonants. The most prominent acoustic effects are to cause an additional resonance near to the first formant, and to cause additional first formant damping. Nasalization in vowels is not so common near to fricative or plosive consonants, because the velum has to be raised to allow the pressure build-up needed during consonant production.

The resonant frequencies (and therefore the poles of the transfer function) of the vocal tract for a given configuration are independent of the position of the sound source. However, there are three differences which make the spectral envelope of the radiated sound very different for voiced excitation, compared with plosive or fricative excitation. First, the spectrum of the voiced source has nearly all of its power in the lowest few harmonics, and the slope of the spectral intensity above about 1 kHz is at least -12 dB/ octave. At low vocal effort the fall-off is often a lot faster. Second, the vocal tract is much less constricted for most of the time during voiced sounds, except at the glottis, where it is closed or almost closed. During voiceless sounds the glottis is normally fairly wide open, such that the acoustic system behind the source of frication includes the sub-glottal system. Because of the coupling with the bronchi and the lungs, this system is quite heavily damped. The third effect is that (except when the constriction is in the laryngeal region) the sound source is further forward in the vocal tract. Figure 2.7 illustrates a typical vocal tract configuration for producing the sound [s].

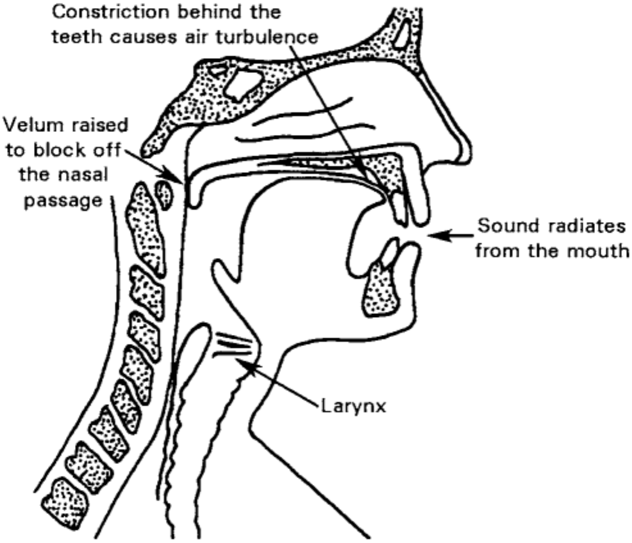

Constriction behind the teeth causes air turbulence
Velum raised to block off the nasal passage
Sound radiates from the mouth
Larynx

Figure 2.7 Articulator positions for producing the fricative consonant, [s].

There are two alternative ways of analysing the effect of the different position of the sound source. One way is to consider the whole acoustic system, which has transfer function poles that are independent of placement of the source. That part of the vocal tract behind the source acts as though it were a closed tube in series with the source to the total system, and its transfer function poles will cause absorption of power from the source at the pole frequencies, and thus add zeros at these frequencies to the transfer function from source to lips. However, as there is normally a close constriction at any point of frication or stop release, the acoustic systems on each side of the constriction are almost independent (i.e. there is very little coupling between them). In this circumstance the poles of the overall system are almost the same as the poles of the two part-systems considered in isolation; the poles associated with the part of the tract behind the constriction will thus be almost coincident with the zeros, and will therefore have their effects substantially cancelled. It is then possible to regard the tract length as consisting only of the part from the constriction to the lips, and to ignore the part behind the constriction. This shorter vocal tract will, of course, have more widely spaced resonances, and for sounds where the constriction is very near the mouth opening (e.g. [s, f]) the resonant length will be so short and the ratio of mouth opening to cavity volume will be so large that there will be only one or two obvious resonant modes, tuned to high frequencies (above 3.5 kHz) and very broad because of their heavy damping.

2.4 INTERACTION OF LARYNGEAL AND VOCAL TRACT FUNCTIONS

During voiceless sounds, for which the glottis is normally wide open, there is strong acoustic coupling between the vocal tract and the sub-glottal system. However, as explained above, the constriction needed to produce excitation for these sounds substantially decouples the region behind the constriction, and so the sub-glottal system has very little effect on the acoustics of the radiated sound. In the case of sounds excited primarily by the glottal air flow, the time-varying impedance presented by the glottis to the lower end of the pharynx will affect the overall sound properties. Of course, the transfer function relating volume flow at the glottis to sound radiated from the lips does not depend on glottal opening, but the finite acoustic impedance at the glottis when it is open will mean that the volume flow through the glottis will depend on the frequency-dependent load impedance presented by the vocal tract. In consequence there can be prominent ripple components of the open-phase glottal flow at the formant frequencies, particularly for F_1. It is easier to appreciate the effects of the varying glottal impedance, not by taking into account this modification to the effective source waveform, but by estimating the effect of the glottal impedance on the poles of the whole acoustic system, which includes the glottis. An electrical equivalent circuit of the acoustic system is shown in Figure 2.8.

The impedance looking down the trachea below the glottis will be fairly low, because of the large cross-sectional area and the heavy damping on any resonances caused by the structure of the lungs. Even at its maximum opening in the phonatory cycle the glottis area is very much smaller than that of the trachea, so the acoustic impedance presented to the bottom of the pharynx is substantially that of the glottis itself. This impedance will be a combination of resistance, and reactance

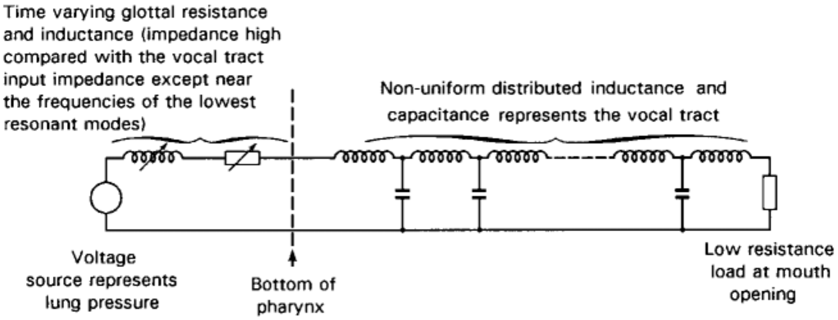

Time varying glottal resistance and inductance (impedance high compared with the vocal tract input impedance except near the frequencies of the lowest resonant modes)
Non-uniform distributed inductance and capacitance represents the vocal tract
Voltage source represents lung pressure
Bottom of pharynx
Low resistance load at mouth opening

Figure 2.8 Electrical equivalent circuit of the glottis coupled to an idealized vocal tract, with all losses at the terminations.

resulting from the mass of the air in the glottal opening. The effect of the open glottis impedance for typical vowels is to cause a small increase of F_1 frequency, but a very noticeable increase of F_1 damping. The higher formants, which are more heavily damped anyway because of other losses, show much smaller effects.

When looking at the speech waveform for the period from one main glottal excitation point to the next, it is not at all easy to know whether to attribute observed departures from a simple decaying resonant system response to variations in glottal impedance, or to changes in volume flow stemming directly from phonation. For example, if the vocal folds part sharply enough at the beginning of their open phase, the start of the glottal flow can cause significant secondary excitation of F_1, which may be in such a phase relationship to the decaying F_1 response from the previous glottal closure that it causes partial cancellation, producing a sudden reduction in amplitude. It would be difficult in such a case to distinguish the effect from that caused by a sudden increase of formant damping. The same phenomenon occurring with slightly different phonation or formant frequencies might cause an amplitude increase which would be clearly seen as secondary excitation, and would normally be followed by an obvious increase in rate of amplitude decay because of the extra damping. The extra formant damping in the open glottis period is most noticeable for open vowels, such as the [a] sound in the first syllable of "father". These vowels are associated with a smaller pharynx cross-section, which is more closely matched to the glottal area, so substantially reducing wave reflections at the glottis. When the glottis is closed, the damping of F_1 is such that the 3 dB bandwidth of the formant is about 50 Hz, but for the open glottis it can be at least four times greater. Typical closed-glottis bandwidths for F_2 are around 80 Hz, and for the higher formants can be 150 Hz or more.

In addition to glottal opening having an effect on F_1, F_1 can also have some effect on the glottis. If a low-order harmonic of the fundamental frequency of phonation is near the frequency of F_1, the F_1 flow through the glottis causes a slight tendency for the relaxation oscillation of the vocal folds to be 'pulled' by the formant frequency, thus making the harmonic move with the formant.

Another interaction between the fundamental frequency and the formants occurs because the muscular force needed to raise the pitch also raises the larynx. (This movement might be as much as 20 mm.) The raising of the larynx shortens the pharynx, and thus tends to increase the formant frequencies. This modification to the formant frequencies is one reason why it is possible to perceive pitch changes even when speech is whispered.

2.5 RADIATION

So far the discussion has been on the properties of the sound sources, and the effect of the resonant system on the properties of the volume velocity at the lips and nostrils. The volume flow leaving these openings causes a pressure wave to be radiated, which can be heard by a listener or cause a response from a microphone. The waveform shape of the radiated pressure wave from a small opening in a large baffle can be found by taking the time-derivative of the volume flow from the radiating orifice. The spectrum of the radiated sound therefore differs from that of the volume velocity by a 6 dB/octave lift. When the mouth and nose are both radiating at the same time (in nasalized vowels) the two pressure waves will combine. At all frequencies where the audio power is significant the wavelength will be so great compared with the spacing between mouth and nostrils that simple addition of the volume velocities from the two sources will suffice. Diffraction round the head will reduce the level in front of the head by up to 3 dB for wavelengths that are large compared with the head dimensions, due to a significant fraction of the power at these frequencies being radiated behind the speaker. However, this level change is fairly small compared with the effect of the wide variety of different acoustic environments that the human listener is easily able to allow for. For example, the low-frequency spectrum drop can be compensated to a large extent by having the speaker stand with his/her back to a wall, and a low-frequency boost would arise when a speaker stands facing outwards from a corner.

2.6 WAVEFORMS AND SPECTROGRAMS

An annotated typical speech waveform, representing the sentence “The new bricks fell over” spoken by an adult male with a southern British English accent, is shown in Figure 2.9. The variety of structure associated with the various speech sounds is very obvious, and some information about the phonetic content can be derived from waveform plots. However, the waveform is not useful for illustrating the properties of speech that are most important to the general sound quality or to perception of phonetic detail.

In view of the crucial significance in speech communication of resonances and their time variations, it is very important to have some means of displaying these features. The short-time spectrum of the signal, equivalent to the magnitude of a Fourier transform of the waveform after it has been multiplied by a timewindow function of appropriate duration, cannot, of course, show any information that is not in the original signal. It will, however, be more suitable for displaying the resonances. Because the time variations of the resonances are responsible for carrying the phonetic information that results from moving the articulators, it is important to have a means of displaying a succession of Fourier transforms at short time intervals (at most a few milliseconds apart).

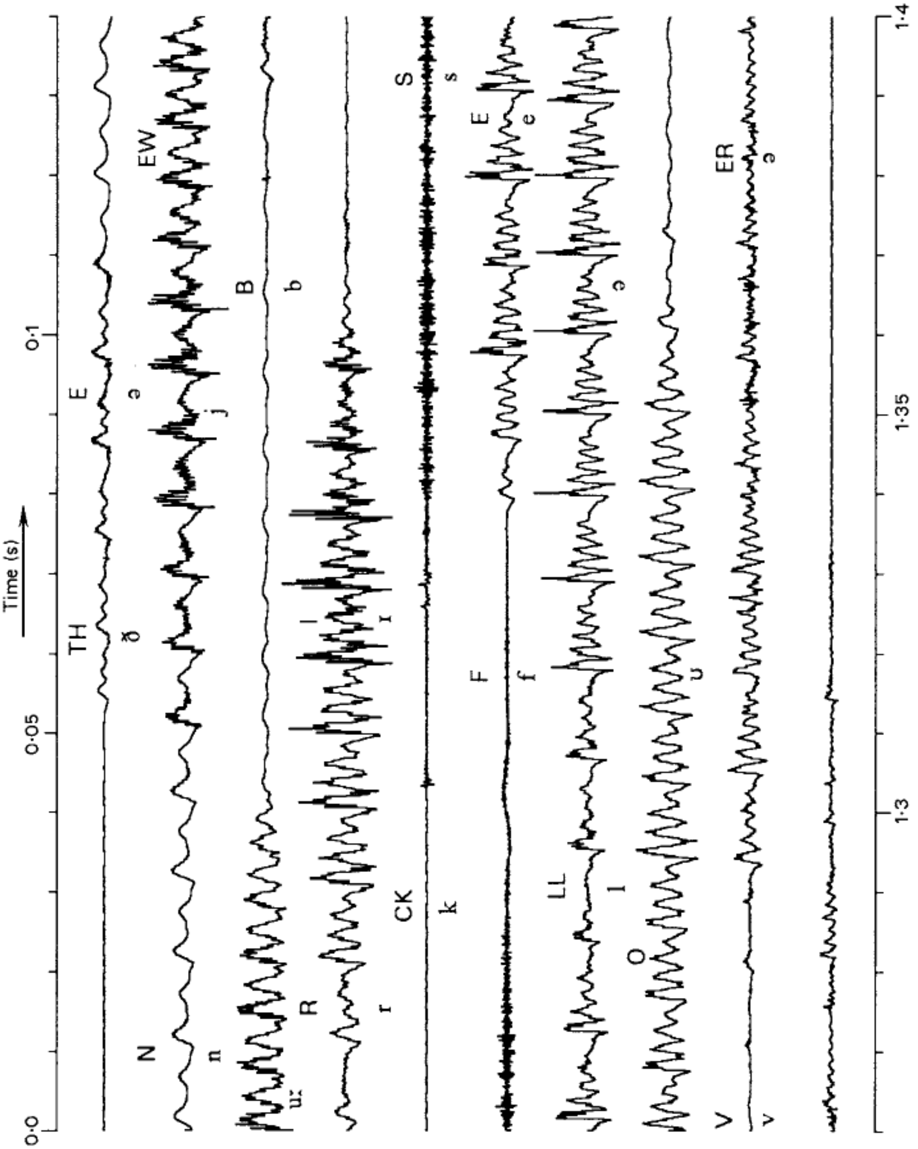


Figure 2.9 Waveform of the sentence “The new bricks fell over” spoken by an adult male talker with a southern British English accent. Although the sentence is less than 1.4 s long, the timescale is too compressed to show the detail. The text is marked above the graph in conventional orthography and below in phonemic notation, both in approximate time alignment with the waveform.

There are many ways in which a succession of Fourier transforms can be displayed. Using current computer technology, the speech waveform for analysis could be input to the computer using an analogue-to-digital converter, and the required Fourier transforms could be calculated and plotted, each one just below the next, on a screen or on paper. This method of spectral analysis can be useful for some purposes, but it is not easy to interpret formant movements from such a succession of spectral cross-sections, partly because when they are plotted far enough apart to be distinguishable, the total amount of display area needed for even a fairly short sequence of phonetic events is excessive. It has been found much easier for general study of the acoustic properties of speech signals to use the horizontal dimension for time, the vertical dimension for frequency, and to represent the short-time spectral intensity at each frequency by visual intensity, or colour, or some combination of the two. In this manner it is possible to get a very compact display of a few seconds of speech, in a way that allows the phonetically important acoustic features to be easily interpreted. It is obviously not possible to judge relative intensities of different parts of the signal so precisely from such a variable-intensity display as it would be from a spectral cross-section plot, but in practice a variable-intensity plot is usually adequate for most purposes. If a combined colour and intensity scale is available, it is, of course, possible to get finer spectral level discrimination.

So far this discussion has assumed that a computer and Fourier transforms will be used for generating **spectrograms**. This requires a lot of computation, but these days computational power is not expensive, and the method is widely used in research laboratories with normal computing facilities. The earliest spectrograms (in the 1940s) were, however, made by purpose-built **spectrographs** to obtain equivalent pictures by a completely different technique. The magnitude of the short-time Fourier transform at a particular frequency and time is equivalent to the signal amplitude at the appropriate time from a suitable band-pass filter centred on the required frequency. The width and shape of the filter pass-band have to be chosen so that the envelope of its impulse response is of similar shape to the time window used on the input to the Fourier transform, although a very close match is not normally possible using a hardware filter.

The original spectrographs stored a short passage of speech on a magnetic drum so that it could be played back repetitively. On the same shaft as the magnetic drum was another drum, on which was wrapped a sheet of electrosensitive paper. Held in contact with the paper was a stylus mounted on a slide driven by a screw geared to the drum. When a sufficient voltage was applied to the stylus, the resultant sparking caused the paper to burn, so turning it black. For each revolution of the drum the stylus moved along by about 0.25 mm, so that it eventually covered the whole surface of the paper. The movement of the stylus was also used to vary the frequency of a heterodyne oscillator, which effectively varied the tuning of the band-pass analysis filter to select successive frequencies for analysis. The signal from the magnetic drum was fed into the filter, and after a sufficient number of drum revolutions a complete picture was built up, in the form of closely spaced horizontal lines of varying blackness. Although some spectrographs working on this principle may still be in use, they have now almost completely been superseded by the Fourier transform method.

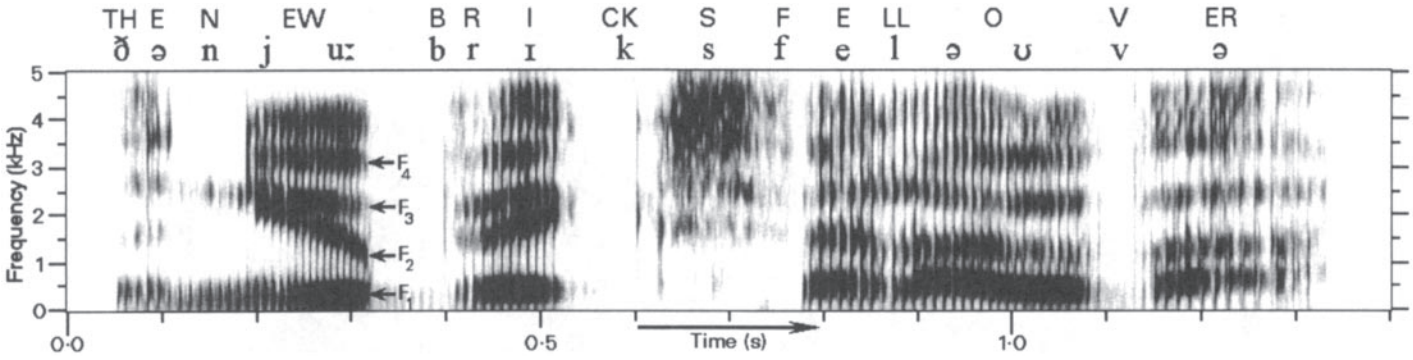


Figure 2.10 Wide-band spectrogram of the speech waveform shown in Figure 2.9. The dynamic range of the grey scale in the display is 50 dB, so very weak sounds are clearly visible.

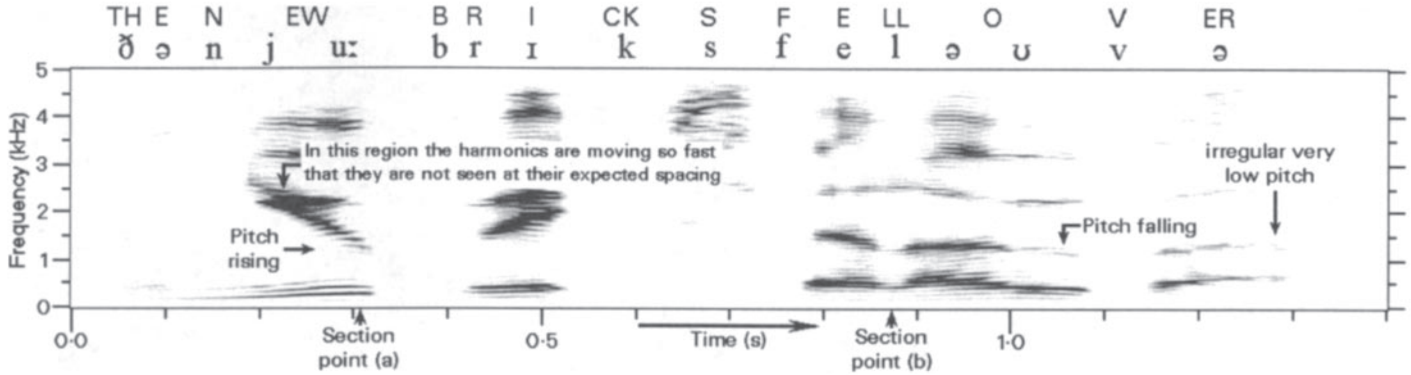


Figure 2.11 Narrow-band (30 Hz) spectrogram of the speech waveform shown in Figure 2.9. The dynamic range of marking in this picture is only 30 dB, so weak sounds are not visible, but the harmonic structure of vowels is clearly seen.

One very important parameter in short-time Fourier analysis is the width (and also the shape) of the time window. A long window corresponds to a narrow bandpass filter, and if the bandwidth is appreciably less than the fundamental frequency of phonation the analysis will separate the individual harmonics of the voiced excitation source. If the time window is short it will only contain at most the response to one excitation pulse, which cannot display the harmonic structure. In effect, the bandwidth of the equivalent filter is wider than the fundamental frequency and so the harmonics will not be separated. With a wide filter, because its impulse response is shorter, the instrument will display the fine time-structure of the signal in more detail than with a narrow filter. Figures 2.10 and 2.11 show wide-band and narrow-band spectrograms of the same short sentence from a typical adult male talker. From the wide-band spectrogram it is easy to see the formant movements, and the responses to the individual glottal excitation pulses can be seen in the time pattern of the display for each formant. On the other hand, the narrow-band picture shows the harmonic structure clearly, but blurs the rapid changes. Although the formant movements are still embodied in the variations of harmonic intensities, they are much more difficult to discern because of the distracting effect of the independent movements of the harmonics. The useful range of filter bandwidths for speech analysis lies between about 25 and 400 Hz. Narrow-band spectrum cross-sections of the marked points in Figure 2.11 are shown in Figure 2.12.

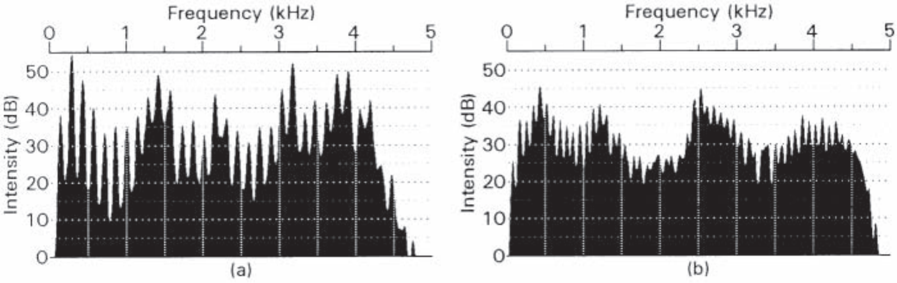


Figure 2.12 Narrow-band spectral cross-sections, taken at the points marked on Figure 2.11.
(a) Section through the end of the vowel in the word “new”.
(b) Section through the final consonant of “fell”.

2.7 SPEECH PRODUCTION MODELS

If the various functions of human speech production can be modelled, either acoustically, electronically, or in a computer program, it is possible to produce speech synthetically. Although early such attempts used acoustic models (see Linggard (1985) for a comprehensive review), electronic models took over in the 1930s, and computer models are more widely used today. In all these models the utterances to be spoken must be provided in the form of control signals that, in effect, represent the muscular control of the human vocal system. Because of the inertia of the articulators, such control signals do not change extremely fast, and each control signal can be represented well enough for almost all practical purposes within a bandwidth of 50 Hz, or by sample values every 10 ms.

In order to use a speech production model for speech synthesis, it is necessary to model both the sound sources and the resonant structure of the vocal tract. The methods to be used for these two operations are not independent. If it is required to model the human speech production process very closely it should ideally be necessary to model the detailed mechanics of vocal fold vibration, and also to have models of turbulent and plosive excitation that can be inserted in the appropriate places in an articulatory model of the complete vocal tract. In principle such models are possible, but there are enormous practical difficulties in making them adequately represent the complete process of realistic speech production.

2.7.1 Excitation models

Ishizaka and Flanagan (1972) modelled vocal fold vibration on a computer by considering each fold as two adjacent masses, representing the upper and lower parts, coupled by springs. They found that this model well represented the gross behaviour of real vocal folds, in that it would start and stop phonation in generally plausible ways as the sub-glottal pressure and vocal fold spacing were given appropriate values. However this model cannot represent the more subtle aspects of the motion of real vocal folds, such as are seen on high-speed motion pictures of the larynx. For more complicated models the computation time is greatly increased and it becomes much more difficult to deduce realistic values for the parameters of a more elaborate structure. For these reasons it is not usual to use models of the actual vocal folds in practical speech synthesis, but instead to adopt other means for generating a functional approximation to the voiced excitation signal.

Bearing in mind that the most important property of the glottal flow is its excitation of the formants at every glottal closure, it is possible to represent the flow merely by a train of impulses repeating at the fundamental frequency. As a real glottal pulse has most of its energy at low frequencies, an impulse train, with its flat spectral envelope, will not produce the correct relative intensities of the formants. However, the spectral balance can be approximately corrected by a simple linear filter. As the ear is not very sensitive to moderate amounts of phase distortion, even using a minimum-phase filter can give an excitation source that approximates fairly well to the important features of glottal excitation. A much better approximation, that is still far easier to generate than a vocal fold model, is achievable by representing the air-flow pulse shape by some simple mathematical function (e.g. by a small number of segments from a cosine wave). By varying the numerical parameters specifying each segment it is then even possible to model the variations in shape with vocal effort, and some of the differences between different speakers. Closer approximations to more complex pulse shapes can be produced by storing one or more typical shapes as sets of waveform samples, and repeating those sample sequences at the required fundamental frequency. To get good results by this technique it is necessary to have some method of varying the timescale of the pulses, particularly as the fundamental frequency varies.

As most speech synthesizers these days are implemented digitally, using sampled-data filters, it is naturally attractive for implementation to make voiced excitation pulses have a spacing equal to an integer number of sampling periods. At typical sampling rates of around 10 kHz this quantization of pulse positions can cause noticeable roughness in the perceived pitch of synthetic speech, particularly for high pitches, and so it is also desirable to have some means of interpolating excitation points between the signal samples. If these precautions are taken, the stored-shape models of voiced excitation can be used to generate synthetic speech almost totally indistinguishable from high-quality recorded natural speech.

Turbulent excitation can be very well modelled by random electrical noise or, in digitally implemented synthesizers, by a pseudo-random sequence generator. On occasions when fricative and voiced excitation are produced simultaneously, the fricative intensity in natural speech will be modulated by the periodically varying air flow through the glottis. It is not difficult to include this effect in a synthesizer, particularly in digital implementations, although there seems to be little evidence so far that it influences perception of even the highest quality synthetic speech.

Plosive excitation can be well represented by a single impulse, in conjunction with an appropriate filter to achieve the desired spectrum shape. However, as the spectral fine structure of a single pulse is the same as that of random noise, it is quite usual to use the same noise generator for plosive as for fricative excitation. The difference will be that the phase coherence of impulsive excitation will not then be achieved on plosive bursts. As the duration of such bursts is normally only a few milliseconds, the difference can only be perceived under very favourable listening conditions.

2.7.2 Vocal tract models

As with the vocal folds, a computer model can be made to represent the physical structure of the remainder of the vocal system. The limitations of a simple acoustic tube model have already been discussed. Some improvement can be obtained by modelling the losses distributed along the tube, and Flanagan *et al*. (1975) achieved this by representing the vocal tract by a simulation of a 20-section lumped constant transmission line with realistic losses in each section. With their model, comprising the vocal folds and the simulated pharyngeal, oral and nasal tracts, they have produced complete utterances by controlling dimensions and other parameters of the model to copy the articulatory behaviour of real speakers. However, the problems of determining and controlling the details of the model parameters are extreme, and their model does not attempt the very difficult task of representing the departures from plane wave propagation. For these reasons, although continuing developments in articulatory synthesis are providing valuable insights into the mechanisms of speech production, it is likely to be very many years before it will be practicable to use these techniques for routine generation of synthetic speech.

There is, however, a particular use of the acoustic tube model for speech synthesis which is both practical and very widely used. An acoustic tube with N sections has N degrees of freedom in its specification (usually represented by the N reflection coefficients at the section boundaries). These N degrees of freedom are responsible for determining the N co-ordinates of the independent poles in the transfer function of the tube. (Each pair of poles can be complex conjugate or real, in both cases requiring two co-ordinates to specify two poles.) When fed with a suitable excitation signal, the dimensions of the tube can be chosen so that those poles give the best possible approximation to spectral properties of a short segment of speech signal, according to some suitable error criterion. If N is high enough, but the sampling rate of the synthesizer is kept near the Nyquist rate for the bandwidth of signal required (e.g. around 10 kHz) it is possible to make an all-pole minimum-phase filter that will give a very close approximation to any actual spectral shape of a speech signal. If N is made sufficient to represent the number of formants within the given bandwidth, and if the speech sound being produced is the result of an unbranched vocal tract excited at the glottis, then the resultant acoustic tube shape will approximate fairly well to the area function of the vocal tract that produced the sound. In general, however, this will not be so, and some of the poles of the filter will often be real, with a role for controlling general spectral balance of the signal instead of providing resonant modes. The analysis method used for this type of synthesizer, known as **linear predictive coding** (see Chapter 4), is not concerned with modelling the articulation, but merely with optimizing the filter specification to make the best approximation to acoustic properties of the signal. It is just an incidental property of the method that, under the right conditions (i.e. for non-nasal vowels), the articulatory approximation of the equivalent tube is not too bad.

A practical alternative to articulatory synthesis is merely to generate the signal by means of excitation provided to a set of resonators for representing the individual formants. If there are, say, five formants to be modelled within the desired audio bandwidth, then five resonators will be needed. There are two basic ways in which such resonators can be connected—ca**scade** or **parallel**. If they are connected in cascade (Figure 2.13) there is only one amplitude control, and the relative intensities of the formants are determined entirely by their frequencies and their damping factors or bandwidths. If sampled-data resonators are used, the resultant all-pole filter will be exactly equivalent to the acoustic tube model referred to in the last paragraph, except that the filter will be specified in terms of formant frequencies and bandwidths instead of tube reflection coefficients. This different method of specification has the advantage that it can be easily related to the acoustic properties of speech, as seen on spectrograms, but it will not in general be easy to model the effects of varying vocal effort on the voiced source spectrum, of excitation inserted further forward than the glottis, or of nasal coupling. For these reasons cascade formant synthesis in its simplest form is normally used only

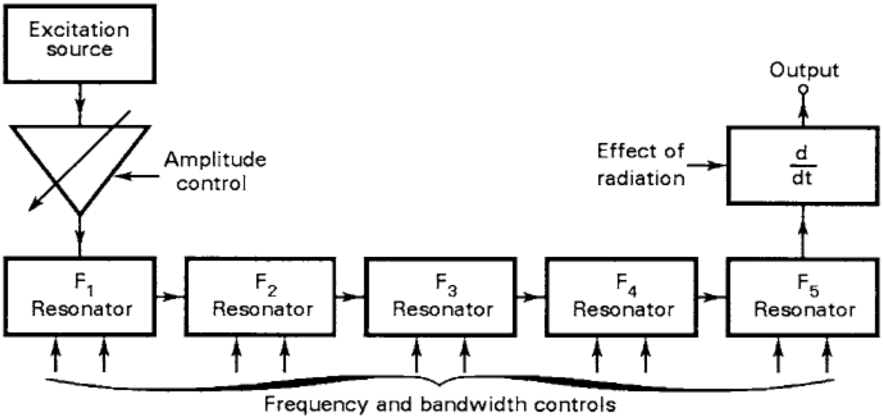


Figure 2.13 Cascade connection of formant generators.

for modelling non-nasal vowels, and complete synthesizers using this method are usually provided with other methods for generating plosives, fricatives and nasals.

The claimed advantage of cascade formants over the parallel connection for non-nasal vowels is that they offer a theoretical representation of the unbranched acoustic tube. While this is true, a synthesizer with parallel resonators is in fact able to approximate just as well to the unbranched tube for all practical purposes, if sufficient care is taken over the design details.

If a small number (e.g. 4 or 5) of analogue resonators is used to model the formants in a cascade synthesizer, the transfer function will not have the infinite series of poles that is present with sampled-data systems. The result in the speech frequency range will be to remove the combined effect of the lower skirts of the infinite number of periodic poles that the sampled-data implementation would give, so making the upper formants much less intense. When analogue cascade synthesizers were still in common use, the solution to this problem was to include a special **higher-pole correction circuit** which gave an approximation to the effect of the missing higher poles that was fairly accurate up to about 5 kHz.

With a parallel formant synthesizer, outputs from the separate resonators are added, and each one has a separate gain control to vary the formant intensity. The increase in formant amplitude when two formants move close in frequency, which occurs automatically in an all-pole cascade formant synthesizer, has to be specified explicitly in the amplitude controls of a parallel system. Thus more control information needs to be provided. The transfer function of a parallel connection of resonators has the same poles as the cascade connection, but will also in general have zeros as a direct consequence of the parallel paths. The zero co-ordinates can be found by putting the second-order transfer functions of the individual resonators over a common denominator, and then factorizing the resultant numerator.

The phase characteristics of individual resonators show a change from 90° lead to 90° lag as the frequency passes resonance, and so between any pair of adjacent formants the phase characteristics will differ by approximately 180°. If the formant waveforms are combined by simple addition there will be a deep dip in the spectrum, caused by the zeros, between every pair of formants, and the lower skirts of all formants will add in phase below the frequency of F_1 (Figure 2.14). This pattern of response does not occur in the spectrum of human speech. If on the other hand the gain coefficients of adjacent formants have opposite signs the response from the skirts of neighbouring formants will reinforce each other, and there will be substantial lower-skirt cancellation below F_1. In effect this change will move the

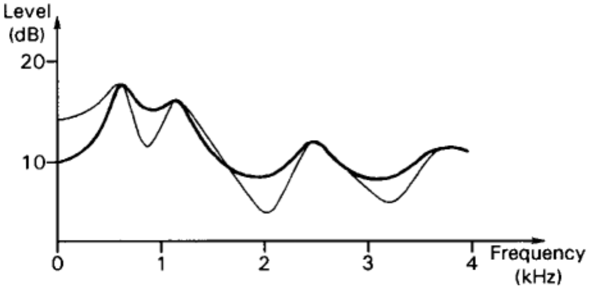


Figure 2.14 Thick line: typical spectral envelope during a vowel. Thin line: the result of simple addition of the outputs from parallel formant generators.

zeros of the transfer function well away from the frequency axis in the *s* plane (or from the unit circle in the *z* plane for sampled-data synthesizers). In fact, if all resonators are arranged to have the same bandwidth, it is possible to choose the gain coefficients so that the transfer function reduces to an all-pole one, and the parallel circuit will then have the identical response to a cascade connection.

The advantage of a parallel formant synthesizer is that it is possible to achieve a reasonable approximation to the spectral shape of sounds for which the all-pole design is not well suited. As the perceptual properties of speech are largely determined by the frequencies and intensities of the formants, it is possible to represent these properties very well by a suitable excitation signal feeding a parallel set of resonators with the appropriate frequency and amplitude parameters. In fact it is most convenient in such a system to incorporate the different spectral trends of the different types of excitation simply by varying the formant amplitude controls, so that both voiced and voiceless excitation are arranged to have the same spectral envelope. The difficulties with this method arise when formants in different parts of the spectrum are set to have very different amplitudes. During voiceless fricatives the high formants will be very intense, while F_1 will be extremely weak, and the reverse will occur in nasal consonants. The skirts of the frequency responses of some resonators may then be of comparable level to the signal required at the peaks of the other, less intense, resonances. There will then be a danger that the assumptions about spectral shape between the formants will not be justified, and the overall response will not be acceptable, particularly at the very low and very high ends of the spectrum.

By putting very simple filter circuits in each formant output to modify the shape of the formant skirts before mixing, it is possible to achieve a very acceptable approximation to the overall spectral shape of any speech sounds using a parallel formant synthesizer. A block diagram of a complete formant system of this type is shown in Figure 2.15. There are two features of the spectral shape specification that deserve special comment. The first is the use of a special heavily damped low-frequency resonator (F_N) below the frequency of F_1. By choosing the appropriate setting of A_{LF} it is possible to control the level in the bottom few hundred Hz independently of the level of F_1, to give a better control of the low-frequency spectral shape of vowels, and more particularly of nasalized vowels and nasal consonants. The second feature is the use of a fixed filter with multiple resonances in the F_4 region. Because of the cross-dimensions of the vocal tract the region above 3 kHz often has many more resonant modes than are predicted by the simple acoustic tube theory, but the detail of the spectral shape in this region is not important. The approximation given by this fixed filter, when it is supplied with a suitable amplitude control, is perceptually sufficient to achieve the highest quality for speech band-limited to about 4 kHz. Additional fixed filters with associated amplitude controls can be used for synthesis at frequencies above 4 kHz.

The excitation circuits for the synthesizer shown in Figure 2.15 have provision for mixed voiced and voiceless sound sources. In human speech, when voiced and turbulent sources are simultaneously in operation, the lower formants will be predominantly voiced, whereas the upper formants will be predominantly voiceless. This difference arises as a direct consequence of the different spectral balance and different points of application of the two sources. The synthesizer shown in Figure 2.15 has a separate excitation mixing circuit for each formant

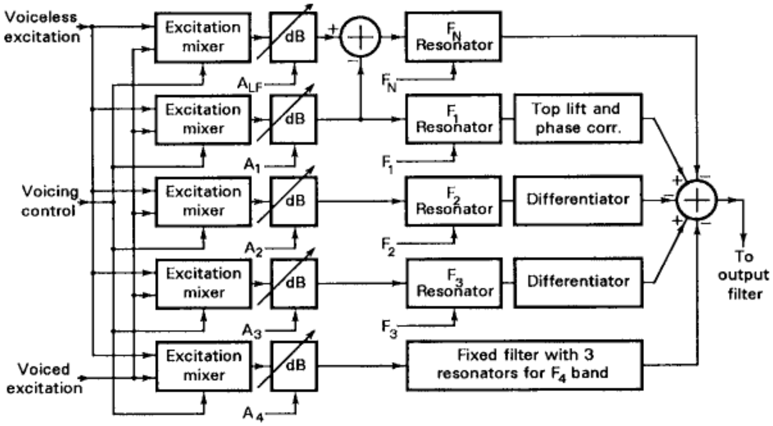


Figure 2.15 Block diagram of the parallel-formant filter system described in Holmes (1983).

generator. The mixing fraction is arranged to be different for the different formants, and is controlled by an overall **degree-of-voicing** signal. When the speech is half voiced the lowest formant receives only voiced excitation, the middle ones have a mixture, and the highest formant is completely voiceless. Experience has shown that this method is extremely successful in generating voiced fricatives and stops, and also breathy vowels.

CHAPTER 2 SUMMARY

- Models of human speech production help understanding of the nature of speech signals as well as being directly useful for speech generation.
- Muscular force on the lungs provides air flow which is modulated by vibrating vocal folds in the larynx, or by turbulence or a blockage in the vocal tract. The resultant sound sources have their spectra modified by the acoustic resonances (or formants), whose frequencies are controlled by position of the tongue, etc.
- Vowels and some consonants normally use the vocal fold sound source, which is periodic with a fundamental frequency (which determines the pitch) typically in the range 50–400 Hz, and thus has a line spectrum. Most other consonants use turbulence as the main sound source.
- For many sounds the resonant structure can be approximately analysed using electrical transmission line theory by representing it as a set of acoustic tubes of different cross-section butted together. It has a transfer function which has only poles and, in spite of the idealizations in the analysis, this function gives a fairly good specification of the three or four main resonances.
- For most consonant sounds the simple acoustic tube analogue is not really adequate. Nasal sounds require the very complicated structure of the nose to be considered, and the way it couples with the pharynx and mouth. Other consonants normally have a close constriction in the mouth, which effectively decouples the back part of the acoustic system and makes it easier to consider only the part in front of the constriction.

- During many voiced sounds the effect of the glottal impedance on the resonances of the vocal tract causes a substantial extra damping when the vocal folds are at their farthest apart.
- The effect of radiation at the mouth can be well represented by simple differentiation of the volume flow waveform.
- Spectrograms are generally a more useful way of displaying the significant properties of speech sounds than are waveforms. Narrow-band spectrograms clearly show the changes in pitch, whereas wide-band spectrograms are better for illustrating formant structure.
- The speech production process can be modelled electronically, and such models are used as practical speech synthesizers. These days they are mostly implemented digitally using sampled-data techniques.
- The details of vocal fold vibration need not be copied closely for realistic speech synthesis, provided the main acoustic consequences are represented. Turbulence is easily represented by random electrical noise or a pseudorandom sequence generator, and plosive excitation by a single impulse or by a very short burst of random noise.
- Speech synthesis can in principle use vocal tract (articulatory) models for the resonant system, but these are mostly too complicated to control except for the special case of linear predictive coding (LPC, see Chapter 4). More practical models use explicit separate resonators for the formants. The resonators can be connected in cascade, which is theoretically attractive for vowels but unsuitable for most consonant sounds. When carefully designed, a parallel resonator system using individual amplitude controls can give excellent results for all types of speech sound. A realistic representation of mixed vocal-fold and turbulent excitation can also be provided in a parallel formant system.

CHAPTER 2 EXERCISES

E2.1 What are the main differences between the spectra of voiced and voiceless excitation?

E2.2 What are the possible contributions to voiced excitation besides air flow through the glottis?

E2.3 Discuss the idealizing assumptions that are made in the simple theory of vocal tract acoustics.

E2.4 Describe the effects of having the sound source remote from the glottis during consonant production.

E2.5 In what ways are fundamental frequency and formant frequencies interdependent?

E2.6 Give examples of factors which influence formant bandwidth.

E2.7 Why is a spectrogram more useful than the waveform when studying the communication function of speech signals?

E2.8 Discuss the different uses of wide-band and narrow-band spectrograms.

E2.9 Summarize the relative merits of cascade and parallel formant synthesis.

E2.10 Why has articulatory synthesis been less successful than formant synthesis?

CHAPTER 3

Mechanisms and Models of the Human Auditory System

3.1 INTRODUCTION

When considering the requirements for speech synthesis, or methods for automatic speech recognition, much insight can be gained from knowledge about the workings of the human auditory system. Unfortunately, because of the invasive nature of most physiological studies and the large number and extremely small size of the neurons involved, study in this area has been extremely difficult, and our knowledge is very incomplete. Even so, over the recent decades much progress has been made, with a combination of psychophysical studies on humans, neurophysiological studies on experimental animals, and computer modelling to investigate plausible hypotheses.

3.2 PHYSIOLOGY OF THE OUTER AND MIDDLE EARS

Figure 3.1 illustrates the structure of the human ear. The outer ear consists of the **pinna,** which is the visible structure of the ear, and the passage known as the **auditory canal**. Sound impinging on the side of the head travels down the auditory canal to reach the **eardrum,** or **tympanic membrane**. However, the pinna plays a significant role because the effect of reflections from the structures of the pinna is to introduce spectral changes at high frequencies, which can be used to judge the direction of a sound source. The effect is confined to frequencies above 3 kHz or so, as it is only at these high frequencies that the wavelength of the sound is short enough for it to interact with the structures of the pinna. The length of the auditory canal is such that it forms an acoustic resonator, with a rather heavily damped main resonance at about 3.5 kHz, and some slight secondary resonances at higher frequencies. The principal effect of this resonant behaviour is to increase the ear's sensitivity to sounds in the 3–4 kHz range. Sound arriving at the eardrum causes it to vibrate, and the vibrations are transmitted through the middle ear by three inter-connected small bones, known as the **ossicles** and comprising the **malleus, incus** and **stapes**. The stapes is in contact with the **oval window,** which is a membrane-covered opening at one end of the **cochlea**. The cochlea is the main structure of the inner ear. The ossicles vibrate with a lever action, and enable the small air pressure changes that vibrate the eardrum to be coupled effectively to the oval window. In this way the ossicles act as a transformer, to match the low acoustic impedance of the eardrum to the higher impedance of the input to the cochlea.

Although the pinna and the ossicles of the middle ear play an important role in the hearing process, the main function of processing sounds is carried out within the cochlea and in higher levels of neural processing.

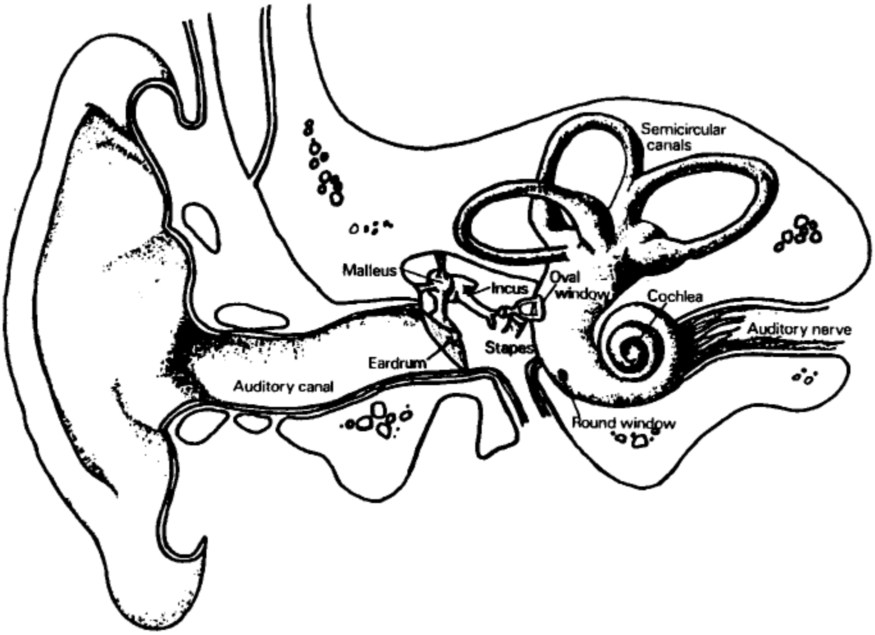


Figure 3.1 Structure of the peripheral auditory system. (Figure from HUMAN INFORMATION PROCESSING: AN INTRODUCTION TO PSYCHOLOGY by Peter H.Lindsay and Donald A.Norman, copyright © 1972 by Harcourt, Inc., reproduced by permission of the publisher.)

3.3 STRUCTURE OF THE COCHLEA

As can be seen from Figure 3.1, the cochlea is a spiral tapered tube, with the stapes in contact with the outer, larger cross-section, end. At this end also are the **semicircular canals,** whose main function is control of balance, rather than hearing. Figure 3.2 shows a section through one turn of the spiral, and it can be seen that it is divided along its length into three parts by two membranes. The three parts are known as the **scala vestibuli,** the **scala media** and the **scala tympani**. The scala media is filled with a fluid known as **endolymph**. The structure separating the scala vestibuli and the scala tympani stops just short of the inner end of the cochlear spiral, to leave a small interconnecting opening known as the **helicotrema**. Both of these scalae are filled with another fluid, **perilymph**. One of the membranes, **Reissner's membrane,** is relatively wide, and serves to separate the fluids in the scala media and the scala vestibuli but has little effect acoustically. The other membrane, the **basilar membrane (BM),** is a vital part of the hearing process. As can be seen, the membrane itself only occupies a small proportion of the width of the partition between the scala media and the scala tympani. The remainder of the space is occupied by a bony structure, which supports the **organ of Corti** along one edge of the BM. Rather surprisingly, as the cochlea becomes narrower towards the helicotrema, the BM actually becomes wider. In humans it is typically 0.1 mm wide at the basal end, near the oval window, and is 0.5 mm wide at the apical end, near the helicotrema.

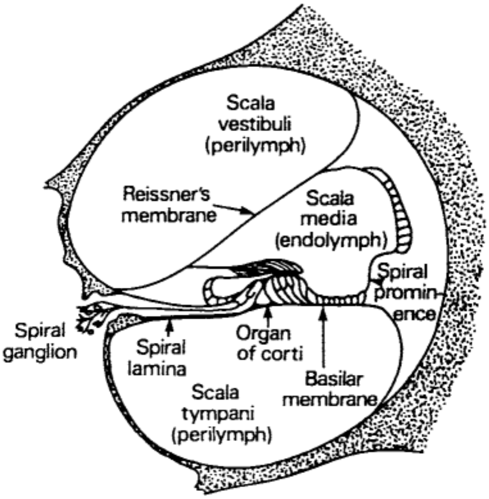


Figure 3.2 Cross-section of one turn of the cochlear spiral.

As the stapes vibrates, and so causes movements in the incompressible perilymph, there are compensatory movements of the small membrane covering the **round window,** which is situated near the basal end of the scala vestibuli. If the stapes is given an impulsive movement, its immediate effect is to cause a distortion of the basal end of the BM. These initial movements are followed by a travelling wave along the cochlea, with corresponding displacements spreading along the length of the BM. However, the mechanical properties of the membrane in conjunction with its environment cause a resonance effect in the membrane movements; the different frequency components of the travelling wave are transmitted differently, and only the lowest audio frequency components of the wave cause any significant movement at the apical end.

Because of the frequency dependence of the wave motion in the cochlea it is informative to study the response of the BM to sinusoids of different frequencies. The pioneering measurements of the movement of the membrane in cadaver ears by von Békésy (1947) showed a filtering action, in which each position along the membrane was associated with a different frequency for maximum response. The highest audio frequencies cause most response near the basal end, and the lowest frequencies cause a peak response near the helicotrema. However, the frequency selectivity is not symmetrical: at frequencies higher than the preferred frequency the response falls off more rapidly than for lower frequencies. The response curves obtained by von Békésy were quite broad, but more recent measurements from living animals have shown that in a normal, healthy ear each point on the BM is in fact sharply tuned, responding with high sensitivity to a limited range of frequencies. The sharpness of the tuning is dependent on the physiological condition of the animal, as is evident from the **tuning curves** shown in Figure 3.3. The sharp tuning is generally believed to be the result of biological structures actively influencing the mechanics of the cochlea. The most likely structures to play this role are the **outer hair cells,** which are part of the organ of Corti.

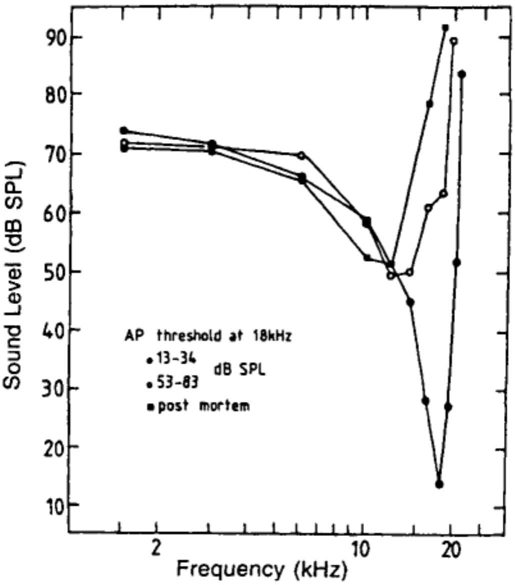


Figure 3.3 Tuning curves representing measurements at a single point on the BM of a guinea pig. Each curve shows the input sound level required to produce a constant velocity on the BM, plotted as a function of stimulus frequency. The curve marked by solid circles was obtained at the start of the experiment when the animal was in good physiological condition, the one marked by open circles after deterioration during the experiment, and the one marked with solid squares following death of the animal. (Reprinted with permission from Sellick *et al.*, 1982. Copyright © 1982, Acoustical Society of America.)

The magnitude of the BM response does not increase directly in proportion with the input magnitude: although at very low and at high levels the response grows roughly linearly with increasing level, in the mid-range it increases more gradually. This pattern shows a **compressive non-linearity,** whereby a large range of input sound levels is compressed into a smaller range of BM responses.

For the purpose of hearing, the frequency-selective BM movements must be converted to a neural response. This transduction process takes place by means of the **inner hair cells** in the organ of Corti. In a normal human ear this organ contains around 25,000 outer hair cells and 3,500 inner hair cells, arranged in rows spread along the cochlear spiral and attached to one side of the BM. Movement of the BM causes bending of the hair cells, and so stimulates firing of the neurons in the auditory nerve. This transduction function is performed by the inner hair cells, while the outer hair cells have the very different role of actively influencing cochlear mechanics to maximize sensitivity and selectivity as described above.

3.4 NEURAL RESPONSE

Just as the mechanical movement of a point on the BM can be investigated as a function of frequency, so it is also possible to study the rate of firing for single

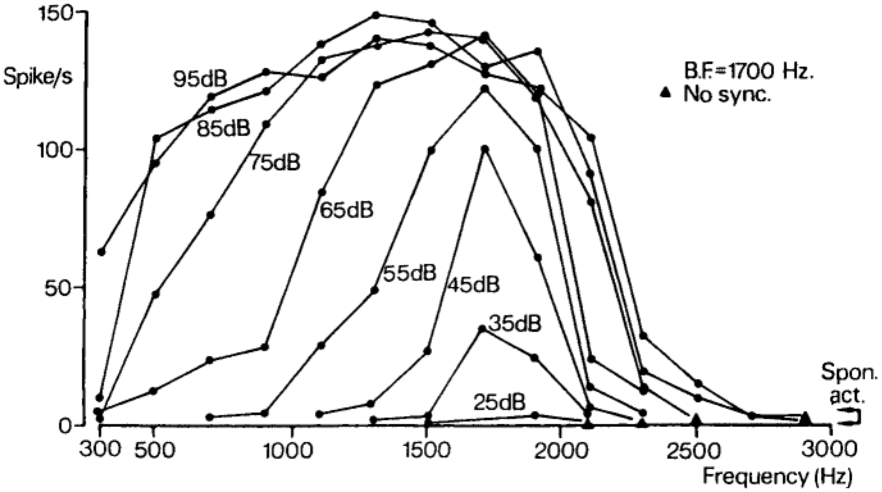


Figure 3.4 Firing rates from a single auditory nerve fibre of a squirrel monkey at various intensity levels. (Rose *et al.*, 1971. Reproduced by permission of the American Physiological Society.)

nerve fibres in the auditory nerve. It is found that each fibre has a **characteristic frequency,** at which it is most easily stimulated to fire; as might be expected, this frequency is closely related to the part of the BM associated with the corresponding inner hair cell. In addition, characteristic frequencies are distributed in an orderly manner in the auditory nerve, so that the place representation of frequency along the BM is preserved as a place representation in the auditory nerve. If a tuning curve is plotted for a single neuron, showing response threshold as a function of frequency, the sharpness of tuning is found to be very similar to the sharpness of tuning for the healthy BM as shown in Figure 3.3.

The neural transduction process is extremely non-linear and so the shape of a curve plotting firing rate as a function of frequency depends very much on signal level (see Figure 3.4). Most neurons show some spontaneous firing, even in the absence of stimulation, and they rarely respond at mean rates in excess of a few hundred firings per second even for very intense stimuli. When a neuron no longer responds to an increase in sound level with an increase in firing rate, it is said to be **saturated**. Most neurons have a fairly high rate of spontaneous firing, and a **dynamic range** of levels between threshold and saturation of around 20–50 dB. A small proportion of neurons have a lower spontaneous rate and a much wider dynamic range, and are useful at high sound levels.

Figure 3.4 does not, of course, attempt to indicate the precise times of firing of the neurons in relation to the instantaneous value of the sinusoidal stimulus. There is a strong tendency for individual firings to be at roughly the same points on the sine wave cycle, so the ‘spikes’ of waveform detected on the nerve fibre show interspike intervals that are very close to integer multiples of one period of the stimulating signal. At stimulation frequencies above about 4 kHz, this tendency to **phase locking** is no longer apparent, mainly because capacitance of the inner hair cells prevents them from changing in voltage sufficiently rapidly.

Research is continuing into the processes of neural coding of audio signals, and they are certainly not yet fully understood. There is still considerable debate over the extent to which information provided by the timing of neural impulses is used in perceptual processes. For example, timing information may contribute to our ability to distinguish between different sounds of high intensity. This ability is still quite good, even when the intensities of the frequency components are such that at least most of the neurons at the appropriate characteristic frequencies can be expected to be fully saturated, and firing at their maximum rates. It is known that, as the level of a frequency component of a stimulus is increased, both the degree and the regularity of phase locking to that component show an increase. Thus changes in the pattern of phase locking could contribute to the detection of changes in spectral content (for frequencies up to about 4 kHz). It is likely that phase locking is relevant to our perception of the pitch of pure and complex tones.

3.5 PSYCHOPHYSICAL MEASUREMENTS

In the absence of complete knowledge of the physical processes of auditory analysis, it is useful to measure the functional performance of the hearing system by means of psychophysical experiments. In such experiments, various types of auditory stimuli are given to human subjects, who are asked to respond in various ways according to what they hear.

One of the most basic types of auditory measurement is known as an **audiogram,** which displays the r.m.s. pressure of sound which is just audible as a function of frequency of sinusoidal stimulation. This display can be extended to include plots of subjective judgements of equal loudness at different frequencies, for levels well above the threshold of detection. Such a display is shown in Figure 3.5. The units of loudness are **phons,** which are defined as the level in

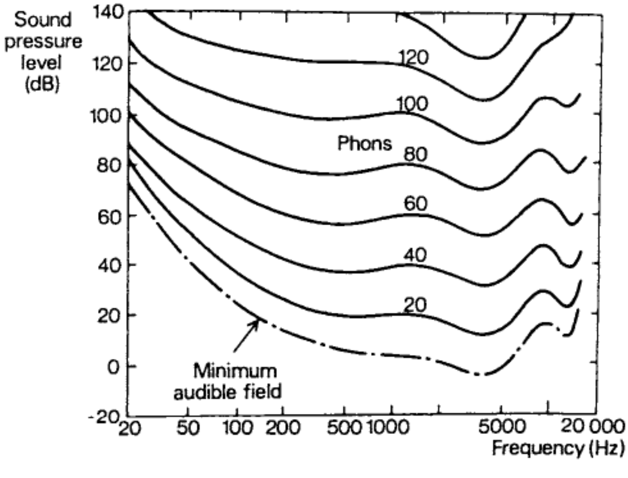


Figure 3.5 Curves showing the sound pressure level needed for various perceived loudness values, (redrawn from Robinson and Dadson, 1956. © Crown Copyright 1956. Reproduced by permission of the Controller of HMSO.)

decibels (dB) of a tone at 1 kHz that would be judged to be of the same loudness as the test stimulus. The reference level for 0 dB **sound pressure level** (**SPL**) has been arbitrarily adopted to be 2×10^{-5} N/m^2. This level was chosen because it is approximately equal to the average threshold of hearing for humans with normal hearing at 1 kHz (which is a frequency at which our ears are nearly at their most sensitive). On Figure 3.5 the small ripples in auditory sensitivity in the range from 1 to 10 kHz are caused by the standing wave resonances in the auditory canal.

Perceptual experiments can be conducted to investigate the frequency selectivity of the auditory system, and to estimate the characteristics of the **auditory filters**. These measurements are generally made using the technique of **masking**. There are many different types of masking experiment for determining the frequency resolution capabilities of the ear, of which the following is a typical example. If a low-level sinusoid (or **pure tone**) is mixed with a narrow band of random noise of much higher level and centred on the same frequency, perception of the tone will be masked by the noise. In general the presence of the tone cannot be detected if the noise power is more than a few dB above that of the tone. If the centre frequency of the noise is now shifted away from the tone frequency, it will not cause its main effect at the same place on the BM, and so its masking action is reduced. A **psychophysical tuning curve** (**PTC**) can be derived for any tone frequency by plotting the level of noise that is just sufficient to mask the tone, as a function of noise centre frequency. These curves can be plotted for different tone frequencies and various stimulus levels. The PTCs so derived (see Figure 3.6 for some typical examples) are generally similar in form to the BM response curves shown in Figure 3.3 and also to neural tuning curves.

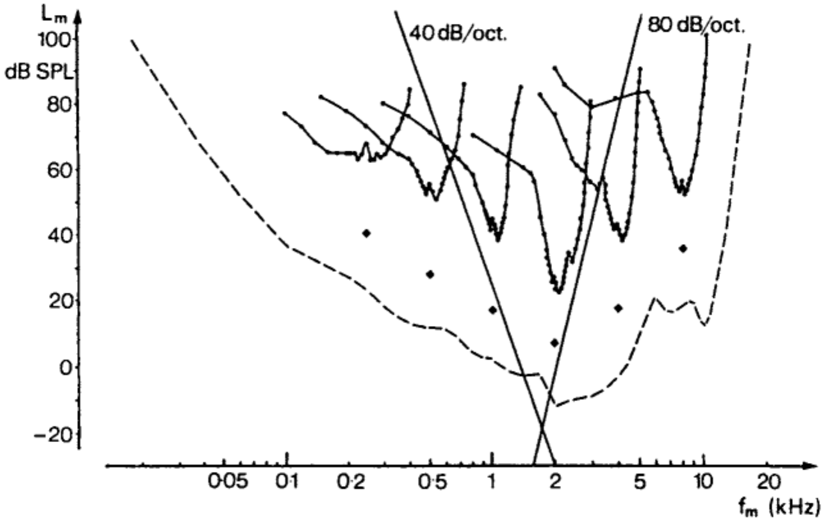


Figure 3.6 Psychophysical tuning curves found by a masking experiment. The dashed line shows the absolute threshold of hearing for the subject. The filled diamonds indicate the frequencies and levels of the six short probe tones that were used. The curves show the corresponding levels of masker tones needed at various frequencies to just obscure the probe tones. The superimposed sloping lines represent slopes of 40 and 80 dB/octave. (Adapted from Vogten (1974), copyright © 1974 by Springer-Verlag GmbH & Co., reproduced by permission of the publisher and of the author.)

The useful dynamic range of representation and variation of resolution with frequency are such that it is most useful to plot PTCs with logarithmic intensity and frequency scales. The examples shown in Figure 3.6 were derived by using a pure tone as the masker, and a brief low-level tone as the probe signal. The response at frequencies above the peak of sensitivity falls off at more than 80 dB/octave. The steepest slope below the peak response is nearer 40 dB/octave, but the slope then reduces to a lower value as the frequency is further reduced. This asymmetry indicates that intense low frequencies are able to mask higher-frequency test tones much more than high frequencies will mask lower ones.

The bandwidth between 3 dB points (i.e. the points indicating a change in power by a factor of two) of the PTC is typically between 10% and 15% of the centre frequency for frequencies above about 1 kHz, and a somewhat larger percentage for lower centre frequencies. However, the estimates of bandwidth vary somewhat according to the precise experimental method used.

Due to the masking effect within the bandwidth of a single filter, the human auditory system is not sensitive to the detailed spectral structure of a sound within this bandwidth, except to the extent that beats between spectral components cause variations of the intensity envelope that are slow enough for the neural system to respond to. The bandwidth over which the main masking effect operates is usually known as the **critical bandwidth**. This term was introduced by Fletcher (1940), who also used the phrase **critical band** to refer to the concept of the auditory filter. The critical bandwidth provides an indication of the effective bandwidth of the auditory filter. However, because the auditory filters do not have a rectangular response in the frequency domain, they are not completely specified by their critical bandwidths. Since Fletcher first described the critical band concept, a variety of different experiments have been carried out to investigate critical band

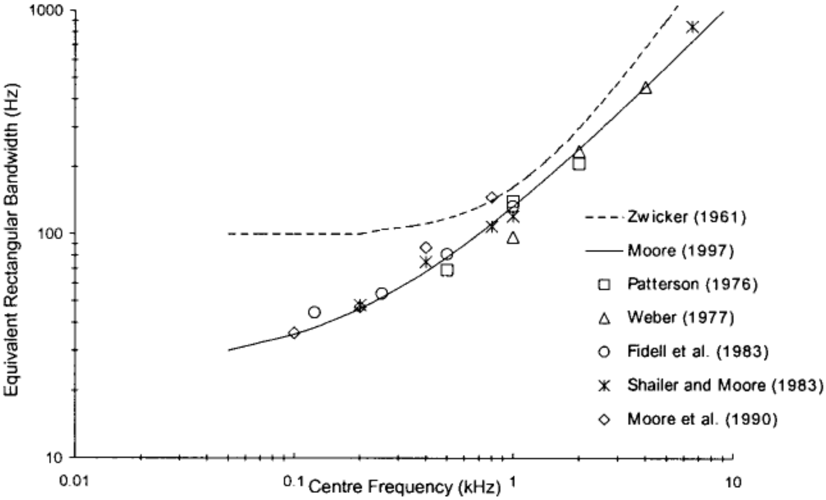


Figure 3.7 Estimates of auditory filter bandwidths. The symbols indicate estimates of the ERB of the auditory filter at various centre frequencies, as obtained by the workers specified. The solid curve shows the frequency-dependent function for ERB given by Moore (1997), which is compared with the dashed curve showing the traditional critical bandwidth function as tabulated by Zwicker (1961).

phenomena and to estimate critical bandwidth. Based on the results of early experiments, Zwicker (1961) tabulated critical bandwidth as a function of centre frequency, and these values are shown graphically as the dashed curve in Figure 3.7. The critical bandwidth was estimated to be constant at 100 Hz for centre frequencies below 500 Hz, while for higher frequencies the bandwidth increases roughly in proportion with centre frequency. Zwicker proposed the **Bark** scale, whereby a difference of 1 Bark represents the width of one critical band over the entire frequency range. The Bark scale corresponds very closely with another perceptual scale, the **mel** scale, which represents the **pitch** (perceived frequency) of a tone as a function of its acoustic frequency.

The 'traditional' critical bandwidth function tabulated by Zwicker was derived when there were relatively few estimates available for low centre frequencies, but more recent experiments have provided evidence that the critical bandwidth continues to decrease at frequencies below 500 Hz. Many of the more recent estimates of critical bandwidth are based on masking experiments to determine the shape of the auditory filter and estimate the **equivalent rectangular bandwidth (ERB)**. The ERB is defined as the bandwidth of an ideal rectangular-passband filter that will transmit the same power of a flat-spectrum input as the auditory filter would, when the gains of both filters are equal at the peak-response frequency. Figure 3.7 shows estimates of the ERB of the auditory filter as a function of frequency taken from a variety of experiments, and also shows a function suggested by Moore (1997) to approximate these data.

3.6 ANALYSIS OF SIMPLE AND COMPLEX SIGNALS

In pitch perception experiments in the mid-audio frequency range, subjects are able to perceive changes in frequency of pure tones of approximately 0.1%. It is thus clear that there is some frequency-determining mechanism that is far more powerful than the mere frequency-selective filtering of the inner ear and its associated low-level neural interactions. At frequencies above 4 kHz pitch discrimination reduces substantially. This fact gives a hint that neural phase-locking may be responsible for passing precise timing information to higher centres, so that some measurement of time intervals is probably involved. Further support for some time-domain processing can be found in the ability to infer sound direction as a result of very small relative delays in signals reaching the two ears.

In the case of complex signals such as speech, it is much less clear what the capabilities and processes of the auditory system are. The range of SPL between signals that are just audible and those that actually cause physical discomfort is more than 100 dB (the SPL of music as produced in a modern discotheque is frequently at the upper end of this range). In the middle of the range, it is possible to vary the intensity of speech signals by at least 20 dB without listeners regarding them as being significantly changed in quality, even though they would regard these as quite substantial changes of loudness. The just-discriminable difference in formant frequency is roughly constant at about 10 Hz in the F1 region (up to around 800 Hz) and then increases to about 20 Hz at 2000 Hz.

The filtering action of the ear is such that the lower harmonics are generally resolved, and so there will be no single peak in response that corresponds to the frequency of F1, which suggests that determination of F1 frequency may be based on relative amplitudes of harmonics rather than on locating a single spectral peak. The higher formants (at least for male voices) do produce clear peaks in the auditory response. There is evidence that peaks in the spectrum of the audio signal are detected more easily than features between spectral peaks, and such preferential treatment of peaks is probably caused by the shape of the auditory filter in combination with higher-level neural processes. By analogy with the fact that the higher levels of the visual system are known to have particular neurons that fire in response to objects of particular shape or orientation, it seems plausible that the auditory system may be able to respond specifically to formant frequency movements of particular rates, such as occur in many consonant transitions.

3.7 MODELS OF THE AUDITORY SYSTEM

Modelling the auditory system's behaviour when exposed to sound is of interest for two reasons. The first is to assist in understanding how sound is interpreted by humans and other living organisms. The second reason is to use the model directly in machines intended for processing sound that would usually be interpreted by human beings—in particular for automatic speech recognition. Humans are extremely competent at interpreting speech-like sounds, even when there may be multiple sound sources, or the sounds are modified by interfering noises or by other influences such as reverberation. A functional model of the auditory system might be a very good first-stage processor in an automatic speech recognizer, because it should retain those features of a speech signal which are used for human speech recognition, but would discard information that humans make no use of.

3.7.1 Mechanical filtering

The modelling of the outer and middle ears is fairly straightforward, at least for low or moderate sound levels, because these parts of the auditory system can then be assumed to be approximately linear and can be represented as a fairly simple electrical filter with appropriate characteristics. The main function of this filter is to model the lowest resonance of the auditory canal.

The filtering of the cochlea presents a more difficult problem. This filtering is a direct consequence of the way the waves travelling along the tapered tube interact with the mechanical properties of the BM. It is possible to represent each small section along the coclear spiral as a section of transmission line, where the constants of each successive section are scaled to represent the narrowing of the cochlea and the changing mechanical properties of the membrane. The function of the outer hair cells to provide adaptive gain control and ensure high sensitivity can be simulated by means of an appropriate feedback circuit. With careful design, cochlear models (implemented either as a lumped-constant electrical analogue or in a computer program) can be shown to yield quite close approximations to physiological measurements. The ability of such models to reproduce real-ear measurements gives considerable confidence that the mechanical filtering in the cochlea is now fairly well understood.

Due to the high computational load involved in implementing travelling wave models in transmission lines, it is often more convenient to achieve the filtering effect by using a number of independent filters, each representing the filtering characteristic at a single point on the BM. Because the individual filters have a fairly broad passband it is possible to represent the continuously distributed filter system reasonably well with as few as about 40 separate channels. Even with this small number the overlap between adjacent channels is so great that all significantly different filtered signals are available in at least one channel.

3.7.2 Models of neural transduction

The relationship between BM motion and neural firing is quite complicated. A typical functional model of the transduction process involves first half-wave rectifying the waveform output from the filtering stage. Physiological evidence suggests that the inner hair cells are only stimulated to release neurotransmitter for movement of the BM in a single direction, and the rectification acts to simulate this characteristic. The next stage is to apply a compression function to reduce the very large dynamic range of the input signal. The output of this process can be used to influence the probability of firing of a model neuron. Models of this type are able to represent saturation of firing rate at high signal levels and phase locking to particular points in the vibration. The probability of firing should, however, also be influenced by the time interval after the immediately previous firing of the same model neuron, so that a much greater stimulus will be needed to cause two firings to occur close in time than is needed for a longer interval. To be realistic, it is also necessary to include a certain amount of randomness, to represent the fact that the time of firing of any one fibre is not precisely determined by the stimulus history. With suitable parameters for this type of model it is possible to simulate the observed firing statistics of real nerve fibres, including the slow spontaneous firing that occurs in the absence of stimulation, the saturation at high levels, and the tendency to phase locking. By also making the firing probability depend on the average rate over the previous few tens of milliseconds it is possible to incorporate the short-term adaptation that occurs to steady sound patterns.

There have been many animal studies on the responses of individual fibres in the auditory nerve. There is sufficient similarity in the physiological structure of the ears of humans and experimental animals for us to assume that the effects in human ears are very similar. There is thus a reasonable amount of confidence that the modelling of neural transduction is fairly accurate. Unfortunately it is much more difficult to get physiological data to define the further processing of these neural signals, so any modelling of the higher levels is inherently more speculative, and must be guided by the results of psychophysical experiments.

3.7.3 Higher-level neural processing

A number of models have been suggested for the representation of speech-evoked activity in the auditory nerve. There are two main characteristics that distinguish different modelling approaches: the extent to which a model uses explicit knowledge about a nerve fibre's place of origin in the cochlea, and whether a model is based on instantaneous firing rate alone or on temporal properties of the firing pattern. Thus it is common to divide these models into three categories according to the nature of the representation they use, which may be:

- place/rate (using explicit knowledge of place, and only instantaneous rate),
- place/temporal (using place and local temporal firing pattern), or
- place-independent/temporal (ignoring place and using only temporal properties of the global firing pattern).

Place/rate models

A model based on the pattern of average firing rate as a function of the nerve fibre's characteristic frequency (which is in turn related to 'place' on the BM) can be shown to give a well-defined formant pattern for vocalic sounds at low sound-pressure levels. However, at higher sound-pressure levels typical of normal conversation, saturation effects are such that there is a loss of definition in the spectral pattern. A representation that uses only average firing rate information is thus unable to account for observations that speech intelligibility *improves* as sound-pressure level increases, and cannot fully account for the frequency resolution and dynamic range of the auditory system. Several workers have therefore developed models that make use of the synchrony between the firings of different neurons. We have no clear understanding of how real nervous systems might detect such synchrony, but functional models have been developed that seem to have the right properties. Examples of these models are described briefly below.

Place/temporal models

The idea behind place/temporal models is to compare intervals between firings of an auditory neuron with the reciprocal of the neuron's characteristic frequency (i.e. the period corresponding to the preferred frequency for the neuron). One such model is the **generalized synchrony detector (GSD)** developed by Seneff (1988).

In Seneff's model, the signals from the neural transduction stage are assumed to represent estimates of the probability of firing for neurons connected to the corresponding parts of the BM. If there is a dominant peak in the input spectrum at the preferred frequency for a point on the BM, the response waveform representing the probability of firing for that point will be a half-wave rectified signal, roughly periodic at the frequency of the spectrum peak. Let this waveform be represented by *u(t)*. Define a quantity τ as the reciprocal of the characteristic frequency for the neurons being considered. It follows that *u(t)* and *u(t-*τ) will be very similar. For a nearby point on the membrane, the cochlear filter will still be dominated by the same spectral peak, so the periodicity of the neural response will be the same. However, the characteristic frequency of the neurons associated with this point on the membrane will be slightly different. The value of *t* corresponding to this point will therefore no longer correspond to one period of the waveform, so *u(t)* will now not be nearly the same as *u(t-*τ*)*. Seneff's GSD uses the following ratio:

$$r = \frac{\left|u(t) + u(t-\tau)\right| - \delta}{\left|u(t) - u(t-\tau)\right|}, \tag{3.1}$$

where δ is a small constant representing a threshold that is included in order to reduce the value of the ratio for very weak signals. Aside from the thresholding of very weak signals, the use of the sum waveform in the numerator of Equation (3.1) ensures that the peak value of r is independent of the peak magnitude of u. The value of r can be very large for points on the simulated membrane whose preferred frequencies are close to the frequency of a spectral peak, but will become much smaller when moving quite small distances along the membrane. To avoid problems with r becoming arbitrarily large in the case of exact periodicity at a channel centre frequency, Seneff found it useful to compress the range of the output by applying a saturating non-linear function prior to subsequent processing.

The outputs from Seneff's GSD and from similar devices developed by other workers have been found to be extremely sensitive for detecting spectral peaks associated with formants, although in the form described here they give no indication of formant intensity. Seneff was also able to make use of intensity by having her complete auditory model give an additional set of output signals depending on the mean firing rate of the simulated neurons.

Place/temporal models are able to provide a good account of data for speech intelligibility across a wide range of sound pressure levels. However, it is more difficult to use these models to explain evidence, both from studies of people with selective hearing loss and from masking experiments, that speech intelligibility is not necessarily correlated with the effectiveness of neurons with particular characteristic frequencies. It is also unclear exactly how the necessary information about a nerve fibre's characteristic frequency would be obtained in a real system.

Place-independent/temporal models

An alternative to a place/temporal spectral representation is to use information about firing synchrony without any reference to the nerve fibres' characteristic frequencies. One model based on this idea is the **ensemble interval histogram (EIH)** model developed by Ghitza (1988, 1992). In Ghitza's model, an interval histogram is constructed for the neurons corresponding to each auditory channel, and an EIH is then obtained by combining these histograms across all channels. At moderate and high sound pressure levels, a spectral peak will cause a pattern of common synchrony across neurons for several channels. The timing can be used to determine the frequency of the peak, while the extent of the common synchrony across different neurons provides information about amplitude.

It seems plausible that all the types of representation described above could contribute to give a robust system for the perception of speech signals, with the optimum representation being dependent on the acoustic environment. Many separate research groups have experimented with various different functional models for the properties of the higher levels of the auditory system, and research effort is still continuing in this area. While auditory models have not yet been widely adopted as the front-end representation for automatic speech recognition systems, some encouraging results have been reported from preliminary speech recognition experiments based on the use of such models, especially under noisy conditions. The choice of front-end representation for use in automatic speech recognizers will be discussed further in Chapters 10 and 11.

CHAPTER 3 SUMMARY

- The outer ear has a damped resonance which enhances the response of the tympanic membrane at around 3.5 kHz. The ossicles of the middle ear couple the vibrations of the tympanic membrane to the spiral-shaped cochlea of the inner ear.
- The cochlea is filled with fluids, and divided along its length by the basilar membrane (BM) except for a small gap at the inner end of the spiral (the helicotrema). The BM in a normal healthy ear shows a sharply-tuned resonant behaviour, and its resonant frequency varies along its length, being high at the outer end of the cochlea and low near the helicotrema.
- Hair cells in contact with the membrane are coupled to nerve cells, and convert the membrane vibrations into neural firings in the auditory nerve. The mean rate of firing is a very non-linear function of vibration amplitude, and individual firings tend to be at a fixed part of the vibration cycle of the corresponding part of the BM.
- Psychophysical tuning curves are derived using masking techniques to show human ability to separate the responses to individual frequency components of a complex signal. Separation is not effective for components closer than the bandwidth of the auditory filter, which is about 10% of centre frequency above 1 kHz and a somewhat larger percentage for lower centre frequencies.
- Human ability to judge the pitch of tones and the frequency of resonances is much better than indicated by the width of critical bands, and is believed to be the result of analysing the timing pattern of neural firings.
- A simple filter can provide a good model of the outer and middle ears. The filtering of the cochlea can be modelled as a series of transmission line sections or as a set of discrete filters. Suitable non-linear functions applied to the filter outputs can provide estimates of the probability of neurons firing. Further processing to emulate the human ability to detect frequency changes needs to somehow exploit temporal information in the pattern of neural firings.
- Auditory models are showing promise as acoustic analysers for automatic speech recognition.

CHAPTER 3 EXERCISES

E3.1 Discuss the relationship between psychophysical tuning curves, neural tuning curves, and basilar membrane response as a function of frequency.

E3.2 Suggest possible explanations for the wide dynamic range of the human ear, given that the firing rates of individual hair cells saturate at moderate sound levels.

E3.3 Comment on the difference in frequency discrimination for simple tones and for spectral features of complex sounds, such as speech.

E3.4 Discuss the role of non-linearity in neural transduction in the ear.

E3.5 Why should it be advantageous to use models of the auditory system for speech signal analysis in automatic speech recognition?

CHAPTER 4

Digital Coding of Speech

4.1 INTRODUCTION

There are two justifications for including a chapter on speech coding in a book on speech synthesis and recognition. The first is that some specialized low-data-rate communication channels actually code the speech so that it can be regenerated by synthesis using a functional model of the human speaking system, and some systems even use automatic speech recognition to identify the units for coding. The second justification arises, as will be explained in Chapter 5, because a common method of automatic speech synthesis is to replay a sequence of message parts which have been derived directly from human utterances of the appropriate phrases, words or parts of words. In any modern system of this type the message components will be stored in digitally coded form. For these reasons this chapter will briefly review some of the most important methods of coding speech digitally, and will discuss the compromises that must be made between the number of digits that need to be transmitted or stored, the complexity of the coding methods, and the intelligibility and quality of the decoded speech. Most of these coding methods were originally developed for real-time speech transmission over digital links, which imposes the need to avoid appreciable delay between the speech entering the coder and emerging from the decoder. This requirement does not apply to the use of digital coding for storing message components, and so for this application there is greater freedom to exploit variable redundancy in the signal structure.

To reproduce an arbitrary audio signal it is possible to calculate the necessary information rate (bits/s) in terms of the bandwidth of the signal and the degree of accuracy to which the signal must be specified within that bandwidth. For typical telephone quality the bandwidth is about 3 kHz and the signal-to-noise ratio might be 40 dB. The information rate in this case is about 40,000 bits/s. For a high-fidelity monophonic sound reproducing system the bandwidth would be about five times greater, and the noise would probably be 60–70 dB below the peak signal level. In this case a rate of about 300,000 bits/s is required to specify any of the possible distinct signals that could be reproduced by such a system.

In contrast to these very high figures, it is known that human cognitive processes cannot take account of an information rate in excess of a few tens of bits per second, thus implying a ratio of information transmitted to information used of between 1,000 and 10,000. This very large ratio indicates that the full information capacity of an audio channel should not be necessary for speech transmission. Unfortunately for the communications engineer, the human listener can be very selective in deciding what aspects of the signal are chosen for attention by the few tens of bits per second available for cognitive processing. Usually the listener concentrates on the message, which, with its normal high degree of linguistic redundancy, falls well within the capacity available. However, the listener may pay attention specifically to the voice quality of the speaker, the background noise, or even to the way certain speech sounds are reproduced.

There are two properties of speech communication that can be heavily exploited in speech coding. The first is the restricted capacity of the human auditory system, explained in Chapter 3. Auditory limitations make the listener insensitive to various imperfections in speech reproduction. When designing speech coding systems it can also be advantageous to make use of the fact that the signal is known to be produced by a human talker. As explained in Chapter 2, the physiology of the speaking mechanism puts strong constraints on the types of signal that can occur, and this fact may be exploited by modelling some aspects of human speech production at the receiving end of a speech link. The potential reduction in digit rate that can ultimately be achieved from this approach is much greater than is possible from exploiting auditory restrictions alone, but such systems are only suited to auditory signals that are speech-like.

Coding methods can be divided into three general classes, thus:

1. simple waveform coders, which operate at data rates of 16 kbits/s and above;
2. analysis/synthesis systems, which are most useful at low rates from 4 kbits/s down to less than 1,000 bits/s and, in the extreme, as low as about 100 bits/s;
3. intermediate systems, which share some features of both of the first two categories and cover a wide range of rates in the region of 4–32 kbits/s.

Members of each class exploit aspects of production constraints and of perception tolerance, but to varying extents for different types of coders. In the following discussion of individual coding methods some mention will be made of the extent to which properties of perception and production are exploited.

4.2 SIMPLE WAVEFORM CODERS

4.2.1 Pulse code modulation

Waveform coders, as their name implies, attempt to copy the actual shape of the waveform produced by the microphone and its associated analogue circuits. If the bandwidth is limited, the sampling theorem shows that it is theoretically possible to reconstruct the waveform exactly from a specification in terms of the amplitudes of regularly spaced ordinate samples taken at a frequency of at least twice the signal bandwidth. In its conceptually simplest form a waveform coder consists of a bandlimiting filter, a sampler and a device for coding the samples. The sampler operates at a rate higher than twice the cut-off frequency of the filter. The amplitudes of the samples are then represented as a digital code (normally binary) with enough digits to specify the signal ordinates sufficiently accurately. There is obviously no point in making the specification much more accurate than can be made use of for the given input signal-to-noise ratio. This principle of coding, known as **pulse code modulation (PCM),** was suggested by Reeves (1938), and is now widely used for feeding analogue signals into computers or other digital equipment for subsequent processing (in which case it is known as **analogue-to-digital (A-D)** conversion). The process is not normally used in its simplest form for transmission or for bulk storage of speech, because the required digit rate for acceptable quality is too high. Simple PCM does not exploit any of the special properties of speech production or auditory perception except their limited bandwidth.

The distortion caused by PCM can be considered as the addition of a signal representing the successive sample errors in the coding process. If the number of bits per sample in the code is fairly large (say>5) this **quantizing noise** has properties not obviously related to the structure of the speech, and its effect is then perceptually equivalent to adding a small amount of flat-spectrum random noise to the signal. If the number of digits in the binary code is small or if the input signal level exceeds the permitted coder range, the quantizing noise will have different properties and will be highly correlated with the speech signal. In this case the fidelity of reproduction of the speech waveform will obviously be much worse, but the degradation will no longer sound like the addition of random noise. It will be more similar perceptually to the result of non-linear distortion of the analogue signal. Such distortion produces many intermodulation products from the main spectral components of the speech signal, but even when extremely distorted the signal usually contains sufficient of the spectral features of the original signal for much of the intelligibility to be retained.

The sound pressure waveform of a speech signal has a substantial proportion of its total power (for some speakers more than half) in the frequency range below 300 Hz, even though the information content of the signal is almost entirely carried by the spectrum above 300 Hz. As quantizing noise has a flat spectrum its effect on the signal-to-noise ratio is much more serious for the weaker but more important higher-frequency components. A considerable performance improvement for PCM can be obtained by taking into account this property of speech production, and applying **pre-emphasis** to the speech signal with a simple linear filter to make the average spectrum more nearly flat. After PCM decoding the received signal can be restored to its original spectral shape by **de-emphasis,** so reducing the higher-frequency components of the quantizing noise. For normal communication purposes it is not, however, necessary that the de-emphasis should match the preemphasis, as speech intelligibility is actually improved by attenuating the low-frequency components, because it reduces the upward spread of auditory masking.

The amplitude of the quantizing noise of simple PCM is determined by the step size associated with a unit increment of the binary code. During low-level speech or silence this noise can be very noticeable, but in loud speech it is masked, partially or in some cases completely, by the wanted signal. For a given perceptual degradation in PCM it is therefore permissible to allow the quantizing noise to vary with signal level, so exploiting a property of perception. The variation can be achieved either by using a non-uniform distribution of quantizing levels or by making the quantizing step size change as the short-term average speech level varies. Both methods have been adopted, and have enabled excellent quantizing-noise performance to be achieved at 8 bits/sample, and useful communications performance at 4 bits/sample. Civil telephony uses PCM with 8 bits/sample at 8 kHz sampling rate, so needing 64 kbits/s. In this system there is an instantaneous **companding** characteristic that gives an approximately exponential distribution of quantizing intervals except at the lowest levels. (The two slightly different variants of this law used by different telephone administrations are known as **A-law** and **μ-law**.) The sampling rate is generous for the 300–3,400 Hz bandwidth required, but this high sampling rate simplifies the requirements for the band-limiting filters.

The time resolution properties of the auditory system ensure that masking of quantizing noise by the higher-level wanted signals is effective for at least a few milliseconds at a time, but instantaneous companding will give finer quantization near zero crossings even for large-amplitude signals. It is obvious that more effective use will be made of the transmitted digits if the step size is not determined by the instantaneous waveform ordinate height, but is changed in sympathy with the short-term average speech level. In this case, however, some means must be devised to transmit the extra information about the quantizing step size. This information can be sent as a small proportion of extra digits interleaved in the digital waveform description, but more usually it is embodied in the waveform code itself. The latter process is achieved by using a feedback loop that modifies the quantal step size slowly up or down according to whether the transmitted codes are near the extremities or near the centre of their permitted range. As the same codes are available at the receiver it is in principle easy to keep the receiver quantizing interval in step with that at the transmitter, but digital errors in the transmission path disturb this process and will thus affect the general signal level besides adding noise to the received audio waveform. Another disadvantage of this method of **backward adaptation** is that when the signal level increases suddenly it will overload the coder for at least a few samples before the quantizing interval has had time to adapt. Use of a separate channel for **forward adaptation** of the quantizing control can avoid this problem, but needs a small signal delay to enable the quantizer to be correctly set before the signal is coded, in addition to the small amount of extra information needed to specify the quantizer step size.

4.2.2 Deltamodulation

Deltamodulation is a very simple alternative type of waveform coding. A deltamodulator uses its transmitted digital codes to generate a local copy of the input waveform, and chooses successive digital codes so that the copy reproduces the input waveform as closely as possible, within the constraints of the coder. The basic scheme is illustrated in Figure 4.1. In its original and simplest form the quantizer uses only one bit per sample, and merely indicates whether the copy is to be increased or decreased by one quantum. Such a coder offers the possibility of extremely simple hardware implementation, and if run at a high enough sampling

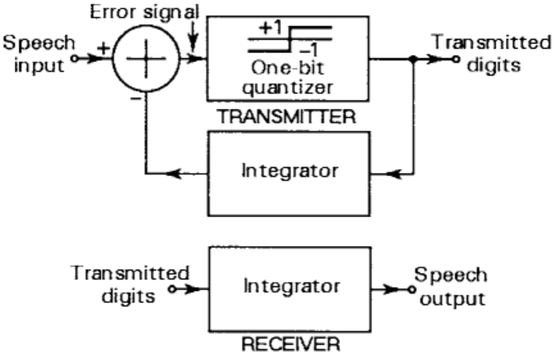


Figure 4.1. Block diagram of a simple deltamodulator.

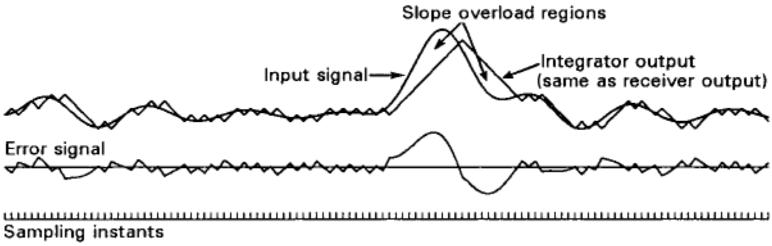


Figure 4.2. Waveforms in a simple deltamodulator.

rate can approximate waveforms very closely. The process of following the waveform in small steps makes deltamodulation work best on signals in which differences between successive ordinates are small. Thus the low-frequency dominance in speech signals is accommodated directly by deltamodulation without pre-emphasis, and it is acceptable to use a quantal step that is only a very small fraction of the waveform amplitude range. In contrast, a flat-spectrum input would cause frequent slope overloading if used with the same step size and sampling rate. Typical waveforms in simple deltamodulation are illustrated by Figure 4.2.

The use of a single bit per sample in deltamodulation is basically inefficient because a sampling rate much in excess of twice the highest frequency in the input signal is needed for close following of the input waveform. However, the intrinsic feedback loop in the coding process gives the coder some ‘memory’ of coding overload on previous waveform ordinates, for which it continues to compensate on later samples. This advantage of deltamodulation can be combined with those of PCM if a PCM coder is used instead of a one-bit quantizer in the feedback loop. Current terminology describes this arrangement as **differential PCM (DPCM)**.

The advantages of and techniques for level adaptation apply to deltamodulation in the same way as to PCM, and adaptive forms of coder are normally used, so exploiting the noise-masking properties of auditory perception and the slow level changes of speech production. **Adaptive DPCM (ADPCM)** incorporating an adaptive quantizer seems to be the most efficient of the simpler waveform coding processes. At 16 kbits/s the quantizing noise is noticeable, but slightly less objectionable than the noise given by adaptive deltamodulation or adaptive PCM at the same digit rate.

Many authors have also used the term ADPCM to describe waveform-following coders where the adaptation is based on much more complicated models of speech generation, with consequent much greater complexity than the simple coders described in this section. Coders of this more complicated type, but referred to as ADPCM, include a group of coders which have been recommended by the International Telecommunications Union (ITU) as standards for network telephony.

There are in fact a variety of waveform-following coders which incorporate adaptation but applied to a speech generation model of some complexity. It seems most useful, therefore, to describe these more elaborate systems in terms of the types of speech generation models they use and, in view of their higher complexity, they will be considered in the intermediate category (Section 4.4).

4.3 ANALYSIS/SYNTHESIS SYSTEMS (VOCODERS)

An alternative to direct waveform coding is to analyse the speech signal in terms of parameters describing its perceptually important characteristics. These parameters are transmitted and used to generate a new waveform at the receiver. The regenerated waveform will not necessarily resemble the original waveform in appearance, but should be perceptually similar. This type of coding system was first described by Homer Dudley of Bell Telephone Laboratories (Dudley, 1939), who called his system a **vocoder** (a contraction of VOice CODER). The term vocoder has since been widely used to refer to analysis/synthesis coders in general.

Most vocoders are based on a model of speech production which exploits the fact that it is possible substantially to separate the operations of sound generation and subsequent spectrum shaping. The basic elements of such a vocoder are shown in Figure 4.3. The sources of sound are modelled by periodic or random excitation, and in several of the more recent vocoders it is also possible to have mixtures of both types of excitation. The excitation is used as input to a dynamically controllable filter system. The filter system models the combined effects of the spectral trend of the original sound source and the frequency response of the vocal tract. The specifications for the sound sources and for the spectral envelope are both derived by analysis of the input speech. By separating the fine structure specification of the sound sources from the overall spectral envelope description, and identifying both in terms of a fairly small number of slowly varying parameters, it is possible to produce a reasonable description of the speech at data rates of 1,000–3,000 bits/s. The general principles of synthesis used in the receiver to regenerate speech from this description were discussed in Section 2.7.

There are many different types of coder that use analysis/synthesis, but our discussion will concentrate on four of the most influential ones. These are **channel vocoders, sinusoidal coders, linear predictive coding (LPC) vocoders** and **formant vocoders**. With all these types the data are coded into **frames** representing speech spectra measured at intervals of 10–30 ms. There are also techniques for efficient coding of frames and sequences of frames, and some specialized vocoders which code whole sequences of frames as single units. Discussion of these techniques will be postponed until Sections 4.3.5 and 4.3.6.

Of the four types of vocoder mentioned above, up until around the late 1980s LPC and channel vocoders predominated. The advantages and disadvantages of the two types were nearly equally balanced; both gave usable but rather poor speech

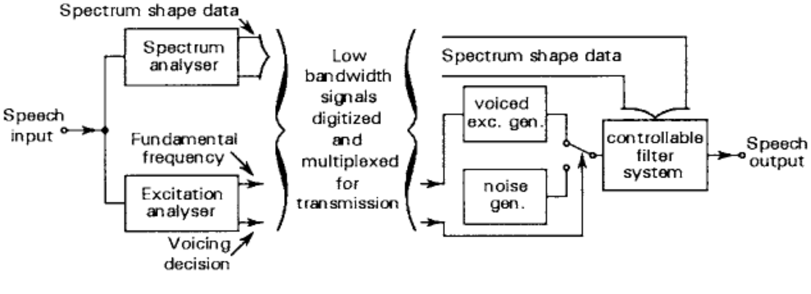


Figure 4.3 Block diagram of the basic elements of a vocoder.

transmission performance at 2,400 bits/s using fairly complex processing. The advent of digital signal processor (DSP) chips made implementation very easy, and standard LPC is somewhat simpler than a channel vocoder when these devices are used. More recently, various enhancements have been added to LPC vocoders and sinusoidal coders have emerged to take over from channel vocoders as an alternative at comparable data rates. Formant vocoders have a long history, but there are special advantages and problems with their use in practical systems and their implementation is much more complex, as will be explained in Section 4.3.4.

4.3.1 Channel vocoders

In a channel vocoder (of which Dudley's was the first example) the spectrum is represented by the response of a bank of contiguous variable-gain bandpass filters. The way in which the desired overall response can be approximated using the separate contributions from individual channels is shown in Figure 4.4. The control signals for the channels are derived by measuring the short-term-average power from a similar set of filters fed with the input speech signal in the transmitter.

Unless a very large number of channels can be used (with consequent high digit rate) it is difficult to achieve a good match to the spectrum shapes around the formant peaks with a channel vocoder. However, the quality achievable with around 15–20 channels is reasonable for communications purposes, and has been achieved in several systems operating at data rates of around 2,400 bits/s or lower.

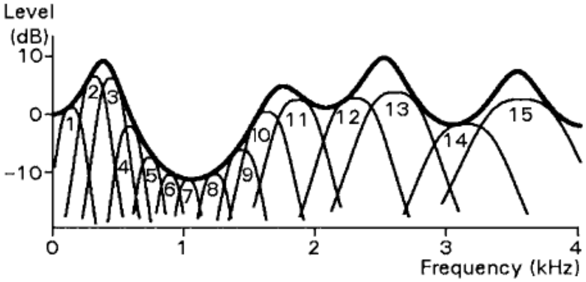


Figure 4.4. Contributions of individual channels to a channel vocoder output spectrum. Thick line: desired spectrum shape. Thin lines: contributions from the separate channels.

4.3.2 Sinusoidal coders

The key feature of sinusoidal analysis/synthesis models is the concept of representing the short-term spectrum of a speech signal as a sum of sinusoids specified in terms of frequency, amplitude and phase. One such coding method is known as **sinusoidal transform coding (STC)**. For each frame, a set of frequencies, amplitudes and phases is estimated corresponding to peaks in the short-term Fourier transform. An economical code is achieved by representing voiced speech as a constrained set of harmonically related sinusoids, and unvoiced speech as a set of sinusoids with appropriately defined random phases.

Multi-band excitation (MBE) coding is another method that uses a sinusoidal model. The main features of the MBE approach that distinguish it from STC are the treatment of the excitation and the method used to generate unvoiced components of the speech. The fundamental frequency is estimated, then the spectrum is divided into harmonic bands and a binary voiced/unvoiced decision is made separately for each band. The voiced components of the speech are regenerated as a combination of the relevant set of harmonic sinusoids, with the amplitudes of all sinusoids associated with an 'unvoiced' harmonic being set to zero. The unvoiced components are generated separately using a frequency-domain method. In this method, the spectrum shape taken from the unvoiced samples of the estimated vocal tract spectral envelope is multiplied by the spectrum of a whitenoise excitation signal. The resulting transform is used to synthesize the unvoiced speech signal components, which are added to the voiced components to produce the final synthesized speech waveform.

Efficient coding techniques have been developed both for STC and for MBE coding, and have enabled these methods to achieve considerable success at data rates in the 2,000–4,000 bits/s range, providing more natural-sounding speech than traditional channel vocoders at these data rates.

4.3.3 LPC vocoders

In LPC vocoders, spectral approximation to a speech signal is given by the response of a sampled-data filter, whose all-pole transfer function is chosen to give a least-squared error in waveform prediction. The configuration of such a filter is illustrated in Figure 4.5.

The principle of linear prediction applied to a resonant system depends on the fact that resonance causes the future output of a system to depend on its previous history, because resonant modes continue to 'ring' after their excitation has ceased. The characteristics of this ringing are determined by a linear difference equation of appropriate order. If a system is ringing entirely as a result of previous excitation, and can be represented exactly by a small number of resonant modes with constant characteristics, a difference equation can be derived to predict its future output exactly. In speech, however, these assumptions are only approximately true. Although the formants cause a strong resonance effect, they vary slowly with time. Their damping is also modified by opening and closing of the glottis. The resonances are always being excited to some extent by the sound sources, and receive substantial excitation at the instants of glottal closure. In spite of all these deviations from the ideal, it is possible to derive a useful description of the spectrum by choosing the parameters of a predictor filter to minimize the average prediction error power over a frame of input samples of around 10–20 ms duration. The predictor filter is a finite-impulse-response filter whose output is a weighted linear combination of previous input speech samples. At the receiver the predictor is connected in a feedback loop to give an all-pole recursive filter, whose resonant modes approximate those for the input signal. The two main methods that are used for deriving the predictor filter coefficients are known as the **covariance** and **autocorrelation** methods, but their details are outside the scope of this book.

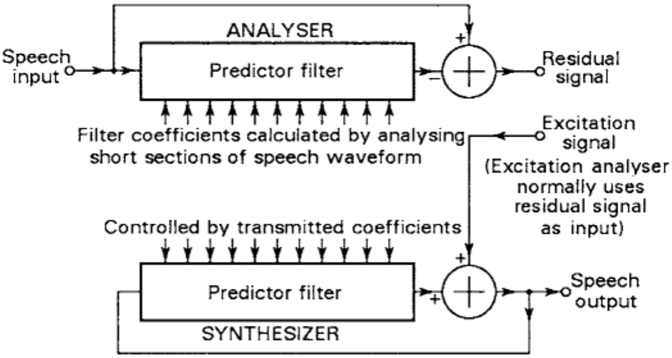

ANALYSER
Speech input
Predictor filter
+
−
Residual signal
Filter coefficients calculated by analysing short sections of speech waveform
Excitation signal
(Excitation analyser normally uses residual signal as input)
Controlled by transmitted coefficients
+
Predictor filter
+
Speech output
SYNTHESIZER

Figure 4.5. Predictor filter as used in LPC systems. When connected as shown in the synthesizer, the finite-impulse-response predictor filter provides the feedback path in an allpole recursive filter. Because the formant resonances are effectively removed, the residual signal is a more suitable input for excitation analysis than the original speech waveform.

In any practical LPC-based coding system, the parameters describing the predictor filter must be quantized prior to transmission to the decoder. The filter coefficients are not quantized directly, because small quantization errors can give rise to large changes in the spectral response of the filter. The coefficients are therefore converted to some alternative representation, which is chosen to be more robust under quantization. For recent implementations, including the LPC-derived intermediate-rate coders described later, the most popular choice is a parameter set called **line spectrum pairs,** details of which may be found in the literature.

When the predictor filter has been adjusted to predict the input as best it can from the immediately preceding samples, the difference between the input speech and the predictor output (known as the **residual**) will have a roughly flat spectrum. The obvious spectral peaks caused by the resonances of speech production will have been removed. For this reason the complete filtering process is sometimes referred to as **inverse filtering**.

In LPC vocoders the resonant properties of the synthesis filter make possible fairly good approximations to the spectral shapes of the formants. However, the correct analysis to achieve this result will only be obtained when the overall speech spectrum really is like the response of an all-pole filter. During vowel sounds this approximation is often very close, although at normal frame rates standard LPC cannot deal correctly with the fact that the formant bandwidths in natural speech change significantly as the glottis opens and closes in each excitation period. There are frequent other occasions when the spectral modelling is quite poor, particularly during nasalized vowels and many voiced consonant sounds. On these occasions the LPC synthesis frequently produces spectral peaks whose bandwidths are too large, with a consequent ‘buzziness’ in the speech quality. Another inherent property of normal LPC vocoders is that all regions of the spectrum are treated equally with regard to accuracy of frequency specifications, and so no advantage is taken of the variable frequency resolution of auditory perception.

Simple LPC coding produces speech which is buzzy but intelligible, and a version of the algorithm was used for many years as a U.S. Government standard at 2,400 bits/

s for secure voice communications in both military and civilian use. Recently, this standard has been replaced by a new standard at the same data rate but using an enhanced LPC algorithm that gives speech of considerably better quality. This algorithm, known as **mixed excitation linear prediction (MELP),** uses a frequency-dependent mixture of pulse and noise excitation (similar in concept to that described in Section 2.7.2 for a parallel-formant synthesizer). Other new features of the MELP coder include a spectral enhancement filter to improve the match to the natural speech in regions close to the formant frequencies, and a dispersion filter applied after the LPC synthesis filter in order to improve the modelling of regions between the formants. The new developments have removed a lot of the buzziness traditionally associated with LPC-coded speech.

4.3.4 Formant vocoders

In formant vocoders the spectrum shape is specified in terms of the frequencies and amplitudes associated with the resonant modes of the speaker's vocal tract. The relationship between the formant control signals and the synthesized spectral shape can be seen from Figure 4.6.

Formant vocoders are different from the other types described above, as they use a synthesizer that is much more closely related to human speech production. In addition to modelling periodic and noise sources, the synthesizer has a spectral filter system with resonators that are explicitly related to the principal formants of the input speech. Thus the coding system can be constrained to deal only with the known frequency range and necessary accuracy of specification for each formant. The systematic variation of formant bandwidth with glottal opening can easily be provided in a formant synthesizer and requires no extra information to be transmitted. Apart from this effect, the bandwidths of the formants do not vary much and such variation as does occur is fairly predictable; provided they are within the limits of natural variation, preservation of the actual formant bandwidths is not perceptually important. In consequence this property of the resonances is not usually transmitted in formant vocoders. For formant vocoders it is not practicable to use a simple cascade connection of resonators (Figure 2.13), because occasional

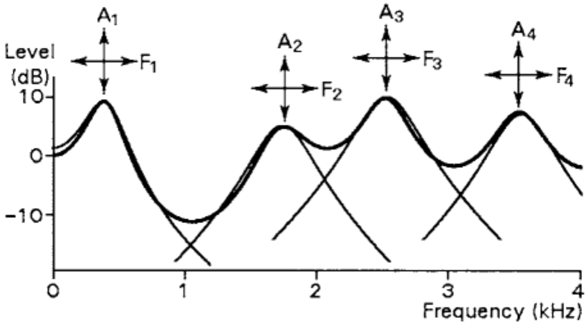


Figure 4.6. Illustration of how formant amplitude and frequency control signals affect the output spectrum in a formant vocoder. Thick line: desired spectrum. Thin lines: contributions from individual formants.

errors of formant frequency could then cause serious formant amplitude errors. The parallel connection, such as is shown in Figure 2.15, is therefore necessary.

Analysis is the main difficulty with formant vocoders. When the spectral envelope of a speech sound shows a small number of well-defined peaks it is trivial to assign these to formants in a sensible way. However, there are occasions, particularly in consonants, near vowel/consonant boundaries or even in the middle of vowels if the fundamental frequency is very high, when it is not clear what is the most appropriate way to assign the parameters of the synthesizer to spectral peaks of the input signal. Because of the analysis difficulties formant vocoders are not yet used operationally, but a few have been demonstrated in a research environment, and some of these have been extremely successful, though computationally expensive. Formant analysers have been used to derive stored components for message synthesis, because the analysis can then be carried out more slowly than real time on only a moderate amount of speech material, and serious analysis errors can be corrected by subsequent interactive editing. This possibility is, of course, not available for coding real-time conversation.

4.3.5 Efficient parameter coding

For all types of analysis/synthesis systems it is possible to achieve some saving in digit rate by exploiting the redundancy in the measured parameters. Any technique of this type will add complexity, but the analysis process itself gives such a reduction of digit rate compared with the original speech that the computational speed needed for further processing can be quite low. With modern implementation technology fairly complex coding is possible even in single-chip microprocessors.

Vector quantization (**VQ**) considers only one frame at a time and exploits the fact that the multi-dimensional parameter space is not uniformly occupied. By choosing from a subset of possible combinations of parameter values (which are stored in a **codebook**), fewer bits are required per frame than are needed for independent coding of the parameters. However, quite a lot of computation is needed to select the most appropriate member of the codebook for each frame.

The data rate can also be reduced by taking into account the relationship between the data in a sequence of frames, although it will always be necessary to provide buffer delay for this reduction to be exploited in a constant-rate real-time link. A simple method, often referred to as **frame fill,** involves only transmitting every alternate frame and sending an additional code of just 1 or 2 bits to indicate how to reconstruct the missing frames (typically by repeating one of the frames either side or by interpolating between them). There are also more elaborate schemes for transmission at a **variable frame rate**. In these methods, a frame is only transmitted when the spectrum change since the preceding transmitted frame exceeds some threshold and missing frames are obtained by interpolation. Thus frames are transmitted more frequently when the speech characteristics are changing rapidly, and less frequently when the characteristics are changing only slowly. However, in order to obtain the full benefit of the method, a longer delay must be included than with fixed frame-fill methods. Variable-frame-rate schemes are especially suited to formant vocoders because the control parameters tend to vary in a much more orderly way than they do in other types of vocoders.

Reduced-frame-rate schemes can achieve very good speech quality for realtime systems at about 1,200 bits/s. Lower rates are possible for coding stored message components, because the delay due to variable-rate coding is acceptable and interactive editing can overcome some of the limitations of automatic coding.

4.3.6 Vocoders based on segmental/phonetic structure

Even lower bit rates can be achieved by coding whole sequences of frames as single units. In **segment vocoders,** consistent segments of speech are identified and an extended form of VQ is used to provide a compact codebook that contains variable-length segments. An utterance can then be coded by finding the sequence of these segments that provides the best fit according to some suitable distance criterion. Segment vocoders have been produced with reasonable intelligibility at rates of less than 300 bits/s. However, they work best when used to code speech from a known speaker, so that a speaker-specific codebook can be used. If coding of speech from any (unknown) speaker is required, it becomes difficult to retain speaker characteristics while keeping the codebook at a manageable size.

The lowest bit rates are possible by explicitly taking advantage of the phonetic structure of spoken language and using a phoneme-based coding scheme. For example, given that English contains about 44 phonemes (which can be generously specified using 6 bits) and a typical speaking rate might be an average of around 12 phonemes/s, the phoneme sequence can be coded at a rate of approximately 70 bits/s. Including some additional bits for pitch and timing information may only increase the bit rate to not much more than 100 bits/s. The term **phonetic vocoder** is generally used to describe a type of segment vocoder in which the segments are explicitly defined in phonetic terms. These coders involve applying automatic speech recognition at the transmitter and using some synthesis technique at the receiver, and are therefore sometimes referred to as **recognition-synthesis coders**. The coding performance is obviously critically dependent both on the quality of the synthesizer, and especially on the recognition performance as any recognition errors will lead to coding errors. As with other segment vocoders, it is difficult for phonetic vocoders to retain speaker characteristics if they are to be used to code speech from any arbitrary speaker.

Segment and phonetic vocoders typically use complex processing to produce speech of limited quality, and they operate at a variable frame rate, so some delay is unavoidable. They do, however, make possible speech transmission at very low data rates, and have so far been of most interest for their potential use in specialized military communications where there may be severe bandwidth restrictions but the task can be quite tightly controlled and a delay is tolerable.

4.4 INTERMEDIATE SYSTEMS

There are many ways of combining some of the detailed signal description possibilities of simple waveform coders with some of the signal redundancy exploitation of vocoders. The resultant intermediate systems normally give much better speech reproduction in the 4–16 kbits/s range than is possible with either of the other two classes of system at these digit rates. Their complexity is, of course, always greater than for simple waveform coders, and most of the higher-performance systems are more complicated than the majority of vocoders.

The importance of these systems has increased dramatically over the past 10 years because of the explosive growth in mobile telephony of various sorts during this period. The restrictions of radio bandwidth, and the need for extra bits for the detection and correction of the inevitable digital transmission errors, has made it important to keep the digit rate for speech coding to much less than the 64 kbits/s or 32 kbits/s used in line transmission. But the requirement for acceptable speech quality from the general population means that the performance currently obtainable from vocoders would not be adequate for these applications. Although mobile telephones must be small and not too expensive, it is fortunate that modern integrated circuit technology is making it possible to implement many very complicated coding algorithms while still satisfying the cost and size constraints.

4.4.1 Sub-band coding

Waveform coding can make use of masking effects and the ear's tolerance to less accurate signal specification at higher frequencies by filtering the speech signal into many bands and coding each band separately (Figure 4.7). Systems that use this technique are known as **sub-band coders**. Each sub-band is coded by a waveform coding process, using a sampling rate equal to twice the bandwidth in each case. If the bands are made as narrow as the ear's critical bands, the quantizing noise in each band can be largely masked by the speech signal in the same band. In addition, the fact that most power in the higher bands is from the 'unvoiced' randomly excited sounds means that the waveform shape in these bands need not be specified so accurately, and therefore requires fewer bits per sample. In practice, complexity of suitable filter designs makes a choice of about five bands more attractive than the 15–20 needed for critical bands, and even with this number the speech quality in the 16–32 kbits/s range is much improved over that possible

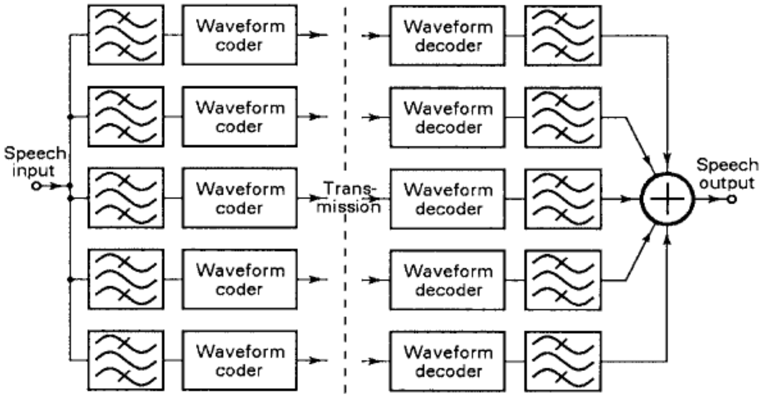


Figure 4.7. Block diagram of a sub-band coder.

with the best simple waveform coders at the same bit rates. Because they do not exploit the constraints of speech production, sub-band coders have mainly been used where non-speech signals must also be transmitted.

4.4.2 Linear prediction with simple coding of the residual

The technique of linear prediction analysis for vocoders inherently makes available in the analyser the low-power prediction error signal, the **residual.** If applied as input to the LPC vocoder synthesis filter, this residual signal will regenerate the original speech waveform exactly. This fact can be exploited to give an improvement in speech quality by transmitting to the receiver a digitally coded representation of the residual to replace the conventional vocoder excitation. A further stage of prediction can be included to predict the periodicity at the fundamental frequency of excitation for voiced sounds, and the residual will then be of even lower power and will be much more random in structure. Although the residual must inherently contain important detail in its time structure, its gross spectral shape will always be fairly flat because it is the output of the LPC ‘inverse filter’. The objective in coding the residual is to retain as much as possible of its perceptually important features, because the quality of the output speech will depend on the extent to which these features are available at the receiver.

There is a whole family of related techniques that all depend on linear prediction analysis, with some form of excitation derived from the residual. At one extreme is, of course, the LPC vocoder, where the excitation is generated taking into account only the periodicity of the residual, which is transmitted as a low-data-rate parameter. Another technique is to transmit the low-frequency waveform structure of the residual, sampled at a reduced rate, and to regenerate a wide-band excitation by non-linear action or by spectral folding. This method has usually been called **residual-excited linear prediction (RELP)**. It exploits the fact that the detail of the spectrum structure is perceptually not so important at the higher audio frequencies. Other simple residual coding systems have also been developed, but are no longer of interest because of more effective methods described below.

4.4.3 Adaptive predictive coding

There is another group of techniques depending on linear prediction analysis, which are essentially waveform-following coders, somewhat similar in principle to ADPCM (see Section 4.2.2). However, as mentioned in Section 4.2.2, it seems more informative to name them to include their speech modelling method in the title, and the early research papers on these systems in general used such names. One of the earliest systems of this type (developed in the 1970s and early 1980s) was referred to as **adaptive predictive coding (APC)**. The early versions used a comparatively simple coding scheme for the residual, and lacked many of the refinements introduced over the next ten years. Now systems of this general class use a much more complicated structure than these early APC systems, and they normally include most or all of the features illustrated in Figure 4.8.

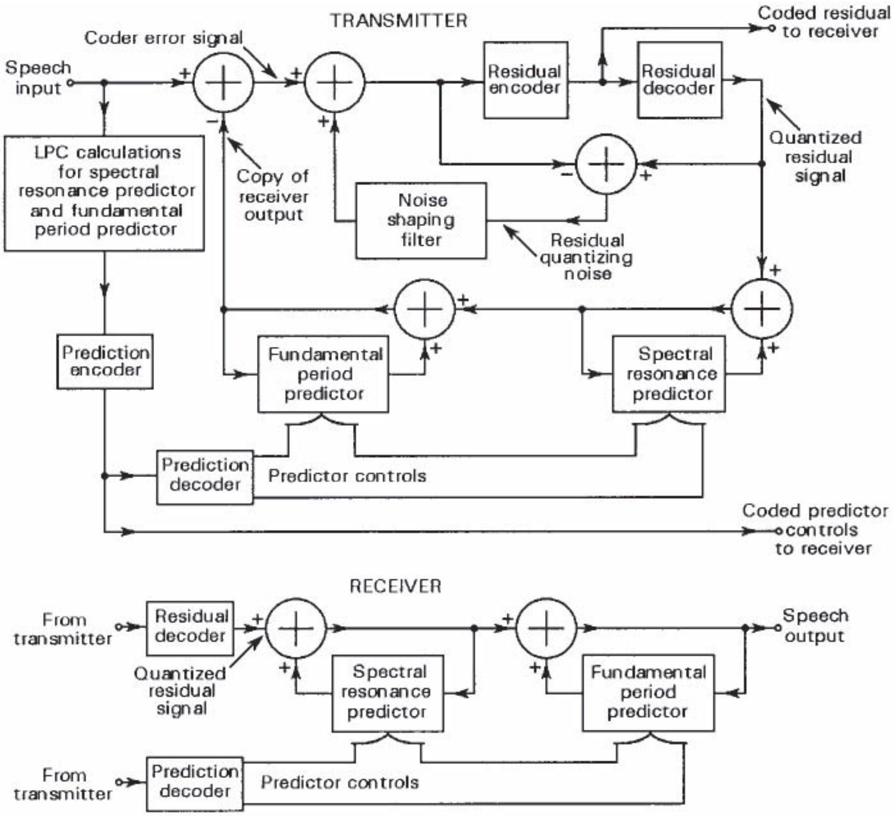

TRANSMITTER
Coded residual to receiver
Coder error signal
Speech input
+
+
−
+
Residual encoder
Residual decoder
LPC calculations for spectral resonance predictor and fundamental period predictor
Copy of receiver output
Quantized residual signal
−
+
Noise shaping filter
Residual quantizing noise
+
+
+
+
Prediction encoder
Fundamental period predictor
Spectral resonance predictor
Prediction decoder
Predictor controls
Coded predictor controls to receiver
RECEIVER
From transmitter
Residual decoder
+
+
Speech output
Quantized residual signal
+
Spectral resonance predictor
+
Fundamental period predictor
From transmitter
Prediction decoder
Predictor controls

Figure 4.8. General APC-type coder and decoder with two predictors in the feedback loop and spectral shaping of the quantizing noise.

The important features of this group are:

1. They are essentially waveform coders, in that they have a feedback loop that attempts to copy the input waveform, using a suitable error criterion.
2. They normally include a predictor for both the spectral resonances and the fundamental period of voiced speech, and parameters for these two predictors are coded and sent as **side information**.
3. The predictors in the feedback loop are controlled by the quantized predictor coefficients. The quantizer used to code the residual is also placed in the feedback loop. As a result, the waveform of the residual takes into account the imperfections of both quantizers. The overall coding accuracy can then be much better than would be possible by using the same number of bits to code the residual in isolation.
4. The minimization of the overall quantizing noise is achieved through a **noiseshaping filter** that gives less weight to the noise in high-level regions of the spectrum. The action of the noise-shaping filter causes an increase in the total noise power, but the noise is concentrated in regions of the spectrum where the speech power is high and

> hence the noise is more effectively masked by the speech signal. Comparatively more weight is given to the error signal in low-level regions between the formants, so giving a better signal-to-noise ratio in these regions than would otherwise be achieved. A suitable noise-shaping characteristic can easily be derived by adjusting the response of the resonance predictor to give slightly reduced peak heights at the formant frequencies.

If the fundamental-period predictor is included the residual power will be very low during long periodic sounds, but there will still be higher-level transients whenever either the fundamental frequency or the formant frequencies change. As the ear is in general less sensitive to noise and distortion in transient sounds, the presence of this extra stage of prediction is just what is wanted to make the coding more accurate when it matters most. It is therefore always worth putting in the fundamental predictor if the extra complexity is acceptable.

4.4.4 Multipulse LPC

There are many techniques for coding the residual in the APC family of systems, and some have attained such importance that they have been given their own separate names. One of these is **multipulse LPC,** where the residual is represented by a much smaller number of pulses than is indicated by the Nyquist rate for the signal. Analysis by synthesis is used to optimize the amplitudes and positions of these pulses to minimize the spectrally shaped quantizing noise, and it is these amplitudes and positions that are coded for transmission. Typically only 10–20 pulses are used for every 20 ms of speech, so reducing the sampling rate of the residual by a factor of more than 10. The success of multipulse LPC is fairly easy to understand, and stems directly from the fact that voiced speech is typically excited by only a small number of glottal pulses every 20 ms. The multipulse algorithm will place the largest excitation pulses to correspond with the main glottal excitation points, and will then add extra smaller pulses to correct for the inadequacy of the LPC filter in predicting the waveform detail. The inadequacy arises from the fact that neither the excitation nor the vocal tract response can be accurately represented by the simple models that are assumed in LPC vocoders.

4.4.5 Code-excited linear prediction

Another variant of the APC family is **code-excited linear prediction (CELP)**. The principle is to use analysis by synthesis to select the best excitation from one or more codebooks. The original version of CELP used a codebook populated with random noise sequences, but more recent designs usually employ deterministic or structured codebooks that are chosen to permit faster codebook search techniques.

A typical simple implementation of the basic CELP algorithm uses a set of 1024 waveform code sequences, selected by a 10-bit code. Each sequence corresponds to a waveform section of around 40 samples (5 ms at 8 kHz sampling rate), thus involving only one quarter of a bit per sample of coded residual. For each 40-sample section of residual, the available codes are tested to choose the one that minimizes the weighted quantizing noise power, just as in multipulse LPC.

In the period since CELP coding was first proposed in the mid-1980s many refinements and different variants of the original algorithm have been developed. However, most of the current variants of CELP include the following elements:

1. *A fixed excitation codebook.* One example is the random-noise codebook used in the original version of CELP. A widespread implementation uses linear combinations of the vectors from two small codebook tables. Another implementation uses a sparse codebook with only a few non-zero pulses, similar to the method used in the multipulse LPC described in Section 4.4.4.
2. *An adaptive codebook.* This 'codebook' is arranged to operate as a fundamental period (pitch) predictor (as mentioned in Section 4.4.3 describing APC). The codebook contains the previous excitation used, and this excitation is tested at a range of delay values (the codebook is therefore regarded as *adaptive,* as it changes depending on the speech). The codebook is searched to find the optimal delay (and hence the pitch period).
3. *An error-weighting filter.* This filter has the same objective as the noise-shaping filter used in APC, allowing the coder to choose excitation which will not minimize the squared error, but instead gives rise to a perceptually better spectral distribution of the quantization noise. This distribution of the noise takes into account the speech spectrum shape, so that the noise is concentrated in those spectral regions that have high energy. The filter may be adaptive, with different weightings for voiced and unvoiced speech.
4. *An adaptive post-filter.* This filter is appended to the decoder, but does not require any additional transmitted parameters. Individual implementations vary, but the general aim is to reimpose speech-like characteristics that may have been lost in the coding process. For example, a voiced-speech detector may be applied followed by a comb filter to enhance the pitch harmonics. Other characteristics which may be modified include spectral tilt and bandwidths of spectral peaks. Any post-filter may need to be disabled for non-speech signals such as music or DTMF (touch-tone) signals.

In comparison with the other intermediate-rate systems described in this section, CELP coders usually offer better performance at the cost of greater computational complexity. Improvements in the technology of signal processing chips, and the general increase in available computational power, have made complexity less important than before, and CELP coders now dominate in the field of mobile telephony and in related application areas.

4.5 EVALUATING SPEECH CODING ALGORITHMS

In order to make meaningful comparisons between the performance of different speech coding algorithms, it is important to have a means of formally evaluating the ability to preserve the characteristics of the original speech. The ultimate criterion for judging the performance of a speech transmission or reproduction system must be the satisfaction of the human users, which can only be assessed by **subjective** testing. Subjective test measures are based on listeners' responses to questions about the speech, and may be subdivided into those that measure **intelligibility** and those that are intended to gauge

perceived quality (including attributes such as naturalness and recognizability of the speaker). Subjective tests are essential for assessing the reactions of listeners, but it can be helpful to also use **objective** test measures to provide a mathematical comparison between the original and the coded speech signals. Objective tests are especially valuable when there are only small differences in quality to be assessed, and have the advantage of being easier and less time-consuming to carry out than tests involving human listeners. There are a variety of both subjective and objective measures available, and a few examples of the different measures are described briefly below.

4.5.1 Subjective speech intelligibility measures

Intelligibility tests are often based on listeners' responses to single-syllable rhyming words of the form consonant-vowel-consonant (e.g. "bat", "cat", etc.). One widely used test is the **diagnostic rhyme test (DRT),** whereby a listener hears a succession of test stimuli and, for each stimulus, selects from a choice of just two words. The members of each word pair differ only in the initial consonant and this difference is further restricted to be in only one **distinctive feature** (such as voicing or nasality). Example pairs are "goat-coat" (which differ in the voicing feature) and "moss-boss" (which differ in the nasality feature). The overall DRT score is obtained as follows:

$$\text{DRT score (\%)} = \frac{N_{\text{correct}} - N_{\text{incorrect}}}{N_{\text{tests}}} \times 100\,, \tag{4.1}$$

where N_{tests} is the number of tests, N_{correct} is the number of correct responses, and $N_{\text{incorrect}}$ is the number of incorrect responses. A system that produces speech of 'good' quality would typically have a DRT score in the region of 85–90%. An advantage of the DRT is that the results can be analysed to determine how well different phonetic distinctions are preserved in a speech coding system.

4.5.2 Subjective speech quality measures

Speech 'quality' is more difficult to quantify, but can be assessed using an opinion rating method such as the **mean opinion score (MOS)**. With this method, listeners rate the quality of the speech under test on a five-point scale ranging from 1 (unsatisfactory) to 5 (excellent). Care must be taken when conducting and interpreting the results of these tests, as listeners can vary greatly in their interpretation of the subjective scale, and any one listener may not be consistent across evaluation sessions. Using set reference signals as part of each evaluation session can help to normalize for these types of variation.

4.5.3 Objective speech quality measures

A widely used objective measure of speech quality is the **signal-to-noise ratio (SNR)**. The SNR for a speech coder is the ratio of the average energy in the original speech waveform to the average energy in the error (or 'noise') signal representing the distortion introduced by the coding algorithm (see Section 4.2.1). If *s(n)* represents the original speech signal at time *n* and $\hat{s}(n)$ is the corresponding coded signal, the error signal *e(n)* can be written as

$$e(n)=s(n)-\hat{s}(n). \tag{4.2}$$

The SNR is usually given in decibels, thus:

$$\mathrm{SNR(dB)} = 10\log_{10}\frac{E(s)}{E(e)} = 10\log_{10}\frac{\sum\limits_{n} s^2(n)}{\sum\limits_{n}\left[s(n)-\hat{s}(n)\right]^2}, \tag{4.3}$$

where *E(s)* and *E(e)* denote the energy in the speech and error signals respectively.

The criterion for success of an objective quality measure is that it should be a good predictor of subjective speech quality. Classical SNR is an energy ratio which is computed over an entire signal and so makes no distinction between errors that occur in high-energy regions and those in the low-energy regions, where any errors will have a greater perceptual effect. An improved measure is the **segmental SNR,** whereby the SNR is measured over short intervals (typically 15–25 ms) and the individual SNR measures are averaged. There are also several other SNR variants, all of which are aimed at improving the estimate of perceptual speech quality.

Because they are based on a waveform comparison, SNR measures are only appropriate for coders that are intended to reproduce the original input waveform. For coders which aim to reproduce the perceptually important features of a speech signal without necessarily copying the waveform detail, some form of spectral comparison measure can be used. Various objective measures have been proposed which attempt to be influenced only by perceptually relevant differences between the spectral characteristics of the coded and original signals. One such measure is the **perceptual speech-quality measure (PSQM),** which has been standardized by the ITU as recommendation P.861. This objective measure has been shown to predict subjective speech quality quite well for a limited range of speech coders and test circumstances. However, objective quality measures are still an active research area and care is required to assess their validity and applicability to any particular case.

4.6 CHOOSING A CODER

It can be seen that there is a bewildering variety of speech coding methods available, each with its own particular advantages and disadvantages, and it is very difficult for a system designer to make the best compromise between the conflicting factors which should influence the choice. Characteristics such as cost, size, weight and digit rate can be assessed using mainly engineering and economic criteria. The communications delay caused by the coder must also be considered, both from an engineering perspective and in terms of usability. These assessments need to be balanced against an evaluation of the quality of the coded speech, which is not straightforward but which should include properly controlled subjective testing to ensure making the correct choice of speech coder.

CHAPTER 4 SUMMARY

- Speech coding always involves a three-way compromise between data rate, speech quality and algorithm complexity.
- Simple waveform coders, using pulse code modulation or deltamodulation, can achieve fairly good quality with very simple equipment, but require a high data rate. Adaptation of the quantizer in these coders improves the performance at any data rate with only a small increase in complexity.
- Analysis/synthesis systems (‘vocoders’) provide much lower data rates by using some functional model of the human speaking mechanism at the receiver. The excitation properties and spectral envelope are usually specified separately. Different types of vocoder describe the slowly varying spectral envelope in different ways. Channel vocoders specify the power in a set of contiguous fixed band-pass filters, and sinusoidal coders specify frequencies, amplitudes and phases of sinusoids. LPC vocoders use an all-pole sampled-data filter to model the short-term speech spectrum. Formant vocoders specify the frequencies and intensities of the lowest-frequency formants.
- Currently the most successful coders for real-time speech communication at 2,400 bits/s use sinusoidal coding or mixed-excitation linear prediction.
- Intermediate systems have some of the advantages both of vocoders and of simple waveform coders, and often use digit rates in the 4–16 kbits/s range.
- Many intermediate systems use linear prediction analysis to exploit the resonant properties of speech production, but with different ways of coding the prediction residual for use as excitation in the receiver. Adaptive predictive coding, multipulse linear prediction and code-excited linear prediction can all give excellent speech quality at data rates well below 16 kbits/s.
- Very low data rates of a few hundred bits/s can be achieved by coding whole sequences of frames as single units using segment or phonetic vocoders, but at the expense of complex processing and often quite poor speech quality.
- Ideally speech coders need to be evaluated by subjective tests of both quality and naturalness, but objective comparison measures can also be useful.

CHAPTER 4 EXERCISES

E4.1 Why are simple speech waveform coders extravagant with digit rate, and why should economies be possible?

E4.2 Why is it not useful to specify the ratio of signal to quantizing noise as a performance criterion for PCM when there are very few bits per sample?

E4.3 Why are simple waveform coders greatly improved by some form of adaptation to variations in speech level?

E4.4 What are the essential features of a vocoder?

E4.5 Why do the theoretically attractive features of LPC vocoders not necessarily result in improved performance over channel vocoders or sinusoidal coders?

E4.6 Describe possibilities for reducing the transmission data rate in vocoders.

E4.7 Discuss the role of linear prediction in intermediate-rate coding techniques.

CHAPTER 5

Message Synthesis from Stored Human Speech Components

5.1 INTRODUCTION

Several years ago the term “speech synthesis” was used almost exclusively for the process of generating speech sounds completely artificially in a machine which to some extent modelled the human speaking system, as described in Chapter 2. The applications were mainly for research in speech production and perception. These days, particularly in an engineering environment, speech synthesis has come to mean provision of information in the form of speech from a machine, in which the messages are structured dynamically to suit the particular circumstances required. The applications include information services, reading machines for the blind and communication aids for people with speech disorders. Speech synthesis can also be an important part of complicated man-machine systems, in which various types of structured dialogue can be made using voice output, with either automatic speech recognition or key pressing for the human-to-machine direction of communication.

A conceptually simple approach to message synthesis is to concatenate fragments of human speech for the message components. This chapter describes a variety of **concatenative synthesis** techniques, categorizing them according to the size of the units to be concatenated and the type of signal representation used. These synthesis techniques can be used for preparing limited sets of known messages, but they are also frequently used as the speech-generation component of more general systems for speech synthesis from unrestricted text (see Chapter 7).

5.2 CONCATENATION OF WHOLE WORDS

5.2.1 Simple waveform concatenation

An obvious way of producing speech messages by machine is to have recordings of a human being speaking all the various words, and to replay the recordings at the required times to compose the messages. The first significant application of this technique was a speaking clock, introduced into the UK telephone system in 1936, and now provided by telephone administrations all over the world. The original UK Speaking Clock used optical recording on glass discs for the various phrases, words and part-words that were required to make up the full range of time announcements (see Figure 5.1). Some words can be split into parts for this application, because, for example, the same recording can be used for the second syllables of “twenty”, “thirty”, etc. The next generation of equipment used analogue storage on magnetic drums. For general applications of voice output there is a serious disadvantage with analogue storage on tapes, discs or drums: the words can only start when the recording medium is in the right position, so messages need

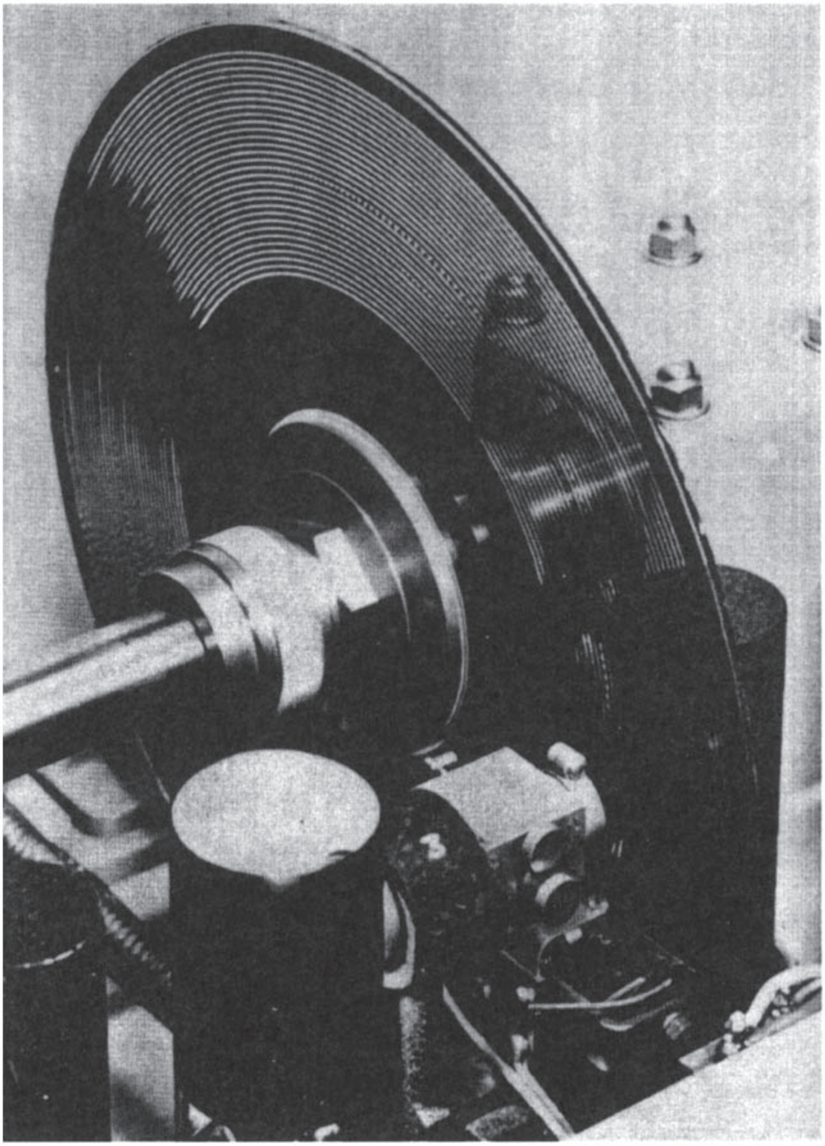


Figure 5.1 A glass disc used for message storage in the 1936 UK speaking clock. *By courtesy of BT Archives.*

to be structured to use words at regular intervals in order to avoid delays approaching the duration of one word or more. When the desired messages can be successfully made merely by replaying separately stored words in a specified order, the use of recorded natural speech means that the technical quality of the reproduction can be extremely high. It is apparent from the excellent speech from speaking clocks that there are applications where this method has worked extremely well. In the late 1960s it was used for some announcing machine applications in association with general-purpose computers, such as to provide share prices from the New York Stock Exchange to telephone enquirers.

The development of large cheap computer memories has made it practicable to store speech signals in digitally coded form for use with computer-controlled replay. As long as sufficiently fast memory access is available, this arrangement overcomes the timing problems of analogue waveform storage. Digitally coded speech waveforms of adequate quality for announcing machines generally use digit rates of 16–32 kbits per second of message stored, so quite a large memory is needed if many different elements are required to make up the messages.

For several years now there have been many computer voice-response systems commercially available that work on the principle of stored digitally coded message elements derived from human speech. The simplest of these systems involve merely recording the required components of the messages, which are then concatenated together without any modification to the individual elements. This simple concatenation can work well when the messages are in the form of a list, such as a simple digit sequence, or if each message unit always occurs in the same place in a sentence, so that it is comparatively easy to ensure that it is spoken with a suitable timing and pitch pattern. Where a particular sentence structure is required, but with alternative words at particular places in the sentence, it is important that the alternative words should all be recorded as part of the right sort of sentence, because they would otherwise not fit in with the required sentence intonation. For list structures it is desirable to record two versions of every element that can occur either in the final or non-final position. The appropriate falling pitch can then be used for the final element in each list. Even for messages that are suitable for simple stored waveform concatenation, great care has to be taken in recording and editing the separate message components, so that they sound reasonably fluent when presented in sequence. For any large body of messages it is worthwhile to provide a special interactive editing system, in which any section of waveform can be marked and replayed, either in isolation or joined to other sections. By this means it is possible to select the best available recording and choose the precise cutting points for greatest fluency. Even with these special tools the editing is labour-intensive, and it can be very time-consuming to achieve good results with a message set of moderate size.

There are a number of difficulties associated with using stored speech waveforms for voice output when a variety of different messages are required. In normal human speech the words join together, and the inherently slow movements of the articulators mean that the ends of words interact to modify the sound pattern in a way that depends on the neighbouring sounds. The pitch of the voice normally changes smoothly, and intonation is very important in achieving fluency and naturalness of speech. It therefore follows that if single versions of each word are stored they cannot produce fluent speech if simply joined together in all the different orders that might be needed for a wide variety of messages. Over 30 years ago laboratory experiments with arbitrary messages generated in this way demonstrated that the completely wrong rhythm and intonation made such messages extremely difficult to listen to, even though the quality of the individual words was very high. In recent years techniques have been developed which have made it possible to modify the pitch and timing of stored waveforms, and these methods will be described in Section 5.5.

Another problem with waveform storage is merely the size of memory needed to store a large vocabulary, although the current trend in memory costs is making this disadvantage less serious for many applications.

5.2.2 Concatenation of vocoded words

The large amount of digital storage needed for speech waveforms can be greatly reduced by using a low-bit-rate coding method for the message elements. Some ingenious methods have been developed to reduce the digit rate of stored waveforms, by exploiting various forms of redundancy. A widespread technique is to use some type of vocoder (most often using LPC, see Chapter 4), which can reduce the digit rate of the stored utterances to 2,400 or even 1,200 bit/s, albeit with some reduction in speech quality compared with the high-digit-rate stored waveform approach. A good example of the use of LPC vocoder methods to reduce the memory requirements is in the "Speak & Spell" educational toy that was produced by Texas Instruments. Since the first introduction of single-chip LPC synthesizers in "Speak & Spell" in the late 1970s, devices of this type have become widely used for message synthesis, and there are now many products which include speech synthesis based on LPC vocoder storage.

Besides memory size reduction there is another great potential advantage with vocoder storage of message elements: the pitch and timing of messages can easily be changed without disturbing the spectral envelope pattern of the stored words. It is thus possible in principle to modify the prosody to suit a word's function in a sentence, without storing alternative versions of each word. Although techniques are now available for making prosodic modifications directly to stored waveforms (see Section 5.5), the process is much simpler with a vocoder. In a vocoder the pitch is changed merely by varying the fundamental frequency parameter fed into the synthesizer. The timing can be varied by omitting or repeating occasional frames of the control data. There are, of course, difficulties in deriving suitable methods to control the timing and intonation patterns. If there is a fixed sentence structure that sometimes requires different words in particular places, the intonation pattern can be specified in advance, and merely imposed on the words that are chosen. If the sentence structures of the messages are not determined in advance it is necessary to derive the pitch and timing according to a set of rules. This aspect will be discussed in Chapter 7. Even when an appropriate prosody can be imposed, there are still problems at word boundaries because speech properties will in general not match where words join, but it is easier to avoid discontinuities than when concatenating stored waveforms. Co-articulation at word boundaries can be crudely simulated with concatenation of vocoder words by applying some smoothing to the control signals where the words join. A smoothing time constant of about 50 ms will remove the more serious effects of discontinuity at the word boundaries. Alternatively, one can have an overlap region, where the parameters for the end of one word can be gradually blended with those for the start of the next. Such crude methods will, of course, be very unrepresentative of the actual co-articulation that occurs between words of human speech.

5.2.3 Limitations of concatenating word-size units

Whether using waveform or vocoder storage, there is an insuperable limitation with all systems using stored human speech as words or larger units: every message component must have been previously spoken by a human speaker. It is thus not possible to add even a single new word without making a new recording. This process needs a suitable acoustic environment and either finding the original talker to say the new material or re-recording and editing the entire vocabulary with a new talker. This restriction prevents whole-word message storage from giving good results whenever it is necessary to add new items locally to a system already in service. The vocabulary size for word concatenation systems is limited by memory availability and by the problem of recording and editing all the words.

5.3 CONCATENATION OF SUB-WORD UNITS: GENERAL PRINCIPLES

5.3.1 Choice of sub-word unit

One obvious way of overcoming the limitations of word concatenation systems is to reduce the size of the stored units. Harris (1953) described early experiments with "building blocks of speech", in which he tried to synthesize words by concatenating waveform recordings of the length of individual phones. He found that it was essential to have several allophones of most phonemes, and even then intelligibility of some words was poor, due to the lack of natural co-articulation. A way of using small units, while still achieving natural co-articulation, is to make the units include the transition regions. Many speech sounds contain an approximately steady-state region, where the spectral characteristics are not greatly influenced by the identities of the neighbouring sounds. Thus concatenation of small units is better if each unit represents the transition from one phone to the next, rather than a single phone in isolation. A popular unit is the **diphone** (sometimes called a **dyad**), defined to contain the transition from the steady-state portion of one phone to the steady-state portion of its immediate neighbour. Storing transition regions in this way requires the number of units to be of the order of the square of the number of individual phonemes in the language, so might typically be about 1,600. This number makes it possible to achieve an unlimited vocabulary. Diphones provide a straightforward way of capturing the most immediate effects of co-articulation with manageable storage requirements as the individual units are quite short and not too numerous.

Another type of transition unit is the **demisyllable,** which represents half a syllable split in the centre of the vowel. Demisyllables are slightly more numerous than diphones, because they may need to provide several consonant clusters. The use of demisyllables can be a great advantage for languages like English, where consonant clusters are common, because some consonant phonemes are acoustically quite different when they occur in clusters, compared with when they are in simple sequences of alternating consonants and vowels.

Diphone-type methods work well provided that any units to be concatenated have similar acoustic characteristics in the region of the join. However, in fact there may be significant variation even at the centres of some phones (i.e. at the

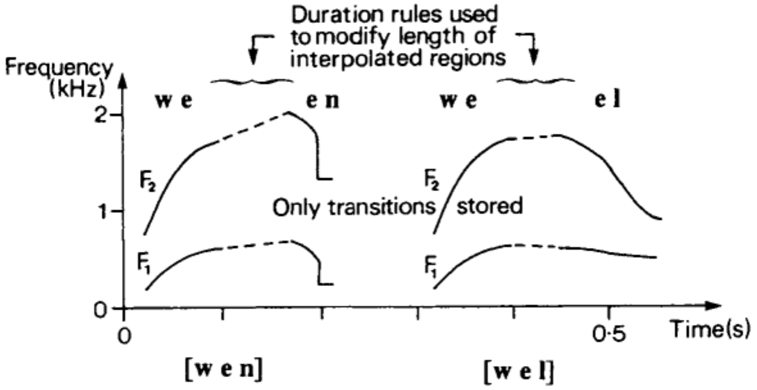


Figure 5.2 Construction of the words "when" and "well" using interpolation between formant-coded diphones to reduce discontinuity effects at junctions.

diphone boundaries) according to the identities of the adjacent phonemes. As a result there can often be a considerable discontinuity in the acoustic specification where two diphones join. Consider the words "well" and "Ben". The **/w/** and **/l/** phonemes are normally associated with long formant transitions, due to the large articulatory movements associated with these consonants. On the other hand the stop consonants **/b/** and **/n/** involve much more rapid transitions. In the middle of "well" therefore, the normal articulatory position associated with the isolated form of the **/e/** phoneme is not reached, whereas in "Ben" any undershoot will be quite minor. If we now consider the word "when", it is obvious that a [we] diphone appropriate for "well" will not join correctly to the [en] diphone suitable for "Ben". If a vocoder representation is used, one way of obtaining reasonable transitions in these cases is to store minimal-length transitions and interpolate the synthesis parameters between the ends of the stored diphones, as illustrated in Figure 5.2.

Another approach to the problem of discontinuities is to extend the set of diphones to include allophones whenever there is too much variation for a single diphone to be sufficient. A similar effect can be achieved by using larger units for these problematic contexts, so that one or more complete phones are retained together with the surrounding transitions. Variable-size units (often referred to as **polyphone** or ***N*-phone** units) have become popular in recent years as computing power and memory have increased. By careful selection of larger units where necessary, speech quality can be improved considerably, although at the expense of greater memory requirements and added complexity in choosing the units to use.

5.3.2 Recording and selecting data for the units

The quality of the speech obtained from a concatenative synthesis system is critically dependent upon obtaining good examples of the synthesis units, spoken clearly and consistently by a single human talker. It is usual to construct a special corpus, generally designed to cover all possible diphone contexts and possibly other context groupings known to show allophonic variation (e.g. valid consonant clusters in the language). There is some debate about whether the recorded data should consist of isolated words or of sentences, and about whether real words or nonsense words should be used. A popular approach is to use nonsense words, with the diphones embedded in a few carefully selected carrier phrases. This approach has the advantage that it is easy to systematically vary phonemic context, while keeping prosodic context and stress level as constant as possible for all the diphone units. The level of stress needs to be chosen carefully so that the speech is perceived to be clear but not over-articulated, and naturalness may be improved by also incorporating separate units for unstressed vowels.

Speech is highly variable, and there may be considerable acoustic variability even between different repetitions of the same phrase spoken in the same way by the same talker. Any variability is a problem for unit concatenation, and it is often helpful to record a few different examples of each phone sequence for later selection of the units. Once the data have been recorded the units need to be excised from the recordings, choosing the units and the breakpoints carefully to obtain the smoothest joins between units that need to be concatenated. Extracting suitable units has traditionally been a labour-intensive process, with some initial segmentation being performed automatically but followed by fine-tuning of the segmentation and final selection of the units by a human expert. Recently, automatic techniques have become increasingly used for all aspects of the process, including selecting the units to use, choosing the best example of each one, and locating the breakpoints for each example. These methods are generally based on minimizing distance measures based on spectral discontinuities, while also satisfying practical criteria such as total memory requirements.

5.3.3 Varying durations of concatenative units

In human speech, durations of sounds vary according to their positions in relation to the prosodic pattern of the sentence they are in. In a practical synthesis system it will therefore generally be necessary to vary the durations of the synthesis units. In the case of a text-to-speech system, phone durations will be generated by rule (see Section 7.5.1), and these durations must then be imposed on the synthesis units. The positions of phone boundaries would normally be included with the diphones or other units spanning phone boundaries, so it is straightforward to calculate the amount of duration modification required for each portion of a unit. One simple way of obtaining the required duration is to apply a uniform lengthening or shortening over the whole phone, which is appropriate for modifying the duration of a fricative for example. To change the duration of a vowel, it is probably better to lengthen or shorten the central region (around the join between two diphones), because inherent limitations in the speed of articulatory movements tend to mean that transitions vary less in duration than the more steady-state regions of vowels. There is no doubt, however, that transitions produced at one rate of articulation will not be of exactly the same form as would be produced at a very different speed. Thus concatenative methods may be less suitable for systems in which it is required to provide extensive variation in the speed of talking. The techniques for achieving the lengthening or shortening of the synthesized signal depend on the synthesis method used, and are described in the following sections.

5.4 SYNTHESIS BY CONCATENATING VOCODED SUB-WORD UNITS

Vocoder parameters for the sequence of synthesis units can be simply joined together, applying any necessary duration modifications. When shortening is required frames can be removed, and lengthening can be achieved by interpolating the synthesis parameters for the region to be lengthened. Interpolation across the boundary between two units has the advantage of reducing any discontinuities in the parameters. Thus, by only storing short transition regions (see Figure 5.2), interpolation will usually be required to lengthen the units and at the same time minimize discontinuities. Any remaining discontinuities can be reduced after concatenation by applying a smoothing function to the parameters, in the same way as for concatenating vocoded words (see Section 5.2.2). Pitch modifications are easily achieved by varying the separate fundamental frequency parameter.

The quality of speech synthesized by vocoder-based concatenation cannot be better than the vocoder method employed. Although formant synthesizers can produce very natural-sounding speech if the controls are set appropriately, the quality of speech from formant vocoders suffers due to the difficulties involved in deriving these controls automatically. If careful hand-editing is used to correct analysis errors, a formant vocoder could be applied to generate the synthesis units. However, mainly because of the ease of analysis and availability of very-low-cost synthesis chips, LPC methods are much more widely used. The underlying quality is then limited to that possible from an LPC vocoder (see Section 4.3.3).

5.5 SYNTHESIS BY CONCATENATING WAVEFORM SEGMENTS

Consider the problem of joining together two segments of vowel waveform. Discontinuities in the combined waveform will be minimized if the join occurs at the same position during a glottal cycle for both the segments. This position should correspond to the lowest-amplitude region when the vocal-tract response to the current glottal pulse has largely decayed and just before the following pulse. Thus the two segments are joined together in a **pitch-synchronous** manner. To obtain a smooth join, a tapered window is applied to the end of the first segment and to the start of the second segment, and the two windowed signals are overlapped before being added together (see Figure 5.3). Because the method involves a combination of pitch-synchronous processing with an **overlap-add (OLA)** procedure to join the waveform segments, it is known as **pitch-synchronous overlap-add (PSOLA)**.

The PSOLA technique can be used to modify pitch and timing directly in the waveform domain, without needing any explicit parametric analysis of the speech. The position of every instance of glottal closure (i.e. pitch pulse) is first marked on the speech waveform. These **pitch markers** can be used to generate a windowed segment of waveform for every pitch period. For each period, the window should be centred on the region of maximum amplitude, and the shape of the window function should be such that it is smoothly tapered to either side of the centre. A variety of different window functions have been used, but the **Hanning** window (shown in Figure 5.3) is a popular choice. The window length is set to be longer than a single period's duration, so that there will always be some overlap between adjacent windowed signals. The OLA procedure can then be used to join together a

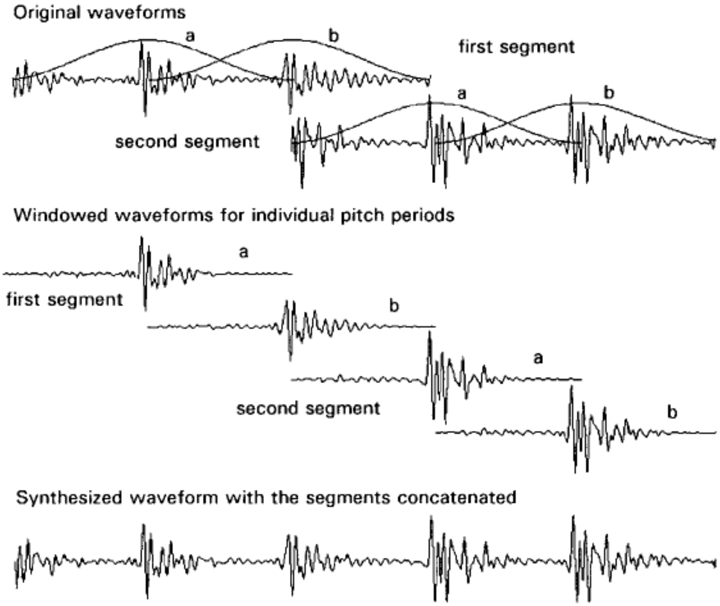

Original waveforms
a
b
first segment
a
b
second segment
Windowed waveforms for individual pitch periods
a
first segment
b
a
second segment
b
Synthesized waveform with the segments concatenated

Figure 5.3 Decomposing speech waveforms into a sequence of pitch-synchronous overlapping windows. For two voiced speech segments, pitch markers and window placement are shown in the top plots, and the outputs of the analysis windows are shown in the middle plots. The bottom plot shows the waveform that is obtained if the PSOLA technique is used to join the last analysis window of the first segment to the first analysis window of the second.

sequence of windowed signals, where each one is centred on a pitch marker and is regarded as characterizing a single pitch period. By adding the sequence of windowed waveform segments in the relative positions given by the analysed pitch markers, the original signal can be reconstructed exactly. However, by adjusting the relative positions and number of the pitch markers before resynthesizing, it is possible to alter the pitch and timing, as described below.

5.5.1 Pitch modification

The pitch of the signal can be raised by reducing the spacing between the pitch markers, and lowered by increasing this spacing. Examples of these modifications are shown in Figure 5.4. As the degree of overlap between successive windows is altered, the energy in the resynthesized signal will tend to vary, but a normalization factor can be applied to compensate for this artefact of the technique.

To be successful, the pitch-modification technique needs to change the pitch of the signal (given by the repetition rate of the pitch pulses) while not altering the spectral envelope (i.e. the formant frequencies and bandwidths). Thus the analysis window length needs to be short enough to be dominated by only a single pitch pulse, but long enough to capture the formant structure with sufficient accuracy. The popular window length of twice the local pitch period has been found to be a

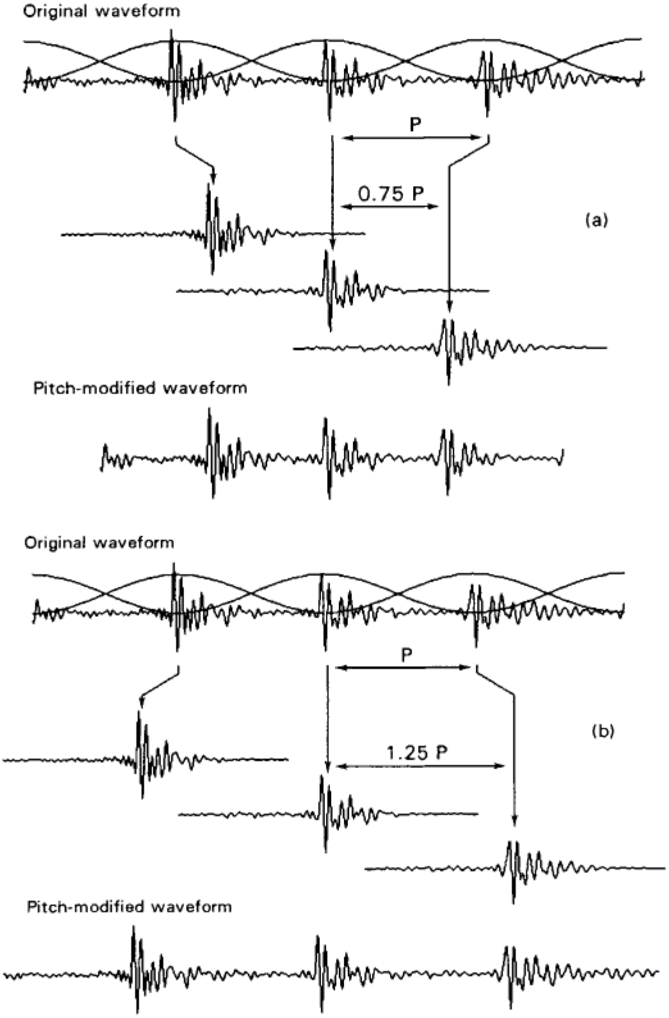


Figure 5.4 Using PSOLA to modify the pitch of a speech signal: (a) raising the pitch by 20% (the period, P, is multiplied by 0.75), (b) lowering the pitch by 20% (P is multiplied by 1.25). To modify the pitch, the pitch markers for the original signal are first repositioned according to the new pitch. The new signal is then constructed by adding the outputs of the analysis windows at this new pitch spacing.

good compromise, and can be used to achieve pitch modifications ranging from one half to twice the pitch of the original signal. The effect of this windowing of the signal tends to cause some widening of the formant bandwidths when the pitch is modified, but a moderate degree of widening does not seem to be perceptually significant. Widening of formant bandwidths becomes more severe as the pitch of the analysed signal increases, so the analysis window becomes shorter and hence there is a decrease in the accuracy with which the formant structure is preserved.

5.5.2 Timing modification

It is straightforward to use PSOLA to modify the timing of an utterance by careful selection of the sequence of pitch markers to use for synthesis. Pitch markers can be replicated where lengthening is required, and removed when a region is to be shortened. The sequence of pitch markers gives the order of the analysis windows to use when constructing the synthesized signal. Synthesis is achieved by applying the OLA procedure to join these windowed segments together at a spacing corresponding to the required synthesis pitch period. When choosing the sequence of pitch markers to use in order to achieve the required timing, it is necessary to take into account the changes in duration that will occur as a by-product of any pitch modifications. If the pitch is altered, some adjustment to the sequence of pitch markers will be needed even to keep the timing the same as for the original signal.

Timing can be modified with little acoustic distortion using the above method to achieve the effect of increasing speaking rate by a factor of up to about four, but to reduce speaking rate by rather less. When slowing down unvoiced regions by more than a factor of about two, the regular repetition of identical segments of signal tends to introduce a buzzy quality to the synthesized speech. This buzziness can be avoided by reversing the time-axis for every alternate segment, after which reasonable quality is obtained for slowing down by a factor of up to about four.

5.5.3 Performance of waveform concatenation

For PSOLA to work well, the positions of instances of glottal closure must be marked accurately on all the waveform segments. There are methods for determining these pitch markers automatically from the speech waveform, but these methods generally make some errors which need to be corrected by hand based on expert visual inspection of the waveform. More reliable automatic extraction of pitch markers is possible by using a **laryngograph** to record glottal activity simultaneously with the speech recordings. Whatever method is used to derive the pitch markers, part of this process will involve identifying unvoiced regions of the speech. For these regions, the positions of the analysis windows are not critical, and it is generally sufficient to place the pitch markers in arbitrary positions at a constant rate (although some care is needed for stop consonants).

Once speech segments and associated pitch markers are available, the PSOLA method described above is extremely simple to implement and requires very little computation, but it does need a lot of memory for storing the units. Some memory saving is possible by using a simple waveform coding technique such as DPCM (which typically more than halves the amount of memory required). However, the more complex coding methods that would be needed to obtain greater compression are not generally used with time-domain waveform synthesis, mainly because they would add considerable complexity to an otherwise simple synthesis procedure.

Because the individual message parts are obtained directly from human utterances, speech synthesized by waveform concatenation can be very natural-sounding. However, this naturalness is only achieved if any two segments to be concatenated have similar pitch periods and spectral envelopes that match at the join. Concatenation of waveforms provides no straightforward mechanism for avoiding spectral discontinuities. Thus achieving natural-sounding fluent synthetic speech often requires a painstaking trial-and-error process to select examples that are known to join together smoothly for the most common combinations.

The PSOLA method described in this section operates directly on the speech waveform, and is therefore known as **time-domain PSOLA (TD-PSOLA)**. There are now several other variants of the general PSOLA technique, a few of which are briefly mentioned in the next section.

5.6 VARIANTS OF CONCATENATIVE WAVEFORM SYNTHESIS

An alternative to performing the signal manipulations directly in the time domain is to first apply a Fourier transform to compute the short-term spectrum. Prosodic modifications and segment joining are then carried out in the frequency domain, before applying an inverse Fourier transform to convert back to the time domain. For this **frequency-domain PSOLA (FD-PSOLA)** approach, a longer window of typically four times the local pitch period is used so that the pitch harmonics are resolved in the spectral representation. The short-term spectral envelope is then estimated (using linear prediction for example). Taking the short-term spectrum that was obtained from the original Fourier transform and dividing by the estimated spectral envelope gives an estimate of the spectrum of the glottal source. The spacing between the harmonics in this source spectrum can then be modified to change the pitch. FD-PSOLA has the advantage of providing the flexibility to modify the spectral characteristics of a speech signal, including applying spectral smoothing at diphone boundaries. However, although the technique has proved to be a useful research tool, it has not been widely adopted for practical systems as it is very demanding computationally as well as having high memory requirements for segment storage.

In **linear-predictive PSOLA (LP-PSOLA),** speech is parameterized using LPC and the TD-PSOLA method for prosodic modification is applied to the linear-prediction error signal. Thus, as with FD-PSOLA, the excitation is separated out from the spectral shaping due to the vocal tract, so it is easy to modify the spectral envelope (to smooth segment boundaries for example). However, in the case of LP-PSOLA, prosodic modifications are easier as they are applied in the time domain. In addition, substantial memory savings are possible by using coded forms of the prediction error signal (using CELP for example). Due to these advantages, versions of LP-PSOLA have been adopted in a number of synthesis systems.

Multi-band resynthesis PSOLA (MBR-PSOLA) uses simple waveform concatenation, but first applies MBE coding (which represents voiced speech as a sum of harmonically related sinusoids: see Section 4.3.2) to the segment database. The idea is to modify (and resynthesize) the segments so that they will then be more amenable to waveform concatenation. The pitch is set to be constant throughout the database and, in addition, the phases of the harmonics are reset to the same (appropriately chosen) values at the beginning of each pitch period. Advantages of making these modifications are:

1. Explicit pitch marking is no longer required because all segments have the same known pitch value.

2. The method completely avoids potential problems due to mismatches between the pitch or phase structure of the segments to be concatenated.
3. Because all the voiced segments have the same pitch and the same harmonic phases, the effect of spectral envelope interpolation between two segments can be achieved by performing interpolation in the time domain. This temporal interpolation provides an efficient yet effective method for smoothing spectral discontinuities at segment boundaries.

The modifications are applied just once to the stored database, so these advantages are achieved without adding to the complexity of the simple timedomain synthesis operation itself. Furthermore, because the database has a constant pitch and fixed harmonic phases, and because characteristics due to the vocal tract evolve only slowly with time, the sample value at a given position in any pitch period will generally be similar to the value for the corresponding position in the previous pitch period. It is therefore possible to make a large saving in memory by applying a version of DPCM in which the differential coding is applied to corresponding samples in adjacent pitch periods (rather than to adjacent samples). This technique has been used in a modified, storage-efficient variant of MBR-PSOLA, termed simply **multi-band resynthesis overlap-add (MBROLA),** which has been used for synthesis systems in a variety of different languages.

5.7 HARDWARE REQUIREMENTS

Considering word-based systems first, stored waveforms are typically coded at 16–32 kbits/s. Assuming the lower figure, 1 Mbyte will store about 8 minutes of speech. This duration would be suitable for a reasonable range of pre-determined messages, and could be sufficient for an announcing system with a large variety of alternative words used in a few standard sentence types, such as for routine railway station announcements. The decoders for use with simple waveform storage are extremely cheap and many are available as single-chip devices. The word-selection process is simple, so memory costs are likely to dominate for a complete system.

For a system providing information to the general public over the telephone network the economics of system design are very different. Here many enquirers may access the system simultaneously, all wanting different messages. However, the memory can be common to the whole system, and if the number of channels is very large the memory cost will not contribute greatly to the cost per channel, so it becomes practicable to provide many more messages using waveform storage.

A set of diphones provides a simple sub-word system and might represent somewhere in the region of 3–4 minutes of speech from one talker for a single language. The memory requirements obviously depend on how the diphones are stored. At one extreme the original speech waveforms at a 16 kHz sampling rate with 16 bits per sample typically occupy around 5 Mbytes, whereas an LPC version of this database would need less than 200 kbytes. Methods such as LP-PSOLA and MBROLA require more memory than a simple LPC system, but would generally need less than 1 Mbyte for a complete diphone set. Of course, memory requirements increase if a larger inventory of synthesis units is used.

Computational requirements are very low for TD-PSOLA, and increase only slightly for MBROLA (for 16 kHz-sampled speech, real-time operation has been achieved on an

Intel486 processor). LP-PSOLA increases the computation by a factor of about 10 relative to TD-PSOLA, and real-time implementations of LP-PSOLA have generally used a dedicated DSP. A low-cost alternative is provided by simple LPC synthesis, for which suitable DSP chips are widely available.

CHAPTER 5 SUMMARY

- Message synthesis from stored waveforms is a long-established technique for providing a limited range of spoken information. The simplest systems join together word-size units. The technical quality of the speech can be high, but it is not possible to produce good results for a wide range of message types.
- Synthesis from diphones gives complete flexibility of message content, but is limited by the difficulty of making the diphones represent all the coarticulation effects that occur in different phonetic environments. To obtain the best quality from diphone synthesis, care must be taken in selecting and excising the examples. Quality may be improved by adding to the simple diphone set to include allophone-based units or longer units spanning several phones.
- Vocoders require a small fraction of the memory needed for simple waveform storage, and also make it easy to vary the pitch and timing, and to smooth the joins between any units being concatenated. Synthesis quality is, however, limited by the inherent vocoder quality.
- The technique known as pitch-synchronous overlap-add (PSOLA) allows good synthesis quality to be achieved by concatenating short waveform segments. Smooth joins are obtained by concatenating segments pitch-synchronously and overlapping the end of one segment with the start of the next.
- By decomposing the speech signal into individual pitch periods with overlapping windows, prosodic modifications are easy with PSOLA. Timing can be modified by repeating or removing individual pitch periods, and pitch can be changed by altering the spacing between windows before resynthesis.
- Time-domain PSOLA is simple to implement, but needs a lot of memory and cannot smooth any spectral discontinuities occurring at segment boundaries.
- Other variants of PSOLA address the above limitations by incorporating some parametric representation of the speech, such as LPC or MBE coding, while retaining the PSOLA technique for prosodic modifications.
- The hardware cost for synthesis from stored human speech is dominated by the memory requirements except for multi-channel systems.

CHAPTER 5 EXERCISES

E5.1 Discuss the advantages and disadvantages of message synthesis by waveform concatenation of whole words.

E5.2 Why can it be beneficial to use vocoders for concatenative synthesis?

E5.3 What are the advantages and disadvantages of diphone synthesis?

E5.4 What are the potential problems with synthesis by concatenating waveform fragments and how are these problems addressed by the PSOLA technique?

CHAPTER 6

Phonetic Synthesis by Rule

6.1 INTRODUCTION

Concatenative synthesis techniques join together (often in quite sophisticated ways) fragments of human utterances, either as raw waveforms or in some coded form. An alternative is to generate synthetic speech by applying a set of rules to convert from a symbolic description to control parameters to drive an articulatory or formant synthesizer. We will use the term **phonetic synthesis by rule** to refer to the use of acoustic-phonetic rules for generating synthesizer control parameters from a phonemic description of an utterance together with any required prosodic information. For some applications requiring limited sets of messages or special effects, it can be appropriate to use such a description directly as the input to a synthesis system. It is, however, more usual for phonetic synthesis by rule to form one component of a more general system for generating speech from text or some other higher-level message description. Rules can also be used in the other components of such a system. These uses of synthesis by rule will be discussed in the next chapter, while the current chapter concentrates on the acoustic-phonetic level. A characteristic of the methods considered in this chapter is that they do not store utterances of human speech in any form, although they do, of course, usually make extensive use of human utterances for guidance in formulating the rules.

6.2 ACOUSTIC-PHONETIC RULES

Human speech is produced as a result of muscular control of the articulators. The acoustic properties caused by even quite simple gestures can, however, be very complicated. For example, in the release of a stop consonant, such as [t], there may be very noticeable acoustic differences caused by slight variation of the relative timing of the tongue movement away from the alveolar ridge and the bringing together of the vocal folds in preparation for the voicing of a following vowel. If the **voice onset time** (VOT) is short there will be very little aspiration, and the perceptual quality will be much closer to [d]. Because the voicing then starts during the early stages of the tongue movement, it excites the transition of the first formant, which can be clearly seen on spectrograms. For a longer VOT the glottis will be wide open at release, and the resultant greater air flow gives rise to aspiration, i.e. turbulent noise excitation of the higher formants for 60–100 ms.

The complex consequences of simple gestures have led some people to suggest that rules for the phonetic level of synthesis would be easiest to specify in articulatory terms, for driving an articulatory synthesizer. This viewpoint obviously has merit, but articulatory rules have not generally been adopted in practical speech synthesis systems for a number of reasons.

The most fundamental argument against using articulatory rules is that when humans acquire speech it is the auditory feedback that modifies their behaviour, without the speaker being consciously aware of the articulatory gestures. There are frequent cases of significantly non-standard articulatory strategies being used by some individuals to produce particular phonetic events. Although careful analysis sometimes reveals that the articulation of such people produces acoustic properties that differ consistently from the norm, the differences are not sufficient to cause phonetic confusion and will frequently not be noticed. In other cases the differences from the normal acoustic pattern are within the variation that occurs naturally between users of the more common articulation, and are not even detectable perceptually. The prime example of differing articulation for similar phonetic percepts occurs in the case of a good ventriloquist, who can produce a full range of speech sounds without externally obvious mouth movements. In developing an articulatory synthesis-by-rule system, it is thus often not easy to decide what the articulatory gestures should be for any particular phonetic event.

The second argument against using articulatory synthesis is merely the difficulty of accurately measuring articulatory gestures. Various techniques are available, such as X-rays, electro-myography, fibre-scopes etc., but all are inconvenient to use and of limited accuracy. By contrast, synthesis methods that specify sounds directly in terms of measurable acoustic properties can have their control rules simply related to the acoustic features that are required, even though these features might be quite complicated in some cases. It is not then necessary even to consider the possible underlying articulatory gestures.

The third main reason for not using articulatory synthesis for machine voice output is that existing articulatory synthesizers have been much less successful than formant-based methods for modelling the perceptually important acoustic features (as already mentioned in Chapter 2).

6.3 RULES FOR FORMANT SYNTHESIZERS

For the reasons outlined above, most acoustic-phonetic rule systems are designed for directly driving some form of formant synthesizer. The input at this level is normally a sequence of allophones, each associated with prosodic information to specify duration, pitch and possibly also intensity. Allophonic variation in speech arises from two causes, both of which are normally systematic in operation. The first cause results from the phonological rules of a language, which may specify that a particular **extrinsic** allophone of a phoneme should be used in certain environments, even though other allophones could be produced by a speaker without much difficulty. The second cause, giving rise to **intrinsic** allophones, is a direct consequence of the constraints of articulation. The actual formant frequencies in a short vowel are greatly influenced by the articulatory gestures for the consonants on either side, and in consequence the acoustic properties in the centre of the sound representing a particular vowel phoneme may differ substantially for different consonant environments. The extrinsic allophones must be specified by the input to a phonetic rule system, but it is possible to generate many of the co-articulation effects that give rise to intrinsic allophones automatically as a consequence of the way the rules operate.

Phonetic rule systems have been developed in a number of laboratories, and some have been incorporated in commercial text-to-speech products. With some phonetic rule systems, the researchers have found it advantageous to write a special notation for expressing the rules, to facilitate writing rule systems for a variety of languages. In others the rules have been written in a standard algorithmic computer language (e.g. Pascal or Fortran), with large numbers of conditional expressions to determine what synthesizer control signals should be generated for each type of phonetic event. A third approach is to have a very small number of computational procedures, driven by a large set of tables to represent the inventory of possible allophones. The numbers from the tables are then used by the computation to determine how all the synthesizer control signals should vary for any particular allophone sequence. An example of this table-driven method will be described in more detail, to illustrate the types of rules that are typically found useful.

6.4 TABLE-DRIVEN PHONETIC RULES

The rules described in this section are appropriate for a parallel formant synthesizer such as the one illustrated in Figure 2.15. They are based on the technique described by Holmes, Mattingly and Shearme (1964) for a simpler parallel formant synthesizer. The following description includes some minor improvements to the computational algorithm given in the 1964 paper.

The synthesizer control parameters are calculated as a succession of **frames,** each of which has a duration of 10 ms. The input to the system comprises the sequence of speech sounds required, and for each sound there is a duration (in frames), and a specification of how to derive the fundamental frequency contour. There is also an option to vary the loudness of each sound from its default value.

This table-driven system is based on the idea that most speech sounds can be associated with some target acoustic specification, which might not be reached, but which can be regarded as an ideal that would be aimed for in long isolated utterances. Simple vowels and continuant consonants are obvious examples where this concept seems generally appropriate. The target values for the 10 synthesizer control signals shown in Figure 2.15 are stored in a table for each such phone.

Other sounds, such as diphthongs and stop consonants, clearly have a sequence of acoustic properties, and each member of the sequence may be associated with a target specification and some transition rules for changing between targets. These sounds can be represented by a sequence of two or more component parts, each having its own table. Because a table in this system sometimes corresponds to a complete phone, and sometimes only to part of a phone, the term **phonetic element** has been used to describe the chunk of sound generated by the use of one table.

The table for each phonetic element also contains information relating to how transitions between target values are calculated around phonetic-element boundaries. In general, for transitions between a consonant and a vowel, it is the identity of the consonant that decides the nature of the transition. For example, nasal consonants have acoustic properties that change only slightly during the oral occlusion, but cause rapid changes at the boundary between consonant and vowel and fairly rapid but smooth formant transitions during the vowel. Fricative-vowel boundaries, on the other hand, can also have quite clearly discernible formant transitions during the frication. These types of transition are largely independent of the identity of the vowel involved, although the actual numerical parameter values in each transition obviously depend on the associated vowel target value.

The Holmes-Mattingly-Shearme (HMS) system was designed to achieve the above properties without requiring special tables for each possible vowel/consonant combination. As the type of transition in a vowel-consonant (VC) or consonant-vowel (CV) sequence is determined mainly by the consonant, the table entries associated with the consonant are used to define the transition type. The only quantities used from the vowel tables are their target values. The operation of a transition calculation in the HMS system is explained below.

6.4.1 Simple transition calculation

Consider a transition between a consonant, [w], and a following vowel, [e], for the second-formant frequency parameter. Appropriate values for the targets of the two sounds might be 750 Hz and 2000 Hz respectively. The system has a nominal boundary, where the two elements join, and has a method for calculating the parameter value at that boundary, taking into account the target value for the vowel and the identity of the consonant. The values either side of the boundary are derived by simple interpolation between the boundary value and the two target values, where the two interpolations are carried out over times that are specified in the consonant table. For each parameter the table for [w] will include:

1. its own target value;
2. the proportion of the vowel target used in deriving the boundary value;
3. a ‘fixed contribution’ to the boundary value, specified for the consonant;
4. the ‘internal’ transition duration within the consonant (in frames);
5. the ‘external’ transition duration within the vowel (in frames).

The boundary value is given by: fixed contribution+(proportion×vowel target). The [w] table might have the following values for F_2: target=750 Hz, proportion=0.5, fixed contribution=350 Hz, internal duration=4 frames, external duration=10 frames. If the F_2 target for the [e] is 2000 Hz, the boundary value is 350+(0.5×2000), which is 1350 Hz. The complete transition would then be as shown in Figure 6.1. For simplicity, this diagram does not illustrate the level quantization or time quantization that will be used in any practical system.

In the original HMS system, the entries controlling the form of a transition were only specified once for each parameter in a table, and so the transition calculation was of exactly the same form for CV and for VC pairs. To a first approximation, this symmetry is reasonable as the articulatory movements between the vowel and consonant configurations are likely to be of broadly similar form irrespective of the direction of the change. This arrangement was used in the 1964 system because it allowed an appreciable saving of memory for the limited-performance computers which were available at that time. However, now that computer memory is so cheap, there is no need to maintain this restriction and so each table can include separate specifications for the initial and final transitions.

The description above has focused on a transition calculation for one control signal in one CV pair. Different values in the tables are used to achieve appropriate transitions for other control signals, and different tables are provided for all other

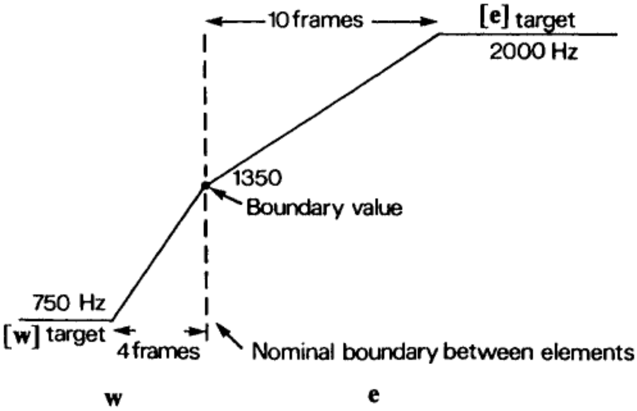


Figure 6.1 Second-formant transition for the sequence [we], using the Holmes-Mattingly-Shearme (HMS) rule system.

vowels and consonants. Of course, speech does not consist entirely of alternating consonants and vowels. Consonants often occur in clusters, and vowel sequences also occur, both within diphthongs and between syllables or words. The HMS system makes provision for these events by associating every phonetic element with a **rank**. Some phonemes that are undoubtedly consonants from a phonological point of view, such as [w], are acoustically more like vowels, and hence will contain formant transitions caused by adjacent consonants just as vowels will. Consonants such as stops will also tend to cause formant transitions in fricatives. The ranking system gives highest rank to those elements which have the strongest effect in determining the nature of transitions, and lowest rank to vowels. For any sequence of two elements the transition calculation is as described above; the table of the higher-rank element determines the nature of the transition, and the table of the lower-rank element is used only to provide parameter targets. If two adjacent elements have the same rank, the earlier one is arbitrarily regarded as dominant.

6.4.2 Overlapping transitions

The input to phonetic-level synthesis is required to specify a duration for each element. It can happen that the transition durations as specified in the tables are so long in relation to the element duration that there is insufficient time for the element to contain both the initial and final transitions without them overlapping in time. This effect is illustrated in Figure 6.2 for the F_2 transition of the sequence [wel] (the word "well"). The original HMS paper advocated a very crude method of producing some sort of transition in such cases, but a more satisfactory method is as follows. For the element under consideration ([e] in Figure 6.2), first construct separately that part of each transition which lies within the specified duration for the element, filling in the target value if necessary for the remainder of the frames of the element. Then construct the final parameter track by taking a weighted sum of the two component transitions over the duration of the element. The weighting

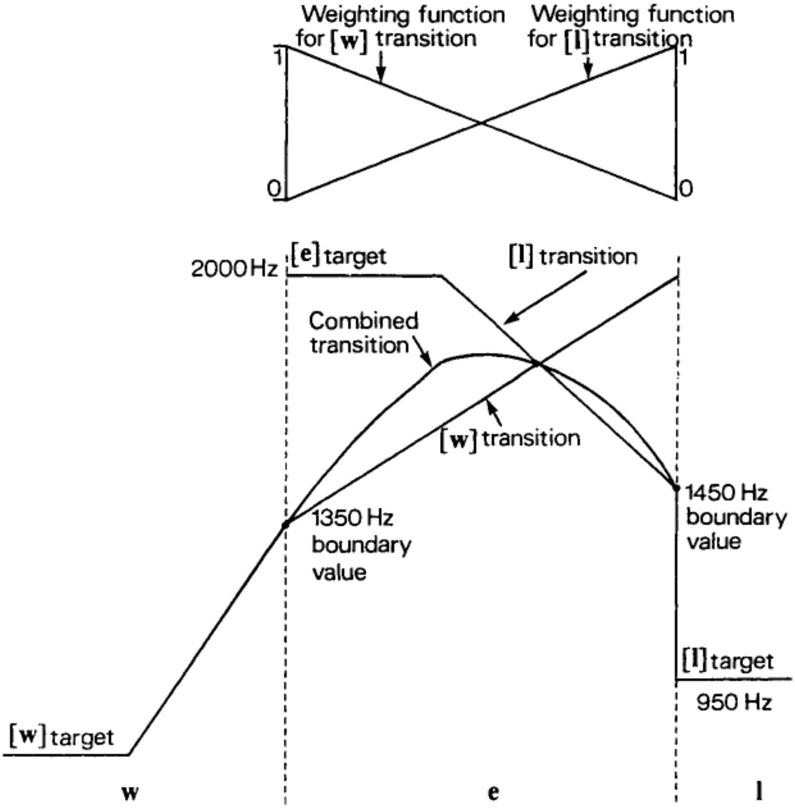


Figure 6.2 Method for dealing with overlapping transitions in a variant of the HMS technique.

function goes linearly from 1 to 0 for the initial transition, and from 0 to 1 for the final transition. The result of applying the linear weighting function to the linear transitions is to make the final components of the two transitions parabolic.

The use of the weighting functions when constructing parameter tracks produces smoother trajectories with a more natural-looking shape, whether or not the two transitions overlap. It is therefore preferable to apply this method of determining trajectories throughout, although listening comparisons have indicated that the difference is not obvious perceptually in the non-overlapping case.

6.4.3 Using the tables to generate utterances

Examples of table entries for the 10 parameters for a few typical elements are shown in Figure 6.3, and the resultant parameter tracks generated for the word "wells" for F_1, A_1, F_2 and A_2 are shown in Figure 6.4. The graphs in Figure 6.4 include the time quantization into frames, and also show magnitude quantization of each parameter into 6-bit integer values. The phonemic significance of the element

		Tgt.	L.Prop.	L.F.C.	L.I.D.	L.E.D.	R.Prop.	R.F.C.	R.I.D.	R.E.D.
Element name = **Q**	F_N	250	1	0	63	0	1	0	63	0
Rank = 63	A_{LF}	1	1	−10	0	3	1	−10	0	3
	F_1	900	1	0	63	0	1	0	63	0
	A_1	1	1	−10	0	3	1	−10	0	3
	F_2	2100	1	0	63	0	1	0	63	0
	A_2	1	1	−10	0	3	1	−10	0	3
	F_3	2900	1	0	63	0	1	0	63	0
	A_3	1	1	−10	0	3	1	−10	0	3
	A_{HF}	1	1	−10	0	3	1	−10	0	3
	V	1	1	0	63	0	1	0	63	0
Element name = **w**	F_N	250	0.5	125	0	4	0.5	125	4	4
Rank = 10	A_{LF}	51	0.5	21	4	4	0.5	23	4	4
	F_1	200	0.5	100	4	4	0.5	50	4	6
	A_1	43	0.5	21	4	4	0.5	22	4	4
	F_2	750	0.5	550	4	8	0.5	350	4	10
	A_2	40	0.5	24	4	4	0.5	20	4	4
	F_3	2000	0.5	800	4	4	0.5	1000	4	4
	A_3	36	0.5	22	4	4	0.5	18	4	4
	A_{HF}	1	0.5	0	4	4	0.5	0	4	4
	V	63	0.5	32	0	0	0.5	32	0	0
Element name = **e**	F_N	250	0.5	125	0	0	0.5	125	0	0
Rank = 2	A_{LF}	52	0.5	23	3	3	0.5	21	4	4
	F_1	650	0.5	325	3	3	0.5	290	4	4
	A_1	49	0.5	23	3	3	0.5	21	4	4
	F_2	2000	0.5	1000	3	3	0.5	950	4	4
	A_2	48	0.5	24	3	3	0.5	24	4	4
	F_3	2500	0.5	1375	3	3	0.5	1375	4	4
	A_3	53	0.5	24	3	3	0.5	24	4	4
	A_{HF}	54	0.5	23	3	3	0.5	23	4	4
	V	63	0.5	32	0	0	0.5	32	0	0
Element name = **l**	F_N	250	0.5	125	0	6	0.5	125	0	0
Rank = 11	A_{LF}	49	0.5	1	0	0	0.5	20	0	2
	F_1	350	0.5	225	0	6	0.5	175	0	6
	A_1	46	0.5	−2	0	0	0.5	−13	0	2
	F_2	950	0.5	450	0	6	0.5	400	0	6
	A_2	47	0.5	12	0	0	0.5	−9	0	2
	F_3	2500	0.5	1250	0	6	0.5	1400	0	6
	A_3	50	0.5	15	0	0	0.5	−6	0	2
	A_{HF}	47	0.5	12	0	0	0.5	−6	0	2
	V	63	0.5	32	0	0	0.5	32	0	0
Element name = **z**	F_N	250	0.5	125	2	3	0.5	125	0	0
Rank = 20	A_{LF}	36	0.5	1	0	0	0.5	20	0	0
	F_1	275	0.5	150	2	3	0.5	125	2	3
	A_1	33	0.5	−2	0	0	0.5	14	0	0
	F_2	1700	0.5	950	2	3	0.5	850	2	3
	A_2	37	0.5	12	0	0	0.5	16	2	2
	F_3	2550	0.5	0	2	3	0.5	1300	2	3
	A_3	40	0.5	15	0	0	0.5	16	2	2
	A_{HF}	57	0.5	12	0	0	0.5	25	2	2
	V	32	1	−31	10	2	0.5	32	0	0

Figure 6.3 Example table entries for five phonetic elements in a modified HMS system. For each parameter of each element, the table specifies the target value and, for both the initial (left) and final (right) transitions, the proportion, the fixed contribution and the internal and external transition durations.

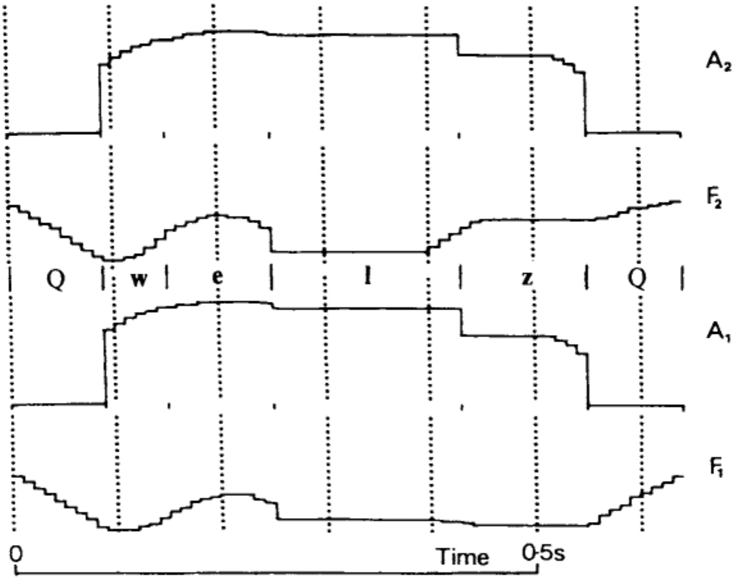


Figure 6.4 Tracks of four parameters for the word “wells”, [welz], generated from the tables in Figure 6.3.

names shown in Figures 6.3 and 6.4 should be obvious from the conventions of English orthography, except for Q which is used to signify silence.

The method of calculating parameter tracks in this table-driven system is extremely versatile. By suitable choice of table entries a wide variety of transitions, appropriate for formant frequencies, formant amplitudes and degree of voicing, can be constructed. The dominance system, determined by the ranks, is able to provide many of the co-articulatory effects that occur, particularly where high-ranking consonants are adjacent to vowel-like sounds. However, values chosen to suit CV boundaries will not normally give sensible results if used to define the transitions between, for example, pairs of stop consonants, or between stops and nasals.

The system described above can be vastly improved by providing different elements for different allophones of some of the phonemes. Element selection rules can then be used to take into account the phonetic environment of every phoneme in determining which element or sequence of elements should be used to specify the appropriate allophone. Without these selection rules around 60 elements are needed to provide one element or element sequence for each English phoneme. Increasing the number of elements to be a few hundred, where many of the extras are provided specifically to deal with the particular problems that would otherwise occur with consonant clusters, can considerably improve the naturalness of the synthesized speech. The additional elements only require a modest amount of extra memory, and do not significantly increase the cost of a speech synthesis system, once the table values have been determined. The number of elements is still far less than the number of items needed in a diphone system because so much of the coarticulation between vowels and consonants can be generated by the rules.

As it stands, the technique described above does not deal with co-articulation spreading across several phonemes. For example, lip-rounding for [w] is often maintained for several of the following speech sounds if none of them specifically requires spread lips. Here again, however, element selection rules can take into account remote phonetic environment to generate a suitable allophone in each case. Extra elements can also be added for any situation in which the notion of a single target specification for a phoneme is not sufficient to capture all realisations of that phoneme in all different phonetic contexts.

Even if several hundred tables were needed to cover allophonic variation, the memory requirements are modest compared with those for concatenative synthesis systems. For example, using 6-bit controls with the tables shown in Figure 6.3, each table would occupy a total of less than 70 bytes. Thus, for example, 500 tables would occupy less than 35 kbytes. If memory saving were important, the tables could in fact be stored more economically by employing efficient coding methods or by using fewer than 6 bits for certain parameters (such as the degree of voicing).

6.5 OPTIMIZING PHONETIC RULES

So far the usual method of choosing rules for a phonetic synthesis-by-rule system has required a large amount of effort and the skill of an experimental phonetician, assisted by speech analysis tools providing, for example, spectrograms. Human subjective judgement provides, of course, the ultimate criterion for success, but it is very dangerous to rely too much on listening to guide small improvements because the listener easily gets perceptually saturated by repeated listening to the same short utterance. Phonetic theory, specifying the normal acoustic consequences of various articulatory events, has been widely used to formulate initial sets of rules for subsequent improvement. It can, however, be restrictive to structure a rule system primarily to deal with phonetic generalizations, because further study of imperfections in the rules may in many cases show the theory to be incorrect in detail, so requiring many special cases to be added to the rules.

Table-driven systems have a very large number of table values to be determined, and so at first sight it would seem that preparing rules for this method involves far more work than incorporating rules derived from theory directly in a computer program. However, it is quite practicable to use theory to guide the choice of initial table entries, and the experimenter can then introduce exceptions as they are shown to be necessary merely by modifying selected table entries.

6.5.1 Automatic adjustment of phonetic rules

Another possibility for adjusting phonetic rules is to prepare a fairly large corpus of phonetically transcribed speech data. The same phonetic sequence can then be generated by the current set of rules, and the rules successively modified to achieve the optimal fit to the natural data as measured by some suitable objective distance criterion. One option is to analyse the natural speech in terms of the parameters that are calculated by the rules (e.g. formant frequencies and amplitudes) and compare these parameters with the rule-generated ones. An alternative, which does not rely on analysing the natural speech in this way, is to make a direct comparison between the spectra of natural speech and rule-generated synthetic speech.

These methods are potentially very powerful, as a completely automatic process could be envisaged to optimize the parameters of a rule system to match natural speech data. The idea has some similarities to the automatic determination of waveform segments for concatenative synthesis which was described in Chapter 5. However, the situation is more complex for synthesis by rule because, rather than simply identifying segments to be used for synthesis, the task here is to optimize the *parameters* of a synthesis system. Although there are many issues that need to be addressed to develop such a system, some of the techniques used in automatic speech recognition are relevant. For example, one obvious problem, of aligning the timescales of the natural and synthesized speech, can be solved by using the method of **dynamic programming** that will be described in Chapter 8.

A major difficulty for the automatic optimization of phonetic rule systems is deciding in advance how many different allophones will be needed to achieve good synthesis for each phoneme. It should be possible to automate the choice of allophones, again based on a measure of distance between rule-generated synthetic speech and the natural data. One option would be simply to allocate a new allophone for a particular phonemic environment whenever it is found that rule-generated utterances match the natural data less well for one environment compared with others. Another possibility would be to start with many more allophones, optimize the parameters of the rules for each allophone, and then combine those allophones that are similar into a single allophone. The issue of automatically determining allophones will be considered again in Chapter 12, for the task of selecting models to use for large-vocabulary speech recognition.

6.5.2 Rules for different speaker types

In any practical synthesis system the user may want many different types of voice for different occasions, and in particular may wish to switch from male to female or vice versa. For either concatenative or rule-based methods, one option would be to set up the system from scratch each time a new voice is required, which is easiest if an automatic procedure is available.

Another option for a rule-based formant synthesis system is to change the voice quality by applying some transformation to the parameters of the rule system. An extreme would be to convert a system for male speech into one for female speech. Although the phonetic descriptions of male and female speech for the same accent are very similar, their acoustic realizations are quite different. The fundamental frequency of female speech is normally about an octave higher than for male. Because of a shorter vocal tract, the formant frequencies are also higher, usually by about 20%. The different dimensions of the vocal folds in a female larynx also cause the voiced excitation spectrum to be different in female speech, with far less power at the frequencies of the higher formants.

It seems almost certain that most of the effects mentioned above are systematic, so that transformations could be devised for converting rule systems between male and female speech. However, attempts so far to generate acceptable voice quality of female speech from male rules have had only limited success. It has been suggested by various workers that at least a part of the speech differences between the sexes is socially conditioned, in that the two sexes actually learn different styles of speech. If this effect is significant it could account for some of the difficulty in devising rule transformations between the sexes.

6.5.3 Incorporating intensity rules

A large proportion of the intensity variation between phones depends merely on the identity of the phone being considered. For example, voiceless fricatives are fairly weak, and most vowels are quite strong. Those variations will be incorporated in the rules for generating the phones, and therefore do not require to be specified in the input to the acoustic-phonetic system. Intensity specifications could, however, be desirable as an optional modification to the default intensity for each individual allophone.

There are three obvious classes of intensity variation that may be desirable. First, for some applications it may be required to vary the overall loudness in a way that gives the impression of variation of vocal effort by the synthetic talker. This variation does not have the same effect as applying an overall scale change to the output waveform, because the spectral balance of the excitation changes with vocal effort. Loud speech has relatively more power in the higher-frequency region. The second factor affecting intensity is stress. Stressed syllables in a sentence are normally a little louder than unstressed syllables, although a part of this increase is simply a consequence of the increase in voiced excitation power that automatically arises from the pitch increase often associated with stressed syllables. The third cause of intensity change is the result of the lowering of vocal effort that normally occurs towards the end of each breath group.

In general, the above intensity variations only involve changes of a few dB from phone to phone, and most current synthesis systems completely ignore them without serious damage to the output quality. However, the small intensity changes that might be desirable could easily be accomplished by additional rules to adjust the appropriate parameters within a formant synthesis-by-rule system, provided that the syllable stress pattern and position in the breath group were known. This information could be provided by higher-level components of a synthesis system, as described in the next chapter.

6.6 CURRENT CAPABILITIES OF PHONETIC SYNTHESIS BY RULE

The first synthesis-by-rule programs for synthesizing speech from a phonemic representation were written in the early 1960s, and many different systems were developed during the late 1960s and early 1970s. By the 1980s, there were several text-to-speech systems that used synthesis by rule for the acoustic-phonetic component, often following years of careful research to refine the rules. The most well-known of these systems is MITalk (Allen *et al*., 1987), which formed the basis for Digital Equipment Corporation's commercial text-to-speech system, DECtalk. This system has demonstrated a variety of different voice qualities, including men, women and children. At present, the best text-to-speech systems using phonetic synthesis by rule produce speech which is very intelligible for much of the time, but which does not sound natural and has a ‘machine-like’ quality.

More recently, with the advent of PSOLA and low-cost computer memory, phonetic synthesis by rule has been largely abandoned for text-to-speech systems in favour of waveform-based concatenative techniques, which currently give more natural-sounding synthetic speech. However, formant synthesis by rule has important advantages in its inherently smooth model of co-articulation, and also in the flexibility to easily incorporate effects due to changes in speaking rate, voice quality, vocal effort and so on, by applying appropriate transformations to just the relevant controls. Although this flexibility is shared to some degree by parametric concatenative methods, it can be achieved in a more disciplined way with rule-driven synthesis. Techniques for automatic optimization using natural speech data may offer the opportunity for much higher-quality formant synthesis by rule to be achieved in the future. Related issues will be discussed further in Chapter 16.

CHAPTER 6 SUMMARY

- Phonetic synthesis by rule involves applying acoustic-phonetic rules to generate synthesizer control parameters from a description of an utterance in terms of a sequence of phonetic segments together with prosodic information.
- Most acoustic-phonetic rule systems are designed for a formant synthesizer.
- A convenient implementation is to store the rules as tables of numbers for use by a single computational procedure.
- Typically, a table for each phone holds some target synthesizer control values, together with transition durations and information used to calculate the controls at the nominal boundary between any pair of phones. Such a system can capture much of the co-articulation effects between phones.
- Separate tables can be included for any allophonic variation which is not captured by the co-articulation rules. The total number of different units will still be far fewer than the number required in a concatenative system.
- Acoustic-phonetic rule systems have tended to be set up ‘by hand’, but automatic procedures can be used to derive the parameters of these systems, based on optimizing the match of the synthesized speech to phonetically transcribed natural speech data.

CHAPTER 6 EXERCISES

E6.1 Discuss the benefits and difficulties that arise from using articulatory rules for speech synthesis.

E6.2 In view of the very large number of table entries that must be provided, why are table-driven phonetic rules practically convenient?

E6.3 How can allophonic variation be provided for in acoustic-phonetic synthesis rules?

E6.4 Explain the concept of ‘rank’ in the Holmes-Mattingly-Shearme synthesis technique.

CHAPTER 7

Speech Synthesis from Textual or Conceptual Input

7.1 INTRODUCTION

The previous two chapters have described two different methods for generating an acoustic waveform from an input phoneme sequence together with prosodic information. Either of the methods can form one component of a more general speech synthesis system in which the input is at some higher level, which may be orthographic text or even concepts that are somehow represented in the machine.

7.2 EMULATING THE HUMAN SPEAKING PROCESS

When human beings speak, many factors control how the acoustic output is related to the linguistic content of their utterances. At one level, there are constraints determined by the physiology of their vocal apparatus. Although the physiology is generally similar between people, there are also clear differences of detail, partly related to age and sex, but also caused by genetic differences between individuals.

For a given vocal system, the speech depends on the sequence of muscular actions that control the articulatory gestures. These gestures are learnt from early childhood, and their details are determined partly by the properties of the inherited central nervous system, but also very much by the speech environment in which the child grows up. The latter feature is entirely responsible for determining the inventory of available phonetic productions of any individual, which is closely tied to his/her native language. At a higher level, the relationship between the ideas to be expressed by the choice of words, with their pitch, intensity and timing, is entirely determined by the language.

In acquiring competence in speech the human has two forms of feedback. On the one hand, auditory self-monitoring is paramount for comparing the acoustic patterns produced with those heard as model utterances. The second main form of feedback is the response by other human beings to imperfect utterances produced during language acquisition. Once the right types of utterances can be produced and the necessary gestures have been learnt, kinaesthetic feedback can be used for detailed control of articulatory positions, and can ensure continuation of competent speech even if auditory feedback is not available for any reason.

All the above aspects of speech acquisition imply that the human develops a set of rules at many different levels, to convert concepts to speech. Although some parts of these rules are determined by inherited physiology and some by learning from the environment, it is not easy to separate these two aspects. However, it is clear that there must be a set of rules to guide humans generating speech, although in many cases the utterances will be modified by chance or by creative variation within the limits of what is acceptable to retain the desired effect on the listeners. To embody the complete process of human speaking, these rules must be fantastically complicated—particularly in the linguistic process of expressing subtle shades of meaning by choice of words and prosody.

The aim for computer speech synthesis from either textual or conceptual input is to imitate the characteristics of the typical human speaking process well enough to produce synthetic speech that is acceptable to human listeners. **Synthesis from text** should be able to apply the rules used by a good reader in interpreting written text and producing speech. In its most advanced form such a system should be able to apply semantic interpretation, so that the manner of speaking appropriate for the text can be conveyed where this is not immediately obvious from the short-span word sequences alone. **Synthesis from concept** poses rather different challenges, as the computer will already have some representation of the meaning to be conveyed, but an appropriate sequence of words must be generated for the required concepts before the words can be further converted into their acoustic realisation. Most work on speech synthesis has concentrated on **text-to-speech (TTS)** conversion, and TTS will form the main focus for this chapter, although synthesis from concept will be mentioned briefly later in the chapter.

7.3 CONVERTING FROM TEXT TO SPEECH

The generation of synthetic speech from text is often characterized as a two-stage **analysis-synthesis** process, as illustrated in Figure 7.1. The first part of this process involves analysis of the text to determine underlying linguistic structure. This abstract linguistic description will include a phoneme sequence and any other information, such as stress pattern and syntactic structure, which may influence the way in which the text should be spoken. The second part of the TTS conversion process generates synthetic speech from the linguistic description. This synthesis stage can be further subdivided into prosody generation followed by generation of a synthetic speech waveform from the phonemic and prosodic specifications.

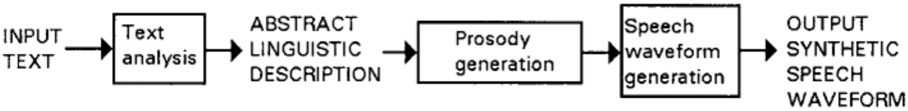

INPUT TEXT
Text analysis
ABSTRACT LINGUISTIC DESCRIPTION
Prosody generation
Speech waveform generation
OUTPUT SYNTHETIC SPEECH WAVEFORM

Figure 7.1 The conversion from text to speech as an analysis-synthesis process.

7.3.1 TTS system architecture

Both the analysis and synthesis processes of TTS conversion involve a number of processing operations, and most modern TTS systems incorporate these different operations within a modular architecture such as the one illustrated in Figure 7.2. When text is input to the system, each of the modules takes some input related to the text, which may need to be generated by other modules in the system, and generates some output which can then be used by further modules, until the final synthetic speech waveform is generated. However, all information within the

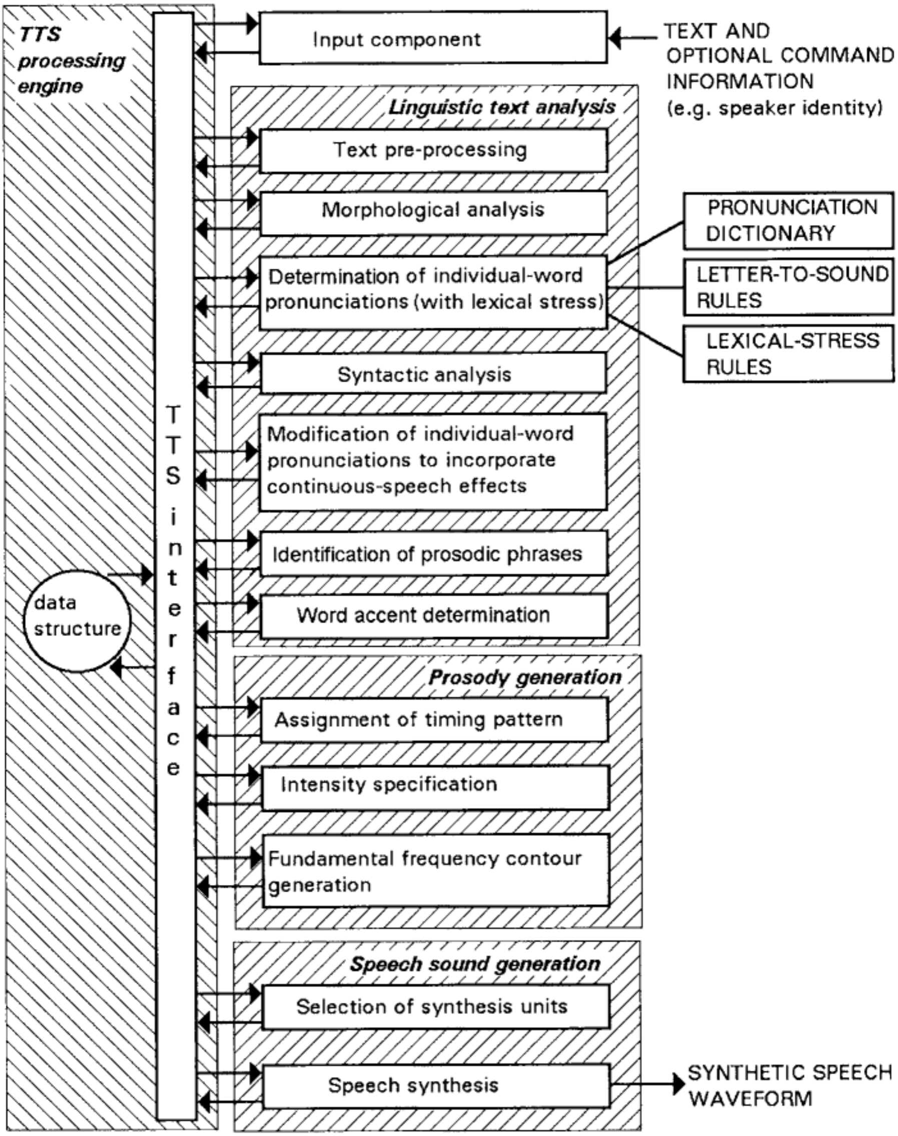

TTS processing engine
TTS interface
data structure
Input component
TEXT AND OPTIONAL COMMAND INFORMATION (e.g. speaker identity)
Linguistic text analysis
Text pre-processing
Morphological analysis
Determination of individual-word pronunciations (with lexical stress)
PRONUNCIATION DICTIONARY
LETTER-TO-SOUND RULES
LEXICAL-STRESS RULES
Syntactic analysis
Modification of individual-word pronunciations to incorporate continuous-speech effects
Identification of prosodic phrases
Word accent determination
Prosody generation
Assignment of timing pattern
Intensity specification
Fundamental frequency contour generation
Speech sound generation
Selection of synthesis units
Speech synthesis
SYNTHETIC SPEECH WAVEFORM

Figure 7.2 Block diagram showing a modular TTS system architecture with typical modules for performing text analysis and speech generation operations.

system passes from one module to another via a separate processing 'engine' and the modules do not communicate directly with each other. The processing engine controls the sequence of operations to be performed, stores all the information in a suitable data structure and deals with the interfaces required to the individual modules. A major advantage of this type of architecture is the ease with which individual modules can be changed or new modules added. The only changes that are required are in the accessing of the modules in the TTS processing engine; the operation of the individual modules is not affected. In addition, data required by the system (such as a **pronunciation dictionary** to specify how words are to be pronounced) tend to be separated from the processing operations that act on the data. This structure has the advantage that it is relatively straightforward to tailor a general TTS system to a specific application or to a particular accent, or even to a new language. There is a growing interest in **multilingual** TTS synthesis, whereby the aim is to use the same TTS system for synthesis in a range of languages, just by changing language-specific data and possibly varying a few of the modules. Our description will concentrate mainly on English, but the overall design and many of the techniques for the individual modules are also applicable to other languages.

7.3.2 Overview of tasks required for TTS conversion

This section will give an overview of typical TTS tasks (as shown in Figure 7.2), together with some explanation of why they are needed and how they are used in the TTS conversion process. The aim of this section is to provide background for the more detailed description of individual tasks that will be given in later sections.

Linguistic text analysis

Text consists of alphanumeric characters, blank spaces and possibly a variety of special characters. The first step in text analysis usually involves pre-processing the input text (including expanding numerals, abbreviations etc.) to convert it to a sequence of words. The pre-processing stage will normally also detect and record instances of punctuation and other relevant formatting information such as paragraph breaks. The following text analysis modules then convert the sequence of words into a linguistic description. A major function of these modules is to determine the pronunciation of the individual words. In a language such as English the relationship between the spellings of words and their phonemic transcriptions is extremely complicated. Furthermore, this relationship can be different for different words with the same structure, as is illustrated by the pronunciation of the letter string “ough” in the words “through”, “though”, “bough”, “rough” and “cough”.

Word pronunciation is normally obtained using some combination of a pronunciation dictionary and **letter-to-sound rules**. In early TTS systems the emphasis was on deriving pronunciation by rule and using a small **exceptions dictionary** for common words with irregular pronunciation (such as “one”, “two”, “said”, etc.). However, now that large computer memory is available at low cost, it is more usual for the main work of the pronunciation task to be accomplished using a very large dictionary (which may include several tens of thousands of words) to ensure that known words are pronounced correctly. Rules are nevertheless still required to deal with unknown words, as for example new words are continually being added to languages and it would be impossible to rely on including all proper nouns (e.g. place names and surnames) in a dictionary. The task of determining word pronunciation is made easier if the structure, or **morphology,** of the words is known, and most TTS systems include some morphological analysis. This analysis determines the 'root form' of each word (for example, the root for "gives" is "give"), and avoids the need to include all derived forms in the dictionary. Some syntactic analysis of the text will also be required to determine the pronunciation of certain words. For example, "live" is pronounced differently depending on whether it is a verb ("they live here") or an adjective ("live wire"). Once pronunciations have been derived for the individual words as if they were spoken in isolation, some adjustments are then needed to incorporate phonetic effects occurring across word boundaries, in order to improve the naturalness of the synthetic speech.

In addition to determining the pronunciation of the word sequence, the text-analysis modules must determine other information relevant to how the text should be spoken. This information, which includes phrasing, lexical (word-level) stress and the pattern of accentuation of the different words (sentence-level stress), will then be used to generate the prosody for the synthesized speech. Markers for lexical stress can be included for each word in the dictionary, but rules will also be needed to assign lexical stress to any words not found in the dictionary. Some words, such as "permit", have their primary stress on a different syllable depending on whether they are being used as a noun or a verb, and so syntactic information will be needed in order to assign the correct stress pattern. The result of a syntactic analysis can also be used to group words into prosodic phrases, and to determine which words are to be accented so that a stress pattern can be assigned to the word sequence. While syntactic structure provides useful clues to accenting and phrasing (and hence prosody), in many cases truly expressive prosody cannot be obtained without really *understanding* the meaning of the text. However, although some simple semantic effects are sometimes incorporated, comprehensive semantic and pragmatic analyses are beyond the capabilities of current TTS systems.

Speech synthesis

Information derived in the text analysis can be used to generate prosody for the utterance, including the timing pattern, overall intensity level and fundamental-frequency (pitch) contour. The final modules in a TTS system perform speech sound generation by first selecting the appropriate synthesis units to use, and then synthesizing from these units together with the prosodic information. Nowadays, this synthesis stage is usually achieved by concatenative techniques (explained in Chapter 5), although an alternative is to use phonetic synthesis by rule (Chapter 6).

As methods for speech generation have already been described in the previous two chapters, they will not be discussed in any detail here. The following sections describe the text analysis and prosody generation stages in more detail.

7.4 TEXT ANALYSIS

7.4.1 Text pre-processing

Text will enter the TTS system as a string of characters in some electronically coded format, which in the case of English would normally be ASCII. The first stage in text analysis is **text segmentation,** whereby the character string is split into manageable chunks, usually sentences with each sentence subdivided into individual words. For a language such as English the separation into words is fairly easy as words are usually delimited by white space. The detection of sentence boundaries is less straightforward. For example, a full stop can usually be interpreted as marking the end of a sentence, but is also used for other functions, such as to mark abbreviations and as a decimal point in numbers.

Any unrestricted input text is likely to include numerals, abbreviations, special symbols such as %, *, etc., capitalization and a variety of punctuation and formatting information (white space, tab characters, etc.). It is therefore usual for the text pre-processing to also include a process of **text normalization,** in which the input text is converted to a sequence of pronounceable words. The normalized text will typically consist of a sequence of explicitly separated words, consisting only of lower-case letters, and with punctuation associated with some of the words. For example, the text "Dr. Smith lives at 16 Castle St." could be converted to:

{[doctor] [smith] [lives] [at] [sixteen] [castle] [street]}

where square brackets have been used to delimit each individual word and curly brackets to delimit the sentence. Each word can be marked with tags to indicate detection of an expanded abbreviation, expanded numerals, capital letters and so on. In this way, all of the information can be passed on but at the same time the text is put into a format which is more suitable for further processing. Most TTS systems include a large number of rules to deal with the variety of text formats that may be encountered, and a few examples are given in the following paragraphs.

In the case of numerals, the correct pronunciation will depend on the context. In many contexts a four-digit number beginning in 1 represents a year and should therefore be pronounced according to the conventions for dates, but in other cases it will be "one thousand" followed by the hundreds, tens and units (e.g. "1999" could be the year "nineteen ninety nine" or "one thousand nine hundred and ninety nine"). Telephone numbers in English are usually pronounced as a sequence of separate digits. A number with two decimal places will be pronounced as a sum of money if preceded by a currency symbol (e.g. "$24.75" becomes "twenty-four dollars and seventy-five cents"), but will otherwise include the word "point" (e.g. "24.75" becomes "twenty-four point seven five".

Conversions for abbreviations and special symbols can be provided in a look-up table. Special symbols are replaced by the relevant words (e.g. "%" is changed to "per cent", and "&" to "and"), and certain abbreviations need to be expanded as appropriate (e.g. "Mr." to "mister", and "etc." to "et cetera"). Some abbreviations are ambiguous and context needs to be taken into account to determine the correct expansion. Commonly cited examples are "Dr.", which can expand to "doctor" or to "drive", and "St.", which can expand to "saint" or to "street". While some abbreviations need to be expanded, others (e.g. "USA", "GMT") must be spelled out and these will be replaced by the appropriate sequence of letter names.

In general, dealing with abbreviations is quite straightforward as long as they are known in advance and have been included in a conversion table. It will, however, be impossible to predict all abbreviations that might occur in any arbitrary text, and so it is usual to include rules for detecting abbreviations. The presence of full stops between the letters can be taken as a good indication that the letter names should be pronounced separately. A word in capitals is also likely to be an abbreviation, at least if the surrounding words are in lower case. If the sequence of letters forms a pronounceable word, it is probably an acronym (e.g. "NATO") and should therefore be treated as a word, but otherwise the abbreviation can be pronounced as a sequence of letter names. However, some pronounceable sequences should also be spelled out as individual letters (e.g. "MIT"). The best strategy is probably to treat abbreviations of four or more letters as words if they are pronounceable. For shorter abbreviations, or if the letter combination is unpronounceable, it is more appropriate to spell out the individual letters.

Text pre-processing rules of the types described above can cope adequately with many text formatting phenomena, but unrestricted text is always likely to contain some formatting features which will be difficult to decode without sophisticated analysis of syntax and even meaning. It may be possible to overcome any ambiguity by delaying decisions that cannot be resolved at a pre-processing stage until the later stages of text analysis. Currently, however, the best results are still obtained if the designer prepares the TTS system for a known restricted range of applications, so that the pre-processing can be tailored appropriately.

7.4.2 Morphological analysis

Morphemes are the minimum meaningful units of language. For example, the word "played" contains two morphemes: "play" and a morpheme to account for the past tense. Morphemes are abstract units which may appear in several forms in the words they affect, so that for example the word "thought" comprises the morpheme "think" together with the same past-tense morpheme as was one used in the previous example. When there is a direct mapping between the abstract morphemes and segments in the textual form of the word, these text segments are referred to as **morphs**. In many words, such as "carrot", the whole word consists of a single morph. Others, such as "lighthouse", have two or more. Morphs can be categorized into **roots** and **affixes,** and the addition of common affixes can vastly increase the number of morphs in a word. For example, "antidisestablishmentarianism" has six morphs if "establish" is regarded as a single root morph. A high proportion of words in languages such as English can be combined with prefixes and/or suffixes to form other words, but the pronunciations of the derived forms are closely related to the pronunciations of the root words.

Rules can be devised to correctly decompose the majority of words (generally at least 95% of words in typical texts) into their constituent morphs. This morphological analysis is a useful early step in TTS conversion for several reasons:

- It is then not necessary for all derived forms of regularly inflected words to be stored in the pronunciation dictionary. Instead, the pronunciation of any derived word can be determined from the pronunciation of the root morphs together with the normal pronunciations of the affixes. For example, inclusion of the word "prove" would enable the correct pronunciation of "improvement", "proving", etc. to be determined. (Note that it is necessary to take account of the fact that in many words a final "e" needs to be removed before the addition of certain suffixes.)
- If the pronunciation of individual morphs is known, it is possible to deal with the many compound words of English and cover a high proportion of the total vocabulary while keeping the dictionary at a manageable size. A lexicon of N morphs can generate between 5*N* and 10*N* words. Complete words need then only be included in

the dictionary if they do not follow the regular morpheme composition rules of the language. A morph lexicon is also useful in predicting the pronunciation of unknown words. While the words in a language are continually changing, it is rare for a new morpheme to enter a language.

- Even when it is necessary to apply letter-to-sound rules (see Section 7.4.3), some attempt to locate morph boundaries is beneficial as many of the rules for the pronunciation of consonant clusters do not apply across morph boundaries. For example, the usual pronunciation of the letter sequence "th" does not apply in the word "hothouse", due to the position of the morph boundary.
- Morphological analysis gives information about attributes such as syntactic category, number, case, gender (in the case of some languages) and so on. This information is useful for later **syntactic analysis** (see Section 7.4.4).

Extensive use of morphological analysis and morph dictionaries was pioneered in the MITalk system (Allen *et al.*, 1987), which covered over 100,000 English words with a morph lexicon of about 12,000 entries and hence moderate cost in terms of storage. While storage cost is no longer such an issue, the other advantages of morph decomposition are such that the better TTS systems all include at least some morphological analysis.

7.4.3 Phonetic transcription

It is usual for the task of determining the pronunciation of a text to begin by assigning an idealized phonemic transcription to each of the words individually. Nowadays, most TTS systems use a large dictionary. This dictionary will generally only contain the root forms of words and not their morphological derivatives, except for those derivatives that cannot be correctly predicted by rule. For words with alternative pronunciations, both possibilities can be offered by the dictionary and syntactic analysis may be used to choose between them (see Section 7.4.4).

A typical strategy for determining word pronunciation is to start by searching the dictionary to check whether the complete word is included. If it is not, the component morphs can be searched for. Provided that the individual morphs are in the dictionary, the pronunciation of the derived word can then be determined by rule from the pronunciation of its component morphs. When deriving pronunciations of derived words from their root form, it is necessary to take into account any pronunciation-modification rules associated with the affixes. For example, the suffix "ion" changes the phonemic interpretation of the final /t/ sound in words like "create".

For any words (or component morphs) whose pronunciation cannot be determined using the dictionary, letter-to-sound rules are needed. The complexity of the relationship between the spellings of words and their phonemic transcription is different for different languages. However, even in a language such as English which has a particularly complicated mapping between letters and phonemes, it is obvious that human readers must have some rules for relating spelling to phoneme sequences because they can usually make a reasonable guess at the pronunciation of an unfamiliar word. While it cannot be guaranteed that letter-to-sound rules will always give the pronunciation that most people would regard as correct, a human reader will also often make errors with unfamiliar words. However, these words are usually quite rare and the nature of any errors tends to be such that the incorrect phoneme sequence is often sufficient to indicate the intended word. Predicting the pronunciation of proper names is especially challenging, as names often follow quite different pronunciation rules from ordinary words and may be from many different languages. Many TTS systems include special rules for names, sometimes using a scheme based on analogy with known names (e.g. the pronunciation of “Plotsky” can be predicted by analogy with the pronunciation of “Trotsky”).

In naturally spoken continuous speech, word pronunciations are influenced by the identities of the surrounding words. TTS systems incorporate these effects by applying **post-lexical rules** to make phonetic adjustments to the individual-word phonemic transcriptions. For example, the correct pronunciation of the vowel in the word “the” depends on whether the following word begins with a vowel (e.g. “the apple”) or a consonant (e.g. “the dog”). Other effects on pronunciation are related to the consequences of co-articulation and the preferred option may depend on the speaking style. For example, the consonant sequence in the middle of “handbag” may be pronounced [ndb] in highly articulated speech, but would more usually be reduced to [nb], and may even become [mb] in casual speech.

7.4.4 Syntactic analysis and prosodic phrasing

Some syntactic analysis is needed both to resolve pronunciation ambiguities and to determine how the utterance should be structured into phrases. Possible syntactic classes can be included with each entry in the dictionary, and the morphological analysis will also provide useful information about likely parts of speech. However, very many English words may be used as both nouns and verbs, and several can also be adjectives, so very little definite information about syntax can be resolved without taking into account the relationships between the words.

Assignment of syntactic classes, or **part-of-speech tags,** is often achieved using a statistical model of language, based both on probabilities for particular tags appearing in a certain context and on probabilities of the tags being associated with the given words. The model probabilities can be derived from large amounts of correctly marked text, and the modelling technique itself is one that is widely used for language modelling in automatic speech recognition (see Chapter 12).

Once the part of speech has been decided for each word in a sentence, the phrase structure of the sentence can be determined. In order for suitable prosody to be generated, it is necessary to decide on sentence type (declarative, imperative or question), and to identify phrases and clauses. Some systems have included full syntactic parsing, while others perform a more superficial syntactic analysis, for example to locate noun phrases and verb phrases and possibly group these phrases into clauses. There are also methods for using statistical models trained to predict prosodic phrases directly from information about parts of speech, stress, position in the sentence and other relevant factors. The general aim is to produce a reasonable analysis for *any* text, even if the text contains syntactic errors. There will always be instances for which correct assignment of appropriate phrasing cannot be achieved without incorporating semantic and pragmatic constraints, but current TTS systems do not have more than very limited capability to apply such constraints.

7.4.5 Assignment of lexical stress and pattern of word accents

In the case of polysyllabic words, there is normally one syllable that is given **primary stress,** and other syllables are either unstressed, or carry a less prominent **secondary stress**. These lexical stress markings can be included for each entry in the dictionary. When the pronunciation of a word is obtained by combining morphs, the stress pattern for the individual morphs may be changed, so it is necessary to apply rules to determine the stress pattern for the complete word. For example, the addition of the suffix "ity" to "electric" moves the primary stress from the second syllable to the third. For some words, such as "permit", the stress assignment depends on syntactic category, so the choice between alternative stress patterns must be made following the syntactic analysis.

With any word whose pronunciation has to be obtained by letter-to-sound rules, additional rules are also needed to assign lexical stress. For many polysyllabic words of English the placement of primary and secondary stresses on the syllables can be determined reasonably accurately using very complicated rules that depend on how many vowels there are in the word, how many consonants follow each vowel, the vowel lengths, etc. There are, however, many words for which the normal rules do not apply, as exemplified by the fact that some pairs of words are of similar structure yet are stressed differently. Examples are "Canada" and "camera", contrasting with "Granada" and "banana". Words such as these will need to be included in the dictionary to ensure correct lexical stress assignment.

One of the last tasks in text analysis is to assign sentence-level stress to the utterance, whereby different words in a sentence are accented to different extents. Assignment of accents depends on a number of factors. **Function words** (such as articles, conjunctions, prepositions and auxiliary verbs) serve to indicate the relationships between the **content words** that carry the main information content of an utterance. Function words are not normally accented, whereas content words tend to be accented to varying degrees dependent on factors such as parts of speech and the phrase structure. In addition to the syntax-driven placement of stress, emphasis may be placed on important words in the sentence. For example, when a speaker wishes to emphasize his or her attitude towards the truth of something, words such as "surely", "might" and "not" may be used with stress. Stress may also be used to make a distinction between new and old information, or to emphasize a contrast. Some TTS systems include rules to model a number of these types of effects. The pattern of accents on the different words will usually be realized as movements in fundamental frequency, often referred to as **pitch accents** (see Section 7.5.2 for further discussion).

7.5 PROSODY GENERATION

The acoustic correlates of prosody are intensity, timing pattern and fundamental frequency. Intensity is mainly determined by phone identity, although it also varies with stress for example. It is fairly easy to include rules to simulate such effects, as was discussed in Section 6.5.3. From the perspective of prosody, intensity variations are in general less influential than variations in timing pattern and in fundamental frequency contour, which are discussed in the following sections.

7.5.1 Timing pattern

Both in concatenative synthesis and in most synthesis-by-rule methods, utterances are generated as sequences of speech segments. For any utterance, a duration needs to be chosen for each segment such that the synthesized speech mimics the temporal structure of typical human utterances. The temporal structure of human speech is influenced by a wide variety of factors which cause the durations of speech segments to vary. Observations about this variability include the following:

1. The inherent durations of different speech sounds differ considerably. Some vowels are intrinsically short and others long. The vowels in the words “bit” and “beet” in English differ in this way. Diphthongs are usually longer than monophthongs, and consonant sounds also show systematic differences.
2. Durations differ according to speed of speaking, but sounds that are mainly steady in character, such as fricatives and vowels, tend to vary in duration more than inherently transient sounds, such as the bursts of stop consonants.
3. If a particular word in a sentence is emphasized, its most prominent syllable is normally lengthened.
4. Durations of phones vary according to their position in a word, particularly if there are several syllables.
5. When at the end of a phrase, a syllable tends to be longer than when the same syllable occurs in other locations in a phrase.
6. Vowels before voiced consonants are normally longer than occurrences of the same vowels before unvoiced consonants. For example, in the English words “feed” and “feet” the vowel is substantially longer in “feed”. There are also other systematic duration modifications that depend on the identities of neighbouring phones.
7. Some evidence suggests that, in a ‘stress-timed’ language such as English, unstressed syllables tend to be shorter if there are several of them between two stressed syllables. However, the empirical evidence is less conclusive for this effect than for the other effects listed above.

A number of systems have been developed for deriving segment durations by applying a succession of rules. These rules operate on phonetic transcriptions with the stressed syllables marked, and assume that some decision has been made about speed of speaking. It is then possible to estimate a suitable duration for each phone by having some **intrinsic duration** for the phone, and to modify it by various amounts according to each of the circumstances mentioned above. The amount of the modification could in general depend on the circumstances causing it and on the identity of the phone whose duration is being calculated. Sets of rules have been devised and refined based on phonetic knowledge in combination with statistics of speech segment durations and the results of small-scale experiments investigating the effect of varying different factors on synthesis quality. While reasonable success has been achieved in producing acceptable timing patterns, this approach is not able to guarantee that the rules are optimized simultaneously to the very wide range of utterances that general TTS systems must be able to deal with.

In recent years, as large speech corpora and increased computational resources have become available, there has been a growth in alternative approaches using automatic optimization to derive the parameters of a general model based on large databases of segmented and labelled speech. It is quite straightforward to apply these data-driven methods to derive a reasonable duration model for a new language, provided that sufficient labelled speech data are available.

Automatic methods have achieved some improvement over the older rule-based systems. However, current TTS systems are still not able to produce the rhythm that humans can adopt naturally in sentences containing rhyming clauses, or to generate other systematic variations related to meaning. Speech synthesized from text also lacks the pattern of pauses and decelerations that are found in speech from a good human reader, and which serve to enhance a listener's comprehension. More elaborate linguistic analysis would be necessary to produce all these effects.

7.5.2 Fundamental frequency contour

The fundamental frequency of voiced speech, which determines the perceived pitch, is widely used by all languages to convey information that supplements the sequence of phonemes. In some languages, such as Chinese, pitch changes are used to distinguish different meanings for syllables that are phonetically similar. In most Western languages pitch does not help directly in identifying words, but provides additional information, such as which words in a sentence are most prominent, whether a sentence is a question, statement or command, the mood of the speaker, etc. Even for these Western languages, the type of intonation pattern that is used to achieve particular effects varies considerably from one language to another, and even between accents of the same language. Obviously the model for generating a suitable intonation pattern must be developed to suit the required language. For the purposes of this book, examples will be given for typical southern British English.

Most sentences in English show a general tendency for pitch to fall gradually from beginning to end of each sentence, but with many local variations around this trend. Two major factors determining these variations are the way in which the sentence is subdivided into phrases and the sentence stress pattern. The most significant pitch variations occur at major phrase boundaries and on words that the user wishes to be more prominent. In the case of polysyllabic words, the syllable with primary stress carries the main pitch movement.

The normal structure of English is such that the last syllable carrying primary stress in any breath group is given the biggest pitch change, and is known as the **nuclear syllable**. Usually the **nuclear tone** (i.e. the pitch pattern on the nuclear syllable) on a simple statement is a pitch fall, but a number of other patterns are possible to indicate other types of utterance. (The number of possible nuclear tones is at least three, but some workers have claimed that there are up to six significantly different patterns.) The nuclear tone for a question expecting a yes/no answer shows a substantial pitch rise. On the non-final stressed syllables the pitch usually shows a local small rise and then continues its steady fall. The amount of this rise and the subsequent rate of fall can depend on the syntactic function of the word in the sentence: verbs, for example, generally have less pitch variation than nouns and adjectives. At the beginning of an utterance the pitch often starts fairly low, and then rises to a high value on the first stressed syllable.

In addition to these pitch changes caused by the pattern of stressed syllables, there are smaller pitch variations that are influenced by the phonetic detail of an

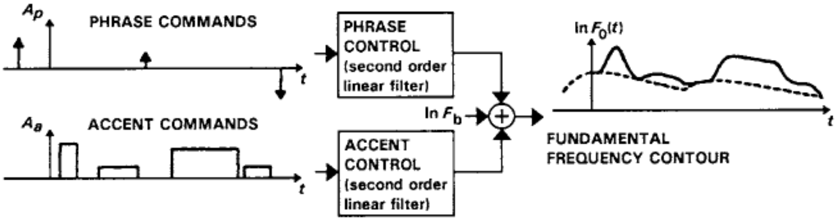

A_p
PHRASE COMMANDS
t
PHRASE CONTROL (second order linear filter)
$\ln F_b$
A_a
ACCENT COMMANDS
t
ACCENT CONTROL (second order linear filter)
$\ln F_0(t)$
t
FUNDAMENTAL FREQUENCY CONTOUR

Figure 7.3. Generating a fundamental frequency *(F_0)* contour as a filtered sequence of phrase commands and accent commands, combined with a baseline F_0 value *(F_b)*. The generated F_0 contour is indicated by the solid line in the right-hand graph. The dotted line shows the contour given by the phrase commands alone (adapted from Fujisaki and Ohno (1995)).

utterance. When voicing restarts after a voiceless consonant there is a tendency for the pitch to be a little higher for a few tens of milliseconds than it would be after a voiced consonant. Also, due to muscular interactions between the articulators and the larynx, some vowels tend to have intrinsically higher pitch than others.

Various models have been proposed to generate fundamental frequency *(F_0)* characteristics of the type described above. There are differences between some of the models that are related to differences between the theories of intonation on which they are based. However a general characteristic of all the models is that they operate in two stages. The first stage generates an abstract description of an intonation contour (which will include some expression of pitch accents), and the second stage converts from the abstract description into a sequence of F_0 values.

Superposition models are hierarchically organized and generate F_0 contours by overlaying multiple components of different types. An example of this approach to generating intonation is shown in Figure 7.3. This model distinguishes between phrase commands and accent commands. The commands are discrete events, represented as pulses for the phrase commands and step functions for the accent commands. The F_0 contour is obtained by filtering each sequence of commands and combining the output of the two filters, superimposed on a baseline F_0 value.

In contrast, **tone sequence models** generate an F_0 contour from a sequence of discrete tones that are locally determined and do not interact with each other. One especially influential model was developed by Pierrehumbert (1980). Here a tone is defined as being either high or low, and of a different type, depending on whether it is associated with a pitch accent, a phrase boundary or an intermediate position between a pitch accent and a boundary tone. To use the model for synthesis, Pierrehumbert (1981) defined a time-varying F_0 range and used rules to assign a high or low target tone within that range to each stressed syllable. An F_0 contour was generated by applying a quadratic function to interpolate between successive targets, as shown in Figure 7.4. Pierrehumbert's approach to labelling intonation has formed the basis for the intonation transcription system called **TOBI (TOnes and Break Indices),** which was proposed by Silverman *et al.* (1992). There are speech databases transcribed according to the TOBI system, which have facilitated the development of various automatic methods for training models both to predict abstract TOBI labels from information such as stress pattern and phrase structure, and for generating an F_0 contour once TOBI symbols are available.

Although some research groups have demonstrated very natural-sounding utterances using intonation generated by models such as the ones described above,

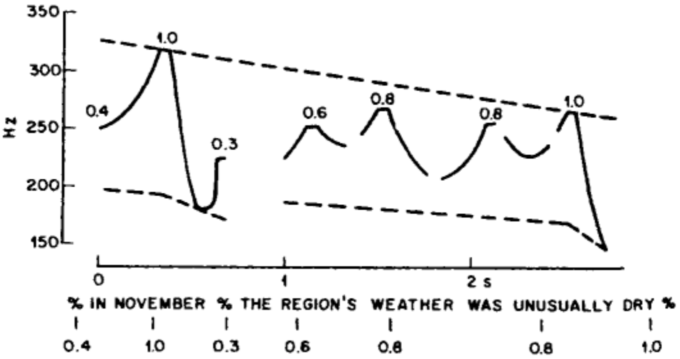


Figure 7.4. Generating a fundamental frequency *(F_0)* contour by interpolating between targets. The dashed lines indicate the F_0 range, and the numbers represent local target F_0 values, expressed as a proportion of the current range. The targets are located on the stressed syllables of content words and at phrase boundaries (indicated by % in the text). (Reprinted with permission from Pierrehumbert, 1981. Copyright © 1981, Acoustical Society of America.)

to achieve these results the input to the model has needed a very detailed specification of the utterance structure. Complete TTS systems do not generally produce such natural-sounding speech, and the intonation sounds less 'interesting' than one would expect from a human talker. It seems likely that a major limitation is the difficulty of making a sufficiently accurate linguistic analysis of the text to provide the appropriate information as input to the intonation prediction model.

7.6 IMPLEMENTATION ISSUES

The pronunciation dictionary can account for a substantial proportion of the total memory requirements. The dictionary might require an average of around 30 bytes per word, so a modest 10,000-word dictionary would need 300 kbytes of memory, and a 100,000-word dictionary would involve 3 Mbytes. The memory requirements for the synthesis component are different for different synthesis methods. Synthesis by rule tends to be the most efficient in terms of memory: for example, the table-driven system described in Chapter 6 could provide a useful set of allophone tables using less than 50 kbytes. Much more memory may be needed with concatenative synthesis (see Section 5.7), and the waveform-based methods can require several Mbytes when using a large inventory of synthesis units. The total memory usage for systems giving the best quality may even be as high as around 100 Mbytes.

The processing requirements will also depend on the synthesis method. Time-domain waveform synthesis obviously involves less computation than LPC or formant synthesis. However, although a large amount of arithmetic would be needed for the signal processing in LPC or formant synthesis, the calculations are simple and can easily be handled by current microprocessors. Alternatively, DSP chips may be used for some or all of the calculations.

The higher-level processing may be quite complicated, but the input and output data rates are very low (only about 20 symbols per second for each stage of the processing).

Semantic processing is still insufficiently developed for it to be possible to predict the computational load needed, but if semantics are excluded fairly complex high-level processing can be implemented serially in real time.

Currently available TTS systems include software-only versions for personal computers, small stand-alone devices and plug-in processing boards. The best of present-day capabilities for TTS can be implemented on a single processing card of appropriate design, generally including software, memory (which may be quite large), DSP chips and a controlling microprocessor.

7.7 CURRENT TTS SYNTHESIS CAPABILITIES

For several years now there have been many TTS systems working in research laboratories, and also various commercial products. Often a limited choice between different voices is included. There are systems for a variety of different languages, and some systems are multilingual. As explained in Chapters 5 and 6, the majority of current TTS systems use concatenative waveform synthesis for the speech generation component, as it has not yet been possible to achieve the same degree of naturalness using phonetic synthesis by rule.

As with speech coders, speech synthesizers need to be evaluated in terms of both intelligibility and naturalness, and the techniques described in Section 4.5 are also applicable to assessing the capabilities of TTS systems. The best systems produce speech which is highly intelligible and quite natural-sounding for straightforward material with a simple linguistic content. However, even the best examples of speech from TTS systems are unlikely to be mistaken for natural speech. On more than just a very short utterance the synthesized speech quickly becomes very boring to listen to, and the intonation rarely matches the naturalness of a human speaker. On difficult material, especially texts such as poetry or plays which require a special speaking style, both intelligibility and naturalness fall far short of the performance of a good human reader of the same text.

Improvements are still needed at all levels of the TTS synthesis process, but most especially in the synthetic prosody. This aspect in particular is likely to be limited for many years to come by the difficulties of automatically analysing semantics and, most fundamentally, achieving some ‘understanding’ of the text.

7.8 SPEECH SYNTHESIS FROM CONCEPT

As an alternative to generating synthetic speech from text input, in systems for speech synthesis from concept the input is in the form of ideas or concepts that the machine must express in spoken language. For example, in an information-retrieval system, a computer may be required to respond to queries about some stored database of facts (such as sports scores or weather reports). Thus it will be necessary to determine the concepts to be conveyed and convert these into natural language. Because concepts inherently provide semantic information, and the generation of language must include syntactic structure, it should be easier to predict word pronunciation and to determine the most appropriate prosody than when this information has to be extracted from text. However, natural language generation, including the choice of words and sentences that are appropriate for each situation, is a very challenging research topic in its own right and outside the scope of this book. Current systems for synthesis from concept tend to be set up for particular applications, so restrictions can be placed on the language to be generated.

CHAPTER 7 SUMMARY

- Human beings have acquired rules to convert from ideas to speech, and also to read from written text.
- Machine models for the same processes involve several levels, and are usually implemented with a modular architecture to separate the different components.
- At a high level, the input to a synthesis system can be text or concept, but most systems currently only deal with text input.
- Text-to-speech synthesis involves analysis of the text to determine underlying linguistic structure, followed by synthesis from the linguistic information. Synthesis includes generation of prosody and of a synthetic speech waveform.
- The first step in text analysis is pre-processing, including expanding abbreviations, etc., to give a sequence of words with punctuation marked.
- For English it is difficult to determine correct pronunciation and stress pattern from ordinary text. The task is made easier by including a pronunciation dictionary, often with morph decomposition and syntactic analysis.
- Prosody generation involves using information about stress pattern and sentence structure to specify the fundamental frequency pattern and duration of each phone, and possibly also intensity modifications.
- Several Mbytes of storage may be needed for the pronunciation dictionary and for the synthesis units. Because most of the more complicated processing only needs to operate slowly, real-time TTS conversion is practical.
- Current products give speech that is intelligible and reasonably natural on short passages of simple text, but quality still suffers on more difficult texts.

CHAPTER 7 EXERCISES

E7.1 Describe a technique for converting from conventional to phonetic spelling for a language such as English, highlighting any special difficulties.

E7.2 Why is morphological analysis useful in TTS systems?

E7.3 What types of text pre-processing are needed for TTS systems to handle unrestricted text? Why is this processing not always straightforward?

E7.4 Discuss the relative importance of pitch, intensity and duration when generating synthetic prosody.

E7.5 What information in text can be used to determine utterance prosody?

E7.6 Discuss the main influences on processing and memory requirements in TTS systems for English.

E7.7 Why should conventional orthography not be used as an intermediate representation in systems for synthesis from concept?

CHAPTER 8

Introduction to Automatic Speech Recognition: Template Matching

8.1 INTRODUCTION

Much of the early work on **automatic speech recognition** (**ASR**), starting in the 1950s, involved attempting to apply rules based either on acoustic/phonetic knowledge or in many cases on simple *ad hoc* measurements of properties of the speech signal for different types of speech sound. The intention was to decode the signal directly into a sequence of phoneme-like units. These early methods, extensively reviewed by Hyde (1972), achieved very little success. The poor results were mainly because co-articulation causes the acoustic properties of individual phones to vary very widely, and any rule-based hard decisions about phone identity will often be wrong if they use only local information. Once wrong decisions have been made at an early stage, it is extremely difficult to recover from the errors later.

An alternative to rule-based methods is to use **pattern-matching** techniques. Primitive pattern-matching approaches were being investigated at around the same time as the early rule-based methods, but major improvements in speech recognizer performance did not occur until more general pattern-matching techniques were invented. This chapter describes typical methods that were developed for spoken word recognition during the 1970s. Although these methods were widely used in commercial speech recognizers in the 1970s and 1980s, they have now been largely superseded by more powerful methods (to be described in later chapters), which can be understood as a generalization of the simpler pattern-matching techniques introduced here. A thorough understanding of the principles of the first successful pattern-matching methods is thus a valuable introduction to the later techniques.

8.2 GENERAL PRINCIPLES OF PATTERN MATCHING

When a person utters a word, as we saw in Chapter 1, the word can be considered as a sequence of phonemes (the linguistic units) and the phonemes will be realized as phones. Because of inevitable co-articulation, the acoustic patterns associated with individual phones overlap in time, and therefore depend on the identities of their neighbours. Even for a word spoken in isolation, therefore, the acoustic pattern is related in a very complicated way to the word’s linguistic structure.

However, if the same person repeats the same isolated word on separate occasions, the pattern is likely to be generally similar, because the same phonetic relationships will apply. Of course, there will probably also be differences, arising from many causes. For example, the second occurrence might be spoken faster or more slowly; there may be differences in vocal effort; the pitch and its variation during the word could be different; one example may be spoken more precisely than the other, etc. It is obvious that the waveform of separate utterances of the same word may be very different. There are likely to be more similarities between spectrograms because (assuming that a short time-window is used, see Section 2.6), they better illustrate the vocal-tract resonances, which are closely related to the positions of the articulators. But even spectrograms will differ in detail due to the above types of difference, and timescale differences will be particularly obvious.

A well-established approach to ASR is to store in the machine example acoustic patterns (called **templates**) for all the words to be recognized, usually spoken by the person who will subsequently use the machine. Any incoming word can then be compared in turn with all words in the store, and the one that is most similar is assumed to be the correct one. In general none of the templates will match perfectly, so to be successful this technique must rely on the correct word being more similar to its own template than to any of the alternatives.

It is obvious that in some sense the sound pattern of the correct word is likely to be a better match than a wrong word, because it is made by more similar articulatory movements. Exploiting this similarity is, however, critically dependent on how the word patterns are compared, i.e. on how the 'distance' between two word examples is calculated. For example, it would be useless to compare waveforms, because even very similar repetitions of a word will differ appreciably in waveform detail from moment to moment, largely due to the difficulty of repeating the intonation and timing exactly.

It is implicit in the above comments that it must also be possible to identify the start and end points of words that are to be compared.

8.3 DISTANCE METRICS

In this section we will consider the problem of comparing the templates with the incoming speech when we know that corresponding points in time will be associated with similar articulatory events. In effect, we appear to be assuming that the words to be compared are spoken in isolation at exactly the same speed, and that their start and end points can be reliably determined. In practice these assumptions will very rarely be justified, and methods of dealing with the resultant problems will be discussed later in the chapter.

In calculating a distance between two words it is usual to derive a short-term distance that is local to corresponding parts of the words, and to integrate this distance over the entire word duration. Parameters representing the acoustic signal must be derived over some span of time, during which the properties are assumed not to change much. In one such span of time the measurements can be stored as a set of numbers, or **feature vector,** which may be regarded as representing a point in multi-dimensional space. The properties of a whole word can then be described as a succession of feature vectors (often referred to as **frames**), each representing a time slice of, say, 10–20 ms. The integral of the distance between the patterns then reduces to a sum of distances between corresponding pairs of feature vectors. To be useful, the distance must not be sensitive to small differences in intensity between otherwise similar words, and it should not give too much weight to differences in pitch. Those features of the acoustic signal that are determined by the phonetic properties should obviously be given more weight in the distance calculation.

8.3.1 Filter-bank analysis

The most obvious approach in choosing a distance metric which has some of the desirable properties is to use some representation of the short-term power spectrum. It has been explained in Chapter 2 how the short-term spectrum can represent the effects of moving formants, excitation spectrum, etc.

Although in tone languages pitch needs to be taken into account, in Western languages there is normally only slight correlation between pitch variations and the phonetic content of a word. The likely idiosyncratic variations of pitch that will occur from occasion to occasion mean that, except for tone languages, it is normally safer to ignore pitch in whole-word pattern-matching recognizers. Even for tone languages it is probably desirable to analyse pitch variations separately from effects due to the vocal tract configuration. It is best, therefore, to make the bandwidth of the spectral resolution such that it will not resolve the harmonics of the fundamental of voiced speech. Because the excitation periodicity is evident in the amplitude variations of the output from a broadband analysis, it is also necessary to apply some time-smoothing to remove it. Such time-smoothing will also remove most of the fluctuations that result from randomness in turbulent excitation.

At higher frequencies the precise formant positions become less significant, and the resolving power of the ear (**critical bandwidth**—see Chapter 3) is such that detailed spectral information is not available to human listeners at high frequencies. It is therefore permissible to make the spectral analysis less selective, such that the effective filter bandwidth is several times the typical harmonic spacing. The desired analysis can thus be provided by a set of bandpass filters whose bandwidths and

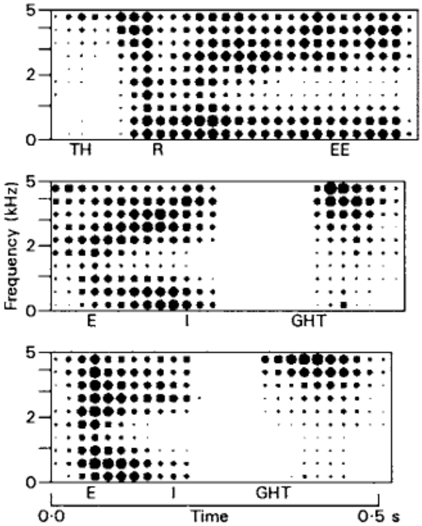


Figure 8.1 Spectrographic displays of a 10-channel filter-bank analysis (with a non-linear frequency spacing of the channels), shown for one example of the word “three” and two examples of the word “eight”. It can be seen that the examples of “eight” are generally similar, although the lower one has a shorter gap for the [t] and a longer burst.

spacings are roughly equal to those of critical bands and whose range of centre frequencies covers the frequencies most important for speech perception (say from 300 Hz up to around 5 kHz). The total number of band-pass filters is therefore not likely to be more than about 20, and successful results have been achieved with as few as 10. When the necessary time-smoothing is included, the feature vector will represent the signal power in the filters averaged over the frame interval.

The usual name for this type of speech analysis is **filter-bank** analysis. Whether it is provided by a bank of discrete filters, implemented in analogue or digital form, or is implemented by sampling the outputs from short-term Fourier transforms, is a matter of engineering convenience. Figure 8.1 displays word patterns from a typical 10-channel filter-bank analyser for two examples of one word and one example of another. It can be seen from the frequency scales that the channels are closer together in the lower-frequency regions.

A consequence of removing the effect of the fundamental frequency and of using filters at least as wide as critical bands is to reduce the amount of information needed to describe a word pattern to much less than is needed for the waveform. Thus storage and computation in the pattern-matching process are much reduced.

8.3.2 Level normalization

Mean speech level normally varies by a few dB over periods of a few seconds, and changes in spacing between the microphone and the speaker's mouth can also cause changes of several dB. As these changes will be of no phonetic significance, it is desirable to minimize their effects on the distance metric. Use of filter-bank power directly gives most weight to more intense regions of the spectrum, where a change of 2 or 3 dB will represent a very large absolute difference. On the other hand, a 3 dB difference in one of the weaker formants might be of similar phonetic significance, but will cause a very small effect on the power. This difficulty can be avoided to a large extent by representing the power logarithmically, so that similar power ratios have the same effect on the distance calculation whether they occur in intense or weak spectral regions. Most of the phonetically unimportant variations discussed above will then have much less weight in the distance calculation than the differences in spectrum level that result from formant movements, etc.

Although comparing levels logarithmically is advantageous, care must be exercised in very low-level sounds, such as weak fricatives or during stop-consonant closures. At these times the logarithm of the level in a channel will depend more on the ambient background noise level than on the speech signal. If the speaker is in a very quiet environment the logarithmic level may suffer quite wide irrelevant variations as a result of breath noise or the rustle of clothing. One way of avoiding this difficulty is to add a small constant to the measured level before taking logarithms. The value of the constant would be chosen to dominate the greatest expected background noise level, but to be small compared with the level usually found during speech.

Differences in vocal effort will mainly have the effect of adding a constant to all components of the log spectrum, rather than changing the shape of the spectrum cross-section. Such differences can be made to have no effect on the distance metric by subtracting the mean of the logarithm of the spectrum level of each frame from all the separate spectrum components for the frame. In practice this amount of level compensation is undesirable because extreme level variations are of some phonetic significance. For example, a substantial part of the acoustic difference between [f] and any vowel is the difference in level, which can be as much as

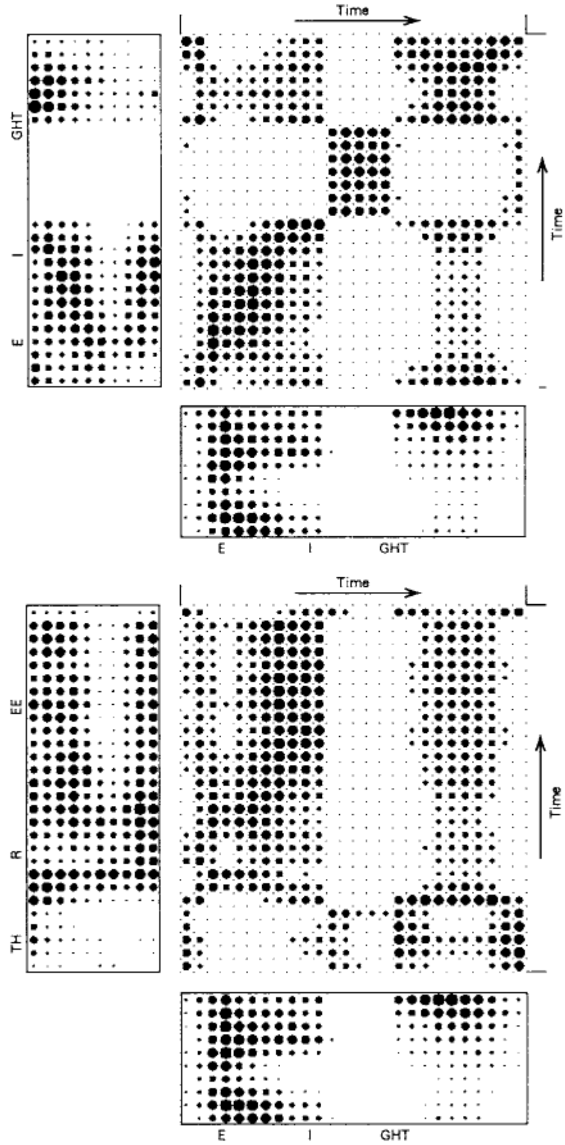


Figure 8.2 Graphical representation of the distance between frames of the spectrograms shown in Figure 8.1. The larger the blob the smaller the distance. It can be seen that there is a continuous path of fairly small distances between the bottom left and top right when the two examples of “eight” are compared, but not when “eight” is compared with “three”.

30 dB. Recognition accuracy might well suffer if level differences of this magnitude were ignored. A useful compromise is to compensate only partly for level variations, by subtracting some fraction (say in the range 0.7 to 0.9) of the mean logarithmic level from each spectral channel. There are also several other techniques for achieving a similar effect.

A suitable distance metric for use with a filter bank is the sum of the squared differences between the logarithms of power levels in corresponding channels (i.e. the square of the **Euclidean distance** in the multi-dimensional space). A graphical representation of the Euclidean distance between frames for the words used in Figure 8.1 is shown in Figure 8.2.

There are many other spectrally based representations of the signal that are more effective than the simple filter bank, and some of these will be described in Chapter 10. The filter-bank method, however, is sufficient to illustrate the pattern-matching principles explained in this chapter.

8.4 END-POINT DETECTION FOR ISOLATED WORDS

The pattern comparison methods described above assume that the beginning and end points of words can be found. In the case of words spoken in isolation in a quiet environment it is possible to use some simple level threshold to determine start and end points. There are, however, problems with this approach when words start or end with a very weak sound, such as [f]. In such cases the distinction in level between the background noise and the start or end of the word may be slight, and so the end points will be very unreliably defined. Even when a word begins and ends in a strong vowel, it is common for speakers to precede the word with slight noises caused by opening the lips, and to follow the word by quite noisy exhalation. If these spurious noises are to be excluded the level threshold will certainly have to be set high enough to also exclude weak unvoiced fricatives. Some improvement in separation of speech from background noise can be obtained if the spectral properties of the noise are also taken into account. However, there is no reliable way of determining whether low-level sounds that might immediately precede or follow a word should be regarded as an essential part of that word without simultaneously determining the identity of the word.

Of course, even when a successful level threshold criterion has been found, it is necessary to take account of the fact that some words can have a period of silence within them. Any words (such as “containing” and “stop”) containing unvoiced stop consonants at some point other than the beginning belong to this category. The level threshold can still be used in such cases, provided the end-of-word decision is delayed by the length of the longest possible stop gap, to make sure that the word has really finished. When isolated words with a final unvoiced stop consonant are used in pattern matching, a more serious problem, particularly for English, is that the stop burst is sometimes, but not always, omitted by the speaker. Even when the end points are correctly determined, the patterns being compared for words which are nominally the same will then often be inherently different.

Although approximate end points can be found for most words, it is apparent from the above comments that they are often not reliable.

8.5 ALLOWING FOR TIMESCALE VARIATIONS

Up to now we have assumed that any words to be compared will be of the same length, and that corresponding times in separate utterances of a word will represent the same phonetic features. In practice speakers vary their speed of speaking, and often do so non-uniformly so that equivalent words of the same total length may differ in the middle. This timescale uncertainty is made worse by the unreliability of end-point detection. It would not be unusual for two patterns of apparently very different length to have the underlying utterances spoken at the same speed, and merely to have a final fricative cut short by the end-point detection algorithm in one case as a result of a slight difference in level.

Some early implementations of isolated-word recognizers tried to compensate for the timescale variation by a uniform time normalization to ensure that all patterns being matched were of the same length. This process is a great improvement over methods such as truncating the longer pattern when it is being compared with a shorter one, but the performance of such machines was undoubtedly limited by differences in timescale. In the 1960s, however, a technique was developed which is capable of matching one word on to another in a way which applies the optimum non-linear timescale distortion to achieve the best match at all points. The mathematical technique used is known as **dynamic programming (DP),** and when applied to simple word matching the process is often referred to as **dynamic time warping (DTW)**. DP in some form is now almost universally used in speech recognizers.

8.6 DYNAMIC PROGRAMMING FOR TIME ALIGNMENT

Assume that an incoming speech pattern and a template pattern are to be compared, having *n* and *N* frames respectively. Some distance metric can be used to calculate the distance, *d(i, j),* between frame *i* of the incoming speech and frame *j* of the template. To illustrate the principle, in Figure 8.3 the two sets of feature vectors of the words have been represented by letters of the word “pattern”. Differences in timescale have been indicated by repeating or omitting letters of the word, and the fact that feature vectors will not be identical, even for corresponding points of equivalent words, is indicated by using different type styles for the letters. It is, of course, assumed in this explanation that all styles of the letter “a” will yield a lower value of distance between them than, say, the distance between an “a” and any example of the letter “p”. To find the total difference between the two patterns, one requires to find the sum of all the distances between the individual pairs of frames along whichever path between the bottom-left and top-right corners in Figure 8.3 that gives the smallest distance. This definition will ensure that corresponding frames of similar words are correctly aligned.

One way of calculating this total distance is to consider all possible paths, and add the values of *d(i, j)* along each one. The distance measure between the patterns is then taken to be the lowest value obtained for the cumulative distance. Although this method is bound to give the correct answer, the number of valid paths becomes so large that the computation is impossible for any practical speech recognition machine. Dynamic programming is a mathematical technique which guarantees to

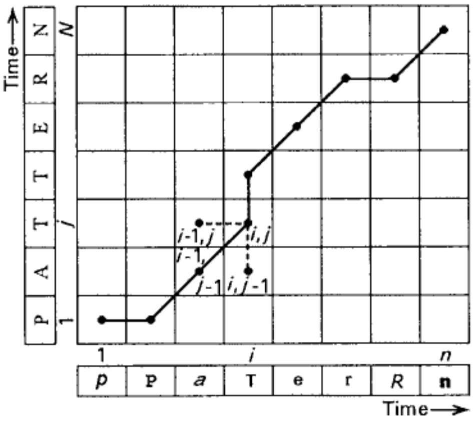


Figure 8.3 Illustration of a time-alignment path between two words that differ in their timescale. Any point *i, j* can have three predecessors as shown.

find the cumulative distance along the optimum path without having to calculate the cumulative distance along all possible paths.

Let us assume that valid paths obey certain common-sense constraints, such that portions of words do not match when mutually reversed in time (i.e. the path on Figure 8.3 always goes forward with a non-negative slope). Although skipping single frames could be reasonable in some circumstances, it simplifies the explanation if, for the present, we also assume that we can never omit from the comparison process any frame from either pattern. In Figure 8.3, consider a point *i, j* somewhere in the middle of both words. If this point is on the optimum path, then the constraints of the path necessitate that the immediately preceding point on the path is *i*-1, *j* or *i*-1, *j*-1 or *i, j*-1. These three points are associated with a horizontal, diagonal or vertical path step respectively. Let *D(i, j)* be the cumulative distance along the optimum path from the beginning of the word to point *i, j,* thus:

$$D(i,j) = \sum_{\substack{x,y=1,1\\ \text{along the}\\ \text{best path}}}^{i,j} d(x,y) \ . \tag{8.1}$$

As there are only the three possibilities for the point before point *i, j* it follows that

$$D(i,j) = \min\left[D(i-1,j), D(i-1,j-1), D(i,j-1)\right] + d(i,j) \ . \tag{8.2}$$

The best way to get to point *i, j* is thus to get to one of the immediately preceding points *by the best way,* and then take the appropriate step to *i, j*. The value of *D*(1, 1) must be equal to *d*(1, 1) as this point is the beginning of all possible paths. To reach points along the bottom and the left-hand side of Figure 8.3 there is only one possible direction (horizontal or vertical, respectively). Therefore, starting with the value of D(l, 1), values of *D(i,* 1) or values of *D*(1, *j)* can be calculated in turn for increasing values of *i* or *j*. Let us assume that we calculate the vertical column, *D*(1, *j),* using a reduced form of

Equation (8.2) that does not have to consider values of *D(i*-1, *j*) or *D(i*-1, *j*-1). (As the scheme is symmetrical we could equally well have chosen the horizontal direction instead.) When the first column values for *D(1, j)* are known, Equation (8.2) can be applied successively to calculate *D(i, j)* for columns 2 to *n*. The value obtained for *D(n, N)* is the score for the best way of matching the two words. For simple speech recognition applications, just the final score is required, and so the only working memory needed during the calculation is a one-dimensional array for holding a column (or row) of *D(i, j)* values. However, there will then be no record at the end of what the optimum path was, and if this information is required for any purpose it is also necessary to store a two-dimensional array of back-pointers, to indicate which direction was chosen at each stage. It is not possible to know until the end has been reached whether any particular point will lie on the optimum path, and this information can only be found by tracing back from the end.

8.7 REFINEMENTS TO ISOLATED-WORD DP MATCHING

The DP algorithm represented by Equation (8.2) is intended to deal with variations of timescale between two otherwise similar words. However, if two examples of a word have the same length but one is spoken faster at the beginning and slower at the end, there will be more horizontal and vertical steps in the optimum path and fewer diagonals. As a result there will be a greater number of values of *d(i, j)* in the final score for words with timescale differences than when the timescales are the same. Although it may be justified to have some penalty for timescale distortion, on the grounds that an utterance with a very different timescale is more likely to be the wrong word, it is better to choose values of such penalties explicitly than to have them as an incidental consequence of the algorithm. Making the number of contributions of *d(i, j)* to *D(n, N)* independent of the path can be achieved by modifying Equation (8.2) to add twice the value of *d(i, j)* when the path is diagonal. One can then add an explicit penalty to the right-hand side of Equation (8.2) when the step is either vertical or horizontal. Equation (8.2) thus changes to:

$$\begin{aligned} D(i,j) = \min\bigl[&D(i-1,j) + d(i,j) + hdp, \\ &D(i-1,j-1) + 2d(i,j), \\ &D(i,j-1) + d(i,j) + vdp\bigr]. \end{aligned} \tag{8.3}$$

Suitable values for the horizontal and vertical distortion penalties, *hdp* and *vdp,* would probably have to be found by experiment in association with the chosen distance metric. It is, however, obvious that, all other things being equal, paths with appreciable timescale distortion should be given a worse score than diagonal paths, and so the values of the penalties should certainly not be zero.

Even in Equation (8.3) the number of contributions to a cumulative distance will depend on the lengths of both the example and the template, and so there will be a tendency for total distances to be smaller with short templates and larger with long templates. The final best-match decision will as a result favour short words. This bias can be avoided by dividing the total distance by the template length.

The algorithm described above is inherently symmetrical, and so makes no distinction between the word in the store of templates and the new word to be identified. DP is, in fact, a much more general technique that can be applied to a wide range of applications, and which has been popularized especially by the work of Bellman (1957). The number of choices at each stage is not restricted to three, as in the example given in Figure 8.3. Nor is it necessary in speech recognition applications to assume that the best path should include all frames of both patterns. If the properties of the speech only change slowly compared with the frame interval, it is permissible to skip occasional frames, so achieving timescale compression of the pattern. A particularly useful alternative version of the algorithm is asymmetrical, in that vertical paths are not permitted. The steps have a slope of zero (horizontal), one (diagonal), or two (which skips one frame in the template). Each input frame then makes just one contribution to the total distance, so it is not appropriate to double the distance contribution for diagonal paths. Many other variants of the algorithm have been proposed, including one that allows average slopes of 0.5, 1 and 2, in which the 0.5 is achieved by preventing a horizontal step if the previous step was horizontal. Provided the details of the formula are sensibly chosen, all of these algorithms can work well. In a practical implementation computational convenience may be the reason for choosing one in preference to another.

8.8 SCORE PRUNING

Although DP algorithms provide a great computational saving compared with exhaustive search of all possible paths, the remaining computation can be substantial, particularly if each incoming word has to be compared with a large number of candidates for matching. Any saving in computation that does not affect the accuracy of the recognition result is therefore desirable. One possible computational saving is to exploit the fact that, in the calculations for any column in Figure 8.3, it is very unlikely that the best path for a correctly matching word will pass through any points for which the cumulative distance, $D(i, j)$, is much in excess of the lowest value in that column. The saving can be achieved by not allowing paths from relatively badly scoring points to propagate further. (This process is sometimes known as **pruning** because the growing paths are like branches of a tree.) There will then only be a small subset of possible paths considered, usually lying on either side of the best path. If this economy is applied it can no longer be guaranteed that the DP algorithm will find the best-scoring path. However, with a value of score-pruning threshold that reduces the average amount of computation by a factor of 5–10 the right path will almost always be obtained if the words are fairly similar. The only circumstances where this amount of pruning is likely to prevent the optimum path from being obtained will be if the words are actually different, when the resultant over-estimate of total distance would not cause any error in recognition.

Figures 8.4(a), 8.5 and 8.6 show DP paths using the symmetrical algorithm for the words illustrated in Figures 8.1 and 8.2. Figure 8.4(b) illustrates the asymmetrical algorithm for comparison, with slopes of 0, 1 and 2. In Figure 8.4 there is no time-distortion penalty, and Figure 8.5 with a small distortion penalty shows a much more plausible matching of the two timescales. The score pruning used in these figures illustrates the fact that there are low differences in cumulative

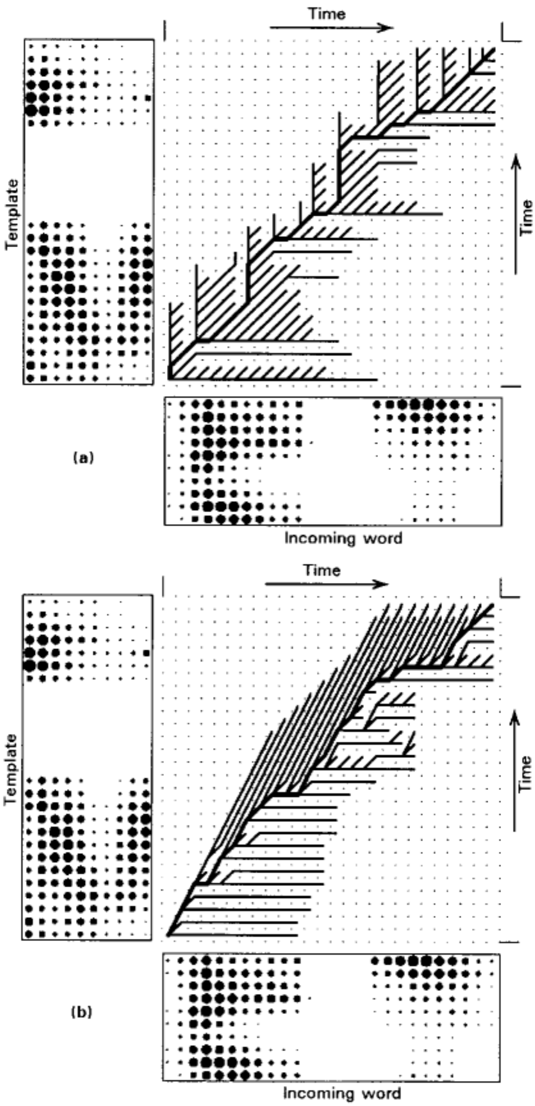


Figure 8.4 (a) DP alignment between two examples of the word “eight”, with no timescale distortion penalty but with score pruning. The optimum path, obtained by tracing back from the top right-hand corner, is shown by the thick line, **(b)** Match between the same words as in (a), but using an asymmetric algorithm with slopes of 0, 1 and 2.

distance only along a narrow band around the optimum path. When time alignment is attempted between dissimilar words, as in Figure 8.6, a very irregular path is obtained, with a poor score. Score pruning was not used in this illustration, because any path to the end of the word would then have been seriously sub-optimal.

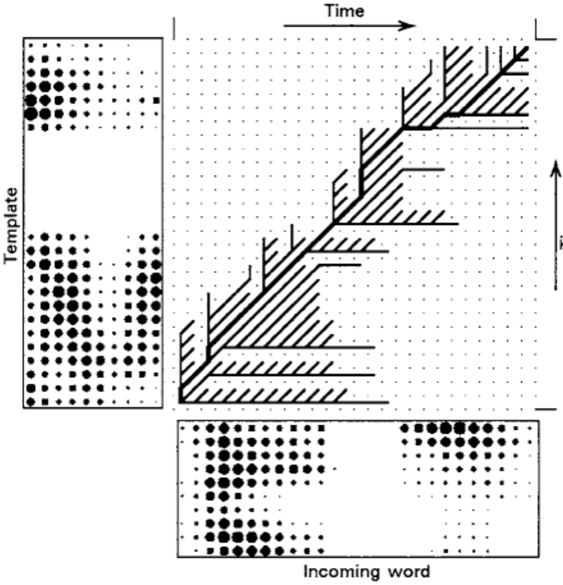


Figure 8.5 As for Figure 8.4(a), but with a small timescale distortion penalty.

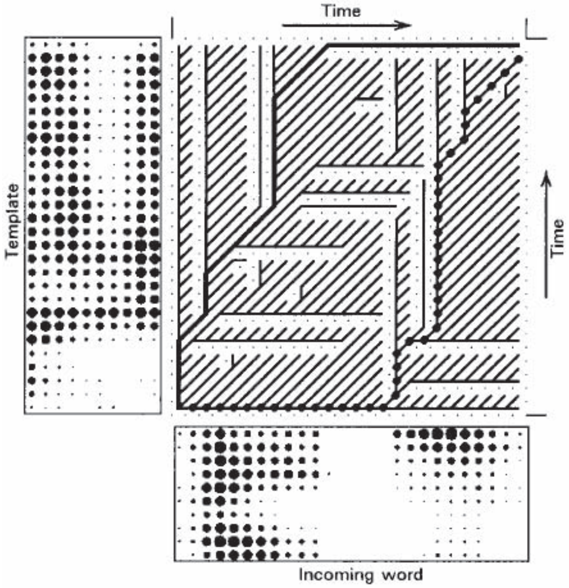


Figure 8.6 The result of trying to align two dissimilar words ("three" and "eight") within the same DP algorithm as was used for Figure 8.5. The score pruning was removed from this illustration, because any path to the end of the word would then have been seriously sub-optimal. It can be seen that if the last frame had been removed from the template, the path would have been completely different, as marked by blobs.

8.9 ALLOWING FOR END-POINT ERRORS

If an attempt is made to match two intrinsically similar words when one has its specified end point significantly in error, the best-matching path ought to align all the frames of the two words that really do correspond. Such a path implies that the extra frames of the longer word will all be lumped together at one end, as illustrated in Figure 8.7. As this extreme timescale compression is not a result of a genuine difference between the words, it may be better not to have any timescale distortion penalty for frames at the ends of the patterns, and in some versions of the algorithm it may be desirable not to include the values of *d(i, j)* for the very distorted ends of the path. If the chosen DP algorithm disallows either horizontal steps or vertical steps, correct matching of words with serious end-point errors will not be possible, and so it is probably better to remove the path slope constraints for the end frames.

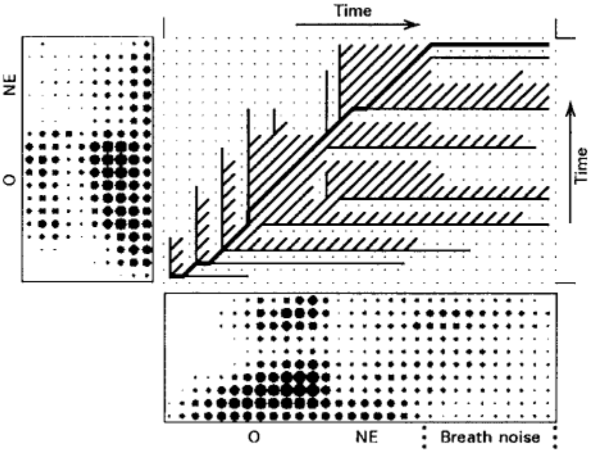


Figure 8.7 An example of the word "one" followed by breath noise, being aligned with a "one" template. A timescale distortion penalty was used except for the beginning and end frames.

8.10 DYNAMIC PROGRAMMING FOR CONNECTED WORDS

Up to now we have assumed that the words to be matched have been spoken in isolation, and that their beginnings and ends have therefore already been identified (although perhaps with difficulty). When words are spoken in a normal connected fashion, recognition is much more difficult because it is generally not possible to determine where one word ends and the next one starts independently of identifying what the words are. For example, in the sequence "six teenagers" it would be difficult to be sure that the first word was "six" rather than "sixteen" until the last syllable of the phrase had been spoken, and "sixty" might also have been possible before the [n] occurred. In some cases, such as the "grade A" example given in Chapter 1, a genuine ambiguity may remain, but for most tasks any ambiguities are resolved when at most two or three syllables have followed a word boundary.

There is another problem with connected speech as a result of co-articulation between adjacent words. It is not possible even to claim the existence of a clear point where one word stops and the next one starts. However, it is mainly the ends of words that are affected and, apart from a likely speeding up of the timescale, words in a carefully spoken connected sequence do not normally differ greatly from their isolated counterparts except near the ends. In matching connected sequences of words for which separate templates are already available one might thus define the best-matching word sequence to be given by the sequence of templates which, when joined end to end, offers the best match to the input. It is of course assumed that the optimum time alignment is used for the sequence, as with DP for isolated words. Although this model of connected speech totally ignores co-articulation, it has been successfully used in many connected-word speech recognizers.

As with the isolated-word time-alignment process, there seems to be a potentially explosive increase in computation, as every frame must be considered as a possible boundary between words. When each frame is considered as an end point for one word, all other permitted words in the vocabulary have to be considered as possible starters. Once again the solution to the problem is to apply dynamic programming, but in this case the algorithm is applied to word sequences as well as to frame sequences within words. A few algorithms have been developed to extend the isolated-word DP method to work economically across word boundaries. One of the most straightforward and widely used is described below.

In Figure 8.8 consider a point that represents a match between frame i of a multi-word input utterance and frame *j* of template number *k*. Let the cumulative distance from the beginning of the utterance along the best-matching sequence of complete templates followed by the first *j* frames of template *k* be *D(i, j, k)*. The best path through template *k* can be found by exactly the same process as for isolated-word recognition. However, in contrast to the isolated-word case, it is not known where on the input utterance the match with template *k* should finish, and for every input frame any valid path that reaches the end of template *k* could join to the beginning of the path through another template, representing the next word. Thus, for each input frame *i,* it is necessary to consider all templates that may have just ended in order to find which one has the lowest cumulative score so far. This score is then used in the cumulative distance at the start of any new template, *m:*

$$D(i,1,m) = \min_{\text{over } k}\left[D(i-1,L(k),k)\right] + d(i,1,m)\,, \tag{8.4}$$

where *L(k)* is the length of template *k*. The use of *i*-1 in Equation (8.4) implies that moving from the last frame of one template to the first frame of another always involves advancing one frame on the input (i.e. in effect only allowing diagonal paths between templates). This restriction is necessary, because the scores for the ends of all other templates may not yet be available for input frame i when the path decision has to be made. A horizontal path from within template *m* could have been included in Equation (8.4), but has been omitted merely to simplify the explanation. A timescale distortion penalty has not been included for the same reason.

In the same way as for isolated words, the process can be started off at the beginning of an utterance because all values of D(0, *L(k), k*) will be zero. At the end of an utterance the template that gives the lowest cumulative distance is assumed to represent the final word of the sequence, but its identity gives no indication of the templates that preceded it. These can only be determined by storing pointers to the preceding templates of each path as it evolves, and then tracing back when the final point is reached. It is also possible to recover the positions in the input sequence

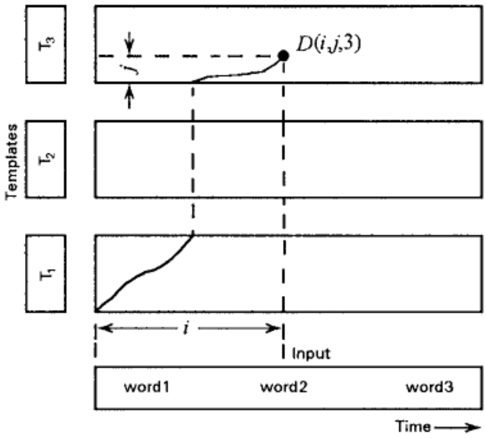


Figure 8.8 Diagram indicating the best-matching path from the beginning of an utterance to the j^{th} frame of template T_3 and the i^{th} frame of the input. In the example shown i is in the middle of the second word of the input, so the best path includes one complete template (T_1) and a part of T_3. The cumulative distance at this point is denoted *by D(i, j,* 3), or in general *by D(i, j, k)* for the k^{th} template.

where the templates of the matching sequence start and finish, so segmenting the utterance into separate words. Thus we solve the segmentation problem by delaying the decisions until we have seen the whole utterance and decided on the words.

The process as described so far assumes that any utterance can be modelled completely by a sequence of word templates. In practice a speaker may pause between words, so giving a period of silence (or background noise) in the middle of an utterance. The same algorithm can still be used for this situation by also storing a template for a short period of silence, and allowing this **silence template** to be included between appropriate pairs of valid words. If the silence template is also allowed to be chosen at the start or end of the sequence, the problem of end-point detection is greatly eased. It is only necessary to choose a threshold that will never be exceeded by background noise, and after the utterance has been detected, to extend it by several frames at each end to be sure that any low-intensity parts of the words are not omitted. Any additional frames before or after the utterance should then be well modelled by a sequence of one or more silence templates.

When a sequence of words is being spoken, unintentional extraneous noises (such as grunts, coughs and lip smacks) will also often be included between words. In an isolated-word recognizer these noises will not match well to any of the templates, and can be rejected on this basis. In a connected-word algorithm there is no provision for not matching any part of the sequence. However, the rejection of these unintentional insertions can be arranged by having a special template, often called a **wildcard template,** that bypasses the usual distance calculation and is deemed to match with any frame of the input to give a standard value of distance. This value is chosen to be greater than would be expected for corresponding frames of equivalent words, but less than should occur when trying to match quite different sounds. The wildcard will then provide the best score when attempting to match spurious sounds and words not in the stored template vocabulary, but should not normally be chosen in preference to any of the well-matched words in the input.

8.11 CONTINUOUS SPEECH RECOGNITION

In the connected-word algorithm just described, start and finish points of the input utterance must at least be approximately determined. However it is not generally necessary to wait until the end of an utterance before identifying the early words. Even before the end, one can trace back along all current paths through the tree that represents the candidates for the template sequence. This tree will always involve additional branching as time goes forward, but the ends of many of the 'twigs' will not represent a low enough cumulative distance to successfully compete with other twigs as starting points for further branching, and so paths along these twigs will be abandoned. It follows that tracing back from all currently active twigs will normally involve coalescence of all paths into a single 'trunk', which therefore represents a uniquely defined sequence of templates (see Figure 8.9). The results up to the first point of splitting of active paths can therefore be output from the machine, after which the back-pointers identifying that part of the path are no longer needed, nor are those representing abandoned paths. The memory used for storing them can therefore be released for re-use with new parts of the input signal.

The recognizer described above can evidently operate continuously, with a single pass through the input data, outputting its results always a few templates behind the current best match. Silence templates are used to match the signal when the speaker pauses, and wildcards are used for extraneous noises or inadmissible words. The time lag for output is determined entirely by the need to resolve ambiguity. When two alternative sequences of connected words both match the input well, but with different boundary points (e.g. "grey day" and "grade A") it is necessary to reach the end of the ambiguous sequence before a decision can be reached on any part of it. (In the example just given, the decision might even then

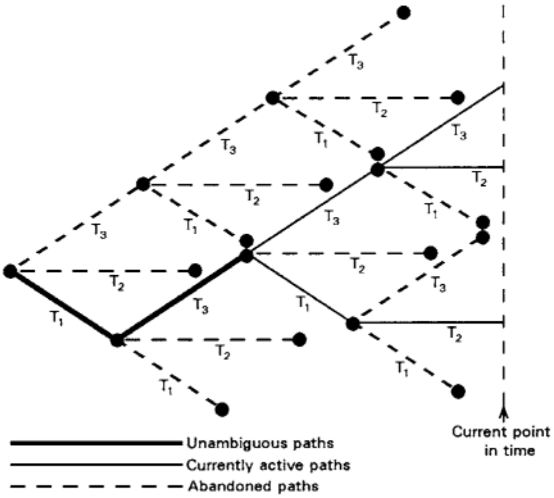


Figure 8.9 Trace-back through a word decision tree to identify unambiguous paths for a three-word vocabulary continuous recognizer. Paths are abandoned when the cumulative distances of all routes to the ends of the corresponding templates are greater than for paths to the ends of different template sequences at the same points in the input. Template sequences still being considered are T_1-T_3-T_{1-} T_3, T_1-T_3-T_3-T_1 and T_1-T_1-T_1-T_1. Thus T_2 is being scored separately for two preceding sequences.

be wrong because of inherent ambiguity in the acoustic signal.) On the other hand, if the input matches very badly to all except one of the permitted words, all paths not including that word will be abandoned as soon as the word has finished. In fact, if score pruning is used to cause poor paths to be abandoned early, the path in such a case may be uniquely determined even at a matching point within the word. There is plenty of evidence that human listeners also often decide on the identity of a long word before it is complete if its beginning is sufficiently distinctive.

8.12 SYNTACTIC CONSTRAINTS

The rules of grammar often prevent certain sequences of words from occurring in human language, and these rules apply to particular syntactic classes, such as nouns, verbs, etc. In the more artificial circumstances in which speech recognizers are often used, the tasks can sometimes be arranged to apply much more severe constraints on which words are permitted to follow each other. Although applying such constraints requires more care in designing the application of the recognizer, it usually offers a substantial gain in recognition accuracy because there are then fewer potentially confusable words to be compared. The reduction in the number of templates that need to be matched at any point also leads to a computational saving.

8.13 TRAINING A WHOLE-WORD RECOGNIZER

In all the algorithms described in this chapter it is assumed that suitable templates for the words of the vocabulary are available in the machine. Usually the templates are made from speech of the intended user, and thus a **training session** is needed for enrolment of each new user, who is required to speak examples of all the vocabulary words. If the same user regularly uses the machine, the templates can be stored in some back-up memory and re-loaded prior to each use of the system. For isolated-word recognizers the only technical problem with training is end-point detection. If the templates are stored with incorrect end points the error will affect recognition of every subsequent occurrence of the faulty word. Some systems have tried to ensure more reliable templates by time aligning a few examples of each word and averaging the measurements in corresponding frames. This technique gives some protection against occasional end-point errors, because such words would then give a poor match in this alignment process and so could be rejected.

If a connected-word recognition algorithm is available, each template can be segmented from the surrounding silence by means of a special training syntax that only allows silence and wildcard templates. The new template candidate will obviously not match the silence, so it will be allocated to the wildcard. The boundaries of the wildcard match can then be taken as end points of the template.

In acquiring templates for connected-word recognition, more realistic training examples can be obtained if connected words are used for the training. Again the recognition algorithm can be used to determine the template end points, but the syntax would specify the preceding and following words as existing templates, with just the new word to be captured represented by a wildcard between them. Provided the surrounding words can be chosen to give clear acoustic boundaries where they join to the new word, the segmentation will then be fairly accurate. This process is often called **embedded training**. More powerful embedded training procedures for use with statistical recognizers are discussed in Chapters 9 and 11.

CHAPTER 8 SUMMARY

- Most early successful speech recognition machines worked by pattern matching on whole words. Acoustic analysis, for example by a bank of bandpass filters, describes the speech as a sequence of feature vectors, which can be compared with stored templates for all the words in the vocabulary using a suitable distance metric. Matching is improved if speech level is coded logarithmically and level variations are normalized.
- Two major problems in isolated-word recognition are end-point detection and timescale variation. The timescale problem can be overcome by dynamic programming (DP) to find the best way to align the timescales of the incoming word and each template (known as dynamic time warping). Performance is improved by using penalties for timescale distortion. Score pruning, which abandons alignment paths that are scoring badly, can save a lot of computation.
- DP can be extended to deal with sequences of connected words, which has the added advantage of solving the end-point detection problem. DP can also operate continuously, outputting words a second or two after they have been spoken. A wildcard template can be provided to cope with extraneous noises and words that are not in the vocabulary.
- A syntax is often provided to prevent illegal sequences of words from being recognized. This method increases accuracy and reduces the computation.

CHAPTER 8 EXERCISES

E8.1 Give examples of factors which cause acoustic differences between utterances of the same word. Why does simple pattern matching work reasonably well in spite of this variability?

E8.2 What factors influence the choice of bandwidth for filter-bank analysis?

E8.3 What are the reasons in favour of logarithmic representation of power in filter-bank analysis? What difficulties can arise due to the logarithmic scale?

E8.4 Explain the principles behind dynamic time warping, with a simple diagram.

E8.5 Describe the special precautions which are necessary when using the symmetrical DTW algorithm for isolated-word recognition.

E8.6 How can a DTW isolated-word recognizer be made more tolerant of end-point errors?

E8.7 How can a connected-word recognizer be used to segment a speech signal into individual words?

E8.8 What extra processes are needed to turn a connected-word recognizer into a continuous recognizer?

E8.9 Describe a training technique suitable for connected-word recognizers.

CHAPTER 9

Introduction to Stochastic Modelling

9.1 FEATURE VARIABILITY IN PATTERN MATCHING

The recognition methods described in the previous chapter exploit the fact that repeated utterances of the same word normally have more similar acoustic patterns than utterances of different words. However, it is to be expected that some parts of a pattern may vary more from occurrence to occurrence than do other parts. In the case of connected words, the ends of the template representing each word are likely to have a very variable degree of match, depending on the amount that the input pattern is modified by co-articulation with adjacent words. There is also no reason to assume that the individual features of a feature vector representing a particular phonetic event are of equal consistency. In fact, it may well occur that the value of a feature could be quite critical at a particular position in one word, while being very variable and therefore not significant in some part of a different word.

Timescale variability has already been discussed in Chapter 8. It must always be desirable to have some penalty for timescale distortion, as durations of speech sounds are not normally wildly different between different occurrences of the same word. However, there is no reason to assume that the time distortion penalty should be constant for all parts of all words. For example, it is known that long vowels can vary in length a lot, whereas most spectral transitions associated with consonants change in duration only comparatively slightly.

From the above discussion it can be seen that the ability of a recognizer to distinguish between words is likely to be improved if the variability of the patterns can be taken into account. We should not penalize the matching of a particular word if the parts that match badly are parts which are known to vary extensively from utterance to utterance. To use information about variability properly we need to have some way of collecting statistics which represent the variability of the word patterns, and a way of using this variability in the pattern-matching process.

The basic pattern-matching techniques using DTW as described in Chapter 8 started to be applied to ASR in the late 1960s and became popular during the 1970s. However, the application of statistical techniques to this problem was also starting to be explored during the 1970s, with early publications being made independently by Baker (1975) working at Carnegie-Mellon University (CMU) and by Jelinek (1976) from IBM. These more powerful techniques for representing variability have gradually taken over from simple pattern matching. In the period since the early publications by Baker and by Jelinek, there has been considerable research to refine the use of statistical methods for speech recognition, and some variant of these methods is now almost universally adopted in current systems.

This chapter provides an introduction to statistical methods for ASR. In order to accommodate pattern variability, these methods use a rather different way of defining the degree of fit between a word and some speech data, as an alternative to the 'cumulative distance' used in Chapter 8. This measure of degree of fit is based on the notion of **probability,** and the basic theory is explained in this chapter. For simplicity in introducing the concepts, the discussion in this chapter will continue to concentrate on words as the recognition unit. In practice, the majority of current recognition systems represent words as a sequence of **sub-word units,** but the underlying theory is not affected by the choice of unit. The use of sub-word units for recognition, together with other developments and elaborations of the basic statistical method will be explained in later chapters. In the following explanation, some elementary knowledge of statistics and probability theory is assumed, but only at a level which could easily be obtained by referring to a good introductory textbook (see Chapter 17 for some references).

9.2 INTRODUCTION TO HIDDEN MARKOV MODELS

Up to now we have considered choosing the best matching word by finding the template which gives the minimum cumulative 'distance' along the optimum matching path. An alternative approach is, for each possible word, to postulate some device, or model, which can generate patterns of features to represent the word. Every time the model for a particular word is activated, it will produce a set of feature vectors that represents an example of the word, and if the model is a good one, the statistics of a very large number of such sets of feature vectors will be similar to the statistics measured for human utterances of the word. The best matching word in a recognition task can be defined as the one whose model is most likely to produce the observed sequence of feature vectors. What we have to calculate for each word is thus not a 'distance' from a template, but the *a posteriori* probability that its model could have produced the observed set of feature vectors. We do not actually have to make the model produce the feature vectors, but we use the known properties of each model for the probability calculations. We will assume for the moment that the words are spoken in an 'isolated' manner, so that we know where the start and end of each word are, and the task is simply to identify the word. Extensions to sequences of words will be considered in Section 9.11.

We wish to calculate the *a posteriori* probability, $P(\boldsymbol{w}|Y)$, of a particular word, w, having been uttered during the generation of a set of feature observations, $\boldsymbol{Y}$. We can use the model for w to calculate $P(\boldsymbol{Y}|w)$, which is the probability of $\boldsymbol{Y}$ conditioned on word w (sometimes referred to as the **likelihood** of w). To obtain $P(w|\boldsymbol{Y})$, however, we must also include the *a priori* probability of word w having been spoken. The relationship between these probabilities is given by Bayes' rule:

$$P(w|\boldsymbol{Y})=\frac{P(\boldsymbol{Y}|w)P(w)}{P(\boldsymbol{Y})}. \tag{9.1}$$

This equation states that the probability of the word given the observations is equal to the probability of the observations given the word, multiplied by the probability of the word (irrespective of the observations), and divided by the probability of the observations. The probability, $\boldsymbol{P(Y)}$, of a particular set of feature observations, $\boldsymbol{Y}$, does not depend on which word is being considered as a possible match, and therefore only acts as a scaling factor on the probabilities. Hence, if the goal is to find the word w which maximizes $P(w|\boldsymbol{Y})$, the $P(\boldsymbol{Y})$ term can be ignored, because it does not affect the choice of word. If for the particular application all permitted words are equally likely, then the *P(w)* term can also be ignored, so we merely have to choose the word model that maximizes the probability, *P(**Y**|w),* of producing the observed feature set, ***Y***. In practice for all but the simplest speech recognizers the probability of any particular word occurring will depend on many factors, and for large vocabularies it will depend on the statistics of word occurrence in the language. This aspect will be ignored in the current chapter, but will be considered further in Chapter 12.

The way we have already represented words as sequences of template frames gives us a starting point for the form of a possible model. Let the model for any word be capable of being in one of a sequence of states, each of which can be associated with one or more frames of the input. In general the model moves from one state to another at regular intervals of time equal to the frame interval of the acoustic analysis. However, we know that words can vary in timescale. In the asymmetrical DP algorithm mentioned in Chapter 8 (Figure 8.4(b), showing slopes of 0, 1 and 2) the timescale variability is achieved by repeating or skipping frames of the template. In our model this possibility can be represented in the sequence of states by allowing the model to stay in the same state for successive frame times, or to bypass the next state in the sequence. The form of this simple model is shown in Figure 9.1. In fact, if a word template has a sequence of very similar frames, such as might occur in a long vowel, it is permissible to reduce the number of states in the model by allowing it to stay in the same state for several successive frames.

The mathematics associated with a model such as the one shown in Figure 9.1 can be made more tractable by making certain simplifying assumptions. To be more specific, it is assumed that the output of the model is a **stochastic process** (i.e. its operation is governed completely by a set of probabilities), and that the probabilities of all its alternative actions at any time *t* depend only on the state it is in at that time, and not on the value of *t*. The current output of the model therefore depends on the identity of the current state, but is otherwise independent of the sequence of previous states that it has passed through to reach that state. Hence the model's operation is a **first-order Markov process,** and the sequence of states is a **first-order Markov chain**. Although the model structure shown in Figure 9.1 is quite appropriate for describing words that vary in timescale, the equations that represent the model's behaviour have exactly the same form in the more general case where transitions are allowed between all possible pairs of states.

At every frame time the model is able to change state, and will do so randomly in a way determined by a set of **transition probabilities** associated with the state it is currently in. By definition, the probabilities of all transitions from a

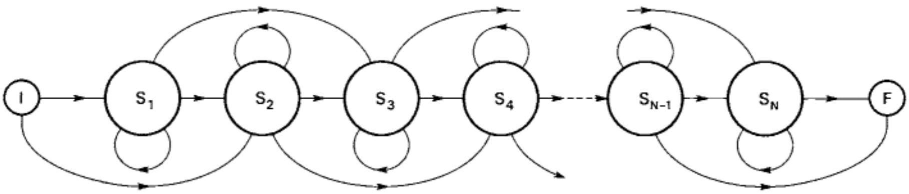


Figure 9.1. State transitions for a simple word model, from an initial state, I, to a final state, F.

state at any frame time must sum to 1, but the sum includes the probability of a transition that re-enters the same state. When the model is activated a sequence of feature vectors is emitted, in the same form as might be observed when a word is spoken. However, in the type of model considered here, observing the feature vectors does not completely determine what the state sequence is. In addition to its transition probabilities, each state also has associated with it a **probability density function (p.d.f.)** for the feature vectors. Each p.d.f. can be used to calculate the probability that any particular set of feature values could be emitted when the model is in the associated state. This probability is usually known as the **emission probability.** The actual values of the observed features are, therefore, probabilistic functions of the states, and the states themselves are hidden from the observer. For this reason this type of model is called a **hidden Markov model (HMM)**.

The emission p.d.f. for a state may be represented as a discrete distribution, with a probability specified separately for each possible feature vector. Alternatively, it is possible to use a parameterized continuous distribution, in which feature vector probabilities are defined by the parameters of the distribution. Although there are significant advantages, which will be explained in Section 9.7, in modelling feature probabilities as continuous functions, it will simplify the following explanation if we initially consider only discrete probability distributions.

9.3 PROBABILITY CALCULATIONS IN HIDDEN MARKOV MODELS

In order to explain the HMM probability calculations, we will need to introduce some symbolic notation to represent the different quantities which must be calculated. Notation of this type can be found in many publications on the subject of HMMs for ASR. Certain symbols have come to be conventionally associated with particular quantities, although there is still some variation in the details of the notation that is used. In choosing the notation for this book, our aims were to be consistent with what appears to be used the most often in the published literature, while also being conceptually as simple as possible.

We will start by assuming that we have already derived good estimates for the parameters of all the word models. (Parameter estimation will be discussed later in the chapter.) The recognition task is to determine the most probable word, given the observations (i.e. the word w for which $P(w|\boldsymbol{Y})$ is maximized). As explained in Section 9.2, we therefore need to calculate the likelihood of each model emitting the observed sequence of features (i.e. the value of $P(\boldsymbol{Y}|w)$ for each word w).

Considering a single model, an output representing a whole word arises from the model going through a sequence of states, equal in length to the number of observed feature vectors, T, that represents the word. Let the total number of states in the model be N, and let s_t denote the state that is occupied during frame t of the model's output. We will also postulate an initial state, I and a final state, F, which are not associated with any emitted feature vector and only have a restricted set of possible transitions. The initial state is used to specify transition probabilities from the start to all permitted first states of the model, while the final state provides transition probabilities from all possible last emitting states to the end of the word. The model must start in state I and end in state F, so in total the model will go through a sequence of T+2 states to generate T observations.

The use of non-emitting initial and final states provides a convenient method for modelling the fact that some states are more likely than others to be associated with the first and the last frame of the word respectively[1]. These compulsory special states will also be useful in later discussions requiring *sequences* of models.

The most widely used notation for the probability of a transition from state *i* to state *j* is a_{ij}. The emission probability of state *j* generating an observed feature vector $\boldsymbol{y}_t$, is usually denoted $b_j(\boldsymbol{y}_t)$.

We need to compute the probability of a given model producing the observed sequence of feature vectors, $\boldsymbol{y}_1$ to $\boldsymbol{y}_T$. We know that this sequence of observations must have been generated by a state sequence of length *T* (plus the special initial and final states) but, because the model is hidden, we do not know the identities of the states. Hence we need to consider all possible state sequences of length *T*. The probability of the model generating the observations can then be obtained by finding the **joint probability** of the observations and any one state sequence, and summing this quantity over all possible state sequences of the correct length:

$$P(\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_T) = \sum_{\substack{\text{over all possible}\\ \text{state sequences}\\ \text{of length } T}} P(\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_T,s_1,s_2,\cdots,s_T)$$

$$= \sum_{\substack{\text{over all possible}\\ \text{state sequences}\\ \text{of length } T}} P(\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_T,|\, s_1,s_2,\cdots,s_T)P(s_1,s_2,\cdots,s_T)\,, \qquad (9.2)$$

where, for notational convenience, in the equations we are omitting the dependence of all the probabilities on the identity of the model.

Now the probability of any particular state sequence is given by the product of the transition probabilities:

$$P(s_1,s_2,\cdots,s_T) = a_{Is_1}\Big(\prod_{t=1}^{T-1} a_{s_t s_{t+1}}\Big)a_{s_T F}\,, \qquad (9.3)$$

where $a_{s_t s_{t+1}}$ is the probability of a transition from the state occupied at frame *t* to the state at frame *t*+1; a_{Is_1} and $a_{s_T F}$ similarly define the transition probabilities from the initial state I and to the final state *F*. If we assume that the feature vectors are generated independently for each state, the probability of the observations given a particular state sequence of duration *T* is the product of the individual emission probabilities for the specified states:

$$P(\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_T \mid s_1,s_2,\cdots,s_T) = \prod_{t=1}^{T} b_{s_t}(\boldsymbol{y}_t)\,. \qquad (9.4)$$

[1] Some published descriptions of HMM theory do not include special initial and final states. Initial conditions are sometimes accommodated by a vector of probabilities for starting in each of the states (e.g. Levinson *et at*., 1983), which has the same effect as the special initial state used here. For the last frame of the word, approaches include allowing the model to end in any state (e.g. Levinson *et al*., 1983) or enforcing special conditions to only allow the model to end in certain states. The treatment of the first and last frames does not alter the basic form of the probability calculations, but it may affect the details of the expressions associated with the start and end of an utterance.

Thus the probability of the model emitting the complete observation sequence is:

$$P(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_T) = \sum_{\substack{\text{over all possible} \\ \text{state sequences} \\ \text{of length } T}} a_{Is_1} \left(\prod_{t=1}^{T-1} b_{s_t}(\boldsymbol{y}_t) a_{s_t s_{t+1}} \right) b_{s_T}(\boldsymbol{y}_T) a_{s_T F} \,. \tag{9.5}$$

Unless the model has a small number of states and T is small, there will be an astronomical number of possible state sequences, and it is completely impractical to make the calculations of Equation (9.5) directly for all sequences. One can, however, compute the probability indirectly by using a recurrence relationship. We will use the symbol $\alpha_j(t)$ to be the probability[2] of the model having produced the first t observed feature vectors and being in state j for frame t. The recurrence can be computed in terms of the values of $\alpha_i(t\text{-}1)$ for all possible previous states, i.

$$\alpha_j(t) = P(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_t, s_t = j) \tag{9.6}$$

$$= \left(\sum_{i=1}^{N} \alpha_i(t-1) a_{ij} \right) b_j(\boldsymbol{y}_t) \quad \text{for } 1 < t \le T \tag{9.7}$$

The value of $a_j(1)$, for the first frame, is the product of the transition probability a_{Ij} from the initial state I, and the emission probability $b_j(\boldsymbol{y}_1)$.

$$\alpha_j(1) = a_{Ij} b_j(\boldsymbol{y}_1) \tag{9.8}$$

The value of $a_j(T)$, for the last frame in the observation sequence, can be computed for any of the emitting states by repeated applications of Equation (9.7), starting from the result of Equation (9.8).

The total probability of the complete set of observations being produced by the model must also include the transition probabilities into the final state F. We will define this quantity as $a_F(T)$, thus:

$$P(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_T) = \alpha_F(T) = \sum_{i=1}^{N} \alpha_i(T) a_{iF} \,. \tag{9.9}$$

Equation (9.9) gives the probability of the model generating the observed data, taking into account all possible sequences of states. This quantity represents the probability of the observations given the word model (the $P(\boldsymbol{Y}|w)$ term in Equation (9.1)). Incorporating the probability of the word, $P(w)$, gives a probability that is a scaled version of $P(w|\boldsymbol{Y})$, the probability of the word having been spoken. Provided that the model is a good representation of its intended word, this probability provides a useful measure which can be compared with the probability according to alternative word models in order to identify the most probable word.

[2] In the literature, this probability is almost universally represented by the symbol *a*. However, there is some variation in the way in which the *a* symbol is annotated to indicate dependence on state and time. In particular, several authors (e.g. Rabiner and Juang (1993)) have used $a_t(j)$, whereas we have chosen $a_j(t)$ (as used by Knill and Young (1997) for example). The same variation applies to the quantities *ß*, *?* and *?*, which will be introduced later. The differences are only notational and do not affect the meaning of the expressions, but when reading the literature it is important to be aware that such differences exist.

9.4 THE VITERBI ALGORITHM

The probability of the observations, given the model, is made up of contributions from a very large number of alternative state sequences. However, the probability distributions associated with the states will be such that the probability of the observed feature vectors having been produced by many of the state sequences will be microscopically small compared with the probabilities associated with other state sequences. One option is to ignore all but the single most probable state sequence. Equation (9.2) can be modified accordingly to give the probability, $\hat{P}$, of the observations for this most probable state sequence:

$$\hat{P}(y_1, y_2, \cdots, y_T) = \max_{\substack{\text{over all possible}\\ \text{state sequences}\\ \text{of length } T}} \left(P(y_1, y_2, \cdots, y_T, s_1, s_2, \cdots, s_T)\right). \tag{9.10}$$

The probability associated with the most probable sequence of states can be calculated using the **Viterbi algorithm** (Viterbi, 1967), which is a dynamic programming algorithm applied to probabilities. Let us define a new probability, $\hat{\alpha}_j(t)$ *as* the probability of being in the j^{th} state, after having emitted the first t feature vectors and *having been through the most probable sequence* of t-1 preceding states in the process. Again we have a recurrence relation, equivalent to the one shown in Equation (9.7):

$$\hat{\alpha}_j(t) = \max_{\text{over } i} \left(\hat{\alpha}_i(t-1)a_{ij}\right) b_j(y_t) \quad \text{for } 1 < t \le T \ . \tag{9.11}$$

The conditions for the first state are the same as for the total probability, which was given in Equation (9.8):

$$\hat{\alpha}_j(1) = \alpha_j(1) = a_{Ij} b_j(y_1) \ . \tag{9.12}$$

Successive applications of Equation (9.11) will eventually yield the values for $\hat{\alpha}_j(T)$. *D*efining $\hat{\alpha}_F(T)$ *a*s the probability of the full set of observations being given by the most probable sequence of states, its value is given by:

$$\hat{P}(y_1, y_2, \cdots, y_T) = \hat{\alpha}_F(T) = \max_{\text{over } i} \left(\hat{\alpha}_i(T)a_{iF}\right). \tag{9.13}$$

The difference between the total probability and the probability given by the Viterbi algorithm depends on the magnitude of the contribution of the 'best' state sequence to the total probability summed over all possible sequences. If the feature-vector p.d.f.s of all states are substantially different from each other, the probability of the observations being produced by the best sequence might not be appreciably less than the total probability including all possible sequences. The difference between the total probability and the probability for the best sequence will, however, be larger if the best path includes several consecutive frames shared between a group of two or more states which have very similar p.d.f.s for the feature vectors. Then the probability of generating the observed feature vectors would be almost independent of how the model distributed its time between the states in this group. The total probability, which is the sum over all possible allocations of frames to states, could then be several times the probability for the best sequence. This point will be considered again in Section 9.14. However, the design of models used in current recognizers is such that sequences of states with similar emission p.d.f.s generally do not occur. As a consequence, in spite of the theoretical disadvantage of ignoring all but the best path, in practice the differences in performance between the two methods are usually small. Some variant of the Viterbi algorithm is therefore usually adopted for decoding in practical speech recognizers, as using only the best path requires less computation. (There can also be considerable advantages for implementation, as will be discussed in Section 9.12.)

9.5 PARAMETER ESTIMATION FOR HIDDEN MARKOV MODELS

So far, we have considered the probability calculations required for recognition. We have assumed that the parameters of the models, i.e. the transition probabilities and emission p.d.f.s for all the states, are already set to their optimum values for modelling the statistics of a very large number of human utterances of all the words that are to be recognized. In the discussion which follows we will consider the problem of deriving suitable values for these parameters from a quantity of training data. We will assume for the moment that the body of training data is of sufficient size to represent the statistics of the population of possible utterances, and that we have sufficient computation available to perform the necessary operations.

The training problem can be formulated as one of determining the values of the HMM parameters in order to maximize the probability of the training data being generated by the models ($P(\boldsymbol{Y}|w)$ in Equation (9.1)). Because this conditional probability of the observations $\boldsymbol{Y}$ given word w is known as the 'likelihood' of the word w, the training criterion that maximizes this probability is referred to as **maximum likelihood** (other training criteria will be considered in Chapter 11). If we knew which frames of training data corresponded to which model states, then it would be straightforward to calculate a maximum-likelihood estimate of the probabilities associated with each state. The transition probabilities could be calculated from the statistics of the state sequences, and the emission probabilities from the statistics of the feature vectors associated with each state. However, the 'hidden' nature of the HMM states is such that the allocation of frames to states cannot be known. Therefore, although various heuristic methods can be formulated for analysing the training data to give rough estimates of suitable model parameters, there is no method of calculating the optimum values directly.

If, however, one has a set of rough estimates for all the parameters, it is possible to use their values in a procedure to compute new estimates for each parameter. This algorithm was developed by Baum and colleagues and published in a series of papers in the late 1960s and early 1970s. It has been proved by Baum (1972) that new parameter estimates derived in this way always produce a model that is at least as good as the old one in representing the data, and in general the new estimates give an improved model. If we iterate these operations a sufficiently large number of times the model will converge to a locally optimum solution. Unfortunately, it is generally believed that the number of possible local optima is so vast that the chance of finding the global optimum is negligible. However, it is unlikely that the global optimum would in practice be much better than a good local optimum, derived after initialization with suitable starting estimates for the models.

Baum's algorithm is an example of a general method which has come to be known as the **expectation-maximization (EM) algorithm** (Dempster *et al.*, 1977). The EM algorithm is applicable to a variety of situations in which the task is to estimate model parameters when the observable data are 'incomplete', in the sense that some information (in this case the state sequence) is missing.

The detailed mathematical proofs associated with the derivation of the re-estimation formulae for HMMs are beyond the scope of this book, although Chapter 17 gives some references. In the current chapter, we will describe the reestimation calculations and give some intuitive explanation. The basic idea is to use some existing estimates for the model parameters to calculate the probability of being in each state at every frame time, given these current estimates of the model parameters *and* the training data. The probabilities of occupying the states can then be taken into account when gathering the statistics of state sequences and of feature vectors associated with the states, in order to obtain new estimates for the transition probabilities and for the emission probabilities respectively. In the re-estimation equations we will use a bar above the symbol to represent a re-estimated value, and the same symbol without the bar to indicate its previous value.

9.5.1 Forward and backward probabilities

Suppose for the moment that we have just a single example of a word, and that this example comprises the sequence of feature vectors y_1 to y_T. Also, assume that the word has been spoken in isolation and we know that y_1 corresponds to the first frame of the word, with y_T representing the last frame. In Equation (9.7) we showed how to compute $\alpha_j(t)$, which is the probability of the model having emitted the first t observed feature vectors and being in state j. The values of $\alpha_j(t)$ are computed for successive frames in order, going **forward** from the beginning of the utterance. When estimating parameters for state j, we will need to know the probability of being in the state at time t, while the model is in the process of emitting *all* the feature vectors that make up the word. For this purpose we also need to compute $ß_j(t)$, which is defined as the **backward** probability of emitting the remaining T-t observed vectors that are needed to complete the word, given that the j^{th} state was occupied for frame t:

$$\beta_j(t) = P(\boldsymbol{y}_{t+1}, \boldsymbol{y}_{t+2}, \cdots, \boldsymbol{y}_T \mid s_t = j)\,. \tag{9.14}$$

When calculating the backward probabilities, it is necessary to start applying the recurrence from the end of the word and to work backwards through the sequence of frames. Each backward probability at time t is therefore derived from the backward probabilities at time t+1. Because the notation convention is to move from state i to state j, it is usual to specify the recurrence relationship for the backward probabilities with the i^{th} state occupied at time t. Thus the value of $ß_i(t)$ is computed in terms of the values of $ß_j(t+1)$ for all possible following states j:

$$\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{y}_{t+1}) \beta_j(t+1) \qquad \text{for } T > t \geq 1\,. \tag{9.15}$$

In contrast to Equation (9.7), it will be noticed that Equation (9.15) does not include the emission probability for frame *t*. This difference in form between the definitions of $\alpha_i(t)$ and $\beta_i(t)$ is necessary because of the way we will combine these quantities in Equation (9.17).

The first application of Equation (9.15) uses the fact that the model must be in the final state, *F*, at the end of the word. At this point all features will have been emitted, so the value of $\beta_i(T)$ is just the probability of a transition from state *i* to state *F*:

$$\beta_i(T) = a_{iF} . \tag{9.16}$$

The probability of the model emitting the full set of *T* feature vectors and being in the *j*th state for the *t*th observed frame must be the product of the forward and backward probabilities for the given state and frame pair, thus:

$$P(y_1, y_2, \cdots, y_T, s_t = j) = \alpha_j(t)\beta_j(t) . \tag{9.17}$$

Although it is not relevant to parameter re-estimation, it is interesting to note that, as the probability of generating the full set of feature vectors and being in state *j* for frame *t* is given by $\alpha_j(t)\beta_j(t)$, the probability of the observations irrespective of which state is occupied in frame *t* must be the value of this product summed over all states. We can write this probability as:

$$P(y_1, y_2, \cdots, y_T) = \sum_{i=1}^{N} \alpha_i(t)\beta_i(t) \quad \text{for any value of } t, \tag{9.18}$$

where here we use i as the state index for ease of comparison with Equation (9.9). Equation (9.18) is true for any value of the frame time, *t,* and Equation (9.9) is thus just a special case for the last frame, where $t=T$ and in consequence $\beta_i(T)=a_{iF}$.

9.5.2 Parameter re-estimation with forward and backward probabilities

In practice when training a set of models there would be several (say *E*) examples of each word, so the total number of feature vectors available is the sum of the numbers of frames for the individual examples. The re-estimation should use all the training examples with equal weight. For this purpose it is necessary to take into account that the current model would be expected to fit some examples better than others, and we need to prevent these examples from being given more weight in the re-estimation process. The simple product $\alpha_j(t)\beta_j(t)$ does not allow for these differences, as it represents the *joint* probability of being in state *j* at time *t* and generating a particular set of feature vectors representing one example. In order to be able to combine these quantities for different examples, we require the *conditional* probability of occupying state *j* given the feature vectors.

We will define a quantity $\gamma_j(t)$, which is the probability of being in state *j* for frame *t,* given the feature vectors for one example of the word. This quantity can be derived from $\alpha_j(t)\beta_j(t)$ using Bayes' rule, and it can be seen that the result involves simply normalizing $\alpha_j(t)\beta_j(t)$ by the probability of the model generating the observations.

$$\gamma_j(t) = P(s_t = j \mid y_1, y_2, \cdots, y_T) = \frac{P(y_1, y_2, \cdots, y_T \mid s_t = j) P(s_t = j)}{P(y_1, y_2, \cdots, y_T)}$$

$$= \frac{P(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_T, s_t = j)}{P(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_T)} = \frac{\alpha_j(t)\beta_j(t)}{\alpha_F(T)} \tag{9.19}$$

The normalization by $\alpha_F(T)$ thus ensures that when there are several examples of the word, all frames of all examples will contribute equally to the re-estimation.

The probability, $b_j(\boldsymbol{k})$, of observing some particular feature vector, *k,* when the model is in state *j* can be derived as the probability of the model being in state *j* and observing ***k,*** divided by the probability of the model being in state *j*. In order to take into account the complete set of training examples of the word, we need to sum both the numerator and the denominator over all frames of all examples. Hence, assuming *E* examples of the word, the re-estimate for the emission probability is given by:

$$\bar{b}_j(\boldsymbol{k}) = \frac{\sum_{e=1}^{E} \sum_{\{t : \boldsymbol{y}_{te} = \boldsymbol{k},\, t = 1,2,\ldots,T_e\}} \gamma_j(t,e)}{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_j(t,e)} . \tag{9.20}$$

In Equation (9.20), quantities for the e^{th} example of the word are denoted by T_e for the number of frames in the example and $\boldsymbol{y}_{te}$ for the feature vector at the t^{th} frame of the example, with $\gamma_j(t, e)$ being used for the value of $\gamma_j(t)$ for the e^{th} example.

The denominator in Equation (9.20) is the sum of the individual probabilities of being in state *j* for each frame time, given the complete set of training data, and is sometimes referred to as the **state occupancy**. In some publications, the term **count** is also used when referring to this quantity. Although it is in fact a sum of probabilities, because it has been summed over the complete data set it is equivalent to the expected number, or count, of frames for which the state is occupied (although it will not in general be an integer number of frames).

In order to re-estimate the transition probabilities, we need to calculate the probability of a transition between any pair of states. This calculation is basically straightforward, but care needs to be taken to treat the start and end of the word correctly[3]. In the following explanation, transitions from the initial state and to the final state will be treated separately from transitions between emitting states.

Returning for the moment to considering only a single example of the word, let us define $\xi_{ij}(t)$ to be the probability that there is a transition from state i to state j at time *t,* given that the model generates the whole sequence of feature vectors representing the example of the word:

$$\xi_{ij}(t) = \frac{\alpha_i(t) a_{ij} b_j(\boldsymbol{y}_{t+1}) \beta_j(t+1)}{\alpha_F(T)} \quad \text{for } 1 \le t < T . \tag{9.21}$$

[3] The details of the equations given here apply to the use of special initial and final states and there will be slight differences if, for example, the model is allowed to end in any state (as in some publications).

Equation (9.21) can be applied to calculate the probability of a transition between any pair of emitting states at frame times starting from t=1 up until t=T-1. For the final frame, t=T, there cannot be a transition to another emitting state and the only possible transition is to the final state, F, with probability $\xi_{iF}(T)$, thus:

$$\xi_{iF}(T) = \frac{\alpha_i(T)a_{iF}}{\alpha_F(T)} . \tag{9.22}$$

For the initial state, we need to calculate the probability of a transition to each of the emitting states. This transition from the initial state is only possible at the start of the word, before any observations have been generated. If we regard this time as being t=0 then, given that the model must start in state I, another special instance of Equation (9.21) can be derived for all transitions out of state I, thus:

$$\xi_{Ij}(0) = \frac{a_{Ij}b_j(y_1)\beta_j(1)}{\alpha_F(T)} . \tag{9.23}$$

The total probability of a transition between any pair of states i and j is obtained by summing the values of $\xi_{ij}(t)$ over all frames for which the relevant transition is possible. Dividing this quantity by the total probability γ_i of occupying state i gives the re-estimate for the transition probability a_{ij}. Assuming E examples of the word, for a transition between any two emitting states we have:

$$\overline{a}_{ij} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e-1}\xi_{ij}(t,e)}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_i(t,e)} \qquad \text{for } 1 \le i, j \le N , \tag{9.24}$$

where $\xi_{ij}(t, e)$ denotes the value of $\xi_{ij}(t)$ for the eth training example. Note that the summation of $\xi_{ij}(t, e)$ over time only includes frames up until time T_e-1. The last frame is not included as it cannot involve a transition to another emitting state, and so by definition the value of $\xi_{ij}(T, e)$ is zero for all pairs of emitting states.

Transitions from an emitting state to the final state F can only occur at time T_e and so the transition probability a_{iF} may be re-estimated as:

$$\overline{a}_{iF} = \frac{\sum_{e=1}^{E}\xi_{iF}(T_e,e)}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_i(t,e)} \qquad \text{for } 1 \le i \le N . \tag{9.25}$$

Transitions from the initial state I can only occur at the start (time t=0), when the model *must* be in state I, so $\gamma_I(0, e)$=1 for all examples and hence:

$$\overline{a}_{Ij} = \frac{\sum_{e=1}^{E}\xi_{Ij}(0,e)}{E} \qquad \text{for } 1 \le j \le N . \tag{9.26}$$

The use of forward and backward probabilities to re-estimate model parameters is usually known either as the **forward-backward algorithm** or as the **Baum-Welch algorithm**. The second name in “Baum-Welch” recognizes the fact that Lloyd Welch was working with Baum on this subject in the early 1960s.

After re-estimation using the Baum-Welch algorithm, the probability of the training data given the new set of models is guaranteed to be higher than the probability for the previous model set, except at the **critical point** at which a local optimum has been reached and therefore the models (and hence the probability) are unchanged. The procedure can thus be repeated in an iterative manner until the difference between the new and old probabilities is sufficiently small that the training process can be regarded as being close enough to its local optimum.

It can be seen from the expression of Equations (9.24), (9.25) and (9.26) using the quantities defined in Equations (9.21), (9.22) and (9.23) that, if any of the a_{ij} are initially given values of zero, their re-estimated values will also always be zero. Setting initial values of some transition probabilities to zero is thus a convenient way of constraining the structure of the word model to prevent it from producing intrinsically implausible state sequences. For example, it would not seem reasonable to allow the model to occupy a state early in the word, and then return to it after having been through several succeeding states. The sequence possibilities in Figure 9.1 are very limited, only allowing three non-zero values of a_{ij} for any state *i,* yet this structure is very plausible as a word model. Constraining the possible state sequences by setting most of the initial values of the transition probabilities to zero has the added benefit of greatly reducing the computation required for both recognition and training.

Model initialization issues, including the choice of initial conditions for the emission p.d.f.s, will be discussed in more detail later on in this chapter.

9.5.3 Viterbi training

It is also possible to re-estimate the model parameters using only the most likely path through the states, as given by the Viterbi algorithm. The calculations are substantially simplified by just considering a single path. For any frame of input data the probability of a state being occupied can only be unity or zero, depending on whether that state is on the path. The most likely path can be found by calculating the values of a^$_j$*(t)* for all states and frames to the end of the word using Equation (9.11), and then tracing back from the final state in the same way as for the DTW method described in Chapter 8. In contrast to Baum-Welch re-estimation, the backward probabilities are not required.

Having identified the most likely path, each input frame will have been allocated to a single state to provide a state-level segmentation of the training data. It will therefore be known which state produced each observed feature vector, and also which states preceded and followed each state along the path. For the re-estimation it is then only necessary, for all examples of each training word, to accumulate the statistics of the feature vectors that occur for each occupied state, and of the transitions between states along the most likely path. Using the identified path, there will need to be counts of the following events, totalled over all *E* examples of the word:

i. the number of frames for which each state gives rise to each of the possible feature vectors, with the count for state *j* and feature vector *k* being denoted by $n_j(y_t=k)$;
ii. the number of frames for which a transition occurs between each pair of states, which for transitions between states i and j will be denoted by n_{ij};
iii. the number of occasions for which each state is occupied for the first frame of each example of the word, which for state *j* will be denoted by n_{Ij};
iv. the number of occasions for which each state is occupied for the last frame of each example of the word, which for state *i* will be denoted by $n_{iF;}$
v. the number of frames for which each state is occupied, which will be denoted by n_i and n_j for states *i* and j respectively.

The re-estimation formulae are then simply given by:

$$\bar{b}_j(\boldsymbol{k}) = \frac{n_j(\boldsymbol{y}_t = \boldsymbol{k})}{n_j}, \tag{9.27}$$

$$\bar{a}_{ij} = \frac{n_{ij}}{n_i} \quad \text{for all pairs of emitting states, } 1 \le i, j \le N, \tag{9.28}$$

$$\bar{a}_{iF} = \frac{n_{iF}}{n_i} \quad \text{for all } i \text{ such that } 1 \le i \le N, \tag{9.29}$$

$$\bar{a}_{Ij} = \frac{n_{Ij}}{E} \quad \text{for all } j \text{ such that } 1 \le j \le N. \tag{9.30}$$

Note that the above re-estimation equations for Viterbi training are in fact equivalent to the corresponding Baum-Welch equations (9.20, 9.24, 9.25, 9.26) with the values of all the frame-specific state occupancy probabilities *(?j(t, e),* etc.) set either to one or to zero, depending on whether or not the relevant states are occupied at the given frame time. As with the Baum-Welch re-estimation, the Viterbi training procedure (determination of the most likely state sequence followed by estimation of the model parameters) can be applied in an iterative manner until the increase in the likelihood of the training data is arbitrarily small.

Because the contribution to the total probability is usually much greater for the most likely path than for all other paths, an iterative Viterbi training procedure usually gives similar models to those derived using the Baum—Welch recursions. However, the Viterbi method requires much less computation and it is therefore often (and successfully) adopted as an alternative to full Baum-Welch training.

9.6 VECTOR QUANTIZATION

In the discussion above it was assumed that the data used for training the models include a large enough number of words for reliable values to be obtained for all the parameters. For any statistical estimation to give sensible results it is obvious that the total number of data items must be significantly larger than the number of separate parameters to be estimated for the distribution. If the number of possible feature vectors is very large, as a result of many possible values for each of several individual features, many feature vectors will not occur at all in a manageable amount of training data. In consequence all the generation probabilities for these feature vectors will be estimated as zero. If such a feature vector then occurred in the input during operational use of the recognizer, recognition would be impossible.

The multi-dimensional feature space for any practical method of speech analysis is not uniformly occupied. The types of spectrum cross-section that occur in speech signals cause certain regions of the feature space, for example those corresponding to the spectra of commonly occurring vowels and fricatives, to be highly used, and other regions to be only sparsely occupied. It is possible to make a useful approximation to the feature vectors that actually occur by choosing just a small subset of vectors, and replacing each measured vector by the one in the subset that is ‘nearest’ according to some suitable distance metric. This process of **vector quantization (VQ)** is also used in systems for efficient speech coding (see Section 4.3.5).

Setting up a vector quantizer usually involves first applying a **clustering** algorithm to group similar vectors together, then choosing a representative quantized vector for each cluster. The performance of such a quantizer depends on the number of different vectors and how they are chosen, but the details of these decisions are outside the scope of this book. It is, however, clear that if a fairly small **codebook** of vectors is chosen to represent the well-occupied parts of the feature space, all of these quantized vectors will occur frequently in a training database of moderate size. For each model state it will thus be possible to obtain good estimates for the probability of all feature vectors that are likely to occur.

Even after vector quantization, a fully trained model for a particular word will often have some feature vectors that are given zero probability for all states of the word. For example, the word “one” would not be expected to contain any examples of a feature representing the typical spectrum of an [s] sound. It is, however, important not to allow the probabilities to remain exactly at zero. Otherwise there is the danger of error on an input word that matches fairly well to the properties of one of the models except for just one non-typical frame that is represented by a zero-probability feature vector. In such a case the model will yield zero probability for that sequence of vectors, and the recognizer will therefore not be able to choose the correct word. A simple solution is to replace the zero value by a very small number. The model will then yield a low probability of generating the observed features, but if the rest of the word is sufficiently distinctive even this low value can be expected to be greater than the probability of generating the same set of features from any of the competing models. Better estimates for the probability of an unseen feature vector can be obtained by using a measure of distance from the vectors that are observed for the word, so that the unseen vector is given a higher probability if it is similar to those vectors which do occur in the training examples.

9.7 MULTI-VARIATE CONTINUOUS DISTRIBUTIONS

Vector quantization involves an approximation which unavoidably loses some information from the original data, and any method for estimating the probability of an unseen feature vector will inevitably be somewhat *ad hoc*. These limitations associated with discrete distributions can be overcome by representing the distribution of feature vectors by some suitable parametric description. Provided that an appropriate parametric distribution can be found for describing the true distribution of the features, a useful estimate can be computed for the probability of *any* feature vector that may occur in the training and recognition processes.

Many natural processes involve variable quantities which approximate reasonably well to the **normal** (or **Gaussian**) distribution. The normal distribution has only two independently specifiable parameters, the **mean,** μ, and the **standard deviation,** σ. For a quantity *x,* the probability density, ϕ *(x),* is given by:

$$\phi(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right). \tag{9.31}$$

When quantities are distributed normally, this simple mathematical description of the distribution makes it possible to calculate the probability of the quantity lying in any range of values provided the mean and standard deviation of the distribution are known. To calculate the probability of one particular value (i.e. a measured acoustic feature vector) occurring, we need to consider the limiting case in which the size of the interval for the range of values is infinitesimally small.

The definition of the continuous probability density function, ϕ*(x),* of a variate, *x,* is such that the probability of an observation lying in an infinitesimal interval of size *dx* centred on *x* is ϕ*(x)dx,* and is thus infinitesimally small. However, if continuous probability density functions are used instead of discrete probability distributions in the HMM equations given in Sections 9.3 to 9.5, the computation will still give the correct relative likelihoods of the different words, as the infinitesimal interval, *dx,* is common to all probability calculations. The probability of observing the features, ***P(Y),*** independently of which word is spoken, is also affected in the same way by the size of *dx*. The probability of the word given the features is therefore still correctly given by the formula expressed in Equation (9.1), even if these probability densities are used instead of actual probabilities for ***P(Y)*** and ***P(Y****|w).* Although their theoretical interpretations are different, it is thus equally suitable to use either discrete or continuous probability distributions in the calculations of word probability and in parameter re-estimation. In the following discussion of continuous distributions, it will be convenient to continue to use the term “probability” even where the quantities are, strictly speaking, probability densities.

9.8 USE OF NORMAL DISTRIBUTIONS WITH HMMS

It is obvious that many naturally occurring quantities are not normally distributed. For example, speech intensity measured over successive fixed time intervals of, say, 20 ms during continuous speech will certainly not approximate to a normal distribution because it clearly has a hard limit of zero during silences, will be low for much of the time during weak sounds, but will go up to quite high values during more intense vowels. The intensity on a logarithmic scale would have a more symmetrical distribution, which might be nearer to normal, but in this case the low-level end of the distribution will be very dependent on background noise level.

Normal distributions usually fit best to measurements which can be expected to have a preferred value, but where there are various chance factors that may cause deviation either side of that value, with the probability progressively decreasing as the distance either side of the preferred value increases. Thus it might be reasonable to use a normal distribution to approximate a distribution of speech features which are derived from the same specific part of a specific word spoken in the same way by the same person. When it is assumed that features are normally distributed for each state of an HMM, the distributions are often termed **single Gaussian**.

When different speakers are combined in the same distribution the departures from normal will be greater, and for different regional accents there is a fairly high probability that the distribution will be multi-modal, and therefore much less suitable for modelling as a normal distribution. However, when multi-modal distributions are likely, as is the case with many current speech recognition systems, it is now almost universal to model the distributions with a weighted sum, or **mixture,** of several normal distributions with different means and variances (usually referred to as **Gaussian mixtures**). Provided that there is a sufficient number of mixture components, any shape of distribution can be approximated very closely. This characteristic of sums of Gaussian distributions, combined with the attractive mathematical properties of the Gaussian itself, is largely responsible for their widespread and successful use for describing emission probability distributions in HMM-based speech recognition systems.

The theory underlying the use of mixture distributions is a straightforward extension of the single-Gaussian case and will be discussed in Section 9.10, after first introducing the probability calculations and model parameter re-estimation equations using single Gaussian distributions.

9.8.1 Probability calculations

The features are multi-dimensional and so, in the case of single-Gaussian distributions, they will form a multi-variate normal distribution. In general the features may not vary independently, and their interdependence is specified by a **covariance matrix**. The entries along the main diagonal of this matrix represent the variance of each feature, while the remaining entries indicate the extent to which the separate feature distributions are correlated with each other.

Let us first consider the output probability $b_j(\boldsymbol{y})$ *f*or the j^{th} state, where $\boldsymbol{y}$ is a single feature vector. Assume that the column vector $\boldsymbol{y}$ comprises K features, $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k$. Let $\boldsymbol{\mu}_j$ be the column vector of means, $\mu_{j1}, \mu_{j2}, \ldots, \mu_{jk}$, and Σ_j be the covariance matrix for the distribution of features associated with that state. The definition of the multi-variate normal distribution gives the output probability compactly in matrix notation:

$$b_j(\boldsymbol{y}) = \frac{1}{|\boldsymbol{\Sigma}_j|^{1/2}(2\pi)^{K/2}} \exp\left(\frac{-(\boldsymbol{y}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_j)}{2}\right), \tag{9.32}$$

where $|\Sigma_j|$ is the determinant of Σ_j and $(\boldsymbol{y}\text{-}\boldsymbol{\mu}_j)^T$ is the transpose of $(\boldsymbol{y}\text{-}\boldsymbol{\mu}_j)$. In the special case when the features are uncorrelated, the covariance matrix becomes zero except along its main diagonal (and is therefore often referred to as a **diagonal covariance matrix**). The probability of a feature vector then reduces to a product of probabilities given by the univariate distributions of the separate features:

$$b_j(\mathbf{y}) = \prod_{k=1}^{K} \frac{1}{\sigma_{jk}\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{y_k - \mu_{jk}}{\sigma_{jk}} \right)^2 \right), \tag{9.33}$$

where y_k is the k^{th} feature of $\mathbf{y}$, and μ_{jk} and s_{jk} are the mean and standard deviation of the distribution of the k^{th} feature for state *j*.

Equation (9.33) is evidently computationally simpler than Equation (9.32). The extent of the computational saving provides a strong motivation for choosing methods of speech analysis for which the features are substantially uncorrelated. Some of these methods will be described in Chapter 10. Most current speech recognition systems adopt such a method and use diagonal covariance matrices. Having defined an expression for the emission probability in terms of the distribution parameters, recognition can be performed in the same way as when using discrete distributions. Thus, in the case of the Viterbi algorithm, the new definition of $b_j(\mathbf{y})$ is simply used in Equations (9.11) and (9.12).

9.8.2 Estimating the parameters of a normal distribution

When modelling emission probabilities with continuous distributions, the training task is to optimize the parameters of the feature distribution model, rather than the probabilities of particular feature vectors. If we had a set of *T* feature vectors that were known to correspond to state7, then the maximum-likelihood estimates for the parameters of a normal distribution are easily calculated. The mean vector $\hat{\boldsymbol{\mu}}_j$ is equal to the average of all the observed vectors (i.e. the sample mean), and the covariance matrix $\hat{\boldsymbol{\Sigma}}_j$ is obtained based on the deviation of each of the observed vectors from the estimated mean vector (i.e. the sample covariance matrix):

$$\hat{\boldsymbol{\mu}}_j = \frac{1}{T}\sum_{t=1}^{T} \mathbf{y}_t\ , \tag{9.34}$$

$$\hat{\boldsymbol{\Sigma}}_j = \frac{1}{T}\sum_{t=1}^{T} (\mathbf{y}_t - \hat{\boldsymbol{\mu}}_j)(\mathbf{y}_t - \hat{\boldsymbol{\mu}}_j)^T\ . \tag{9.35}$$

Obviously, in the case of HMMs, the state sequence is not known, but the standard methods for estimating mean and covariance given in Equations (9.34) and (9.35) can be extended for use in either Baum-Welch or Viterbi re-estimation procedures, as explained below.

9.8.3 Baum-Welch re-estimation

For the Baum-Welch algorithm, the parameters are re-estimated using contributions from all frames of all the *E* examples of the word in the training data. Each contribution is weighted by the probability of being in the state at the relevant frame time, as given by Equation (9.19). Therefore the re-estimates of the mean vector $\boldsymbol{\mu}_j$ and the covariance matrix Σ_j associated with state j are given by:

$$\bar{\boldsymbol{\mu}}_j = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)\boldsymbol{y}_{te}}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)}, \tag{9.36}$$

$$\bar{\boldsymbol{\Sigma}}_j = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)\ (\boldsymbol{y}_{te}-\bar{\boldsymbol{\mu}}_j)(\boldsymbol{y}_{te}-\bar{\boldsymbol{\mu}}_j)^T}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_j(t,e)}, \tag{9.37}$$

where $\boldsymbol{y}_{te}$ is the feature vector for the t^{th} frame of the e^{th} example of the word. Just as for discrete emission p.d.f.s, it can be shown that iterative application of the above formulae leads to a locally optimum solution. Baum's (1972) analysis included a proof for univariate normal distributions, which was later generalized by Liporace (1982) to a wider class of distributions, including multi-variate normal distributions.

Note that Equation (9.37) for re-estimating the covariance matrix is based on deviation of observed vectors from the *re-estimated* mean vector $\bar{\boldsymbol{\mu}}_j$. In practice, when accumulating the contributions for covariance re-estimation it is easier to use the current estimate $\boldsymbol{\mu}_j$ instead of the (yet to be computed) new value $\bar{\boldsymbol{\mu}}_j$. It is then straightforward to correct for the difference between the old and new mean values at the end of the calculation.

9.8.4 Viterbi training

For Viterbi re-estimation, the requirement is just to use the state-level segmentation obtained from the most likely path according to the current set of models as the basis for collecting the statistics needed to apply Equations (9.34) and (9.35) for each model state. The statistics for state j are therefore gathered over all examples of the word using all frames for which state j is occupied. Using s_{te} to denote the state occupied at frame t of example e, the re-estimation formulae are as follows:

$$\bar{\boldsymbol{\mu}}_j = \frac{1}{n_j}\sum_{e=1}^{E}\sum_{t \ni s_{te}=j} \boldsymbol{y}_{te}\ , \tag{9.38}$$

$$\bar{\boldsymbol{\Sigma}}_j = \frac{1}{n_j}\sum_{e=1}^{E}\sum_{t \ni s_{te}=j} (\boldsymbol{y}_{te}-\bar{\boldsymbol{\mu}}_j)(\boldsymbol{y}_{te}-\bar{\boldsymbol{\mu}}_j)^T, \tag{9.39}$$

where, as in Section 9.5.3, n_j is the number of frames for which state j is occupied.

9.9 MODEL INITIALIZATION

There must be enough states in the model to capture all the acoustically distinct regions in the word. For example, a typical word of two or three syllables could need around 10–20 states to model the acoustic structure adequately. Because the training process only finds a local optimum, the initial estimates for the model parameters can have a strong influence on the characteristics of the final set of trained models. It is very important to give careful consideration to how the model parameters, including both transition and emission probabilities, should be initialized before training. The trained model for each word needs to capture the spectral and temporal characteristics of all spoken utterances of that word while at the same time, in order to minimize recognition errors, it must be a constraining model which does not allow inappropriate sequences of states for the word.

In an HMM, the probability of a path through the model is computed on a frame-by-frame basis and therefore cannot take into account any of the previous states occupied other than the one at the immediately preceding time frame. Thus, if a model allows many different transitions from each state, recognition errors can result if a sequence of frames gives a good acoustic match even if the complete state sequence is very inappropriate for a genuine example of the word. Even the limited degree of flexibility included in the model structure shown in Figure 9.1 can cause problems if used throughout a word.

The dangers associated with allowing flexibility of transitions within a word model are such that most current uses of HMMs only allow a very restricted set of possible transitions, by initializing most of the transition probabilities to zero. A popular HMM structure for speech recognition uses a left-to-right topology with the probability of all transitions set to zero except those to the next state or returning to the current state (i.e. as for Figure 9.1 but omitting the ‘skip’ transitions). If this model structure is used to represent a word, the word will be modelled as a sequence of acoustic regions which can vary in duration but which must always all occur and always in the same fixed order. With this strong temporal constraint provided by the model structure, re-estimation (using either the Baum-Welch or the Viterbi approach) can give a useful local optimum even with a simple initialization approach for the emission probabilities. One popular strategy is to start with a uniform segmentation of each training example, with the number of segments being equal to the number of states in the model. This segmentation can then be used to compute the required statistics for each state emission probability, with the allowed transition probabilities of all emitting states initialized to identical values.

In the case of Baum-Welch training, an even simpler initialization strategy may be used for the parameters of discrete or normal distributions. For this method, sometimes called a **flat start** (Knill and Young, 1997), all emission p.d.f.s for all states are set to average values computed over the entire training set, in addition to using identical transition probabilities for a limited set of allowed transitions. Thus all permitted paths through the model start with equal probability and the training algorithm is left to optimize the parameters from this neutral starting position with constraints imposed by the model structure. This approach has been found to work well if there are several utterances for each model unit (e.g. Paul and Martin, 1988).

An important advantage of the initialization approaches described above is that the training process can be carried out completely automatically, without requiring any pre-segmented data. Alternatively, if there are any data available for which suitable state boundaries are ‘known’ (for example, the boundaries could be marked by hand for a small subset of the training corpus), this segmentation can be used as the basis for initializing some or all of the model parameters.

If all models use a restricted structure that only allows transitions back to the same state or on to the next state, it is implicitly assumed that such a model structure is appropriate for representing all words. There are many cases in human language where pronunciation varies from occasion to occasion, even for one speaker. The variations may be at the phonemic level: for example, in the word “seven” many speakers often omit the vowel from the second syllable and terminate the word with a syllabic [n]. Allophonic variations can also occur: for example, in words ending in a stop consonant, the consonant may or may not be released. If a word with alternative pronunciations is represented by a single sequence of states with the model structure described above, some states will have to cope with the different pronunciations, and so their p.d.f.s will need to be multi-modal to model the distributions well. In these cases a normal distribution will not be suitable, and Gaussian mixtures will be essential for good modelling of the data.

A rather different approach is to explicitly model alternative pronunciations as alternative state sequences, using either whole-word or sub-word models. Initialization then involves choosing a constraining topology separately for each word model to take into account the possible phonetic structure of the word and its expected variation. It will thus be necessary to decide on the number of states required to represent each phonetic event and on the allowed transitions between the states, with state skips being allowed only where a particular phonetic event is sometimes omitted. For this approach to work it is essential that the emission p.d.f.s of the models are initialized with values roughly appropriate for the phonetic events expected for each state, because otherwise the training frames may not be allocated to the states in the intended way. Such models can be initialized by carefully hand-labelling a few examples of the training words in terms of state labels, and collecting the statistics of these data to initialize the emission p.d.f.s. However, the whole method requires a lot of skilled human intervention, and a simpler model topology is usually adopted, with any limitations in this approach being addressed by using Gaussian mixtures for the p.d.f.s. Methods used for including some simple provision for alternative pronunciations will be considered further in Chapter 12.

9.10 GAUSSIAN MIXTURES

9.10.1 Calculating emission probabilities

The expression for the emission probability defined in Equation (9.32) is easily extended to include a weighted sum of normal distributions, where each component distribution has a different mean and variance. We will use the notation $N(y;\ \mu,\ \Sigma)$ to represent the probability density of the observed vector y given a normal distribution with mean vector μ and covariance matrix Σ. Thus, for an emission p.d.f. defined according to a Gaussian mixture distribution, the emission probability given by the m^{th} component for state j is:

$$b_{jm}(\boldsymbol{y}) = N\left(\boldsymbol{y}\,;\boldsymbol{\mu}_{jm},\boldsymbol{\Sigma}_{jm}\right)\ . \tag{9.40}$$

The total emission probability for a distribution with *M* components is defined as:

$$b_j(\boldsymbol{y}) = \sum_{m=1}^{M} c_{jm} b_{jm}(\boldsymbol{y}) \, , \tag{9.41}$$

where c_{jm} denotes the weight of the m^{th} mixture component for state *j.* The mixture component weights can only take positive values, c_{jm}=0, and must sum to 1:

$$\sum_{m=1}^{M} c_{jm} = 1 \; . \tag{9.42}$$

In the special case where there is only one mixture component, the emission probability specified by Equation (9.41) is defined in terms of a single Gaussian distribution with weight equal to 1 and is therefore equivalent to Equation (9.33).

Once the parameters of multiple-component mixture distributions have been trained, Equation (9.41) can be used as the basis for the recognition calculations in exactly the same way as with the simpler emission p.d.f.s that we have already discussed. Parameter estimation for mixture distributions requires more detailed consideration, and is discussed in the following sections. Firstly we will assume that initial estimates are available and address the re-estimation problem, before considering ways of obtaining suitable initial estimates in Section 9.10.4.

9.10.2 Baum-Welch re-estimation

Assuming that initial estimates are available for all the parameters of all the *M* components of a Gaussian mixture representing the emission p.d.f. for state j, Baum-Welch re-estimation can be used to find new estimates for these parameters, c_{jm}, $\boldsymbol{\mu}_{jm}$ and Σ_{jm}. When using Gaussian mixtures, the contribution from each observation $\boldsymbol{y}_t$ needs to be weighted by a probability that is specific to the mixture component *m.* By analogy with the quantity ?$_{j}$*(t)* which was introduced in Section 9.5.2, let us define ?$_{jm}$(t) *to* be the probability of being in state *j* at time *t* and using component *m* to generate $y_{t,}$ given that the model generates the whole sequence of *T* feature vectors representing an example of the word.

$$\gamma_{jm}(t) = \frac{\sum_{i=1}^{N} \alpha_i(t-1) a_{ij} c_{jm} b_{jm}(\boldsymbol{y}_t) \beta_j(t)}{\alpha_F(T)} \tag{9.43}$$

Now, if we have *E* examples of the word, summing the values of ?$_{jm}$*(t, e)* over all frames of all examples gives the total probability for the m^{th} component of state *j* generating an observation. Dividing this quantity by the corresponding sum of γ_j*(t, e)* terms gives the re-estimate for the mixture component weight c_{jm}:

$$\bar{c}_{jm} = \frac{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}(t,e)}{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_j(t,e)} \; . \tag{9.44}$$

The re-estimation equations for the mean vector and covariance matrix are the same as for the single-Gaussian case given in Equations (9.36) and (9.37), but using the component-specific state occupation probabilities $?_{jm}(t, e)$:

$$\bar{\boldsymbol{\mu}}_{jm} = \frac{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}(t,e) \boldsymbol{y}_{te}}{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}(t,e)} , \tag{9.45}$$

$$\bar{\boldsymbol{\Sigma}}_{jm} = \frac{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}(t,e)(\boldsymbol{y}_{te} - \bar{\boldsymbol{\mu}}_{jm})(\boldsymbol{y}_{te} - \bar{\boldsymbol{\mu}}_{jm})^T}{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{jm}(t,e)} . \tag{9.46}$$

Juang (1985) extended Liporace's (1982) analysis to show that iterative application of the re-estimation formulae leads to a locally optimum solution when emission p.d.f.s are defined in terms of sums of normal distributions.

9.10.3 Re-estimation using the most likely state sequence

The use of a Gaussian mixture to represent the HMM emission p.d.f. incorporates another 'hidden' element in the model, as it is not known from the observations which mixture component generated each observation. The probability calculation in Equation (9.41) uses the total probability taking into account all the mixture components that could have produced the observation. As a result the Baum-Welch re-estimation formulae in Equations (9.44) to (9.46) use probabilities not only of state occupancy but also of mixture components. The formulae can be simplified if the state sequence is known, but the situation is more complex than for the single-Gaussian case because the mixture components are still unknown. Thus the state sequence alone does not lead to an analytic solution for these emission p.d.f.s.

One option is to retain the EM algorithm for estimating the parameters of the mixture distribution. Equations (9.44) to (9.46) can be simplified accordingly: the summations are now just over those frames for which state *j* is occupied, and for each frame the component-dependent state occupation probability $?_{jm}(t, e)$ simplifies to a component-dependent emission probability $c_{jm} b_{jm}(y_{te})$. (The total state occupation probability $?_j(t, e)$ is replaced by the emission probability $b_j(y_{te})$.)

Alternatively, to estimate the distribution parameters without requiring an EM algorithm, each observation must be assigned to a single mixture component. This assignment can be achieved by using a clustering procedure to divide the observed feature vectors corresponding to any one model state into a number of groups equal to the number of mixture components for that state. ***K*-means clustering** is a well-established technique for dividing a set of vectors into a specified number of classes in order to locally minimize some within-class distance metric, and was originally applied to vector quantization (see Section 9.6). The term **segmental**

***K*-means** is often used to refer to the use of *K*-means clustering in conjunction with a Viterbi alignment procedure to identify the state-level segmentation. After clustering, each frame will be labelled, not only with the state that was occupied, but also with the mixture component that generated the observation. The re-estimation formula for the weight associated with the m^{th} mixture component of state j is then given by:

$$\bar{c}_{jm} = \frac{n_{jm}}{n_j}, \tag{9.47}$$

where n_{jm} represents the number of frames for which state j was occupied and mixture component m generated an observation. Using s_t to denote the state occupied and x_t to denote the mixture component used at time t, the re-estimation formulae for the mean feature vector and covariance matrix are straightforward extensions of the single-Gaussian case (Equations (9.38) and (9.39)), as follows:

$$\bar{\boldsymbol{\mu}}_{jm} = \frac{1}{n_{jm}} \sum_{e=1}^{E} \sum_{t \ni s_t = j,\, x_t = m} \mathbf{y}_{te}, \tag{9.48}$$

$$\bar{\boldsymbol{\Sigma}}_{jm} = \frac{1}{n_{jm}} \sum_{e=1}^{E} \sum_{t \ni s_t = j,\, x_t = m} (\mathbf{y}_{te} - \bar{\boldsymbol{\mu}}_{jm})(\mathbf{y}_{te} - \bar{\boldsymbol{\mu}}_{jm})^T. \tag{9.49}$$

9.10.4 Initialization of Gaussian mixture distributions

The segmental *K*-means procedure outlined above uses an initial set of models to obtain the state-level segmentation, but does not rely on any existing estimates for the mixture components. It therefore provides a convenient method for initializing the parameters of HMMs using mixture distributions. If no models are available, the process can even be started from a uniform segmentation, as described in Section 9.9. Once initial estimates have been obtained for all the mixture components, the estimates can be refined using further iterations of the segmental *K*-means procedure. At this point the models could be used for recognition, but they can be trained further using full Baum-Welch re-estimation, or even using the EM algorithm to update the mixture parameters without changing the segmentation.

A segmental *K*-means procedure is often used to initialize mixture models prior to Baum-Welch training. However, this approach requires the number of mixture components to be decided in advance. An alternative is to start with trained single-Gaussian models and to incrementally increase the number of mixture components using a method often referred to as **mixture splitting**. Starting with a single Gaussian distribution for the emission p.d.f., a two-component mixture model is initialized by duplicating the parameters of the original distribution and perturbing the means by a small amount in opposite directions (typically±0.2 standard deviations). The variances are left unchanged and the mixture weights are set to 0.5 for both components. The means, variances and mixture weights are all re-estimated, and the mixture-splitting procedure is then applied to the component with the largest weight (setting the weights of both new components to half the value for the component from which they were derived). The model parameters are re-estimated again, and so on until the desired level of complexity is reached.

For a given number of mixture components, Young and Woodland (1993) reported that an iterative mixture-splitting training procedure with Baum-Welch re-estimation gave similar results to using segmental K-means followed by Baum-Welch training. However, a useful advantage of the mixture-splitting approach is that the number of mixture components can be chosen for each state individually according to some objective criterion based on how well the data are modelled. Examples of useful criteria for deciding on the number of components are the magnitude of the increase in training-data likelihood from adding a new component, or the quantity of training data available for the model concerned. This flexibility of mixture modelling is particularly beneficial for modelling large vocabularies; its use will be discussed further in Chapter 12.

9.10.5 Tied mixture distributions

Increasing the number of components used in a Gaussian mixture distribution allows for greater flexibility in the shapes of distributions that can be modelled, but a larger quantity of training data is required to ensure that the parameters are trained robustly. In any practical recognizer there are often only limited data available for training each model, which imposes limitations on the number of state-specific mixture components that can be included. However, similarities between different speech sounds are such that many of the component distributions will be similar for several different states. One straightforward way of taking advantage of these similarities to provide more data for training the model parameters is to use the same Gaussian distributions to represent *all* the states of all the models, with only the mixture weights being state-specific. Thus the distribution parameters are tied across the different states, and this type of model is often referred to as a **tied mixture** (Bellegarda and Nahamoo, 1990). The term **semi-continuous HMM** has also been used (Huang and Jack, 1989), because the one set of continuous distribution parameters for all states can be regarded as an alternative to the VQ-generated codebook used with discrete emission probabilities.

When using tied mixtures, the emission probability $b_j(\mathbf{y})$ for any one state j is calculated in the same way as for Equation (9.41), but although the mixture weights c_{jm} are state-specific, the $b_{jm}(\mathbf{y})$ terms will be the same for all states.

Using the new definition of the emission probability, re-estimation formulae can be derived for the mean μ_m and covariance matrix Σ_m of the m^{th} component (the re-estimation of the mixture weights c_{jm} is unchanged). For example, tied-mixture versions of the Baum-Welch formulae in Equations (9.45) and (9.46) are as follows:

$$\bar{\boldsymbol{\mu}}_m = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\sum_{j=1}^{N}\gamma_{jm}(t,e)\mathbf{y}_{te}}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\sum_{j=1}^{N}\gamma_{jm}(t,e)}, \tag{9.50}$$

$$\bar{\boldsymbol{\Sigma}}_m = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\sum_{j=1}^{N}\gamma_{jm}(t,e)(\boldsymbol{y}_{te}-\bar{\boldsymbol{\mu}}_m)(\boldsymbol{y}_{te}-\bar{\boldsymbol{\mu}}_m)^T}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\sum_{j=1}^{N}\gamma_{jm}(t,e)} . \tag{9.51}$$

Thus the only difference from the original (untied) mixture-distribution re-estimation formulae is that the contributions are now summed over all states as well as over all frames of all examples.

Tied mixtures have been used with some success to address the problems associated with training a large number of model parameters from a limited quantity of training data. However, although any practical system will have some model states which share many similarities, there will obviously be others which are quite different, and this characteristic will be reflected in the mixture weights for the different states. Thus rather more parameters are being tied together than is necessary, and such extensive tying may not be desirable for maximum discrimination. It is important to note that tied mixtures are just one example of the much more general concept of **parameter tying,** whereby any parameters of any model states can be tied together and the only effect on the re-estimation formulae is in the nature of the summations and in the indexing of the model parameters. The ability to tie together the parameters of HMM states is a significant factor in the success of current large-vocabulary speech recognition systems, and this use of tying is explained in Chapter 12.

9.11 EXTENSION OF STOCHASTIC MODELS TO WORD SEQUENCES

In the same way as was described for the dynamic programming methods in Chapter 8, HMMs extend easily to connected sequences of words. For recognition the word sequences can be represented by a higher-level model in which the states correspond to whole words, and the transition probabilities are the language-model probabilities (recognition using a language model will be discussed in Chapter 12).

In the case of recognition of isolated words we were not interested in the state sequences as such, but only in the likelihood of each word model emitting the observed feature vectors. When applying HMMs to connected words, however, we need to know the most likely sequence of words, so at the word level the Viterbi algorithm is necessary. The word boundary procedure is then exactly analogous to that described in Section 8.10, making use of the back-pointers to determine the word sequences.

The HMM training algorithms can also be used when the training data are spoken as natural connected sequences of words. It is not generally necessary to segment the data into the individual words prior to training. Instead, an **embedded training** approach can be used, whereby a composite model is obtained for the whole utterance by concatenating the required sequence of word models. This concatenation is very easy if special non-emitting initial and final states are used for the individual models, as it simply involves linking the final state of one word to the initial state of the next. The parameters of the composite model are trained using the same procedure that would be carried out if this composite model represented a single word. If a state occurs more than once in the composite model (i.e. if the utterance contains more than one example of any particular word), all occurrences of that state will contribute to the parameter re-estimation. Provided that each word is spoken in a variety of different contexts, embedded training is very successful (at least for a constrained left-to-right model structure), even with the simplest ‘flat’ initialization procedure of setting the parameters of all models to the same values. The ability of the HMM training framework to automatically find the patterns in the data to associate with individual models is fundamental to the successful use of HMMs for substantial recognition tasks.

9.12 IMPLEMENTING PROBABILITY CALCULATIONS

The calculation of the forward and backward probabilities for sequences of feature vectors involves multiplication of a very large number of probability components, the majority of which are much less than 1. The results will in general have very low values, and mostly will be smaller than the minimum size of floating point number that can be held in any normal computer.

One solution to the number range problem is to check the probabilities at each stage of the recursion, and to multiply them by a scale factor that will bring the numbers back into the centre of the available range. However, scale factors must be noted and taken into account in estimating the relative likelihoods that each frame of feature vectors has been generated by each word model.

An alternative way of avoiding problems with numerical underflow is to represent all probabilities in logarithmic form, so that no explicit scaling is necessary. The following sections will discuss the implementation of HMM probability calculations using logarithms of probabilities.

9.12.1 Using the Viterbi algorithm with probabilities in logarithmic form

Because the logarithmic function is monotonic and increasing, the task of maximizing a probability can be achieved by maximizing its logarithm, and the main Viterbi probability calculation given in Equation (9.11) can therefore be replaced by:

$$\hat{\alpha}_j^L(t) = \max_{\text{over } i}\left(\hat{\alpha}_i^L(t-1) + a_{ij}^L\right) + b_j^L(\boldsymbol{y}_t) \; , \tag{9.52}$$

where $\hat{\alpha}_j^L(t)$ is used for log($\hat{\alpha}_j(t)$), a_{ij}^L for log(a_{ij}) and $b_j^L(\mathrm{y}_t)$ *fo*r log(b_j (y_t)).

When using discrete emission p.d.f.s with vector quantization, the calculation of Equation (9.52) is very straightforward and can easily be made very efficient: the quantities log(a_{ij}) and log($b_j(\boldsymbol{y}_t)$) are fixed for given values of *i, j* and $\boldsymbol{y}_t$, and hence the logarithms need to be calculated just once and stored ready for use as required. The dynamic programming algorithm then only involves summations and comparisons, with no multiplications or logarithmic functions.

If normal distributions are used for the emission p.d.f.s, we must take the logarithm of the expression for the emission probability, but this is also straightforward. For example, in the case of uncorrelated normal distributions for the individual features, taking logarithms[4] of Equation (9.33) gives:

$$b_j^L(y_t) = \log(b_j(y_t)) = -\frac{K}{2}\log(2\pi) - \sum_{k=1}^{K}\log(\sigma_{jk}) - \frac{1}{2}\sum_{k=1}^{K}\left(\frac{y_{kt} - \mu_{jk}}{\sigma_{jk}}\right)^2 \qquad (9.53)$$

where we are now taking into account the fact that the observations are time-dependent, and are using the symbol y to denote the k^{th} feature at the t^{th} frame.

Comparing Equation (9^{kt}.53) with (9.33), it can be seen that use of logarithms has eliminated the need for the observation-dependent exponential operation, while the logarithmic terms are independent of the observed feature values and so can be pre-computed. Thus, while the computational load when using normal distributions is somewhat greater than for discrete emission p.d.f.s, the use of logarithms leads to a considerable computational saving as well as solving the number range problem.

9.12.2 Adding probabilities when they are in logarithmic form

When calculating emission probabilities using Gaussian mixture distributions, and for all calculations of forward and backward probabilities in Baum-Welch re-estimation, probabilities must be summed as well as multiplied and so the use of logarithms is more complicated. If we consider two probabilities, *A* and *B,* the task is to compute log*(A+B)* given log*(A)* and log*(B)*. In theory, we could exponentiate both log*(A)* and log*(B),* add them and take the logarithm. However, aside from the computational issues, the exponential operation puts the probabilities back onto a linear scale and so presents problems for the wide range of probabilities that may be encountered. This difficulty can be addressed by first rewriting log*(A+B)* thus:

$$\log(A+B) = \log(A(1+B/A)) = \log(A) + \log(1+B/A)\,. \qquad (9.54)$$

Assume that we have ordered the probabilities such that *A*=*B*. The issue is now one of evaluating the ratio *B/A,* which can be no greater than 1 and therefore the calculation will only present problems if this ratio is smaller than the smallest number which can be represented in the computer. This situation can only arise if *B* is so much smaller than *A* that it can safely be ignored by setting log*(A+B)*=log*(A)*. A procedure for finding *log(A+B)* is therefore as follows:

1. If log*(B)*>log*(A) t*hen transpose log*(A)* and log(B).
2. Find log*(B/A)* by forming log*(B)*-log*(A)*. Store this value in *C*.
3. If *C*<a suitable threshold, set *C*=0.
4. Otherwise *C*=log(l+exp(C)).
5. Add *C* to log*(A)*.

The threshold in step 3 is used to prevent underflow when taking the exponential in step 4. The smallest value to which this threshold can be set is the logarithm of the smallest number that can be represented in the computer.

The procedure described above for performing probability calculations in logarithmic form is effective and widely used. However, whenever there is the need

[4] When using normal distributions, the calculations are simplest if natural logarithms are used, and the use of natural logarithms has been assumed in Equation (9.53).

to add two probabilities, one exponential and one logarithmic operation are usually required. These operations can be avoided by using a method which allows the numbers to be added while in their logarithmic form[5]. Considering step 4 in the sequence of calculations described above, both the exponential and the logarithmic operation can be avoided by using a pre-computed look-up table to store the values of log(l+*B/A*) in terms of log*(B/A)*. Thus steps 3 and 4 can be replaced by a single table look-up operation, with log*(B/A)* as input (i.e. the value already stored in *C* at step 2). The output is log(l+*B/A*), which can again be stored as the new value of *C*. Moderate accuracy in the value of log*(A+B)* can be achieved with a small look-up table. For example, a 1% accuracy for *A*+*B* enables values of *B/A* of less than 0.01 to be ignored, and the look-up table for the larger values of *B/A* only needs entries for 115 equally spaced values of log*(B/A)*.

If the above method is implemented using a suitable scale factor for the logarithms, it is then even possible to make all the probability calculations for recognition and parameter estimation using integer arithmetic on logarithmically coded numbers. No multiplications would be required with the VQ method, and no exponential functions would be needed when using Gaussian distributions. The 1% error proposed above should have very little effect on the re-estimation, but the error could easily be reduced if necessary by using a larger look-up table.

9.13 RELATIONSHIP BETWEEN DTW AND A SIMPLE HMM

It is interesting to compare the Markov probability calculation with the cumulative distance formula for a simple asymmetric dynamic programming algorithm in which each input frame occurs exactly once in the distance calculation. If the DP uses a squared Euclidean distance metric, the recognition process can be regarded as a special case of HMM Viterbi decoding, in which the word models have one state per template frame, and the features are assumed to be normally distributed with unit variance.

To clarify this relationship, we will return to the Viterbi calculation using logarithms of probabilities, given in Equation (9.52). The value of b_j^L $(\mathbf{y}_t)$ according to an uncorrelated normal distribution is given by Equation (9.53), where we are now assuming that $\sigma_{jk}=1$ for all states *j* and for all features *k*. Hence

$$\sum_{k=1}^{K} \log(\sigma_{jk}) = 0\,,$$

and recursive calculation of $\hat{\alpha}_j^L(t)$ simplifies to:

$$\hat{\alpha}_j^L(t) = \max_{\text{over } i}\left(\hat{\alpha}_i^L(t-1) + a_{ij}^L\right) - \frac{K}{2}\log(2\pi) - \frac{1}{2}\sum_{k=1}^{K}\left(y_{kt} - \mu_{jk}\right)^2 . \tag{9.55}$$

The term *K/2* log(2π) is a constant, which will scale the likelihood calculation but will not affect the choice of optimal state sequence. Thus the only quantities that need be considered at each frame are the logarithms of the transition probabilities

[5] This method was described by Kingsbury and Rayner (1971) for a completely different application.

and the square of the Euclidean distance between the observed features at time t and the means for model state j.

As maximizing is equivalent to minimizing— , this recognition task can be regarded as one of minimizing a distance comprising— , the negative logarithm of the transition probability (which must itself be positive), *plus* the squared Euclidean distance of the features from their model mean values. Thus we have a simple DP algorithm in which the— terms are interpreted as timescale distortion penalties. Where only slopes of 0, 1, and 2 are permitted, as is the case for the HMM in Figure 9.1, the time distortion penalties for other values of slope are -log(0), and are therefore infinite.

9.14 STATE DURATIONAL CHARACTERISTICS OF HMMS

The probability of a model staying in the same state, *i,* for successive frames is determined only by the transition probability, a . The expected number of frames it will stay in state i is 1/(1-a), so a value of a =0.9 ^{ii}would be suitable for using one state to model, for example, a^{ii}steady fricative s^{ii}ound whose expected duration is 10 frames. Although the expected total duration in state i in this case is 10 frames, the most likely duration is only one frame, with a probability of 0.1. The probabilities for longer durations decrease exponentially, as shown in trace (i) of Figure 9.2(b). This distribution is often referred to as a geometric distribution because the probabilities for successive numbers of frames form a geometric progression.

For any state representing a particular phonetic event, this type of duration distribution is obviously not sensible. For any such event there will be a most probable duration, with reducing probability for both shorter and longer durations. If many more states are available, the durational characteristics of the model can be improved, but only if a long steady region is modelled by a sequence of states with very similar feature p.d.f.s and the total likelihood method is used to calculate the word probability. For example, consider a group of four identical states with a repeat probability of 0.6, as shown in trace (ii) of Figure 9.2. The expected duration for the group is 10 frames, as it is for the single state shown in trace (i). However, in the case of the group of states, the variation of probability with duration is much more appropriate for speech sounds within a word. This more realistic distribution arises because, while there is only one possible way of going through the states in the minimum number of frames, there are more possible paths for longer frame sequences. However, the improved shape of duration distribution given by this state-splitting approach relies on using total likelihood probability calculations, whereas the Viterbi algorithm is generally used for recognition.

A simple method which can be used with the Viterbi algorithm involves merely imposing a minimum and a maximum duration on state occupancy. Such duration constraints can be achieved with an easy modification to the recognition algorithm, and can give worthwhile performance benefits. Many other methods have been proposed for improving the duration characteristics of HMMs, including some that model duration distributions of each state explicitly. These methods generally give greater benefits than simple duration constraints, but at the expense of more computation and some increase in the number of model parameters.

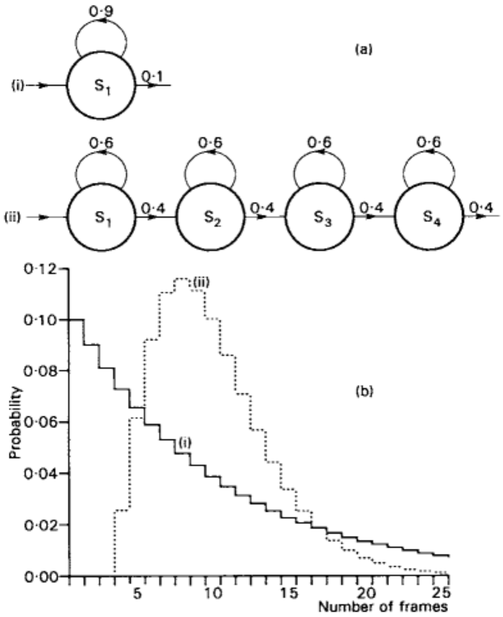


Figure 9.2 **(a)** Two arrangements of states, each with an expected occupation time of 10 frames. **(b)** Probability of occupancy of groups of states in the model sections shown in (a).

CHAPTER 9 SUMMARY

- The performance of pattern-matching speech recognizers is improved by representing typical characteristics of speech patterns in a way that also takes account of variability, which can be achieved by using a stochastic model of each word. Hidden Markov models (HMMs) represent each word as a sequence of states, with transition probabilities between each state and its permitted successors, and probability distributions defining the expected observed features for each state. A recursive formula can be used to calculate the probability that each word model will produce the observed data. The model with the highest probability is assumed to represent the correct word.
- Computation can be saved by using the Viterbi dynamic programming algorithm to calculate the probability of producing the data from only the most likely path through the states. This probability will always be less than the true probability, but the effect on recognition performance is usually very small and the Viterbi algorithm is generally adopted for HMM recognition.
- For each word model the transition probabilities and the probability distributions of the feature vectors can be found by the Baum-Welch re-estimation process. This process iteratively refines initial guesses to improve the model's representation of a set of training examples of the word, taking into account all possible paths through the states of the model.
- An alternative approach to estimating model parameters is to use a Viterbi training procedure, in which the initial guesses are used to find the most likely state-level

segmentation of the data and then the model parameters are re-estimated for this alignment of data frames to model states.

- Vector quantization is one method for reducing the set of possible feature vectors to a number for which robust training is possible. A parametric model, such as the normal (Gaussian) distribution or, more generally, a weighted sum (mixture) of Gaussian distributions, can also be used to describe the feature statistics. The re-estimation is then applied to the distribution parameters.
- HMMs can be extended to deal with word sequences, in which each state of the model represents one word, and the transition probabilities are determined by word sequence statistics of the language.
- One way of overcoming scaling problems because of very small numbers in the probability calculations is to represent all the numbers by their logarithms, and to use a special technique for finding the logarithm of the sum of two numbers.
- The dynamic programming recognition method described in Chapter 8 can be shown to be equivalent to using a very simplified form of HMM.
- The durational characteristic of an HMM state is determined only by the selfloop transition probability, and is such that the most likely duration is always only one frame and probabilities for longer durations decrease exponentially, so forming a geometric progression.

CHAPTER 9 EXERCISES

E9.1 What is the significance of the word ‘hidden’ in hidden Markov models?

E9.2 Why is it not necessary to explicitly consider all possible state sequences when calculating the probability of an HMM generating observed data?

E9.3 What is the essential difference between the Viterbi algorithm and the total likelihood method when calculating the probability of a word model generating observed data? What practical advantages can be gained by using the Viterbi algorithm for recognition?

E9.4 How can the form of an HMM be constrained by choice of initial parameters provided for re-estimation?

E9.5 What is the purpose of the ‘vector quantization’ sometimes used in HMMs?

E9.6 What are the benefits of using normal distributions to model feature statistics for HMMs? What are the limitations of simple normal distributions and how can these be overcome?

E9.7 How do the calculations required for Viterbi training differ from those for Baum-Welch re-estimation?

E9.8 What are the practical difficulties associated with implementing forward and backward probability calculations? What solutions are usually adopted?

E9.9 How can a simple HMM be interpreted as equivalent to a DTW recognizer?

E9.10 Why are the state durational characteristics of HMMs not very appropriate for modelling speech? What are the effects on duration characteristics if a single state is replaced by a sequence of several identical states?

CHAPTER 10

Introduction to Front-end Analysis for Automatic Speech Recognition

10.1 INTRODUCTION

The term "front-end analysis" refers to the first stage of ASR, whereby the input acoustic signal is converted to a sequence of acoustic **feature vectors**. As explained in Section 8.3, the short-term spectrum provides a convenient way of capturing the acoustic consequences of phonetic events. Ideally the method of front-end analysis should preserve all the perceptually important information for making phonetic distinctions, while not being sensitive to acoustic variations that are irrelevant phonetically. As a general policy for ASR, it seems desirable not to use features of the acoustic signal that are not used by human listeners, even if they are reliably present in human productions, because they may be distorted by the acoustic environment or electrical transmission path without causing the perceived speech quality to be impaired. Over the years many different front-ends have been tried, for use first with DTW recognizers and, more recently, with HMM systems. These front-ends vary in the extent to which they incorporate knowledge about human auditory perception, but currently the most successful analysis methods include at least some of the known properties of perception. These successful methods are, however, also characterized by a compatibility with the mathematical techniques that are generally used in HMM recognizers (as will be explained later). In this chapter we will introduce various aspects of front-end analysis for ASR.

10.2 PRE-EMPHASIS

The spectrum of voiced speech is characterized by a downward trend, whereby frequencies in the upper part of the spectrum are attenuated at about 6 dB/octave. This downward trend is due to a combination of the typical -12 dB/octave slope of the glottal source spectrum with the +6 dB/octave lift given by the radiation effect due to the lips (see Chapter 2). For the purpose of front-end analysis, it is common to compensate by applying a **pre-emphasis** of 6 dB/octave so that the analysed signal has a roughly flat spectral trend. This pre-emphasis is easily applied to the speech signal as the first processing stage. Although the above argument for pre-emphasis only applies to voiced regions, in practice it is usually applied throughout without causing any obvious problems for the analysis of voiceless regions.

10.3 FRAMES AND WINDOWING

Due to physical constraints, the vocal tract shape generally changes fairly slowly with time and tends to be fairly constant over short intervals (around 10–20 ms). A

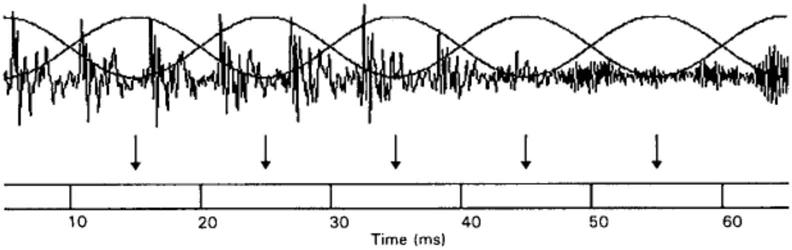


Figure 10.1 Analysis of a speech signal into a sequence of frames. This example shows a 20 ms Hanning window applied at 10 ms intervals to give a frame rate of 100 frames/s.

reasonable approximation is therefore to analyse the speech signal into a sequence of **frames,** where each frame is represented by a single feature vector describing the average spectrum for a short time interval.

Prior to any frequency analysis, each section of signal is multiplied by a tapered **window** (usually a **Hamming** or **Hanning** window). This type of windowing is necessary to reduce any discontinuities at the edges of the selected region, which would otherwise cause problems for the subsequent frequency analysis by introducing spurious high-frequency components into the spectrum. The length of each analysis window must be short enough to give the required time resolution, but on the other hand it cannot be too short if it is to provide adequate frequency resolution. In addition, because the analysis is normally performed at a fixed time interval, during voiced speech the window must be long enough so that it is not sensitive to exact position relative to the glottal cycle (i.e. there needs to always be at least one complete glottal cycle in the main part of the window). Long windows also have the advantage of smoothing out some of the random temporal variation that occurs in unvoiced sounds such as fricatives, but at the expense of blurring rapid events such as the releases of stop consonants. A common compromise is to use a 20–25 ms window applied at 10 ms intervals (giving a **frame rate** of 100 frames/s and an overlap between adjacent windows of about 50%), as shown in Figure 10.1.

10.4 FILTER BANKS, FOURIER ANALYSIS AND THE MEL SCALE

In Section 8.3 we introduced a speech signal representation using a filter bank with channels whose bandwidth and spacing increase with frequency (motivated by psychophysical studies of the frequency resolving power of the human ear). A convenient implementation of filter-bank analysis involves applying a Fourier transform. The output of the Fourier analysis will usually be at a finer frequency resolution than is required, especially at high frequencies. Thus the Fourier magnitudes are summed into a smaller number of channels, whose bandwidth and spacing conform to a perceptual scale such as the Bark or mel scale (see Section 3.5). Typically no more than 20 such channels are used for speech with a 4 kHz bandwidth, with a few additional channels being needed for higher-bandwidth signals. As already explained in Section 8.3, it is advantageous for the filter-bank output to represent power logarithmically, which reflects the phonetic

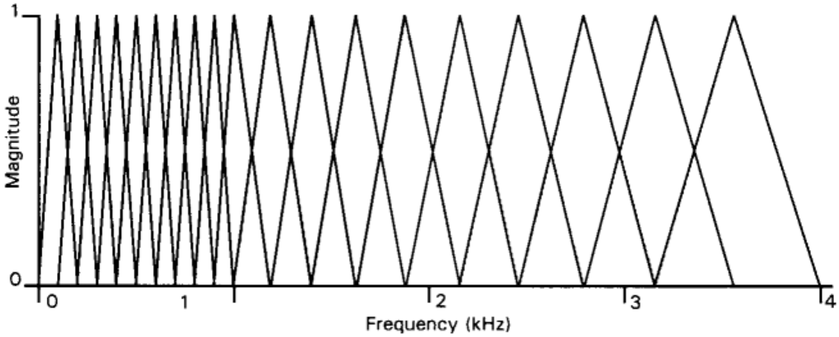


Figure 10.2 Triangular filters of the type suggested by Davis and Mermelstein (1980) for transforming the output of a Fourier transform onto a mel scale in both bandwidth and spacing.

significance of level variations and accords with evidence of a similar compressive non-linearity in auditory systems (see Chapter 3). A consequence of the logarithmic compression is that, when sampled from representative speech over a long period of time, the distribution of the energy in each of the channels tends to follow a Gaussian distribution, and is therefore compatible with any Gaussian assumptions that are made in the modelling.

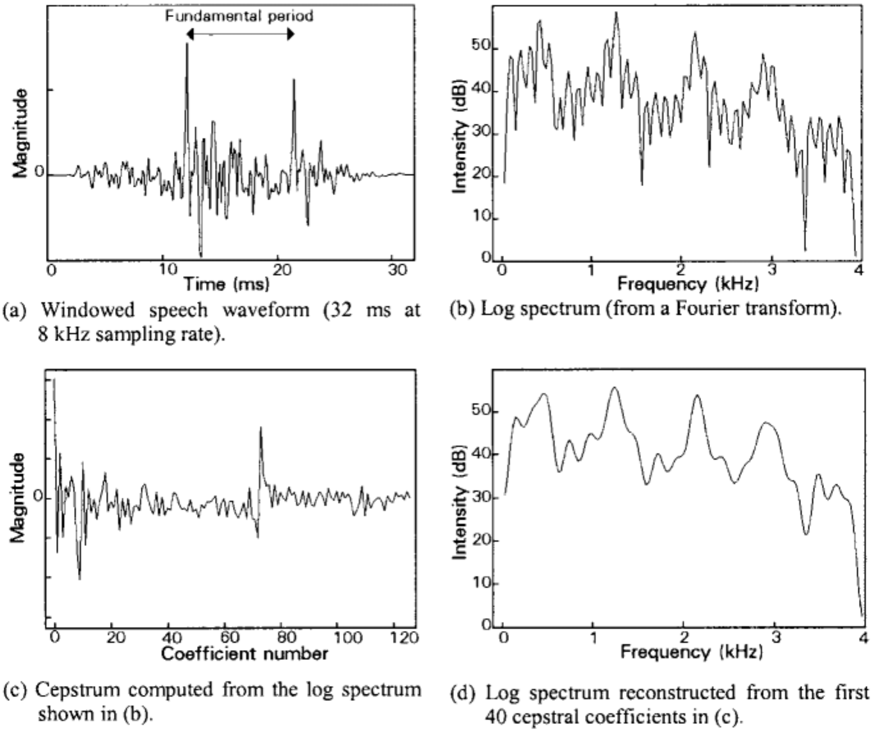
Figure 10.2 shows a set of triangular 'filters' that can be used to compute a weighted sum of Fourier spectral components, so that the output of the process approximates to a mel scale. Here the centre frequencies of the filters are spaced equally (at intervals of 100 Hz) on a linear scale from 100 Hz to 1 kHz, and equally on a logarithmic scale above 1 kHz. (Other slightly different spacings are also often used.) Each filter's magnitude frequency response is triangular in shape, and is equal to unity at the centre frequency and decreases linearly to zero at the centre frequencies of the two adjacent filters. This configuration of mel filters, which is now very widely used in ASR, was suggested by Davis and Mermelstein (1980).

One option is to use the output of a filter-bank analysis to provide the recognition features directly. However, although filter-bank energies were widely used and achieved a fair amount of success as acoustic features in early recognition systems, there are substantial advantages to be gained by applying further transformations and this approach is the more usual choice nowadays.

10.5 CEPSTRAL ANALYSIS

The frequency resolution that is given by Fourier analysis applied to a 20–25 ms window of speech is generally sufficient to resolve the individual harmonics of the voiced excitation source, as well as showing the spectral shaping that is due to the vocal tract. Because the filtering operation of the vocal tract is the most influential factor in determining phonetic properties of speech sounds, it is desirable to separate out the excitation component from the filter component. The vocoders described in Chapter 4 are based on this principle. **Cepstral analysis** is another technique for estimating a separation of the source and filter components. Here the starting point is the observation that passing an excitation signal through a vocaltract filter to generate a speech signal can be represented as a process of **convolution** in the time domain, which is equivalent to multiplying the spectral magnitudes of the source and filter components. When the spectrum is represented logarithmically, these components are *additive,* because the logarithm of a product is equal to the sum of the logarithms ($\log(A\times B)=\log(A)+\log(B)$). Once the two components are additive, it is relatively straightforward to separate them using filtering techniques.

A typical logarithmic spectrum cross-section shows the rapidly oscillating component due to the excitation superimposed on a more gradual trend representing the influence of the vocal tract resonances (see Figure 10.3(b)). If we now imagine that this combined shape represents a time-domain signal, the rapid oscillations would correspond to high-frequency components, while the more gradual changes would be due to low-frequency components. If a Fourier transform were applied, the two components would therefore appear at opposite ends of the resulting spectrum. Thus by starting with the log magnitude spectrum and computing a Fourier transform, to obtain the so-called **cepstrum** (an anagram of "spectrum"), the excitation is effectively separated from the vocal-tract filtering, as shown in Figure 10.3(c). In fact, because the log magnitude spectrum is a symmetric function, the Fourier transform can be conveniently simplified to a



(a) Windowed speech waveform (32 ms at 8 kHz sampling rate).

(b) Log spectrum (from a Fourier transform).

(c) Cepstrum computed from the log spectrum shown in (b).

(d) Log spectrum reconstructed from the first 40 cepstral coefficients in (c).

Figure 10.3 Analysing a section of speech waveform to obtain the cepstrum and then to reconstruct a cepstrally smoothed spectrum.

discrete cosine transform (**DCT**). For a spectral representation comprising N channels with log magnitudes A_1 to A_N, the DCT can be computed as follows:

$$c_j = \sqrt{\frac{2}{N}} \sum_{i=1}^{N} A_i \cos\left(\frac{\pi\, j(i-0.5)}{N}\right) \quad \text{for } 0 \leq j < N, \tag{10.1}$$

where c_j is the j^{th} **cepstral coefficient**. When j=0, Equation (10.1) simplifies so that c_0 is proportional to the mean of the individual log channel signals A_1 to A_N. The c_1 term reflects the balance between energy at low frequencies and energy at high frequencies. As j increases, c_j captures increasingly fine spectral detail: first overall spectrum shape, then general formant structure, then more detailed spectral structure between the formants and, at high values of j, the excitation structure. There is no simple relationship between the c_j terms and the formants. However, for periodic speech the effect of the excitation source tends to be seen as a clear 'spike' at the pitch period duration (see Figure 10.3(c)). Cepstral analysis is therefore one method that can be used to estimate fundamental frequency. For example, in Figure 10.3(c) the spike occurs at c_{73} which, for the sampling frequency of 8 kHz (i.e. a sample duration of 0.125 ms), corresponds to a fundamental period of 73×0.125=9.125 ms. This value can be seen to be roughly equal to the interval between the pitch pulses in Figure 10.3(a).

Although the cosine transformation given by Equation (10.1) ensures that the Euclidean distance in transformed space is exactly equal to the distance between the sets of untransformed channel signals, the information that is of phonetic significance becomes concentrated in the lower-order terms. Filtering the cepstrum (a process usually referred to as **liftering**) can be applied to remove certain components or alter the relative influence of the different components. A simple lifter is one which simply truncates the cepstral sequence, by giving a weight of one to the low coefficients (up to some specified index) and a weight of zero to all the higher coefficients. By setting the cut-off point to just below the coefficient corresponding to the pitch period, most of the influence of the fundamental period is effectively removed from the spectrum. This process is shown in Figure 10.3(d), in which the spectrum has been re-constructed (by a Fourier transform) from just the low-order cepstral coefficients. The resulting spectrum can be seen to be much smoother than the original and show the formant peaks more clearly. The lower the cut-off point is set, the more detail will be removed and the smoother the spectrum will be. The process of smoothing the spectrum by truncating the sequence of cepstral coefficients is often referred to as **cepstral smoothing**.

The effectiveness of cepstral analysis for separating out the fundamental-frequency component of a speech signal depends on the frequency of the fundamental relative to the frequencies of the formants. The method generally works best for adult male speech (as shown in the example in Figure 10.3). For typical female and children's speech both the pitch and formant frequencies are higher, but the pitch increases more relative to the formant frequencies and so the cepstrum gives a less clear separation of the excitation component. It is therefore more difficult to set a cut-off point for cepstral smoothing that removes the pitch influence without also removing useful information about the formant structure.

In addition to the beneficial effect of concentrating on the information that is of greatest phonetic significance, discarding the high-order cepstral coefficients reduces the number of features, so less computation is needed for the pattern-matching process.

The cepstrum has another desirable property for use in speech recognition. For typical speech signals it is found that, in contrast with the original channel signals, the variation of the separate coefficients tends to be uncorrelated. As a consequence, when using HMMs with continuous-density probability distributions, full covariance matrices can be replaced by much simpler diagonal covariance matrices (see Section 9.8.1) without any great loss in performance. Using diagonal covariance matrices substantially reduces both the computational requirements and the number of parameters needed to represent each distribution.

Cepstral coefficients have the property that (ignoring coefficients that are associated with pitch) both the variance and average numerical values decrease as the coefficient index increases (see Figure 10.3(c)). A consequence for a DTW recognizer using a simple Euclidean distance metric is that the distance calculation is affected most by the lowest-order coefficients and the coefficients that are more related to formant structure tend to be given insufficient weight. A solution that has often been adopted is to apply a lifter with a weighting for each coefficient that acts to roughly equalize the variances for the different coefficients. The problem does not arise when using probability distributions in HMM systems, because the variance is accommodated in the probability calculations. The liftering is often still applied, however, because the effect of making the variances of all the features cover a similar range makes it easier to study model parameters and to place restrictions on variances as part of re-estimation (see Section 11.4.1).

As explained above, the c_0 coefficient is proportional to the mean of the log channel signals and therefore provides an indication of overall level for the speech frame. Sometimes c_0 is included in the feature set, but often it is discarded and replaced by a different energy measure that is derived from the true signal energy. The energy in each frame will depend on overall speaking level, but for identifying sounds the most relevant factor is the *relative* level for different frames in an utterance. Therefore, for those applications for which the whole utterance becomes available before recognition needs to start, the measured energy is often normalized with respect to the maximum energy found over all frames in the utterance. (See Section 8.3.2 for further discussion about measures of speech level.)

In order to retain the advantages of a perceptually motivated filter-bank analysis, for ASR the cosine transform is usually applied to the output of non-linearly spaced filter-bank channels (see Section 10.4 above). A popular choice is to use **mel-frequency cepstral coefficients (MFCCs),** which are obtained by applying a DCT to the output of mel filters such as the ones shown in Figure 10.2. An acoustic representation using MFCCs is often simply referred to as a **mel cepstrum**. As explained above, it is generally advantageous to discard the higher-order coefficients. For example, with 8 kHz bandwidth speech, there might be 24 mel channels but only the first 12 MFCCs are generally used in the final feature set. Although the use of the non-linear filter-bank means that the cosine transform no longer gives a simple separation of the excitation from the vocal-tract filtering (and much of the excitation effect will usually have already been smoothed out by the mel averaging), the truncation of the cepstral sequence has a general spectral smoothing effect that is normally desirable because it tends to remove phonetically irrelevant detail.

10.6 ANALYSIS BASED ON LINEAR PREDICTION

An alternative to filter-bank methods for representing the short-term spectrum is to derive linear prediction (LP) coefficients (usually called LPC analysis because of its origin in linear predictive coding, see Chapter 4). In the past, mainly during the 1970s, many recognizers were built using LPC-derived features and these systems generally gave performance comparable with that obtained from recognizers using filter-bank methods. During the 1980s it became more popular to use LPC-derived cepstral coefficients rather than the LP coefficients themselves because, as in the case of the filter-bank representation, the addition of the cepstral transformation was found to improve recognition performance. A convenient method exists for computing cepstral coefficients directly from the LP coefficients. LP analysis has the advantage that it produces an estimate of the smoothed spectrum, with much of the influence of the excitation removed. However, there is less freedom to apply non-linear processing to combat noise than there is with a filter-bank front-end. In addition, LPC inherently gives uniform weighting to low- and high-frequency regions of the spectrum. A non-linear frequency scale can be incorporated, but complicates the analysis to a greater extent than when using filter-bank methods.

Perceptual linear prediction (PLP) (Hermansky, 1990) is one LP-based analysis method that successfully incorporates a non-linear frequency scale and other known properties from the psychophysics of hearing. In PLP analysis, a Fourier transform is first applied to compute the short-term power spectrum, and the perceptual properties are applied while the signal is represented in this filter-bank form. The spectrum is transformed to a Bark scale, and this spectrum is pre-emphasized by a function that approximates the sensitivity of human hearing at different frequencies (see Figure 3.5). The output is compressed to approximate the non-linear relationship between the intensity of a sound and its perceived loudness. The all-pole model of LPC is then used to give a smooth, compact approximation to the simulated auditory spectrum, and finally the LP parameters are usually transformed to cepstral coefficients for use as recognition features. Apart from the use of LPC to achieve spectral smoothing, PLP analysis is very similar to MFCC analysis, but with perceptual properties incorporated in a way that is more directly related to psychophysical results (see Table 10.1 for a comparison of the two methods). In recent years a number of recognition systems have used PLP-based cepstral coefficients as acoustic features, and experimental evidence suggests that overall they give performance that is comparable with that obtained using MFCCs.

Table 10.1 Comparison between the properties of PLP cepstral coefficients and typical MFCCs.

MFCCs	PLP cepstral coefficients
Cepstrum-based spectral smoothing	LPC-based spectral smoothing
6 dB/octave pre-emphasis applied to speech waveform	equal-loudness pre-emphasis applied to spectrum
triangular mel filters	critical-band filters
logarithmic amplitude compression	cube root amplitude compression

10.7 DYNAMIC FEATURES

In the HMM probability calculations (see Section 9.3), the probability of a given acoustic vector corresponding to a given state depends only on the current vector and the current state, and is otherwise independent of the sequence of acoustic vectors preceding and following the current vector and state. It is thus assumed that there is no dependency between the observations, other than through the underlying state sequence. In reality, however, an acoustic feature vector representing part of a speech signal is highly correlated with its neighbours. In fact, it is often the dynamic characteristics of the features that provide most information about phonetic properties of speech sounds (related to, for example, formant transitions or the closures and releases of stop consonants). These correlations can be captured to some extent by augmenting the original set of ('static') acoustic features (such as MFCCs) with dynamic features that are a measure of the change in the static features. These dynamic features are often referred to as **time derivatives** or **deltas**. One way of computing the delta features is by simple differencing between the feature values for two frames either side of the current frame:

$$\Delta y_t = y_{t+D} - y_{t-D}, \qquad (10.2)$$

where *D* represents the number of frames to offset either side of the current frame and thus controls the width of the window over which the differencing operation is carried out. Typically *D* is set to a value of 1 or 2.

Although time-difference features have been used successfully in many systems, they are sensitive to random fluctuations in the original static features and therefore tend to be 'noisy'. A more robust measure of local change is obtained by applying linear regression over a sequence of frames:

$$\Delta y_t = \frac{\sum_{\tau=1}^{D} \tau \left(y_{t+\tau} - y_{t-\tau} \right)}{2\sum_{\tau=1}^{D} \tau^2} \qquad (10.3)$$

With linear regression, a value of *D*=2 is the usual choice for an analysis frame rate of 100 frames/s. This regression window of five frames (50 ms) is long enough to smooth out random fluctuations, yet short enough to capture local dynamics.

The delta features described above are first-order time derivatives, which can in turn be used to calculate second-order time derivatives (sometimes referred to as **delta-deltas**). Including first-order time derivative features usually gives a large gain in recognition performance, and adding second-order derivatives (which capture changes in the first-order dynamics) tends to give an additional but smaller improvement. The majority of current HMM systems incorporate first-order derivative features, most often applied to a basic feature set of MFCCs and an energy feature, and many also include second-order derivatives. Most of the benefit from derivative features is due to their ability to capture dynamic information. However, these features also have the useful property that they are not affected by any constant or slowly changing disturbances to the signal (such as linear filtering in microphone pre-amplifiers and on telephone channels, for example), provided that these distortions are additive in the feature domain (see Section 11.2).

10.8 CAPTURING THE PERCEPTUALLY RELEVANT INFORMATION

Both in Chapter 3 and at the beginning of the current chapter we explained the desirability of capturing properties of human phonetic perception in the front-end analysis for ASR. Analysis methods such as PLP take into account several known facts about the lower levels of human auditory processing. However, there is no attempt to model higher-level auditory processing or more specific properties of speech perception in any of the analysis methods that have been described above.

It is now well established that the frequencies of the speech formants, particularly the first and second, are vitally important phonetically. Relative formant amplitudes are much less important, and the detailed structure of the lower-level spectral regions between formants is of almost no consequence. There would therefore seem to be potential for better performance in ASR if these factors could be taken into account when designing acoustic analysis methods and distance metrics. Although auditory models have shown considerable promise for incorporating into systems for ASR (see Section 3.7), these types of features have not yet replaced more general spectral features such as MFCCs or PLP-cepstra as the preferred choice in HMM-based systems. It is possible that substantial changes in the design of the recognizers themselves will be required before it will be possible to gain the full benefit from incorporating auditory models. We will return to this issue in Chapter 16, when we will also discuss the prospects and issues for extracting and using formant information more explicitly in ASR.

10.9 GENERAL FEATURE TRANSFORMATIONS

The DCT is one orthogonal transformation that reduces the dimensionality of a filter-bank output by concentrating the most useful information into a small number of features. Other orthogonal transformations for data reduction include **principal components analysis** (PCA) and **linear discriminant analysis (LDA)**. PCA performs a linear transformation on an input feature set, to produce a different feature set of lower dimensionality in a way that maximizes the proportion of the total variance that is accounted for. LDA also applies a linear transformation on the input feature set, but here the transformation is chosen to maximize a measure of class separability, and hence to improve discrimination. In order to determine the transformation, this procedure requires each input feature vector to have first been associated with a single class. PCA and LDA are both general data-reduction techniques that can usefully be applied to reduce the dimensionality of any diverse feature set, including for example static spectral or cepstral features with first- and second-order time derivatives, or even the output of auditory models. Both PCA and LDA generate new feature sets that are uncorrelated, thus allowing diagonal covariance matrices to be used for HMM state emission p.d.f.s.

10.10 VARIABLE-FRAME-RATE ANALYSIS

It has been assumed so far that all the frames in an utterance are of equal importance when making a comparison with stored templates or models. However, a slight difference of vowel quality, for example, may not affect the identity of a word, whereas formant transitions at vowel-consonant boundaries may be crucial in identifying the consonant. Because, for many consonants, such transitions are very rapid, they do not occupy many frames. Although the addition of time-derivative features increases the importance of matching the transition characteristics, still rapid transitions may make only a small contribution to the cumulative distance or probability, even when they are matched very badly. The vowels and steady-state parts of long consonants can, in contrast, make a large contribution overall even when they match fairly well on each frame.

To overcome this difficulty it is necessary to give more weight to parts of the signal that are changing rapidly, and less weight to long steady regions. One way that is sometimes used to achieve this effect is to perform the original acoustic analysis at a fairly high frame rate (e.g. 100–200 frames/s), but then to discard a variable proportion of the frames depending on the distance between consecutive pairs of frames. Thus all frames are retained in rapid transitions, but perhaps only one in five is kept in very steady long vowels. This **variable-frame-rate** analysis method is similar to the scheme described in Section 4.3.5 for efficient speech coding. In the case of speech analysis for ASR, not only is there a computational saving, but also the overall match of an input utterance to stored templates or models shows much greater relative sensitivity to mismatch in transition regions.

CHAPTER 10 SUMMARY

- When deriving features for speech recognition, input speech is often first preemphasized by 6 dB/octave, so that the signal for subsequent analysis has a roughly flat spectral trend. Speech is analysed into a sequence of frames: most usually a 20–25 ms tapered window is applied at 10 ms intervals.
- One popular method of representing the speech spectrum is to use a filter bank with triangular filters whose width and spacing follow a mel scale. To obtain features for ASR, the output of such a filter bank is often subjected to a cosine transform (so deriving mel-frequency cepstral coefficients: MFCCs). An alternative is to derive cepstral coefficients from perceptual linear prediction.
- The cosine transform causes the features to become largely decorrelated so that diagonal covariance matrices can be used in the HMMs. In addition, information of phonetic significance is concentrated in the lower-order terms, so a more efficient representation can be obtained with fewer features.
- ASR performance is often greatly improved by adding ‘delta’ (first-order time-derivative) features, which are usually computed for each frame by applying linear regression over a window of five frames centred on the current frame.

CHAPTER 10 EXERCISES

E10.1 Why is cepstral analysis a useful tool in speech processing?

E10.2 Explain the stages that are typically used to analyse a speech signal into MFCCs and their first- and second-order time derivatives.

E10.3 How are properties of auditory perception simulated in front-ends for ASR?

CHAPTER 11

Practical Techniques for Improving Speech Recognition Performance

11.1 INTRODUCTION

Almost all of the successful current speech recognition systems are based on the HMM theory that was introduced in Chapter 9. It is now over 20 years since these methods were first applied to speech recognition. During this time there has been a progressive increase in the difficulty of the tasks that can be attempted with acceptable recognition performance. However, although there has been an increase in system complexity, the underlying theory and the HMM structure are generally unchanged from those used in the early systems. The main advances have been in tuning the application of these methods to the task of recognizing spoken language, including coping with all the variability that exists due to differences between speakers, environments and many other factors. In this chapter and in the following chapter we will introduce some of the most influential techniques that have been used to improve recognition performance. Some of the developments are closely tied to the demands of recognizing *large* vocabularies, and these aspects will be considered in Chapter 12. The current chapter concentrates on methods that are of general relevance to recognition systems irrespective of vocabulary size, and also discusses some techniques used when recognizing limited vocabularies.

11.2 ROBUSTNESS TO ENVIRONMENT AND CHANNEL EFFECTS

When a person is in a quiet environment speaking into a good-quality close-talking microphone connected directly into an ASR system, the signal that is input to the recognizer should be very close to the one that is output from the person's mouth. In many real ASR applications, however, this ideal situation does not exist and the speech that is input to the recognizer will have been corrupted in some way:

1. There may be external noise present in the signal. Sources of environmental noise include computers and other office equipment, machinery, car engines, music, and even other people speaking in the background. A speech signal captured in these conditions will contain **additive noise**. Noise can also be added by a poor-quality microphone or a noisy transmission channel.
2. The uttered speech signal itself may undergo some spectral distortion during its transmission from the talker's mouth to the speech recognizer. Sources of such distortion include room reverberation, the microphone transducer and the transmission channel. The characteristics of the speech will also be affected by any application of waveform-coding techniques such as CELP. There are many causes for the degradation that is typical of telephone speech, but a particular problem is the

> *variability* of this degradation: different telephone handsets and channels may have different characteristics and therefore cause different disturbances to the speech spectrum; cellular (mobile) telephones are especially problematic. Of the range of possible disturbances, many, such as channel bandwidth limitation and the spectral shaping introduced by microphones, have a linear-filtering effect on the speech signal. The effect of these disturbances can thus be viewed as one of convolving the original speech signal with a filter representing the characteristics of the disturbance, and hence this type of corruption to speech signals is often referred to as **convolutional noise**.

The difficulties of dealing with noise and other imposed signal disturbances are exacerbated by the tendency for talkers to modify the way they speak, and in particular to increase their vocal effort, when the acoustic environment worsens. This phenomenon is known as the **Lombard effect,** named after Etienne Lombard who first described it (Lombard, 1911). As environmental noise level increases, people's natural response is to talk more loudly and often with a more exaggerated style of articulation. The consequence for the acoustic signal is an increase in overall level, but also, perhaps more significantly, changes in spectrum shape. One effect that has been observed is a change in spectral tilt. This change can be regarded as being due to another type of convolutional disturbance, the characteristics of which are dependent upon other (external) causes of corruption. There may also be changes in formant frequencies and in the durations of many of the speech sounds, due to factors such as more precise articulation, increased muscular tension and reduced speaking rate.

Speech recognizers generally perform better in quiet, 'clean' conditions than when the speech signal is noisy or distorted. However, the greatest problem arises when there is a **mismatch** between the conditions in which the recognizer is used and those under which it was trained. It is always best to train an ASR system using speech material that is recorded under conditions that are as close as possible to the predicted operational conditions. It is, however, not always possible to predict these conditions in advance and anyway the conditions may change over time: the noise in an office or car is continually changing for example, and transmission channels may introduce variable distortion. Techniques are therefore needed to make the best use of the most reliable information in the signal whatever corruption may be present, even when the nature of the corruption may be unknown and variable.

Additive noise is easiest to deal with in the linear spectral domain, where spectral components due to the noise can be seen as being added to the components representing the speech. Convolutional noise on the other hand is easier to cope with in the log spectral or cepstral domain. As explained in Section 10.5, the effect of convolution in the time domain becomes one of addition in the log spectral (or cepstral) domain, so making it easier to separate out the different convolved components. Techniques for handling either type of 'noise' can be roughly divided into two categories. The first category encompasses methods that are applied purely at the level of the features (with no reference to the models), and includes both the selection of features that are inherently robust to corruption and the application of speech enhancement techniques to existing feature sets. The second category covers model-based methods whereby some way of dealing with the effects of noise or distortion is incorporated into the recognition process itself.

11.2.1 Feature-based techniques

A simple way of compensating for additive noise is to estimate the spectrum of the noise and then to subtract this spectrum from the spectrum of the signal, with the **spectral subtraction** being performed in the *linear* spectral domain. This method requires an estimate of the noise spectrum, which can be obtained by averaging over a length of signal that contains only noise. In some cases a second microphone may be available for assisting with this process. If there is only a single channel available, non-speech regions of signal can be detected to a reasonable degree of reliability by automatic methods, which are typically based on some heuristic measure of spectral characteristics. The noise estimate will represent an average spectrum, so subtraction of this estimate from the spectrum for any one frame may give negative values for some channels. Any negative values can simply be set to zero, or alternatively it may be better to set a small positive threshold for all the subtracted channel levels, to allow for the fact that low-level channels will be affected more than high-level channels by any errors in the estimated noise spectrum.

Because convolution in the time domain is equivalent to addition in the logarithmic spectral domain, the principle of subtracting noise from the spectrum can be applied to log spectral or cepstral features to remove convolutional noise. In this case, any estimate of the spectral characteristics of the 'noise' needs to be made in speech regions as speech must be present for the effects of the convolutional distortion to be evident. Such distortion tends to be fairly constant over the duration of an utterance, while the speech information is captured in the changing *short-term* spectral characteristics. Thus any convolutional distortion can be removed, without removing useful speech information, by simply subtracting the mean feature vector computed over the duration of a reasonably long utterance. This technique is widely applied in speech recognition systems using cepstral features, when the method is known as **cepstral mean subtraction (CMS)** or **cepstral mean normalization (CMN)**.

CMS can be viewed as performing a type of high-pass filtering of the temporal characteristics of the signal spectrum, to remove just the constant component. This concept of filtering temporal characteristics can be extended to also remove components that are changing only slowly or very rapidly, both of which are unlikely to be related to phonetic properties of the speech. By designing an appropriate band-pass filter for the temporal characteristics of speech, it is possible to maximize sensitivity to characteristics that are changing at a rate that is most likely to be related to phonetic properties. The technique is usually referred to as **relative spectral,** or **RASTA** (RelAtive SpecTrAl), processing.

Time derivatives of log spectra or cepstra are inherently robust to any constant or slowly changing convolutional distortion, because by definition these features are only affected by local changes in the feature values.

11.2.2 Model-based techniques

When speech is spoken in noisy conditions, the low-energy parts of the speech spectrum may be completely corrupted by the noise. However, the higher-level parts of the speech spectrum, in the vicinity of intense formants, will generally be above the noise level and so still provide useful information for speech recognition. It should therefore be advantageous to use a probability calculation that gives highest weight to those feature differences which are most likely to be reliable, while ignoring differences where noise corruption is such that it is not possible to know whether there is really a difference in the underlying speech. Provided that the acoustic analysis does not mix noise-corrupted and reliable parts of the signal into the same features (e.g. filter-bank analysis is suitable), the recognition calculations can be easily modified to have appropriate properties. As with spectral subtraction methods, it is necessary to have an estimate of the current noise level in each spectral channel. For each comparison between an observed speech frame and a reference model state, and for each spectral channel, the probability calculation can take into account the observed and model values in relation to the noise level.

A simple strategy involves just replacing any measured channel levels that are below the noise level by the noise estimate. This procedure is applied both to observed signals and to the stored model. Provided that the values for both the observation and model are above the noise level, the comparison between them is not affected. Conversely, if they are both below the noise level, then the difference will become zero (in this instance it is not possible to know whether there are really any differences in the underlying speech). If only one out of the observation and the model is above the noise level, this high-level signal will be compared with the level of the noise. There are several variants of the above technique, which is often referred to as **noise masking** because a noise estimate is applied as a ‘mask’ both to input speech and to the models (or templates in the case of a DTW recognizer).

Another possibility is to model the fact that the true values for noise-corrupted spectral regions of the speech are not known, by modifying the recognition calculations to allow for all possible values for these ‘missing’ components of the spectrum.

A natural extension of noise masking is to model variation in the noise. If an HMM is used to represent the characteristics of the noise, both spectral and temporal variability can be accommodated. A single state should be sufficient if the noise characteristics are fairly constant over time **(stationary noise),** while multiple states can be used to handle noise with changing spectral characteristics **(non-stationary noise)**. By using parallel sets of HMMs, one set for the speech and one set for the noise, the modelling process can be viewed as performing **decomposition** of a noisy speech signal into its constituent parts. If the noise HMM has more than one state, the search needs to be extended to deal with all possible pairings of speech and noise model states. For each of the possible state pairings, the probability of the observed noisy speech feature vector needs to be computed. This probability calculation is made tractable by treating each filter-bank channel separately and making the assumption that the energy in a channel is dominated either by the speech or by the noise. An approximation to the probability of a noisy speech observation is then computed as a weighted sum of the probability of the observation being generated by the speech model and the probability of it being generated by the noise model.

Both decomposition and noise masking schemes have been shown to improve HMM recognition performance in noisy conditions when using filter-bank features. However, because these schemes deal with each filter-bank channel separately and

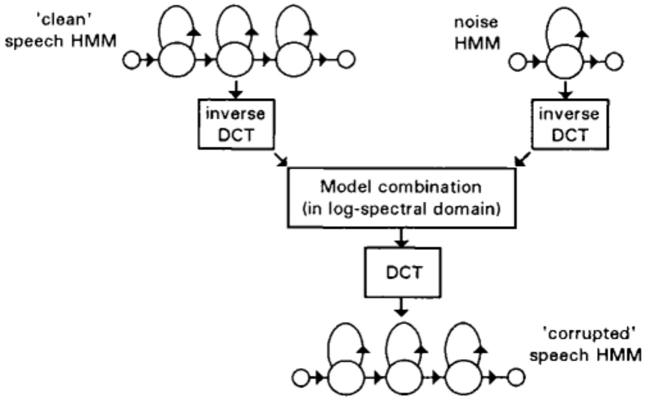


Figure 11.1 Using parallel model combination to generate a model for noise-corrupted speech by combining a speech model with a noise model (adapted from Gales and Young (1995)).

independently, they cannot represent the correlation that exists between the filter-bank channels. Furthermore, the methods operate in the log-spectral domain and cannot be applied directly to features that have been obtained after a transformation such as a DCT. Rather than attempting to separate out the speech and noise components of a noisy speech signal, an alternative is to combine speech models with noise models in order to estimate models for the corrupted speech. Having applied the noise compensation to the models 'off-line', the new models are then used for recognition in a standard HMM system. The technique for combining the models is known as **parallel model combination (PMC)**.

An important advantage of PMC is that it can be applied to models that use cepstral features, by adopting the approach shown in Figure 11.1. An inverse DCT is first applied to convert both sets of model parameters back into the log-spectral domain before combining them to derive a single set of models representing the corrupted speech. This new set of model parameters can then be transformed back to the cepstral domain ready for use in a standard recognition system. The recognizer itself is not altered but, if the noise model has more than one state, the corrupted speech models will have more states than the original speech models because it will be necessary to have a model structure that represents all possible pairings of speech and noise model states. The PMC process involves some approximations in order to estimate the parameters of the new model for the corrupted speech, and it is assumed that the original models for both the speech and the noise provide accurate representations of their true distributions.

11.2.3 Dealing with unknown or unpredictable noise corruption

The model-based noise compensation schemes that have been described in the previous section can be classed as **predictive,** because they start with a model of speech in a quiet environment and some model of the noisy environment and attempt to predict what will happen to the speech in this noisy environment. Such predictive schemes are useful because they can make some reasonable approximation to the effects of a change in the environment without needing any speech data from the new environment. However, the success of the technique is dependent upon the accuracy of the predictive model that is used.

Predictive noise-compensation schemes such as PMC rely on some estimate of the characteristics of the noise. For noise that is additive in the spectral domain, the missing-data method mentioned above can be used even when the noise characteristics are not known, as this approach is dependent only on identifying which spectral regions have been corrupted. This process might involve using noise estimates, but could alternatively be achieved by using cues (such as harmonicity) that are related to speech properties.

Even if the noise characteristics are known, to be able to apply the predictive noise-compensation schemes described in Section 11.2.2 it is necessary to assume that the speech and the noise are independent from each other. The reality is that talkers' speech usually changes in the presence of noise (the Lombard effect mentioned earlier), but it is difficult to predict the exact nature of the interactions between the speech and the noise. If speech data are available for the new environment, this difficulty can be avoided by using **adaptive** schemes whereby the new data are used to adjust the parameters of the models to be more representative of those data. Adaptation methods will be described in Section 11.4. Some form of predictive compensation for environmental or channel characteristics can provide a good starting point for any subsequent adaptation.

11.3 SPEAKER-INDEPENDENT RECOGNITION

Many ASR systems are required to work with speech from a wide variety of individuals without re-training, and such systems are generally referred to as **speaker independent**. The acoustic realization of a word may show a wide degree of variation across different talkers due to many factors, including physical differences between peoples' vocal tracts, differences in accent or dialect and variation in speaking style. Often a single set of HMMs is used to represent speech from all the different talkers, with multiple-component Gaussian mixture distributions being used to describe the emission p.d.f.s. Although Gaussian mixtures allow complicated distributions to be described very accurately, the variation across individuals will cause the distributions to be broader than corresponding distributions for speech from a single individual. The result is a greater degree of overlap between the distributions representing different speech sounds and hence the discrimination power for any one individual is reduced.

If a recognizer uses a single set of models to represent speech from a variety of different talkers, the system could work just as well if each successive word were spoken by a different person. This facility is never normally required for any real task, so better performance should be obtainable by somehow taking account of consistent speaker-dependent factors that affect the spectral characteristics of the speech. Provided that sufficient training data are available, one option is to use different sets of models for different clearly identifiable categories of talkers. A simple separation that has been found to be beneficial involves using one set of models for male talkers and another set for female talkers. Whenever multiple model sets are used, either the speech must be classified into one of the different categories prior to recognition, which will inevitably not be possible with perfect accuracy, or alternatively different recognizers must operate in parallel (one recognizer for each model set), which increases the computational load.

ASR systems are often designed with the aim of minimizing the influence of acoustic differences between speech from different individuals. Standard front-end processing is influenced to some extent by a desire not to be too sensitive to talker differences by, for example, removing fundamental-frequency effects. Techniques such as computing time derivatives, CMS and RASTA also help reduce sensitivity to differences between talkers by removing long-term bias in feature values.

Performance of speaker-independent recognizers can be greatly enhanced by adapting the model parameters in order to improve the match between the models and speech from a particular individual. We will consider adaptation methods in general in Section 11.4. Other methods use feature transformations that are aimed specifically at normalizing for differences between individuals, as explained below.

11.3.1 Speaker normalization

Individuals differ in the physical dimensions of their vocal tracts, and the frequencies of the formants are related to vocal-tract length (see Section 2.3). Thus vocal-tract length differences are manifested in speech as fairly consistent differences in the positions of the spectral peaks corresponding to the formant frequencies. The aim of **speaker normalization** is to compensate for these differences by applying some appropriate, speaker-specific, warping of the frequency scale of a filter bank. Any transformation (such as a DCT) is then computed after applying the warping.

One way of estimating the required frequency warping is to calculate average formant frequencies, but this approach requires automatic formant analysis which is difficult to achieve reliably. An alternative method involves finding an 'optimal' warping, which is defined to be the warping factor that maximizes the probability of the feature vectors given the models. It is usual to select a small set of possible warpings, and then use a simple search procedure to find the one that gives the highest probability. The procedure is applied at both the training and testing phases. For training, a set of models is first trained as normal, then an iterative procedure is used whereby the warpings are computed, then the models retrained, and so on. The situation for the testing stage is more difficult because the text of the utterance, and therefore the identity of the model, is not known. A solution is to begin by performing recognition without applying any warping of the test utterance, and to use the recognized model as the basis for computing the optimal frequency warping of the utterance. A second recognition stage is then applied to the warped version of the utterance in order to decide on the word identities. Procedures of this type have been found to consistently improve speaker-independent recognition performance, although at the expense of a considerable increase in computation.

Techniques of the type described above are often referred to as performing **vocal tract length normalization (VTLN)**. While methods based on formant frequencies are likely to be performing compensation that is quite closely related to vocal tract length, search-based methods of the type described here are really compensating more generally for spectral differences. These differences could be due to many factors including, but not necessarily confined to, vocal tract length.

11.4 MODEL ADAPTATION

As discussed in the previous two sections, a major issue for ASR is coping with acoustic variability due to differences between different environments, different transmission channels, different talkers and even the same talker on different occasions. The techniques described in Sections 11.2 and 11.3 can all help to make machines more robust to such variation. However, given some speech data that are representative of any particular recognition task, further performance gains are possible by *adapting* the parameters of the models in the recognizer to provide a better match to those data. If the adaptation process works well, any changes in the characteristics of the acoustic signal should be compensated for, whatever the cause of the mismatch between the current acoustics and the original set of models.

For any adaptation to be accurate, ideally it should only take place when words have been recognized correctly and thus the text of the adaptation data is 'known'. This scenario is referred to as **supervised adaptation** and is possible, for example, when it is practical to obtain feedback from the user to confirm the recognition results. An alternative that may also be an option for some applications, especially those that are used interactively and for a reasonable length of time by any one person, is to require a new user to start by speaking some known text.

Supervised adaptation will not generally be possible in situations where a recognizer is used to transcribe speech without interactive involvement of the talker, especially if either the person or the environment are changing frequently. When the text is not known **unsupervised adaptation** is required, which is more difficult because recognition must proceed concurrently with adaptation. A typical unsupervised adaptation procedure involves an iterative process whereby recognition is first performed using the current set of models, and the recognized transcription is then used as the basis for adapting the models before performing recognition again. This procedure relies on obtaining sufficiently good recognition accuracy with the first set of models to be useful for the subsequent adaptation.

If sufficient data were available for the new conditions, then 'adaptation' could be performed by simply retraining the models on the new data. However, many recognizers (especially those designed for large vocabularies) are originally trained on huge quantities of training data, which is completely impractical for adaptation. In practice it is often desirable to adapt models using only a very small quantity of data, which may not even include any examples for some of the model units. Thus the adaptation process needs somehow to combine the information provided by the new data with the information provided by the original set of models. Two types of method for achieving this aim are described below.

11.4.1 Bayesian methods for training and adaptation of HMMs

In Section 9.5 we described the widely used HMM training method, whereby the model parameters are trained with the aim of obtaining the closest possible fit to a given set of training data. Thus, given some training data ***Y***, values are found for some model parameters ? in order to maximize P(***Y***|?), the probability of the data being generated by the models (i.e. the **likelihood** of the data). This **maximum likelihood (ML)** approach to model training can work well provided that there are sufficient examples of each model unit to give reasonable estimates of the true distributions of the features. However, ML estimates tend to be unreliable when the data are sparse. As an extreme example, consider the problem of estimating the parameters of a Gaussian distribution from just one frame of speech. The ML solution would be to set the mean of the distribution to be equal to the observed value and the variance equal to zero, so giving a perfect match to this observation but a probability of zero for any other feature vector. In reality, however, one observation provides no information about the variance of the features, but from general knowledge about speech it may be possible to make some reasonable guess. A practical solution to the problem of variances becoming unrealistically small during training is to set some suitable allowed minimum value as part of the training process.

Prior knowledge can be incorporated into model training in a more formal way if the training is performed to maximize the *a posteriori* probability of the models given the training data. Given the observed data ***Y*****, maximum *a posteriori* (MAP)** estimation of some HMM parameters θ involves finding values of θ that maximize P(θ|***Y***). Using Bayes' rule to express this probability in terms of P(***Y***|θ) gives the following expression for θ $_{\text{MAP}}$, the MAP estimate of the model parameters θ:

$$\begin{aligned}\Theta_{\text{MAP}} &= \arg\max_{\Theta} P(\Theta \mid \boldsymbol{Y}) \\ &= \arg\max_{\Theta} P(\boldsymbol{Y} \mid \Theta) P(\Theta),\end{aligned} \tag{11.1}$$

where P(θ) is the *a priori* probability of the model parameters θ. This method of training by incorporating prior information is often referred to as **Bayesian** learning. If no prior information is available, effectively P(θ) is assumed to be a constant and so the MAP estimate of θ is equivalent to the ML estimate.

Intuitively, if we have some idea about likely parameter values before encountering any speech data, combining this information with whatever data are available should help in the model estimation process, especially when there is only a limited quantity of training data. The difficulty lies in knowing what the prior information should be for typical model feature vectors and in specifying that prior information in such a way that it can then be used in obtaining the MAP estimate of the model parameters. However, for the task of model adaptation, amenable prior information exists in the form of the current estimates of the HMM parameters. By treating these HMM parameters as the prior information, it is possible to derive MAP re-estimation formulae that give new values for the HMM parameters to incorporate some adaptation data. The new estimates are a weighted sum of the original model estimates and the observed data, with the relative contribution of the observed data depending on how much data is available.

Bayesian adaptation provides an optimal mathematical framework for combining information from new data with existing models. However, it is only possible to adapt models for which adaptation data are available. Large-vocabulary systems may contain thousands of models, in which case even several minutes of adaptation data are unlikely to provide examples of all the model units.

11.4.2 Adaptation methods based on linear transforms

A straightforward method for improving the match between some speech data and a set of models is to apply a suitable linear transformation to the model parameters. One useful technique for adapting continuous-density HMM parameters is **maximum likelihood linear regression (MLLR),** whereby a linear transformation for the Gaussian distribution parameters is estimated in order to maximize the likelihood of the adaptation data. For example, given a model mean vector $\boldsymbol{\mu}$, a new mean $\hat{\boldsymbol{\mu}}$ can be derived as follows:

$$\hat{\boldsymbol{\mu}} = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}\,, \tag{11.2}$$

where $\boldsymbol{A}$ is a transformation matrix and $\boldsymbol{b}$ is a bias vector, both of which can be estimated given some speech data for adaptation.

An MLLR transform can also be estimated for the model variances. The variance transform can either be estimated separately from the mean transform, or alternatively the system can be constrained so that the same transformation matrix $\boldsymbol{A}$ in Equation (11.2) is also applied to transform the covariance matrix. It is also possible to share a single transform between different Gaussian mixture distributions and hence between different models. Thus the number of transforms can be chosen dependent upon the amount of data available for adaptation. If the amount of adaptation data is very limited, a single transform can be shared across all the Gaussian distributions in the models. As the amount of data increases, the number of transforms can be increased in such a way that transforms are shared between models that are similar.

For speaker adaptation, MLLR has been found to give worthwhile gains in recognition performance with just a few seconds of adaptation data, and performance then improves as the quantity of data increases. The method has also been found to be useful for adaptation to changes in the environment, although to provide a reasonable starting point it is beneficial to start by applying some predictive compensation method such as PMC.

The conventional starting point for speaker adaptation is a set of speaker-independent models that have been trained using training data from a wide variety of different individuals. It is, however, possible to use **speaker-adaptive training (SAT),** whereby the adaptation technique is incorporated within the training process as well as being used to adapt to individuals at recognition time. A linear transform is therefore computed for each speaker in the original training set. The speaker-independent model parameters are estimated in a way that takes into account the subsequent application of the (speaker-dependent) transform to improve the fit to the data for any one speaker. In this way, the speaker-independent models should be less influenced by speaker-specific characteristics (as these should be captured in the transforms), and should mainly capture the phonetic properties of different speech sounds. Such models may provide a better starting point for subsequent adaptation to any new speaker.

11.5 DISCRIMINATIVE TRAINING METHODS

We have already mentioned the problems that tend to occur with ML training methods when only a small quantity of training data is available. In practice the quantity of training data will always be limited, especially for speaker-independent and for large-vocabulary systems. In both Baum-Welch and Viterbi training each model is trained only to match the given data for its own class, so there is no guarantee that a model trained in this way will match badly to the alternative, incorrect classes. In recognition, however, the task is to find the model *M* for which *P(M*|***Y***) is the highest, and the system must be able to *discriminate* between the correct model and all the incorrect models. Training the models to maximize the likelihood of the training data will not necessarily give optimum recognition performance.

ML training is widely used and has achieved considerable success. However, the discrepancy between the ML training criterion and the requirements in recognition has led several workers to investigate alternative parameter estimation methods. These **discriminative training** techniques aim to maximize the ability of the models to distinguish between the different classes to be recognized, by estimating model parameters to match training data in a way that improves the likelihood of the correct models *relative* to the likelihood for the incorrect models. This training criterion is thus closely related to the needs of the recognition task.

11.5.1 Maximum mutual information training

One training method that is aimed at maximizing discrimination of a set of models is **maximum mutual information** (MMI) training. For a set of model parameters ? and some observed data $\boldsymbol{Y}_w$ representing a word w, the **mutual information** between the data $\boldsymbol{Y}_w$ and its corresponding word w is given by:

$$I(\boldsymbol{Y}_w, w \mid \Theta) = \log \frac{P(\boldsymbol{Y}_w, w \mid \Theta)}{P(\boldsymbol{Y}_w \mid \Theta) P(w \mid \Theta)} . \qquad (11.3)$$

The numerator in Equation (11.3) represents the joint probability of the data $\boldsymbol{Y}_w$ and the word w occurring together, while the denominator is the product of the probabilities of the two separate events. The mutual information *I(*$\boldsymbol{Y}_w$*,w*|*?)* represents the proportion of the total information that is 'shared' between the data and its corresponding model. If the model provides a good representation of the data, the mutual information should be high. The goal of MMI training is to maximize the total mutual information for the entire model set and all the data. To see that this training criterion is discriminative, we first note that:

$$P(\boldsymbol{Y}_w, w \mid \Theta) = P(\boldsymbol{Y}_w \mid w, \Theta) P(w \mid \Theta) . \qquad (11.4)$$

Thus, substituting back into Equation (11.3) and writing the logarithm of the ratio as the difference of the logarithms, we have:

$$I(\boldsymbol{Y}_w, w \mid \Theta) = \log P(\boldsymbol{Y}_w \mid w, \Theta) - \log P(\boldsymbol{Y}_w \mid \Theta) . \qquad (11.5)$$

Now if there are a total of *V* models, the last term in Equation (11.5) can be written as the sum of the joint probabilities of $\boldsymbol{Y}_w$ and each of the models, *v,* thus:

$$I(\boldsymbol{Y}_w, w \mid \Theta) = \log P(\boldsymbol{Y}_w \mid w, \Theta) - \log \sum_{v=1}^{V} P(\boldsymbol{Y}_w \mid v, \Theta) P(v \mid \Theta) . \tag{11.6}$$

Hence, to increase the mutual information, the probability of the observations given the correct model must increase more than the sum of the joint probabilities of the observations and all the possible alternative models.

Equation (11.6) is straightforward to compute for an isolated-word system. In the case of a connected-word system, this equation needs to be formulated in terms of model sequences, but the sum of probabilities over all word sequences becomes difficult to compute, because there may be a vast number of possible word sequences. Various approximations have been used to obtain some useful estimates for this term. MMI training has been incorporated in a number of research systems and often gives some benefit over ML methods, especially for model sets that use a fairly small number of parameters to represent the recognition vocabulary.

11.5.2 Training criteria based on reducing recognition errors

MMI training seeks to maximize the total discrimination capability of the models over the entire training set. Another approach to discriminative training is to use a training criterion that is specifically aimed at minimizing the recognition error rate on the training data. The aim is to improve recognition performance by giving most weight to those training examples that are most easily confused with other words. It does not matter if as a result the match becomes somewhat worse for examples that are easily recognized, provided that the match for the correct model is still better than the match for all incorrect models. The procedure known as **corrective training** starts by applying the standard forward-backward training algorithm to derive model estimates. These models are then used to perform recognition on the training data and hence to identify the confusable examples. These examples will include utterances that are misrecognized, but also those 'near-miss' utterances for which the match for an incorrect model is unacceptably close to the match for the correct model. For each of the selected examples, an adjustment is made to the model parameters. The value for this adjustment is chosen to increase the probability for the correct model and reduce the probability for the incorrect model. Corrective training is an intuitively appealing but largely experimental method, because both the model adjustment and the threshold for deciding on the 'near-miss' utterances are parameters that need to be determined experimentally.

A recent research direction for ASR is the development of discriminative training methods that generalize the corrective training and MMI approaches to provide a formal optimization criterion that is designed to minimize classification errors. These **minimum classification error (MCE)** methods use **generalized probabilistic descent (GPD)** training algorithms and represent a general framework that includes as special cases a number of discriminative approaches, including MMI and corrective training. The details of MCE and GPD training are outside the scope of this book, but Chapter 17 gives some references.

11.6 ROBUSTNESS OF RECOGNIZERS TO VOCABULARY VARIATION

There are many applications that only need a limited vocabulary of different words (see Chapter 15). However, even when people are given very specific instructions as to the words that they are allowed to say, there will inevitably be some occasions when those instructions are not followed. Often users say extra words in addition to the ones that are requested. For instance, the system might say "Please say one or two", to which the reply could be "I'll have two please". Input of this type will cause problems for a recognizer that expects only a single word, "one" or "two". However, by using **keyword spotting** techniques, it is possible to allow users freedom in their phrasing of responses, while only attempting to recognize certain keywords from a smaller vocabulary.

A common approach to keyword spotting uses continuous speech recognition that incorporates additional models to represent the acoustic background (often called "filler" or "garbage" models). The structure of the background models can vary from a simple one-state HMM with a Gaussian mixture output distribution trained on a suitable range of speech material, to networks of phonetic models or even networks representing word sequences that are typical of the application in which the system is being used. The recognition process produces a continuous stream of keywords and fillers, from which the keywords can be extracted to provide the recognizer output.

The ease and reliability with which keywords can be spotted will depend on the nature of the other speech material in which they are embedded and on more general acoustic properties of the signal. It is possible for a word spotter to compute a measure of **confidence** in its recognition decision, in order to provide an estimate of the probability that a word in the recognizer output is correct. This confidence measure is often derived from a likelihood ratio comparing the likelihood of a hypothesis including the keyword with that of a hypothesis that only includes filler models. The higher the value of the ratio, the greater the confidence in the recognition decision. The confidence estimate can then be evaluated against a threshold in order to decide whether or not the keyword is finally detected by the system. The value of the threshold may be set according to the application (i.e. whether it is more critical to miss a genuine keyword or to falsely detect a keyword when one is not present).

Another use for confidence measures is to detect and hence to reject **out-of-vocabulary** words (e.g. "three" for the example given in the first paragraph). It is also possible to reject utterances that cannot be recognized with enough confidence for some other reason (such as an unclear pronunciation or a very noisy signal).

CHAPTER 11 SUMMARY

- Any mismatch between the conditions under which a speech recognizer is used and those in which it was trained tends to cause problems. Possible mismatches include acoustic environment, transmission channel and speaker identity.
- Techniques for making features more robust to environmental changes include spectral subtraction for additive noise and cepstral mean subtraction for convolutional disturbance (due to channel-filtering effects for example).
- A popular and successful model-based method for dealing with noise is parallel model

combination (PMC). Models for clean speech are combined with models of the anticipated noise to obtain models for the corrupted speech.

- Robustness to acoustic variation caused by differences between speakers can be improved by normalizing the frequency scale for the feature analysis.
- Performance for any one speaker or environment can be greatly improved by adapting the models to the new conditions. Adaptation works best when the text of the adaptation data is known. However, provided the initial recognition performance is reasonable, unsupervised adaptation can also be useful.
- Bayesian (maximum *a posteriori:* MAP) adaptation provides a mathematical framework for combining some new adaptation data with an existing set of models, but requires data to be available for every model in the system.
- Alternatively a linear transform can be estimated to transform the model parameters to improve the match to some adaptation data. The technique of maximum likelihood linear regression (MLLR) computes a linear transformation for the parameters of continuous-density HMMs in order to maximize the likelihood of the adaptation data. By sharing transforms between models, this approach can be used with very little adaptation data.
- As an alternative to maximum-likelihood training, discriminative training methods determine HMM parameters to maximize the separation between correct and incorrect models. This training criterion is more closely related to the requirements for recognition and hence may lead to better performance.
- When users cannot be relied upon to keep to a specified vocabulary, keyword-spotting techniques can be used to recognize vocabulary words while accommodating non-vocabulary words with some general background models.

CHAPTER 11 EXERCISES

E11.1 What are the differences between predictive and adaptive noise compensation schemes? Explain the relative merits of each type.

E11.2 Explain the differences between the MAP and MLLR methods for speaker adaptation.

E11.3 What is the main advantage of discriminative training over maximum-likelihood training?

CHAPTER 12

Automatic Speech Recognition for Large Vocabularies

12.1 INTRODUCTION

The previous four chapters have concentrated on introducing underlying theory and algorithms for ASR, together with some of the techniques for using the algorithms successfully in real situations. The discussion so far has deliberately concentrated either specifically on distinguishing between a small number of different words or on more general methods irrespective of the particular recognition task. In this chapter, we consider issues relevant to systems for recognizing continuously spoken utterances using large vocabularies, which may be anywhere from a few thousand up to around 100,000 different words.

12.2 HISTORICAL PERSPECTIVE

One of the earliest major efforts aimed at large-vocabulary ASR was initiated during 1971 in the United States by the Advanced Research Projects Agency (ARPA), with funding for a five-year programme of research and development. The overall objective was to make significant progress in the field of speech understanding by developing several alternative systems. The specific goal was to achieve a level of performance that was expressed in terms of semantic errors (less than 10%) on a continuous speech recognition task with a total vocabulary size of at least 1,000 words but using constrained-language input.

Although isolated-word recognition using pattern-matching techniques had achieved some initial success by the time of this ARPA programme, it was not generally obvious then how to extend the approach to accommodate the contextual effects that were known to occur in continuous speech. Therefore most systems adopted what at that time was the more traditional approach, using two separate stages. The first stage began by detecting **phonetic features** (e.g. formant frequencies, energy in different frequency bands, etc.) that were known to be important for distinguishing different speech sounds. Rules were used to convert from the measured features to a hypothesized phonetic transcription, which usually included some alternatives. The second stage then converted this transcription to a recognized word sequence. Inevitably there would be errors in the initial phonetic transcription, but the hope was that these errors would be corrected by the higher-level post-processing. However, in practice the first stage was so error-prone that information was lost which could not be recovered later. As a consequence, all the systems using this **knowledge-based** approach gave disappointing performance. In fact, the only system to achieve the required level of performance used a completely different method, based on a systematic search of a large network of states with strong syntactic constraints, and it was one of the early large-vocabulary speech recognition systems using HMMs. The system was developed (somewhat separately from the main ARPA projects) at CMU by Lowerre (1976) as a Ph.D. project, extending the earlier pioneering work on HMMs by Baker (1975).

The results of the 1970s ARPA programme, while disappointing in terms of achievements for the money invested, provide a convincing demonstration of the benefits of **data-driven** statistical pattern matching over knowledge-based methods. In particular, the principle of delayed decision making is crucial, as it allows the overall best solution to be found incorporating all constraints, including those on construction of individual words and on allowed word sequences. This principle is fundamental to the design of all modern large-vocabulary speech recognizers.

Concurrent with the ARPA projects, research was in progress at IBM on the use of statistical methods for ASR. Early work was published by Jelinek (1976), independently of the work being carried out at CMU during the same period by Baker (1975). Work at IBM continued with an emphasis on applying HMMs to large-vocabulary speech recognition, and in the early 1980s the group focused on developing a system for dictation of office correspondence. The resulting system, “Tangora”, as described by Jelinek (1985), was a speaker-dependent, isolated-word, near-real-time recognizer with a 5,000-word vocabulary. Although this system required users to leave pauses between words, it established the principles underlying the use of HMMs for a large-vocabulary task. Since the mid-1980s, further developments in many laboratories have led to significant further progress, and systems are now able to recognize fluent, naturally spoken continuous speech with very large vocabularies. There are a variety of systems for **large-vocabulary continuous speech recognition (LVCSR)** in existence, both as commercial products and as research systems in laboratories. At present, the successful systems are all based on HMMs, usually incorporating many of the refinements described in Chapter 11, but also with components that are specific to demands imposed by the need to cope with large vocabularies.

12.3 SPEECH TRANSCRIPTION AND SPEECH UNDERSTANDING

Large-vocabulary speech recognition tasks fall into two quite distinct categories:

1. *Speech transcription:* The user wishes to know exactly what the speaker said, in the form that it would be transcribed by an audio typist to produce orthographic text. Such a system may be used for dictation, and for tasks such as producing transcripts of broadcast news programmes.
2. *Speech understanding:* The semantic content of the message is required, and any recognition errors do not matter provided that the meaning is not changed. In fact often the real requirement is for the system to perform the correct action, irrespective of what words are recognized. Speech understanding systems may involve an interactive dialogue between a person and a machine to retrieve information from some computerized database. Other uses include automatic information extraction, for example to summarize spoken reports or broadcasts.

The interactive nature of many speech-understanding tasks, together with the fact that the subject area is often restricted, means that the relevant vocabulary at any one point can be much smaller than the total vocabulary that is needed for more general transcription tasks. However, in order to interpret meaning of utterances, more detailed syntactic and semantic analyses are necessary than are required when just transcribing the words that were spoken. The principles of large-vocabulary recognition using HMMs apply both to transcription and to understanding, but the way in which the recognizer output is used is rather different. The first, main part of this chapter concentrates on transcription, while the latter part of the chapter briefly describes the use of large-vocabulary ASR in speech understanding systems.

12.4 SPEECH TRANSCRIPTION

The input speech waveform (typically sampled at 16 kHz) is first analysed into a sequence of acoustic feature vectors such as MFCCs (see Chapter 10). A popular choice is the first 12 cepstral coefficients and an overall energy feature together with first and second time derivatives of these features, giving a 39-element vector.

Once the input speech has been analysed into a sequence of feature vectors, the recognition task is to find the most probable word sequence *W* given the observed vector sequence ***Y***. Revisiting Bayes' theorem (see Section 9.2), but applying it to the task of finding a word *sequence,* the most probable sequence can be derived from the probability $P(W|\boldsymbol{Y})$ of any one sequence *W* as follows:

$$\hat{W} = \arg\max_{W} P(W \mid \boldsymbol{Y}) = \arg\max_{W} \frac{P(\boldsymbol{Y} \mid W)P(W)}{P(\boldsymbol{Y})} = \arg\max_{W} P(\boldsymbol{Y} \mid W)P(W)\,. \quad (12.1)$$

Equation (12.1) states that the most likely word sequence is the one which maximizes the product of $P(\boldsymbol{Y}|W)$ and $P(W)$. The first term denotes the probability of observing vector sequence ***Y*** given the word sequence *W^*, and is determined by an **acoustic model**. The second term represents the probability of observing word sequence *W* independently from the acoustic signal, and is determined by a **language model**. Chapter 9 focused on the task of calculating acoustic-model probabilities, which is fundamental to any speech recognition system based on statistical models. However, for all but the most simple of applications, the language-model probability is also a major factor in obtaining good performance: restrictions imposed by the language model can greatly reduce the number of different alternatives to be distinguished by the acoustic model. As with the acoustic model, the language model for LVCSR is usually a statistical model that is automatically trained on data. In the case of the language model, these data usually take the form of *text* material chosen to be representative of the recognition task.

Assuming that models have been trained, Figure 12.1 illustrates a framework for classifying an unknown utterance by computing $P(\boldsymbol{Y}|W)P(W)$. The language model postulates a word sequence ("ten pots" in this example[1]) and determines its probability $P(W)$. In order to calculate the acoustic-model probability $P(\boldsymbol{Y}|W)$, a

[1] The phrase "ten pots" will be used in this chapter to illustrate a variety of different points. This phrase was chosen to provide a simple example for which the phonetic and orthographic transcriptions are very similar. For convenience of notation, we will represent the vowel in "pots" with its orthographic transcription /o/ in place of the correct phonetic notation for southern British English /?/.

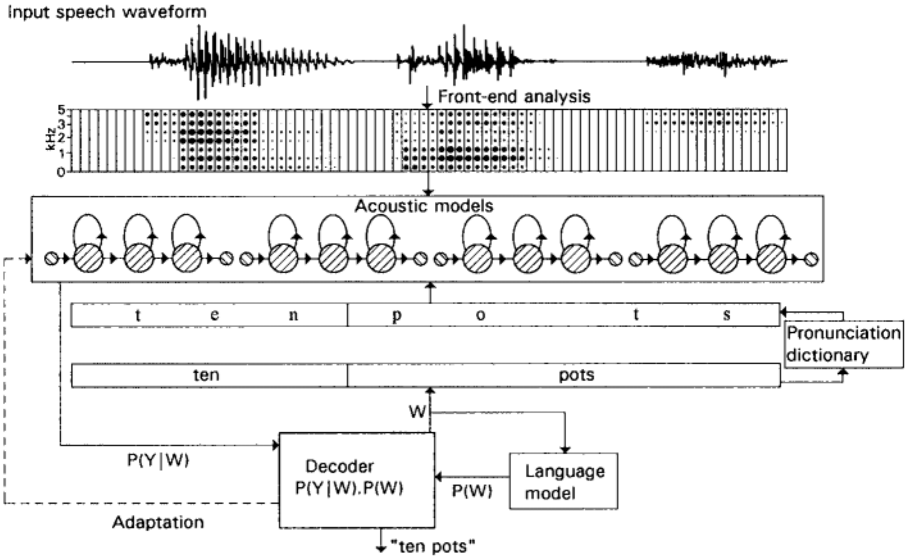


Figure 12.1 Framework for decoding a speech signal by computing the probability of a word sequence in terms of language-model and acoustic-model probabilities, shown for recognition of the phrase "ten pots". A simple filter-bank analysis is shown here for clarity of illustration, although in practice other features such as MFCCs would be used. Due to space limitations, only four of the seven models needed to represent the phone sequence in "ten pots" are shown.

composite model for the word sequence is generated. Rather than having a separate HMM for each word, the component models represent phone-size units and a pronunciation dictionary is used to specify the sequence of models for each word in the vocabulary. For any word sequence, the dictionary is used to look up the required sequence of phone models for each word, and these phone models are concatenated together to form the model for the word sequence. The probability of that model generating the observed acoustic sequence is calculated, and this probability is multiplied together with the language-model probability. In principle, this process can be repeated for all possible word sequences allowed by the language model, with the most likely sequence being selected as the recognizer output. In practice, decoding for LVCSR requires a very efficient search strategy for evaluating the vast number of different possibilities, as will be explained later.

12.5 CHALLENGES POSED BY LARGE VOCABULARIES

Issues for the design of large-vocabulary recognition systems include the following:

1. In continuous fluent speech, there are many instances when words cannot be distinguished based on acoustic information alone and it is necessary to rely on a language model for discrimination. Difficulties in making acoustic distinctions arise for two main reasons. Firstly, due to co-articulation between adjacent words, word boundaries are not usually

apparent in the acoustic signal. In some cases, two utterances may be linguistically different but acoustically very similar or even identical (as in the “grey day” versus “grade A” example given in Chapter 1). Secondly, the pronunciation of many words, particularly function words, can be so reduced that there is very little acoustic information at all.

2. The memory and computation requirements can become excessive. In recent years, advances in computer technology have greatly reduced the impact of this limitation. However, memory and computation are still influential, especially in determining the choice of search mechanism for use in decoding.
3. As the vocabulary size increases, it becomes increasingly difficult to provide enough representative examples of all the words, both as text to train the language model and as spoken examples to train the acoustic model.

Many of the design features of modern LVCSR systems are determined by the need to deal with these issues. The design of the acoustic model, the language model and the decoding operation are all crucial factors for the success of an LVCSR system. The following three sections describe each of these three components in turn.

12.6 ACOUSTIC MODELLING

Although some early systems used HMMs with discrete distributions for their emission p.d.f.s (e.g. Lee (1989)), current systems generally use fully continuous distributions or tied-mixture distributions, usually with diagonal covariance matrices. These latter types will be the focus of the explanation given here, which is based mainly on descriptions of the research system developed at Cambridge University (e.g. Young (1996)). This system is one of the most successful systems to date, but there are many other systems that have fairly similar structure and give broadly comparable performance, although they differ in various details.

The need to make the best use of any available acoustic training data has important consequences for the design of the acoustic-model component. With a large vocabulary, it is impractical to expect any one person to provide enough examples to train models for all the words from scratch, even if the system is intended for speaker-dependent operation. Therefore, a speaker-independent model set is used, at least to provide a starting point. Speaker-adaptation techniques are often used to improve performance for any one individual. Unsupervised adaptation may be performed using the recognizer output, as shown by the dotted data path in Figure 12.1. In addition, for a system to be used by one known person, that person can be required to speak some specific utterances, which can be used for supervised model adaptation before the person uses the system to perform any real task.

Even with several speakers to provide the data, it is not practical to train a separate model for each word in a large-vocabulary system. Even if it were practical, this approach would not make the best use of the data, as it does not take account of the fact that different words can share sub-components. Therefore large-vocabulary systems are based on **sub-word** models. The usual method, as shown in Figure 12.1, is to use models of phone-size units, with the sequence of phones for each word being specified in a pronunciation dictionary. Thus, the requirement for the training is to provide sufficient examples of all the phone-size units, and all the words in the vocabulary will not necessarily have occurred in the training data. In fact, provided suitable models are available, words can be added to the vocabulary at any time simply by extending the pronunciation dictionary.

12.6.1 Context-dependent phone modelling

As approximately 44 phonemes are needed to represent all English words, this number of models would be the minimum needed to build word models for English. However, the effects of co-articulation are such that the acoustic realization of any one phoneme can vary greatly with acoustic context. Therefore **context-dependent** HMMs are generally used, with different models for different phonetic contexts. Additional variation tends to arise because many speakers will, either consistently or occasionally, use word pronunciations that are different from those given in the dictionary. Although alternative pronunciations can be included in the dictionary, it is difficult to include every possible pronunciation and any that are not covered will need somehow to be accommodated in the chosen set of context-dependent HMMs.

The simplest and most popular approach is to use **triphones,** whereby every phone has a distinct HMM for every unique pair of left and right neighbours. For example, consider the word "ten". When spoken in isolation, this word could be represented by the sequence sil ten sil, with the sil model being used for silence at the start and end. Using triphones, with the notation ${}_{x}y_{z}$ to denote phone y preceded by phone x and followed by phone z, the word would be modelled as

$$\mathrm{sil}\quad {}_{\mathrm{sil}}\mathrm{t}_{\mathrm{e}}\quad {}_{\mathrm{t}}\mathrm{e}_{\mathrm{n}}\quad {}_{\mathrm{e}}\mathrm{n}_{\mathrm{sil}}\quad \mathrm{sil}.$$

Now consider the phrase "ten pots", for which the triphone sequence would be

$$\mathrm{sil}\quad {}_{\mathrm{sil}}\mathrm{t}_{\mathrm{e}}\quad {}_{\mathrm{t}}\mathrm{e}_{\mathrm{n}}\quad {}_{\mathrm{e}}\mathrm{n}_{\mathrm{p}}\quad {}_{\mathrm{n}}\mathrm{p}_{\mathrm{o}}\quad {}_{\mathrm{p}}\mathrm{o}_{\mathrm{t}}\quad {}_{\mathrm{o}}\mathrm{t}_{\mathrm{s}}\quad {}_{\mathrm{t}}\mathrm{s}_{\mathrm{sil}}\quad \mathrm{sil}.$$

The two instances of the phone [t] are represented by different models because their contexts are different. Note that the triphone contexts span word boundaries, so that the first and last triphones used to represent a word depend on the preceding and following words respectively. For example, if the phrase were "ten dogs", the last triphone used to model "ten" would be ${}_{e}n_{d}$ rather than ${}_{e}n_{p}$. This use of **cross-word triphones** enables co-articulation effects across word boundaries to be accommodated, but creates complications for the decoding process as the sequence of HMMs used to represent any one word will depend on the following word.

The decoding task can be greatly simplified by using only **word-internal triphones,** whereby 'word boundary' acts as a context and so the sequence of HMMs is fixed for each word. Thus, in the above example the triphones ${}_{e}n_{p}$ and ${}_{n}p_{o}$ would be replaced by ${}_{e}n_{-}$ and ${}_{-}p_{o}$ respectively, with—being used to represent a word boundary. Early triphone systems were restricted to word-internal triphones, but the inability to model contextual effects across word boundaries is a serious disadvantage and current systems generally include cross-word context-dependent models. The consequences for decoding are explained in Section 12.8.

12.6.2 Training issues for context-dependent models

For a language with 44 different phones, the number of possible triphones is 44^3= 85,184. In fact, phonotactic constraints are such that not all of these triphones can occur. However, an LVCSR system which includes cross-word triphones will still typically need around 60,000 triphones. This large number of possible triphones poses problems for training the models:

- The total number of parameters needed for the models is very large: it is usual to use three-state models with somewhere in the region of 10 mixture components to represent the output distribution for each state. This number of mixture components tends to be needed to represent the wide range of speakers (including a range of different accent types) who must be accommodated within a single model. Assuming that diagonal covariance matrices are used with 39-element acoustic feature vectors and 10 mixture components, each state would require around 790 parameters (39×10 means, 39×10 variances, and 10 mixture weights). Thus 60,000 three-state models would have a total of over 142 million parameters. Any feasible quantity of training data would not be large enough to train this number of parameters adequately.
- In any given set of training data, many triphones will inevitably not occur at all, especially if cross-word triphones are allowed (as it is very difficult to include all the phone combinations that might occur in all possible sequences of words). Thus some effective method is required for generating models for these **unseen triphones** that do not occur in the training data.

Similar issues have already been mentioned as difficulties with using whole-word models for large vocabularies. However, when using smaller model units that are meaningful in phonetic terms, it is easier to see how the problems can be reduced. The challenge is to balance the need for detail and specificity in the models against a requirement to have enough training data to obtain robust parameter estimates. To tackle the problem various different **smoothing** techniques have been used:

1. *Backing off:* When there are insufficient data to train a context-specific model, it is possible to **back off** and use a less-specific model for which more data are available. For example, one approach is to replace a triphone by the relevant **biphone**[2]**,** representing the phone dependent on only one adjacent context, which may be either to the left or to the right. Given a choice between the left or the right biphone context, it is generally better to choose the right context as articulation tends to be anticipatory, such that following context has a greater influence than preceding context. If there are insufficient examples to train a biphone, it is possible to resort to the context-independent phone model, or **monophone.** The backing-off mechanism ensures that every model in the final system is adequately trained, but can result in only a relatively small number of full triphone models, so that several contexts are not modelled very accurately.
2. *Interpolation:* A greater degree of context dependency can be retained by **interpolating** the parameters of a context-specific (triphone) model with those of a less-specific model (such as a biphone or monophone), to give model parameters which represent some compromise between the two sets. Thus some of the context dependency of the original triphone models is preserved, while increasing their robustness by taking into account additional (less specific) data.
3. *Parameter sharing:* An alternative is to take all the triphones representing any one phone, apply some form of **clustering** procedure to group the individual

[2] Note that the term **biphone** is used to refer to a phone that is dependent upon a single context (either left or right), and that this unit is different from the **diphone** unit discussed in Chapter 5, which represents the region from the middle of one phone to the middle of the next.

models (or parts of models) into clusters with similar characteristics, and share the parameters between them. This sharing of parameters, often referred to as **parameter tying,** allows the data associated with similar states to be pooled to improve the robustness of the parameter estimates. This approach can retain a higher degree of context specificity than is possible with the first two methods.

Although both backing off and parameter interpolation have been used with some success, the greater power of more general parameter sharing to obtain a better compromise between accuracy and robustness is such that this method is now widely used in LVCSR. The method is described in more detail below.

12.6.3 Parameter tying

The technique of tying provides a general mechanism for sharing parameters between models or parts of models. One example is provided by the tied-mixture distributions introduced in Chapter 9, where the means and variances of each mixture component are tied across all model states. Smoothing the parameters of context-dependent models represents another application of tying. Here tying is usually applied to all model parameters for a subset of the triphones representing a phone. The aim is to tie together those models or states for which any differences in the acoustic realizations are not significant for phonetic discrimination.

Initial developments in parameter sharing between context-dependent models concentrated on clustering together triphone *models,* to give **generalized triphones**. However, this approach assumes that the degree of similarity between any two models is the same for all the states in the models. In fact, the different effects of left and right context are such that this assumption is rarely justified. For example, consider three triphones of /e/: $^{t}e^{n}$, $^{t}e^{p}$ and $^{k}e^{n}$. The first state of the $^{t}e^{n}$ and $^{t}e^{p}$ triphones can be expected to be very similar, while the last state of the $^{t}e^{n}$ and $^{k}e^{n}$ triphones will be similar. Thus tying at the state level offers much more flexibility to make the most effective use of the available data for training a set of models. We will now consider two important issues associated with the use of state tying: firstly the general procedure used to train the tied-state multiple-component mixture models (assuming that it is known which states to tie together), and secondly the choice of clustering method used to decide on the state groupings. The discussion focuses on state tying, but the principles apply in the same way when the tying is applied to complete models.

12.6.4 Training procedure

Careful design of the training procedure is essential to maximize the robustness and accuracy of the final set of tied-state context-dependent HMMs. When training subword models, it is not usual for the individual speech segments to have been identified and labelled in the available training data. In fact it is most likely that the data will have been transcribed as a sequence of words but not segmented at all. Rather than attempting to segment these data, they can be used directly for parameter estimation by adopting the **embedded training** approach described in Section 9.11. When using sub-word models, it is necessary first to construct a model for each word from the sub-word units, before then constructing a model for the whole utterance from the individual words. Similarly to the procedure outlined in Section 12.4 for recognition, the pronunciation dictionary is used to look up the phone sequence required to represent each training utterance. A composite HMM is constructed by concatenating the appropriate sub-word models, and the relevant statistics for re-estimation are accumulated over all occurrences of each model.

Phone sequence constraints across different utterances are such that the embedded training method should generally be effective in associating appropriate speech frames with each model state, provided that in the early stages of training each model is used in a sufficient range of different contexts. For this situation it is even adequate to use the very simple 'flat' initialization of all the model parameters to identical values (see Section 9.9). It is usual to start with single-Gaussian distributions and train simple monophone models. Because there are very many examples of each one, these monophones can be trained very robustly, and provide a good basis for initializing the more specific context-dependent models. A typical procedure for training context-dependent models is illustrated in Figure 12.2, and summarized below:

1. A set of monophone HMMs, using single-Gaussian output distributions with diagonal covariance matrices, is created and trained.
2. All the training utterances are transcribed in terms of the complete set of possible triphones. For each triphone, an initial model is created by cloning the appropriate monophone. The transition probability matrix is typically not cloned, but is tied across all triphones of a phone. The triphone model parameters are re-estimated and the state occupancies, which represent the expected number of observations used to estimate the parameters of each triphone (see Section 9.5.2), are retained for later use.
3. For the set of triphones representing each phone, similar states are clustered together to create tied states (see Sections 12.6.5 and 12.6.6 for more explanation). As part of the state tying process, it is important to check that there are sufficient data associated with each tied state. This situation can be achieved by only allowing clusters for which the total state occupancy (i.e. the estimated 'count' of number of frames for which the state is occupied) exceeds a suitable threshold value (typically around 100). The parameters of the tied-state single-Gaussian models can then be re-estimated. The use of tying does not alter the form of the re-estimation formulae and can be made transparent to the re-estimation process if the data structures used to store the information are set up appropriately. Storage can be set up for accumulating the numerator and denominator for re-estimating each parameter of each tied state, with individual states simply pointing to the relevant storage.
4. Finally, multiple-component mixture models are trained using the iterative mixture splitting procedure explained in Section 9.10.4.

Delaying the introduction of the multiple-component Gaussians until the final stage has a number of advantages:

- The difficulties associated with training untied triphone mixture distributions are avoided, as multiple mixture components are only introduced once the model inventory has been set up to ensure adequate training data for each state.
- The state tying procedure is simplified because, by using single-Gaussian

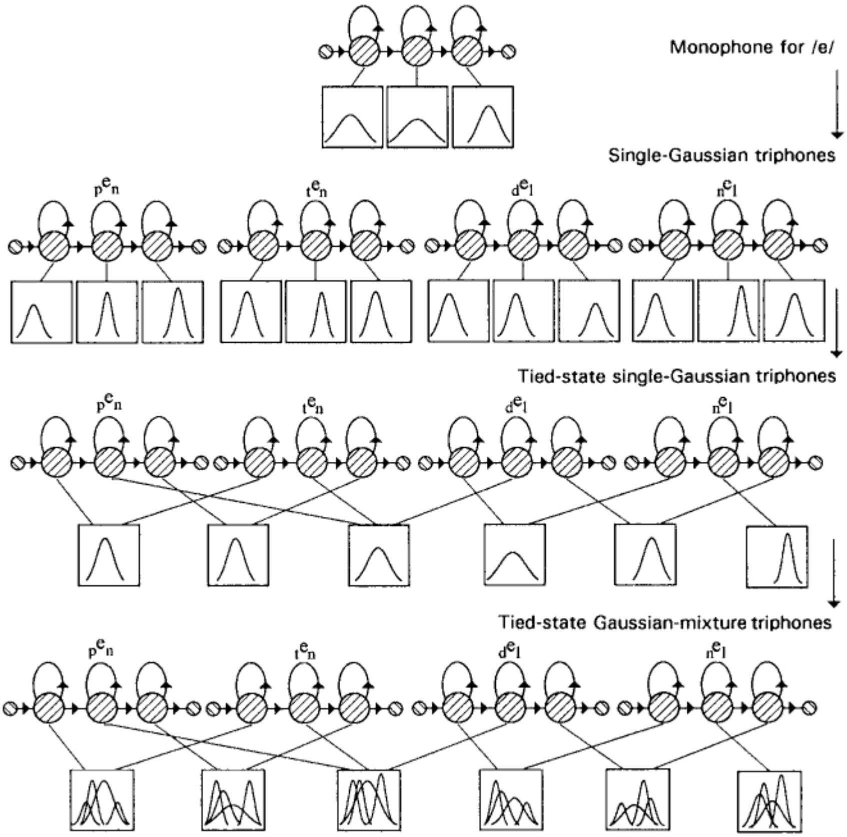


Figure 12.2 Sequence of stages for training tied-state Gaussian-mixture triphones, illustrated for a group of triphones representing the /e/ phoneme.

distributions, it is much easier to compute a similarity measure and to calculate the parameters of the combined states (see Section 12.6.6).

- By not introducing the mixture distributions at the monophone stage, the process avoids potential complications that could arise if the mixture components were used to accommodate contextual variation which would at a later stage be covered by the context-dependent models. It is generally better to accommodate contextual influences explicitly so that the predictable nature of these effects can be exploited as far as possible. The multiple mixture components are then needed mainly to allow for the fact that any one model represents data from a wide variety of different speakers.

In addition to the benefits in terms of robustness, computation and storage, state tying has the potential to lead to models with better discrimination. The potential advantages of sub-word over whole-word models in terms of discrimination power have already been mentioned. These arguments extend to the use of state tying. If the differences between the acoustic realizations associated with two different model states are simply a consequence of random variation, it is detrimental for these differences to be included in the models. By combining them into a single model state, discrimination will be more focused on those regions of words containing the most useful acoustic cues. This benefit is dependent upon finding an appropriate method for determining which states to tie together.

12.6.5 Methods for clustering model parameters

Clustering methods for grouping together similar states can be divided into two general types. These methods can be used to cluster triphone states as follows:

1. *Bottom-up clustering:* Starting with a separate model for each individual triphone, similar states are merged together to form a single new model state. The merging process is continued until some threshold is reached which ensures that there are adequate data to train each new clustered state. This data-driven approach is often referred to as **agglomerative clustering.** The method should ensure that there are sufficient data to train every state in the final set, while keeping the models as context-specific as possible given the available training data. However, for any triphones that do not occur at all in the training data, it is still necessary to back off to more general models such as biphones or monophones.
2. *Top-down clustering:* Initially all triphones for a phoneme are grouped together and a hierarchical splitting procedure is used to progressively divide up the group according to the answers to binary yes/no questions about either the left or the right immediately adjacent phonetic context. The questions are arranged as a **phonetic decision tree,** and the division process starts at the root node of the tree and continues until all the leaf nodes have been reached. All the states clustered at each leaf node are then tied together. A tree is generated for each state of each phone. An example showing the use of a decision tree to cluster the centre state of some /e/ triphones is shown in Figure 12.3. The context questions in the tree may relate to specific phones (e.g. "Is the phone to the right /l/?"), or to broad phonetic classes (e.g. "Is the phone to the left a nasal?"). Using the tree, the correct tied state to use for any given context can be found by tracing the path down the tree until a leaf node is reached (see Figure 12.3).

The main advantage of the top-down approach to clustering is that a context-dependent model will be specified for *any* triphone context, even if that context did not occur in the training data. It is thus possible to build more accurate models for unseen triphones than can be achieved with the simple backing-off strategy, assuming that the questions in the tree are such that contexts are grouped appropriately. Although the tree could be constructed by hand based on phonetic knowledge, this approach does not work very well in practice, as it does not take into account the acoustic similarity of the triphones in the data. It is, however, possible to construct trees automatically by combining the use of phonetic questions with tests of acoustic similarity and a test for sufficient data to represent any new division. This automatic construction provides generalization to unseen contexts while maintaining accuracy and robustness in the acoustic models. A popular version of this effective technique for constructing phonetic decision trees is explained in more detail in the next section.

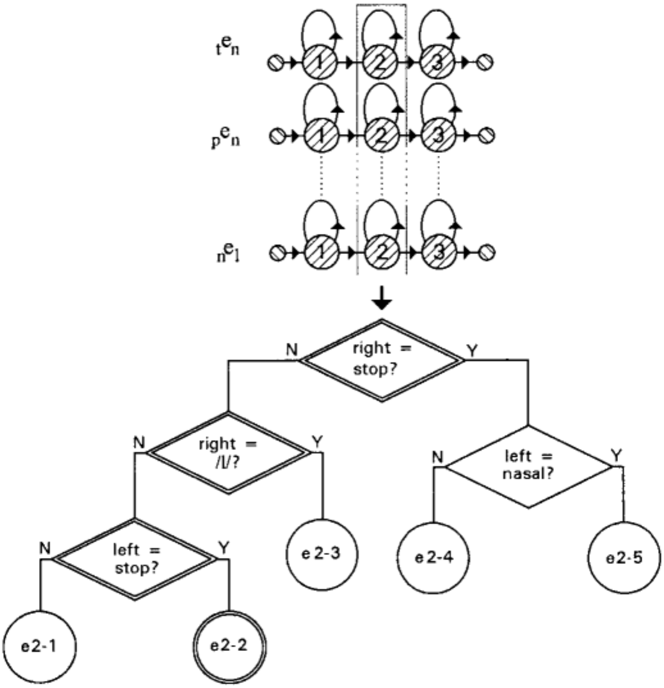


Figure 12.3 Example illustrating the use of a phonetic decision tree to cluster the centre state for a group of triphones of the /e/ phoneme. Each triphone is moved down the tree until a terminal 'leaf' node is reached (shown as circles), and a new model state is formed from the members of the cluster. The tree shown here is a very simple one intended simply as an illustration of the principles of clustering, and in any real application there will be many more questions before a terminal node is reached. The tree can then be used to find the appropriate clustered state for any given context. As an example, the route down this simple tree is shown (using double lines round the chosen decision boxes and the final leaf node) for the context of a preceding /t/ and a following /n/. In this context the clustered state e2–2 will be used.

12.6.6 Constructing phonetic decision trees

First, linguistic knowledge is used to choose a set of possible context questions that might be used to divide the data. This question set will usually include tests for each specific phone, tests for phonetic classes (e.g. stop, vowel), tests for more restricted classes (e.g. voiced stop, front vowel) and tests for more general classes (e.g. voiced consonant, continuant). Typically, there will be over 100 questions for the left context and a similar number for the right context. For each state of each phone, the aim is to build a tree where the question asked at each node is chosen to maximize the likelihood of the training data given the final set of tied states. A condition is imposed to ensure that there are sufficient data associated with each tied state (i.e. that the total occupancy of the tied state exceeds some threshold).

It would in principle be possible to build all possible tree architectures (for all states), train a set of models for each architecture, and choose the set for which the likelihood of the training data is the highest while satisfying the occupancy condition for each of the final tied states. This strategy would, however, be computationally intractable. Fortunately, by making the assumption that the assignment of acoustic observations to states is unchanged from the assignment for the original triphone models, it is possible to build a tree in a computationally efficient manner using just the state occupancies and the triphone model parameters. When using single-Gaussian distributions, this information is sufficient to calculate new model parameters for any putative combination of the individual triphone states.

The tree-building process starts by placing all the states to be clustered together at the root node of the tree. The mean and variance vectors are calculated assuming that all the states $\boldsymbol{S}$ are tied. Using these values of the model parameters it is then possible to estimate $L(\boldsymbol{S})$, the likelihood of the data associated with the pool of states $\boldsymbol{S}$. The next step is to find the best question for splitting $\boldsymbol{S}$ into two groups. For each question q, the states are divided up according to the answer to the question and new model parameters are computed. The likelihoods $L(\boldsymbol{S}\ (q))$ and $L(\boldsymbol{S}\ (q))$ can then be calculated for the sets of data corresponding to the answers^{y} “yes” and “^{n}no” respectively. For question q the total likelihood of the data associated with the pool of states will increase by:

$$\Delta L_q = L(\boldsymbol{S}_y(q)) + L(\boldsymbol{S}_n(q)) - L(\boldsymbol{S})\,. \tag{12.2}$$

Thus by computing ΔL_q for all possible questions, the question for which this quantity is the maximum can be selected. Two new nodes are created and the splitting process is then repeated for each of the new nodes, and so on. The splitting procedure is terminated when, for all of the current leaf nodes, the total occupancy of the new tied state which could be created at that node falls below the designated occupancy threshold. An additional termination condition is also used, whereby the splitting is halted when the increase in likelihood which would result from a split falls below a specified likelihood threshold. This second termination condition avoids the unnecessary use of different models for contexts which are acoustically similar (even if sufficient data are available for separate models to be used).

Once the tree has been constructed, this tree can be used to accomplish the state tying required for step 3 of the training process described in Section 12.6.4.

12.6.7 Extensions beyond triphone modelling

A useful feature of the phonetic decision tree approach is that it can be extended beyond simple triphone contexts. For example, decision trees can be built using questions relating to the next-but-one left and right contexts as well as the immediately adjacent contexts. The resulting models are often referred to as **quinphones,** as they can incorporate information over a sequence involving up to five phones. Questions relating to the presence of word boundaries can also be included. When building these complex decision trees with such a large number of possible contexts, it is not usually practical to start by training a fully context-dependent system because such a system would typically require a vast number of models and state occupancies to be stored. It is preferable to begin with some more manageable model set and use Viterbi alignment to provide a state-level segmentation of the data. If a tied-state triphone system has been built first, this system can be used to provide a good alignment upon which to base the derivation of a decision tree for a word-boundary-dependent quinphone system.

As well as incorporating context dependency in the acoustic models, it can be beneficial to deal with male and female speech separately, because the differences between male and female speech tend to be much greater than the differences between talkers of the same sex. Models built in this way are commonly described as **gender dependent**. One way of training gender-dependent models robustly is to introduce gender dependency at the final training stage by cloning the trained multiple-component mixture models and then re-estimating the state means and mixture weights for each model set using the data for the male and female speech separately. State variances are usually kept gender independent in order to avoid the robustness issues that would otherwise arise due to the more limited data available to train variances for each gender separately. Once the gender-dependent models have been trained, a straightforward method for using them in recognition is to run two recognizers in parallel, one using the 'male' models and the other using the 'female' models. The system output is then taken from the recognizer which gives the highest probability for an utterance (where the utterance represents a sequence of words known to have been spoken by the same person).

12.7 LANGUAGE MODELLING

In any language, there are syntactic, semantic and pragmatic constraints that have the effect of making some sequences of words more likely than others. For an ASR system that is intended for a particular application domain, the language that the system can recognize may be quite limited, in terms both of vocabulary size and of utterance syntax. However, for more general recognition of large vocabularies, *any* word sequence must be allowed, but different probabilities need to be assigned to different sequences. The purpose of the language model is to make effective use of linguistic constraints when computing the probability of the different possible word sequences. Assuming a sequence of *K* words, $W=w_1, w_2, \ldots, w_k$, the sequence probability *P(W)* can be expanded in terms of conditional probabilities as follows:

$$P(W)=P(w_1,w_2,\ldots,w_K)=\prod_{k=1}^{K}P(w_k\mid w_1,\ldots,w_{k-1})\,. \tag{12.3}$$

This expression simply states that the probability of the word sequence can be decomposed into the probability of the first word, multiplied by the probability of the second word given the first word, multiplied by the probability of the third word given the first and the second words, and so on for all the words in the sequence. Initially, the probabilities of the words will be influenced mostly by general constraints on the types of words that are most likely to start utterances. As the utterance continues, an increasing number of words will have already been spoken and so it becomes easier to predict the next word.

For any natural language, there is of course a vast number of possible word combinations and hence a huge number of conditional probabilities that might be required. It is not feasible to specify all of these probabilities individually, and so some modelling assumptions are needed to make the task of specifying the conditional probabilities more manageable.

12.7.1 *N*-grams

A simple solution to the problem of estimating word-sequence probabilities is to use ***N*-grams** (where *N* is a small number). Here it is assumed that the probability of observing any word w_k depends only on the identity of the previous *N*-1 words (so that each conditional probability depends on a total of *N* words, including the current word). Using *N*-grams Equation (12.3) can be approximated as follows:

$$P(W) = \prod_{k=1}^{K} P(w_k \mid w_1, \ldots, w_{k-1}) \approx \prod_{k=1}^{K} P(w_k \mid w_{k-N+1}, \ldots, w_{k-1}) . \qquad (12.4)$$

If *N*=1 the model is a **unigram** and just represents the probability of the word. A **bigram** (*N*=2) models the probability of a word given the immediately preceding word, while a **trigram** *(N*=3) takes into account the previous two words.

To illustrate the use of *N*-grams to estimate word-sequence probabilities, consider the phrase "ten pots fell over". For a unigram model, the probability of the sequence is obtained simply by multiplying together the word probabilities:

$$P(\text{ten pots fell over}) \approx P(\text{ten})\, P(\text{pots})\, P(\text{fell})\, P(\text{over}) . \qquad (12.5)$$

In the case of a bigram model the probability of the sequence is estimated as:

$$\begin{aligned} P(\text{ten pots fell over}) \approx\ & P(\text{ten} \mid \text{START})\, P(\text{pots} \mid \text{ten}) \\ & P(\text{fell} \mid \text{pots})\, P(\text{over} \mid \text{fell}), \end{aligned} \qquad (12.6)$$

where START is used to indicate the beginning of the sequence. For a trigram model, the probability estimate becomes:

$$\begin{aligned} P(\text{ten pots fell over}) \approx\ & P(\text{ten} \mid \text{START})\, P(\text{pots} \mid \text{START ten}) \\ & P(\text{fell} \mid \text{ten pots})\, P(\text{over} \mid \text{pots fell}). \end{aligned} \qquad (12.7)$$

In principle, *N*-grams can be estimated using simple frequency counts from training data to obtain maximum-likelihood estimates for the required probabilities. Considering the bigram (w_{k-1}, w_k) and the trigram (w_{k-2}, w_{k-1}, w_k), the conditional probabilities $P(w_k|w_{k-1})$ and $P(w_k|w_{k-}2_{,}\ w_{k-1})$ could be estimated thus:

$$\hat{P}(w_k \mid w_{k-1}) = \frac{C(w_{k-1}, w_k)}{C(w_{k-1})}, \quad \hat{P}(w_k \mid w_{k-2}, w_{k-1}) = \frac{C(w_{k-2}, w_{k-1}, w_k)}{C(w_{k-2}, w_{k-1})}, \qquad (12.8)$$

where the notation *C(x)* is used to represent the count of number of examples of *x*. For the examples of the bigram (fell over) and the trigram (pots fell over) we have:

$$\hat{P}(\text{over} \mid \text{fell}) = \frac{C(\text{fell over})}{C(\text{fell})}, \quad \hat{P}(\text{over} \mid \text{pots fell}) = \frac{C(\text{pots fell over})}{C(\text{fell over})} . \qquad (12.9)$$

12.7.2 Perplexity and evaluating language models

The task of the language model can be viewed as one of predicting words in a sequence, and a good language model will be one that provides a good predictor of the word in any position based on the words observed so far. Given a particular sequence of *K* words in some test database, the value of *P(W)* for that sequence provides a measure of how well the language model can predict the sequence: the higher the value of *P(W),* the better the language model is at predicting the word sequence. Word sequences differ in length, and an average measure of the probability per word is obtained by taking the K^{th} root of the probability of the sequence. The inverse of this probability defines the **perplexity,** *PP(W),* thus:

$$PP(W) = \left[P(w_1, w_2, \ldots, w_K)\right]^{-1/K} = \left[\prod_{k=1}^{K} P(w_k \mid w_1, \ldots, w_{k-1})\right]^{-1/K} . \qquad (12.10)$$

For any given language model, it is possible to calculate the perplexity of some corpus to be recognized. For artificially constrained tasks with a rigid syntax, perplexity can be interpreted as being equivalent to the average number of different words that would need to be distinguished at any point in the word sequence, if all words at any particular point were equiprobable. Perplexity is therefore sometimes referred to as representing an **average branching factor**. The lower the value of perplexity, the fewer the number of alternatives that must be considered. The lowest possible value of perplexity is 1, but this value would only be obtained if all the individual word probabilities were equal to 1, such that only one word sequence could be recognized by the system. At the other extreme, if any word in a sequence is assigned a probability of zero by the language model, then the probability of the complete sequence will be equal to zero and the perplexity will be infinitely large.

As we have already seen, a major challenge for any model of natural language is to avoid probabilities of zero by not excluding any of the vocabulary words, while making the prediction of the next word as good as possible by having only a few high-probability alternatives at any one point. A good language model should give a low value of perplexity on a large corpus of representative text material (with no part of that material having previously been used to train the model).

The perplexity measure provides a means for evaluating alternative possible language models on some test corpus without needing to run a complete recognition experiment. A perplexity evaluation allows the language-model component to be assessed independently from the acoustic-model component, but it cannot take into account any interactions between the two models: good discrimination by the language model may not have much effect on recognition performance if the words concerned are acoustically very distinct, but could have a large effect for acoustically confusable words. Furthermore, any effects of the search algorithm cannot be allowed for in the perplexity calculation. Thus perplexity on a test data set is helpful for comparing alternative language models and also provides a useful indicator of the difficulty of the recognition task to be performed by the acoustic models, but the final test must be in terms of the recognition accuracy of the complete system.

12.7.3 Data sparsity in language modelling

Maximum-likelihood estimates obtained from frequency counts using expressions such as those in Equation (12.8) are a good approximation to the true probabilities, provided that the sample size is large in relation to the number of possible outcomes. Unfortunately, in the case of N-gram modelling there are a vast number of possible outcomes. A vocabulary of V words provides V^2 potential bigrams and V^3 potential trigrams. For a 20,000-word vocabulary there are thus 400 million possible word bigrams and eight million million possible trigrams! It would be completely impracticable to provide sufficient speech data to determine the language-model probabilities, and instead very large text corpora are used. Even so, while typical text corpora may contain over 100 million words, most of the possible bigrams and the vast majority of possible trigrams will not occur at all in the training data and many others will only appear once or twice.

Data sparsity is a much greater problem for the language model than for the acoustic model, due to the much larger size of the inventory of basic units (words rather than phones). As a consequence of this severe data-sparsity problem, it is not possible to rely only on simple frequency counts for estimating language-model probabilities, as many of the bigram and trigram probabilities would be set to zero (so that it would then be impossible to recognize any word sequence containing one of these combinations) and many other probabilities would be poorly estimated. The successful use of N-grams for LVCSR is dependent upon the use o**f smoothing** techniques for obtaining accurate, robust (non-zero) probability estimates for *all* the possible N-grams that can occur for a given vocabulary. Zero probabilities and low non-zero probabilities are adjusted upwards, and high probabilities are reduced. Thus the overall effect is to make the probability distributions more uniform, and hence ‘smoother’. Some different aspects of smoothing algorithms are described briefly in the following sections.

12.7.4 Discounting

For any set of possible ‘events’, such as bigrams or trigrams, the sum of the probabilities for all the possibilities must be equal to one. When only a subset of the possible events occurs in the training data, the sum of the probabilities of all the observed events must therefore be less than one. This effect can be achieved by reducing the observed frequency counts. The process is generally known as **discounting,** and is often described in terms of ‘freeing’ **probability mass** from the observed events which can then be redistributed among the unseen events.

Several different methods have been used to achieve discounting, and these methods differ in the way in which the reduced probabilities are calculated. Full coverage of discounting methods is outside the scope of this book but, for example, one simple but effective technique is **absolute discounting**. Here some small fixed amount (between zero and one) is subtracted from each frequency count (the numerators in Equation (12.8) or in the examples shown in Equation (12.9)). Thus the probability of every observed event is reduced, but the effect decreases as the observed frequency count increases (when the maximum-likelihood estimate should be more reliable). However, the same discount value may not be optimum for the full range of frequency counts. In particular, there is some evidence that absolute discounting imposes too great a reduction in the probability of events that occur only once or twice. A variant of the technique overcomes this problem by having separate discount values for these rare events.

12.7.5 Backing off in language modelling

Probability mass which has been made available as a result of discounting can be allocated to the unseen events. In estimating the probabilities of the unseen events, it is desirable to make use of any information that is available about the relative probabilities of the different events. In the case of *N*-grams, useful information may be available in the form of the probability according to a more general distribution. Thus it is possible to formulate a recursive procedure of **'backing off':** if a trigram is not observed, the model backs off to the relevant bigram probability, and if the bigram is not available, it backs off further to the unigram probability. For any words which do not occur in the training texts at all, it is possible to back off to a uniform distribution whereby all words are assumed to be equally likely.

Backing off can be illustrated by considering the task of estimating the conditional trigram probabilities $P(w_k|w_{k-2}, w_{k-1})$ for word w_k in the context of the sequence of preceding words (w_{k-2}, w_{k-1}). We will assume that some appropriate discounting method has been used to assign probabilities to all observed trigrams, and these probability estimates will be denoted $\tilde{P}(w_k|w_{k-2}, w_{k-1})$. Thus, using $P_s(w_k|w_{k-2}, w_{k-1})$ to denote an estimate for the probability of word w_k given the sequence of preceding words $(w_{k-2,}\ w_{k-1})$, a backing-off scheme for obtaining this probability estimate is as follows:

$$P_s(w_k \mid w_{k-2}, w_{k-1}) = \begin{cases} \tilde{P}(w_k \mid w_{k-2}, w_{k-1}) & \text{if } C(w_{k-2}, w_{k-1}, w_k) > 0 \\ B(w_{k-2}, w_{k-1})\, P_s(w_k \mid w_{k-1}) & \text{otherwise.} \end{cases} \tag{12.11}$$

$P_s(w_k|w_{k-1})$ is the estimated bigram probability of w_k given preceding word w_{k-1}. $B(w_{k-2}, w_{k-1})$ is a normalizing constant for trigram context (w_{k-2}, w_{k-1}), and scales the bigram probabilities so that the sum of the $P_s(w_k|w_{k-2}, w_{k-1})$ terms is equal to 1 when computed over all possible words w_k. The sum of all the probabilities that are calculated by backing off must be equal to the total probability mass that has been freed from discounting the relevant trigram context. The normalizing constant is therefore chosen to be equal to the appropriate fraction of this freed probability.

Backing off from bigram to unigram and from unigram to uniform distributions can be accomplished in an analogous manner.

By setting the count threshold to zero, backing off only occurs for *N*-grams with no examples at all. However, as observing just one or two examples is unlikely to provide a reliable probability estimate, it can be beneficial to apply a higher threshold and so disregard those *N*-grams that occur just a few times in the training data. In this way only robustly estimated *N*-grams should be retained, which has the added benefit of a substantial saving in the memory needed for the language model.

12.7.6 Interpolation of language models

Backing off involves choosing between a specific and a more general distribution. An alternative is to compute a weighted average of different probability estimates, obtained for contexts that can range from very specific to very general. The idea is to improve the accuracy and robustness of a context-specific probability estimate by combining it with more general estimates for which more data are available. One option involves taking a linear combination of different probability estimates. For example, a trigram probability could be estimated by linear interpolation between relative frequencies of the relevant trigrams, bigrams and unigrams, thus:

$$P_s(w_k \mid w_{k-2}, w_{k-1}) = \lambda_3 \frac{C(w_{k-2}, w_{k-1}, w_k)}{C(w_{k-2}, w_{k-1})} + \lambda_2 \frac{C(w_{k-1}, w_k)}{C(w_{k-1})} + \lambda_1 \frac{C(w_k)}{K}, \quad (12.12)$$

where K is the number of different words, and the sum of the non-negative weights λ1+λ2+λ3=1. The values for the weights need to be chosen to make the best compromise between specificity and ability to generalize to new data. The training data are therefore divided into two parts. The first (larger) part of the data is used to derive the frequency counts, which are then used when finding the optimum values of the weights in order to maximize the probability for the second part of the data. Because the parameters that are estimated will tend to depend on how the data are partitioned, often alternative sets of parameters are estimated for several different ways of splitting the data, and then the individual estimates are combined. This smoothing method is often called **deleted interpolation**.

For simplicity, interpolation has been introduced using probability estimates obtained from simple frequency counts. However, smoothing by interpolation can also be applied to other probability estimates, including *N*-gram probabilities that have been obtained by, for example, the absolute discounting technique mentioned in Section 12.7.4. Interpolation schemes can also be used with probabilities from other types of language model, such as those discussed in Section 12.7.8 below.

12.7.7 Choice of more general distribution for smoothing

In the previous sections we have described ways of smoothing *N*-grams using lower-order *N*-grams. The lower-order distribution is usually defined in a way which is exactly analagous to the definition for the higher-order distribution that is being smoothed. There is however an alternative method which may give more reliable estimates of the lower-order probabilities. This method is best explained by describing an example. Consider a word, such as “Francisco”, that almost always occurs following just one other word (“San”). If the training text contains several examples of “San Francisco”, both the bigram probability *P*(Francisco|San) and the unigram probability *P*(Francisco) will be high. Thus if we then use a discounting method such as absolute discounting to derive the probability of “Francisco” occurring after some other bigram history, this probability will also tend to be fairly high. However, intuitively it seems more plausible that, if the only information we have implies that “Francisco” always follows “San”, the probability of “Francisco” following any other word should be low.

It is possible to define the unigram probability used for smoothing not in terms of number of examples of a word but rather in terms of the number of *different* contexts in which the word occurs. Chen and Goodman (1999) have proposed using this approach together with smoothing by interpolation, and incorporating a variant of absolute discounting that uses three separate discounts (one for *N*-grams occurring just once, one for *N*-grams occurring twice, and a third for all other *N*-gram counts). In comparative experiments, they demonstrated that this method performed consistently better (in terms of both perplexity and recognition performance) than a wide range of other smoothing techniques.

12.7.8 Improving on simple *N*-grams

N-grams have proved to be a very popular approach to language modelling. Bigrams and trigrams are the most widely used but 4-grams and even 5-grams are sometimes also included. *N*-grams provide a simple representation of language structure by focusing on local dependencies based only on word identity. Effects due to syntax, semantics and pragmatics are captured simultaneously but with no distinction between them. Although the method is really very crude, in practice it is very effective for languages such as English for which word order is important and the strongest contextual effects tend to be from immediately adjacent words.

While *N*-grams are good at modelling local context, an obvious deficiency is their inability to capture longer-term effects: syntactic constraints (e.g. subject-verb agreement) and semantic influences (certain words tending to occur together) may both operate over a span of several words. Various methods have been suggested to incorporate these effects, usually to provide additional information that can be interpolated with *N*-gram probabilities. A few of the developments are briefly mentioned in the following paragraphs.

N-grams are simple to compute directly from text data, without any need for explicit linguistic knowledge about the individual words. However, if information about syntactic classes (or other word groupings) is available, it is possible to estimate class-based *N*-gram probabilities and to back off or interpolate using these rather than needing to go to a shorter context (e.g. using a class-based trigram rather than resorting to a bigram). Taking this idea further, decision-tree methods can be applied to language models to find the best way of partitioning the data into different clusters based on questions about syntactic and/or semantic context. Other methods have involved attempting some grammatical analysis to determine the syntactic structure of the hypothesized sentence so far.

For generality, a language model needs to be trained on a large body of text from diverse sources. However, language is actually very dynamic in character, with the probability of many words being very different depending on the subject matter. One way of taking account of short-term patterns of word usage is to introduce a cache component (by analogy with “cache memory” in hardware terminology). The cache is simply a buffer which stores the frequency of occurrence of some number of the most recent words (typically around 200 different words). Word probabilities can be estimated by interpolating conventional N-gram probabilities with the probabilities as given by frequency of occurrence in the cache. A related concept is the idea of word ‘triggers’, whereby certain words tend to be very strong indicators, or triggers, for other words to occur in the general vicinity, but not necessarily within the span covered by a trigram. Triggers may be related to subject matter (e.g. “airline” associated with “flights”) or to linguistic constructs (e.g. “neither” tends to be followed fairly soon afterwards by “nor”). Trigger models are used in conjunction with a cache component to keep a record of the recent words from which to adjust the probabilities for likely ‘triggered’ words.

The use of a cache and the incorporation of triggers are two ways of capturing dynamics in language usage. It is also possible to apply language-model adaptation techniques (analogous to the acoustic-model adaptation methods described in Chapter 11) to adapt *N*-gram probabilities based on adaptation data, which might for example relate to a new topic or reflect talker-specific language patterns.

12.8 DECODING

The recognition task is to find the most probable sequence of words *W,* as given by Equation (12.1). As the vocabulary size becomes larger, the number of different possible word sequences soon becomes prohibitive and any practical system needs an efficient means of searching to find the most probable sequence. The component of the system that performs this search operation is often referred to as the **decoder**.

Although recognition using whole-word HMMs is not practicable with a large vocabulary, the same principles can be applied to units of any size. In the previous sections we have explained how statistical models can be used at different levels. One level captures the relationship between phone-size units and their acoustic realization, then there are the different phone sequences that can make up a word, and finally there are the probabilities of different word sequences. Thus a language can be modelled as a network of states representing linguistic structure at many levels. At the lowest level a small network of states represents a triphone or similar unit. At the next level a network of triphones forms a state to represent a word. A complete sentence can then be generated by a network of word states, where the connections between the states correspond to the language-model probabilities. The decoding task is to find the best path through this multiple-level network, and the recognized word sequence is given by the best path at the highest level.

In order to apply the principles of DP using the Viterbi algorithm to a multiple-level network, probabilities need to be evaluated for all valid partial paths at all levels, with no decisions being reached until all earlier parts of a path enter the same highest-level state at the same time, thus delaying decisions until all relevant evidence has been used. The use of delayed decisions is a fundamental principle of this HMM approach, as it enables the knowledge and constraints at all levels to be employed simultaneously to find the best explanation of the data. In practice, even at any one point in time, in a large-vocabulary system there are so many possibilities (both at the word and at the sub-word level) that it is impossible to evaluate all of them. However, the language model provides strong constraints that act to make many of the possibilities extremely unlikely, and it is necessary to find an efficient way of applying these constraints within the decoding.

In a first-order HMM the probability of a transition from any one state to any other state depends only on the identities of the source and destination states. This model cannot accommodate a trigram (or higher-order) language model, because here the word transition probability depends on more than just the immediately preceding state. In order to use such a language model, some way of dealing with the higher-order dependencies needs to be found. An additional complication arises when using cross-word triphones because the identity of a word-final triphone, and hence the word probability, depends on the identity of the following word.

The issues associated with large-vocabulary decoding have been addressed in various different ways. Three main types of method are briefly described below.

12.8.1 Efficient one-pass Viterbi decoding for large vocabularies

In order to accommodate cross-word triphone models, the state network for a Viterbi search needs to include multiple entries for each word to cover all possible

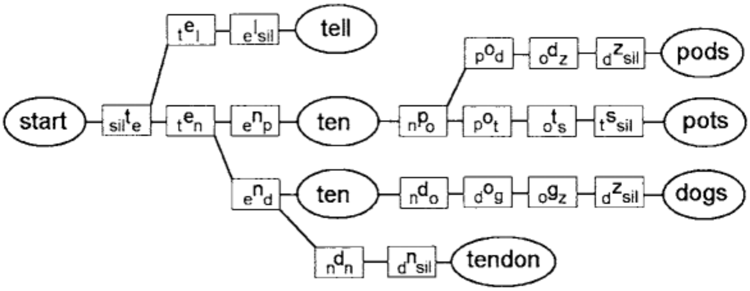


Figure 12.4 A very small fragment of a network for decoding using cross-word triphone models. This fragment shows the sequence "ten pots" and a few possible alternatives that branch at different positions in the network. Note that the word "ten" is represented by different nodes in the network, depending on whether the following word is "pots" or "dogs".

different triphones that may end any one word. Similarly, a trigram language model can be used by expanding the network to keep multiple copies of each word so that each transition between words has a unique two-word history. To make the network manageable, it is usually represented as a tree structure, as shown in Figure 12.4. In the tree, different hypotheses that start with the same sequence of sub-word models share those models. This tree network is built dynamically as required.

In Section 8.8 we introduced the concept of score pruning to reduce the number of hypotheses to be evaluated in a DP search. Efficient pruning is essential for LVCSR systems. The usual scheme employs a **beam search,** whereby at each time frame all paths whose likelihood score is not within some specified threshold of the best-scoring path are pruned out. In practice, it is generally the case that the likelihoods for all except just the few most likely states tend to be very small, so it is possible to concentrate the search on a narrow beam of possible alternatives.

Language constraints act to restrict the set of words that are likely at any given point in an utterance. It is therefore advantageous to use the language model to prune out unlikely possibilities as soon as possible when decoding an utterance, but in conventional Viterbi decoding the language-model probability is not known until the end of a word is reached. However, if a record is kept of the current possible words associated with each sub-word model, it is possible to use the language-model probabilities to prune out unlikely hypotheses at an earlier stage.

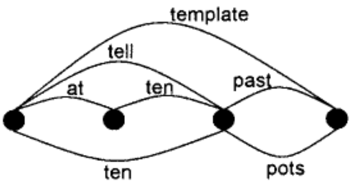
12.8.2 Multiple-pass Viterbi decoding

An alternative to attempting complete and accurate recognition in a single pass is to use a multiple-pass approach. The idea here is to start by using a simple recognition system to identify a small number of likely hypotheses, and then to use a more elaborate system to choose between these possibilities. For example, the first recognition pass could use only word-internal triphones and a bigram language model. A second recognition pass might then incorporate cross-word triphones and other more specific acoustic models as well as trigram and cache language models.

The output of the first recognition pass is usually expressed either as a rank-ordered ***N*-best** list of possible word sequences, or as a **word graph** or **lattice**

1	ten pots
2	tell pots
3	template
4	at ten past
5	ten past
6	tell past

(a) *N*-best list of hypotheses.



(b) Lattice representation of the alternatives shown in (a).

Figure 12.5 Possible alternative utterances that might be generated for a short utterance.

describing the possibilities as a network. Figure 12.5 shows an example of an *N*-best list and a word lattice that could be generated for an utterance often pots". In order to output more than one possible sequence, the first-pass DP search needs to be extended to retain several hypotheses at each stage.

12.8.3 Depth-first decoding

The Viterbi approach is known as a **breadth-first** search, because all possibilities are considered in parallel, with all paths to time *t* being evaluated before proceeding to time *t*+1. An alternative is to adopt a **depth-first** search, for which the principle is to pursue the most promising hypothesis until the end of the utterance. This strategy is often referred to as **stack decoding:** the idea is to keep an ordered 'stack' of possible hypotheses and to iteratively take the best hypothesis from the stack, choose the most likely next word and then add this hypothesis as a new entry to the stack, reordering the hypotheses as necessary. Once the end of the utterance is reached, the best-scoring hypothesis is output as the recognized word sequence. At any point in the search it may be necessary to compare hypotheses of different lengths. Because the score for any one path is a product of probabilities, it decreases with time and so a simple comparison of different scores will be biased towards shorter hypotheses. This problem can be overcome by normalizing the score for each path based on the number of frames that it accounts for.

With depth-first decoding a new hypothesis is generated each time a word is added to one of the existing hypotheses, so it is possible to evaluate the different options that are required for long-range language models. The method can, however, be expensive in terms of both memory and processing requirements.

12.9 EVALUATING LVCSR PERFORMANCE

12.9.1 Measuring errors

When recognizing connected speech, there are three types of recognition error: **substitution** (the wrong word is recognized), **deletion** (a word is omitted) and **insertion** (an extra word is recognized). The **word error rate (WER)** is given by:

$$\mathrm{WER} = 100 \times \frac{C(\text{substitutions}) + C(\text{deletions}) + C(\text{insertions})}{N}\,\%\,, \qquad (12.13)$$

where *N* is the total number of words in the test speech and *C(x)* is the count of errors of type *x*. Because there will not in general be a one-to-one correspondence between the recognized and actual word sequences, a DP alignment process is used to find the best way of aligning the two before the WER can be calculated.

The percentage **word accuracy** is equal to 100-WER. Sometimes the percentage of words correctly recognized is quoted, but this measure may not be such a good indicator of true performance as it does not include insertion errors.

12.9.2 Controlling word insertion errors

The probability that is assigned to any word sequence will depend on the relative contributions from the language and acoustic models. However, both models involve approximations and assumptions, and hence only provide estimates of the relevant probabilities. In particular, the probability given by the acoustic model will depend on the number and choice of acoustic features that are used and typically has a disproportionately large influence relative to the language-model probability. When the language model is given insufficient weight, the consequence is often a large number of errors due to insertion of many short function words. The short duration and high variability that are typical of these words mean that a sequence of their models may provide the best acoustic match to short regions of speech, even though the word sequence is given a low probability by the language model.

There are two practical solutions that are often adopted to tackle the problem of word insertion errors. One approach is to impose a **word insertion penalty,** whereby the probability of making a transition from one word to another is explicitly penalized by multiplying the word probability by some number less than 1. Alternatively or in addition, the influence of the language model on the combined probability can be increased by raising its probability to some power greater than 1. Values, both for this power and for any word insertion penalty, are usually chosen experimentally to maximize performance on some evaluation data.

12.9.3 Performance evaluations

By the mid-1980s HMMs were showing promise for practical application to large-vocabulary recognition tasks. It was, however, difficult to compare competing systems from different laboratories because individual systems tended to be trained and tested on different sets of data, due to the absence of any common databases for training or for testing. This widely recognized problem was addressed when ARPA (also known as DARPA, the *Defense* Advanced Research Projects Agency) began funding a second major research programme in the area of ASR. As part of this programme, a recognition task was defined and speech data were recorded and made publicly available. The data were divided into a **training set** for training a set of models, a **development test set** for testing a recognition system during its development, and an **evaluation test set** on which the final system could be tested and scored just once. The advent of this type of database enabled direct comparisons to be made between different algorithms and systems, with controlled assessments of performance that could also be used to measure progress over time.

In 1989 the first formal competitive evaluation to measure the performance of different systems was organized. This evaluation marked the start of a series of tests that is still continuing today. Any organization participating in an evaluation has to test its recognition system on the designated evaluation test data (following the pre-specified rules of the evaluation) and then submit the recognition results for scoring by the National Institute of Standards and Technology (NIST), a U.S. government organization. This scoring of the results by an independent body helps to ensure the applicability of direct and fair comparisons between different systems.

All the recognition tasks that have been chosen by ARPA have involved speaker-independent recognition, but over the years the tasks have progressively become more challenging both by increasing the vocabulary size and by changing the nature of the speech material. The first recognition task was called Resource Management (RM). The data for this task consisted of read queries on the status of U.S. naval resources, using a vocabulary of about 1,000 words and a fairly constrained syntax (giving a test-set perplexity of about 60). Recognition tests were first carried out on this task in 1987. Formal evaluations began in 1989 and ended in 1992. New testing data were provided for each of the evaluations, and each year the recognition performance improved as shown in Figure 12.6.

The next ARPA evaluations focused on recognition of spoken newspaper texts. This application allowed much larger-vocabulary tasks to be studied and also ensured that there was easy availability of large quantities of text data for training statistical language models. Initially the texts were all obtained from the *Wall Street*

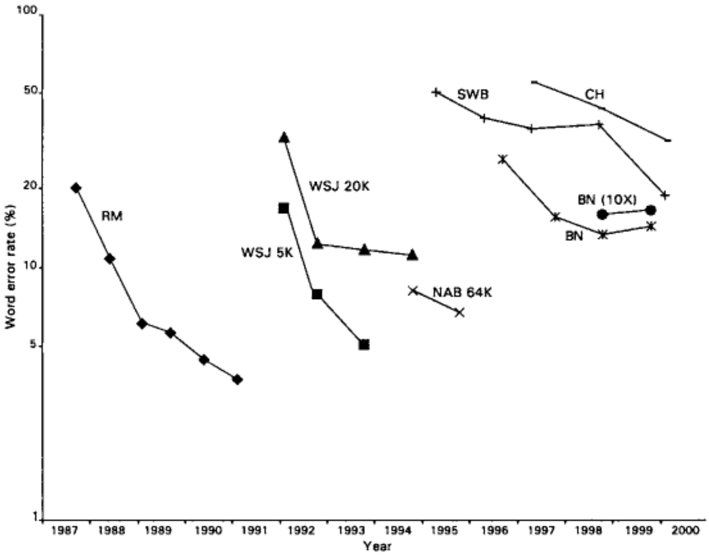


Figure 12.6 Graph showing recognition performance (in terms of word error rate, plotted on a logarithmic scale) for various transcription tasks used in ARPA evaluations. Each data point has been taken from published results for the relevant evaluation, with the aim of showing the performance of the system that was the best on that particular test. Note that, even within any one task, different evaluation data have been used each year, and in some cases there were differences in the task details. Thus, while the lines in the graph give an indication of general trends in performance, strict comparisons of performance in successive years are more difficult.

Journal (WSJ), and included both a 5,000-word recognition test and a 20,000-word test. More recently, the test set was expanded to include a range of North American business (NAB) news and restrictions on training for the acoustic and language models were removed. The recognition vocabulary became essentially unlimited and performance was generally evaluated using systems designed for recognizing about 64,000 words. WSJ and NAB evaluations were conducted in the period between 1992 and 1995. Over the years the complexity and variety of the tests increased, but recognition performance still improved each year (see Figure 12.6).

Later ARPA evaluations moved away from read speech, and concentrated on transcribing ‘found’ speech from broadcast news (BN) shows on radio and television. Such data provided a lot of variation in speech quality, microphones and background noise, as well as non-native speech and periods containing only non-speech data such as sound effects and music. For the first BN evaluation in 1996, performance dropped considerably from that which had been obtained in the previous evaluations using read speech. However, researchers soon found ways of coping with the greater variety of material, and by the final BN evaluations in 1998 and 1999 typical error rates had roughly halved from those obtained in 1996.

ARPA have also sponsored a research programme on the recognition of conversational speech, including the collection of two different corpora. Data for one corpus, known as Switchboard (SWB), first became available in 1992. The data comprised telephone conversations between two strangers about some topic that had been specified in advance. A few years later, in 1995, collection of the Callhome (CH) corpus was initiated to provide more natural conversational speech by recording telephone conversations between family members. The CH set includes a number of different languages. Conversational speech is rather different in nature to either read speech or television and radio broadcasts. Conversations tend to include many disfluencies, false starts and so on, as well as a lot of phonological reduction and special use of prosody. There is also turn-taking behaviour and reliance on shared knowledge to assist in the communication, especially when the participants know each other very well (as in the CH data). As a result, errors are much higher than for the other tasks (see Figure 12.6), but again substantial progress has been made in improving performance on the conversational recognition tasks. As of 2001, these evaluations are still continuing.

The ARPA initiative has been very influential in the development of large-vocabulary recognizers, both by providing a focus for refining the techniques to cope with increasingly difficult problems and also by making available huge quantities of speech data for training. Over the years, the result has been a progressive improvement in performance on increasingly difficult tasks, as can be seen in Figure 12.6. There are now several research systems which give impressive performance on substantial transcription tasks. However, to perform at their best these systems generally require both a lot of memory (with typically several million parameters in both the acoustic and language models), and a lot of processing power (often operating at a few hundred times real time). In the last two BN evaluations, the effect of processing demands was tested by introducing a category for systems that operated at no slower than 10 times real time (shown as BN(10×) in Figure 12.6). Enforcing this constraint led to around 15% more errors than if there were no processing restrictions. We will consider performance in relation to the requirements of different applications in Chapter 15.

12.10 SPEECH UNDERSTANDING

In order to understand spoken language, it is necessary not only to recognize the words, but also to determine the linguistic structure, including both syntactic and semantic aspects. Given a word sequence, the field of computational linguistics provides techniques for deriving a representation of the linguistic structure. Traditionally these techniques involve a complex linguistic analysis based on qualitative models that are usually stated in the form of a grammar that is specified manually by human experts, although in recent years data-driven statistical methods have also been used. Even so, any detailed linguistic analysis will involve modelling long-range effects that may operate over the entire span of an utterance. It is very difficult to incorporate such a model into a speech recognition search.

Due to the difficulties involved in attempting to incorporate detailed linguistic analysis into the recognition search, speech understanding systems generally treat these tasks as separate components: an ASR system of the type described earlier in this chapter, and a second component that uses artificial-intelligence techniques to perform linguistic analysis in order to 'understand' the recognizer output. To reduce the problems caused by errors made at the recognition stage, the output of the recognizer should include alternatives, which can be provided in the form of a word lattice or an *N*-best list. The final recognized output is typically taken to be the highest-scoring alternative that is also allowed by the linguistic model.

The process of determining the linguistic structure of a sentence is known as **parsing**. Syntactic structure is usually expressed in terms of a formal grammar, and there are a variety of grammar formalisms and associated methods for **syntactic parsing**. However, traditional parsers aim to recover complete, exact parses. This goal will often not be achievable for spoken language, which tends to contain grammatical errors as well as hesitations, false starts and so on. The problem is made even worse when dealing with the output of a speech recognizer, which may misrecognize short function words even when overall recognition performance is very good. It can therefore be useful to adopt **partial parsing** techniques, whereby only segments of a complete word sequence are parsed (for example, noun phrases might be identified). Other useful tools include **part-of-speech taggers,** which aim to disambiguate parts of speech (e.g. the word "green" acts as an adjective in the phrase "green coat" but as a noun in "village green"), but without performing a parsing operation. Some taggers are rule-based, but there are also some very successful taggers that are based on HMMs, with the HMM states representing tags (or sequences of tags). Transition probabilities are probabilities of tag(s) given previous tag(s) and emission probabilities are probabilities of words given tags. Partial parsing and part-of-speech tagging enable useful linguistic information to be extracted from spoken input without requiring a comprehensive linguistic analysis.

General **semantic analysis** for the derivation of meaning representations is a complex aspect of computational linguistics. However, many speech-understanding systems have used a simpler approach that is more application dependent. A popular technique uses templates, or 'frames', where each frame is associated with an action to be taken by the system. A frame has 'slots' for different items of relevant information. In a flight enquiry system for example, one frame could represent a request for flight information. This frame could include slots for the type of information (e.g. flights, fares) and for any constraints (e.g. date, price). The task is to find the frame that gives the best match to the input, and to obtain values for the slots in that frame. Once a frame has been recognized and its slots have been filled, the system's response can be determined.

Many applications of speech understanding involve the user asking questions in order to access information from a computer system. Often a query may be insufficiently specified or ambiguous. However, by entering into an interactive dialogue with the user, the system should be able to obtain the additional information or resolve any ambiguities. A **dialogue manager** can greatly assist in the usability of these **spoken dialogue systems,** by keeping track of the interaction, guiding the user and asking questions where necessary.

12.10.1 Measuring and evaluating speech understanding performance

When evaluating speech understanding systems, word and sentence recognition errors are obviously of interest. However, some measure of a system's ability to recognize meaning is also required. This measure is usually the percentage of utterances for which the complete system gives an acceptable output, as judged by a trained human operator. For a spoken dialogue system that is intended for a particular application, it may be most informative to evaluate performance using task-based measures such as the percentage of tasks successfully completed and time taken to accomplish the tasks, together with measures of user satisfaction.

In parallel with its speech transcription evaluations, ARPA has also been conducting a series of evaluations of spoken language understanding systems. In the late 1980s, work began on an Air Travel Information System (ATIS) task, and formal evaluations were carried out in the period between 1990 and 1994. The ATIS task involved recognizing and responding to spontaneous queries about airline reservations (e.g. "List the flights from Boston to Dallas." or "Is lunch served on that flight?"), using a vocabulary of about 2,500 words. In 1994, the best speech recognition performance was a word error rate of 2.3% (Moore *et* al., 1995), which improved to 1.9% when the output of the speech recognizer was subjected to post-processing by a natural-language system that was tuned to the application domain. Although this task involves spontaneous speech, the constraints of the task domain enable high recognition accuracy to be achieved. The best understanding performance, taken over all the answerable spoken queries, was an error rate of 8.6% (Pallett *et* al., 1995). Although there were more understanding errors than word recognition errors, this level of performance is probably adequate for many database query applications. However, achieving such performance requires the system to be tuned to its particular application domain, with a lot of 'knowledge' that is specific to the one domain (flights in the case of the ATIS task).

More recent ARPA evaluations have focused on other types of tasks, which involve processing large quantities of speech material in a way that requires some understanding capability. Past and ongoing tasks include retrieval of spoken documents, detection and tracking of topics in broadcasts, and extraction of the content (meaning) from transcripts of spoken news material. Automatic speech understanding is a pre-requisite for all of the most advanced applications of spoken language processing, of which one of the hardest is automatic speech translation (see Chapters 15 and 16 for further discussion).

CHAPTER 12 SUMMARY

- Some large-vocabulary recognition tasks may require accurate transcription of the words that have been said, while others will need understanding of the semantic content (but not necessarily accurate recognition of every word).
- For speech transcription the task is to find the most likely sequence of words, where the probability for any one sequence is given by the product of acoustic-model and language-model probabilities.
- The principles of continuous speech recognition using HMMs can be applied to large vocabularies, but with special techniques to deal with the large number of different words that need to be recognized.
- It is not practical or useful to train a separate model for every word, and instead sub-word models are used. Typically phone-size units are chosen, with the pronunciation of each word being provided by a dictionary.
- Triphone models represent each phone in the context of its left and right neighbours. The large number of possible triphones is such that many will not occur in any given set of training data. Probabilities for these triphones can be estimated by 'backing off or interpolating with biphones (dependent on only the left or the right context) or even context-independent monophones.
- Another option, which allows greater context specificity to be achieved, is to group ('cluster') similar triphones together and share ('tie') their parameters. A phonetic decision tree can be used to find the best way to cluster the triphones based on questions about phonetic context. The idea is to optimize the fit to the data while also having sufficient data available to train each tied state.
- An embedded training procedure is used, typically starting by estimating parameters for very general monophone models for which a lot of data are available. These models are used to initialize triphone models. The triphones are trained and similar states are then tied together. Multiple-component mixture distributions are introduced at the final stage.
- The purpose of the language model is to incorporate language constraints, expressed as probabilities for different word sequences. The perplexity, or average branching factor, provides a measure of how good the language model is at predicting the next word given the words that have been seen so far.
- *N*-grams model the probability of a word depending on just the immediately preceding *N*-1 words, where typically *N*=2 ('bigrams') or *N*=3 ('trigrams').
- The large number of different possible words is such that data sparsity is a massive problem for language modelling, and special techniques are needed to estimate probabilities for *N*-grams that do not occur in the training data.
- Probabilities for *N*-grams that occur in the training text can be estimated from frequency counts, but some probability must be 'freed' and made available for those *N*-grams that do not occur. Probabilities for these unseen *N*-grams can then be estimated by backing off or interpolating with more general models.
- The principles of HMM recognition extend to large vocabularies, with a multiple-level structure in which phones are represented as networks of states, words as networks of phones, and sentences as networks of words. In practice the decoding task is not straightforward due to the very large size of the search space, especially if cross-word triphones are used. Special treatment is also required for language models

whose probabilities depend on more than the immediately preceding word (i.e. for models more complex than bigrams). The one-pass Viterbi search can be extended to operate with cross-word triphones and with trigram language models, but the search space becomes very large and is usually organized as a tree. Efficient pruning is essential.

- An alternative search strategy uses multiple passes. The first pass identifies a restricted set of possibilities, which are typically organized as an N-best list, a word lattice or a word graph. Later passes select between these possibilities. Another option is to use a depth-first search.
- Automatic speech understanding needs further processing of the speech recognizer output to analyse the meaning, which may involve syntactic and semantic analysis. To reduce the impact of recognition errors, it is usual to start with an N-best list or word lattice. Partial parsing techniques can be used for syntactic analysis to deal with the fact that the spoken input may be impossible to parse completely because parts do not fit the grammar, due to grammatical errors, hesitations and so on.
- Meaning is often represented using templates, which will be specific to the application and have ‘slots’ that are filled by means of a linguistic analysis.
- In spoken dialogue systems, a dialogue manager is used to control the interaction with the user and ensure that all necessary information is obtained.
- ARPA has been influential in promoting progress in large vocabulary recognition and understanding, by sponsoring the collection of large databases and running series of competitive evaluations. Error rates of less than 10% have been achieved for transcribing unlimited-vocabulary read speech and for understanding spoken dialogue queries. Recognition of more casually spoken spontaneous speech is still problematic.

CHAPTER 12 EXERCISES

E12.1 Explain the different requirements and problems in speech transcription and speech understanding.

E12.2 What are the special issues for the design of the acoustic model, the language model and the search component when a recognizer needs to cope with a large vocabulary size?

E12.3 Give examples of how acoustic/phonetic knowledge can be used when choosing an acoustic model set for a large-vocabulary recognition system.

E12.4 When estimating trigram language-model probabilities based on counts in some training text, explain how probabilities can be estimated for trigrams that do not occur in the training data. How does this operation affect the probabilities that need to be assigned to trigrams that do occur?

E12.5 What are the similarities and differences between the concept of ‘backing off in language modelling and in acoustic modelling?

E12.6 Explain the relative merits of performing large-vocabulary recognition using a one-pass Viterbi search versus using a multiple-pass search strategy.

E12.7 What measures are required to evaluate the performance of a speech understanding system?

CHAPTER 13

Neural Networks for Speech Recognition

13.1 INTRODUCTION

Humans can recognize and understand what people are saying with remarkable accuracy, even when the acoustic signal is severely degraded. A prominent theme for this book is the importance of emulating human performance. An obvious question concerns whether such emulation can be achieved by actually modelling the type of mechanism in the human central nervous system. **Artificial neural networks (ANNs)** have long been a subject of research interest for many pattern recognition tasks, including speech recognition. A period of particular interest in ANNs began in the mid-1980s and considerable advances have been made since that time, but these methods have not yet achieved success as a complete system for the recognition of continuous speech. However, ANNs have been shown to have practical advantages for performing certain tasks as a component of a system operating within the HMM framework. Neural networks now represent a large subject area in their own right, and we do not have space to do more than touch on the subject here. In this chapter we will give a brief introduction to typical neural network structures and their application to speech recognition tasks.

13.2 THE HUMAN BRAIN

Although there is as yet very inadequate knowledge about the functioning of the brain, there is now quite a lot of knowledge about its most basic components—the individual nerve cells. The nerve cell or **neuron** is an electro-chemical device with one long output fibre, the **axon,** and many shorter input fibres, the **dendrites**. When it receives sufficient stimulation from its dendrites, the neuron will 'fire', causing a small voltage pulse to travel along its axon by a progressive electro-chemical reaction. The speed of travel of a pulse along an axon varies between different axons but is typically of the order of 100 m/s for the faster ones, and is thus many orders of magnitude below the speed of electric pulses in wires. After a neuron has fired it will become inactive for a period of a millisecond or so, during which it is not sensitive to further stimulation. The magnitude and form of the response pulse that is transmitted along the axon is independent of the size of the stimulus that caused it. The only possible variable in a neuron's output is the time of firing, and the short-term-average rate of firing is a useful measure of activity in a neuron. The quiescent period after each firing imposes a maximum firing rate, but after each firing the threshold of stimulation needed to fire it again gradually reduces.

The axons pass in proximity to the input dendrites of other neurons, and the coupling from a pulse in the axon from one neuron will stimulate the tendency to fire in the others.

These neural junctions are known as **synapses**. The strengths of couplings at synapses vary a lot, both from one synapse to another and over time as a result of brain development and learning.

Because of the involvement of electro-chemical processes, there is a large disparity (in excess of 10,000:1) between the maximum speed of operation of nerve cells and that of modern electronic circuits. However, the neural system's speed disadvantage is compensated for by having enormous numbers of neurons operating in parallel. It is estimated that the number of neurons in the human brain is at least 10^{10}, and the number of effective synapses is at least 1,000 times greater.

Although the above facts do not explain how the brain performs its cognitive functions, they do give some indication of the possible processes involved. First, the individual neurons are extremely limited in what they can do. If stimulated enough they will fire, and if not they make no response. There are two possible mechanisms for memory in the brain. Long-term memory can exist only in the nature of the synaptic couplings between neurons. These couplings are partly innate, but other synapses develop over time, and changes in the strengths of synaptic couplings seem to provide the only feasible mechanism for long-term memory of learnt behaviour. Short-term memory could be achieved by continually re-circulating data, whereby one group of neurons stimulates the firing of another group, which in turn stimulates the first group again after the transmission delay through their axons.

A very obvious way in which operation of the brain differs from the operation of normal computers is the degree of parallel processing. Although modern high-power computing systems tend to include parallel operation, the method is merely to divide the task into parts which can be largely performed separately, and to process data serially in each part. In the brain, by contrast, all parts are working together, with a continuous high degree of inter-communication between them.

13.3 CONNECTIONIST MODELS

The nature and properties of the brain have inspired many research groups to investigate whether cognitive processes could be achieved in electronic or computational models that have many of the known neural properties. Although the term "artificial neural network" is widely used, for most groups working in this area the aim has not been to copy the precise action of the neural mechanism, but rather to generate functional models of cognitive processes, using knowledge of neurophysiology as a guide to what types of operation might be plausible.

An obvious requirement for an ANN is to include large numbers of highly interconnected units, whose coupling weights can be modified by 'learning'. Another important feature is to make the response of the units a non-linear function of the combined input stimulation, by analogy with the firing threshold of neurons.

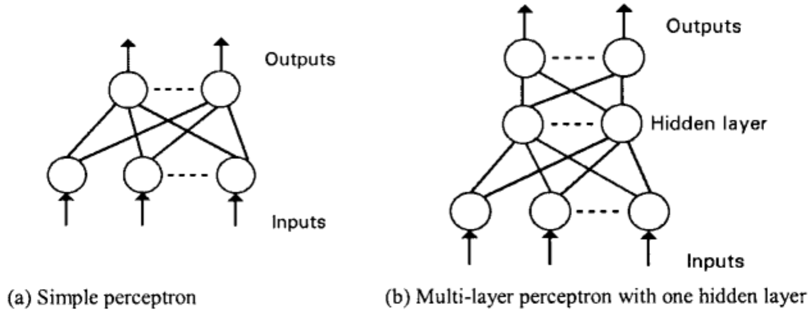
The two most significant characteristics of neural processes are that the units are working in parallel, and that each operation is distributed between many such units. Hence this arrangement accomplishes **parallel distributed processing (PDP)**. Acknowledging the vital role of the numerous connection weights between units, models of cognitive processes of this type are sometimes known as **connectionist models**.

13.4 PROPERTIES OF ANNS

An ANN needs input units to allow information to be put into the system. In the case of speech processing, the input could be the responses of auditory sensors. There must also be output units, through which the response of the system is made externally available. Any connection between two units has an associated weight to determine the contribution of each input, and the response of a unit is governed by a non-linear function (by analogy with the firing threshold of neurons). For any practical system it is necessary to have a means for the network to learn its required behaviour by adjusting the weights of the inter-connections. Training a network involves supplying example patterns to the input units, together with the desired output patterns. A learning algorithm is used to modify the connection weights in a direction that makes the model give a closer approximation to the desired output.

One early ANN was the **perceptron** (Rosenblatt, 1962), for which the input units are connected directly to the output units. Each output unit computes a single output as a function of one or more inputs, as shown in Figure 13.1 (a). There is thus just one layer (the output layer) that performs any computation, as the only function of the input layer is to store the values that are input to the system. There was much early enthusiasm for the capabilities of these perceptions. However, although such a network can perform linear classification, it was shown by Minsky and Papert (1969) that a single-layer topological structure puts a serious limitation on the types of computation that can be performed. An essential feature of the more advanced present-day ANN systems is that they require **hidden** units, not connected to input or output. Incidentally, it is known that a high proportion of brain cells are not connected directly to either input or output nerves.

A popular ANN architecture is a structure that is similar to Rosenblatt's perceptron, except that it includes one or more layers of hidden units as shown in Figure 13.1(b). The addition of the hidden layers enables this **multi-layer perceptron (MLP)** to learn arbitrarily complex decision boundaries between different classes. There are straightforward training procedures for MLPs. An error function (such as the mean-squared error) is defined, and this function is differentiated with respect to each of the network weights. These derivatives can then be used to find values of the weights that minimize the error function. By propagating the errors back through the network, it is possible to optimise the weights for any number of layers in the network. This type of training procedure is therefore known as error **back-propagation**.



(a) Simple perceptron

(b) Multi-layer perceptron with one hidden layer

Figure 13.1 Examples of a simple perception (no hidden layers) and multi-layer perceptron.

MLPs have been shown to be very effective for a wide range of classification problems. They can be used to classify sequences of speech frames representing individual words (or other linguistic units) by presenting the entire sequence as a pattern for input to the MLP. However, because MLPs are by nature static networks, they cannot capture temporal properties of time-varying signals directly.

MLPs are feed-forward networks: connections are all in one direction from input to output. Other types of ANN include **recurrent** networks that also allow feedback connections from the output of one layer back to the previous layer and between nodes in any one layer. A consequence of the feedback connections is that activity can remain in the network after removal of an input stimulus, and hence the response of the network to any one input depends on previous inputs. This structure is therefore attractive for representing a time-varying signal such as speech, but it is more difficult to train than a structure that only has feed-forward connections. There are also structures that are intermediate between MLPs and fully recurrent networks. These structures include partially recurrent networks, which have mainly feed-forward units but include some specific feedback connections, and **time-delay neural networks (TDNNs),** which have only feed-forward connections but include a specified set of previous activation values as part of the input.

13.5 ANNS FOR SPEECH RECOGNITION

In comparison with the HMMs discussed in the previous chapters, ANN methods have the advantage that they can learn much more general types of structure, which can implement very complex non-linear conditional rules. By varying connection weights they can also modify whatever structure they are provided with initially. In principle, therefore, these systems have the power to accomplish most, if not all, of the operations involved in recognizing and interpreting speech. However, such a system would require huge numbers of units and be very difficult to train. It seems almost certain that a large part of the human ability in linguistic processing arises due to the innate neural connections in our brains, and actual linguistic competence then follows as a result of many years of training during which children are almost continually using language for everyday communication. Acquiring both the innate and learnt connectivity patterns seems to be a task that will not be solved in machines for many years. A practical problem in developing ANN systems is that, because all the modelling is hidden within the network, it tends to be very difficult to understand what is actually happening in that network and thus to gain insights into *how* a speech recognition task is being performed.

So far, the most successful uses of neural network models have been for parts of the speech-recognition problem, within frameworks that allow some knowledge to be built into the system. Attractive properties of ANNs include the following:

1. The learning algorithms for ANNs are inherently discriminative between the different classes that the network is trained to recognize. In fact it has been shown that the output of various ANNs can be interpreted as direct estimates of the *a posteriori* probabilities of output classes given the input. Thus neural networks have the desirable property discussed in Section 11.5 that they are trained to maximize discriminability between the correct output class and the alternative incorrect classes, and hence to minimize classification error.

2. There is no need to make any strong assumptions about the statistical distributions of the features that are input to the network. Also, because ANNs can incorporate multiple constraints for performing classification, it is not necessary to assume that features are independent.
3. By including several frames in the input or using feedback connections, the network will inherently model context dependence across speech frames.
4. Although in practice most present ANN systems are simulated on conventional von Neumann computer architecture, the parallel and regular nature of ANN structures makes them amenable to easy hardware implementation.

For recognition of isolated words, and for phoneme classification of speech that has been pre-segmented to identify phone boundaries, ANNs have been shown to give performance that is at least comparable with and sometimes better than that obtained with HMM systems. ANNs have also proved useful for front-end processing tasks, including noise reduction in corrupted speech signals. However, ANNs on their own have not yet been shown to be effective for any substantial continuous speech recognition tasks. In order to train a connectionist network it is necessary to specify the correspondence between input (the feature vectors) and output (the classes to be recognized). This information tends to be unavailable for continuous speech because the locations of word boundaries are not usually known in advance. More generally, although recurrent architectures and TDNNs capture the time-varying nature of speech to some extent, connectionist models do not cope with timescale variations very easily. In contrast, one of the strengths of HMMs is their separation of spectral and temporal properties, and hence their efficiency as tools for jointly recognizing and segmenting continuous speech.

13.5.1 Hybrid HMM/ANN methods

The motivation for the development of **hybrid** models for continuous speech recognition is to combine the discriminative classification abilities of ANNs with the time-domain modelling capabilities of HMMs. The usual approach involves training a neural network to compute emission probabilities in an HMM system.

In an HMM/ANN hybrid system, the HMM structure determines possible paths through the models and the Viterbi algorithm can be used to find the best path in the usual way. The crucial difference from a conventional HMM system is that the neural network is used to compute the emission probability for any given correspondence between an HMM state and an observed acoustic feature vector. The HMM transition probabilities are used as normal, but the ANN replaces the emission p.d.f.s that would be used in a conventional HMM system. To obtain an emission probability for use in recognition, the *a posteriori* probability that is the output of the ANN needs to be converted back to a likelihood. Given an observed feature vector $\boldsymbol{y}$ and a model state representing some class c that might be recognized, the ANN provides an estimate of $\mathrm{P}(c|\boldsymbol{y})$. The HMM emission probability corresponds to $\mathrm{P}(\boldsymbol{y}|c)$, the probability of the observed vector given the class (i.e. the model state). Applying Bayes' rule we have:

$$P(\boldsymbol{y} \mid c) = \frac{P(c \mid \boldsymbol{y}) P(\boldsymbol{y})}{P(c)} . \tag{13.1}$$

*P(**y**)* is independent of the class *c*. Dividing the ANN output by the prior probability *P(c)* therefore gives a scaled likelihood that can be used as an HMM emission probability. The prior probability of each class can be estimated from its frequency in the training data. In hybrid systems for large-vocabulary recognition, it is usual for each state to represent one phone, so the priors are phone probabilities.

To train the system, a Viterbi training procedure is used to iteratively segment the training data, then train the neural network and re-estimate the HMM transition probabilities. The accuracy of the segmentation at the first iteration does not appear to be too critical, and can for example be provided by a simple set of HMMs.

Several workers have developed hybrid systems combining HMMs with neural networks, typically MLPs or recurrent nets. In recent years, these hybrid systems have been competitive with conventional HMM systems on unlimited-vocabulary continuous speech recognition tasks of the type described in the previous chapter. While the hybrid systems have not yet been shown to clearly outperform the best HMM systems, they can achieve good performance with far fewer parameters than are typically required in conventional HMM systems. Hybrid systems have so far tended to use the same acoustic features, model structure and so on that have been optimized for use with pure HMM systems. Further research may identify ways of deriving greater benefit from the neural network capabilities.

CHAPTER 13 SUMMARY

- Artificial neural networks (ANNs) are loosely modelled on characteristics of the human neural system. A network of interconnected units relates some input (e.g. acoustic feature vectors) to the system output (e.g. recognized words). Each connection between units has a weight whose value is trained from data.
- A popular architecture is the multi-layer perception (MLP), which includes one or more ‘hidden’ layers of units between the input and output connections. MLPs have proved to be effective for many classification problems, but are static networks and therefore cannot capture time-varying properties directly.
- There are other ANN structures that can capture temporal properties more explicitly, but these structures tend to be more difficult to train.
- ANNs are by nature discriminative, and can learn complex non-linear relationships between data and recognition classes. They have been successful for isolated-word recognition, but so far seem to be less suited to segmenting and recognizing continuous speech.
- ANNs have however proved to be successful for computing HMM emission probabilities within a hybrid system.

CHAPTER 13 EXERCISES

E13.1 Why are ANNs attractive for speech recognition tasks, and what are the practical difficulties that must be addressed?

E13.2 What are the main differences between discriminative training using ANNs as opposed to purely HMM-based methods such as MMI training?

CHAPTER 14

Recognition of Speaker Characteristics

14.1 CHARACTERISTICS OF SPEAKERS

When humans hear speech they usually recognize what the words are, and at the same time they also recognize characteristics of the talker who is speaking those words. If they know the person who is speaking, they can normally recognize that person. Even if they do not know the person, they can recognize attributes such as the sex of the speaker, and often recognize the language in which the person is speaking and the person's accent. Other attributes such as the speaker's emotional state can usually be determined from the person's speech. Analyses of speaker characteristics can be performed automatically, using techniques which are closely related to many of the methods that have been applied in ASR. These analyses may be used as one component of a speech recognition system; for example, some large-vocabulary recognition systems employ **gender identification** (identification of the sex of the talker) prior to speech recognition using gender-dependent models. Systems for recognizing speaker characteristics can also be used alone for applications in their own right. Typical applications include security, surveillance and forensic work. The technologies that have received most interest are automatic **speaker recognition** (recognition of speaker identity) and automatic **language recognition**. In this chapter we will introduce some general principles before briefly describing methods used for automatic speaker and language recognition.

14.2 VERIFICATION VERSUS IDENTIFICATION

Tasks requiring recognition of speaker attributes can be divided into two types:

1. *Verification:* The task is to verify whether or not an utterance belongs to a specified category, and the utterance is then accepted or rejected accordingly. Verification can be viewed as a signal detection task, as the requirement is to detect whether or not a required category is present in a given speech signal.
2. *Identification:* The task is to identify which one out of a set of possible categories is present. The categories may be a **closed set,** whereby it is known that the utterance must belong to one of the given categories and the task is simply to decide which one. A harder identification task involves an **open set** of categories, for which there is also the possibility that the utterance may not fit into any of the given set and should therefore be rejected.

The input utterance is first analysed to give a sequence of feature vectors. This stage is similar to feature analysis for ASR, although the choice of features may be somewhat different, depending on the nature of the task. A comparison is then made with one or more reference templates (evaluating distances) or statistical models (evaluating probabilities), and a recognition decision is made. The nature of this decision depends on whether the task involves verification or identification.

Taking the example of speaker recognition, **speaker verification** involves deciding whether or not a given voice sample was spoken by one known individual, based on how well this sample matches the reference for the voice of the one speaker. If the match is good enough, the utterance is accepted. **Speaker identification** (within a closed set) entails deciding on the speaker identity from a set of 'known' speakers, by finding the one from the reference set that gives the closest match to an 'unknown' voice sample. Open-set speaker identification requires both types of decision to be made, so as to make a choice between speakers or to reject the sample if it does not match any of the known speakers well enough. Similar distinctions apply in language recognition, although most research has concentrated on (closed-set) **language identification**.

In either identification or verification tasks, it is possible to make an estimate of confidence in the recognition decision. For certain applications (such as speaker verification for secure access), further input can be requested when a confident decision cannot be made based only on the original input utterance. Performance tends to improve as the amount of speech material increases, and hence more information becomes available, until some maximum performance level is reached.

14.2.1 Assessing performance

For closed-set identification tasks, performance can be assessed by testing on a suitable number of samples of speech from each of the categories of interest and calculating the percentage of samples that are correctly recognized. For *N* categories, just guessing would give 100/N% correct. As N increases, the guessing level of performance gets worse and in general the task becomes more difficult.

For verification (and open-set identification), assessing performance is more complicated because the input speech may be of *any* category and the decision is one of acceptance versus rejection. Any evaluation must include samples of speech both from the category of interest and from a range of alternative categories. In the case of a speaker verification system for example, it will be necessary to include speech both from the **target** speaker and from several likely **imposter** speakers. The decision to accept or reject a speech sample is usually made by deciding whether the degree of fit to the required category exceeds a **threshold**. One way of specifying the threshold would be simply in terms of how good the match has to be. However, choosing a value for this threshold can be very difficult because the goodness of the match between the reference and a valid utterance may vary considerably from utterance to utterance, especially if there are changes in the environment or channel characteristics. What is really required is not how good the match is in absolute terms, but how good it is for the one category of interest relative to the match for the full range of possible different categories. A solution is therefore to normalize the score before making the threshold comparison. For example, in a speaker verification system based on a probability measure, the normalization term could be the sum of probabilities for a suitable set of speaker-dependent models, or it could be the probability for one speaker-independent model intended to characterize the general population. Thus for any speech sample the match for the speaker of interest is always evaluated relative to some estimate of the match for the whole population who may have spoken the utterance.

14.2.2 Measures of verification performance

When designing a verification system, the aim is to choose a value for the acceptance threshold that minimizes the number of verification errors. However, verification errors can be one of two types: **false acceptances** (incorrectly accepting an imposter in the case of speaker verification) and **false rejections** (incorrectly rejecting the target speaker). In more general signal-detection terminology, a false acceptance error is referred to as a **false alarm,** and a false rejection error as a **miss.** The value of the threshold will affect the relative number of occurrences of the two types of error. Here we will assume that two patterns are being compared on the basis of their *similarity,* so a high score indicates a good match. (Alternatively we could refer to the *distance* between the patterns, in which case a low score would be required.)

Figure 14.1 shows a typical plot for the probabilities of the two types of verification error against the value of the threshold specifying the degree of match required for acceptance. With a high threshold, the match has to be very good for an utterance to be accepted, and hence there will be fewer false acceptances but at the expense of more false rejections. Conversely, a low threshold allows a poorer match to be accepted, which reduces the number of false rejections but leads to more false acceptances. A measure of performance that is sometimes quoted for verification tasks is the **equal error rate (EER),** which is the error rate obtained when the threshold parameter is set so that the two types of error occur with equal probability. In a graph such as the one shown Figure 14.1, the EER is given by the error rate at the point at which the two error curves cross each other.

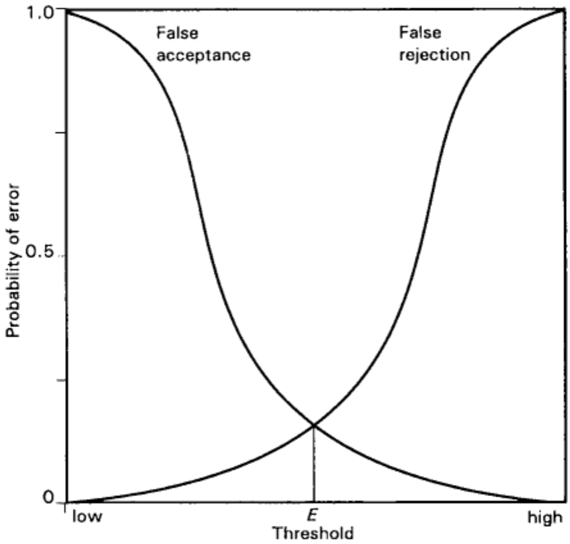


Figure 14.1 Typical plots for a verification system showing relationship between the probabilities of false-acceptance and false-rejection errors as a function of the similarity threshold for acceptance. The point at which the two curves cross gives the equal error rate (EER) for the system. The value of the threshold corresponding to the EER is marked *E.*

In real applications the preferred setting for the threshold parameter will depend on the cost associated with the two types of error, which will be different for different applications. For example, in a speaker verification system for secure access, a false acceptance would usually be more costly than a false rejection, and hence the similarity threshold should be set to a relatively high value. In a surveillance application on the other hand, false rejections would normally be more costly and thus it is appropriate to set the threshold to a fairly low value.

An error measure that captures the performance of the verification technology when used in a particular application is the **detection cost,** which is a weighted arithmetic mean of the probabilities of false rejections and false acceptances. The detection cost can be defined as follows:

$$\text{detection cost} = c_{FR} \,.\, P_{FR} \,.\, P_{\text{target}} + c_{FA} \,.\, P_{FA} \,.\, (1 - P_{\text{target}}),$$

where c_{FR} and c_{FA} are 'costs' associated with false rejections and false acceptances respectively. These costs can be set dependent upon the application. P_{FR} and P_{FA} represent the probabilities of the two types of error, which are obtained from the number of occurrences of each of these errors divided by the total number of trials, P_{target} is the *a priori* probability of a target: in a speaker verification system for example, this probability represents the proportion of trials made by the true speaker. The values of P_{FR} and P_{FA}, and hence the detection cost, will depend on the value of the acceptance threshold. This threshold value can thus be chosen to minimize the detection cost. The chosen threshold is sometimes referred to as the **operating point.**

The performance of verification systems is often shown graphically as a **receiver operating characteristic** (**ROC**) curve. ROC analysis originated from psychophysics, and is now applied in various fields to characterize accept/reject decision-making performance by both humans and machines. In a conventional ROC curve it is most usual to plot the probability of a correct acceptance against the probability of a false acceptance. By varying the acceptance threshold, a set of points will be obtained which define a curve from the bottom-left corner to the top-right corner of a square, as shown in Figure 14.2. The better the system, the nearer the curve to the top left of the square (i.e. to a probability of 1.0 for correct acceptances and 0.0 for false acceptances). The ROC curve thus provides the means for a graphical comparison between the performance of different systems.

Figure 14.2 shows a typical format that has traditionally been used for ROC curves. There are, however, various different ways in which the same information may be displayed. For example, the y-axis may be used to show the probability of a false rejection (where *P*(false rejection)=1-*P*(correct acceptance)), which can be useful for directly illustrating the trade-off between the two types of error. Thus, in signal-detection terminology, the probability of a miss is plotted against the probability of a false alarm. It may also be helpful to use a logarithmic or other non-linear scale for the error rates. For example, Figure 14.3 illustrates the performance of the same verification systems shown in the traditional ROC curves of Figure 14.2, but here performance is expressed in terms of detection errors plotted on a logarithmic scale. It can be seen that the effect of this non-linear errorrate scale is to magnify the regions in which the error rates are low, and to make the 'curves' become nearer to straight lines. Thus it is easier to see differences between systems that are all performing fairly well.

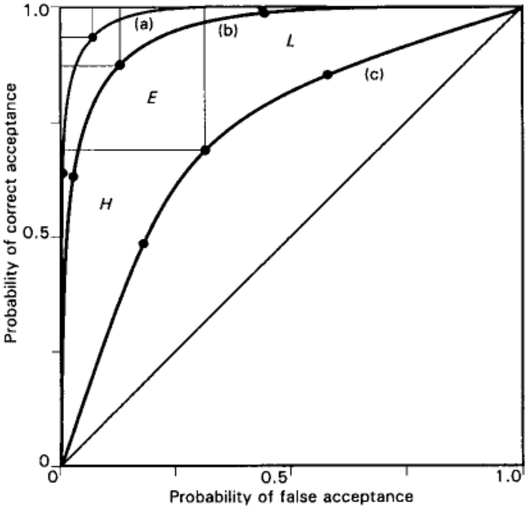


Figure 14.2 Three ROC curves. Curve (a) shows a system with better verification performance than curve (b), which is in turn much better than (c). The diagonal line shows the performance that would be obtained by guessing (i.e. equal probabilities for correct and false acceptances). Points marked *E* indicate the performance at the equal-error-rate setting for the acceptance threshold. Points *H* correspond to performance for a high threshold of similarity (strict acceptance criterion), while points *L* show performance for a low threshold (lax acceptance criterion).

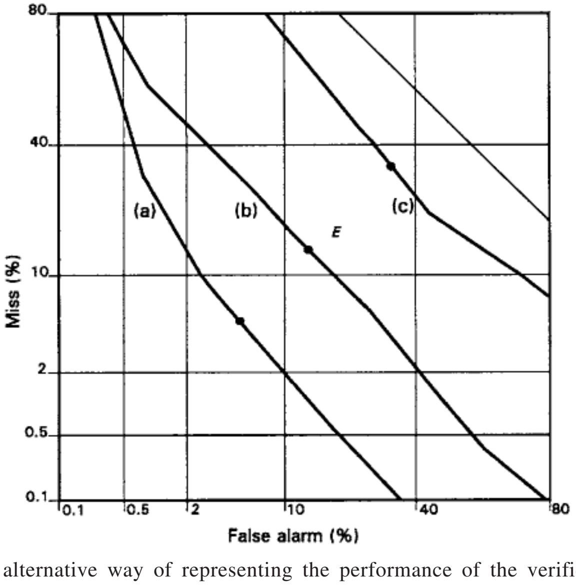


Figure 14.3 An alternative way of representing the performance of the verification systems shown in Figure 14.2. In these alternative plots, misses are plotted against false alarms, with both types of errors plotted on a logarithmic scale. In the extreme regions of few false alarms or few misses, the effect of the non-linear scale is to show a clearer separation between the three systems than is seen in Figure 14.2. Points marked *E* indicate the performance at the equal-error-rate setting for the acceptance threshold, and the diagonal line furthest to the right shows the performance that would be obtained by guessing.

14.3 SPEAKER RECOGNITION

14.3.1 Text dependence

In speaker recognition it is usual to distinguish between **text-dependent** methods and **text-independent** methods, for which the main distinction is as follows:

1. *Text-dependent* The text to be spoken by the user is 'known' by the system.
2. *Text-independent* There are no constraints on the text when the system is in use, so it must be trained to be able to cope with utterances of any text.

The feasibility of using text-dependent methods will depend on the type of application, and especially on whether or not the users can be regarded as being **'co-operative'**. For example, text-dependent methods are appropriate for systems used in security applications, where speakers are expected to be co-operative and to know their own passwords. For surveillance and forensic applications, however, the speakers will most often not be co-operative (and may be ignorant of any surveillance), so it is not possible to use predetermined keywords and text-independent methods are required.

For security applications, it is important to guard against attack by someone playing back a suitable voice recording of an authorized speaker. Text-independent methods are therefore not really suitable for these applications. Text-dependent methods are also very vulnerable to attack if the user always speaks the same fixed text. This problem can be addressed by using a **text-prompted** method, whereby the user is prompted to enter a sequence of key words that is chosen randomly every time the system is used. To reduce the risk of deception it is best to use sequences of words spoken in a naturally connected manner, so that it would be very difficult to record all possible combinations or to generate natural-sounding imitations even with a fairly sophisticated concatenation machine. The greater the variety of different sequences that may be requested, the less opportunity there is for the system to be deceived, but the greater the enrolment effort.

14.3.2 Methods for text-dependent/text-prompted speaker recognition

For text-dependent speaker recognition, it is usual to first train one or more reference templates or HMMs for the required text. If the text is fixed, only one reference is required. An input speech sample is time-aligned with this reference and a similarity measure (a distance for the template method, or a likelihood in the case of the HMM approach) is accumulated for the duration of the utterance. Fixed-text methods tend to give the best speaker recognition performance because the text is known and hence these methods can fully exploit the speaker individuality that is associated with each sound in the utterance.

Text-prompted speaker recognition is more difficult because the system must both recognize the identity of the speaker and verify that the utterance is indeed a spoken version of the specified text. Thus an utterance should be rejected if its text differs from the prompted text, even if it is spoken by the registered speaker. If very many different text prompts are used, speaker-dependent sub-word HMMs may be needed so that a model can be created for each text prompt as required.

14.3.3 Methods for text-independent speaker recognition

Text-independent speaker recognition is especially challenging because the system does not ‘know’ what the words should be. Some of the main approaches to this task are summarized below.

Long-term statistics

Long-term statistics (such as the mean and variance of a set of spectral features) are computed over the duration of the test utterance, and compared against stored statistics for some reference speakers. This simple technique has the advantage that it is not necessary to make any hypothesis about the speech sounds that may have been uttered. However, long-term averages will inevitably blur the speaker-specific characteristics of individual sounds. Hence this approach lacks the discrimination power that is possible with the sequences of short-term features that are used for text-dependent methods.

Vector quantization (VQ)

A speaker’s voice characteristics are captured in a codebook of representative short-term feature vectors. At the recognition stage an input utterance is quantized using this codebook, and the total VQ distortion accumulated over the entire utterance provides the basis for the recognition decision. If the task is speaker identification, the distortion can be compared across different codebooks for all possible speakers. In the case of speaker verification, the distortion is evaluated against a threshold (possibly involving a comparison with the VQ distortion for codebooks for other speakers, or with the distortion for a generic speaker-independent codebook). Because VQ uses clusters of short-term features, this method can capture characteristics of different speech sounds, while not requiring any explicit knowledge of the identity of those sounds. It is, however, difficult to capture temporal information.

Statistical models

This method is similar to the VQ approach, but the codebook is replaced by a statistical model that is used to compute a likelihood for each speaker. This model may simply be a **Gaussian mixture model (GMM),** which is equivalent to a single-state HMM with a Gaussian mixture output distribution. A generalization of this approach uses a multiple-state **ergodic HMM** (an HMM which allows *all* possible transitions between states). After training, each state will tend to represent different spectral characteristics (associated with different phonemes) and the transition structure of the HMM allows some modelling of temporal information.

In general, statistical methods outperform VQ techniques, provided that sufficient training data are available. Experiments comparing the relative merits of GMMs and ergodic HMMs have suggested that in both cases the main factor affecting performance is the number of mixture components used to model the distributions. Thus in practice it seems that the introduction of the transition probabilities in ergodic HMMs has little effect on performance.

Large-vocabulary speech recognition

An alternative to the first three methods is to recognize the linguistic content of the utterance explicitly, and make a comparison based on models for the recognized linguistic units. The units for the recognition may be words, phonemes, or general phonetic categories. If sufficient training data are available, speaker-dependent recognition can be used. An alternative approach which requires less speaker-dependent training data, and for which these training data need not have been transcribed, is as follows. Speaker-independent models are first used to transcribe the training data (with the option of manual correction), and these models are then adapted to obtain speaker-dependent models. For recognition, the speaker-independent system is again used first to transcribe the data. This transcription is then used to compute the acoustic likelihood for the speaker-dependent models.

Speaker identification can be performed by finding the set of speaker-dependent models that gives the highest likelihood. For speaker verification the speaker-dependent likelihood is normalized by the corresponding likelihood for the speaker-independent models and compared against a threshold. The incorporation of a large-vocabulary speech recognizer makes the speaker recognition system quite complicated, but systems of this type perform well provided that the recognition is fairly accurate. Because open-vocabulary speech recognition is difficult for very short utterances (as less benefit can be gained from the language model), the performance of this approach to speaker recognition is more sensitive to the length of the testing utterance than are the other approaches.

For any given application, the best method to use will depend on factors such as availability of training data, length of typical test utterances and computational considerations. Overall, at the moment the GMM approach seems to be the most widely used and successful, although large-vocabulary recognition-based methods are competitive if there is enough test material (typically about 30 seconds).

14.3.4 Acoustic features for speaker recognition

For speaker recognition, the feature analysis needs to capture the characteristics of different talkers. Ideally features should be chosen that maximize the separation between individuals, while not being too sensitive to occasion-to-occasion variation within the speech of any one person. For some applications, robustness to within-speaker variation will need to include variability that may be introduced by an individual attempting to disguise his or her voice. There are many applications, such as those requiring speaker recognition over the telephone, for which the feature representation also needs to be robust to noise and to channel variations. Otherwise these variations could cause changes to the features that are larger than the differences between speakers.

Human perceptions of similarities and differences between speakers are influenced by many factors. Fundamental frequency and formant frequencies have a major influence, but other factors such as amplitude spectra of vowels and nasals and properties of the glottal source spectrum are also relevant. For automatic speaker recognition a variety of spectral and prosodic features have been tried. However, currently the most popular choice seems to be cepstrum-based features, similar to those that have proved to be so successful for ASR (see Section 10.5). The similarity in the choice of features may seem rather counter-intuitive, given that the requirements for speaker recognition are somewhat different from those for 'speaker-independent' ASR (for which the aim is to capture the properties of different speech sounds, irrespective of speaker differences). The success of cepstral features is probably due mainly to their compatibility with the use of multivariate Gaussian mixture distributions. Another relevant factor is the availability of straightforward techniques for improving the robustness of the features to channel distortions: it is usual to include time-derivative features as well as the cepstral features themselves, and to apply some form of cepstral mean subtraction. While it is widely acknowledged that additional features, especially those related to prosody, should be beneficial for speaker recognition, in practice so far only rather limited success has been achieved with such features.

14.3.5 Evaluations of speaker recognition performance

Several speech databases suitable for evaluating speaker recognition systems are now publicly available. One database for evaluating text-dependent speaker recognition is the YOHO corpus (Campbell, 1995). This database comprises 'combination lock' phrases (e.g. "twenty-six, eighty-one, fifty-seven"), recorded by 138 speakers using a high-quality telephone handset in a fairly quiet office environment (and not over a telephone channel). For 10-s utterances (four phrases), typical EERs that have been achieved using this database for speaker verification are lower than 0.5%, while closed-set speaker identification error rates are lower than 1.0%. When using just one phrase (2.5 s), error rates are somewhat higher.

Starting in 1996, the U.S. organization NIST has conducted evaluations of text-independent speaker recognition. These evaluations are similar in format and organization to the speech recognition evaluations described in Chapter 12. The speaker recognition evaluations have used recordings of unscripted telephone conversations, with variability in telephone lines and handsets. About one minute of training data is available for each speaker. One of the recognition tasks involves the standard verification paradigm of detecting whether or not a sample of speech is spoken by a specified individual. On this difficult database, the EER of the best systems is typically around 10% (Reynolds *et al*., 2000). Recent evaluations have also included even more challenging tasks, involving detection and tracking of speakers in recordings containing both sides of a telephone conversation. Error rates are higher for these tasks; for example, in 2000 the system with the best detection performance gave an EER of 14.0% (Reynolds *et al*., 2000).

Just as for ASR systems, speaker recognition systems tend to perform much better for some speakers ('sheep') than they do for other speakers ('goats'). In the case of speaker verification, the 'goats' tend to suffer a disproportionately large number of false-rejection errors. It has also been observed that there are some speakers who tend to have high false-acceptance rates (and therefore make good impersonators), while others are particularly prone to suffering from these false acceptances (and are thus easy to impersonate). The underlying acoustic reasons for these differences between speakers are still not fully understood.

14.4 LANGUAGE RECOGNITION 14.4.1 Techniques for language recognition

Humans can very quickly decide whether speech is from a language they know, and if it is, they identify that language simultaneously with recognizing what the words are. Even if they do not know the language, people can often make a good guess as to likely languages. Languages each have their own characteristic patterns of sounds, and people are sensitive to these differences between languages without necessarily speaking the languages. Systems for automatic language recognition also need to be able to detect and exploit these differences. A variety of methods have been employed, differing in the extent to which they use knowledge about the linguistic content of each language. Some of these methods are summarized below.

Large-vocabulary speech recognition

Several recognition systems are run in parallel. Each recognizer is configured for a different language, with its own acoustic and language models. The language is identified as the one corresponding to the recognizer that gives the best score for its recognized word sequence. This approach is computationally intensive, but can perform well provided that a suitable large-vocabulary recognition system can be set up for each language that is included in the task. It is important that the recognizer for each of the different languages produces output that can be easily compared with the output of the other recognizers, and in particular that the different recognizers are trained on similar speech corpora for their respective languages. To configure a language-identification system for a new language requires orthographically transcribed speech training data, a pronunciation dictionary and suitable texts for training a language model. Such material is not available for very many of the world’s languages, so identification of these languages requires techniques which do not rely on knowledge of the words in the language and which do not need such large amounts of training data.

Acoustic pattern matching

Using methods similar to those employed in speaker recognition, a VQ codebook or a statistical acoustic model (typically a GMM or an ergodic HMM) is trained for each language of interest. A test utterance can then be identified as belonging to the language for which the codebook or model matches the best. These methods are based purely on acoustics and therefore do not require knowledge of the linguistic structure of a language. However, because there is no temporal modelling (or, in the case of ergodic HMMs, only weak temporal modelling), it is not possible to utilize the phone sequence information that is a major distinguishing characteristic of different languages.

Phone recognition

The key component here is a phone language model, which is trained to specify legal phone sequences (and associated probabilities) for each language. Typically, *N*-gram models are used (see Section 12.7.1). This approach captures the **phonotactic** constraints of different languages, but without needing any explicit knowledge of their lexical content. If phonetically transcribed data are available for training in all the languages of interest, both acoustic models and phone language models can be trained for each language. Language identification is then possible by running all the phone recognition systems in parallel, incorporating the phonotactic constraints within the recognition search for each system, and finding the system that gives the best recognition score. This method is similar to the large-vocabulary recognition approach described above, but with phones (rather than words) as the largest recognition unit.

If phone-labelled training data are not available for all the languages to be identified, a different method is needed. A simple option requires a phone recognizer for just one language and some unlabelled training speech from each of the languages that the system needs to recognize. To train the system to recognize a language, the phone recognizer (without any phonotactic constraints) is applied to the training data for that language and its output is used to train an N-gram phone language model. To perform recognition, the phone recognizer is applied to the test utterance (again without any phonotactic constraints) and its output is scored by all the different phone language models. The language is identified as the one corresponding to the language model with the highest likelihood. Problems tend to arise if any of the languages to be identified contain sounds that are very different to the sounds that occur in the single language on which the phone recognizer was trained. This limitation is addressed in a variant of the method that uses several phone recognizers, one for each of a range of different languages. Each language model then scores the output of all the phone recognizers, and these scores are combined to obtain the total likelihood for the language.

Overall, the performance of the large-vocabulary recognition approach is hard to beat if suitable recognition systems are available for all the languages of interest. Otherwise, phone-based recognition has so far been the most successful, with the preferred method depending on what training data are available. Performance depends on the number of languages to be distinguished and on the acoustic similarity of those languages, as well as on the length of the test utterances. For example, Zissman (1996) obtained error rates of 2% for 45-s utterances and 5% for 10-s utterances when choosing between just two languages (with error rates averaged over all possible pairings of 11 different languages). When distinguishing between all 11 languages, the error rates increased to 11% for 45-s utterances and 21% for 10-s utterances.

14.4.2 Acoustic features for language recognition

As with speech and speaker recognition, currently most systems for language recognition use MFCCs and their time derivatives. MFCCs represent the spectrum that is related to vocal tract shape, but they do not capture the prosodic information that is also a distinguishing characteristic of languages. By definition, pitch is a necessary feature in the recognition of tone languages, but the way in which speakers use pitch also varies across other (non-tone) languages. In practice, however, so far only limited success has been achieved by using fundamental frequency as a feature for language recognition. Timing patterns are also different in different languages, and information about phone and syllable duration has been used with some success to assist in language identification.

CHAPTER 14 SUMMARY

- Speaker characteristics include speaker identity, sex of the speaker, and the language and accent in which that person is speaking.
- Automatic recognition of speaker characteristics requires acoustic analysis followed by pattern matching using either templates or statistical models.
- Identification tasks involve choosing one out of a set of possible categories (e.g. speakers or languages). These categories may form a closed set, or they may be an open set which includes the additional possibility that the utterance does not fit into any of the given categories and so should be rejected.
- Verification tasks involve detecting whether or not an utterance belongs to one specified category. Errors may be either false rejections (misses) or false acceptances (false alarms). The relative proportions of the two types of verification error depend on the setting of the acceptance threshold. The effect of varying this threshold can be shown graphically using some form of receiver operating characteristic (ROC) curve. A simplified measure of performance is provided by the equal error rate (EER), which is the error rate obtained when the threshold is set so that the probability of false acceptances is equal to the probability of false rejections.
- Speaker recognition can be text-dependent, text-prompted or text-independent.
- Both large-vocabulary ASR and Gaussian mixture models (GMMs) have been used with some success for automatic speaker recognition.
- For automatic language recognition, running several large-vocabulary ASR systems in parallel can work well when systems are available for all languages to be distinguished. Otherwise, phone-based recognition followed by language-specific scoring of phone sequences is a reasonable alternative.
- Currently most systems for speaker recognition and for language recognition use cepstrum features such as MFCCs, although prosodic features are also important for distinguishing speakers and languages.

CHAPTER 14 EXERCISES

E14.1 Explain the different types of errors that can be made by a system for identifying whether an utterance is spoken in English, in French or in some other (unspecified) language.

E14.2 How would you assess performance of a speaker verification system, and make a choice of value for the acceptance threshold for a given application?

E14.3 What is the crucial difference between text-dependent and text-independent speaker recognition? Which tends to be more accurate and why?

E14.4 How is large-vocabulary ASR applied to speaker and language recognition?

CHAPTER 15

Applications and Performance of Current Technology

15.1 INTRODUCTION

In the past few years there has been a large and continuing increase in the number and range of products and services that incorporate speech technology. More and more people have experience of an application that uses speech technology in some way. This increase in applications is due partly to the advances in the methods that are used in speech synthesis and recognition, but also to the more general progress that has been made in computer technology. The increases in the computer power and memory that have become available at decreasing cost have contributed to the growth of speech technology in two ways. Firstly, the advances in computers have been a crucial factor in much of the recent progress in the speech synthesis and recognition techniques themselves. Secondly, the widespread use of computers has opened up new opportunities for exploiting speech technology. The fantastic growth of the Internet has created a demand for easy ways of accessing and retrieving all the information and services that are becoming available. Also highly relevant to the application of speech technology are the more general developments that have taken place in telecommunications, including the growth in mobile telephony. There are now a vast number of automated telephone-based services, for which voice is the most natural means of communication.

The main aim of this book has been to introduce essential concepts and techniques in speech synthesis and recognition, and it is neither possible nor appropriate to give a detailed treatment of the many applications here. However, the demands of applications are usually the driving force behind developments in the techniques. In this chapter we will give an overview of some of the different applications for speech technology, concentrating on the relationship between what is needed for these various applications and the capabilities of the technology.

15.2 WHY USE SPEECH TECHNOLOGY?

As we pointed out at the start of Chapter 1, speech is not always the most appropriate or the easiest means of communication between humans and machines. Speech technology must offer some tangible advantage over alternative options if it is to be successful in any given application. Potential advantages include:

1. Cost savings may be obtained by automating services or by making human operators more efficient.
2. Effectiveness may be improved, for example in terms of speed and quality of output or in terms of ease with which a goal can be achieved.
3. Safety may be increased by using an additional modality for communication.

Situations in which there are obvious advantages to be gained from applying speech technology can be categorized as follows:

1. *Hands busy, eyes busy:* The usual mode of communication with a computer or other machine is to input commands using the hands, and to receive output visually. However, in situations when it is not possible to use the hands and/or the eyes, speech can provide a valuable alternative means of communication. Such situations arise when a person's hands and eyes are occupied, for example operating some piece of equipment, but also when hands or eyes cannot be used for some other reason such as disability, or the need to operate in darkness or to wear special equipment that makes manual operation difficult.
2. *Remoteness:* The telephone makes it possible to communicate with computers remotely. Although touch-tone phones are now widespread and can be used to input information, speech is more natural and is much easier for many types of information. For the machine-human direction of communication, speech is the only really viable option.
3. *Small devices:* Computers are becoming miniaturized. For communicating with palmtop computers and other small devices, there are many circumstances in which speech is easier than using a pointing device or limited keyboard. Similarly, when there is only a small display available, speech can provide a better means for output of many types of information.

Speech technology performance does not yet approach human performance, but there are many tasks for which the technology is useful. In situations where speech technology is providing users with a facility they would not otherwise have, those users will generally be more tolerant of limitations in the technology than will users of applications for which there are other alternatives. However, for any application, achieving success is critically dependent upon designing the system and its **user interface** to take into account the strengths and weaknesses of the particular technology that is to be used given the requirements of the application.

15.3 SPEECH SYNTHESIS TECHNOLOGY

When spoken messages are required, currently the main choice is between a text-to-speech (TTS) system and digitally recorded speech, possibly compressed using some speech coding algorithm. Recorded speech offers the best quality, or alternatively, with some loss in quality, can be used in coded form very cheaply on simple DSP chips. The principal disadvantage is lack of flexibility: if a new word or a different type of message is required, it is necessary to make a new recording. In addition, there may be practical difficulties in using recorded speech if a large number of different messages are required. For applications where the messages are unpredictable or are likely to be changed frequently, TTS is the only practical option. As has already been discussed in Section 7.7, the best TTS systems produce speech that is highly intelligible and sounds fairly natural on short, straightforward utterances. However, for longer passages, especially those requiring the expression of emotion, the perceived quality can drop dramatically. Thus at present the most successful applications for TTS synthesis are those needing only short utterances with simple intonation patterns, or those applications that need more complex utterances but for which lower quality will be tolerated because the system provides the users with a facility which they would not otherwise have.

For some speech synthesis applications (such as telephone-based information services), the requirement is really for a message preparation system. Such a system needs the flexibility of message content that TTS offers, but not the full range of capabilities to deal with any arbitrary text because the service provider has control over the input to the synthesis system. Given suitable tools together with the TTS system, a service provider can easily correct errors that may occur in word pronunciations or in the initial text analysis, and also 'mark up' the text to indicate, for example, which words should be stressed or where to put pauses. These types of facilities can be used to get around many of the limitations of current TTS while still providing much greater flexibility than is possible with pre-recorded messages.

Another issue that is relevant to the application of speech synthesis concerns memory requirements. At the moment the TTS systems which give the best quality use a lot of memory (see Section 7.6). A system of this type may be practical when it can be held centrally and used to service many telephone lines for example, but will generally not be an option for incorporating in a small, low-cost product.

15.4 EXAMPLES OF SPEECH SYNTHESIS APPLICATIONS 15.4.1 Aids for the disabled

One of the longest-established applications of TTS synthesis is in reading machines for the blind. The first such machine, combining an optical character reader with a TTS synthesizer, was produced by Kurzweil Computer Products in the 1970s. Even now, this speech synthesis task is very difficult as the machine must cope with any arbitrary text, and the quality of the speech that is generated would be regarded as insufficient by many people. However, these systems provide the visually impaired with the facility to read text that would not otherwise be available to them. Because these users are very motivated, they tend to be much more tolerant of errors and will learn to understand even poor quality TTS output very well. Indeed, someone who is familiar with the speech may choose to increase the speed of speaking to several times faster than normal speed and still understand the speech well enough to successfully search for some particular part of a document which is of interest.

The requirements for aids for people with speech impairments are rather different. Here the speech synthesizer acts as a means to communicate with other people, so the speech must be intelligible, preferably natural-sounding and ideally with a voice that is appropriate to the person who is using it. However, because the user has control over what the machine is required to speak, text pre-processing is not an issue and mark-up facilities can be used to improve quality and expressiveness. Commonly required utterances can even be prepared in advance.

15.4.2 Spoken warning signals, instructions and user feedback

Speech synthesis can be used to provide spoken warnings in emergencies. Spoken warnings are especially useful in eyes-busy, stressful environments where visual warnings may go unnoticed. A good example is the cockpit of a fighter aircraft. Speech synthesis may also be used more generally in hands-busy, eyes-busy situations such as when operating or repairing complicated equipment, to provide spoken instructions, feedback and so on. For all these types of applications, the messages may be recorded specially or a TTS-based message-preparation system can be used, depending on whether there is expected to be a requirement to change the messages. Care is needed to choose a voice that is the most effective for attracting attention in the environment in which the system is to be used.

15.4.3 Education, toys and games

Beginning with Texas Instruments' "Speak & Spell" in the 1970s, dedicated speech synthesis chips have been used in educational toys and in other toys and games. These synthesis chips are typically used to provide a fixed set of coded messages at low cost. There are also many opportunities for applying speech synthesis in the field of education. Possibilities include teaching foreign languages, teaching vocabulary and pronunciation to children learning their native language, and tools to assist in correcting speech defects. These applications can also incorporate a speech recognition component to provide feedback to learners about the accuracy of their pronunciations. For educational applications, high-quality output is normally very important, and so recorded speech is generally used at present. However, TTS synthesis allows much greater flexibility and the recent advances in speech quality are making it more viable as an alternative.

15.4.4 Telecommunications

Information services and interactive voice response systems

There is a vast quantity and variety of information that is stored on computers and for which there is a demand to be able to access remotely over the telephone. If the message structure is controlled and the words are not likely to change, it is practical for these systems to use recorded speech. Speaking clocks and directory enquiries services are examples for which a large number of different messages are required but the structure and vocabulary is sufficiently constrained for recorded speech to be applicable. For other applications, requiring a large vocabulary or messages of an unpredictable nature, it is more appropriate to use TTS synthesis. Examples include services providing access to public information, such as current stock market prices, news and weather reports, sports results, and so on. Other examples involve accessing more personal information, such as recent bank-account transactions, or the status of an order made through a mail-order catalogue.

In many situations, rather than just passively accessing information over the telephone, a person may wish to interact with and influence the remote system. Automated systems of this type, based on spoken output, are generally referred to as **interactive voice response (IVR)** systems. Examples include making banking transactions, booking travel tickets and placing orders from a mail-order catalogue. In many IVR systems the person is required to communicate with the machine using a touch-tone keypad, but alternatively ASR can be used (see Section 15.6.5).

Remote e-mail readers

A specialized but very useful application of TTS synthesis is to provide remote access to e-mail from any fixed or mobile telephone. For an e-mail reader, a full TTS conversion facility is required because the messages may contain any text characters. E-mail messages are often especially challenging, due to the tendency to errors of spelling and grammar as well as the special nature of the language, abbreviations and so on that are often used. There are also many formatting features that are specific to e-mail. For example the message header needs to be processed appropriately to extract relevant information, such as who the message is from and when it was sent. Other important facilities include an ability to navigate through messages, with options such as repeating, going back to a previous message or on to the next one. Commands from the user to the system may be implemented using speech recognition technology, or using the telephone keypad.

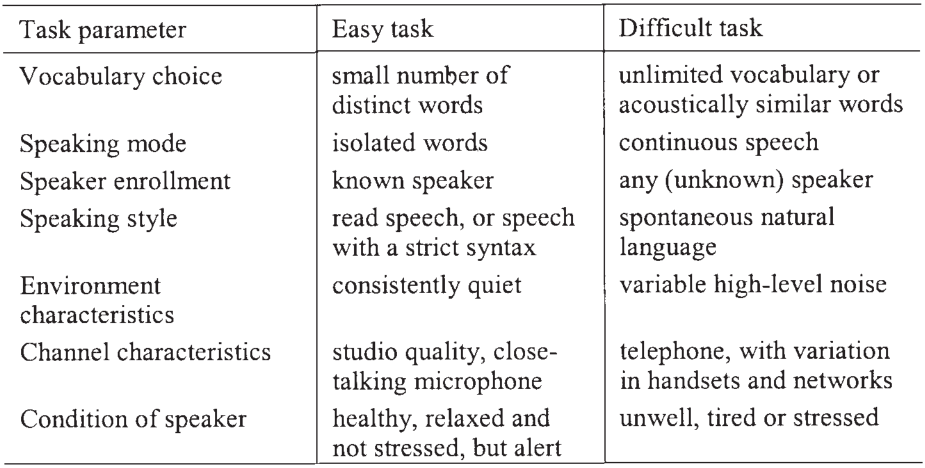
There are a number of commercial products available for remote reading of e-mail. Although the quality that can be achieved using TTS synthesis is still rather limited for this application, these products can be very useful because they make it possible to keep in touch with e-mail without needing to carry a computer around.

15.5 SPEECH RECOGNITION TECHNOLOGY

15.5.1 Characterizing speech recognizers and recognition tasks

The task of an ASR system is to respond appropriately to spoken human input, and the difficulty of this task is affected by a whole range of factors related to characteristics of the users' speech and the environment in which they are speaking. The main parameters which influence the difficulty of ASR tasks are summarized in Table 15.1.

Table 15.1 Parameters to characterize ASR tasks, with examples of easy and difficult tasks.



Task parameter	Easy task	Difficult task
Vocabulary choice	small number of distinct words	unlimited vocabulary or acoustically similar words
Speaking mode	isolated words	continuous speech
Speaker enrollment	known speaker	any (unknown) speaker
Speaking style	read speech, or speech with a strict syntax	spontaneous natural language
Environment characteristics	consistently quiet	variable high-level noise
Channel characteristics	studio quality, close-talking microphone	telephone, with variation in handsets and networks
Condition of speaker	healthy, relaxed and not stressed, but alert	unwell, tired or stressed

Each of the task parameters listed in Table 15.1 is explained in more detail below:

1. *Vocabulary choice:* It is easier to distinguish between a small number of words that are acoustically very different than to choose between a much larger number of words or between words that are acoustically very similar.
2. *Speaking mode:* In **isolated-word recognition** tasks, the speaker leaves a gap between each word and so co-articulation effects between words are avoided. **Continuous speech recognition** is more difficult due to between-word co-articulation and the difficulty of locating word boundaries.
3. *Speaker enrolment:* If the task is to be performed by known individuals and each person can provide sufficient suitable speech to train the recognizer, a **speaker-dependent** system can be set up for each person. **Speaker-independent** recognition is much more difficult, as here it may be necessary to recognize speech from any arbitrary person without knowledge of relevant factors such as gender, age group, dialect or even language. Substantial improvement over raw speaker-independent performance is possible by employing **speaker adaptation** techniques (see Section 11.4). Speaker adaptation is most effective if a person uses the system over a long period of time and corrects any recognition errors.
4. *Speaking style:* Read speech is generally easier to recognize than spontaneous speech, which usually contains more hesitations, errors and corrections, mispronunciations and so on. The recognition task is also easier when the talker can be trained to follow a strict syntax specifying allowed utterance constructs, than when unconstrained natural language must be accommodated. For situations in which it is applicable, a carefully designed syntax can ensure that the effective vocabulary at any one point is small and distinct, even if the total vocabulary size is large. Another aspect of speaking style is speech level: recognition performance tends to be best for speech spoken at a consistent moderate sound level, and worse when speech is shouted or whispered.
5. *Environment characteristics:* Recognition accuracy tends to be higher in a quiet environment than a noisy one, but the most important factor is to match the training environment as closely as possible to the environment in which the recognizer will be used. Conditions of time-varying noise are especially problematic. Another difficulty, which is associated with environmental characteristics, is that users often change the way they speak when the environment changes (the **Lombard effect,** see Section 11.2), for example shouting in an effort to be heard above the level of any noise.
6. *Channel characteristics:* In general, if the bandwidth of the speech signal is limited (as in speech transmitted over the telephone), the recognition task becomes more difficult because less information is available. Other problems that can occur with telephone-based systems include distortions due to handsets and telephone networks; cellular networks are especially problematic. More generally, the type of microphone affects the speech quality and, for example, speech recognizers tend to work better when a close-talking microphone is used than if it is necessary to use a far-field microphone. As with other factors, performance tends to degrade with any *variation* in the microphone that is used.
7. *Physiological/psychological condition of speaker:* Mild illness such as a common cold can change an individual's voice characteristics. Other relevant factors include fatigue and both emotional and physical stress (such as the high g force experienced

in jet aircraft). Any change to the speaker's voice is yet more variation that can present problems to a speech recognizer.

For an ASR application to be successful, the recognizer capabilities must match the requirements for the task in terms of the parameters listed above and also in terms of recognition accuracy and any other relevant considerations, such as cost, memory and processing requirements, real-time operation, etc. It may be difficult to find a system that both satisfies the necessary economic criteria and meets all the task requirements, while giving a sufficiently high level of recognition performance. However, by making some compromises in what is required of the task, it may be possible to achieve high-enough recognition accuracy. In general the different parameters can be traded against each other so that, for example, in a hostile, stressful, noisy environment a small vocabulary of command words may be practical. On the other hand, in quiet conditions with a known user and a close-talking microphone it may be possible to achieve useful performance recognizing natural language with a large vocabulary.

15.5.2 Typical recognition performance for different tasks

When assessing the accuracy of a recognizer in operational use, it is difficult to control all the factors that may affect the performance level. However, a useful indication of performance can be obtained from laboratory tests on databases of speech that have been previously collected under known conditions. A variety of databases are available, including the ones used for the competitive evaluations that have already been mentioned in Sections 12.9 and 12.10. Figure 12.6 showed how performance has improved over time, and Table 15.2 shows some recent (speaker-independent) recognition performance figures for speech databases which differ both in vocabulary size and in speaking style. From these performance figures it is evident that the technology performs well enough to be applicable to digit-recognition tasks and to tasks requiring recognition of considerably larger vocabularies in a fairly constrained domain, such as airline travel information.

For large-vocabulary tasks, involving recognition of 10,000 words or more, recognition performance is greatly affected by the type of speech material, as we have already seen in Section 12.9. Read newspaper texts are easier to recognize than television and radio broadcasts because newspaper texts have a quite specific, consistent style. Live broadcasts, on the other hand, may contain a great variety of different material as well as particularly difficult background noise (including speech, music and so on). Conversational speech is even more challenging, especially when the conversations are between individuals who know each other very well. In these situations the familiarity between talker and listener is such that speech tends to be produced very casually, and the talker often relies on the listener using shared knowledge and experience to understand a message with minimal acoustic cues. Current ASR systems do not possess the personal knowledge that people rely on in these situations and so for this type of speech the percentage of recognition errors is several times that for read speech, even when the vocabulary size is much smaller. Thus, while large-vocabulary recognition is good enough to be deployed when the situation is constrained and the environment is controlled, performance is not yet sufficiently high for transcription of less restricted material.

Table 15.2 State-of-the-art word error rates in laboratory recognition tests for different speech corpora.

Corpus	Type	Vocab. size	Error rate	Source for error-rate figure
Connected digit strings	spontaneous	10	0.3%	Rabiner (1997)
Airline travel information	spontaneous	2,500	2.0%	Rabiner (1997)
Newspaper texts	read text (close-talking microphone)	64,000	6.6%	Woodland *et al.* (1996)
Television and radio broadcasts	mixed read/ spontaneous	64,000	13.5%	Pallett *et al.* (1999)
Strangers' telephone conversations	conversational telephone	10,000	19.3%	Fiscus *et al.* (2000)
Family telephone conversations	conversational telephone	10,000	31.4%	Fiscus *et al.* (2000)

15.5.3 Achieving success with ASR in an application

To be successful in an application, ASR technology must give adequate recognition performance for the required task. Word accuracy is a useful measure, especially for a task requiring accurate transcription of what a person says (dictation for example). However, when speech is used to retrieve data or to give commands, success means achieving the required result with each spoken input, not necessarily recognizing every word accurately. Some recognition errors (of function words for example) will not matter, whereas others will be critical.

Any recognition system will make errors sometimes. Users' perception of ASR technology depends very much on how errors are handled and on other aspects of the user interface. It is important to provide appropriate feedback to the user so that he or she is made aware of any recognition errors, and to provide a means for the user to correct these errors. Sometimes the user may not speak clearly or may give a response that does not match any of the allowed options, and it is therefore often helpful to estimate 'confidence' in the recognition accuracy (see Section 11.6). If confidence is low, the system can respond by, for example, requesting clarification or repeating the allowed options.

The system must always respond to the user quickly, and allow any input to be 'undone' in the case of errors either by the system or by the user. In addition it is important to always make clear to the users what is expected of them at any point in an interaction, especially in systems designed for naive users. At the same time provision must be made for the expert user, for example to barge-in over spoken prompts. These **human factors** considerations are crucial to the successful application of ASR, and the detailed design of the system and its user interface will depend on the application. We will touch on some of the issues again in the following discussion, but more detailed treatment of the role of human factors in ASR is outside the scope of this book.

15.6 EXAMPLES OF ASR APPLICATIONS

15.6.1 Command and control

The term “command and control” is used to refer to applications in which a person uses simple voice commands to control functions of a machine. These applications tend to be associated with situations in which hands-free, eyes-free operation is required. Voice control is best suited to functions requiring selection between a discrete set of choices, rather than to selection of continuous quantities or to positional control, and ASR is of course not suitable for safety-critical functions.

Command-and-control systems often have to work in difficult, noisy environments, possibly with the users under stress. However, many of these applications are successful with current technology because the vocabulary size tends to be small, the users are generally known to the system and in some cases may even be highly trained. Well-established applications of this type can be found in the military environment; for example, ASR has been operated successfully in fighter aircraft for functions such as setting radio frequencies and controlling flight displays, and has been included in the Eurofighter 2000 aircraft from its earliest design stages. Another traditional area for command-and-control applications is in factories and other industrial environments, to enable machinery to be operated without requiring hands and eyes to be distracted from the primary task.

There are also commercial command-and-control applications. For example, software packages exist which enable users to customize their PCs for voice control of functions such as menu selection, Web browsing, etc. One application area where voice control is of obvious benefit is in cars, for controlling equipment such as the car radio and, in particular, for voice-controlled dialling of mobile telephones. A number can be entered by speaking the required digit sequence or by speaking some previously programmed repertory entry, such as a name or a descriptor such as “home”. Although repertory dialling requires the user first to train the system by speaking the required words, subsequent recognition performance will generally be better than can be obtained for long digit strings and the usability of the voice-dialling facility is greatly enhanced. Voice dialling is an attractive facility that is now included with many mobile telephones.

15.6.2 Education, toys and games

Speech recognition can be used in the field of education for a variety of applications, closely linked to the speech synthesis applications mentioned in Section 15.4.3. Current uses of ASR generally involve assessing the accuracy of pronunciation of specified words. PC-based software products are available, both for foreign-language teaching and for assisting children in learning to read.

There is potentially a very large consumer market for ASR technology in games and interactive toys. Low-cost special-purpose speech recognition chips are available and have been used in toys incorporating some simple speech recognition capability. Although in the past attempts to incorporate ASR in toys have not achieved widespread success, the situation is rapidly changing with the capabilities of current technology and the growing demand for toys that are interactive.

15.6.3 Dictation

An alternative to typing large amounts of text is to speak the words and have them transcribed automatically. Dictation applications of ASR are now established as a commercial reality. Early products required the user to speak words in an 'isolated' style, leaving short pauses between each word, but in 1997 both Dragon Systems and IBM introduced PC software products that accept continuous speech. Several companies now offer ranges of products (with different capabilities and in various languages), many of which can be purchased from a computer store for less than £100. Quoted word accuracies are around 95–98%.

Current dictation products typically have active vocabularies of tens of thousands of words, but are intended for use with a close-talking microphone, in a quiet environment and in a speaker-dependent mode. Before first using the system, it is necessary to train it by speaking some specified text, and it will then continue to adapt to the individuals voice (both acoustics and choice of words) as that person uses it over a period of time. In the initial period it is likely that the system will make many errors, and care and patience are required on the part of the user to correct these errors so that the adaptation can work properly. Speaking style is also very important: speech must be clear and spoken at a steady rate, without extraneous noises such as coughs, "ums" and "ers". Over time, not only does the system adapt to the user, but committed and successful users of these products adapt their speaking style to optimize the performance of the technology. At the moment, it seems that such dedication and prolonged training are necessary to get good recognition rates. Another crucial component of these products is the user interface, and in particular the ease of error correction. If it is easy to correct errors, users' perceptions are greatly enhanced, even if the product makes mistakes.

If voice dictation is used for preparing many documents of a particular type, productivity is much improved by using 'macros' to call up standard formats (such as letters, including commonly used addresses), as well as standard paragraphs and phrases. With facilities of this type, voice dictation applications have proved to be very successful in specialized areas such as medical reporting. For example, radiologists are responsible for interpreting X-rays and reporting their findings, and this would traditionally have been achieved by speaking into a tape recorder for later typing by a transcriptionist. There are now specially tailored ASR systems that allow radiologists to dictate directly into a computer, with resulting savings in cost and efficiency. This application, and other dictation applications involving professionals (such as doctors and lawyers) who are used to dictating documents in a very standard format, have proved very successful with current ASR technology.

15.6.4 Data entry and retrieval

We use the term **"data entry"** to refer to the input of information to a computer's data file (rather than dictation, which involves direct transcription of the words spoken). **Data retrieval** is the reverse process of accessing information that is stored in a computer system. Aside from telephone applications, which we will consider separately in the next section, typical application areas for data entry and retrieval *via* speech recognition involve hands-busy, eyes-busy scenarios similar to those already mentioned in Section

15.6.1. For example, speech recognition can be used in manufacturing to enter quality control information while inspecting product parts, and in dentistry to allow a dentist to carry out an oral examination of a patient and record the results at the same time without needing an assistant.

Data-retrieval applications of speech recognition include requesting instructions or detailed information such as specific measurements while conducting assembly or repairs. The information from the computer system can be provided using pre-recorded speech or speech synthesis (see Section 15.4.2).

When ASR is used to communicate with computers in military, industrial or medical applications, restrictions can be placed on the vocabulary and it is reasonable for the users to be trained to follow a defined syntax. A very different type of data-retrieval application for ASR involves cataloguing and extracting information from broadcasts or other recorded speech material. This task is very challenging because information must be extracted from material that is often completely uncontrolled. At a simple level, some classification of speech material into ‘topics’ (e.g. weather forecasts) is possible by extending keyword-spotting techniques (see Section 11.6) to look for groups of words that typify particular topics. More sophisticated systems for topic tracking and information retrieval use LVCSR, and are a subject of considerable current research (see Section 12.10.1).

15.6.5 Telecommunications

ASR enables people to interact with computers over the telephone in a much more natural and flexible manner than is possible using only a touch-tone keypad. Some applications are aimed at cost saving by removing or reducing the need for human attendants, while others provide new services that were not previously available. The users can be expected to be more tolerant of technology limitations for the latter type of application, but any telephone system for use by the general public has to cope with a very wide variety of voices (including people in different age groups, from different dialect regions and even non-native speakers). Thus very robust speaker-independent recognition is required. In addition, users of the system will often range from experts to first-time users, and the users cannot be relied upon to respond in the way that the system expects, even when given precise instructions. For systems that are intended to recognize only a limited vocabulary, the techniques described in Section 11.6 for keyword spotting and detection of out-of-vocabulary words can enable some sensible response to be given to most input. More elaborate systems include some spoken language understanding capability.

Automation of functions in telephone networks

Voice dialling has already been mentioned in Section 15.6.1, and there are also voice-directory products that are used by several companies for internal callers and by some hotels both for employees and for guests. These systems remove the need for paper directories and make internal communication much easier. A related application is in ‘automated voice attendants’ which some companies and department stores are now using to answer calls made to the main switchboard and then route these calls to a named department or person. AT&T Laboratories in the U.S. have developed a sophisticated voice attendant system which has enabled a great reduction in the need for human operators in the AT&T network. Rather than restricting the user to name a person or department, this system answers a call simply with "How may I help you?" and, based on the reply, enters into a dialogue with the caller to obtain additional information or clarification in order to process the call. The aim is to classify the call and pass it on to another automated system or to a human operator if necessary. Some understanding capability is needed, but not necessarily accurate recognition of every word in the utterances.

Information services and IVR systems

Many IVR systems rely on touch-tone selection, but this method can be very restrictive and is often unpopular with users who may respond by defaulting to the human operator because it is not obvious to them how to achieve their goal with the automatic system. Speech recognition provides a more intuitive interface and an easy way to select between large numbers of different alternatives. For example, there are ASR-based systems for providing stock quotes, and United Parcel Service in the U.S. uses ASR for a service that allows customers to arrange collections and to track packages. A number of companies offer services whereby people can call a single number and speak keywords to access a variety of different types of information (such as restaurant listings, traffic reports, sports scores and so on).

Some companies use speech recognition to handle travel information and reservations. As demonstrated in the ATIS research described in Chapter 12, this type of application requires some understanding and dialogue capabilities if it is to deal with the wide range of likely enquiries. Speech recognition can also be used in automated telephone banking facilities, allowing customers to check on account balances, credits and debits and to conduct simple transactions. One successful example is the "Anser" system from NTT in Japan, which was first introduced in 1981. Although this early system could only cope with isolated words from a very limited vocabulary, it was highly successful because the user interface was well designed and the system offered obvious advantages to the users in providing them with easy and immediate access to information about their banking transactions.

Remote access to e-mail, voice mail and messaging systems

Speech recognition provides a natural way to access the remote e-mail reading application of TTS synthesis that was described in Section 15.4.4. ASR can also be used for remote access to conventional voice mail messages, and to the growing number of 'unified messaging' systems. The key concept in unified messaging is to provide the user with access at any time to a single system for handling e-mail, voice mail, fax and pager messages. TTS synthesis can be used to regenerate communications (such as e-mail and fax) that were not originally in spoken form.

Some companies now offer the service of a personalized 'telecommunications assistant' that integrates several functions under a single voice interface. Typically, these systems handle messaging functions, screen and forward calls, allow voice dialling by name from a contact list, and may also provide other facilities such as news and stock quotes. Automated personal assistants are a fairly new, but expanding, commercial application for speech technology.

15.7 APPLICATIONS OF SPEAKER AND LANGUAGE RECOGNITION

While there are fewer applications for speaker recognition technology than there are for speech synthesis and speech recognition, the deployment of speaker recognition systems has increased in recent years. The applications can be divided into two general categories:

1. *Authentication for access restriction and fraud prevention:* A number of companies now offer speaker verification products for access control and fraud prevention. These products are often combined with speech recognition, and some are available for several different languages. Systems have been deployed for controlling access to telephone-based services, such as telephone banking and home shopping. Other applications include controlling building access, and validating users of the Internet or users of mobile phones.
2. *Monitoring, surveillance and forensics:* Automatic speaker recognition can be used for general monitoring of voice recordings, or more specifically for checking on the whereabouts of particular individuals. For example, speaker verification is used for the automatic monitoring of offenders who have been released under restrictions such as home detention. Forensic evidence based on identification of individuals from voice recordings has a long but controversial history. Traditionally this task is performed by a forensic phonetician, but automatic systems are sometimes used to assist in the process.

Automatic language identification has applications for surveillance and monitoring of communications, which are of interest to the military, for example. To the authors' knowledge there are not yet any automatic language recognition systems in commercial use. Potential applications include automatic routing of multilingual telephone calls. For example, calls to the emergency services could be directed to an operator who can converse in the relevant language. Language identification can also form a component of systems for multilingual speech recognition or spoken language translation, which have so far been demonstrated as research systems but which should achieve commercial realization in the future.

15.8 THE FUTURE OF SPEECH TECHNOLOGY APPLICATIONS

The commercial exploitation of speech technology looks set to continue to expand in the coming years, closely linked to more general developments in information technology. Telephone and Internet applications will continue to grow. Computers are progressing in the direction of small-scale and embedded computing devices, and intelligent software 'agents' are being developed to manage interactions. With these developments, the ability to interact by voice will become more important.

Although many applications already incorporate both speech synthesis and recognition, much more integration and incorporation into multimedia interfaces is expected in the future. Future applications of speech technology will require higher capability in spoken language understanding and natural language generation, including a greater ability to deal with multiple languages and translate between languages. Exactly what will be achieved and in what timescales will depend on progress in speech technology research, which is considered in the next chapter.

CHAPTER 15 SUMMARY

- To be successful in an application, speech technology must offer an advantage (e.g. in terms of cost, effectiveness or safety) over alternative options.
- Speech technology offers most benefit when manual/visual communication between human and machine is difficult: when the hands and eyes are busy, to communicate remotely *via* the telephone, or if the machine is very small.
- For voice output, there is a choice between using recorded natural speech or a TTS system. TTS is more flexible, but the perceived quality is limited, especially for long pieces of text. A compromise that is suitable for some applications is to use a TTS-based message preparation system.
- Speech synthesis applications include: aids for the disabled; systems for giving spoken warning signals, instructions and feedback to users of complex machines; voice output for toys and educational systems for teaching native or foreign languages; telephone services such as remote e-mail readers, information systems and interactive voice response (IVR) systems.
- Many factors influence the difficulty of a speech recognition task: choice and size of vocabulary; speaking mode (isolated words versus continuous speech); whether or not speakers are enrolled to use the system (speaker-dependent versus speaker-independent recognition); speaking style (read versus spontaneous versus conversational speech); environment and channel characteristics; physiological and psychological condition of the speaker.
- Recognition of spontaneous speech in a constrained domain (e.g. airline travel information), and of read speech for large vocabularies but under controlled conditions, is adequate for deployment now. Error rates for transcription of conversational speech are still too high for widespread application.
- A good user interface, including provision for correcting the inevitable recognition errors, is crucial to the successful application of ASR technology.
- ASR applications include: ‘command and control’ of machine functions for military (e.g. fighter aircraft) and commercial applications (e.g. voice dialling for mobile telephones); educational systems for pronunciation learning; voice-controlled toys and games; voice dictation systems; data entry and retrieval; telephone services, such as automated operator services and access to e-mail readers, information services and IVR systems.
- The continuing growth of the Internet, mobile telephony and ever-smaller computers offers much potential for future applications of speech technology.

CHAPTER 15 EXERCISES

E15.1 Give three different types of situation in which speech technology is useful, with examples of both speech synthesis and speech recognition applications.

E15.2 What types of application benefit most from TTS synthesis and why?

E15.3 Why is it not possible to know how good a speech recognizer is from just a word accuracy figure? What other information is needed?

E15.4 Compare and contrast the characteristics of an office dictation application of ASR with those of a telephone-based flight booking service.

CHAPTER 16

Future Research Directions in Speech Synthesis and Recognition

16.1 INTRODUCTION

The last decade of the twentieth century saw a substantial growth in the capabilities of speech technology. Current performance of speech synthesizers and recognizers makes them already extremely useful for a variety of practical tasks, and they are now deployed in many applications. However, the performance of current technology still falls far short of what is normally achieved with ease by human beings. In speech synthesis, very good quality is possible for a restricted set of messages, but if complete flexibility of message content is required, even the best systems are significantly deficient in both intelligibility and naturalness when compared with typical human speech. In recognition, even the most advanced systems cannot provide the same level of accuracy that is achievable by a competent human speaker of the target language, except when the task is so constrained that the machine has very few output choices at any one time. In both synthesis and recognition, the gap between human and machine performance widens as the conditions become more difficult, for example involving spontaneous speech, emotional speech or noisy environmental conditions.

Although the task of improving performance of speech input/output devices is not trivial, there are a number of lines of work that show considerable promise for leading to significant improvements in technology capabilities.

The first point to emphasize is that, although immense complexity will be required in more powerful systems, the availability of computational resources is increasing all the time and not likely to be the limiting factor. In addition, as a result of the large number of data collection exercises that have been conducted in recent years, there are now plenty of speech databases available for training and testing recognition and synthesis systems. What seems to be required is to develop more powerful and robust techniques for recognition and synthesis, making better use of the resources that are already available.

16.2 SPEECH SYNTHESIS

The best TTS systems are now able to produce synthetic speech, in a neutral reading style, that sounds both intelligible and natural for many short passages of text. However, the systems that give the best speech quality are offered with very few different voices, and usually with very little flexibility to change the characteristics of a voice or the speaking style. Furthermore, on passages of more than just a few sentences even the best systems quickly become very boring to listen to, and some words may give problems even for short passages. If an expressive speaking style is required, such as would be appropriate when reading a story for example, the quality of speech produced by TTS systems is still not really good enough to be useful except to highly motivated users. Achieving variety, flexibility and appropriate expressiveness in speech synthesis will require research into improving all levels of the automatic speech generation process.

16.2.1 Speech sound generation

At the moment the best synthetic speech quality is provided by systems that use PSOLA-type concatenative techniques with a large inventory of variable-length segments. Typically the systems either use time-domain waveform concatenation or they include some waveform coding method that preserves most of the detail of the original waveform (e.g. LP-PSOLA, MBR-PSOLA). Good quality is only achieved by both a careful choice of the inventory of segments and careful extraction of suitable examples. Recent research has led to the development of automatic techniques for optimizing this selection and extraction operation, so that the process of setting up concatenative systems for new talkers is becoming easier. However, with a waveform-based coding method, the only changes that can be made are in the selection of the segments and there is no obvious way to enforce appropriate formant transitions across segment boundaries, or to model systematic changes in formant frequencies or bandwidths for example. This deficiency is widely recognized and is motivating current research into representations and methods that allow some co-articulatory and other spectral changes to be made to the concatenated units, while still retaining the high-quality coding of the speech. For example, if units are coded in a way that can be related to formant frequencies fairly reliably, it is possible to apply some small formant modifications such as are required for smoothing transitions across segment boundaries.

The search for representations and methods that facilitate greater manipulation of speech characteristics within a waveform-based concatenative framework will probably continue to be a focus for speech synthesis research for several years to come. However, even with these methods it seems unlikely that it will be possible to model co-articulation phenomena or to vary the voice characteristics to the same extent that can be achieved with rule-driven approaches.

Current output of phonetic synthesis by rule is considerably worse than that of the best concatenative systems and is unlikely to be mistaken for a recording of human speech, even if correct phonemic and prosodic specifications are provided. The potential sources of the deficiencies are in the speech production model and in the rules for controlling it. However, demonstrations made nearly 30 years ago (J.N.Holmes, 1973) showed for a few sentences that a parallel formant synthesizer could produce synthetic utterances that were almost indistinguishable perceptually from recordings of the natural utterances that they were copying. There is thus good evidence that the limitations of synthesis by rule are almost entirely in the rules for converting a phonemic/prosodic description into control signals for the synthesizer.

As discussed in Chapter 6, the difficulty in choosing appropriate phonetic rules to mimic real speakers has been the major obstacle to achieving truly natural-sounding synthetic speech by rule. While automatic training methods are well established in speech recognition, and have more recently also been used for concatenative synthesis systems, they have not been widely adopted in the case of phonetic synthesis by rule. However, as advocated in Section 6.5.1, automatic training methods are applicable to a rule system such as the one described in Chapter 6. More generally, if an appropriate synthesis *model* were available, it should be possible to train its parameters automatically. There is some recent research interest in developing this type of statistical speech model for application both to recognition and to synthesis, as will be explained in more detail in Sections 16.3 and 16.4. In the long term, this line of research may lead to the best solution to achieving natural-sounding speech synthesis, by modelling speech dynamics and capturing the effects of co-articulation. A model for variability may be required to prevent the synthetic speech from sounding too monotonous.

An advantage of automatic techniques is that they can be applied to derive synthesis parameters for any talker of any language or dialect, given enough labelled speech data to train the system. It should also be possible to apply speaker adaptation techniques to transform an existing set of models based on just a small quantity of speech data from a new talker. The long-term aim should be to develop a synthesis model that characterizes all the attributes that distinguish different individuals' voices. This way it may ultimately be possible to achieve truly flexible and natural synthesis of any specified voice quality on demand, rather than relying on a speech database for a particular talker as is the case at the moment.

16.2.2 Prosody generation and higher-level linguistic processing

Prosody is often cited as the major limitation to the quality of the speech generated by current TTS systems. Recent research has led to the development of automatic techniques for deriving the parameters of prosodic models, but further work is still needed to improve these models and to find the best automatic methods for training them. If the sound generation component is developed to provide an accurate model of co-articulation effects, the realization of different sounds should then vary appropriately with changes in the timing of articulatory movements, which should in turn facilitate development of improved models of timing. Intonation prediction maybe more complex because it depends on the choice of an abstract representation that captures every attribute that contributes to the characteristics of intonation. Prosodic transcription schemes such as TOBI (Silverman *et al*., 1992) have proved useful (see Section 7.5.2), but may not capture all the relevant information, especially if different intonational correlates of emotion need to be included.

Whatever prosodic labelling scheme is used, in a practical TTS system it will be necessary to derive the prosodic labels, as well as the phonemic labels, from analysis of text. It is these higher levels of TTS conversion that seem to present the most difficult challenge. Conversion of arbitrary text into a really accurate, detailed phonemic and prosodic description must require at least some understanding of the meaning of the text (see Section 16.5). Even if a representation of the underlying concepts is available, the problem of converting from concept to the phonemic and prosodic specification seems just as daunting. However, applications that require speech output within a limited domain (such as train timetable enquiry systems) are more manageable. Because such systems normally only need to offer a restricted range of messages in a known domain, relevant information such as syntactic phrasing and semantic focus is relatively easy to obtain. There is, however, still a need for better models to capture the relationship between this semantic and syntactic information and the required prosodic structure.

TTS systems of the future will need to be able to speak in different styles depending on the type of text (e.g. news report, e-mail message, children's story). Research is needed to find the best way to model these stylistic differences with controllable synthesis parameters, and to find methods for automatically training the models from suitable text and speech data. Sophisticated text analysis will also be required to automatically determine the style and structure of documents.

16.3 AUTOMATIC SPEECH RECOGNITION

Current ASR performance can be very impressive, even for tasks involving very large vocabularies. However, some marked deficiencies remain. In particular, ASR systems tend to be very sensitive to variation: changes in the acoustic environment, transmission channel, talker identity, speaking style and so on all cause much more problem for recognition by machines than for recognition by humans. The most successful ASR systems to date have been almost universally based on HMMs, as described in Chapters 9–12. Over the years there have been many refinements to the way in which the HMMs are used, and it seems likely that these incremental improvements will continue in the immediate future. However, there is also research interest in more substantial changes to and advances beyond the current HMM methods. The hope is that this research could eventually lead to a step improvement in recognition performance, especially in terms of robustness to all the inevitable, but often systematic, variability that is found in speech.

16.3.1 Advantages of statistical pattern-matching methods

While current HMM pattern-matching methods have serious limitations, they also have some very desirable properties, as already explained in previous chapters. In this section we will revisit some of the important advantages, before discussing the limitations in the next section.

The HMM methodology provides a tractable mathematical framework with straightforward algorithms for recognition and for training to match some given speech data. The spectral characteristics (emission p.d.f.s associated with states) and the temporal characteristics (the Markov chain with its transition probabilities) are treated separately but within the one consistent framework. As a consequence, segmentation of an utterance arises automatically as part of the training and recognition processes. In addition the models can be made to generalize quite naturally to unseen data, either by smoothing estimated discrete distributions (typically used for language models), or by using a parameterized continuous distribution such as a multi-variate Gaussian (now widely used for acoustic models).

During recognition, a result is only output when the partial trace-back through possible word sequences coalesces into a single path. This coalescence can cause the identities of a whole sequence of words to be determined simultaneously, and in fact the implied decision about the phonemic content of an early word in the sequence can be changed as a result of either acoustic or linguistic evidence for a later word. For example, assume that an early word is acoustically ambiguous between two possibilities. If linguistic knowledge (as expressed in a language model) indicates that a later word for which there is strong acoustic evidence could not follow one of the early candidates, the overall decision on the utterance will be biased strongly against that earlier word.

It can thus be seen that for any given utterance one-pass continuous recognition algorithms make a single decision about the word sequence as soon as they can reliably do so, after weighing up all the available evidence, both acoustic and linguistic. Provided no significant information has been lost in the acoustic analysis, the decision is made without prematurely discarding any relevant information. Experiments with human speech perception (e.g. Marslen-Wilson, 1980) strongly suggest that human speech recognition behaves in a similar way.

In Chapter 12 we mentioned that early systems for large-vocabulary recognition used a knowledge-based approach, whereby recognition was attempted by trying to detect and recognize phonetic features. The comparatively poor performance shown by these systems is due to the difficulties involved both in identifying all the required knowledge for performing speech recognition and in finding an appropriate way for specifying that knowledge. In contrast the stochastic methods require only a general structure whose parameters are trained automatically using a large amount of training data. The best HMM systems incorporate knowledge about speech, but this knowledge takes the form of *constraints* on the more general model structure. Examples include the choice of unit inventory (e.g. context-dependent sub-word models) and the selection of a model topology that only allows a subset of plausible transitions. Typical methods of acoustic feature analysis are also chosen taking account of knowledge about the phonetically important characteristics of a speech signal that need to be preserved.

16.3.2 Limitations of HMMs for speech recognition

HMMs provide a structure that is broadly appropriate to represent the spectral and temporal variation in speech. However, some assumptions are made in the HMM formalism that are clearly inappropriate for modelling speech patterns. Firstly, it is assumed that a speech pattern is produced by a **piece-wise stationary** process, with instantaneous transitions between stationary states. This assumption is in direct contradiction with the fact that speech signals are produced by a continuously moving physical system—the vocal tract. Secondly, in a first-order Markov model, the only modelling of dependency between observations occurs through constraints on possible state sequences (see Section 10.7). Successive observations generated by a single HMM state are treated as independent and identically distributed. By making this **independence assumption,** the model takes no account of the dynamic constraints of the physical system that has generated a particular sequence of acoustic data, except inasmuch as these constraints can be incorporated in the feature vector associated with a state. In a typical speaker-independent HMM recognizer in which each modelling unit is represented by a multi-modal Gaussian distribution to include all speakers, the model in effect treats each frame of data as if it could have been spoken by a different speaker. The independence assumption is also the cause of the inappropriate duration distributions discussed in Section 9.14 and shown in trace (i) of Figure 9.2, which arise because the probability of a model staying in the same state for several frames is determined only by the self-loop transition probability.

HMM recognition systems are usually designed to reduce the impact of the inappropriate modelling assumptions. For example, a generous allocation of states allows a sequence of piece-wise stationary segments to make a fair approximation to speech dynamics, and time-derivative features help to mitigate the effects of the independence assumption as well as capturing some information about local dynamics explicitly.

HMMs are well suited to modelling acoustic features obtained by short-time spectral analysis using a fixed window size (typically 20–25 ms) at fixed time intervals (typically 10ms). It is well known that this type of analysis has to compromise between capturing temporal properties and representing spectral detail, and is not a very good model for many properties of human auditory perception. For example, studies of human speech perception have shown the importance of dynamic information over many different timescales, ranging from as short as 2–3 ms to around 20–50 ms. In the case of prosodic information, much longer time intervals are also relevant. In order to incorporate such information operating at this wide range of timescales, it seems necessary to modify not only the methods of feature analysis, but also the nature of the models. For example, prosody provides information that helps in speech understanding, but simply adding prosodic information to an acoustic feature vector at 10ms intervals for modelling with HMM states seems unlikely to capture the necessary prosodic cues.

A first-order Markov process cannot capture more than very immediate linguistic influences, and long-range syntactic and semantic constraints are difficult to incorporate. As is the case for speech synthesis, recognition systems of the future will need more powerful models of language understanding, especially when dealing with spontaneous or noisy speech. We will return to this issue in Section 16.5 after first discussing how the acoustic modelling might be improved.

16.3.3 Developing improved recognition models

At present data-driven statistical methods have proved to give better recognition performance than knowledge-based methods, even though as presently formulated the data-driven systems ignore much of the information in the acoustic signal that we know is important for human speech recognition. In principle the ability to learn by example that is characteristic of the data-driven approach could be extended to incorporate a richer structure and to learn more complicated phonetic features.

One way of addressing the HMM assumptions of independence and piece-wise stationarity is to associate models with variable-length sequences, or ‘segments’, of acoustic feature vectors. It is then possible to characterize both the duration of the segments and the relationship between the vectors in the sequence associated with a state, usually incorporating the concept of a trajectory to describe how the features change over time in the segment. A variety of **segment models** have been investigated, using different trajectories and different ways of describing have been investigated, using different trajectories and different ways of describing the probability distributions associated with those trajectories. In comparison with conventional HMMs, some improvements in recognition performance have been demonstrated by modelling trajectories of typical acoustic feature vectors such as MFCCs. However, success is often dependent upon a careful choice of trajectory model and distribution modelling assumptions. It seems that, when introducing more structural assumptions into the model, the accuracy of the assumptions is critical to success. If the assumptions are not sufficiently accurate, performance may actually be worse than the performance of a conventional HMM, which, although it is a crude model, makes only a few very general assumptions.

The general concept of modelling the relationship between successive acoustic feature vectors seems desirable. However, the motivation for modelling dynamics and trajectories originates in the nature of speech production. It may therefore not be most appropriate to apply these models directly to transformed acoustic features such as MFCCs, which have a very complex relationship with the speech production system. To obtain the full benefit of trajectory modelling, it may be necessary to apply the models to features that are more directly related to speech production. These features could be some form of articulatory features, or alternatively they could be acoustic features that are closely linked with articulation, such as formant frequencies. Some success has been achieved in extracting and using articulatory and formant information for speech recognition, especially as a supplement to general acoustic features. However, formant or articulatory analysis is very difficult to perform reliably without any prior knowledge about the speech sounds, due to the complex many-to-one mapping that exists between articulation and its acoustic realization. (See Section 4.3.4 for more detail about the difficulties of formant analysis.) Furthermore, articulatory or formant features do not provide all the information that is needed to make certain distinctions, such as those relying on excitation source differences.

The desirability of modelling articulatory or formant dynamics, together with the difficulties involved in extracting the relevant information, have led a number of workers to suggest that this information is best incorporated in a multiple-level modelling framework. The idea is to introduce an intermediate level between the abstract phonological units and the observed acoustic features. This intermediate level represents trajectories of articulators, formants or other parameters closely related to speech production. Some complex mapping is required between this underlying level and the observed acoustic features. The aim is for the trajectory model to enforce production-related constraints on possible acoustic realizations without requiring explicit extraction of articulatory or formant information. To gain the full benefit of such a model, it is important to incorporate a model of coarticulation in the underlying trajectories, so that different trajectories arise naturally for different sequences of speech sounds rather than requiring very many context-dependent models. For example, some approaches model a sequence of hidden targets which are then filtered to obtain a continuously evolving trajectory.

While a lot of further research will be needed, multiple-level models that capture some salient aspects of speech production would seem to be a promising line of investigation, which may lead to more powerful, constraining models than are provided by current HMM systems. In particular, these models should provide a meaningful way of capturing differences, both between talkers and within any one individual, due to effects such as stress or simply differences in speaking rate. Variations of this type can be expected to be much more systematic, and hence predictable, at the level of the speech production system. At this level it should therefore be much easier to adapt to changes than when simply linking acoustic variation directly to the phonological units, as is the case for current HMM systems. It should also be possible to move away from the rather artificial notion in current recognition (and synthesis) systems that speech can be represented as a sequence of contiguous but distinct phonetic segments. Many modern phonological theories view speech as being generated in terms of several different articulatory features operating at different and overlapping timescales, and it is easier to see how these might be accommodated in a model with an intermediate layer that is related to articulation.

A related issue concerns the choice of acoustic features upon which any recognition process must operate. Ideally the feature analysis should preserve all the perceptually relevant information in the speech signal. The typical long analysis window blurs highly transient events such as the bursts of stop consonants and rapid formant transitions, yet these portions of the signal convey some of the most important perceptual information. It ought to be advantageous to make more use of auditory models in ASR systems. As mentioned in Section 16.3.2 above, the human auditory system shows sensitivity to transitional information at a range of timescales, and automatic systems should be improved by the development of better methods for both extracting and modelling the relevant information.

The need to model information at different timescales has recently been addressed by research into extending HMMs to use multiple feature sets in parallel, with the associated probabilities being combined at some stage as part of the recognition process. These **multi-stream** methods potentially provide a way of incorporating many diverse information sources, including those obtained at different timescales.

The modelling approaches that have been mentioned above are just a few of the ideas that are currently being pursued as part of research aimed at improved speech modelling for ASR. More generally, there is also a growing interest in applying statistical formalisms that are used in other areas of pattern recognition, including methods that can be viewed as more powerful generalizations of the first-order HMMs that are still the most widely used model for ASR at present.

16.4 RELATIONSHIP BETWEEN SYNTHESIS AND RECOGNITION

Most speech research groups have for many years tended to specialize in one particular facet of speech processing, and it has been fairly unusual for the same research workers to be involved in both speech synthesis and speech recognition. In the past the actual techniques that have been used in the two areas have seemed to be almost completely unrelated. Yet it is evident that, for real advancement in both subjects, the predominant need is for knowledge about the structure of speech and its relationship to the underlying linguistic content of utterances. There has been much debate about the relationship between speech production and speech perception in humans, and similar issues apply to the development of automatic systems. However at the very least it ought to be beneficial to take more account of the constraints of production in speech recognition and to take more advantage of perceptual influences in speech synthesis.

In recent years there has been some convergence of the two fields, as automatic data-driven techniques have become widely adopted in synthesis as well as in recognition. At present, the most successful systems for both technologies use large inventories of acoustic segments and make minimal assumptions about the underlying structure of those segments. HMMs have been used to identify segments for use in concatenative synthesis (e.g. Donovan and Woodland, 1995). Of course an HMM is a generative model and can therefore be viewed as a synthesizer, but a rather crude one that generates an acoustic signal as a sequence of piecewise-stationary chunks. In the previous section we argued that ASR could be improved by using a more appropriate model of speech production, to capture the dynamic properties of speech in a better way by somehow modelling the human speech production mechanisms more closely. In fact this type of model could also fulfil the requirements that we advocated in Section 16.2 for speech synthesis, although the detail of how such a model would be used may well be different, depending on whether it is being used for synthesis or for recognition. For example, in synthesis it is necessary to generate a speech waveform, but for recognition it will probably be better to work directly from some analysis of this waveform.

16.5 AUTOMATIC SPEECH UNDERSTANDING

There is a growing demand for interactive spoken-language systems that involve a two-way dialogue between a person and a computer. In the future greater naturalness will be required both in the language that the human can use and in the responses generated by the system. Such naturalness is only likely to be achievable if the machine has a good model of the interaction and some ‘understanding’ of the information being communicated. The difficulties here are generally regarded as problems of artificial intelligence, but most language processing work in artificial intelligence has so far only considered textual forms of language. Although the achievements in this field are impressive, spoken language poses additional challenges. In particular, spontaneous speech can be vastly different from read speech: not only does the speech tend to be more casual and include hesitations, corrections and so on, but the use of language is very different when it is part of an interactive communication. Future conversational systems will need to model these effects, both for high-performance speech recognition and for natural-sounding speech generation. Another aspect of growing importance is a demand for spoken-language systems to be multilingual. A truly multilingual system needs to include an underlying representation of concepts, together with methods for relating those concepts to utterances in different languages in ways that exploit the commonality between different languages while also modelling the differences between them. Such a capability is probably necessary if major advances are to be achieved in the most challenging problem of spoken language translation (requiring recognition in one language and synthesis in another, with a translation stage in between the two).

Artificial intelligence methods will need to use information about the current speech communication task whether performing synthesis or recognition. In particular, knowledge about the subject matter is extremely important for producing and interpreting utterances in man-machine dialogue, and must include the effect of previous utterances on the expectations of what will follow. So far, the best spoken dialogue systems have been the result of a lot of hand-tuning specific to their application domain (such as air travel), so setting up a system for a new domain is time-consuming and labour-intensive. Good automatic methods are needed for training models for the domain semantics from appropriate existing material for the relevant domain. The processes that will be needed to interpret the phonetic and prosodic properties of speech signals as text or as concepts will have their counterparts in going from text or concept into speech, and both directions of processing need to be taken into account in dialogue design. Improvements in both synthesis and recognition technologies should come about with the development of better *models* of spoken language, capturing all levels in the relationship between abstract linguistic concepts and the generated acoustic signal.

CHAPTER 16 SUMMARY

- Although speech synthesis and recognition technology are now good enough to be useful in many applications, performance is still poor in comparison with that of humans. The problem is not in the computational power achievable with electronic technology. Large quantities of data are now available for training, but better models are needed to make the most effective use of these data.
- Data-driven methods have been applied to concatenative synthesis but have not yet been widely applied to synthesis by rule. A model-driven approach to synthesis offers the scope to capture co-articulation, and to include variability and flexibility in a way that is not possible with concatenative methods.
- The most difficult synthesis problems are in making the style of speech appropriate for the intended meaning. Development of artificial intelligence techniques will be necessary.
- Future developments in ASR will need to retain a data-driven approach to training, and a recognition framework that delays decisions until there is sufficient evidence and does not discard information too early. Improvements over current methods should be possible by incorporating a richer structure in the model. Promising developments include the use of multiple-layer frameworks to incorporate constraints of speech production and models that use parallel streams to capture information at differing timescales.
- Advanced systems for both synthesis and recognition need the same knowledge about speech and language, including an understanding capability. It should therefore be advantageous for the two applications to be studied together.

CHAPTER 16 EXERCISES

E16.1 How might a good acoustic model for speech recognition also be useful for high-quality synthesis, and what attributes would the model need? Explain any potential problems with using the same model for both applications.

E16.2 Why will automatic understanding of messages be necessary for really high performance in speech synthesis and recognition in the future?

CHAPTER 17

Further Reading

The subject of speech synthesis and recognition is so large that a book of this size could not hope to provide more than an overview. Most of the material presented here has made great use of published material from various sources. There are now several significant textbooks available, both on general speech processing and on more specific topics. Further information, including descriptions of the most recent developments, can be found in specialist research journals and conference proceedings. This chapter aims to give sufficient information as a starting point to enable the reader to trace all the important material needed for more specialized study on any of the facets of speech processing presented here.

As a guide to future work, as well as giving assistance in tracing past work, it is useful to search the literature for other papers by authors mentioned in this chapter, gradually increasing one's knowledge of significant people and research groups by adding names of regular co-authors and other workers from the same laboratories.

17.1 BOOKS

These days nearly all speech processing is digital, and a grasp of basic concepts in digital signal processing is necessary in order to obtain a good understanding of the processes involved in generating and analysing speech signals. Good general introductory textbooks on digital signal processing include Lynn and Fuerst (second edition, 1998) and Lyons (1997). One of the most important textbooks on digital processing of speech is Rabiner and Schafer (1978), but even in its early chapters this book requires the reader to be really at home with mathematical notation and manipulation. For the less mathematically minded, Harrington and Cassidy (1999) or Rosen and Howell (second edition, 2001) provide a much easier introduction, though they are less comprehensive than Rabiner and Schafer.

There are now many books available that provide a variety of perspectives on a similar range of topics to the ones presented in the current book, but with more detailed coverage than has been possible here. Gold and Morgan (2000) is an up-to-date introduction to both speech and audio processing, and Huang *et al.* (2001) provides a modern, comprehensive introduction to all aspects of spoken language processing. Other useful recent sources include O'Shaughnessy (second edition, 2000) and Furui (second edition, 2001), both of which include a lot of references to recent research findings as well as more introductory material. Deller *et al.* (1993, reissued in 2000) is a substantial textbook with a signal processing emphasis. Rabiner and Juang (1993) concentrates on speech recognition, but also includes background about speech signals and explanations of several speech analysis methods.

Other books cover more specialized subjects, and we will include references to some of these under headings for the individual chapters. There are also books of collected research papers containing many of the classic original publications in various subject areas. Among the most important are Flanagan and Rabiner (1972), Dixon and Martin (1979), Schafer and Markel (1979) and Waibel and Lee (1990).

17.2 JOURNALS

For those wishing to keep up with the latest developments in speech synthesis and recognition, there are a few journals that cover most of the important material. For papers with an emphasis on engineering and signal processing, one popular location is the bimonthly *IEEE Transactions on Speech and Audio Processing* (and many important references can also be found in its predecessor publications, the *IEEE Transactions on Acoustics, Speech and Signal Processing* and the earlier *IEEE Transactions on Audio and Electro acoustics*). There are also other IEEE publications that sometimes contain papers on speech subjects. In particular, the bimonthly *IEEE Signal Processing Magazine* has tutorial-style papers which aim to be both reliable technically and readable for a non-specialist audience, while the *Proceedings of the IEEE* is a long-established source of more in-depth tutorials and reviews.

The *Journal of the Acoustical Society of America* has a similar status to the *IEEE Transactions* but is devoted to all aspects of the study of sound, with typically around 15–25% of the papers being on speech subjects. This journal specializes in papers with a more acoustical bias, including psycho-acoustics, physiological acoustics, speech production and perception.

There are two journals that concentrate on spoken language processing by both humans and machines, with an emphasis on the interdisciplinary nature of the subject. *Speech Communication* (North Holland, originally quarterly but now monthly) was first published in 1982, is devoted entirely to speech and covers all aspects. *Computer Speech and Language* (Academic Press, quarterly) dates from 1986, tends to be somewhat more computer-oriented than *Speech Communication* and also includes some papers on language processing that do not necessarily involve speech. There are many other journals specializing in such subjects as linguistics, psychology, etc. which contain some papers on speech. They can easily be found by studying the reference lists given in other papers on relevant subjects.

17.3 CONFERENCES AND WORKSHOPS

The most recent research developments are generally reported in the proceedings of conferences and workshops (often appearing in more detail in a journal a year or so later). Undoubtedly the most important and long-established single annual conference covering speech processing is the *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* usually held in the U.S. but occasionally in other countries. Only about 25% of its contributions are on speech, but the total size of the conference is such that there are normally at least two parallel speech sessions running throughout a four-day period, providing roughly 200–300 speech papers for the proceedings. Although there is quite a high proportion of less significant work, many of the most notable new achievements in speech processing are first presented at ICASSP, and the ICASSP reference has often been the only published source of such work for several years.

There are two biennial conferences that run in alternate years and are both devoted entirely to all aspects of spoken language processing by both humans and machines. The *European Conference on Speech Communication and Technology (EUROSPEECH)* has been held at European venues in every odd-numbered year since 1989. In the even-numbered years, starting in 1990, the *International Conference on Spoken Language Processing (ICSLP)* takes place at a location outside Europe. In recent years both of these conferences have grown very large, with multiple oral and poster sessions running in parallel over four days and conference proceedings containing several hundred papers. The sheer size of the conferences is such that the quality and significance of the papers is rather variable, but they provide a very good way of keeping up to date with research trends in all aspects of speech processing with contributions from many different disciplines.

The IEEE runs a series of biennial workshops (which began in 1989) that focus on research in speech recognition and understanding. Also useful, especially for descriptions and performance of complete research systems for particular recognition tasks, are the proceedings of the ARPA/NIST workshops that are held in the U.S. as part of the ongoing performance evaluations co-ordinated by NIST.

Of course there are other conferences that contain material of relevance to speech processing. For example, the twice-yearly meetings of the Acoustical Society of America cover the same subject areas as the Society's journal, but unfortunately the papers are only published in short abstract form, and so have little value for reference purposes. There are also other conferences and workshops on more specific topics, but it is more difficult to give any general guidance because they tend to be either irregular or organized as special one-off events.

17.4 THE INTERNET

In the past few years, a large amount of information has become available electronically through the Internet. The World Wide Web now provides a great variety of information relevant to speech synthesis and recognition, including notes for lecture courses, details of work conducted by different research groups and descriptions of current speech products from manufacturers. Many sites include audio examples, and there is an increasing amount of freely available software that can be downloaded. Because the very nature of the Web is such that it is constantly changing, any list of Web sites would be out of date almost immediately. Therefore, rather than attempting to list all relevant Web sites here, we simply give the site for this book (http://www.speechbook.net). This site includes links to a variety of Web sites that are of relevance to the subjects covered in the book, and the aim is to update it periodically. In addition, using a good search engine and specifying suitable keywords for any topic of interest will usually provide a useful set of Web addresses as a starting point.

17.5 READING FOR INDIVIDUAL CHAPTERS

Chapter 1

Although written for a completely different purpose (for teaching British English as a foreign language), Roach (second edition, 1991) provides a very readable introduction to phonetics and phonology, illustrated with examples from the “Received Pronunciation” variety of English. O’Connor (second edition, 1991) and Ladefoged (third edition, 1993) are more general books on phonetics. O’Connor contains a useful bibliography on phonetics and linguistics, and Ladefoged includes a glossary of phonetics terms. Another good introduction to phonetics and phonology is provided in the textbook by Clark and Yallop (second edition, 1995). A comprehensive set of English word pronunciations using the phoneme set that we have adopted in this book can be found in the pronunciation dictionary by Wells (second edition, 2000).

Chapter 2

Fant (1960) is an important book on the acoustic theory of speech production. Flanagan (1972) is a classic textbook covering a wide range of aspects of speech, including a general view of the mechanism of speech production and its mathematical modelling (chapters 2 and 3). Linggard (1985) contains an excellent chapter on the early history of acoustic models of speech production, besides dealing with many other aspects of synthesis, largely from a signal processing point of view. Ishizaka and Flanagan (1972) and Titze (1973, 1974) describe mathematical models of the vocal folds. J.N.Holmes (1973, 1976) comments on the effects of the voiced excitation waveform in speech generation. Flanagan *et al.* (1975) describe a computer model for articulatory synthesis. Klatt (1980) gives a description of a widely used cascade/parallel formant synthesizer. J.N.Holmes (1983) presents arguments in favour of parallel formant generators for practical synthesis, and explains in some detail the design of his parallel formant synthesizer for synthesis in the frequency range up to about 4 kHz. An extension to include frequencies up to around 8 kHz is described by W.J.Holmes *et al.* (1990).

Chapter 3

Chapter 4 of Flanagan (1972) covers much of the acoustic aspect of hearing in fair detail. Moore (fourth edition, 1997) is an excellent introduction to the psychology of hearing, and also includes material on auditory physiology and the physics of sound. Lyon (1988) describes a model of the cochlea in considerable detail. A good general overview of auditory models can be found in a theme issue of the *Journal of Phonetics* on “Representation of Speech in the Auditory Periphery”, edited by Greenberg (1988). This issue of the journal contains several papers describing different auditory models, including Seneff (1988) and Ghitza (1988). A more recent description of Ghitza’s model can be found in Ghitza (1992).

Chapter 4

There are several useful review papers on speech coding. For example, Flanagan *et al.* (1979) and Holmes (1982) contain numerous references to early work on various types of coder. More recent reviews of speech coding in general are given by Spanias (1994) and by Gersho (1994), and methods for low bit-rate coding in particular have been reviewed by Jaskie and Fette (1992). Barnwell *et al.* (1995) is a textbook that explains the principles of a variety of speech coding algorithms and includes software to allow practical experimentation with the different algorithms.

Markel and Gray (1976) is a classic book on linear predictive coding. The MELP coder adopted for the new U.S. Government standard at 2,400 bits/s is described in Supplee *et al.* (1997), and is based on the method developed by McCree and Barnwell (1995). Multipulse LPC was developed by Atal and Remde (1982), and an early paper on CELP coding is by Schroeder and Atal (1985). Further developments have been described in numerous other papers in more recent ICASSP proceedings and elsewhere. Work on sinusoidal transform coding is described in McAulay and Quatieri (1992), and in several other papers by the same authors. The MBE coding method is described in Griffin and Lim (1988). Information about the use of vector quantization in speech coding can be found in the tutorial paper by Makhoul *et al.* (1985) and in the book by Gersho and Gray (1991).

Hess (1983) produced a very comprehensive review of methods for measuring fundamental frequency in speech, such as are needed in vocoders. Although it only covers the period up until 1983 and so is inevitably now somewhat dated, for the period covered this book includes an outstanding bibliography, which spreads some way beyond its particular specialist topic.

The use of the DRT for speech intelligibility evaluation is described in Voiers (1977). Both subjective and objective quality measures are explained in the book by Quackenbush *et al.* (1988). The PSQM, which was developed more recently, is described by Beerends and Stemerdink (1994).

Chapter 5

Simple waveform concatenation of words is described in Trupp (1970), and a formant-coded technique appears in Rabiner *et al.* (1971). Harris (1953) wrote an early paper about concatenation of sub-word units (as sections of waveform). One of the earliest diphone methods, using formant coding, was described by Estes *et al.* (1964). Olive (1977) presented a method using LPC coded dyads and Browman (1980) used demisyllables. A more recent system that is based on concatenation of LPC diphones is described by Olive *et al.* (1998). Moulines and Charpentier (1990) is the classic reference for the PSOLA technique for waveform synthesis. This paper includes a detailed discussion of TD-PSOLA, FD-PSOLA and LP-PSOLA. MBR-PSOLA is described in Dutoit and Leich (1993). Chapters 7–10 of the book by Dutoit (1997) discuss sub-word concatenative synthesis methods in general as well as describing both LPC synthesis techniques and different variants of PSOLA.

Chapter 6

Liberman *et al*. (1959) is an important early paper on acoustic/phonetic rules, and methods for phonetic synthesis by rule have been reviewed by Klatt (1987). The book by Allen *et al*. (1987) includes a chapter describing the method used for the phonetic-synthesis component of the MITalk text-to-speech system. The original HMS table-driven formant synthesis-by-rule method is described in J.N.Holmes *et al*. (1964). This paper contains the first complete description of computer-implemented formant synthesis rules for all the phonemes of a language. Two papers describing work on automatically estimating the parameters of HMS tables are Bridle and Rails (1985) and W.J.Holmes and Pearce (1990).

Chapter 7

Klatt (1987) is a comprehensive review of methods for conversion from text to speech for English, as of 1987, including an extensive bibliography. Another useful review paper is the one by Allen (1992). The book by Dutoit (1997) describes more recent developments in TTS in considerable detail, with an emphasis on the use of concatenative techniques for the speech output components. Allen *et al*. (1987) describes the MITalk TTS system for American English, including the method of morph decomposition that was pioneered in this system. The book edited by Sproat (1998) describes TTS work carried out at Bell Laboratories in the U.S., with an emphasis on multilingual synthesis. A special issue of the *BT Technology Journal* (Vol. 14, No. 1, January 1996), devoted to speech synthesis, contains a number of papers authored by researchers at BT Laboratories in the U.K. In particular, the design of the BT “Laureate” TTS system is described in the paper by Page and Breen (1996), and two papers by Edgington *et al*. (1996a, 1996b) provide good overviews of techniques for the analysis and synthesis components of TTS conversion. Models for synthesizing English intonation include those of Fujisaki and Ohno (1995) and of Pierrehumbert (1981), while the TOBI prosodic labelling scheme is described in Silverman *et al*. (1992). Young and Fallside (1979) is an early publication on synthesis from concept.

Chapter 8

Early rule-based methods for ASR are reviewed in Hyde (1972). A big improvement in whole-word pattern matching came when dynamic programming was first applied to the time-alignment problem. Vintsyuk (1968, 1971) and Velichko and Zagoruyko (1970), all from the Soviet Union, published some of the earliest work in this area. A widely referenced paper on the use of DTW for template matching is by Sakoe and Chiba (1978), while Sakoe (1979) and Bridle *et al*. (1983) represent work on connected-word recognition. Silverman and Morgan (1990) is a useful review paper on the use of dynamic programming for speech recognition. This paper describes the history and general operation of the technique, and discusses its application both to template matching and to the statistical speech recognition techniques described in Chapter 9.

Chapter 9

This chapter requires some understanding of statistics and probability theory. There are several textbooks available, including Helstrom (second edition, 1991) and Papoulis (third edition, 1991). The theory on which hidden Markov models are based was originally described in a number of papers with Baum as one of the authors. An example is Baum (1972), but this paper requires the reader to have a high degree of mathematical ability. Liporace (1982) has published a mathematical treatment of the modelling of multivariate continuous distributions, including normal distributions. Juang (1985) extended Liporace's analysis to include sums of normal distributions. The use of tied-mixture, or semi-continuous, distributions is explained in Bellegarda and Nahamoo (1990) and in Huang and Jack (1989). The segmental *K*-means algorithm is described in Juang and Rabiner (1990).

The classic reference for the general EM algorithm is Dempster *et al.* (1977), although this paper is mathematically demanding. An easier-to-read explanation of the EM technique can be found in the paper by Moon (1996).

Several of the papers describing HMM methods for limited vocabulary recognition of words were written by researchers at AT&T Bell Laboratories, such as Levinson *et al.* (1983) and Rabiner (1989). There are now several books that include chapters on HMM theory and its application to speech recognition. For example, the book by Rabiner and Juang (1993) contains a fairly detailed description of HMM theory as well as discussing practical aspects of developing systems for recognizing both small and large vocabularies. Knill and Young (1997) is a book chapter that provides an overview of HMMs, and also includes material that is relevant to Chapters 11 and 12. The book by Jelinek (1997) includes chapters covering the mathematical foundations of the HMM techniques, as well as a lot of more advanced material that is relevant to Chapter 12.

Chapter 10

A tutorial on front-end processing for ASR is given by Picone (1993), and books such as Rabiner and Juang (1993) and Deller *et al.* (1993, 2000) also contain a lot of information about speech analysis and acoustic representations for ASR. The very widely cited reference for the use of MFCCs as features for ASR is Davis and Mermelstein (1980). PLP analysis was proposed by Hermansky (1990), and Furui (1986) suggested using time derivative features for ASR.

Chapter 11

Junqua and Haton (1996) is a book devoted to robust ASR, including techniques for dealing with variation across speakers and environments. A survey of techniques for recognition in noisy environments is given in Gong (1995), and there are several useful papers in a special issue of *Speech Communication* on "Robust Speech Recognition", edited by Junqua and Haton (1998). RASTA processing is discussed in Hermansky and Morgan (1994). HMM decomposition was proposed by Varga and Moore (1990), and the PMC method is described in Gales and Young (1995) and in various other papers by the same authors. These two methods are developments of earlier work by Klatt (1976) on noise masking. Cooke *et al.* (2001) have described the use of missing-data techniques in ASR.

One paper describing a technique for VTLN is Lee and Rose (1998). MAP estimation of HMM parameters is presented by Gauvain and Lee (1992, 1994), and adaptation using MLLR is described in Leggetter and Woodland (1995) and in Gales and Woodland (1996). SAT has been described by Anastasakos *et al.* (1996), and further developments of both MLLR and SAT are discussed in Gales (1998).

Both the MMI and corrective training methods were initially proposed by researchers at IBM and are described in papers by Bahl *et al.* (1986) for MMI training and by Bahl *et al.* (1988) for corrective training. More recent work on MMI is described in Normandin *et al.* (1994). Papers describing MCE and GPD methods include Juang and Katagiri (1992) and Juang *et al.* (1997).

Systems for keyword detection include the ones described by Wilpon *et al.* (1990) and by Rose (1995).

Chapter 12

A good review of the 1970s ARPA programme is given in Klatt (1977). Early work on extending the principles of HMMs to large-vocabulary speech recognition is explained in Baker (1975) and in Jelinek (1976). The IBM Tangora dictation system is described in Jelinek (1985). Young (1996) gives an excellent overview of the successful Cambridge University research system for large-vocabulary speech recognition, as it was in 1996, and includes a large number of references.

The book by Jelinek (1997) provides information about different aspects of large-vocabulary speech recognition, but is particularly comprehensive in its coverage of the language-modelling component. Methods for coping with the data-sparsity problem in language modelling have been discussed in a number of papers by Ney and co-authors, including for example Ney *et al.* (1994) and Ney *et al.* (1997). More recent work on smoothing techniques for language models is described in Chen and Goodman (1999). A tree-based language model is presented in Bahl *et al.* (1989) and some work on incorporating grammatical structure has been described by Chelba and Jelinek (1999), for example. The use of a cache in language modelling was proposed by Kuhn and De Mori (1990, 1991), and work on using word triggers includes that of Lau *et al.* (1993).

The September 1999 issue of the *IEEE Signal Processing Magazine* includes two articles, Ney and Ortmanns (1999) and Deshmukh *et al.* (1999), describing different search techniques.

The book by Jurafsky and Martin (2000) covers natural language processing, with an emphasis on integrating computational linguistics and speech recognition. This book is a valuable source for all aspects of spoken-language understanding, including language modelling, part-of-speech tagging and dialogue modelling.

Young and Chase (1998) is a review paper that describes and explains the ARPA programme of speech recognition evaluations. Descriptions and results from individual evaluations can be found in the proceedings of the ARPA (or DARPA) workshops which take place following each formal evaluation, and also in the proceedings of ICASSP and other speech conferences.

The book by Rayner *et al*. (2000) describes the application of techniques for spoken-language understanding to a system for spoken-language translation.

Chapter 13

Good general introductions to ANNs can be found in the books by Bishop (1995) and Haykin (second edition, 1999). The application of ANNs to a variety of speech processing tasks is covered in the book by D.P.Morgan and Scofield (1991). Two papers by N.Morgan and Bourlard (1995a, 1995b) explain hybrid HMM/ANN methods; the first paper is introductory in nature, while the second is more detailed.

Chapter 14

One useful review of speaker recognition is provided by Furui (1994). Over the years a number of reviews of speaker recognition technology have also appeared in various IEEE publications. Early work has been presented by Atal (1976) and by Rosenberg (1976), and somewhat later work is included in Doddington (1985). More recently, Naik (1990) has concentrated on speaker verification and Gish and Schmidt (1994) have described methods for text-independent speaker recognition, while Campbell (1997) is a comprehensive tutorial on speaker recognition, including a lot of detail about acoustic analysis.

The YOHO corpus for evaluating text-dependent speaker recognition systems is described in Campbell (1995). Doddington *et al*. (2000) provides a good overview of the motivations, design and results for the 1998 NIST speaker recognition evaluation; the 1999 evaluation has been described by Martin and Przybocki (2000). Martin *et al*. (1997) have described a method for displaying detection errors (which they call a detection error trade-off curve). This method has been adopted by NIST for displaying the performance of verification systems.

Overviews of language identification are given in Muthusamy *et al*. (1994) and in Zissman (1996). The use of phone-based recognition techniques for speaker, language and gender identification is described in Lamel and Gauvain (1995).

Chapter 15

Rodman (1999) is an introductory textbook (with broadly similar scope to this one but very different in style and emphasis), which contains three chapters about applications of speech technology as well as some discussion about the different factors affecting speech recognition performance. Gibbon *et al*. (1997) explains how to specify requirements for speech technology applications and match these requirements to technology capabilities, as well as being a more general reference source for standards and resources in speech processing. Junqua (2000) is a recent book on the robust application of speech recognition technology. This book includes examples of applications as well as detailed discussion of the difficulties involved and the different techniques that are available for achieving robustness. Some of the material is also relevant to Chapters 10, 11 and 12.

Papers describing telecommunications applications of speech technology include Levinson *et al*. (1993) and Rabiner (1994, 1997). Another useful source of information is a special issue of the *Speech Communication* journal devoted to papers from the "3rd IEEE Workshop on Interactive Voice Technology for Telecommunications Applications", edited by Spiegel and Kamm (1997). This issue contains a paper by Gorin *et al*. describing the AT&T "How may I help you?" system, as well as several other papers discussing applications of both speech synthesis and speech recognition technology in telecommunications services. The NTT ANSER system is described in Nakatsu (1990). Weinstein (1991) has written a survey of military applications for speech technology.

"Speech Recognition Update", published by TMA Associates, is a monthly newsletter covering company developments and products using speech recognition, speaker verification, text-to-speech synthesis and related language technologies. The book by Weinschenk and Barker (2000) is aimed at application developers and explains the principles and processes involved in designing effective speech interfaces for different types of applications.

Chapter 16

It is more difficult to give references for this chapter because it is looking to the future. Recent issues of journals and conference proceedings should give a good indication of current trends.

A review of past progress and discussion of prospects for the future of speech processing research can be found in Juang (1998). Some general predictions and discussion of research issues are given in Cole *et al*. (1995) and in Chapter 10 of Furui (second edition, 2001). For discussion of future research areas in speech synthesis, see Sproat *et al*. (1998). Bourlard *et al*. (1996) focus on ASR, emphasizing the need to try new approaches even though in the short term they may result in "increasing speech recognition error rates".

Wouters and Macon (2001) have described a technique for smoothing formant transitions across segment boundaries when using synthesis based on a sinusoidal coding representation.

A review of segment modelling methods for ASR is given by Ostendorf *et al.* (1996). Zlokarnik (1993) estimated articulatory features for incorporation into a recognition system. Some of the issues for using formant features for ASR are discussed in Hunt (1987) and in Holmes *et al.* (1997). Richards and Bridle (1999) and Deng and Ma (2000) have .described work on developing multiple-level approaches to speech modelling. Deng (1998) discusses ideas for incorporating a phonological model of overlapping features. Recent conference proceedings include several papers describing the use of multi-stream models, of which one example is the paper by Hagen and Bourlard (2000).

Also of relevance to the points discussed in this chapter are experiments by Lippmann (1997) comparing human recognition performance with machine recognition performance. A review paper by Pitton *et al.* (1996) discusses the advantages and limitations of different time-frequency representations of speech, with an emphasis on auditory modelling. This paper includes some quite advanced material that is of relevance to earlier chapters, especially Chapters 2, 3 and 10.

REFERENCES

Allen, J., 1992, Overview of text-to-speech systems. In *Advances in Speech Signal Processing,* edited by Furui, S. and Sondhi, M.M. (New York: Marcel Dekker), pp. 741–790.

Allen, J., Hunnicutt, M.S. and Klatt, D., 1987, *From Text to Speech: the MITalk System* (Cambridge: Cambridge University Press).

Anastasakos, T., McDonough, J., Schwartz, R. and Makhoul, J., 1996, A compact model for speaker-adaptive training. In *Proceedings of the International Conference on Spoken Language Processing,* Philadelphia, pp. 1137–1140.

Atal, B.S., 1976, Automatic recognition of speakers from their voices. *Proceedings of the IEEE,* **64**, pp. 460–475.

Atal, B.S. and Remde, J.R., 1982, A new model of LPC excitation for producing natural-sounding speech at low bit rates. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Paris, pp. 614–617.

Bahl, L.R., Brown, P.P., deSouza, P.V. and Mercer, R.L., 1986, Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Tokyo, pp. 49–52.

Bahl, L.R., Brown, P.P., deSouza, P.V. and Mercer, R.L., 1988, A new algorithm for the estimation of hidden Markov model parameters. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* New York, pp. 493–496.

Bahl, L.R., Brown, P.P., deSouza, P.V. and Mercer, R.L., 1989, A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **37**, pp. 1001–1008.

Baker, J.K., 1975, The DRAGON system—an overview. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **ASSP-23,** pp. 24–29.

Barnwell, T.P., Nayebi, K. and Richardson, C.H., 1995, *Speech Coding: a Computer Laboratory Textbook* (New York: John Wiley and Sons).

Baum, L.E., 1972, An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities,* **III,** pp. 1–8.

Beerends, J.G. and Stemerdink, J.A., 1994, A perceptual speech-quality measure based on a psychoacoustic sound representation. *Journal of the Audio Engineering Society,* **42**, pp. 115–123.

Bellegarda, J.R. and Nahamoo, D., 1990, Tied mixture continuous parameter modeling for speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **38**, pp. 2033–2045.

Bellman, R., 1957, *Dynamic Programming* (Princeton, NJ: Princeton University Press).

Bishop, C.M., 1995, *Neural Networks for Pattern Recognition* (Oxford: Oxford University Press).

Bourlard, H., Hermansky, H. and Morgan, N., 1996, Towards increasing speech recognition error rates. *Speech Communication,* **18**, pp. 205–231.

Bridle, J.S. and Rails, M., 1985, An approach to speech recognition using synthesis-by-rule. In *Computer Speech Processing,* edited by Fallside, F. and Woods, W.A. (London: Prentice-Hall International), pp. 277–292.

Bridle, J.S., Brown, M.D. and Chamberlain, R.M., 1983, Continuous connected word recognition using whole word templates. *The Radio and Electronic Engineer,* **53**, pp. 167–177.

Browman, C.P., 1980, Rules for demisyllable synthesis using Lingua, a language interpreter. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Denver, pp. 561–564.

Campbell, J., 1995, Testing with the YOHO CD-ROM voice verification corpus. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Detroit, pp. 341–344.

Campbell, J., 1997, Speaker recognition: a tutorial. *Proceedings of the IEEE,* **85**, pp. 1437–1462.

Chelba, C. and Jelinek, F., 1999, Recognition performance of a structured language model. In *Proceedings of the European Conference on Speech Communication and Technology,* Budapest, pp. 1567–1570.

Chen, S.F. and Goodman, J., 1999, An empirical study of smoothing techniques for language modeling. *Computer Speech and Language,* pp. 359–394.

Clark, J. and Yallop, C., 1995, *An Introduction to Phonetics and Phonology,* Second Edition (Oxford: Blackwell Publishers Ltd.).

Cole, R., Hirschman, L., Atlas, L., Beckman, M., Biermann, A., Bush, M., Clements, M., Cohen, J., Garcia, O., Hanson, B., Hermansky, H., Levinson, S., McKeown, K., Morgan, N., Novick, D.G., Ostendorf, M., Oviatt, S., Price, P., Silverman, H., Spitz, J., Waibel, A., Weinstein, C., Zahorian, S. and Zue, V., 1995, The challenge of spoken language systems: research directions for the nineties. *IEEE Transactions on Speech and Audio Processing,* **3**, pp. 1–20.

Cooke, M., Green, P., Josifovski, L. and Vizinho, A., 2001, Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication,* **34**, pp. 267–285.

Davis, S. and Mermelstein, P., 1980, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **28,** pp. 357–366.

Deller, J.R., Proakis, J.G. and Hansen, J.H.L., 1993, *Discrete-Time Processing of Speech Signals* (New York: Macmillan).

Deller, J.R., Proakis, J.G. and Hansen, J.H.L., 2000, *Discrete-Time Processing of Speech Signals (an IEEE Press Classic Reissue)* (Piscataway, NJ: IEEE Press).

Dempster, A.P., Laird, N.M. and Rubin, D.B., 1977, Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society,* Series B, **39**, pp. 1–38.

Deng, L., 1998, A dynamic, feature-based approach to the interface between phonology and phonetics for speech modeling and recognition. *Speech Communication,* **24**, pp. 299–323.

Deng, L. and Ma, J., 2000, Spontaneous speech recognition using a statistical coarticulatory model for the vocal-tract-resonance dynamics. *Journal of the Acoustical Society of America,* **108,** pp. 1–13.

Deshmukh, N., Ganapathiraju, A. and Picone, J., 1999, Hierachical search. *IEEE Signal Processing Magazine,* **16,** no. 5, pp. 84–107.

Dixon, N.R. and Martin, T.B. (eds.), 1979, *Automatic Speech & Speaker Recognition* (New York: IEEE Press).

Doddington, G., 1985, Speaker recognition: identifying people by their voices. *Proceedings of the IEEE,* **64,** pp. 460–475.

Doddington, G., Przybocki, M.A., Martin, A.R. and Reynolds, D.A., 2000, The NIST speaker recognition evaluation—Overview, methodology, systems, results, perspective. *Speech Communication,* **31,** pp. 225–254.

Donovan, R.E. and Woodland, P.C., 1995, Automatic speech synthesiser parameter estimation using HMMs. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing,* Detroit, pp. 640–643.

Dudley, H., 1939, Remaking speech. *Journal of the Acoustical Society of America,* **11,** pp. 169–177.

Dutoit, T., 1997, *An Introduction to Text-to-speech Synthesis* (Dordrecht: Kluwer).

Dutoit, T. and Leich, H., 1993, MBR-PSOLA: Text-to-speech synthesis based on an MBE re-synthesis of the segments database. *Speech Communication,* **13,** pp. 435–440.

Edgington, M., Lowry, A., Jackson, P., Breen, A.P. and Minnis, S., 1996a, Overview of current text-to-speech techniques: part I—text and linguistic analysis. *BT Technology Journal,* **14,** pp. 68–83.

Edgington, M., Lowry, A., Jackson, P., Breen, A.P. and Minnis, S., 1996b, Overview of current text-to-speech techniques: part II—prosody and speech generation. *BT Technology Journal,* **14,** pp. 84–99.

Estes, S.E., Kerby, H.R., Maxey, H.D. and Walker, R.M., 1964, Speech synthesis from stored data. *IBM Journal of Research and Development,* **8,** pp. 2–12.

Fant, G., 1960, *The Acoustic Theory of Speech Production* (The Hague: Mouton and Co.).

Fidell, S., Horonjeff, R., Teffeteller, S. and Green, D.M., 1983, Effective masking bandwidths at low frequencies. *Journal of the Acoustical Society of America,* **73,** pp. 628–638.

Fiscus, J.G., Fisher, W.M., Martin, A.F., Przybocki, M.A. and Pallett, D.S., 2000, 2000 NIST evaluation of conversational speech recognition over the telephone: English and Mandarin performance results. In *Proceedings of the 2000 Speech Transcription Workshop,* University of Maryland (available from NIST, http://www.nist.gov).

Flanagan, J.L., 1972, *Speech Analysis, Synthesis and Perception,* Second Edition (Berlin: Springer-Verlag).

Flanagan, J.L. and Rabiner, L.R. (eds.), 1972, *Speech Synthesis* (Stroudsburg: Dowden, Hutchinson and Ross).

Flanagan, J.L., Ishizaka, K. and Shipley, K.L., 1975, Synthesis of speech from a dynamic model of the vocal cords and vocal tract. *Bell Systems Technical Journal,* **54,** pp. 485–506.

Flanagan, J.L., Schroeder, M.R., Atal, B.S., Crochiere, R.E., Jayant, N.S. and Tribolet, J.M., 1979, Speech coding. *IEEE Transactions on Communications,* **COM-27,** pp. 710–736.

Fletcher, H., 1940, Auditory Patterns. *Reviews of Modern Physics,* **12,** pp. 47–65.

Fujisaki, H. and Ohno, S., 1995, Analysis and modeling of fundamental frequency contours of English utterances. In *Proceedings of the European Conference on Speech Communication and Technology,* Madrid, pp. 985–988.

Furui, S., 1986, Speaker independent isolated word recognizer using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **34,** pp. 52–59.

Furui, S., 1994, An overview of speaker recognition technology. In *Proceedings of ESCA Workshop on Automatic Speaker Recognition, Identification and Verification,* Martigny, pp. 1–9.

Furui, S., 2001, *Digital Speech Processing, Synthesis and Recognition,* Second Edition (New York: Marcel Dekker).

Gales, M.J.F., 1998, Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language,* **12,** pp. 75–98.

Gales, M.J.F. and Woodland, P.C., 1996, Mean and variance adaptation within the MLLR framework. *Computer Speech and Language,* **10,** pp. 249–264.

Gales, M.J.F. and Young, S.J., 1995, Robust speech recognition in additive and convolutional noise using parallel model combination. *Computer Speech and Language,* **9,** pp. 289–307.

Gauvain, J.-L. and Lee, C.-H., 1992, Bayesian learning for hidden Markov model with Gaussian mixture state observation densities. *Speech Communication,* **11,** pp. 205–213.

Gauvain, J.-L. and Lee, C.-H., 1994, Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing,* **2,** pp. 291–298.

Gersho, A., 1994, Advances in speech and audio compression. *Proceedings of the IEEE,* **82,** pp. 900–918.

Gersho, A. and Gray, R.M., 1991, *Vector Quantization and Signal Compression* (Norwell, MA: Kluwer).

Ghitza, O., 1988, Temporal non-place information in the auditory-nerve firing patterns as a front-end for speech recognition in a noisy environment. *Journal of Phonetics,* **16,** pp. 109–124.

Ghitza, O., 1992, Auditory nerve representation as a basis for speech processing. In *Advances in Speech Signal Processing,* edited by Furui, S. and Sondhi, M.M. (New York: Marcel Dekker), pp. 453–485.

Gibbon, D., Moore, R. and Winski, R. (eds.), 1997, *Handbook of Standards and Resources for Spoken Language Systems* (Berlin: Mouton de Gruyter).

Gish, H. and Schmidt, M., 1994, Text-independent speaker identification. *IEEE Signal Processing Magazine,* **11,** no. 4, pp. 18–32.

Gold, B. and Morgan, N., 2000, *Speech and Audio Processing* (New York: John Wiley and Sons).

Gong, Y., 1995, Speech recognition in noisy environments: A survey. *Speech Communication,* **16,** pp. 261–291.

Gorin, A.L., Riccardi, G. and Wright, J.H., 1997, How may I help you? *Speech Communication,* **23,** pp. 113–127.

Greenberg, S., 1988 (ed.), Representation of Speech in the Auditory Periphery. Theme issue of the *Journal of Phonetics,* **16,** pp. 1–151.

Griffin, D.W. and Lim, J.S., 1988, Multi-band excitation vocoder, *IEEE Transactions on Acoustics, Speech and Signal Processing,* **36,** pp. 1223–1235.

Hagen, A. and Bourlard, H., 2000, Using multiple time scales in the framework of multi-stream speech recognition. In *Proceedings of the International Conference on Spoken Language Processing,* Beijing, pp. 349–352.

Harrington, J. and Cassidy, S., 1999, *Techniques in Speech Acoustics* (Dordrecht: Kluwer).

Harris, C.M., 1953, A study of the building blocks of speech. *Journal of the Acoustical Society of America,* **25,** pp. 962–969.

Haykin, S., 1999, *Neural Networks: a Comprehensive Foundation,* Second Edition (Upper Saddle River, NJ: Prentice-Hall).

Helstrom, C.W., 1991, *Probability and Stochastic Processes for Engineers,* Second Edition (New York: Macmillan).

Hermansky, H., 1990, Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America,* **87,** pp. 1738–1752.

Hermansky, H. and Morgan, N., 1994, RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing,* **2,** pp. 578–589.

Hess, W., 1983, *Pitch Determination of Speech Signals* (Berlin: Springer-Verlag).

Holmes, J.N., 1973, The influence of glottal waveform on the naturalness of speech from a parallel formant synthesizer. *IEEE Transactions on Audio and Electroacoustics,* **AU-21,** pp. 298–305.

Holmes, J.N., 1976, Formant excitation before and after glottal closure. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Philadelphia, pp. 39–42.

Holmes, J.N., 1982, A survey of methods of digitally encoding speech signals. *The Radio and Electronic Engineer,* **52,** pp. 267–276.

Holmes, J.N., 1983, Formant synthesizers: cascade or parallel? *Speech Communication,* **2,** pp. 251–273.

Holmes, J.N., Mattingly, I.G. and Shearme, J.N., 1964, Speech synthesis by rule. *Language and Speech,* **7,** pp. 127–143.

Holmes, J.N., Holmes, W.J and Garner, P.N., 1997, Using formant frequencies in speech recognition. In *Proceedings of the European Conference on Speech Communication and Technology,* Rhodes, pp. 2083–2086.

Holmes, W.J. and Pearce, D.J.B., 1990, Automatic parameter derivation for synthesis-by-rule allophone models. In *Proceedings of the Institute of Acoustics,* Vol. 12, part 10, pp. 491–498.

Holmes, W.J., Holmes, J.N. and Judd, M.W., 1990, Extension of the bandwidth of the JSRU parallel-formant synthesizer for high quality synthesis of male and female speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Albuquerque, pp. 313–316.

Huang, X. and Jack, M.A., 1989, Semi-continuous hidden Markov models for speech signals. *Computer Speech and Language,* **3,** pp. 239–251.

Huang, X., Acero, A. and Hon, H.-W., 2001, *Spoken Language Processing* (Upper Saddle River, NJ: Prentice-Hall).

Hunt, M.J., 1987, Delayed decisions in speech recognition—the case of formants. *Pattern Recognition Letters,* **6,** pp. 121–137.

Hyde, S.R., 1972, Automatic speech recognition: a critical survey and discussion of the literature. In *Human Communication: A Unified View,* edited by David, E.E. and Denes, P.B. (New York: McGraw Hill), pp. 399–438.

Ishizaka, K. and Flanagan, J.L., 1972, Synthesis of voiced sounds from a two-mass model of the vocal cords, *Bell Systems Technical Journal,* **51,** pp. 1233–1268.

Jaskie, C. and Fette, B., 1992, A survey of low bit rate vocoders. In *Proceedings of Voice Systems Worldwide,* London, pp. 35–46.

Jelinek, F., 1976, Continuous speech recognition by statistical methods. *Proceedings of the IEEE,* **64,** pp. 532–556.

Jelinek, F., 1985, The development of an experimental discrete dictation recognizer. *Proceedings of the IEEE,* **73,** pp. 1616–1624.

Jelinek, F., 1997, *Statistical Methods for Speech Recognition* (Cambridge, MA: MIT Press).

Juang, B.-H., 1985, Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal,* **64,** pp. 1235–1249.

Juang, B.-H. (ed.), 1998, The past, present and future of speech processing. *IEEE Signal Processing Magazine,* **15,** no. 3, pp. 24–48.

Juang, B.-H. and Katagiri, S., 1992, Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing,* **40,** pp. 3043–3054.

Juang, B.-H. and Rabiner, L.R., 1990, The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **38,** pp. 1639–1641.

Juang, B.-H., Chou, W. and Lee, C.-H., 1997, Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing,* **5,** pp. 257–265.

Junqua, J.-C., 2000, *Robust Speech Recognition in Embedded Systems and PC Applications* (Norwell, MA: Kluwer).

Junqua, J.-C. and Haton, J.-P., 1996, *Robustness in Automatic Speech Recognition* (Norwell, MA: Kluwer).

Junqua, J.-C. and Haton, J.-P. (ed.), 1998, Robust Speech Recognition. Special issue *of Speech Communication,* **25,** pp. 1–192.

Jurafsky, D. and Martin, J.H., 2000, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Upper Saddle River, NJ: Prentice-Hall).

Kingsbury, N.G. and Rayner, P.J.W., 1971, Digital filtering using logarithmic arithmetic. *Electronics Letters,* **7,** pp. 56–58.

Klatt, D.H., 1976, A digital filter bank for spectral matching. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Philadelphia, pp. 573–576.

Klatt, D.H., 1977, Review of the ARPA speech understanding project. *Journal of the Acoustical Society of America,* **62,** pp. 1324–1366.

Klatt, D.H., 1980, Software for a cascade/parallel formant synthesizer. *Journal of the Acoustical Society of America,* **67,** pp. 971–995.

Klatt, D.H., 1987, Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America,* **82,** pp. 737–793.

Knill, K. and Young, S.J., 1997, Hidden Markov models in speech and language processing. In *Corpus-based Methods in Language and Speech Processing,* edited by Young, S.J. and Bloothooft, G. (Dordrecht: Kluwer), pp. 27–68.

Kuhn, R. and De Mori, R., 1990, A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **12,** pp. 570–583.

Kuhn, R. and De Mori, R., 1991, Corrections to ‘A cache-based natural language model for speech recognition’. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **14,** pp. 691–692.

Ladefoged, P., 1993, *A Course in Phonetics,* Third Edition (New York: Harcourt Brace Jovanovich).

Lamel, L.F. and Gauvain, J.-L., 1995, A phone-based approach to non-linguistic speech feature identification. *Computer Speech and Language,* **9,** pp. 87–103.

Lau, R., Rosenfeld, R. and Roukos, S., 1993, Trigger-based language models: a maximum entropy approach. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Minneapolis, Vol. II, pp. 45–48.

Lee, K.-F., 1989, *Automatic Speech Recognition: the Development of the SPHINX system* (Norwell, MA: Kluwer).

Lee, L. and Rose, R., 1998, A frequency-warping approach to speaker normalization. *IEEE Transactions on Speech and Audio Processing,* **6,** pp. 49–60.

Leggetter, C.J. and Woodland, P.C., 1995, Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language,* **9,** pp. 171–186.

Levinson, S.E., Rabiner, L.R. and Sondhi, M.M., 1983, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell Systems Technical Journal,* **62,** pp. 1035–1074.

Levinson, S.E., Olive, J.P. and Tschirgi, J.S., 1993, Speech synthesis in telecommunications. *IEEE Communications Magazine,* **31,** no. 11, pp. 46–53.

Liberman, A.M., Ingemann, F., Lisker, L., Delattre, P. and Cooper, F.S., 1959, Minimal rules for synthesizing speech. *Journal of the Acoustical Society of America,* **31,** pp. 1490–1499.

Lindsey, P.H. and Norman, D.A., 1972, *Human Information Processing* (New York and London: Academic Press).

Linggard, R., 1985, *Electronic Synthesis of Speech* (Cambridge: Cambridge University Press).

Liporace, L.A., 1982, Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Transactions on Information Theory,* **IT-28,** pp. 729–734.

Lippmann, R.P., 1997, Speech recognition by machines and humans. *Speech Communication,* **22,** pp. 1–15.

Lombard, E., 1911, Le signe de l'elévation de la voix. *Ann. Maladiers Oreille, Larynx, Nez, Pharynx,* **37,** pp. 101–119.

Lowerre, B.T., 1976, *The Harpy speech recognition system.* Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University.

Lynn, P.A. and Fuerst, W., 1998, *Introductory Digital Signal Processing with Computer Applications,* Second Edition (Chichester: John Wiley & Sons).

Lyon, R.F., 1988, An analog electronic cochlea. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **36,** pp. 1119–1134.

Lyons, R.G., 1997, *Understanding Digital Signal Processing* (Reading, MA: Addison Wesley).

Makhoul, J., Roucos, S. and Gish, H., 1985, Vector quantization in speech coding. *Proceedings of the IEEE,* **73,** pp. 1551–1588.

Markel, J.D. and Gray, A.H. Jr., 1976, *Linear Prediction of Speech* (Berlin: Springer-Verlag).

Marslen-Wilson, W., 1980, Speech understanding as a psychological process. In *Spoken Language Generation and Understanding,* edited by Simon, J.C. (Dordrecht: Reidel).

Martin, A. and Przybocki, M., 2000, The NIST 1999 speaker recognition evaluation—an overview. *Digital Signal Processing,* **10,** pp. 1–18.

Martin, A., Doddington, G., Kamm, T., Ordowski, M. and Przybocki, M., 1997, The DET curve in assessment of detection task performance. In *Proceedings of the European Conference on Speech Communication and Technology,* Rhodes, pp. 1895–1898.

McAulay, R. and Quatieri, T., 1992, Low-rate speech coding based on the sinusoidal model. In *Advances in Speech Signal Processing,* edited by Furui, S. and Sondhi, M.M. (New York: Marcel Dekker), pp. 165–207.

McCree, A.V. and Barnwell III, T.P., 1995, A mixed excitation LPC vocoder model for low bit rate speech coding. *IEEE Transactions on Speech and Audio Processing,* **3,** pp. 242–250.

Minsky, M. and Papert, S., 1969, *Perceptrons* (Cambridge, MA: MIT Press).

Moon, T.K., 1996, The Expectation-Maximization algorithm. *IEEE Signal Processing Magazine,* **13,** no. 6, pp. 47–60.

Moore, B.C.J., 1997, *An Introduction to the Psychology of Hearing,* Fourth Edition (London: Academic Press).

Moore, B.C.J., Peters, R.W. and Glasberg, B.R., 1990, Auditory filter shapes at low center frequencies. *Journal of the Acoustical Society of America,* **88,** pp. 132–140.

Moore, R., Appelt, D., Dowding, J., Gawron, J.M. and Moran, D., 1995, Combining linguistic and statistical knowledge sources in natural-language processing for ATIS. In *Proceedings of the ARPA Spoken Language Systems Technology Workshop,* Austin (San Francisco, CA: Morgan Kaufmann), pp. 261–264.

Morgan, D.P. and Scofield, C.L., 1991, *Neural Networks and Speech Processing* (Norwell, MA: Kluwer).

Morgan, N. and Bourlard, H.A., 1995a, Continuous speech recognition: an introduction to the hybrid HMM/connectionist approach. *IEEE Signal Processing Magazine,* **12,** no. 3, pp. 25–42.

Morgan. N. and Bourlard, H.A., 1995b, Neural networks for statistical recognition of continuous speech. *Proceedings of the IEEE,* **83,** pp. 742–770.

Moulines, E. and Charpentier, F., 1990, Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication,* **9,** pp. 453–467.

Muthusamy, Y., Barnard, E. and Cole, R.A., 1994. Reviewing automatic language identification. *IEEE Signal Processing Magazine,* **11,** no. 10, pp. 33–41.

Naik, J., 1990, Speaker verification: a tutorial. *IEEE Communications Magazine,* **28,** no. 1, pp. 42–48.

Nakatsu, R., 1990, Anser: An application of speech technology to the Japanese banking industry. *IEEE Computer,* **23,** no. 8, pp. 43–48.

Ney, H. and Ortmanns, S., 1999, Dynamic programming search for continuous speech recognition. *IEEE Signal Processing Magazine,* **16,** no. 5, pp. 64–83.

Ney, H., Essen, U. and Kneser, R., 1994, On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language,* **8,** pp. 1–38.

Ney, H., Martin, S. and Wessel, F., 1997, Statistical language modelling by leaving-one-out. In *Corpus-based Methods in Language and Speech Processing,* edited by Young, S J. and Bloothooft, G. (Dordrecht: Kluwer), pp. 174–207.

Normandin. Y., Cardin, R. and De Mori, R., 1994, High-performance connected digit recognition using maximum mutual information estimation. *IEEE Transactions on Speech and Audio Processing,* **2,** pp. 299–311.

O'Connor, J.D., 1991, *Phonetics,* Second Edition (Harmondsworth: Penguin Books).

O'Shaughnessy, D., 2000, *Speech Communications: Human and Machine,* Second Edition (Piscataway, NJ: IEEE Press).

Olive, J.P., 1977, Rule synthesis of speech using dyadic units. In *Proceedings of the*

IEEE International Conference on Acoustics, Speech and Signal Processing, Hartford, pp. 568–570.

Olive, J., van Santen, J., Möbius, B. and Shih, C., 1998, Synthesis. In *Multilingual Text-to-speech Synthesis: The Bell Labs Approach,* edited by Sproat, R. (Norwell, MA: Kluwer), pp. 191–228.

Ostendorf, M., Digalakis, V.V. and Kimball, O.A., 1996, From HMM's to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Transactions on Speech and Audio Processing,* **4,** pp. 360–378.

Page, J.H. and Breen, A.P., 1996, The Laureate text-to-speech system—architecture and applications. *BT Technology Journal,* **14,** pp. 57–67.

Pallett, D.S., Fiscus, J.G., Fisher, W.M., Garofolo, J.S., Lund, B.A., Martin, A. and Przybocki, M., 1995, 1994 benchmark tests for the ARPA spoken language program. In *Proceedings of the ARPA Spoken Language Systems Technology Workshop,* Austin (San Francisco, CA: Morgan Kaufmann), pp. 5–36.

Pallett, D.S., Fiscus, J.G., Garofolo, J.S., Martin, A. and Przybocki, M., 1999, 1998 broadcast news benchmark test results: English and non-English word error rate performance measures. In *Proceedings of the DARPA Broadcast News Workshop,* Herndon (available from NIST, http://www.nist.gov).

Papoulis, A., 1991, *Probability, Random Variables, and Stochastic Processes,* Third Edition (New York: McGraw-Hill).

Patterson, R.D., 1976, Auditory filter shapes derived with noise stimuli. *Journal of the Acoustical Society of America,* **59,** pp. 640–654.

Paul, D.B. and Martin, E.A., 1988, Speaker stress-resistant continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* New York, pp. 283–286.

Picone, J.W., 1993, Signal modeling techniques in speech recognition. *Proceedings of the IEEE,* **81,** pp. 1215–1247.

Pierrehumbert, J., 1980, *The phonology and phonetics of English intonation.* Ph.D. Thesis, Massachusetts Institute of Technology.

Pierrehumbert, J., 1981, Synthesizing intonation. *Journal of the Acoustical Society of America,* **70,** pp. 985–995.

Pitton, J.W., Wang, K. and Juang, B.-H., 1996, Time-frequency analysis and auditory modeling for automatic recognition of speech. *Proceedings of the IEEE,* **84,** pp. 1199–1215.

Quackenbush, S.R., Barnwell, T.P. and Clements, M.A., 1988, *Objective Measures of Speech Quality* (Englewood Cliffs, NJ: Prentice-Hall).

Rabiner, L.R., 1989, A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE,* **37,** pp. 257–286.

Rabiner, L.R., 1994, Applications of voice processing to telecommunications. *Proceedings of the IEEE,* **82,** pp. 199–228.

Rabiner, L.R., 1997, Applications of speech recognition in the area of telecommunications. In *Proceedings of the 1997 IEEE Workshop on Automatic Speech Recognition and Understanding,* Santa Barbara, edited by Furui, S., Juang, B.-H. and Chou, W. (Piscataway, NJ: IEEE Press), pp. 501–510.

Rabiner, L.R. and Juang, B.-H., 1993, *Fundamentals of Speech Recognition* (Englewood Cliffs, NJ: Prentice-Hall).

Rabiner, L.R. and Schafer, R.W., 1978, *Digital Processing of Speech Signals* (Englewood Cliffs, NJ: Prentice-Hall).

Rabiner, L.R., Schafer, R.W. and Flanagan, J.L., 1971, Computer synthesis of speech by concatenation of formant-coded words. *Bell Systems Technical Journal,* **50,** pp. 1541–1558.

Rayner, M., Carter, D., Bouillon, P., Digalakis, V. and Wirén, M. (eds.), 2000, *The Spoken Language Translator* (Cambridge: Cambridge University Press).

Reeves, A.H., 1938, French patent 852183.

Reynolds, D.A., Dunn, R.B. and McLaughlin, J.J., 2000, The Lincoln speaker recognition system: NIST eval2000. In *Proceedings of the International Conference on Spoken Language Processing,* Beijing, pp. II–470–473.

Richards, H.B. and Bridle, J.S., 1999, The HDM: a segmental hidden dynamic model of coarticulation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Phoenix, pp. 357–360.

Roach, P., 1991, *English Phonetics and Phonology,* Second Edition (Cambridge: Cambridge University Press).

Robinson, D.W. and Dadson, R.S., 1956, A re-determination of the equal-loudness relations for pure tones. *British Journal of Applied Physics,* **7,** pp. 166–181.

Rodman, R.D., 1999, *Computer Speech Technology* (Norwood, MA: Artech House).

Rose, J.E., Hind, J.E., Anderson, D.J. and Brugge, J.F., 1971, Some effects of stimulus intensity on response of auditory nerve fibers in the squirrel monkey . *Journal of Neurophysiology,* **34,** pp. 685–699.

Rose, R., 1995, Keyword detection in conversational speech utterances using hidden Markov model based continuous speech recognition. *Computer Speech and Language,* **9,** pp. 309–333.

Rosen, S. and Howell, P., 2001, *Signals and Systems for Speech and Hearing,* Second Edition (London: Academic Press).

Rosenberg, A., 1976, Automatic speaker verification. *Proceedings of the IEEE,* **64,** pp. 475–487.

Rosenblatt, F., 1962, *Principles of Neurodynamics* (New York: Spartan).

Sakoe, H., 1979, Two-level DP matching—a dynamic programming based pattern matching algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **ASSP-27,** pp. 588–595.

Sakoe, H. and Chiba, S., 1978, Dynamic programming algorithms optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **ASSP-26,** pp. 43–49.

Schafer, R.W. and Markel, J.D. (eds.), 1979, *Speech Analysis* (New York: IEEE Press).

Schroeder, M. and Atal, B.S., 1985, Code-excited linear prediction (CELP): high quality speech at very low bit rates. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Tampa, pp. 937–940.

Sellick, P.M., Patuzzi, R. and Johnstone, B.M., 1982, Measurement of basilar membrane motion in the guinea pig using the Mössbauer technique. *Journal of the Acoustical Society of America,* **72,** pp. 131–141.

Seneff, S., 1988, A joint synchrony/mean-rate model of auditory speech processing. *Journal of Phonetics,* **16,** pp. 55–76.

Shailer, M.J. and Moore, B.C.J., 1983, Gap detection as a function of frequency, bandwidth, and level. *Journal of the Acoustical Society of America,* **74,** pp. 467–473.

Silverman, H.F. and Morgan, D.P., 1990, The application of dynamic programming to connected speech recognition. *IEEE Acoustics, Speech and Signal Processing Magazine,* **7,** pp. 6–25.

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J. and Hirschberg, J., 1992, TOBI: a standard for labeling English prosody . In *Proceedings of the International Conference on Spoken Language Processing,* Banff, pp. 867–870.

Spanias, A.S., 1994, Speech coding: a tutorial review. *Proceedings of the IEEE,* **82,** pp. 1541–1582.

Spiegel, M. and Kamm, C., 1997 (eds.), 3rd IEEE Workshop on Interactive Voice Technology for Telecommunications Applications. Special issue of *Speech Communication,* **23,** Issue 1–2.

Sproat, R. (ed.), 1998, *Multilingual Text-to-speech Synthesis: the Bell Labs Approach* (Norwell, MA: Kluwer).

Sproat, R., van Santen, J. and Olive, J., 1998, Further issues. In *Multilingual Text-to-speech Synthesis: the Bell Labs Approach,* edited by Sproat, R. (Norwell, MA: Kluwer), pp. 245–254.

Supplee, L.M., Cohn, R.P., Collura, J.S. and McCree, A.V., 1997, MELP: the new Federal Standard at 2400 bps. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Munich, pp. 1591–1594.

Titze, I.R., 1973, The human vocal cords: a mathematical model, Part 1. *Phonetica,* **28,** pp. 129–170.

Titze, I.R., 1974, The human vocal cords: a mathematical model, Part 2. *Phonetica,* **29,** pp. 1–21.

Trupp, R.D., 1970, Computer-controlled message synthesis. *Bell Laboratories Record,* June/July 1970, pp. 175–180.

Varga, A.P. and Moore, R.K., 1990, Hidden Markov model decomposition of speech and noise. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Albuquerque, pp. 845–848.

Velichko, Z.M. and Zagoruyko, N.G., 1970. Automatic recognition of 200 words. *International Journal of Man-Machine Studies,* **2,** pp. 223–234.

Vintsyuk, T.K., 1968, Speech recognition by dynamic programming methods. *Kibernetika, Cybernetics,* **4,** pp. 81–88.

Vintsyuk, T.K., 1971, Element-wise recognition of continuous speech consisting of words of a given vocabulary. *Kibernetika, Cybernetics,* **7,** pp. 133–143.

Viterbi, A.J., 1967, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory,* **IT-13,** pp. 260–269.

Vogten, L.L.M., 1974, Pure tone masking: a new result from a new method. In *Facts and Models in Hearing,* edited by Zwicker, E. and Terhardt, E. (Berlin: Springer-Verlag).

Voiers, W.D., 1977, Diagnostic acceptability measure for speech communication systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Hartford, pp. 204–207.

von Békésy, G., 1947, The variations of phase along the basilar membrane with sinusoidal vibrations. *Journal of the Acoustical Society of America,* **19,** pp. 452–460.

Waibel, A. and Lee, K.-F. (eds.), 1990, *Readings in Speech Recognition* (San Mateo, CA: Morgan Kaufmann).

Weber, D.L., 1977, Growth of masking and the auditory filter. *Journal of the Acoustical Society of America,* **62,** pp. *424–429.*

Weinschenk, S. and Barker, D., *Designing Effective Speech Interfaces* (New York: John Wiley and Sons).

Weinstein, C.J., 1991, Opportunities for advanced speech processing in military computer-based systems. *Proceedings of the IEEE,* **79,** pp. 1626–1641.

Wells, J.C., 2000, *Longman Pronunciation Dictionary,* Second Edition (Harlow: Pearson Education Limited).

Wilpon, J., Rabiner, L., Lee, C.-H. and Goldman, E., 1990, Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing,* **38,** pp. 1870–1878.

Woodland, P.C., Gales, M.J.F. and Pye, D., 1996, Improving environmental robustness in large vocabulary speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* Atlanta, pp. 65–68.

Wouters, J. and Macon, M.W., 2001, Control of spectral dynamics in concatenative speech synthesis. *IEEE Transactions on Speech and Audio Processing,* **9,** pp. 30–38.

Young, S.J., 1996, A review of large-vocabulary continuous-speech recognition. *IEEE Signal Processing Magazine,* **13,** no. 5, pp. 45–57.

Young, S.J. and Chase, L.L., 1998, Speech recognition evaluation: a review of the U.S. CSR and LVCSR programmes. *Computer Speech and Language,* **12,** pp. 263–279.

Young, S.J. and Fallside, F., 1979, Speech synthesis from concept: a method for speech output from information systems. *Journal of the Acoustical Society of America,* **62,** pp. 685–695.

Young, S.J. and Woodland, P.C., 1993, The use of state tying in continuous speech recognition. In *Proceedings of the European Conference on Speech Communication and Technology,* Berlin, pp. 2203–2206.

Zissman, M.A., 1996, Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Speech and Audio Processing,* **4,** pp. 31–44.

Zlokarnik, I., 1993, Experiments with an articulatory speech recognizer. In *Proceedings of the European Conference on Speech Communication and Technology,* Berlin, pp. 2215–2218.

Zwicker, E., 1961, Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *Journal of the Acoustical Society of America,* **33,** p. 248.

SOLUTIONS TO EXERCISES

The notes that follow give an indication of the main points that should be covered in the answers to the exercises. In many cases this indication is provided by reference to the appropriate part of the text where the information can be found.

Chapter 1

E1.1 Not suitable: scanning text, position control. Beneficial: data entry in hands-busy situations, or for operators without keyboard skill. Necessary: where access is only available by telephone.

E1.2 Phonetics is concerned with properties of speech sounds, phonology with their function in languages.

E1.3 See p. 8.

E1.4 See pp. 3–4.

E1.5 See p. 4.

E1.6 See pp. 6–7.

E1.7 The enormous redundancy in the speech signal, at many levels, means that the new information needed to deduce a message is a minute fraction of that needed to specify the waveform detail. The human auditory and cognitive systems are complex enough to exploit the many forms of redundancy that are present. See pp. 8–9.

Chapter 2

E2.1 Voiced: harmonic, most power at low frequencies, impulsive. Voiceless: continuous spectrum, fairly even spectral distribution, continuous in time.

E2.2 Surface movements of the vocal folds, including ejection of air between the folds during closure motion. See pp. 13–14.

E2.3 See pp. 16–17.

E2.4 See p. 19.

E2.5 Time-varying glottal impedance affects formants; pitch affects larynx height, so altering length of pharynx; vibration of vocal folds is affected by F_1 resonance.

E2.6 Losses in vocal tract, such as from wall movements, viscosity, heat conduction losses; nasal coupling; loss from radiation to outside air; time-varying loss from glottal impedance. The effects of all these losses depend on what vowel is being produced.

E2.7 The information in a speech signal is mainly conveyed by variation of the frequencies of the main resonances, and of the fundamental frequency. The inherent frequency analysis in spectrograms means that these properties are more clearly seen in these representations than in waveforms.

E2.8 See p. 25 and Figures 2.10 and 2.11 on p. 24.

E2.9 See pp. 28–31.

E2.10 The main acoustic effects, such as formant frequencies and intensities, depend on many articulatory features, which are difficult to model accurately. The problem is exacerbated by difficulties in measuring human articulation. See p. 27.

Chapter 3

E3.1 The peak response from PTCs corresponds with the resonant peak in the basilar membrane, and to the peak firing response from corresponding auditory nerve fibres. See pp. 36–39.

E3.2 The main effect is probably caused by differences in the amount of phase-locking between the responses of auditory neurons. Some increase of dynamic range could also be given by the fact that hair cells are not all equally sensitive.

E3.3 See Section 3.6 (pp. 41–42).

E3.4 See pp. 37, 39–40 and Section 3.7.2 (p. 43).

E3.5 Auditory models will make available the information that humans can use, but will discard other information. It is dangerous to use consistent properties of speech production that humans do not use in perception, because they may not be preserved under the influence of what would be judged to be tolerable distortions of a speech signal.

Chapter 4

E4.1 See pp. 47–48.

E4.2 Because the quantizing noise is then highly correlated with the speech signal. See p. 49.

E4.3 Auditory masking (see Chapter 3) means that it is more efficient to make quantizing noise vary with signal level.

E4.4 The essential features of a vocoder are: (i) separation of the fine detail from the general shape of the short-term spectrum; (ii) representing the general spectrum shape by a slowly varying parametric model; (iii) representing periodicity (when present) of excitation by a slowly varying parameter; (iv) re-synthesis from the parametric spectrum model, fed with the signal from an excitation model.

E4.5 LPC makes assumptions that may not be satisfied by real speech (see p. 55). Auditory analysis (see Chapter 3), which is in some ways like channel vocoder analysis with a very large number of channels, appears to be able to cope quite well with the channel approximations to a speech spectrum (see Figure 4.4). Sinusoidal representations achieve a similar result.

E4.6 Transmission rate can be reduced by exploiting redundancy in the basic parametric representation. Correlation between parameters at any time can be exploited by vector quantization, and correlation over time can be exploited by variable-frame-rate transmission. The latter inherently needs buffer delay for constant-rate real-time links.

E4.7 See pp. 60–63.

Chapter 5

E5.1 Advantages: high technical quality of waveform reproduction; low-cost equipment. Disadvantages: large memory requirement; prosody problems and co-articulation prevent flexibility of message structure.

E5.2 Saving in memory for message storage; ability to modify prosody of stored speech; crude model of co-articulation possible.

E5.3 Advantages: unlimited vocabulary possible with relatively modest memory requirement; straightforward process to change speaker type or language; provides some aspects of natural articulatory transitions. Disadvantages: difficulties of matching diphones at joins; only capture immediate phonetic context; editing new diphones is labour-intensive.

E5.4 Potential problems are difficulty in obtaining smooth joins and in modifying pitch and timing. PSOLA: segments are joined pitch synchronously with overlapping tapered windows to obtain smooth joins; pitch and duration can be modified by modifying spacing and number of windows. See pp. 74–77.

Chapter 6

E6.1 See pp. 81–82.

E6.2 Cost of memory for tables is insignificant; initial values for table entries can be guided by phonetic theory, and values can then be changed where shown to be necessary, preferably using automatic methods; transformations can be applied to whole sets of tables to change speaker type.

E6.3 Intrinsic allophonic variation is best provided by having co-articulatory effects built into the rule structure; extrinsic allophones can be selected according to the identities of neighbouring phonemes.

E6.4 See p. 85.

Chapter 7

E7.1 Dictionary look-up is necessary for a high proportion of words, supplemented by letter-to-sound rules for the inevitable words that will not be covered by the dictionary. See pp. 96, 100–101 for more detail.

E7.2 Helps in syntactic analysis and in deriving pronunciation. See pp. 99–100.

E7.3 Segmentation into words; processing of punctuation; expansion of numerals, abbreviations, special symbols. Some of these may be ambiguous and only resolved by syntactic analysis or even understanding of the text.

E7.4 Fundamental frequency pattern is most important for naturalness, although human productions show quite wide variations. Durations affect phonemic cues as well as stress and rhythm. Intensity variations, except those intrinsic to phoneme type, are of much less importance. See also pp. 6–7.

E7.5 Structure of words, phrases and sentences. See pp. 103–106 for more detail.

E7.6 See Section 7.6 on pp. 106–107.

E7.7 Because synthesis from concept can provide much information about the required prosody, which is difficult to derive from conventional text.

Chapter 8

E8.1 See p. 109. With a suitable distance metric, the right word will usually give a better match than other words (even though all words may match poorly).

E8.2 Bandwidth should be chosen to capture formant structure but not resolve pitch harmonics. It is desirable to give more weight to low frequencies by using critical band spacing. See pp. 111–112 and also Sections 2.6 and 3.5.

E8.3 Variation of vocal effort, amplifier gain and distance from the microphone mainly cause a scale factor change to the signal. Logarithmic representation converts this to a constant addition to all channels, which does not change the shape of the spectral cross-section. Difficulties arise in silent regions, where the logarithmic level will vary a lot with background noise level.

E8.4 See pp. 115–117 and Figure 8.3.

E8.5 Normalization for path length and template length is required. See p. 117.

E8.6 The effects of end-point errors can be mitigated by allowing more freedom for timescale distortion towards the template ends, and by discounting mismatch at the template end frames. Use of a connected-word algorithm in conjunction with a silence template avoids the problem (see p. 123).

E8.7 By tracing back along the optimum path to see where it moves between templates.

E8.8 See pp. 124–125.

E8.9 The use of wildcard templates in a training syntax is an effective method (see p. 125). For connected-word recognition the performance is much improved by careful choice of syntaxes for embedded training, in which existing word templates precede and follow the wildcard.

Chapter 9

E9.1 The features emitted do not uniquely determine the state of the model, which is therefore hidden from the observer.

E9.2 The use of a recurrence relationship enables the state probabilities to be determined successively, from frame to frame.

E9.3 The Viterbi algorithm only considers the likelihood for the most likely path through the model, which must be less than the sum of likelihoods over all paths. Advantages: simplifies computation and avoids scaling problems.

E9.4 The model topology can be constrained by setting some initial transition probabilities to zero. Unsuitable initial values of other parameters may cause a poor local optimum to be obtained.

E9.5 The number of different feature vectors is reduced, so sensible statistical distributions can be obtained from acceptable amounts of training data.

E9.6 A normal distribution provides a compact, mathematically convenient way of estimating a probability for any value of feature vector. The main limitation arises when observed distributions are far from normal, but can be overcome by using a mixture of normal distributions. See pp. 142–144.

E9.7 With Viterbi training, only one path through the models is considered. Hence a state-level segmentation can be obtained first and then the relevant statistics computed, rather than needing probabilities for all possible paths.

E9.8 There are very many probabilities to multiply together, which leads to number range problems. One option is to scale the probabilities; another is to represent all probabilities in logarithmic form (see pp. 153–155).

E9.9 See Section 9.13 on pp. 155–156.

E9.10 The most likely duration for state occupancy is just one frame (see p. 156), which is inappropriate for representing most phonetic events. See Figure 9.2 on p. 157 for the effect of splitting a single state into a sequence of states.

Chapter 10

E10.1 Effects due to excitation are separated out from the vocal tract response; dimensionality can be reduced because the most important phonetic information is in the first few features; features are decorrelated, so diagonal covariance matrices can be used. See Section 10.5.

E10.2 The signal is pre-emphasized and windowed (Sections 10.2 and 10.3). For each windowed portion of signal a Fourier transform is computed, outputs are summed as given by the triangular filters shown in Figure 10.2 and a cosine transform is calculated. Linear regression is applied to compute time derivatives, and derivatives of these first-order derivatives (see p. 166).

E10.3 Pre-emphasis, filter bandwidth and spacing, and amplitude compression all have correlates in the perceptual system (see p. 165 and refer to Chapter 3).

Chapter 11

E11.1 See Section 11.2.3 (pp. 173–174).

E11.2 MAP provides an optimal combination of new data with existing models, whereas MLLR computes a linear transform to apply to the model parameters to maximize the likelihood of some new data. See pp. 176–178.

E11.3 Discriminative training is closer to requirements for recognition. See p. 179.

Chapter 12

E12.1 See pp. 184–185.

E12.2 See pp. 186–187.

E12.3 Phonetic knowledge influences selection of context-dependent models (e.g. triphones), and choice of questions for phonetic decision trees. See p. 194.

E12.4 Probabilities for unseen trigrams can be estimated by backing off to bigrams or unigrams, or by interpolating between bigram and unigram probabilities. Probabilities for trigrams that do occur need to be reduced to make probability mass available for the unseen trigrams. See pp. 199–201.

E12.5 In both forms of backing off, a probability for a specific context is replaced by a probability corresponding to a more general context. In the case of language modelling, the quantity required is the probability of a particular context occurring; for acoustic modelling it is a distribution describing the possible acoustic attributes corresponding to a given context.

E12.6 One-pass Viterbi search makes a single recognition decision based on all the evidence, but the search space is large so careful structuring and extensive pruning are necessary. Multiple-pass search may be more efficient and allows complex language models to be applied, but there is the danger of discarding the correct solution at an early stage. See pp. 203–205.

E12.7 Word recognition errors, sentence recognition errors, and system response errors are all relevant. See p. 210.

Chapter 13

E13.1 See pp. 216–217. The main practical difficulties are in capturing the time-varying properties of speech and in performing joint segmentation and recognition.

E13.2 Learning algorithms for ANNs achieve discriminability by directly maximizing *a posteriori* probabilities of output classes given the input. MMI maximizes a different quantity (mutual information) to achieve the result of increasing the probability for the correct model more than that for all possible incorrect models. See p. 216 and also Section 11.5.

Chapter 14

E14.1 This is an open-set identification task, so there are three possible types of error: identification errors occur if English is misrecognized as French or *vice versa;* false rejections occur if an English or French utterance is misrecognized as some other language; false acceptances occur if another language is misrecognized as either English or French. See pp. 220–221.

E14.2 See p. 222 and Figure 14.2.

E14.3 See pp. 224–225. Text-dependent speaker recognition tends to be more accurate because the system 'knows' the words that have been spoken.

E14.4 See p. 226 for speaker recognition and p. 228 for language recognition.

Chapter 15

E15.1 (i) Hands and eyes busy: access to computers while operating equipment; (ii) Remoteness: access to automated telephone services; (iii) Small devices: access to palm-top computers, toys. See p. 232 and examples in the chapter.

E15.2 Applications requiring variety and flexibility of message content. For other applications recorded speech can often be used and gives better quality.

E15.3 Word accuracy depends on the nature of the task. See pp. 235–237.

E15.4 **O**ffice dictation requires transcription whereas a flight booking service requires understanding (see Section 12.3). Office dictation: relatively quiet environment, large vocabulary, read speech, usually trained or adapted for individual speakers. Booking service: telephone speech, strong contextual constraints on vocabulary, spontaneous speech, speaker-independent.

Chapter 16

E16.1 See pp. 252–253.

E16.2 Much of the information needed to produce the appropriate style of speech for synthesis can only be derived from the meaning of the required message. Similarly, the meaning is frequently essential to resolve the choice between alternative utterances in recognition. Understanding also has to be used by human beings doing the same speech production and recognition tasks.

GLOSSARY

acoustic model The model component of a large-vocabulary speech recognition system that is used to compute $P(Y|W)$, the probability of a sequence of acoustic feature vectors Y being generated by a sequence of word models W. The other model component is a **language model**.

allophone An acoustically distinct variant of a phoneme. The realization of a phoneme depends on many factors, and most especially on the characteristics of the surrounding phonemes. For example, the English phoneme /t/ is usually articulated quite differently in the word "eighth" than in the word "eight", and these two realizations would therefore be referred to as different allophones.

beam search A recognition search in which, at each frame time, any path through a sequence of templates or models is discarded if its cumulative score is not within some specified threshold of the best-scoring path to that point. Thus only those scores that are inside a 'beam' are retained and propagate on to the next frame.

bigram model A statistical language model in which the probability of a word (or other linguistic unit) occurring depends only on the identity of the current word and the identity of the immediately preceding word.

biphone A unit of speech that refers to a phone in the context of one immediately adjacent phone, which may either be the preceding or the following phone.

cepstrum The Fourier transform of the log magnitude spectrum, which can be conveniently computed by means of a discrete cosine transform. The cepstrum has some desirable properties for providing features for automatic speech recognition. In particular, the individual **cepstral coefficients** tend to be uncorrelated with each other, and most of the phonetically significant variation is concentrated in the lowest few coefficients.

co-articulation Refers to the context-dependent nature of speech production, whereby talkers naturally articulate a sequence of speech sounds in an overlapping manner.

codebook A set of vectors that represent a subset of all possible vectors and are used for **vector quantization**.

concatenative synthesis Electronic synthesis of speech by joining together sections of pre-recorded human speech, either as waveforms or in a coded form.

connected speech Speech that is produced without pauses between words, as is usually the case in normal human utterances. An alternative is to speak the words in an isolated manner, leaving short pauses between them.

continuous speech recognition Recognition of a continuous stream of naturally spoken connected speech. The location of the end of an utterance will not generally be known in advance, and it is often necessary to make recognition decisions about the identities of earlier words before the utterance is finished.

critical band An auditory filter, which is one of a bank of band-pass filters that are assumed to exist in the peripheral auditory system.

critical bandwidth A measure of the 'effective bandwidth' of an auditory filter, which is often estimated using perceptual masking experiments. Critical bandwidth increases with centre frequency, and hence the frequency-resolving power of the ear decreases as frequency increases.

decibel (dB) One tenth of a bel. The number of dB is equal to 10 times the logarithm (to the base 10) of the ratio of two intensities. This quantity is thus a measure of relative intensity. The absolute intensity of a sound can be specified by stating the number of dB relative to some reference intensity. The reference that is most commonly used is a pressure of 2×10^{-5} N/m^2, which is defined to be 0 dB **sound pressure level** (SPL).

decoding The process of search in a speech recognizer to find the most probable sequence of words. For a large-vocabulary system the decoding task can be difficult due to the very large number of possible words that may need to be considered at any one point.

deletion error In a connected-speech recognizer, a deletion error occurs when there is a word (or other linguistic unit) in the input speech for which there is no word at a corresponding position in the recognition output. **demisyllable** A unit of speech that represents half a syllable split in the centre of the vowel.

diphone (or **dyad)** A unit of speech that stretches from the centre of one phone to the centre of the following phone.

diphthong A vowel which shows a noticeable change from one vowel quality to another within a syllable; e.g. the English words “by”, “boy” and “bough”.

formant A resonance in the vocal tract, often manifested as a peak in the spectral envelope. Formants are most obvious during vowels and vowel-like sounds. By convention formants are numbered from the low-frequency end; the frequencies of the first three formants have the most influence on phonetic properties.

fundamental frequency (**F**$_0$) The frequency of the lowest-frequency component of a complex sound, as evident in the repetition rate of the waveform. By definition, fundamental frequency only applies to **periodic sounds,** and is determined by the rate of vocal fold vibration during sound production. There is a fairly close relationship between fundamental frequency and perceived **pitch**.

harmonic A component of a complex sound whose frequency is an integral multiple of the **fundamental frequency**. The frequency of the first harmonic is equal to the fundamental frequency, the second harmonic corresponds to twice the fundamental frequency, and so on.

insertion error In a connected-speech recognizer, an insertion error occurs when there is a word (or other linguistic unit) in the recognition output for which there is no word at the same position in the input speech.

intonation The distinctive use of pitch patterns.

keyword spotting A form of speech recognition whereby a limited-vocabulary recognizer works on input speech that may include other words that are not in its vocabulary, by looking for vocabulary words and ignoring the other words.

language model A term used in large-vocabulary speech recognition to refer to the model component of a recognition system that is used to compute $P(W)$, the probability of the sequence of words W. Multiplying this probability by the **acoustic-model** probability $P(\boldsymbol{Y}|W)$ gives a quantity that is proportional to $P(W|Y)$ and can therefore be used to make a recognition decision.

Lombard effect Changes in the way a person speaks (in particular an increase in vocal effort) when in a noisy acoustic environment.

mel A perceptual scale that measures the perceived pitch of a sound. Up to about 1 kHz there tends to be a fairly direct correspondence between pitch and frequency, while at higher frequencies the relationship is logarithmic. The mel scale is often used to provide a perceptually motivated frequency scale upon which to compute a spectrum for subsequent calculation of **cepstral coefficients**.

mel-frequency cepstral coefficients (MFCCs) Cepstral coefficients computed from a spectrum that is represented on a mel scale (or other similar perceptually motivated non-linear frequency scale). A popular configuration uses triangular 'filters' whose centre frequencies and bandwidths both conform to a mel scale.

monophthong A vowel without any change in quality within a syllable; e.g. the English words "bee" and "boo". (Note the contrast with **diphthong**.)

morph A text segment representing the realization of a **morpheme** (which is an abstract unit). Most morphemes are realized as single morphs but some, such as the morpheme of plurality, are realized differently in different words.

morpheme The minimal distinctive unit of grammar, which is the smallest functioning unit in the composition of words. For example, the word "lighthouse" contains two morphemes, while "lighthouses" has three, with the addition of the morpheme of plurality.

morphology The branch of grammar which studies the structure of forms of words, primarily through the **morpheme** construct.

***N*-gram model** A statistical language model in which the probability of a word (or other linguistic unit) occurring depends only on the identity of the current word and the identities of the immediately preceding N-1 words.

period The smallest time interval over which a periodic waveform repeats itself. The shorter the period, the higher the **frequency**.

periodic sound A sound whose waveform repeats itself regularly over time.

perplexity A measure of how accurately a language model can predict the next word in a sequence. For a sequence of K words, the perplexity is defined as the inverse of the K^{th} root of the probability of the sequence as given by the language model. The lower the perplexity (i.e. the higher the probability), the better the prediction of the sequence by the language model.

phone A term used to refer to any one manifestation of a **phoneme**. It is usual to use square brackets when referring to a phone, e.g. [t].

phoneme The smallest unit in the sound system of a language for which substitution of one unit for another might make a distinction of meaning. For example, the English word "do" contains two phonemes, and differs from the word "to" in the first phoneme and from the word "doe" in the second phoneme. It is usual to write phoneme symbols between oblique lines, e.g. /t/.

phonetic element A term used in the Holmes-Mattingly-Shearme phonetic synthesis-by-rule system to refer to an acoustically homogeneous region that may correspond to a whole phone or to part of a phone.

phonetics The science which studies the characteristics of human speech sound generation, and provides methods for the description, classification and transcription of the speech sounds that are generated.

phonology A branch of linguistics which studies the sound systems of languages. Phonology is concerned with the patterns of distinctive sound within a language.

pitch The attribute of auditory sensation in terms of which a sound may be ordered on a musical scale from "low" to "high". Variations in pitch give rise to a sense of melody. The pitch of a complex sound is related to its **fundamental frequency,** but pitch is a subjective attribute.

pitch-synchronous overlap-add (PSOLA) A method for joining two waveform segments in a pitch-synchronous manner by applying a tapered window function to the two segments, overlapping them and adding them together.

prosody Refers collectively to variations in pitch, intensity and timing.

spectrogram A graphical display of sound in which time is on the horizontal axis and frequency is on the vertical axis. Intensity is shown by a grey-scale representation (the darker the display, the higher the intensity) or by a colour display (the brighter the colour, the higher the intensity).

stress A term used in **phonetics** to refer to the degree of force used in producing a **syllable**. Stressed syllables are more prominent than unstressed syllables, with the stressed syllables being typically louder and longer and carrying distinctive pitch movements. From the viewpoint of **phonology,** stress provides a way of distinguishing degrees of emphasis in sentences and of distinguishing between different words (for example, the stress pattern for the word "increase" is different depending on whether the word is functioning as a noun or as a verb).

substitution error In a connected-speech recognizer, a substitution error occurs when a word (or other linguistic unit) in the recognition output is different from the word at the corresponding position in the input speech.

syllable A unit of pronunciation that is typically larger than a single sound and smaller than a word. It is difficult to provide a more precise definition of a syllable, and various definitions exist. Typically a syllable comprises a vowel which is preceded and followed by zero or more consonants, although certain consonants can form a complete syllable on their own (for example, the word "button" is often pronounced with an [n] making up the complete final syllable).

synthesis from concept Conversion (by machine) from coded concepts to speech. text-to-speech (TTS) **synthesis** Conversion (by machine) from orthographic text to speech.

trigram model A statistical language model in which the probability of a word (or other linguistic unit) occurring depends only on the identity of the current word and the identities of the two immediately preceding words.

triphone A unit of speech unit that refers to a phone in the context of its immediately preceding and immediately following phones.

unigram model A statistical language model in which the probability of a word (or other linguistic unit) occurring depends only on the identity of that word.

vector quantization (VQ) A method of representing vectors efficiently when a multi-dimensional parameter space is not uniformly occupied. A subset of possible combinations of parameter values is stored in a **codebook,** and any measured vector is represented by the index of the closest codebook entry.

vocoder A term (a contraction of VOice CODER) that is widely used to refer to speech coders that involve analysis and resynthesis operations. Most vocoders are based on a model of speech production that separates out the sound generation process from the subsequent vocal-tract filtering operation.

word accuracy The percentage of words that a speech recognizer has recognized correctly (equal to 100 minus the **word error rate**).

word error rate The percentage of words that a speech recognizer has recognized incorrectly. The word error rate is obtained by adding the number of **substitution, deletion** and **insertion** errors, dividing this sum by the total number of words in the utterance, and converting to a percentage.

INDEX