TRAINABLE SPEECH SYNTHESIS

Robert Edward Donovan

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

This dissertation is submitted for consideration for the degree of Doctor of Philosophy at the University of Cambridge

Summary

This thesis is concerned with the synthesis of speech using trainable systems. The research it describes was conducted with two principle aims: to build a hidden Markov model (HMM) based speech synthesis system which could synthesise very high quality speech; and to ensure that all the parameters used by the system were obtained through training. The motivation behind the first of these aims was to determine if the HMM techniques which have been applied so successfully in recent years to the problem of automatic speech recognition could achieve a similar level of success in the field of speech synthesis. The motivation behind the second aim was to construct a system that would be very flexible with respect to changing voices, or even languages.

A synthesis system was developed which used the clustered states of a set of decision-tree state-clustered HMMs as its synthesis units. The synthesis parameters for each clustered state were obtained completely automatically through training on a one hour single-speaker continuous-speech database. During synthesis the required utterance, specified as a string of words of known phonetic pronunciation, was generated as a sequence of these clustered states. Initially, each clustered state was associated with a single linear prediction (LP) vector, and LP synthesis used to generate the sequence of vectors corresponding to the state sequence required. Numerous shortcomings were identified in this system, and these were addressed through improvements to its transcription, clustering, and segmentation capabilities. The LP synthesis scheme was replaced by a TD-PSOLA scheme which synthesised speech by concatenating waveform segments selected to represent each clustered state. The final system produced speech which, though in a monotone, was natural sounding, remarkably fluent, and highly intelligible. The segmental intelligibility was measured using the Modified Rhyme Test, and a 5.0% error rate obtained. The speech produced by the system mimicked the voice of the speaker used to record the training database. The system could be retrained on a new voice in less than 48 hours, and has been successfully trained on four voices.

Acknowledgements

There are a very large number of people to thank in connection with this work. I shall begin at the beginning, by thanking my original supervisor, the late Professor Frank Fallside. To him I am deeply grateful, both for letting me join the CUED Speech Vision and Robotics (SVR) group, and for having the vision to start me on the subject of speech synthesis using HMMs. After Frank's sad death in March 1993, Phil Woodland became my new supervisor. Phil undoubtably deserves the largest slice of the thanks on this page. Credit where credit's due: it was Phil's suggestion that I should use the HMM system used in the HTK large vocabulary speech recognition system in place of the ergodic system which I had previously been working towards. Thanks to him for all his help, all our discussions, the freedom I was given to make my own decisions, the proof-reading, the form-filling, etc., etc., etc., etc.,...

Many other people deserve thanks for making this research possible. Most of all I must thank the people who recorded databases, Tina Burrows, Phil, and Patricia. I must also thank the members of my regular listening test squad, Tina, George Harpur, Richard Shaw, Dan Kershaw, Gavin Rummery, David James, Jon Lawn, David Pye, and Gary Cook, as well as all those others who did tests at one time or another. Thanks also to my proofreaders, Phil, Gareth Jones, Beth Logan, Jason Humphries, Simon Blackburn, Mark Gales, Carl Seymour, Jon Foote, Kate Knill, and Parham Zolfaghari. Thanks to Phil, Julian Odell, and Bertrand Lecordier, for all their efforts in getting my audio examples onto compact disc. Thanks must also go to Professor Steve Young, Phil, Julian, and everyone else connected with the development of the HTK software, without which this project would have been much more difficult. Thanks also to Sarah Hawkins and Geoff Potter at the Department of Linguistics for the loan of their Laryngograph. I'd also like to thank Andrew Senior and David J., for teaching me shell script, Mark G., for help with HMMs, Kate for help with latex, and everyone else that I pestered at some time or another when I got stuck. Thanks to Gavin, for letting me steal his thesis latex style files! Thanks must also go to Richard Prager, Patrick Gosling, Andrew Gee, Valtcho Valtchev, Carl, and everyone else connected in any way with maintaining the excellent SVR computing system. Thanks also to the staff of CUED library, for always being so helpful.

Special thanks must go to all those people who helped keep me sane while all this was happening. In particular to Gavin and Rachel, for making me wear some very silly clothes, to Tina for being Tina, to Hsue-Hueh for smiling, to Kate and Chris for hashing, and to Mark, Rose, Andrew, Steve, Tania, Carline, Anna and my brother for the numerous hill-walking trips.

Finally, thanks to the Engineering and Physical Sciences Research Council of the British Government for funding this work, and to Cambridge University Engineering Department, Christ's College, the University Access funds, the SVR group's HTK Travel Fund, and Cambridge Philosophical Society, for extra grants along the way.

To my parents

This 65,000 word dissertation is entirely the result of my own work and includes nothing which is the outcome of work done in collaboration.

Robert Donovan
Christ's College
Cambridge
June 12, 1996

Contents

Chapter 1

Introduction

1.1 A Definition

This thesis is concerned with the synthesis of speech. Specifically, it is concerned with the synthesis of speech containing different words, and/or different word orders, from any speech stored with or transmitted to the synthesis system. In this thesis such systems are referred to as arbitrary speech synthesis (ASS) systems.

1.2 Applications of ASS Systems

When considering the possible applications of ASS systems, it is important to distinguish those cases in which a system which can generate arbitrary speech is needed, from those in which a few recorded utterances would suffice. The advantages of recorded utterances are that they are currently still more intelligible, more natural, and considerably more engaging than the speech of any ASS system. However, as described below, it is not always possible, and often not desirable, to use a recorded utterance based system, and then an ASS system is required.

For some applications the sentences to be synthesised are not, and cannot, be known in advance. For these the only automated solution possible is an ASS system. Examples are the proof-reading of documents, both reading and speaking aids for the disabled, and devices to read messages, such as electronic mail, over telephone lines. Applications also exist in language teaching, where ASS systems could demonstrate the correct pronunciations of both words and arbitrary phrases. Other future possibilities include speech output for automatic translation systems and intelligent machines.

For other applications the sentences to be synthesised may all be known in advance, and in these the prior recording of utterances is possible. However, an ASS solution may still be preferable. The number of sentences to be synthesised and the rate at which they need to be updated are the determining factors. For applications involving small numbers of sentences, for example a voice alarm system, or sentences which are unlikely to change over time, such as those used in a CD-ROM based multi-media encyclopedia, recordings may be the best solution. However, for applications involving access to very large or rapidly changing databases an ASS system driven from a text database has many advantages. With an ASS system the need to record all the utterances in advance, and to maintain this set of utterances, is replaced by the much simpler task of preparing and maintaining a text database. Individual words can be changed without having to re-record whole phrases, and the original speaker never needs to be recalled. Furthermore, text is a much more compact storage medium than speech.

One example of a database-access application, which is already in use (Sorin and Gagnoulet 1995), is a system which enables mail order catalogue product descriptions to be obtained over the telephone. Also tried are a system for use in libraries, which telephones borrowers when their books are overdue, and a reverse directory enquiries system, from which the name and address corresponding to a particular telephone number can be obtained over the telephone. There are also possible applications in situations where information needs to be given to a person whose eyes are otherwise engaged. For example, a system can be envisaged which would inform a driver of traffic problems relevant to his/her journey, or dynamically give directions to his/her destination.

Another set of possible applications arises from the increasing capacity of ASS systems to mimic the voice of a particular speaker. As this technology comes to fruition, celebrity ASS systems, or systems which sound like the members of a users family, become possibilities. Integration of ASS systems with automatic speech recognition systems would open the way to voice-driven voice mimicking. Such systems would have great entertainment value, enabling, for example, the construction of celebrity voice Kareoke systems. They could also be used to produce voice-overs for television commercials, or as insurance policies for film companies, or even individuals, against somebody losing their voice. With the advent of digital audio broadcasting providing widespread access to high quality signals of the speech of politicians, military personnel, and celebrities, such systems also have a large potential for abuse. This is an issue which must be faced both by the companies supplying such technology and by society in general.

The number of possible applications for ASS technology has greatly increased in recent years, largely due to the rapid increase in the use of computers in society. With many more potential users than was hitherto the case, research into improving the intelligibility and naturalness of ASS systems is therefore perhaps more important now than ever before.

1.3 Generic ASS System Structure

All ASS systems require some input specification of the speech to be synthesised. Most frequently the input specification used is the text of the desired utterance, in ASCII form. The process of converting this input to speech is known as Text-to-Speech (TTS) synthesis. For many of the applications described in Section 1.2 this is exactly what is required, since the text is all that is known about the required speech. For some applications however, such as those in which the phrases to be synthesised are constructed by a computer, more information may be available about the required speech than just the corresponding text. In these cases, this additional information can be passed from the underlying system to the speech synthesis system, instead of having a TTS system attempt to generate the same information from the corresponding text. Indeed, (Young and Fallside 1979) suggested that the synthesis system should be passed a “concept”, from which it would construct both the required sentence and the synthetic speech. These approaches can simplify (or avoid) many of the problems involved in synthesising arbitrary speech from text. However, since TTS synthesis is required in many applications, this chapter discusses all the issues relevant to TTS conversion.

In order to be able to synthesise arbitrary speech, all ASS systems must make use of some set of sub-phrase synthesis units to construct the acoustic realisation of each phrase. The set used, and the manner in which it is used, varies between different synthesis systems. The existence of these units results in all TTS systems being essentially composed of three sub-systems, although there is some overlap between these. These sub-systems are, the conversion of arbitrary text into synthesis units (see Section 1.8), the conversion of arbitrary text into prosodic parameters (comprising segmental durations, fundamental frequency contours, and energy contours, see Section 1.9), and the conversion of synthesis units and prosodic parameters into speech. The different synthesis techniques which can be used to generate synthetic speech are discussed in Section 1.4, and the different types of system into which these techniques are incorporated in Sections 1.5, 1.6, and 1.7.

1.3.1 Trainable ASS Systems

Trainable approaches to automatic speech recognition (ASR), aided by the increasing power of modern computers, have had a great deal of success in recent years. Typically, these methods involve creating an appropriate model and then estimating its parameters automatically through training on a suitable database. Their success stems from their ability to capture the regularities and variations present in large training databases through the optimisation of well understood criteria. They have largely replaced rule-based approaches to ASR, which were difficult to optimise, and usually based on human analysis of substantially smaller amounts of data. These considerations suggest that applying similar methods to the problem of speech synthesis might have a similar degree of success.

The attraction of using trainable, data-based, methods for speech synthesis, besides the improvements in performance which they may bring, is the flexibility that such systems would have with regard to changing voices or languages. A synthesis system constructed automatically could be adapted to a *specific* new voice, or even a new language, simply by retraining on new data. Such a system would represent a considerable advance over more traditional approaches to speech synthesis. These require extensive new work to achieve such changes (although less precise voice alterations can be more easily obtained with some systems).

The need to develop ASS systems in several languages is obvious; even amongst the developed nations likely to be using ASS technology there are many different languages in use. The need for many different voices is perhaps less obvious. For database access applications (Sorin 1994) states that at least two voices per application are necessary, one for welcome and guidance messages, the other for information delivery. The requirement then increases many-fold since it would be undesirable to have the same synthetic voices used in every database access system. It is also likely that manufacturers would wish different products to have different voices. (Murray and Arnott 1993) cite the need for multiple voices to avoid confusion when different people using TTS systems as speaking aids are in the same room. Finally, the voice mimicking applications described in Section 1.2 clearly

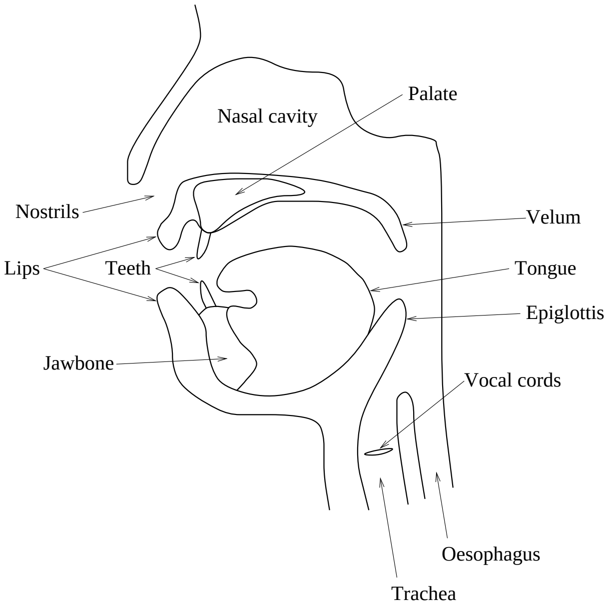


Figure 1.1: The Human Speech Production System.

require specific voices to be produced.

In recent years, research has begun to investigate the use of data-based methods in speech synthesis. Attempts to use data-based methods to solve the problems associated with the conversion of text to synthesis units and prosodic parameters are described in Sections 1.8.2, 1.9.1, 1.9.2 and 1.9.3. The use of data-based methods to select synthesis units for use in a concatenation synthesiser is the principle subject of this thesis, and previous research in this area is therefore discussed in detail in Chapter 2.

1.4 Synthesis Techniques

This section describes the human speech production system, and both historical and present techniques for synthesising speech.

1.4.1 The Human Speech Production System

The human speech production system is illustrated in Figure 1.1. The vocal tract extends from the vocal cords in the throat to the lips in the oral cavity, and the nostrils in the nasal cavity. The shape is modified by the position of the *articulators*, namely the velum, the jaw, the tongue, and the lips. The shape determines the transfer function of the vocal tract response to an excitation signal. This transfer function is usually composed of a number of resonances, known as *formants*, and occasionally of anti-resonances too.

Speech is produced by exciting the resonances and anti-resonances of the vocal tract filter. The excitation either comes from the vibration of the vocal cords during *voiced* speech, at the *fundamental frequency,* F_0, or from turbulent noise created at a constriction somewhere in the vocal tract in the case of *unvoiced* speech. In some sounds both types of excitation may be present at the same time.

The positions of the articulators and the type of excitation signal vary during speech production. It is believed that their positions/status are the result of a constant movement towards a sequence of rapidly changing targets, subject to their dynamical constraints. These targets are defined by the *phonemes* and the *allophones* of a given language.

Phonemes are linguistically defined units of speech. They are essentially labels given to each group of vocal tract articulator targets which are considered to be functionally equivalent by speakers of a given language. A working definition involves the use of *minimal-pairs*. Two vocal tract configurations represent different phonemes if two words can be found which differ only by the use of these two configurations. For example, *rice* and *lice* differ only in their initial vocal tract configurations, but are different words, and therefore contain different phonemes. The number of phonemes in a language varies between about 20 and 60, with English having about 40.

Allophones represent finer configuration distinctions, which although not functionally distinctive in a language, may be discriminated between by speakers of the language. For example, *lice* contains a *light /l/*, and *small* contains a *dark /l/*, which have different vocal tract configurations. Although producing each word with the wrong allophone would sound strange, they would still be the same words, and hence the light and dark versions of /l/ are not different phonemes.

Finally, *phones* are the acoustic realisations of phonemes, and vary considerably both with context and between speakers. In this thesis the term *phone* is used slightly more generally to refer to both realisations of, and classes of, speech sounds which may or may not be true phonemes. For example, in Chapter 6, the phones /tcl/ and /tbst/ are introduced to represent the closure and burst parts of the phoneme /t/.

For a more detailed description of the human speech production process see (O'Connor 1973), (Borden and Harris 1984), or any good book on articulatory phonetics.

1.4.2 An Introduction to Synthesis Techniques

The first attempt to synthesise speech was by Wolfgang von Kempelen in the 18th century. His talking machine was a mechanical device powered by bellows. It had a mouth made of india-rubber, nostrils, and an inflatable side bellows to simulate the expansion of the vocal tract. Voicing was achieved by the vibration of a reed, and fricatives by allowing air to escape in various ways. The device was operated by using levers, by covering the nostrils, and by shaping the mouth, and could synthesise whole phrases in French and Italian.

Attempts to synthesise speech mechanically gave way, in this century, first to analogue-electronic methods, and then to computer-based methods. The first electronic synthesiser was developed by Dudley in the 1930s. His *Voder* was based on a channel vocoder, and synthesised speech by using ten potentiometers to control the gains of ten fixed frequency resonators, which were excited by pulses at the pitch frequency or by noise. The potentiometers were controlled manually, to enable the synthesis of arbitrary speech. Research into electronic methods of synthesising speech continued, both into making direct electronic analogues of the vocal tract, and into implementing both channel and formant synthesisers electronically. For a thorough review of both mechanical and electronic methods of speech synthesis see (Linggard 1985).

In the second half of this century methods employing analogue-electronics to synthesise speech were largely abandoned in favour of methods using digital computers. The broadest subdivision of the strategies used to synthesise speech on computers is into *system-models* which attempt to model the human speech production system, and *signal-models* which attempt only to model the resulting speech signal. The system-model approach is known as *articulatory synthesis*, and is discussed in Section 1.5. The signal-model approach is perhaps the simpler of the two, and as such has been both more thoroughly investigated, and more successful. It can be further subdivided into methods broadly described as *rule-based formant synthesis*, and *concatenation synthesis*.

Rule-based formant synthesis systems were for many years the most successful methods of synthesising speech, and are discussed in Section 1.6. Formant synthesisers use an excitation signal to excite a digital filter constructed from a number of resonances similar to the formants of the vocal tract (see Section 1.4.4). The separation of the vocal tract transfer function and the excitation signal in this way is known as the *source-filter theory* of speech production, and is described in Section 1.4.3.

Concatenation synthesis operates by concatenating appropriate synthesis units to construct the required speech. Section 1.7 describes the various systems which have been investigated, and the different types of synthesis unit which they have used. In these systems signal processing must be applied to alter the fundamental frequencies and durations of the synthesis units to those required in the synthetic speech. Furthermore, unless the units are selected very carefully, the signal processing must also be able to smooth away spectral concatenation discontinuities between units. Two forms of signal processing, and their variations, have been used extensively in concatenative systems. These are *Linear Prediction* (LP) synthesis (see Section 1.4.5), and *Pitch Synchronous Overlap and Add* (PSOLA) synthesis (see Section 1.4.6). LP synthesis is a source-filter based approach, but PSOLA is not and operates simply by windowing and recombining existing synthesis unit waveforms.

1.4.3 The Source-Filter Theory of Speech Production

The source-filter theory of speech production assumes that the excitation source can be considered to be independent from the vocal tract response. In practice, the vocal tract response is usually assumed to be linear, and the z-transform of the speech signal, $S(z)$, can therefore be synthesised as

$$S(z) = U(z)H(z) \tag{1.1}$$

where $U(z)$ is an approximation to the excitation signal, and $H(z)$ the transfer function of a digital filter representing the vocal tract response and the radiation characteristic of the lips/nostrils.

In practice, $H(z)$ is often analysed as $V(z)R(z)$, where $V(z)$ is the transfer function of the vocal tract, and $R(z)$ the radiation characteristic. Furthermore $U(z)$ is often analysed as $P(z)G(z)$, where $P(z)$ is a pulse train and/or white noise, and $G(z)$ (which is only present for voiced speech) is the transfer function of the glottal waveform "filter". For voiced speech equation 1.1 then becomes

$$S(z) = P(z)G(z)V(z)R(z). \tag{1.2}$$

Thus in linear prediction (LP) synthesis, for example, the excitation signal used is often just $P(z)$, and the filter implied by the LP coefficients models the combined effects of the shape of the glottal waveform, the vocal tract response, and the radiation characteristic.

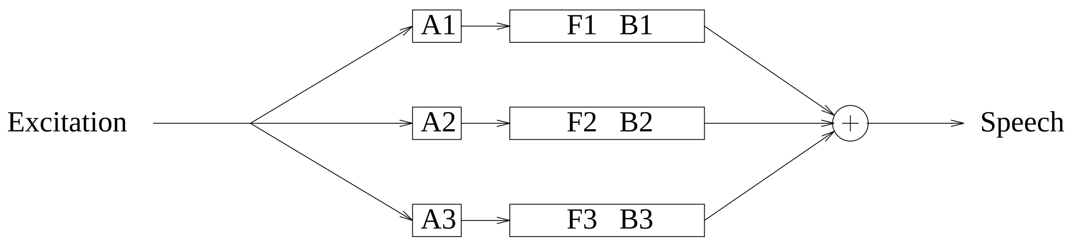
In synthesis applications the digital filter which includes the vocal tract response is usually updated only every 5 or 10 milliseconds. This represents an assumption that speech can reasonably be represented by a single transfer function on such time-scales. This assumption is usually valid, because the articulators of the vocal tract are usually almost stationary over such time-scales. However, for very rapidly articulated sounds, such as bursts, the assumption is more questionable.

The assumption that the source and filter are independent is only approximately true. (Klatt 1987) reports that the resonances of the vocal tract lead to standing pressure waves which can affect both the vibration pattern of the vocal cords, and the shape of the glottal waveform. Similarly, the opening and closing of the vocal cords represents a varying termination impedance for the vocal tract, and can affect its transfer function. These effects are generally small, and Klatt postulates that they may only be of importance during moments of voicing onset, and in causing small pitch-synchronous changes to the first formant. In female voices the vocal cords usually spend a larger fraction of each pitch period open, and with some voices this can lead to the vocal tract interacting with the trachea. In this case the independence assumption is less reliable; indeed (Sorin 1994) states that the assumption is "grossly inadequate" for female speech.

1.4.4 Formant Synthesis

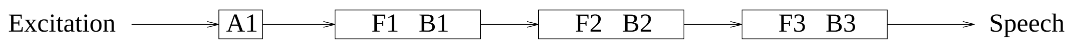
Formant synthesis is a source-filter method of speech production, in which the vocal tract filter is constructed from a number of resonances similar to the formants of natural speech. It is therefore similar to the Dudley *Voder* described in Section 1.4.2, but uses a small number of variable frequency resonators, instead of many fixed frequency resonators. Up to three formants are generally required to synthesise intelligible speech, with four or five being sufficient to produce high quality speech. Each formant is usually modelled using a two pole resonator, which enables both the formant frequency and its bandwidth to be specified. There are two methods of combining the formants to make a model of the vocal tract. In the *parallel formant synthesiser* the excitation is applied to all the formants in parallel and their outputs are summed, enabling individual gains to be specified for each formant. In the *cascade formant synthesiser* the output of one formant is applied to the input of the next. The two forms are shown schematically in Figure 1.2.

There has been much debate over which of the two arrangements in Figure 1.2 is better. (Linggard 1985) summarises that the cascade type has been found to be better for non-nasal voiced sounds, and the parallel type for nasals, fricatives and stops. Efforts to improve on the simple systems shown in Figure 1.2 have also been made. In (Klatt 1980) FORTRAN listings were published for a complex formant synthesiser which incorporated both the cascade and parallel formant arrangements. It also had additional resonances and anti-resonances to aid in the synthesis of nasalised sounds, a sixth formant for synthesising



A1, A2, A3 : Amplitude Scaling Parameters
F1, F2, F3 : Formant Frequencies
B1, B2, B3 : Formant Bandwidths

(a) A Parallel Formant Synthesiser



A1 : Filter Gain
F1, F2, F3 : Formant Frequencies
B1, B2, B3 : Formant Bandwidths

(b) A Cascade Formant Synthesiser

Figure 1.2: Generic Formant Synthesisers.

very high frequency noise, a bypass path to give a flat transfer function, and a radiation characteristic. The system used a complex excitation model, and was controlled by 39 parameters, which were updated every 5ms. The synthesiser could synthesise very high quality speech, and has since been incorporated into several TTS systems (see Section 1.6), and been used by many researchers in their work.

With formant synthesisers, the digital filter specified by the formants usually seeks only to represent the resonances of the vocal tract, and so additional provision is needed for the effects of the shape of the glottal waveform and the radiation characteristic. The radiation characteristic is often approximated as a simple +6dB/octave filter on the output. While the glottal waveform is often approximated by a −12dB/octave filter, natural waveforms often differ from this ideal. For example, spectral zeros are usually present, the open period, abruptness of closure, amount of breathiness, spectral tilt and many other features can vary between speakers and with time. This has led to the creation of complicated voicing models, which enable many of these features to be varied, (Klatt 1987). However, Klatt also stated that the rules necessary to control these complex models were still quite primitive in 1987.

An important demonstration of the capabilities of formant synthesisers was reported in (Holmes 1973). In this experiment people inexperienced in listening to synthetic speech were found to be unable to distinguish a natural utterance and a synthetic utterance based upon it, even when using earphones. The synthetic utterance was generated by manually tuning the control parameters of a parallel formant synthesiser to reproduce the natural utterance. An inverse-filtered non-nasalised vowel waveform was used as the voicing excitation. These results may at first seem to call into question the need for the complex voicing models discussed above. However, (Klatt 1987) suggested that Holmes had effectively modelled the change over time of the voiced excitation of the natural utterance by careful control of the formant amplitudes.

Formant synthesisers are generally controlled by rule (see Section 1.6). Whilst they are theoretically capable of being used in concatenative systems, they have rarely been so because of the difficulties associated with estimating formant parameters automatically from concatenation units.

1.4.5 Linear Prediction Synthesis

Linear Prediction (LP) synthesis is another source-filter method of speech synthesis. The digital filter is estimated automatically from a frame of natural speech using a computationally efficient algorithm. LP synthesis has been used extensively in concatenation systems, since it enables the rapid coding of concatenation units. It is not really suited to rule-based systems, since rules are most easily specified in terms of formants, and the relationship between the coefficients used to define the LP filter and formants is not a simple one. LP synthesis was used in the current work, and the mathematical theory is therefore described in some detail in Appendix D. This section presents an introduction to the concepts involved, and a discussion of the limitations of the model.

The basis of linear prediction theory is the assumption that the current speech sample $y(n)$ can be predicted as a linear combination of the previous P samples of speech, plus a small error term $e(n)$. Thus,

$$e(n) = \sum_{i=0}^{P} a(i) y(n-i) \qquad \text{where } a(0) = 1, \tag{1.3}$$

and the $a(i)$ are termed the *linear prediction coefficients*, and P the *linear prediction order*. The LP coefficients, $a(i)$, are found by minimising the sum of the squared errors over the frame of speech under analysis. Two methods of performing this calculation are commonly used, termed the *covariance method* and the *autocorrelation method*, which differ in the range of n over which the error is minimised (see Appendix D). Coefficients calculated using the autocorrelation method have the advantage that the filter they define is guaranteed to be stable, (Markel and Gray 1976).

It can be shown that LP analysis is equivalent to matching the power spectrum of the all-pole filter defined by the LP coefficients (see Appendix D) to the spectrum of the speech signal. This matching is effectively weighted to achieve the most accuracy in the vicinity of the formant peaks, (Markel and Gray 1976). The digital filter thus models the spectral envelope of the speech signal, and the error signal $e(n)$ (ideally) contains only the harmonic structure of the speech and/or white noise (P(z) in Section 1.4.3). The speech can therefore be re-synthesised at a different fundamental frequency by exciting the filter with a synthetic error signal, providing a new harmonic structure. The stylised excitation often used consists of a simple pulse train at the new fundamental frequency for voiced speech and white noise for unvoiced speech.

The autocorrelation method derived LP coefficients can be reversibly transformed into related parameters called *reflection coefficients*, and *area-functions*, which are so named because the autocorrelation approach can be shown to be analogous to modelling the vocal tract as an acoustical tube of varying cross-section. The reflection coefficient form is particularly useful, because these parameters have the property that a filter defined by a set of reflection coefficients $k_n \;\; 1 \leq n \leq P$ is guaranteed to be stable if $k_n < 1 \;\; \forall n$, (Markel and Gray 1976). Parameter interpolation between stable filters is thus guaranteed to give a stable filter. Reflection coefficients are therefore very useful in smoothing away concatenation discontinuities (see Section 5.6.1). *Lattice* filters (see Section 5.4) enable reflection coefficients to be used directly, without conversion to their LP coefficient equivalents. For more details on LP theory see Appendix D, (Markel and Gray 1976), or any good speech processing textbook, for example (Parsons 1986).

Synthetic speech produced using linear prediction synthesis is far from perfect. (Klatt 1987) reports that autocorrelation method LP synthesis does not reproduce formant frequencies and bandwidths correctly when speech is re-synthesised at a different fundamental frequency to that which it had originally. Even when re-synthesising speech at the original pitch, the speech quality is considerably degraded compared to the original. This is because the stylised excitation used in synthesis is actually an over-simplification of the true error signal, particularly for voiced speech. The true error signal contains additional information to correct for the departure of the speech signal from the assumption of LP theory that the speech within each frame can be modelled by a single all-pole filter. Re-synthesising without this additional information therefore introduces degradation. The most noticeable result is that the synthetic speech is produced with a characteristic buzz. Alternative excitation models have been sought in the hope of reducing this effect, for example, the current AT&T speech synthesiser uses a voicing source model which enables both the spectral balance and the degree of aspiration to be varied, (Sproat and Olive 1995). The all-pole assumption is particularly poor for nasals, and nasalised vowels, which contain spectral zeros, and hence these sounds are not well reproduced by LP synthesis. The model is also particularly poor for many plosives, because the time-scale of events within them can be shorter than the frame sizes used for analysis. They are therefore often poorly reproduced when using a stylised excitation.

A development of LP theory, called *multi-pulse* linear prediction, (Atal and Remde 1982), can solve many of the problems described above. The method constructs a complex excitation consisting of several pulses for each frame of speech analysed, which when combined with the LP coefficients for that frame reproduces almost exactly the original waveform. This is very useful for vocoding, and storage, since it enables very high quality speech to be generated at a reduced bit rate. However, the source and the filter are no longer separate, and hence difficulties arise in ASS applications with both altering prosody and ensuring waveform and spectral continuity at concatenation unit boundaries. (Stella and Charpentier 1985) used multi-pulse linear prediction to code their diphone inventory, using pitch synchronous frames. However, during synthesis the target utterance was synthesised using normal multi-pulse LP synthesis, carrying over the LP filter memories at segment boundaries. The synthesised utterance, which contained the intrinsic prosody of its component diphones, was then subsequently transformed to have the required prosody using a phase-vocoder. Hence, multi-pulse coding was only used as a form of database compression. In contrast, (Varga and Fallside 1987) used multi-pulse methods, and an automatically obtained pitch synchronous labelling, to exactly reproduce each pitch pulse of each concatenation unit during synthesis. Pitch was lowered by allowing the filter associated with each pitch pulse to run on with no excitation, and raised by truncating pitch pulses. Durations were altered by repeating and removing whole pitch pulses. The boundaries between pitch pulses, and between units, were smoothed in the time domain to produce a continuous signal. However, spectral smoothing at concatenation boundaries was not possible. The authors claimed that the prosody modification technique gave excellent results when used for analysis-synthesis of whole utterances, and that good results were obtained in concatenation synthesis.

Other extensions of LP synthesis are *residual excited* linear prediction (RELP), in which the error signal, or residual, is used as the excitation signal, and *codebook excited* linear prediction (CELP), in which one of a number of signals stored in a finite codebook is used as the excitation signal. Both suffer similar problems to multi-pulse LP methods, because the source and filter are no longer completely separated. Nevertheless, these problems have been overcome, and RELP synthesis is used in L&H's commercial TTS system, (Lernout & Hauspie 1996).

1.4.6 PSOLA Synthesis

The Pitch Synchronous Overlap and Add (PSOLA) algorithm was developed by France Telecom at CNET, (Charpentier and Stella 1986). The technique does not synthesise speech itself, but merely enables pre-recorded segments of speech to be smoothly concatenated, while enabling the pitch and duration of the segments to be altered. It is therefore of use in concatenation synthesis in place of linear prediction, which was traditionally used to perform this role. The advantage of PSOLA synthesis over LP synthesis is that the synthetic speech produced is of a much higher quality. The advantages and disadvantages of the various versions of PSOLA which have been developed are discussed below.

All versions of the PSOLA algorithm work in essentially the same way. A natural speech segment is broken into many short-term (ST) signals, by Hanning windowing pitch-synchronously through regions of voiced speech and at a fixed interval through regions of unvoiced speech. The ST-signals are then re-combined to produce the synthetic speech. The size of the Hanning window used has implications for the synthetic speech quality, and is discussed below. The pitch is raised or lowered by altering the spacing of the ST-signals during synthesis, and the duration simultaneously altered by repeating or deleting ST-signals from the synthetic speech. The recombination is performed using one of several overlap-add schemes which add together the new ST-signal sequence at the new spacing. These schemes compensate for the number and amplitude of the Hanning windows contributing to the synthetic signal at each point in time. The most complicated of these, the least-square scheme, goes further, and tries to minimises the error between the ST-signal spectra and the corresponding short-time spectra of the synthetic speech.

The simplest version of PSOLA is time domain, or *TD*-PSOLA, which proceeds exactly as just described. TD-PSOLA is the most computationally efficient version of PSOLA. It was used in the work described in this thesis, and a detailed description of the implementation can be found in Section 7.3. The synthetic speech quality obtained with TD-PSOLA synthesis is far superior to that obtained with LP synthesis. However, this improvement is not without cost. All versions of PSOLA require large amounts of storage for the concatenation unit waveform databases used, although in practice this problem can be reduced by compressing the speech, using, for example, multi-pulse coding. TD-PSOLA also has the disadvantage, compared to LP synthesis, that spectral smoothing at concatenation unit boundaries cannot be performed. Synthesis units must therefore be chosen very carefully if formant discontinuities are to be avoided during synthesis.

As mentioned above, the speech quality obtained with TD-PSOLA is very good. However, it is not perfect, and localised errors can be quite distinct precisely because the general quality is so good. A major problem occurs when significantly increasing the durations of unvoiced sounds. The repetition of unvoiced ST-signals can result in a local periodicity which is heard as a tonal noise. This problem can be largely overcome for purely unvoiced sounds by reversing the time axis of the repeated ST-signal, but this technique cannot be applied to voiced fricatives, which can suffer similar, though less severe, problems. Less localised problems also exist. If large Hanning windows are used (containing multiple pitch pulses), then a mismatch occurs during synthesis between the imposed synthesis pitch frequency, and the inherent pitch frequency contained in each ST-signal. This results in selective alteration of the amplitudes of pitch harmonics in voiced speech, which is heard as reverberation in the synthetic speech. Alternatively, if small Hanning windows are used, these problems are much reduced. However in this case formant bandwidths are broadened in the synthetic speech, since the estimate of the spectral envelope implicit in each ST-signal has a poor frequency resolution due to the shortness of the analysis frames, and the presence of the Hanning Window. Localised problems may also arise in the latter case if the ST-signals are not centred on the moments of principle excitation of the vocal tract (normally the moments of glottal closure).

Some of the problems just described can be overcome by using the frequency domain version of PSOLA, *FD*-PSOLA. In this approach a global spectral envelope is obtained for each ST-signal using, for example, LP techniques, and an estimate of the source spectrum obtained by dividing the Discrete Fourier Transform of the ST-signal by this global spectral envelope. The source spectrum can then be modified to match the synthesis pitch frequency required, in order to remove the mis-match described above. The spectral envelope can also be modified to alter the voice quality, or smooth concatenation unit boundaries. After the required modifications, the two spectra are recombined and an inverse Fourier Transform applied to generate a synthesis ST-signal, which is then treated as before. FD-PSOLA requires considerably more computation than TD-PSOLA.

The *LP*-PSOLA technique is a hybrid of TD-PSOLA and LP methods. In this approach the TD-PSOLA algorithm is applied to the LP residual, or to multi-pulse or CELP coded versions of it, instead of to the speech waveform. The chief advantage of LP-PSOLA is that different window sizes can be used to estimate the spectral envelope (the LP coefficients) than are used to perform the prosodic modifications. Large windows can be used to estimate the spectral envelope in order to obtain accurate formant bandwidths. Small windows can be used to define the ST-signals used to perform the prosodic modifications in order to reduce the synthesis frequency mis-match problem. The use of small windows for the ST-signals is sufficient because the excitation spectrum resonances are usually very broad, and so are not degraded by the lack of resolution. LP-PSOLA also enables the spectral envelope to be smoothed at concatenation unit boundaries. More details of the TD, FD, and LP versions of the PSOLA algorithm can be found in (Moulines and Charpentier 1990).

A later version of PSOLA, called Multi-Band Re-synthesis PSOLA, or *MBR*-PSOLA, was developed by (Dutoit and Leich 1993). In this approach, the segment inventory used in synthesis is altered using a computationally expensive Multi-Band-Excited (MBE) analysis-synthesis procedure, in order to make it more suited for synthesis using the TD-PSOLA algorithm. Specifically, all segments are re-synthesised to have the same constant pitch, with the new pitch-marks, normally the moments in the speech signal corresponding to the moments of principle excitation of the vocal tract, imposed by a phase reset procedure. This removes the problem of trying to locate the pitch-marks in the segment inventory, and reduces the discontinuity problems which can otherwise arise when concatenating spectrally similar segments of speech with very different pitches. Furthermore, and perhaps most importantly of all, the constant pitch and phase reset procedure enable spectral interpolation at concatenation unit boundaries to be achieved directly by the simple interpolation of the time waveforms. This enables the concatenation discontinuities associated with simple diphone inventories to be largely overcome without the use of polyphone units (see Section 1.7.4). However, the speech quality is not as good as that obtained with TD-PSOLA and well chosen units.

1.5 Articulatory Synthesis

Articulatory synthesisers attempt to produce speech by modelling the human speech production system. They typically involve models of the human articulators and vocal cords. These models are moved towards target positions for each phoneme using rules. The rules reflect the dynamical constraints imposed upon the articulators by their masses and associated muscles. In order to generate speech the shape of the vocal tract defined by the positions of the articulators is usually converted into a transfer function, for example by estimating area functions or formant frequencies, (Coker 1976). The vocal cord model may be similarly used to generate an appropriate excitation signal. The synthesis problem is thus converted into one of specifying articulator targets for each phoneme, and accurately modelling the articulators' dynamics. (Klatt 1987) suggests that the latter is the major problem with this form of synthesis, mainly due to a lack of data.

Although articulatory synthesis is perhaps the most satisfying method of speech synthesis, since it models the human system directly, it has received less attention than signal based methods and has not yet achieved the same level of success.

1.6 Rule-based Formant Synthesis

Rule-based formant synthesis is a very successful method of synthesising speech. A set of rules is used to determine the parameters necessary to synthesise a desired utterance using a formant synthesiser. The rules are generally used in conjunction with a phoneme string specification of the desired utterance. They determine which allophones to use in which phonetic and wider contexts (for example, morpheme[1] boundaries, word boundaries, and stress level may all have important effects), and specify exactly how these allophones, and the transitions between them, should be produced.

The first attempt to construct a rule-based formant synthesis system was by (Kelly and Gerstman 1961). A simple three-formant synthesiser was used together with rules based on results from their own experiments with generating control signals by hand, and "known results in speech perception". Early success was also reported by (Holmes et al. 1964). In this system each phoneme was composed of one or more "phonetic elements". Synthesis parameters were stored for the steady state regions of these phonetic elements in tables. The elements were ranked in order of their dominance during transitions, and then the parameters of the dominant element used to calculate the synthesis parameters required to produce transitions between elements. The synthesis parameters were used to drive an electronic formant synthesiser via punched tape. The speech quality and intelligibility of this system were reported by (Klatt 1987) as "remarkably good", though intelligibility tests were never conducted.

Research into rule-based formant synthesis continued, (see (Klatt 1987) for a good review of work in this field) and eventually led to some very high quality TTS systems. These included MITalk, (Allen et al. 1987), the Infovox SA-101, (Magnusson et al. 1984), the Prose-2000, (Groner et al. 1982), and Klattalk, (Klatt 1982), which was licensed to Digital Equipment Corporation to become DECtalk, (Bruckert et al. 1983). Note that MITalk, the Prose-2000, Klattalk and DECtalk all used a (simplified in some cases) version of the Klatt formant synthesiser, (Klatt 1980). The DECtalk system for English was for many years, and possibly still is, the standard by which new systems are judged.

1.7 Concatenation Synthesis

In concatenation systems the existence of synthesis units is explicit, and one or more representations of each unit is stored for use in synthesis. A trade-off exists between longer and shorter units. Longer units are advantageous compared to shorter units because they preserve naturalness over longer time-scales, and result in fewer concatenation points in the synthetic speech. Furthermore, with many types of longer unit the concatenation discontinuities which occur at these points are often relatively small. However, longer units can also be very numerous, to the point of being prohibitively numerous in the case of words, and therefore shorter units, which are much less numerous, are also attractive.

All forms of signal processing introduce degradation, the extent of which usually scales with the amount of modification required. The ideal concatenation system would therefore synthesise each utterance using the set of units which most accurately produced that

[1] Morphemes are the smallest meaningful units of language, see Section 1.8.2

utterance, both spectrally and prosodically, without signal processing. This set of units would comprise as many long units as possible, which would concatenate as smoothly as possible, and preferably be selected from a database optimised to contain those units which gave the best performance on the task for which the ASS system was being used. Such a system does not yet exist, although the augmented diphone systems described in Section 1.7.4, some of the automatic segment inventory construction algorithms described in Chapter 2, and the work described in this thesis, all go some way towards this ideal.

Traditional approaches to concatenation synthesis used manually prepared sets of synthesis units, usually all of the same class. These units were usually either words, syllables, demi-syllables, or diphones, and are discussed below in Sections 1.7.1 to 1.7.4. The use of phone and sub-phone units for concatenation synthesis is mentioned briefly below in Sections 1.7.5 and 1.7.6, and examined in detail in Chapter 2.

1.7.1 Words

The most obvious synthesis unit to choose, and that most often suggested by people not working in the field, is the word. The advantage of using words is that all the within word co-articulation effects are captured in the stored units. Concatenating words is then relatively easy, compared with sub-word synthesis units, because between word co-articulation is usually weaker than within word co-articulation, (Linggard 1985). However, simply concatenating the waveforms of words recorded in isolation produces speech which is very difficult to understand. This is mainly due to the pitch and formant discontinuities at word boundaries, and these problems can be largely solved using signal processing. Further problems arise because words spoken in isolation are much longer than words in sentences, and the acoustic realisation, and even the phonetic realisation, of words in sentences varies with context. To achieve high levels of naturalness it may therefore be necessary to record multiple versions of each word, spoken in different contexts.

(Rabiner et al. 1971) used formant synthesis to enable the pitch, duration, and inter-word formant discontinuity problems to be solved, and reported encouraging results with synthesising telephone numbers. A larger 300 word vocabulary was used in (Fallside and Young 1978), in which LP synthesis was used to perform the signal processing. Multiple versions of some words were stored, durations were shortened, pitch contours were applied, and various techniques, including the smoothing of LP area coefficients, were used to overcome the discontinuities at word boundaries. Again, encouraging results were reported, though the speech was said to lack rhythm. Ten digits, or 300 words, neither present a major recording problem, nor a major storage problem with current computer technology. However, an ASS system requires a very large vocabulary, and in this case the recording and storage problems become formidable. When proper names, foreign words, and new words are included, the problems become insurmountable. It is this limitation which has motivated researchers to look for shorter, less numerous, synthesis units.

1.7.2 Syllables

The use of syllables as synthesis units represents a halfway stage between words and smaller phone-sized units. Again, as with words, the advantage is that the relatively long synthesis units preserve within unit co-articulation. However, unlike words, the between unit co-articulation is not necessarily weaker than the within unit co-articulation, and so smoothing across unit boundaries is not so easy. There are approximately 10,000 syllables in English, which still presents significant recording and storage problems. Given that difficult co-articulation problems must be handled anyway, it would seem sensible to seek alternative units which are both less numerous, and/or more appropriate for concatenation. (Allen et al. 1987) stated that there were no syllable-based concatenation synthesis systems at that time.

1.7.3 Demi-syllables

Demi-syllables represent another step down in unit size, being the initial and final halves of syllables. The advantage of demi-syllables is that only approximately 1,000 are needed to construct the 10,000 syllables of English, (Lovins et al. 1979), and so recording and storing them is both possible and reasonable. A demi-syllable based synthesis system was presented in (Browman 1980) which used LP coded demi-syllables, and twenty rules concerning pitch, duration, and boundary smoothing to concatenate them. Demi-syllables have an advantage over smaller units because they preserve highly co-articulated syllable internal consonant clusters. However, although syllable internal concatenation boundaries are often easily smoothed, co-articulation between syllables can still be problematic. At the time of writing at least one commercial system, the ORATOR TTS system (Bellcore 1996), uses demi-syllable synthesis units.

1.7.4 Diphones

A diphone is roughly the last half of one phone followed by the first half of the next. Diphone units therefore preserve transitions between phones, which are otherwise difficult to produce. The boundaries between diphones during synthesis thus occur in the middle of phones. This tends to result in relatively small concatenation discontinuities because the middles of phones are usually their most spectrally stable regions, and are often relatively spectrally consistent across phonetic contexts. (Peterson et al. 1958) were the first to suggest the use of diphones (dyads) in speech synthesis. In this system speech was synthesised by reproducing each segment unaltered, and up to nine versions of each diphone were required in order to properly model intonation. The authors estimated that a total of approximately 8000 diphones would be needed for American speech. Although the authors envisaged using an electronic synthesiser to reproduce the diphones, the implementation described in (Wang and Peterson 1958) involved the manual splicing together of pieces of audio tape.

A more practical implementation of diphone synthesis was reported in (Dixon and Maxey 1968). A formant synthesiser was used during synthesis to enable diphone durations to be modified, and a pitch track to be applied to the synthetic speech. As a result considerably fewer diphones were needed than was predicted by (Peterson et al. 1958); the authors estimated that the minimum number required was approximately 1000.

Another early attempt at diphone synthesis was made at Bell Labs. by (Olive 1977), using an LP log-area parameter coding. In this system only the end points of the transitions between pairs of phones were stored. During synthesis the transitions were then recreated by linear interpolation between these endpoint values. The steady state of each phone was produced by linear interpolation between the end of one transition and the start of the next. Further simplifications were made for consonant to consonant transitions, resulting in a total of only about 600 transitions to be stored. The durations of the transition and steady state parts of each phone were specified in a pronouncing dictionary for each word in the synthesis vocabulary. Rule generated amplitude and pitch contours were imposed on the synthetic speech during synthesis.

Diphones synthesis systems were also investigated extensively by France Telecom at CNET. (Courbon and Emerard 1982) described a system which used approximately 1200 LP coded diphones to synthesise French. Later, (Stella and Charpentier 1985) used the same set of diphones in a system using multi-pulse coding and a phase-vocoder, described above in Section 1.4.5. Later again, diphones were used in the first PSOLA based system, (Charpentier and Stella 1986). Continuing research, both at Bell Labs. and CNET, sought to supplement the established diphone inventories with selected longer units to improve the synthesis of highly co-articulated phone sequences. These units were typically three or four phones in length and were designed to ensure that rapidly articulated phones, which did not reach their acoustic targets, were embedded between other phones which were more precisely articulated, (Olive 1990), (Bigorgne et al. 1993). At the time of writing, augmented diphone systems form the basis of many leading commercial and research TTS systems, (Sorin 1994), (Sproat and Olive 1995), (Lernout & Hauspie 1996), and simple diphone systems are used by many researchers in their work. In recent years investigations have begun into the automatic segmentation of diphones, and this work is discussed in Section 2.2.2.

1.7.5 Phones

Concatenation synthesis using phone-based segments of speech is difficult due to the large amount of contextual variation in the acoustic realisation of each phoneme, and the consequent problems in selecting appropriate units and ensuring concatenation smoothness. However, automatic techniques to solve these problems have been investigated, with some success, and this research is discussed in detail in Chapter 2.

1.7.6 Sub-Phone Units

In recent years sub-phone units have begun to be investigated for use in speech synthesis. As with phones, these units are also subject to a large degree of contextual variation. Their attraction is that speech generally becomes more acoustically self-similar on these time-scales, and the synthesis units can therefore sensibly be represented by a single vector of spectral parameters. This makes the units easier to work with, and means that state based speech models, such as hidden Markov models (see Chapter 3), can be used. The use of sub-phone units in speech synthesis is discussed further in Section 2.2.3.

1.8 Text to Units

All Text-to-Speech systems require the conversion of arbitrary text into a sequence of synthesis units, which in most practical systems are either phonemes or phoneme dependent. The conversion usually required is therefore from arbitrary text to phonemes, although context information such as morpheme or word boundaries may also be retained for use in synthesis.

Even with the input text constrained to be a string of ASCII characters, text-to-phoneme conversion is a difficult multi-stage operation, and perfect systems have not yet been developed. The first stage of the conversion is usually a text normalisation procedure to interpret such things as paragraphs, punctuation, numbers, and other non-word characters (see Section 1.8.1). The resulting sequence of words and symbols may then be altered by the addition of extra pauses inferred from the words of the utterance by a phrase boundary placement system (see Section 1.9.1); this is one area where text-to-unit and text-to-prosody conversion overlap. The sequence of words and symbols must then be converted into a sequence of phonemes, as described in Section 1.8.2. Data-based approaches to phrase boundary placement and phonetic pronunciation have been investigated, and these are discussed in the appropriate sections.

1.8.1 Text Normalisation

The text normalisation stage of a TTS system converts the input text into a sequence of words and symbols to be processed by the rest of the system. The symbols represent non-word information such as punctuation, which, although usually realised as short periods of silence, retain their identities at this stage to aid in the generation of prosody. For example, the beginning of a paragraph often necessitates a higher fundamental frequency, and therefore paragraph identification is important. Often a tab indicates a new paragraph, but (Klatt 1987) reports that this is not always reliable, and that DECtalk therefore requires an explicit *new paragraph* marker in the text. Sentence identification is also important. In many writing systems this is a trivial task, because a single symbol is used exclusively for marking the ends of sentences, (Sproat and Olive 1995). However, in many other writing systems, including that used for English, the end of sentence marker (a period) can also indicate an abbreviation, an initial in a name, or a decimal point. The problem of sentence identification is therefore often complicated by the problem of abbreviation expansion, which is itself non-trivial. For example, St. can be expanded as Saint or Street, N. can be an initial in a name, or an abbreviation for North or New, etc. Often such ambiguities can be resolved by rules which look at the capitalisation of adjacent words, but this is not always possible. Phrase identification is usually accomplished both by interpreting other punctuation and by the use of phrase break placement algorithms (see Section 1.9.1). Other punctuation, such as commas, question marks, quotes, etc. are also relevant to these tasks, and so must also be interpreted. Apostrophes must be interpreted carefully, since they may be part of a word, or they may be used as quotes.

Numbers require special treatment by TTS systems. The conversion of a number to a string of words is relatively simple, but care is required to use the correct form of pronunciation. For example, times, dates, years, telephone numbers, amounts of currency, and alphanumerics, must all be correctly identified, and then correctly pronounced. However, most of these distinctions can be handled with simple rules which check immediate context, lack of commas, etc., (Allen et al. 1987). Finally, with specialist applications additional problems may arise; for example, the expansion of chemical symbols, or of electrical resistor values such as 2k7.

1.8.2 Word Pronunciation

Given the output of the text normalisation procedure, a pronunciation must be selected for each word. In many languages there is a close correlation between spelling and pronunciation, and so this task is relatively simple. In others, including English, this is not the case, and the problem is considerably more difficult. In all cases, proper names, words borrowed from other languages, and new words, cause further difficulties. Simply storing pronunciations for all the words which might be encountered by a TTS system is not possible for several reasons. Firstly, the number of possible words is infeasibly large. In fact, even if new words and foreign words are discounted the number is still infeasibly large; (O'Malley 1990) reports that the number of words that the average American high school student might encounter has been estimated at five hundred thousand, and that the 1970 United States census listed more than two million different surnames. Secondly, the pronunciation of the same set of letters can vary with context. The letters may become a different conceptual word; for example, *read* can be */r iy d/* or */r eh d/*, *bass* can be */b ae s/* or */b ey s/*. Though the former ambiguity can be resolved by a grammatical analysis of the sentence to be synthesised, the latter can only be resolved by using wider semantic information. Stress assignment may also affect the pronunciation of some words, with, for example, unstressed vowels being reduced to a schwa; this is another area in which text-to-unit and text-to-prosody conversion overlap. Thirdly, in continuous speech word pronunciations can be affected by the phonetic context resulting from adjacent words. For example *the*, which is normally pronounced */dh ax/* becomes */dh iy/* if the following word begins with a vowel. A list containing several examples of such cross-word effects for English is given in (Giachin et al. 1991).

Early attempts to convert words into phonemes were based on letter-to-phoneme rules, which attempt to assign phonemes to each letter, or group of letters, based on their context, (Klatt 1987). However, particularly with English, the performance of such systems is degraded by the large number of exceptions to the rules. In order to improve performance, exceptions dictionaries were introduced, which typically listed a few thousand common words for which the rules did not work. Klatt reported that this improved performance from about 85% of words correct for a rule-only system to about 95-97%, when tested on a random sample of words from a large dictionary. An alternative approach was used by the MITalk system, (Allen et al. 1987), in which a morphemic decomposition of the input text was performed. Morphemes are the smallest meaningful units of language, and *morphs* are their letter string representations, being either prefixes, roots, or suffixes. For example, *houseboats* can be split into the roots *house* and *boat*, and the suffix *s*. The decomposition is often more difficult than this however, and rules are required to cope with spelling changes during morph combination and with words with multiple parses. The system therefore comprised a set of rules and a lexicon listing the pronunciation of about 12,000 morphs. Words which did not follow the rules and led to incorrect parses were added to the lexicon as a whole unit. This approach had several advantages. Firstly, the lexicon of only 12,000 morphs enabled the system to cope with over 100,000 words, representing an improved storage efficiency over a simple lexicon. Secondly, words like *hothouse*, which a letter-to-phoneme rule system might consider to contain a */th/*, are pronounced correctly, and thirdly, morphemic decomposition yields part-of-speech information which is very useful for determining prosody. The accuracy of the MITalk system was never properly measured, although (Klatt 1987) estimated it to be about 95% words-correct.

The above approaches perform reasonably well with normal words, but have difficulties with proper names; (Klatt 1987) reported that the best rule-based system in 1987 still had a 20% error rate with names. This is because names can come from many different languages, and the rules governing their pronunciation are often language specific. Furthermore, exceptions dictionaries are less useful with names than they are with normal words; whereas it takes only 141 words to achieve a 50% level of coverage of running text with normal words, over 2300 names are required to achieve the same level of coverage for names, (Coker et al. 1990). Research conducted at AT&T Bell Labs. addressed the problem of names, and the wider pronunciation problem, in a radical manner, (Coker et al. 1990). Instead of using a dictionary just for exceptions, a dictionary was used whenever possible, and letter-to-phoneme rules used only as a last resort. The dictionary used included the pronunciation of 50,000 names. Many pronunciations could therefore be obtained directly from the dictionary. Many more could be obtained using methods which made use of the dictionary entries. These included looking for morphological decompositions of unknown words, and using rhyming techniques which used letter to phoneme rules at the beginning of words (which is relatively safe) and then compared spellings with dictionary entries for the rest of the word. These methods perform much better than pure letter-to-phoneme rules. Such rules were used only when all the dictionary based methods had failed to give a pronunciation, and this occurred with less than 0.1% of non-name words and only about 2.6% of names. A pronunciation error rate was not given for this system, but the authors claimed that it was at least an order of magnitude lower than those of more traditional approaches. The cost of this performance was the large amount of storage required for the dictionary.

Word Pronunciation : Data-based Methods

Some attempts have been made to construct data-based systems to perform the pronunciation stage of text-to-phoneme conversion. However, unfortunately, these systems have so far not performed as well as traditional rule-based approaches. (Lucassen and Mercer 1984) used a discrete hidden Markov model system to align letters to phonemes for about 70,000 words, producing over 500,000 context-phoneme pairs. Each context was described in terms of a number of automatically determined features incorporating information about the surrounding 8 letters and the preceding 3 phonemes. The context-phoneme pairs were used to construct a decision tree in which each node was split by the feature which maximised the mutual information with the phoneme distribution in the node. Finally, a robustly estimated phoneme distribution was determined for each node by combining the distributions in each of the nodes on the path from the root to the leaf using the method of deleted interpolation (Jelinek and Mercer 1980). An arbitrary spelling could then be converted to a phoneme sequence using a dynamic programming algorithm and the decision tree. In tests, the system correctly predicted 93.7% of phonemes, with approximately half of the errors being only vowel stress errors. Later, a neural network system called *NETtalk*, (Sejnowski and Rosenberg 1986), received widespread publicity. This system used a 3-layer neural network with a hidden layer of 120 units and approximately 25,000 weights. The input layer consisted of 7 groups of units, each of which coded a single letter, punctuation marker, or word boundary marker, from a 7 letter input window. Each input was processed by the network to give an output in terms of 23 articulatory features, stress level, and syllable boundary information, which referred to the pronunciation of the central letter in the input window. The output phoneme was selected to be the one whose vector made the smallest angle in the feature space with the vector output by the network. After back-propagation training on a 20,000 word dictionary the performance reached 90% of phonemes correctly predicted. Note that, with both these systems, the performance figures refer to the percentage of phonemes correctly predicted, and imply much lower performance figures with words.

1.9 Text to Prosody

The *prosody* of an utterance is a term used to describe its perceived pitch, stress, and rhythm. Its physical correlates are fundamental frequency, segmental duration, energy, and to some extent phonetic and acoustic variation. In human speech, the prosody of an utterance often depends not only on its words, but also on its intended meaning, its intended audience, the emotional or physical state of the speaker, and many other factors. Many of these factors are present even in read speech, because humans generally interpret and understand the text that they are reading. Thus, it is likely that TTS systems will only perform as well as humans when they too can understand the input text, using some form of artificial intelligence. Since, at the time of writing, this technology is many years away, most current TTS systems attempt only an emotionless, declarative reading of the input text. However, even this is very difficult, and no perfect solution has yet been found. The following sections attempt to present an overview of the problems involved and briefly describe both traditional approaches, and some more recent data-based approaches, to their solution.

The conversion of text into prosodic parameters is essentially composed of three subsystems. These are, the placement of prosodic phrase boundaries (see Section 1.9.1), the determination of segmental durations (see Section 1.9.2), and the specification of fundamental frequency contours (see Section 1.9.3). Although the energy contour of an utterance is prosodically important, it has been found that the energy contour implied by the fundamental frequency of an utterance (energy scales with fundamental frequency), combined with normal segmental energies, is often sufficient. In fact, including rules to explicitly increase stressed vowel intensities, for example, produces artificially strong stressed vowels, (Klatt 1987). As mentioned in Section 1.8.2, prosodic concerns can cause phonetic alterations to the pronunciation of an utterance, with, for example, unstressed vowels being produced as a schwa. Prosodic concerns may also cause more subtle acoustic variations, such as spectral alterations at different stress levels. Such alterations could be encoded in the rules of a rule-based formant synthesiser, or associated with distinct units in the case of concatenation synthesisers. However, both explicit prosodically motivated energy alterations and acoustic alterations are relatively small effects, and are not discussed further in this section.

1.9.1 Intonational Phrase Boundary Placement

From a prosodic point of view, natural speech can be considered to be composed of a series of intonational phrase groups. These phrase groups are most broadly defined as the regions within which a single intonational tune evolves, with the fundamental frequency and energy being reset at the beginning of each new phrase group. The boundaries between phrase groups often correspond to the moments when speakers draw breath, and are therefore often associated with a short pause in the speech. The boundaries are often marked by punctuation in the corresponding text, but they can also occur at clause or syntactic boundaries which are not explicitly marked. The placement of intonational phrase boundaries by TTS systems is important for two reasons. Firstly, they define the regions within which individual pitch and energy contours should be applied. Secondly, they break long utterances into phrases which could realistically have been spoken by a human in one breath; their absence often results in listeners feeling short of breath. As mentioned in Section 1.8, this is one area where the text-to-prosody and text-to-units problems overlap, since pauses introduced at intonational phrase boundaries affect the synthesis unit sequence.

The placement of phrase boundaries by TTS systems ideally requires a full syntactic parse of the sentence to be synthesised, which is very difficult to obtain automatically. Often, a missing phrase boundary just makes speech sound rushed, and is not as bad as an extra phrase boundary, which can be distracting and confusing, (Klatt 1987). The simplest solution is therefore to place phrase boundaries only where punctuation dictates. A slightly more sophisticated solution is to store a list of function words, and use these to detect the more obvious phrase boundaries not indicated by the punctuation. Klatt reported that the Prose-2000 and the Infovox SA-101 TTS systems used this approach, and that DECtalk used a similar approach with an extended word list which included verbs. The MITalk system was more ambitious, attempting to derive a syntactic parse of each input sentence. Sentence internal pauses were inserted both as dictated by punctuation, and at detectable clause boundaries. An additional algorithm was included to insert extra pauses into very long stretches of unbroken speech. This operated by considering the number of syllables on either side of a potential break and the strength of various syntactic boundaries in the region. The system performed reasonably well but still made a rather large number of errors, both failing to detect existing boundaries and inserting inappropriate boundaries, (Allen et al. 1987).

Intonational Phrase Boundary Placement : Data-based Methods

Some attempts have been made to develop data-based methods to place intonational phrase boundaries. The current AT&T TTS system uses automatically constructed decision trees to determine the locations of its boundaries, (Wang and Hirschberg 1992). Numerous experiments were conducted using Classification and Regression Trees (CARTs), (Breiman et al. 1984), to cluster a labelled database on the basis of many different features. The features investigated were speaker identity, sentence type, both time and word based distance information, part-of-speech information, syntactic information, and pitch-accent information. It was found that there was considerable redundancy amongst the features. A tree constructed using only those features which could be automatically labelled from text could correctly classify 89% of boundaries, only 1% below the best score obtained when using both manually and automatically labelled features. In this tree, part-of-speech information and word-based distance information were found to be the most useful features. More recently, (Sanders and Taylor 1995) investigated the use of a part-of-speech trigram and word based distance measures to predict phrase boundaries. The probability of a phrase boundary occurring between the second and third words of all trigram sequences present in the training database was calculated, and various methods investigated to combine this information with distance information to predict phrase boundaries. Similar results were obtained to the CART methods just described, although manually obtained part-of-speech information was used in this case. Both these investigations sought alternative scoring mechanisms to compensate for the fact that null-boundaries always heavily outnumbered true boundaries. However, whilst these scores were undoubtably useful in evaluating system performance, given that inserting a false boundary is generally perceptually less acceptable than missing a true boundary, it is not clear that these alternatives were appropriate for speech synthesis applications.

1.9.2 Segmental Duration Prediction

Traditionally, segmental durations for speech synthesis were predicted using a set of rules. These rules attempted to allow for all possible factors which could have a perceptually important effect on the segmental durations. (Klatt 1987) reported that many different rule systems were developed, all using slightly different approaches to successfully predict the same phenomena. It was therefore almost impossible to determine which rule system most accurately reflected psychological processes. Factors used in these systems included phonetic context, word frequency, syntactic category, and phrase and clause boundaries, amongst others. Some systems explicitly attempted to enforce rhythm in the synthetic speech. The rules were formulated in terms of many different speech units, including phones, syllables, and words. Indeed, Klatt reported that the size of unit best suited to model different timing phenomena was one of the unsolved problems of duration prediction, and the various data-based approaches described later in this section show that this is still the case today.

One typical rule-based system was that proposed by Klatt, (Klatt 1979), which was implemented in the MITalk system, (Allen et al. 1987). The duration of each segment (phone) was calculated using the equation

$$DUR = (INDUR - MINDUR) * \frac{PRCNT}{100} + MINDUR \tag{1.4}$$

where $INDUR$ and $MINDUR$ were the inherent and minimum durations of a segment, stored in a lookup table, and $PRCNT$ was the percentage alteration determined by applying the system's rules. Ten rules were applied, each of which adjusted $PRCNT$ by a multiplicative factor to adjust for the effects of phonetic environment, stress level, position in word, clause or phrase, and emphasis, followed by one extra rule applied after the calculation of DUR. The rules were suggested by work presented in the literature, but the details of each rule were determined by trial-and-error, matching the output against manually segmented speech read by Dennis Klatt. Testing the rules against new speech from the same speaker showed that the predicted durations differed from the natural durations by a standard deviation of only 17ms.

Segmental Duration Prediction : Data-based Methods

The major problem to be overcome by any data-based approach to duration prediction is the data-scarcity problem. For all but the least context sensitive models, the training data will, in general, contain only a very small fraction of the contexts which could occur during arbitrary speech synthesis. Systems must therefore incorporate some mechanism by which durations can be predicted for unseen contexts using only those contexts contained in the training data. Many solutions to this problem have been investigated in recent years, including statistical clustering based methods, neural network based methods, and methods involving more application specific trainable models. Some examples of significant research in this area are discussed below, in order to illustrate the problems involved and various methods of solution.

Riley, (Riley 1990), (Riley 1992), used CARTs to automatically cluster phone durations according to their context. Contextual effects included were stress level, position in word, position in phrase, and phonetic context up to a distance of 3 phones in each direction. Using individual phone labels to describe phonetic context would result in a serious data-scarcity problem, and therefore each phone was described in terms of four broad features, these being consonant manner, consonant place, "vowel manner", and "vowel place". The tree was built to minimise the variance of the error when predicting the training data, with tree-size being determined using cross-validation techniques. The training data used comprised 1500 short sentences spoken by a single speaker. The author reported that the tree predicted the training data with errors having a standard deviation of 23ms, which, although much lower than the 35ms of a previous rule-based system, did not result in noticeably better synthetic speech than the rule-based system. This was thought to be due to the presence of occasional poor predictions, caused either by there being insufficient training data for a particular context, or by the limited feature set of the model being unable to capture some effects. A similar tree-based method to duration prediction was used in the present work, details of which can be found in Sections 3.4.2, 5.3.1, and 8.3.

(Campbell 1992) investigated an alternative approach in which syllable durations were predicted using a three layer neural network, with phone durations computed within the resulting syllable framework. Neural networks represent an attractive solution to the problem of duration prediction because, hopefully, a neural network could learn something of the underlying interactions between different contextual effects, and hence perform well with unseen context combinations. The network was trained on a set of feature-duration pairs, obtained by manually segmenting a 20 minute single-speaker continuous read speech database. The durations were expressed as log durations throughout in order to increase sensitivity for short syllables, where large relative changes in duration might be small when expressed in milliseconds. The feature vector used to train the network comprised information about syllable length, in terms of number of phonemes, the nature of the syllabic peak, the position in tone-group, the type of foot[2], stress level, and word class, i.e. function word or content word. Within each syllable, phone durations were assigned using the "elasticity principle". Under this principle each phone is produced with the duration occurring at approximately the same probability density in its duration distribution, that is, phone i, with a mean duration μ_i and standard deviation σ_i, is produced with a duration $\mu_i + k\sigma_i$, where k is common across the syllable. This principle was shown to be generally applicable, except for sentence-final syllables, using phone level durational data from a different database. The performance of the system could not be expressed in terms of original database durations, since two different databases were used in its construction. However, the author reported that the durations produced did result in intelligible synthetic speech.

The current AT&T TTS system uses a method involving many "sums-of-products" models, (van Santen 1994). A manually constructed decision tree is used to group phones into categories, the members of each of which are similarly affected by the same contextual factors. For each category, a sums-of-products model is selected and estimated from the available data. Each model combines the various contextual factors, as a sum of a sequence of products of terms associated with each form of context, to compute a log-duration. The author claims that some sum-of-products model must be applicable to each category due to the ordered structure of the data. The underlying systematic interactions captured in each model should enable durations in new contexts to be accurately predicted by interpolation. Log-durations are used for similar reasons to those described above. Perceptual tests showed that the new system was superior to a previous rule-based system, when testing sentences for which the predictions of the two systems were very different.

1.9.3 Fundamental Frequency Contour Prediction

The fundamental frequency (F_0) of a human utterance is determined by a combination of many factors, from all levels of the human speech production process. At the lowest level, unintentional localised segmental effects on F_0, producing what is often termed micro-intonation, are caused by the physical dynamics of the human speech production process. For example, high vowels often raise F_0, unvoiced obstruents often raise F_0 on a following vowel, and voiced obstruents and glottal stops often lower F_0, (Pierrehumbert 1981). The stress pattern of the desired utterance also affects the F_0 contour. Syllables may be stressed due to lexical, emphatic, or contrastive stress, as defined both by the

[2] A foot is a unit of speech containing one stressed syllable followed by a number (possibly zero) of unstressed syllables.

words of an utterance and its intended meaning. A stressed syllable may result in either a raised F_0, or a lowered F_0, although it is widely reported that stressed syllables occurring after the nuclear stress[3] do not affect F_0, (Pierrehumbert 1981). The F_0 contour is also affected by the intonation pattern of a phrase. This is the long range F_0 pattern which distinguishes, for example, a statement from a question. It also includes continuation rises, which are often used to indicate that more speech is about to follow, and phrase final falls, which usually indicate that the sentence is finished. Finally, the F_0 contour is also affected by such things as gender, physical and emotional state, and attitude. In addition to all the influences already described, it has also been demonstrated that many F_0 contours implicitly include an overall declination in fundamental frequency over the course of each intonational phrase. That is, F_0 peaks occurring later in a sentence do not have to be as high as those earlier in the sentence in order to be perceived as having the same degree of prominence, (Pierrehumbert 1981). A full discussion of the various influences and interactions involved in determining the F_0 contours of natural speech is beyond the scope of this thesis; for one such discussion see (Cruttenden 1986). The remainder of this section therefore limits itself to a description of some of the more prominent practical systems which have been devised for use in speech synthesis.

The generation of F_0 contours for speech synthesis has usually been limited to providing only neutral, declarative intonation patterns for most sentences, with some additional provision made for questions. As described in Section 1.9, no attempt is made to incorporate higher level effects, such as emotion and attitude, because they would require the system to have a degree of understanding of the input text beyond that which is currently possible. The F_0 contours generated have also frequently not included lower level segmental effects. Although this absence results in smoother contours than occur in natural speech, this is not very important perceptually; ('t Hart et al. 1990) report that in experiments comparing speech synthesised using heavily stylised F_0 contours with and without micro-intonation, the two contours could be distinguished only by trained listeners performing direct pair comparisons, and only if they contained relatively long stretches free from other intonational effects.

The MITalk system, (Allen et al. 1987), used the outputs of its syntactic parsing and pronunciation modules to construct an F_0 contour. Three intonational "tunes" were used to define different global F_0 patterns for declarative statements, yes/no questions, and wh-questions. Each phrase was accompanied by a rise in F_0 on the first content word, and a fall beginning on the last content word, with a phrase final continuation rise added if the sentence was not finished. A rise-fall contour was produced on the stressed syllable of each content word. The size of this was determined by the syntactic category of the word, the number of syllables in it, and the position of the word in the sentence; peaks were calculated relative to a declination line which fell gradually over the course of each sentence. Further rules implemented a number of segmental level effects, and adjusted F_0 peaks according to their proximity to each other and their syntactic context.

(Pierrehumbert 1981) developed a linguistic description of F_0 contours which considered each phrase to be constructed from a sequence of F_0 targets which were either high

[3]The nuclear stress is the main stress of a phrase.

(H) or low (L). The targets were associated either with stressed syllables, in which case they were called pitch-accents, or with phrase boundaries and the nuclear fall, in which case they were called boundary tones. The exact F_0 values corresponding to each target were seen as fractions of the distance between a baseline and topline, both of which declined over each phrase. The only occasion on which the F_0 contour was allowed outside of this range was during the nuclear fall. Stressed syllables occurring after the nuclear syllable could not have pitch-accents. The F_0 contour was calculated as transitions between the target values, with the transition between an L target and any other being monotonic, but the transition between two H targets involving an F_0 sag. This description could be applied to many different intonation patterns. In the case of neutral declarative intonation all pitch-accents were H, and the phrase final tones either L_L or L_H. However, although the model was very useful descriptively, in synthesis applications it was difficult to determine which pitch-accents should receive what prominence. The solution described in (Pierrehumbert 1981) was to assign the nuclear pitch-accent to the last content word of each phrase, with a fraction of 1.0, and to assign pre-nuclear main word stresses alternating values of 0.4 and 0.7.

Fundamental Frequency Contour Prediction : Data-based Methods

Unlike most aspects of TTS conversion, data-based methods for synthesising F_0 contours have been investigated for many years. A number of methods have been developed which seek to generate an F_0 contour as the output of a digital filter responding to a linguistically motivated excitation signal. Perhaps the most well known of these is the Fujisaki model, developed at the University of Tokyo. In the implementation described in (Hirose and Fujisaki 1982) two critically damped second-order linear filters were used to synthesise sentence length F_0 contours. One filter was used to model the phrase level declination typically observed in neutral declarative intonation, and was excited by pulses placed at phrase boundaries. The other was used to model localised pitch accents, and was excited by a series of step functions. The parameters of the filters were determined by minimising the mean squared error (in the log domain) between contours obtained from natural speech and corresponding model contours. When both filter parameters and excitation signals were optimised for each utterance the model could match natural Japanese F_0 contours very closely. The authors reported that there was no loss of naturalness when fixed filter parameters were used, provided that the excitation signals were specified approximately correctly. However, the problem of determining these signals for arbitrary text in speech synthesis applications was not discussed, and, as with the Pierrehumbert system described above, was likely to be problematic. Nevertheless, a similar filter-based approach was used in the Klattalk system, (Klatt 1982), (Klatt 1987). In this case a single excitation signal was constructed, using rules, from a gradually declining baseline, intonational “hat patterns” associated with syntactic units, localised impulses associated with stressed syllables, and localised segmental perturbations.

A more recent approach, reported in (Ross and Ostendorf 1994), overcame some of the difficulties of specifying a detailed excitation signal. A dynamical system model was estimated to relate F_0 contours to a more abstract prosodic specification of each sentence, which was defined using labels from the TOBI labelling system, (Silverman et al. 1992). The model incorporated phrase level, syllable level, and phoneme level (and hence segmental) effects on F_0. It was trained iteratively on a prosodically labelled database of radio news broadcasts spoken by a single speaker. Listening tests demonstrated that the system could out-perform a recent version of the AT&T TTS system when manually generated prosodic labels were available to both systems for use in synthesis.

1.10 Scope and Structure of Thesis

The research described in this thesis was conducted with two principle aims. These were, to build a hidden Markov model (HMM) based speech synthesis system which could synthesise very high quality speech, and, to ensure that all the parameters used by the system were obtained through training. The motivation behind the first of these aims was to determine if the HMM techniques which have been applied so successfully in recent years to the problem of automatic speech recognition could achieve a similar level of success in the field of speech synthesis. The motivation behind the second aim was to construct a system that would be very flexible with respect to changing voices, or even languages.

These two aims were realised in a system which used HMMs in conjunction with a state-based clustering algorithm and segment selection algorithm to automatically segment an acoustic unit inventory of HMM-state sized segments from a single-speaker continuous-speech database for use in a concatenation synthesiser. During synthesis, the system could convert a string of words, with known phonetic pronunciations selected manually from a pronunciation dictionary, into intelligible, natural sounding speech. No explicit prosodic modelling was undertaken, and the system therefore produced speech in a monotone, with little noticeable durational prosody. However, prosody could easily be imposed on the synthetic speech if the system were part of a larger ASS system.

This thesis is structured as follows. Chapter 2 reviews other work relevant to the problem of automatic acoustic unit inventory construction. Chapter 3 describes HMMs and their application to speech problems, and the state-clustering algorithm used. Listening tests were conducted periodically in order to evaluate the performance of the system, and to provide information about its shortcomings. Chapter 4 reviews the various tests available, and describes in detail the testing procedure used during the course of this work. The construction of the basic synthesis system is described in Chapter 5. This basic system used an LP synthesiser, and an HMM system and clustering algorithm very similar to those used in the HTK large vocabulary speech recognition system, (Woodland et al. 1994). Numerous shortcomings were identified in the performance of this basic system, and many alterations were therefore made in order to improve its transcription, clustering, and segmentation capabilities. These improvements are described in Chapter 6. In order to further improve performance, the LP synthesiser was replaced by a TD-PSOLA synthesiser, as described in Chapter 7. Chapter 8 presents a detailed analysis of the results of the improved systems, and a discussion of the remaining problems. Finally, Chapter 9 describes some areas of possible future work, and presents the conclusions of the current work. A description of the audio examples on the accompanying compact disc is given in Appendix E.

Chapter 2

Automatic Acoustic Inventory Construction

Automatic techniques have begun to be applied to the problems of transcribing, segmenting, and selecting acoustic units for concatenative speech synthesis in the last few years, largely due to recent advances in the field of automatic speech recognition. The research follows the success of concatenation based synthesisers using manually transcribed, segmented, and selected units, and is aimed at enabling any or all of these procedures to be performed both more efficiently and more effectively. The extraction of most types of sub-word unit requires a knowledge of the phonetic transcription of the speech database being used, and automatic methods investigated to perform this transcription are discussed in Section 2.1. Automatic segmentation techniques have been investigated to segment units from both manually and automatically transcribed databases, and this research is discussed in Section 2.2. Research conducted into automatic unit selection algorithms, including context clustering algorithms, is discussed in Section 2.3. Finally, a brief description of the research described in this thesis, in terms of the issues discussed in this chapter, is given in Section 2.4.

2.1 Automatic Phonetic Transcription

The databases traditionally used to prepare segment inventories for speech synthesis were composed of isolated nonsense words, constructed specially to contain the required diphone or polyphone segments. In theory this should mean that the phone sequence is known, but in practice this is often not the case; (Boeffard et al. 1992) reported that even when presenting the words at the recording stage as phonetic strings, some phoneme elision and assimilation occurred, particularly during consonant to consonant transitions. These problems are amplified by lazy speech, and (Boeffard et al. 1993) suggest that their relatively poor performance with segmenting German diphones partly results from using real words, which are produced in a lax manner, instead of using nonsense words. The recent research into methods of automatic unit segmentation and selection has seen the increasing use of continuous speech databases. With these, in general, only the word sequence is known, and the phonetic transcription must be deduced. This is much more difficult than with isolated word databases, since words may have many possible pronunciations, and occurrences of assimilation and elision are considerably more widespread.

Phonetic transcription has traditionally been performed by trained humans. However, this is not ideal, particularly for the large databases often used by automatic segmentation or selection algorithms, because it is both manually intensive, and prone to errors and inconsistencies. The amount of time required to transcribe a new database could be reduced by using more than one human, but then consistency problems become more serious. For example, (Ljolje and Riley 1993) found that the phonetic transcriptions of 50 sentences produced by two humans resulted in 7% of the boundaries placed having one or other of the phonemes labelled differently. It was thought likely that this figure would have been much larger had automatic transcriptions not been provided to the humans as an incentive to keep to a standard transcription. The need is therefore apparent for some form of automatic phonetic transcription system, which, although not likely to be error free, is likely to be more consistent, and much quicker, than a human.

At the time of writing, unconstrained phone recognition systems do not provide a reasonable solution to the problem of phonetic transcription, and therefore alternative methods are required. Since the orthographic transcriptions of continuous speech databases are often known, a likely phone sequence can be obtained by concatenating word pronunciations obtained from a pronunciation dictionary. This was the approach used by (Brugnara et al. 1992), who obtained phone insertion, deletion, and substitution rates of 4.5%, 9%, and 6% respectively, compared to a human transcription. Another possible solution is to use the orthographic transcription and the text-to-phoneme converter of a TTS system. However, both of these solutions do not access the acoustic data in any way, and so are unlikely to give accurate results.

The most accurate automatic methods of obtaining phonetic transcriptions are probably those which use a phone recognition system to select between alternative pronunciations on the basis of the acoustic data. The alternatives can be obtained from a pronunciation dictionary in the case of real words, or suggested by known assimilation or elision rules in the case of nonsense words. This technique is often used in the training of phone based speech recognition systems to ensure the correct model order during parameter reestimation. It is the method used in the work described in this thesis; for more information see Sections 3.3.2 and 5.2.2. A similar approach for synthesis purposes was attempted by (Ljolje and Riley 1991), who used a "phone realisation tree", which had been trained on thousands of transcribed sentences, to produce many alternative pronunciations for an utterance. From these alternatives was generated a list of all possible three-phone sequences which could exist for that utterance, which was then used as the constraint for an HMM based phone recogniser. A transcription error rate of about 10%, with only 2% insertion and deletion, was reported.

2.2 Automatic Segmentation

Research has been conducted into automatically segmenting both phones, and hence diphones, and also smaller, non-traditional units, often defined precisely only by the automatic procedure used. The methods investigated to segment the former, more traditional units, are discussed in Sections 2.2.1 and 2.2.2, and those involving non-traditional units in Section 2.2.3.

2.2.1 Phone Segmentation

Phone segmentation has traditionally been of interest for synthesis uses only as a stage of diphone and polyphone segmentation, to produce units for use in diphone, and augmented diphone, synthesis systems, such as those discussed in Section 1.7.4. For these uses, phone segmentation acts only as a guide for the more accurate diphone or polyphone segmentation to follow, and so only limited accuracy is required. Recently however, with the introduction of the automatic selection algorithms discussed in Section 2.3, phone length units have begun to be used as synthesis units in themselves, and hence the need has arisen for both accurate and consistent phone segmentation.

Traditionally phone segmentation was performed by humans, but, for similar reasons to those with phonetic transcription, this approach is non-ideal. The largest problem with human segmentation is the huge amount of effort, and the correspondingly large time delay, involved in segmenting a new database. An idea of the time-scales involved can be obtained from (Taylor and Isard 1991), who report that it originally took three months to manually segment CSTR's diphone database. Another problem is the lack of consistency inherent in human segmentations. This is particularly important for phone-length unit concatenation synthesisers, where segment boundaries may occur in regions of rapidly changing speech, and hence in which the consistent placement of boundaries in all segments which may be adjacent during synthesis is essential to ensure concatenation smoothness. Hence, the need for an automatic procedure is apparent.

Most recent attempts at automatic segmentation reported in the literature are based upon the use of hidden Markov models (HMMs). These models formed the basis of the work described in this thesis, and are described in detail in Chapter 3. Experiments have been conducted to segment both single and multi speaker databases comprised of both isolated words and continuous speech, using both automatic and manually generated phonetic transcriptions. The performance of the automatic algorithms was usually quoted in terms of how close the segmentation produced was to a human segmentation of the same speech. However, these figures should be considered in the light of results obtained by (Ljolje and Riley 1993), who investigated the segmentation of a single speaker continuous speech database by several automatic systems and *two* humans. They found that 80% of corresponding boundaries placed by the two humans were within approximately 8ms of each other.

The multi-speaker American English TIMIT speech database was investigated by (Brugnara et al. 1992) and (Ljolje and Riley 1991). (Brugnara et al. 1992) used a monophone based system, and obtained 86.9% of their automatic boundaries within 20ms of the manual ones (supplied with the TIMIT database), when manually segmented speech was used to train the HMMs. This dropped slightly to 84.7% when using an automatically generated transcription during testing, using only boundaries between corresponding labels for evaluation. However performance dropped to only 75.6% when the HMMs were trained without reference to any manually obtained boundary information. (Ljolje and Riley 1991) used a more complicated context dependent system, and obtained 80% of all boundaries within 15ms of the manual boundaries both when and when not using manually obtained boundary information to train the HMMs. They also reported only a slight drop in performance, to 80% of all boundaries being within 17ms of the manual boundaries, when using an automatically derived transcription during testing.

In order to construct a concatenation based speech synthesiser, a single speaker speech database is usually required. This is actually advantageous, since speaker dependent HMMs generally perform better than speaker independent ones. (Ljolje and Riley 1993) investigated the segmentation of a single speaker continuous speech database using several automatic systems and two humans. They found that there was little difference between the results obtained using context independent or context dependent models, or between models trained with and without reference to manually produced boundaries. Their best result, of more than 80% of automatic boundaries being within 11.5ms of corresponding manually produced ones, was obtained using context independent models trained using manually produced phone boundaries. This compares well to their result quoted above, of 80% of corresponding boundaries being within approximately 8ms of each other, when comparing two human segmentations of the same speech.

The automatic segmentation of isolated word databases has also been investigated, principally as a precursor to diphone or polyphone segmentation, which is discussed in Section 2.2.2. The words, which are usually nonsense words, are specially prepared to contain the required diphones and polyphones, usually in a neutral phonetic context. Given a transcription, the knowledge of the word boundaries makes the segmentation problem slightly easier than in the continuous speech case, but not drastically so. (Taylor and Isard 1991) obtained 95% of phone boundaries within 30ms of manually placed boundaries, using a simple monophone system trained using manually produced boundary information. They also reported that vowel to semi-vowel, and vowel to nasal pairs were the most difficult boundaries to place automatically. A similar system, but trained without reference to manually produced boundary information, was used by (Boeffard et al. 1993), who obtained 89.5% of boundaries within 30ms of manual boundaries for their French system, with slightly lower scores for Spanish and German. The authors also reported that boundaries between phones belonging to the same broad phonetic class were more difficult to locate automatically than boundaries between phones belonging to different phonetic classes; a similar result was obtained by (Ljolje and Riley 1993).

2.2.2 Diphone Segmentation

The traditional approach to diphone concatenation synthesis was to manually segment diphones during system construction, usually placing boundaries in the relatively steady state regions in the middle of pre-segmented phones. Here again, an automatic procedure offers a much quicker, and more consistent, solution to the segmentation problem. In addition, an automatic procedure also enables an alternative solution to be adopted, in which many boundaries are determined for each diphone to minimise the discontinuities resulting from concatenating all possible diphone pairs. Automation is necessary for this approach because the boundaries must either be computed during synthesis, or pre-computed for all possible diphone pairs, of which there are a very large number.

In the system developed by (Taylor and Isard 1991), the whole two-phone pair associated with each diphone was stored, and the diphone boundaries placed during synthesis.

The first algorithm investigated placed the transition between two diphones at the position in the rightmost phone of the first diphone and the leftmost phone of the second diphone where the MFCC vectors were most similar, using a Euclidean distance metric. However, this algorithm sometimes resulted in very short or very long phones, and so a more complex algorithm was devised in which the sum of the distances between all aligned frames was found for each possible alignment of the two phones to be concatenated. The boundaries were then placed where the smallest distance occurred in the best alignment. The authors went on to note that the algorithms compensated to some extent for errors in the preceding phone segmentation, and also reported that informal listening tests had showed that variable diphone boundaries resulted in synthetic speech which was preferable to that produced using manually placed fixed boundaries. (Boeffard et al. 1992) investigated both fixed boundaries, placed at the moment of minimum spectral derivative in each phone comprising each diphone, and a variable boundary placement algorithm. In the latter, two sets of boundaries were computed for each diphone, for all possible left and right neighbours. For each possible pair of diphones, a matrix was constructed containing all the distances between the frames of the rightmost phone of the first diphone and the leftmost phone of the second diphone, similar to the first algorithm investigated by Taylor and Isard. However, this matrix was then smoothed before finding the minimum distance, in order to avoid problems with local minima. The authors reported that the speech produced using the variable boundaries was not perceptually different to that produced using the fixed boundaries.

2.2.3 Sub-Phone Unit Segmentation

The use of sub-phone units in speech synthesis has arisen largely because finite state speech modelling methods, such as vector quantisation (VQ) and hidden Markov models (HMMs), make use of such units. In VQ or ergodic[1] HMM systems, a large number of states are used to model all the speech available as training data, by quantising the speech into a finite number of acoustically self-similar states. With enough states, each state represents a pool of, usually, sub-phone length segments of speech, because speech generally becomes more self-similar on such time-scales. These systems are directly applicable to vocoding, or speech compression, since only the state sequence of the original speech must be transmitted or stored. However, using such systems for ASS is less straightforward, since the relationship between words or phonemes to states must be established, and this is non-trivial. Nevertheless, ergodic HMM based systems have been used for ASS with some success. The use of multiple model HMM systems, in which the relationship between states and phonemes is explicit, has also been investigated, although only very recently. Such a system also formed the basis of the work described in this thesis.

The first reported application of HMM techniques to speech synthesis appears to be that of (Farges and Clements 1986). A system was developed which used a large 64 state ergodic HMM with 1024-observation discrete output distributions, as the basis of a vocoder. The HMM was trained on a 15 minute single-speaker database. In use, the system worked by transmitting the state sequence of the observed speech, determined

[1]Ergodic HMMs are those in which any state can transit to any other state; see Section 3.3.2.

using the Viterbi algorithm. At the receiver the most likely observation sequence for the transmitted state sequence was found, by maximising a function dependent both on the output probabilities of the model, and a smoothness term. The results were found to be superior to those obtained using a 6-bit (i.e. 64 state) vector quantiser based system, but inferior to those obtained with a 10-bit (i.e. 1024 state) system. Furthermore the HMM transition matrix had a lower entropy than the equivalent matrix computed using the 6-bit VQ system, indicating that more compression was possible with the former.

(Falaschi et al. 1989) also used a 64 state ergodic HMM, but with continuous autoregressive Gaussian output distributions, (Poritz 1982), (Juang 1984), similar to the Linear Prediction based distance measure derived in Appendix D. The ergodic HMM was trained on about 8 minutes of speech, and then used to synthesise isolated words, somewhat indirectly. The model was Viterbi aligned to a single occurrence of a word to be synthesised. The state sequence obtained was used to construct a smaller left-to-right HMM, whose observation vectors were composed of both autoregressive Gaussians, and other features necessary to drive a speech synthesiser, namely energy, voicing, and pitch frequency. The new HMM was then trained on multiple occurrences of the word to be synthesised. Finally, to synthesise the word, a sequence of feature vectors was calculated from the final left-to-right HMM's mean vectors, using weight functions based on mean state durations to determine the contribution of each state to the feature vector at each point in time. The authors stated that the method produced an intelligible and natural speech quality.

The work just described was extended by (Giustiniani and Pierucci 1991), who introduced a mechanism by which speech could be synthesised from a phoneme string specification. An ergodic HMM, with either 64 or 256 states, and continuous autoregressive Gaussian output distributions was trained on a single speaker speech database, as before. In order to relate the acoustically defined states to phonemes, a set of discrete output distributions were also defined, specifying the probability of observing every phoneme in each state. The discrete distributions were trained using a frame synchronous phonetic labelling of a subset of the acoustic training data. During synthesis a text processing module, and a rule-based duration module, were used to construct a phoneme string in which each phoneme was repeated a number of times to indicate its duration. This string was Viterbi aligned to the states using the discrete distributions, to give a state sequence. The LP parameters necessary to drive a synthesiser were then obtained from the autoregressive output distributions of the states in the sequence, without, it appears, any parameter smoothing. The system also included rule based pitch and amplitude determination modules. The authors claimed that spectrograms produced from speech generated by the system demonstrated that it could correctly reproduce the main acoustic correlates of each phoneme, and that co-articulation was handled well; however, they did not comment on the intelligibility of the speech.

Later, (Sharman 1994) developed a similar system, which had the advantage that it did not require the phonetic hand labelling of training data. A vector quantiser was used to cluster the frames of the training data, comprising approximately 40 minutes of speech from a single speaker, into 320 acoustically self-similar units, termed *fenemes*. A global *n-gram HMM* was then constructed, in which each state was associated with an individual phoneme, with discrete output distributions modelling the probabilities of every feneme being generated by each state. The n-gram[2] was incorporated into the HMM structure to enable the model to be constrained to long state sequences. For each sentence in the training data, a phonetic transcription was obtained using the text-to-phoneme module of the TTS system, and a time aligned feneme transcription obtained using the vector quantiser. The HMM was then trained on the pairs of transcriptions, with the n-gram structure ensuring that the state sequence matched the phonetic transcription of each sentence. Once trained, the model was used to align the phonetic transcription of each sentence in the training data to its fenemic transcription. The phoneme-feneme alignments were then used to construct the inverse n-gram HMM, similar to that used by (Giustiniani and Pierucci 1991), in which each state was associated with a feneme, and the output distributions modelled the probabilities of every phoneme being generated by each state. The new model also included durational constraints, enabling the feneme sequence corresponding to an arbitrary phoneme sequence with arbitrary durations to be generated during synthesis. Phoneme durations and a pitch contour were determined by separate modules, and the synthetic speech generated by PSOLA concatenation of waveform segments chosen to represent each feneme. The author stated that informal listening had established that the speech produced was intelligible, and recognisably like that of the original speaker.

Very recently the use of multiple model HMM systems for speech synthesis has begun to be investigated, (Tokuda et al. 1995a), (Tokuda et al. 1995b). A set of multiple mixture continuous output distribution 3-state left-to-right monophone HMMs were trained on a database of Japanese speech recorded from a single speaker. The feature vectors used to code the speech were calculated using a mel-cepstrum analysis developed by the authors which enables speech to be re-synthesised from the cepstral coefficients. The first and second differentials of the cepstral coefficients were also included in the feature vector. During synthesis the monophone HMMs were concatenated in the order defined by the sentence to be synthesised, to create a composite HMM. An approximation to the most likely observation sequence to be generated from this model, using the most likely state sequence, was then found using an iterative algorithm derived by the authors. Durations were handled by including a duration probability term in the likelihood maximised by this algorithm. When only static coefficients were included in the feature vector, the most likely observation sequence was composed only of state means, which caused discontinuities in the synthetic speech at the moments of transition between states. The inclusion of dynamic coefficients in the feature vector, and hence the models, meant that the dynamic coefficients of the observation sequence generated in synthesis were constrained to be realistic, as defined by the parameters of the models. The result was a much smoother spectral evolution of the synthetic speech. The authors did not comment on the overall quality of the synthetic speech, other than to say it was quite smooth. However, demonstrations played at Eurospeech'95 were very encouraging.

[2] An n-gram is a grammar which specifies the probabilities of sequences of n items, in this case phonemes, occurring.

2.3 Automatic Unit Selection

Automatic selection algorithms have been investigated to select both phone-length and variable-length units, from both manually and automatically segmented databases. The work with phone-length units is discussed in Section 2.3.1, and that with variable-length units in Section 2.3.3. The selection of phone-length units has often been approached using context clustering algorithms, and these methods are discussed separately in Section 2.3.2.

2.3.1 Phone-Length Units

Concatenation synthesis was not traditionally attempted with phone-length units because of the large variation in the acoustic realisations of phonemes in different contexts. The variation meant that many realisations of each phoneme would have to be stored for use in synthesis, and that these units would have to be segmented in such a way that they could be concatenated smoothly. As described in Section 2.2.1 the latter is particularly difficult for phone-length segments, since the points of concatenation may be in regions of rapidly changing speech. Since selection and segmentation were generally performed by hand, the diphone, which suffered these problems only to a much lesser degree, was often the unit of choice. However, with the introduction of automated selection algorithms to determine which phone-length segments to use in a particular context, and automatic segmentation algorithms with their more consistent performance, phone-length unit concatenation systems have begun to be investigated. The potential advantage of such systems is that the increased context sensitivity of the unit selection procedure should lead to the selection of more appropriate units, and hence better quality synthetic speech.

The system developed by (Hauptmann 1993) used a large single-speaker speech database of 3,253 sentences, occupying 360MB of disc-space, all of which was stored for use in synthesis. The database was segmented into phones using a speech recognition system with reference to the orthographic transcription of each sentence. Each of the 115,000 phones in the database was then labelled with context information, such as stress level, phonetic context, and position within syllable, word and sentence. The segment used to represent a particular phoneme during synthesis was selected by using an experimentally determined heuristic to assign a context matching score to each realisation of the phoneme in the database, and selecting the one with the best score. The rank order of importance of the various context effects used in the heuristic was, stress $\gg$ phonetic context $\gg$ word boundary context $\gg$ utterance boundary context. The selected segments were then concatenated using the PSOLA algorithm, leaving the pitch and duration of the segments unaltered, apart from localised pitch smoothing at the concatenation boundaries. The author reported that the synthetic speech produced by the system ranged from "nearly indistinguishable from natural speech", to "barely intelligible" in some places. Modified Rhyme Tests (see Chapter 4) were conducted to evaluate the segmental intelligibility of the system, and a respectable error rate of 11.3% obtained.

A similar system was developed by (Black and Campbell 1995), which was used with both English and Japanese, and male and female, databases. In this system a dynamic programming algorithm was used during synthesis to select the sequence of phone-length segments which minimised a cost function. The cost for each segment depended both on the accuracy with which it matched the target specified, and on the continuity distortion between it and the previous segment selected. The target cost was computed as a weighted combination of phonetic context, duration, log power, and mean pitch frequency, and the continuity cost as a similar combination of phonetic context and prosodic context, together with an acoustic join cost. The concatenation points in adjacent segments were chosen to be those points in each with the smallest acoustic join cost, with the search limited to seven frames around the labelled segment boundaries. Thus the method was insensitive to small segmentation errors, and was used with both manually and automatically determined segment boundaries. The weights of the cost function were optimised by removing a sentence from the database, and then using that sentence to specify the targets during synthesis. The synthetic sentence was then compared to the original by calculating the mean Euclidean distance between their time aligned cepstral vectors. The process was repeated for several sentences for each of a large range of weight values, with weightings performing well for many sentences considered good. This method was computationally intensive, but produced better results than hand tuned weights. During synthesis, the waveforms of the selected segments were concatenated either directly, with no additional signal processing, or, more recently, using the PSOLA algorithm. Listening tests were conducted to establish the correspondence between the human perception of quality and that implied by the mean cepstral distance score. Interestingly, it was found that humans tended to place more importance on continuity than accuracy, but the reverse for the cepstral distance measure. The authors stated that they were therefore seeking an alternative to the latter.

2.3.2 Context Clustering

The systems described in Section 2.3.1 selected units by computing selection scores during synthesis, by using a context weighting scheme to find the most appropriate unit from the training database. The precise weighting given to different context factors during unit selection was established either by trial and error, or by optimising the weights by synthesising some test speech. The main drawback of the implementations of this approach discussed above is that the entire training database had to be stored for use during synthesis, although this could be reduced by, for example, synthesising some test speech and discarding infrequently used segments from the database. More fundamental is that the context weightings established were global, applying to every phone in every context. An alternative approach to segment selection, which overcomes both these problems, is that of tree-based context clustering. In this approach, the training data is clustered in a tree fashion, by splitting each node into sub-nodes on the basis of the acoustic data, using partitions suggested by the data's context labels. This produces a number of clusters each comprised of contextually and acoustically similar segments. The storage problem is therefore reduced, since only some representation of each cluster must be stored in order to be able to reproduce all the principle acoustic realisations of each phoneme. Furthermore, the tree structure means that the most important context effects are determined for each phone in each context, as the tree is built, instead of in a global fashion. The size of the tree, and hence the fineness of the modelling, can be set to match the segment inventory size required. The cluster to use for a particular context during synthesis can then be deduced either by descending the appropriate tree, or by using a context matching score to compare the required context label to the available cluster context labels. The latter is necessary if the trees are not stored for use in synthesis. Although this is effectively a globally defined context matching scheme, it represents a considerable improvement over the schemes used with non-clustered data, because the selection is only between clusters with permissible context matches.

The first attempt at using a statistical clustering technique to build a phone-based unit inventory for ASS was that reported in (Nakajima and Hamada 1988). A technique was introduced which the authors named *Context Oriented Clustering (COC)*, which clustered all the versions of each phoneme available in the training data according to their phonetic context. The clustering algorithm worked by building a binary decision tree for each phoneme. Each node of the tree was split by examining all the pairs of possible daughter nodes defined as the group of current node members with a new left (or right) context of a particular phone, and the group of current members without this left (or right) context. Since the parent node could already contain phones in a particular context, the context labels of the daughters could therefore extend beyond the immediate phonetic neighbour. The actual phonetic context used to perform the split was the one which gave the maximum "split evaluation value", defined as the difference between the inner-cluster variance of the LP parameters of the segments in the parent node, and the average of the inner-cluster variances of the daughter nodes. The clustering continued until there were no clusters with more than some number (N_{min}) of members, or until the average of all the inner-cluster variances dropped below some threshold. When clustering was completed, all the members of a particular cluster were time-warped to the average duration of that cluster, and then the centroid of the LP parameters was stored for each point in time within the average duration. Thus both segment duration information and within-segment transitions were preserved. The result was a set of clustered segments, each with phoneme and context labels, which were then concatenated during synthesis to produce an arbitrary phoneme sequence by selecting for each phoneme the stored segment whose context label most closely matched that required. The closeness was determined using a context matching score which compared only the symbolic similarity of contexts, and had no knowledge of methods of phoneme production, or acoustic similarity. The authors used about 5 minutes of manually segmented single-speaker Japanese speech as training data, and clustered this into 627 synthesis units. It was encouraging to note that many of the contexts selected were essentially consonant-vowel pairs, which are the principle syllabic units in Japanese. Several sentences were synthesised by concatenating the units with no interpolation across boundaries. The authors claimed that the speech was highly intelligible and fluent.

The COC technique was later extended to include wider context information, (Nakajima 1993). Three stress levels were distinguished, as well as word-final position, and sentence-final position. Context grouping was also used, to enable broad phonetic class contexts to be used to split clusters. This grouping is advantageous when working with English, where many of the possible phonetic contexts do not appear even in a large speech database, and it is necessary to infer cluster membership from similar contexts. A new expression for the split evaluation value was also introduced, and in the cases of unvoiced fricatives and unvoiced plosives the segment closest to the cluster centroid was used as the synthesis unit, in place of the centroid vector sequence. During synthesis, each phoneme was assigned a segment using a similar context matching score to that described above. The resulting system, called *Multi-Layered Context Oriented Clustering (ML-COC)*, was then applied to English, using a 45 minute manually segmented single-speaker speech database. Both the stress and word boundary contexts were found to be useful in defining distinct synthesis units. Interestingly they were much more important than phonetic contexts beyond the immediate phonetic context.

Recently the ML-COC technique was used as the basis of a waveform concatenation synthesiser, using the PSOLA algorithm, (Itoh et al. 1994). The clustering was carried out as above, and then the waveform segment closest to each cluster centroid selected, using a Mel-LSP[3] distance measure, to represent that cluster during synthesis. The pitch and duration of the segments were altered during synthesis using the PSOLA algorithm, with each phone duration being made equal to the average duration of that cluster. The system was applied to an English single-speaker speech database similar to that described above. Listening tests showed that the synthetic speech was much preferred compared to a similar system using parametric synthesis; however, no intelligibility tests were conducted.

A similar system to ML-COC was investigated by (Wang et al. 1993). This system built a binary decision tree for each phoneme by splitting each node using information about the broad phonetic class of the contexts of the members of that node. For example a node might be split according to the manner of articulation: plosive, nasal, fricative or approximant. All possible class combinations resulting in a binary split were tried for all context factors (eg. manner, place, etc.) across all nodes, and the best node to split selected using a cepstral distance measure. Four time-normalised 11-dimensional feature vectors were used to represent each segment for this procedure. Cross-validation experiments were conducted which showed that stopping the tree building process only by insisting on a minimum cluster occupation count led to over-fitting of the trees to the training data, with a resultant drop in performance when clustering unseen data. Examination of the cross-validation error could therefore be used to determine when to stop growing the trees. During synthesis a phoneme in any context could be mapped to a segment by descending the phoneme's tree. A non-terminal node was used if on descending the tree a node was reached where the context class required had not been seen in training, eg. if a node was split according to manner of articulation of the previous phoneme, and no segments at this node had had preceding nasals during training, then a phoneme with a left nasal context would be assigned to this node. The authors noted that the articulation manner and position of the preceding phoneme, and sometimes the stress of the current phoneme, were the most important factors for clustering. They also noted that some classes of consonants were insensitive to contextual variation. Unfortunately, the authors did not report any attempt to use the system to generate synthetic speech.

[3]Mel scaled line spectral pairs; an LP derived parametric representation of the speech spectrum with properties similar to those of formant frequencies and bandwidths, (Rabiner and Juang 1993).

2.3.3 Variable-Length Units

Research has also been conducted into the automatic identification and selection of multiple-phone length units for use in concatenation synthesis. As discussed in Section 1.7, longer units are desirable because they result in fewer boundaries, and hence fewer concatenation discontinuities, in the synthetic speech. The disadvantage is that a relatively large number of units are required to synthesise arbitrary speech.

(Sagisaka 1988) introduced a novel solution to the unit length versus storage space problem. His system used a speech database of 5240 words, and its phonetic transcription, from which variable length units were selected during synthesis. A tree structured *Synthesis Unit Entry Dictionary* was constructed to hold all the distinct phoneme sequences in the training data, and pointers to all the occurrences of each which could be used as templates in synthesis. The dictionary enabled the rapid construction of a lattice holding all the available unit combinations which could be used to produce the desired utterance. A path through the lattice was then chosen by first reducing the number of templates available for each unit, by retaining only those templates in suitable contexts, and then by applying a number of criteria designed to reduce the number and size of discontinuities resulting in the synthetic speech. These criteria included conserving CV transitions, which form the bulk of syllables in Japanese, conserving transitions between vocalic sounds, giving preference to longer units, and trying to use units with the maximum amount of overlap. The resulting system therefore used long units, often of CVCV type structure followed by voiceless consonants, when they were available, and shorter units otherwise. The system had a separate prosody control module, and used an LP synthesiser, but the author did not comment on the quality of the synthetic speech produced. Analysis of a Japanese dictionary and Japanese sentences showed that the most frequently used 20% of three and four phoneme long sequences, accounted for about 80% of the occurrences of such sequences in the texts. This indicates that a high level of coverage with longer units can be achieved, in Japanese at least, by storing only a small fraction of the number of theoretically possible sequences.

More usually, researchers have sought to supplement diphone concatenation systems with specific longer units. As described in Section 1.7.4, these units have usually been selected manually in order to protect highly co-articulated phones. However, recently, research has begun to be conducted with the aim of automatically identifying the most frequently used longer units, in order to reduce the number of concatenation boundaries during synthesis as much as possible with the minimum increase in storage requirements. (Klavans and Tzoukermann 1994) examined occurrence frequencies of triphones in both two machine readable dictionaries and two text corpora for French. The phonetic transcription of the corpora was obtained using grapheme-to-phoneme conversion software. The results showed that whilst the sets of the 1000 most frequent triphones in the two dictionaries overlapped by about 35%, and those derived from the two corpora by about 29%, the overlaps between the sets from the dictionaries and those from the corpora were only about half as much. This result demonstrates that dictionary based methods alone are insufficient for determining which three-phone segments to store for ASS systems. The authors did not report any attempt to use the selected units to synthesise speech.

2.4 This Thesis

The work described in this thesis used a decision-tree state-clustered HMM based approach to automate the construction of an acoustic inventory. The HMM system performed automatic phonetic transcription, automatic clustering at the HMM-state level, and automatic segmentation of the resulting clustered-states. Synthesis was achieved by concatenating representations of these clustered-states. In order to facilitate the use of a waveform concatenation synthesiser, an automatic algorithm was also developed to select individual waveform segments to represent each clustered state. For further details see Chapters 3, 5, 6, 7, and, 8. The use of variable length units was not investigated during the course of this work. However, a discussion of how the state-based approach could form the underlying basis of a variable unit length system is discussed in Chapter 9.

Chapter 3

Hidden Markov Models

Hidden Markov model (HMM) based approaches to automatic speech recognition have had a great deal of success in recent years, and at the time of writing, most leading speech recognition systems are based on HMMs to some extent. As discussed in Section 1.10, a motivation for the research described in this thesis was to determine if HMMs could bring similar benefits to the field of speech synthesis. This chapter describes the underlying theory of HMMs in Section 3.2, their application to speech problems in Section 3.3, and the decision-tree state-clustering algorithm used in this research in Section 3.4. Firstly, however, mention is made of the HMM toolkit used throughout this research.

3.1 The HTK System

HTK is a hidden Markov model toolkit which was developed over several years at Cambridge University's Engineering Department and is now sold through Entropic Research Laboratory Inc., and Entropic Cambridge Research Laboratory Ltd. It consists of a suite of tools enabling the definition, initialisation, re-estimation and editing of sets of continuous mixture Gaussian HMMs. It also includes tools to perform speech coding, alignments, model clustering, speech recognition, and waveform viewing. It supports a generalised tying mechanism which enables the sharing of parameters between HMMs. This sharing enables a balance to be struck between system complexity and data availability.

The HTK system was used extensively in this research. The code used was essentially that from version 1.5, (Young et al. 1993), although some of the tools were modified during the course of this work and additional code from an unreleased version of HTK used to perform the tree clustering described in Section 3.4.2.[1] The description of HMMs presented in the remainder of this chapter refers to HTK style HMMs, although mention is made where this differs from other styles described in the literature.

3.2 HMM Theory

This section introduces the basic structure of HMMs in Section 3.2.1, discusses HMM training in Section 3.2.2, and describes the algorithm usually used to make use of a trained HMM in Section 3.2.3.

[1]The tree clustering code is included in HTK version 2.0; see (Young et al. 1996) for details.

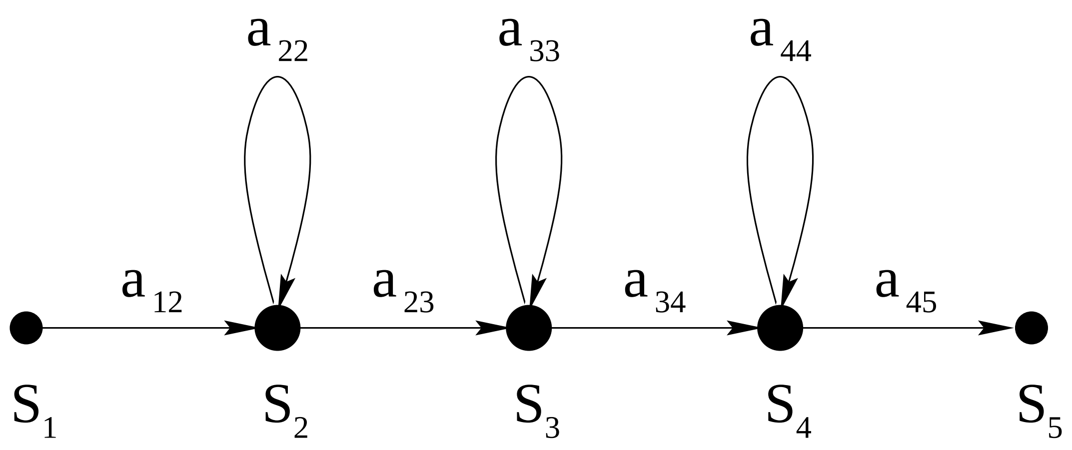


Figure 3.1: A typical HTK style left-to-right hidden Markov model. S_1 and S_5 are entry and exit states. Output distributions are not shown.

3.2.1 HMM Structure

An HMM is a statistical model for discrete-time observation sequences. A model λ is composed of N states, a transition matrix $A = \{a_{ij}\}$, and a set of output probability distributions $B = \{b_j(\cdot)\}$.

In the HTK implementation states 2 to $N-1$ are associated with an output probability distribution $b_j(\mathbf{o}_t)$, specifying the probability density of observation vector $\mathbf{o}_t$ being generated given that the model is in state j. The distributions used in HTK version 1.5 are continuous mixture Gaussians. However, (almost) all of the work described in this thesis was conducted using single Gaussian distributions, and the discussion in this chapter will therefore be limited to these. In this case,

$$b_j(\mathbf{o}_t) = \frac{1}{\sqrt{(2\pi)^n|\mathbf{\Sigma}_j|}} e^{-\frac{1}{2}(\mathbf{o}_t-\boldsymbol{\mu}_j)'\mathbf{\Sigma}_j^{-1}(\mathbf{o}_t-\boldsymbol{\mu}_j)}, \tag{3.1}$$

where $\boldsymbol{\mu}_j$ is the state mean vector, $\mathbf{\Sigma}_j$ is the state covariance matrix, and n is the dimensionality of the data. States 1 and N are defined to be entry and exit states. They are not associated with output probability distributions, and are therefore often termed non-emitting states. Note that in other styles of HMMs described in the literature all states are associated with output distributions and the entry and exit states do not exist. Furthermore, other types of output distribution are also possible, including discrete distributions.

The transition matrix A specifies the probability a_{ij} of the model being in state j at time $t+1$ given that it is in state i at time t. In other styles of HMMs described in the literature an additional distribution $\boldsymbol{\pi} = \{\pi_j\}$ is used to specify initial state occupancy probabilities, (Rabiner 1989). In the HTK implementation the use of entry and exit states means that this information is contained in the transition matrix, and therefore an additional distribution is not required. A typical left-to-right structure HTK style HMM is shown in Figure 3.1.

In operation, an HMM is used to model a discrete observation sequence $\mathbf{O} = \mathbf{o}_1, \ldots, \mathbf{o}_T$

as follows. The model starts in state 1 at time 1^-, just before the observation of the first vector $\mathbf{o}_1$. At time 1 the model moves from state 1 to any state allowed by the transition matrix, and the vector $\mathbf{o}_1$ is generated. This process is repeated at each time step until the entire observation sequence has been generated, and the model ends in state N at time T^+. Note that any observation vector may be generated while the model is in any state, though with a different probability density. The observation sequence can therefore be explained by many possible state sequences; hence the *hidden* in the name Hidden Markov Model. The total likelihood of generating the observation sequence $\mathbf{O}$ given the model λ, can therefore be calculated as a sum over all possible state sequences X,

$$L(\mathbf{O}|\lambda) = \sum_X a_{1x(1)} b_{x(1)}(\mathbf{o}_1) \left[\prod_{t=2}^{T} a_{x(t-1)x(t)} b_{x(t)}(\mathbf{o}_t) \right] a_{x(T)N} \tag{3.2}$$

where $x(t)$ is the state that the model is in at time t, and X represents the set of all possible state sequences $\{1, x(1), x(2), \ldots, x(t), \ldots, x(T), N\}$. Alternatively, it can be approximated by only considering the most likely state sequence,

$$L^*(\mathbf{O}|\lambda) = \max_X \left\{ \sum_X a_{1x(1)} b_{x(1)}(\mathbf{o}_1) \left[\prod_{t=2}^{T} a_{x(t-1)x(t)} b_{x(t)}(\mathbf{o}_t) \right] a_{x(T)N} \right\}. \tag{3.3}$$

Given that any observation vector can be generated from any state, the fact that HMMs have any use at all is because the HMM parameters can be adjusted such that particular states are associated, via their output distributions, with particular features in the observation vectors. Efficient training algorithms exist which enable this adjustment to be performed automatically, and these are described in the next section.

3.2.2 HMM Training

In HTK (and normally in general) HMMs are trained to maximise the likelihood of generating the training data $\mathbf{D}$ given the model λ, $L(\mathbf{D}|\lambda)$. The maximum likelihood approach is taken because an efficient training algorithm, the *Baum-Welch (BW) algorithm*, exists to perform the maximisation. The algorithm was introduced by (Baum et al. 1970), and extended to the case of vector observations and mixture distributions by (Liporace 1982) and (Juang 1985) respectively. Given a model λ, the BW algorithm estimates a new model $\hat{\lambda}$, for which $L(\mathbf{D}|\hat{\lambda}) \geq L(\mathbf{D}|\lambda)$, with the equality occurring when the likelihood has reached a (possibly local) maximum. HMM training therefore involves first initialising the HMM with a reasonable estimate of the model parameters, and then refining these using the BW algorithm. Methods of initialising HMMs in speech applications are discussed in Section 3.3.2

The BW algorithm calculates the parameters of the distributions of the new model $\hat{\lambda}$ as a weighted average of the parameters of the training data. The weights used are the *a posteriori* probabilities of each training vector being observed when the model is in each state, calculated using the old model λ. In general $\mathbf{D}$ will comprise R observation sequences $\mathbf{O}^r$, $1 \leq r \leq R$ each comprising T_r vectors $\mathbf{o}_t^r$, $1 \leq t \leq T_r$. For the single mixture Gaussians defined in equation 3.1, the BW formulae for the new state mean vectors and covariance matrices are then,

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_j^r(t)\mathbf{o}_t^r}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_j^r(t)}, \tag{3.4}$$

and[2],

$$\hat{\boldsymbol{\Sigma}}_j = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_j^r(t)(\mathbf{o}_t^r - \boldsymbol{\mu}_j)(\mathbf{o}_t^r - \boldsymbol{\mu}_j)'}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_j^r(t)}. \tag{3.5}$$

In these equations $\gamma_j^r(t)$ is *a posteriori* probability of the HMM being in state j when generating the tth vector in the rth observation sequence. The new transition probabilities are calculated using similar formulae given below in equations 3.20-3.22. The formulae given extend readily to the case of mixture Gaussians, by considering the different mixture components as states in parallel, with different transition probabilities due to the mixture weights; see (Young et al. 1993) for details.

The number of possible state sequences through the HMM for each training sequence is equal to the number of states to the power of the number of vectors in the sequence. Therefore, to calculate $\gamma_j^r(t)$ directly, by calculating the likelihoods of all possible state sequences through the HMM, would require some multiple of this number of calculations, which is impossible to compute for any realistic amount of training data. Fortunately an efficient recursive solution to this problem exists, which is known as the *forward-backward* algorithm. It works by calculating each $\gamma_j^r(t)$ as a product of two variables, both of which can be evaluated using recursion formulae, one forward in time, and one backward in time.

For each observation sequence $\mathbf{O}$, the forward variable, $\alpha_j(t)$, is defined as the joint likelihood of generating the sequence $\mathbf{o}_1, \ldots, \mathbf{o}_t$ and being in state j at time t,

$$\alpha_j(t) = L(\mathbf{o}_1, \ldots, \mathbf{o}_t, x(t) = j|\lambda). \tag{3.6}$$

The backward variable, $\beta_j(t)$, is defined as the likelihood of generating the sequence $\mathbf{o}_{t+1}, \ldots, \mathbf{o}_T$, given that the model is in state j at time t,

$$\beta_j(t) = L(\mathbf{o}_{t+1}, \ldots, \mathbf{o}_T|x(t) = j, \lambda). \tag{3.7}$$

Note that, although not shown, the forward variable also implicitly includes the condition that the model start in state 1 before the first observation, and the backward variable the condition that the model ends in state N after the last observation. The asymmetry in the definitions exists so that the product of two corresponding variables gives the joint likelihood of starting in state 1, generating the whole observation sequence $\mathbf{O}$, ending in state N, and being in state j at time t, given the model,

$$\alpha_j(t)\beta_j(t) = L(\mathbf{O}, x(t) = j|\lambda). \tag{3.8}$$

This enables $\gamma_j(t)$ to be found, since,

[2]Strictly speaking, $\hat{\boldsymbol{\Sigma}}_j$ should be calculated in terms of the new means $\hat{\boldsymbol{\mu}}_j$, and not the old means $\boldsymbol{\mu}_j$. However, HTK version 1.5 uses equation 3.5 as given; see (Young et al. 1993) for details.

$$\begin{aligned}
\gamma_j(t) &= L(x(t)=j|\mathbf{O},\lambda) && (3.9)\\
&= \frac{L(\mathbf{O},x(t)=j|\lambda)}{L(\mathbf{O}|\lambda)} && (3.10)\\
&= \frac{\alpha_j(t)\beta_j(t)}{L(\mathbf{O}|\lambda)}. && (3.11)
\end{aligned}$$

The only problems remaining are to find the values of $\alpha_j(t)$, $\beta_j(t)$, and $L(\mathbf{O}|\lambda)$. The first two of these quantities can be found by recursions through time, and the third as a by-product of either of these recursions.

By considering the definition of $\alpha_j(t)$ in equation 3.6, it can be seen that $\alpha_j(t)$ can be calculated as a sum of the likelihoods of being in each possible previous state at time $t-1$, weighted by the transition probability of moving to state j, and generating $\mathbf{o}_t$ while in state j. Mathematically,

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1)a_{ij}\right] b_j(\mathbf{o}_t), \qquad 1 < t \leq T,\ \ 1 < j < N, \tag{3.12}$$

with initial condition,

$$\alpha_j(1) = a_{1j}b_j(\mathbf{o}_1) \qquad 1 < j < N. \tag{3.13}$$

A similar recursion exists for the backward variable, namely

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij}b_j(\mathbf{o}_{t+1})\beta_j(t+1) \qquad 1 \leq t < T. \tag{3.14}$$

Strictly, the recursion should terminate at $T-2$, with $\beta_i(T-1)$ defined to be

$$\beta_i(T-1) = \sum_{j=2}^{N-1} a_{ij}b_j(\mathbf{o}_T)a_{jN}, \tag{3.15}$$

since $\beta_i(T)$ is ill defined in terms of equation 3.7. However, $\beta_i(T)$ is required to compute equations 3.4, 3.5, 3.20 and 3.22, and therefore it is useful to let the recursion terminate at $T-1$, and define

$$\beta_i(T) = a_{iN} \qquad 1 < i < N, \tag{3.16}$$

to maintain consistency.

Finally, note that,

$$L(\mathbf{O}|\lambda) = \sum_{i=2}^{N-1} L(\mathbf{O}, x(t)=i|\lambda), \tag{3.17}$$

which, substituting equation 3.8, and setting $t = T$ becomes,

$$L(\mathbf{O}|\lambda) = \sum_{i=2}^{N-1} \alpha_i(T)\beta_i(T) \tag{3.18}$$

$$= \sum_{i=2}^{N-1} \alpha_i(T)a_{iN}. \tag{3.19}$$

Thus $L(\mathbf{O}|\lambda)$ can be computed using only the forward variable (or, by setting $t = 1$, only the backward variable).

Having defined the forward and backward variables, the re-estimation formulae for the transition probabilities can now be given. They compute the new transition probabilities as a weighted average of the probability of a transition occurring between two states given that the model is in the first state. Again, the weights used are the *a posteriori* probabilities of each training vector being generated when the model is in each state. Mathematically,

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r-1} \gamma_i^r(t)a_{ij}b_j(\mathbf{o}_{t+1}^r)\beta_j^r(t+1)/\beta_i^r(t)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_i^r(t)} \qquad 1 < i, j < N, \tag{3.20}$$

$$\hat{a}_{1j} = \frac{1}{R}\sum_{r=1}^{R}\gamma_j^r(1) \qquad 1 < j < N, \qquad \text{and,} \tag{3.21}$$

$$\hat{a}_{iN} = \frac{\sum_{r=1}^{R}\gamma_i^r(T)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_i^r(t)} \qquad 1 < i < N. \tag{3.22}$$

3.2.3 Viterbi Alignment

In many applications, it is useful to be able to align a single state sequence to an observation sequence. The forward-backward algorithm cannot be used, since this determines a probabilistic state alignment, in which each observation vector could have been generated, with different probabilities, from many states. However, a closely related algorithm, called the *Viterbi* algorithm (Viterbi 1967), can be used to compute the maximum likelihood state sequence X^*, as defined in equation 3.3.

The Viterbi algorithm makes use of two variables $\phi_j(t)$ and $\psi_j(t)$. $\phi_j(t)$ is the likelihood of the most likely state sequence ending in state j having generated vectors $\mathbf{o}_1, \ldots, \mathbf{o}_t$,

$$\phi_j(t) = \max_{X(t^-)} L(\mathbf{o}_1, \ldots, \mathbf{o}_t, x(t) = j, |\lambda). \tag{3.23}$$

where $X(t^-)$ represents the set of all possible state sequences before time t. $\phi_j(t)$ can be computed using a recursion similar to that used for the forward variable, in which a maximisation is performed instead of a sum,

$$\phi_j(t) = \max_{1<i<N}\left[\phi_i(t-1)a_{ij}\right]b_j(\mathbf{o}_t) \qquad 1 < t \leq T, \;\; 1 < j < N. \tag{3.24}$$

The initial condition for this recursion is

$$\phi_j(1) = a_{1j}\mathbf{o}_1 \qquad 1 < j < N. \tag{3.25}$$

From the definition of $\phi_j(t)$ in equation 3.23 it can be seen that

$$L^*(\mathbf{O}|\lambda) = \max_{1<i<N} [\phi_i(T)a_{iN}] \,. \tag{3.26}$$

The other variable, $\psi_j(t)$, is required to keep track of which previous state maximised equation 3.24 for each state j at each time t, thus

$$\psi_j(t) = \operatorname*{argmax}_{1<i<N} \quad [\phi_i(t-1)a_{ij}] \,. \tag{3.27}$$

After the whole observation sequence has been generated, the maximum likelihood state sequence X^* can then be recovered as,

$$x^*(T) = \operatorname*{argmax}_{1<i<N} \quad [\phi_i(T)a_{iN}] \tag{3.28}$$

$$x^*(t) = \psi_{(x^*(t+1))}(t+1) \qquad 1 \leq t < T, \tag{3.29}$$

with of course $x^*(1^-) = 1$ and $x^*(T^+) = N$.

3.3 HMMs and Speech

This section first discusses the applicability of HMMs to modelling speech in Section 3.3.1, before describing how they have been applied to speech problems in Section 3.3.2.

3.3.1 HMM Assumptions

HMMs are useful for modelling the speech signal because their doubly stochastic structure allows them to model both the temporal and spectral variations inherent in the realisation of speech sounds. Their use does however imply a number of assumptions about the structure of the speech signal. These assumptions are,

- That the speech signal can be accurately represented as a sequence of observation vectors in time.

  The observation vectors usually used are the spectral parameterisations of consecutive, possibly overlapping, frames of speech of the order of 10ms long. As discussed in Section 1.4.3, speech can usually be considered to be stationary on such time-scales, and therefore this assumption is approximately true.

- That the speech signal can be modelled using a finite number of mixture Gaussian probability distributions.

  Although the acoustic realisation of phonemes varies considerably, realisations in similar contexts often are spectrally similar, especially when considering the speech of a single speaker. Speech can therefore be considered as a concatenation of segments selected from a finite number of segment distributions; a view exploited by many speech synthesis systems. However, representing each of these segment distributions by a mixture Gaussian of mean spectral parameters is a further approximation, which may be particularly poor if the segments all contain large highly correlated formant transitions, for example. In an attempt to improve this situation, dynamic features, calculated over several frames, are often included in the parameterisation of each frame, and hence modelled by the state Gaussians. This assumption is thus fair, but not good, since the parameter distributions used cannot accurately model the segments associated with each state.

- That the observation vectors are independent.

  In an HMM, given the current state, the likelihood of generating an observation vector depends only on the output distribution associated with the state; previous and subsequent observation vectors do not affect the likelihood. This assumption is manifestly untrue; human speech has a high degree of continuity, due to the physical inertia of the articulators in the vocal tract. This situation can also be improved to some extent by including dynamic features in the speech parameterisation, since they hold information about the local signal context.

- That the probability of transition from the current (assumed) distribution to the next (assumed) distribution does not depend on any previous or subsequent (assumed) distributions.

  That is, in an HMM, the transition probability from state i to state j depends only on states i and j. This assumption is also untrue, and means that HMMs cannot take account of long range state sequence probabilities.

Thus some of these assumptions are approximately true while others are completely false, but nevertheless HMMs do perform very well when applied to speech problems.

3.3.2 HMM Applications

HMMs have been used principally in speech recognition, but also in speech coding, and, more recently, in speech synthesis.

As discussed in Section 2.2.3, most HMM-based coding and synthesis systems have used a single global *ergodic* HMM. An ergodic HMM is one in which all states can be reached from every other state in a finite number of transitions. They are typically large, with 64 or more states, and are used to model all the training data. These models are often initialised by using a vector quantisation algorithm to cluster the training data into the required number of states. Once trained, they are used to determine the maximum likelihood state sequence for either the training data or for new data. In coding applications it is not necessary to identify individual states with linguistic labels. However, in synthesis applications it is, and this can be difficult to achieve.

HMM-based recognition systems, and very recently some synthesis systems (including the system described in this thesis), have used multiple HMMs. In these systems many small HMMs, usually with a left-to-right structure, are used to model small units of speech, such as phones or words. Each model thus has a direct correspondence to a linguistic label, thus avoiding the identification problem which occurs when using an ergodic HMM.

In continuous speech modelling, individual (usually phone-sized) HMMs are concatenated to form composite HMMs, which can model whole utterances. This is the reason for the entry and exit states in the HTK implementation — they are used to glue the individual HMMs together. The composite models so formed have non-emitting states in model-internal positions, and this complicates slightly many of the formulae in Section 3.2.2; see (Young et al. 1993) for details.

Multiple model systems are often initialised using manually segmented speech examples corresponding to the linguistic labels associated with the individual models. A typical procedure is to split all the speech examples associated with a particular model into N equal sized segments, and use the frames of corresponding segments to estimate the appropriate distribution in an N state model. The model may then be improved by Viterbi aligning the training examples, and using the frames aligned to each state to re-estimate the distributions; a process which can be repeated several times. Alternatively, with continuous speech modelling, it is possible to initialise every distribution in every model to the global mean and variance of the training data. Training using composite models defined by the orthographic transcription of the training database in conjunction with a pronunciation dictionary usually results in reasonable model parameters after only a few iterations.

The use of composite models also requires changes to the Viterbi algorithm described in Section 3.2.3. In HTK the Viterbi algorithm is implemented using the *Token Passing Model*, (Young et al. 1989), in which moveable tokens are used to determine the maximum likelihood individual model sequence (and if required the state sequence too) through a network of possible model sequences. During training the network is constructed from the orthographic transcription of the training database, in conjunction with a pronunciation dictionary. It enables an accurate phonetic transcription of the database to be determined, in which alternative word pronunciations are chosen appropriately, and inter-word silences inserted. In recognition the network is constructed from a pronunciation dictionary and a word grammar; a detailed discussion of which is beyond the scope of this thesis. Both the forward-backward calculation and the Viterbi calculation are pruned using beam search techniques, a process which can significantly reduce both memory requirements and execution times with negligible effects on performance. For further details of all the above see (Young et al. 1993).

3.4 Context Dependent HMMs

As described in the last section, when using HMMs to model continuous speech a set of phone-sized HMMs is usually used. In simple systems only one model per linguistic phoneme may be used. However, phonemes differ considerably in their acoustic realisation in different contexts, and therefore using only one model per phoneme can result in very blurred models, even for speaker dependent systems.

The use of Gaussian mixture distributions solves this problem to some extent, since each mixture component can model a different acoustic realisation. This is useful in recognition, since the mapping required is from these different possible realisations to a single linguistic label (although a context label might be more useful). However, in synthesis the reverse mapping is required, and since individual mixture components are not associated with specific linguistic labels, there is no way of determining which mixture component to use in which context. Alternative mixture component means (or segments associated with them through a maximum likelihood mixture alignment) could be selected to ensure smoothness during synthesis. Or, as described in Section 2.2.3, mixture distributions of static and dynamic parameters could be used to generate the maximum likelihood vector sequence corresponding to an HMM state sequence, (Tokuda et al. 1995b). However, these methods were not pursued in the current research.

As described in (Schwartz et al. 1984) and (Lee 1990), another solution to the acoustic variation problem which has been investigated is the use of larger sub-word units, such as diphones, demisyllables, or syllables. The arguments involved are similar to those described in Section 1.7 regarding the choice of units to use in concatenation synthesis. The advantage of larger models is that the states in their interiors are less affected by a model's contextual environment, and hence their output distributions are more precise. The disadvantage is the large number of models required for arbitrary speech, and the resulting difficulty in obtaining sufficient training data. In synthesis the additional benefit of greater concatenation smoothness associated with many longer units is also important. However, in recognition this is less important, and the longer units were essentially being used to capture the effect of phonetic context. These considerations led (Schwartz et al. 1984) to return to phone-sized HMMs, but to use several context dependent models per phoneme; a solution which had been investigated earlier by (Bahl et al. 1980). This solution now forms the basis of most high performance speech recognition systems, where it is used in addition to mixture Gaussians.

The use of several context dependent models per phoneme is a more useful solution to the acoustic variation problem for synthesis applications than using mixture Gaussians. This is because each context dependent model is associated with a distinct linguistic label. It was the solution used in the current work, and is therefore the subject of the remainder of this chapter. The data scarcity problem which arises when building context dependent models is discussed in Section 3.4.1, and the method used in the current work is described in detail in Section 3.4.2.

3.4.1 The Data Scarcity Problem

Many types of context can be used to construct context dependent HMMs. Possibilities include various levels of phonetic context, word or phrase boundary information, stress information, other prosodic information, etc. The number of possible models rises very rapidly with the amount of context information included. For example, in English, with perhaps 45 phones, extending to just a right phonetic context (biphones) could involve 2,025 models, and using both left and right phonetic contexts (triphones) could involve 91,125 models. Many of these contexts may not occur even in large training databases, and those which do occur may not occur in sufficient quantity to robustly estimate Gaussian distributions. Both of these problems must be overcome if arbitrary speech is to be synthesised or recognised.

Methods which solve *both* of the above problems are

- Backing off

  Context dependent models are estimated only where sufficient training data exists. When unavailable context dependent models are required, reduced context or monophone models, for which sufficient training data was available, are used instead. This is acceptable in recognition applications, since it results only in a less accurate likelihood calculation for the backed-off phone. However, in synthesis applications the use of inappropriate synthesis parameters or segments for the backed-off phone could cause serious localised problems in the synthetic speech.

- Top Down Clustering

  Tied models (or states) can be created by clustering context dependent models (or states) in a top-down fashion, using decision trees and context questions. The trees generated can then be used to determine the appropriate model (or state) to use in any context, including those not seen in training. Furthermore, clustering stopping criteria can be used to ensure that each tied model (or state) can be robustly estimated. This was the method used in the current work, and it is described in more detail in the next section.

Several techniques exist which enable better use to be made of the training data than the simple scheme in the first method described above. These are

- Parameter smoothing

  Techniques such as deleted-interpolation, (Jelinek and Mercer 1980), enable the parameters of poorly trained context dependent models to be smoothed with those of robustly trained reduced context or monophone models.

- Bottom-up clustering

  Acoustically similar context dependent models can be clustered and tied to form a single model, which then can be more robustly estimated, (Paul and Martin 1988), (Lee 1990). Alternatively, the clustering and tying may take place at the state level, (Hwang and Huang 1992), (Young and Woodland 1993).

However, these methods do not provide models for contexts not seen in training, for which backing off is still required.

3.4.2 Decision Tree State Clustering

Decision tree clustering of phonetic contexts for speech recognition was developed at a number of sites in the late 1980s and early 1990s. Sagayama, who had previously worked on the Context Oriented Clustering technique for speech synthesis (see Section 2.3.2), clustered manually segmented phone segments for use in a template-based speech recogniser, (Sagayama 1989). Clustering at the phone level was also developed by (Bahl et al. 1989), (Lee et al. 1990) and (Bahl et al. 1991) for HMMs with discrete output distributions, and (Odell 1992) and (Downey and Russell 1992) for HMMs with continuous distributions. (Hwang et al. 1993) extended the method to the state level for models with discrete output distributions.

The decision tree clustering method used in this research operates at the state level with continuous output distribution HMMs, (Young et al. 1994), (Odell et al. 1994), (Odell

1995). Although the method is extendible to many forms of context, and indeed was so extended during the course of this work, the following description involves only phones and their immediate phonetic contexts, including contexts across word boundaries, which are known as *cross-word triphones*.

Initially, a set of monophone models are created and trained on the available data. These models are then cloned to produce a triphone model for every distinct triphone in the training data. The transition matrix is not cloned, but remains tied across all the triphones of each base phone. The triphone models are then re-estimated, using a variance floor to prevent variances from dropping to zero in the models of those contexts which occur only a few times in the training data. In the final re-estimation, state occupation counts ($\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_j^r(t)$) are saved for use in the clustering procedure.

For each set of triphones derived from the same base phone, corresponding states are clustered using automatically constructed decision trees. Given the use of maximum likelihood model training, the trees would ideally be constructed to maximise the likelihood of generating the training data $\mathbf{D}$ given the set of tied distributions $\mathbf{S}$ in the leaf nodes of the tree, $L(\mathbf{D}|\mathbf{S})$. Unfortunately, calculating this figure exactly (using a forward pass of the BW algorithm) for every possible tree structure would be extremely computationally expensive. Instead, a sub-optimal greedy algorithm is used to maximise the increase in an approximation to $L(\mathbf{D}|\mathbf{S})$, namely $\mathcal{L}$ (defined below), at each node in the tree.

Initially, all the states to be clustered are tied to form a single Gaussian in the root node of the tree, and the value of $\mathcal{L}$ calculated. Each of a large list of clustering questions, which use phonetic knowledge to group different contexts according to their phonetic similarity, is then used to suggest a binary splitting of this node into two daughter nodes. The change in $\mathcal{L}$ which would result from each of these splits is calculated, and the question which causes the largest increase in $\mathcal{L}$ selected to perform the split. The splitting process is then repeated at the new nodes, terminating when either of two stopping criteria are met. These criteria are thresholds which specify the minimum number of frames of speech that must be assigned to each node, and the minimum increase in $\mathcal{L}$ which must be achieved for a node to be split. The former threshold ensures that sufficient data is associated with each leaf node to properly estimate a Gaussian (or mixture Gaussian) distribution. The latter ensures that nodes are not split for negligible gain.

The likelihood $\mathcal{L}$ is a computationally efficient approximation to $L(\mathbf{D}|\mathbf{S})$. Its use requires the assumption that the assignment of observation vectors to states is not altered during clustering, and that the transition probabilities can be ignored. (Odell 1995) claims that in practice any changes in the assignment of observation vectors to states are not significant. Given this, the transition probabilities can be assumed to be constant during clustering, and hence ignored because the clustering algorithm is only concerned with *changes* in likelihood. The final assumption required is that the log-likelihood of observation vector $\mathbf{o}_t^r$ being generated from $\mathbf{S}$ can be calculated as an average of the log-likelihoods of it being generated from the states $s \in \mathbf{S}$ weighted by the state occupancy probabilities, $\gamma_s^r(t)$. This is true if the values of $\gamma_s^r(t)$ are either 0 or 1, but is an approximation when using probabilistic state assignments. Given these assumptions

$$\mathcal{L} = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{s \in \mathbf{S}} ln[L(\mathbf{o}_t^r | \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)] \gamma_s^r(t), \tag{3.30}$$

where $\gamma_s^r(t)$ is the *a posteriori* probability of the tth observation vector in the rth observation sequence being observed while the model is in clustered state s. Under the above assumptions this can be calculated as

$$\gamma_s^r(t) = \sum_{j \in s} \gamma_j^r(t), \tag{3.31}$$

where j are the states of the unclustered triphone models tied to form s, and $\gamma_j^r(t)$ is as defined in Section 3.2.2. For single Gaussian distributions in each clustered state s, it can be shown that

$$\mathcal{L} = \sum_{s \in \mathbf{S}} -\frac{1}{2}(n + ln[(2\pi)^n |\boldsymbol{\Sigma}_s|]) \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_s^r(t), \tag{3.32}$$

which can be calculated without accessing the training data, by using the unclustered state Gaussian parameters and occupation counts saved from the preceding re-estimation to calculate $\boldsymbol{\Sigma}_s$ and $\sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_s^r(t)$.

After the stopping criteria are reached, similar leaf nodes from different parents are merged if the resulting reduction in $\mathcal{L}$ is less than the minimum increase in $\mathcal{L}$ used as the clustering stopping criteria. The tied Gaussians in the final leaf nodes of the tree become the Gaussians of the new clustered states. Finally, the state-clustered models are re-estimated, and the number of Gaussian mixtures in each state increased if so desired.

The clustering procedure is illustrated in Figure 3.2, which shows the construction of a tree for the middle state of the /aa/ base phone. The HTK notation X−Y+Z is used to indicate the base phone Y with a left phonetic context of X and a right context of Z. The question *Q: R_Nasal* means *Is the right phonetic context a nasal?*, and is defined to be the set of triphones $\{* - * + n, * - * + m, * - * + ng\}$, where $*$ indicates any base phone.

As mentioned in Section 3.4.1, the decision trees can be used to determine which clustered states to use to construct models for phones in any context, including those not seen in training. This is successful because many of the questions used refer to linguistically motivated broad class contexts, (Odell 1995). The trees are therefore saved after construction for use in synthesis or recognition.

The decision-tree state-clustering procedure has been shown to give superior recognition performance to a similar model-based clustering procedure, (Young et al. 1994), (Odell et al. 1994). It has also been shown to give comparable recognition performance to bottom-up, data-driven, clustering methods, when both are used to build similar word-internal triphone systems in which unseen triphones are not required. The system was used (with various levels of context information) in the HTK large vocabulary speech recognition system, which achieved some of the lowest error rates in the November 1993, 1994, and 1995 ARPA evaluations, (Woodland et al. 1994), (Woodland et al. 1995), (Woodland et al. 1996).

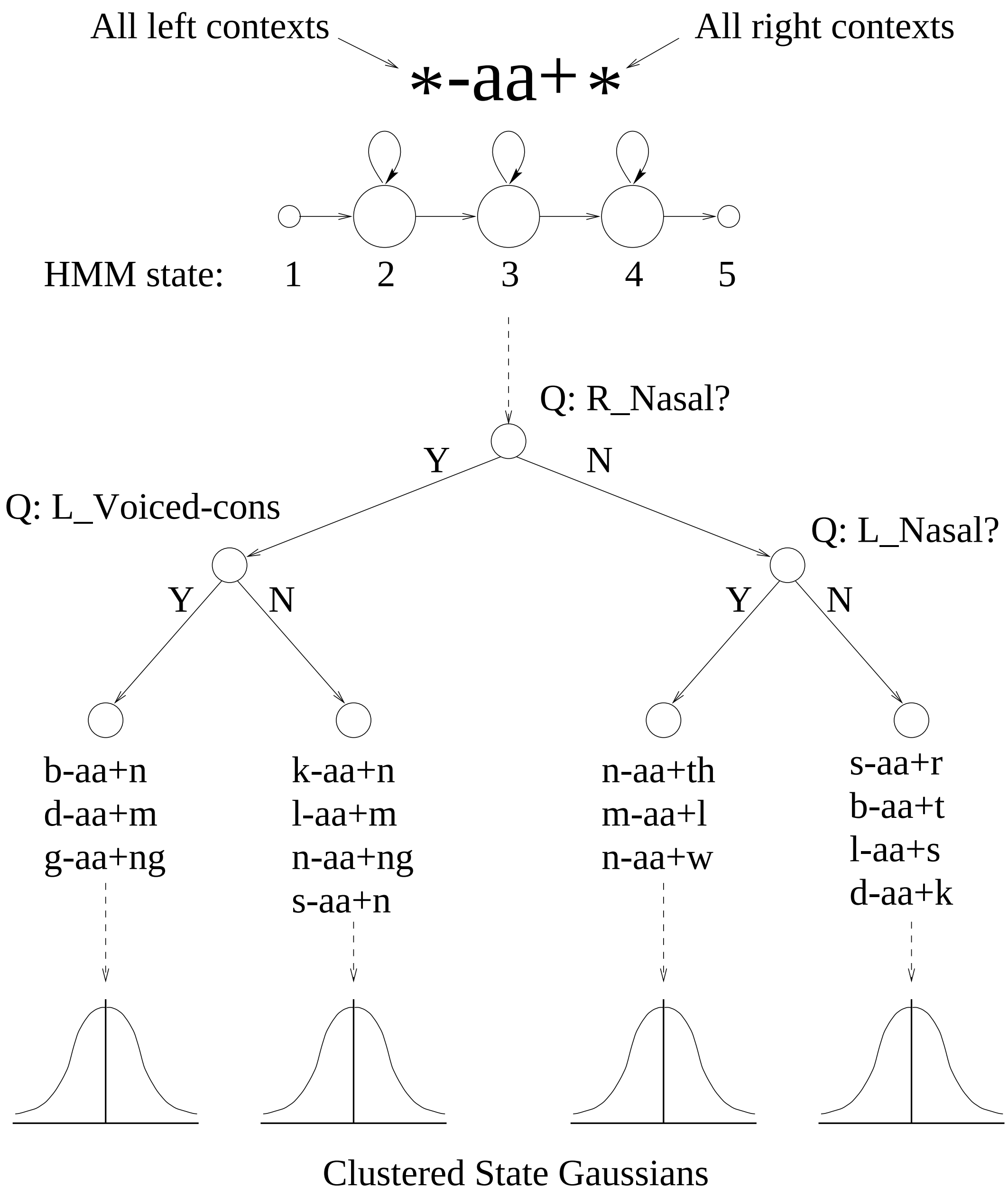


Figure 3.2: An illustration of the decision-tree state-clustering procedure. The middle states of all HMMs modelling the base phone /aa/ are clustered using the questions at the tree nodes into the groups listed under each leaf node.

Chapter 4

Performance Testing

Measuring the quality of synthetic speech provides a mechanism by which different synthesis systems can be compared, both with each other and with natural speech. It can also provide information about the deficiencies of the synthetic speech produced by a system, which can help direct future research effort to those areas of the system most in need of improvement. Given these benefits, it is perhaps surprising how many papers on speech synthesis contain only subjective remarks, such as "produced intelligible speech", and report no quantitative results. This may in part be due to the absence of a set of universally agreed tests. Indeed, as Klatt suggests, (Klatt 1987), many of the tests which are in use are not particularly well designed, and are used only because of the lack of any better alternatives.

With analysis-synthesis schemes, it is possible to calculate objective distortion measures between the original speech and the re-synthesised equivalent, (Deller et al. 1993). However, with ASS the equivalent original speech does not usually exist, and so performance must be analysed subjectively, by performing listening tests with groups of humans. To be useful, subjective measures of speech quality must be obtained in such a way as to give reliable and repeatable results. Whilst subjective testing might at first seem less appealing than an objective measure, in the end it is the human assessment of any synthetic speech which matters; (Deller et al. 1993) state that the performance criterion for an objective measure of speech quality is its correlation with subjective measures.

The quality of synthetic speech includes both its intelligibility and its naturalness. Intelligibility is a measure of how well listeners can recognise the words encoded in the speech. Naturalness is a rather nebulous concept which is not easily defined but includes (amongst others) issues of pitch, duration, and smoothness. This chapter first discusses the many different types of listening tests which have been used to test different aspects of synthetic speech performance, before describing in detail the test procedure selected for use in the current work.

4.1 Types of Listening Tests

This section begins by discussing intelligibility testing, and then goes on to other issues such as comprehension, cognitive load, naturalness, and the measuring of individual ASS sub-system performance.

4.1.1 Intelligibility Tests

The intelligibility of synthetic speech generated from an ASS system is dependent on many factors. These include the quality of the phonetic pronunciations generated by the system, the segmental quality of the synthetic speech, the speaking rate, and, in fact, the prosody of the synthetic speech (Silverman et al. 1993). Most listening tests test intelligibility to some extent, since all speech generated by an ASS system will include the system's intelligibility deficiencies. However, this section describes those tests which have been developed to test segmental intelligibility in particular.

Isolated-Word Tests

Isolated-word listening tests are attractive for several reasons. The lack of context information makes the tests very sensitive to segmental errors in the synthetic speech, and makes it relatively easy to determine the precise cause of errors, enabling the tests to be used diagnostically to improve the synthesis system. The tests are quick and easy to perform, even with untrained listeners. The results are directly relevant to the recognition of digits, letters, or unfamiliar names, which might be spoken in isolation in real-world applications. Finally, the rank ordering of synthesis systems implied by isolated-word test scores has been shown to correlate well with the results of performing many other types of intelligibility test, (Logan et al. 1989).

The most commonly used isolated word tests are the Diagnostic Rhyme Test (DRT), (Voiers 1977), and the Modified Rhyme Test (MRT), (House et al. 1965). In both cases the listener hears a sequence of isolated words, and, for each word, must select the word heard from a number of rhyming alternatives given on an answer sheet. Both tests focus upon consonants, because consonants have generally been found to be more difficult to synthesise than vowels, (Klatt 1987). The answer sheet for the MRT lists six responses for each word, each of which differs in an initial or final consonant. The MRT has also been run in an open-response mode, in order to gather more information about the confusions occurring. The answer sheet for the DRT lists only two options for each word, which differ by only one distinctive feature in an initial consonant. The DRT is so named because the results, whilst providing less information about possible confusions than those from an MRT, do enable the test to be used as a very precise diagnostic tool, (Voiers 1968). In both cases the words used are monosyllabic, of CVC or CV structure, and contain only singleton consonants, not consonant clusters. The direct implications of the results of these tests for the real world cases mentioned above is therefore somewhat limited.

Sentence-Level Tests

Two popular sentence-level tests are those involving the Harvard sentences, (Egan 1948), and the Haskins sentences, (Nye and Gaitenby 1974).[1] The Harvard sentences are a collection of meaningful sentences, whose phonetic balance is similar to that of English as a whole. Test results that analyse the intelligibility of each content word thus reflect

[1] The Harvard sentences are listed in (IEEE 1969), and examples of both Harvard and Haskins sentences listed in (Allen et al. 1987).

the likely intelligibility of the system in normal use when synthesising continuous speech. However, for good synthesis systems this does mean that the test is not very sensitive to segmental errors, since the context present enables very high scores to be obtained. Furthermore, the results are probably also influenced to some extent by the sentence prosody, and so no longer reflect solely the segmental quality of the speech. The Haskins test is more useful, since it uses meaningless sentences of the form "The (adverb) (noun) (verb) the (noun)", and therefore has a higher error rate for a given system and is a more sensitive test. Interestingly, the rank order of systems determined using these methods correlates well with the results of isolated word tests, (Klatt 1987).

4.1.2 Comprehension Tests

It has been suggested that the deficiencies of synthetic speech in intelligibility and naturalness mean that listeners have to devote more concentration to decoding synthetic speech than natural speech, and that this extra cognitive load results in less overall comprehension of the speech. Numerous experiments have been conducted in order to test this theory, and to establish the performance of synthetic speech as an information delivery tool in general.

(Allen et al. 1987) and (Klatt 1987) report various stages of a listening comprehension experiment conducted with two TTS systems, natural speech, and a visual presentation of the same information. After one presentation of the information the subjects were asked to answer multiple-choice questions relating to the passages they had heard or read. The results obtained with the TTS systems were approximately the same as those obtained with natural speech, all of which were slightly worse than those obtained with the readers. Thus, the results seem to indicate that the synthetic speech was as comprehensible as the natural speech. However, although this result is encouraging, it may simply mean that the test was not difficult enough to be sensitive to any difference in comprehensibility. One interesting result, which has been observed elsewhere, (Klatt 1987), is that the subjects performed significantly better on the second half of the listening tests with synthetic speech than on the first half, despite having had no feedback on their earlier results.

(Klatt 1987) also reports the results of experiments which required listeners to respond immediately to the content of synthetic speech or natural speech. In these experiments it was indeed shown that the human processing of synthetic utterances is slower, and the responses less accurate, than for natural speech. Klatt also reports that it has been demonstrated that the capacity of short term memory for earlier items in a list can also be reduced with synthetic speech.

4.1.3 Naturalness Tests

In addition to testing intelligibility and comprehension, it is also possible to compare systems for their overall perceived quality, which is largely a measure of their naturalness. Tests can be conducted by presenting pairs of sentences to listeners, each synthesised from a different system, and asking them to indicate a preference for one or the other, (Klatt 1987). Alternatively, listeners can be asked to describe their impression of the quality of some synthetic speech in terms of a number of categories, for example, labels ranging from "unsatisfactory" to "excellent". For each system, the *Mean Opinion Score* (MOS) can be calculated over all listeners to enable comparison between systems. Including reference signals in the tests can help normalise the MOSs, to enable some comparison to be made between results obtained with different listeners or at different times, (Deller et al. 1993). The pairwise comparison and categorisation methods described here in fact formed the basis of the three methods recommended by the IEEE for making speech quality measurements, (IEEE 1969).

4.1.4 Individual ASS Sub-System Performance

Some of the listening tests described above, such as those using isolated-words and perhaps those using Haskins sentences, can provide useful diagnostic information about the segmental quality of the synthetic speech produced by a system. However, none of them is likely to provide such information about the phonetic pronunciation, duration prediction, and pitch track generation sub-systems of an ASS system. The problem is that it is impossible to test these aspects of system performance in isolation from the effects of the other components of the system; something which is (approximately) possible when testing segmental quality, through the use of isolated monosyllabic words.

An example of the inseparability of the different aspects of ASS system performance was demonstrated by the experiment reported in (Silverman et al. 1993). This experiment examined the performance of various TTS systems when synthesising names and addresses, and might therefore appear to be principally a test of the phonetic pronunciation and segmental quality aspects of the TTS systems involved. However, it was found that performance was also very strongly affected by the prosody of the synthetic speech, to the extent that a mid-performance system was transformed to be the best-performing system by the application of a domain-specific prosodic pre-processor.

An isolated analysis of the performance of the different ASS sub-systems can be obtained by numerically comparing the parameters they produce with parameters extracted from natural speech. This approach has the disadvantage that it is not clear how to score such comparisons in a perceptually relevant way, but nevertheless such comparisons are useful.

4.2 Listening Tests Used

The principle aim of the work described in this thesis was to produce an automatic HMM-based segment inventory construction system. A test was required which reflected this aim, and therefore one of the segmental intelligibility tests described in Section 4.1.1 was appropriate. In fact the Modified Rhyme Test was selected for use for the following reasons. Firstly, (Logan et al. 1989) listed MRT results for 10 synthesis systems, including both leading research systems and commercial systems, and gave a detailed description of the experimental method followed to obtain them. Secondly, MRTs provide detailed information about the nature and location of segmental errors, unclouded by context effects. Thirdly, the tests can be easily and quickly performed with untrained listeners, and finally, the tests can be used again and again with the same group of listeners (this

System	Error Rate (%)
Natural Speech	0.53
DECtalk 1.8, Paul	3.25
DECtalk 1.8, Betty	5.72
Prose 3.0	5.72
MITalk-79	7.00
Amiga	12.25
Infovox SA 101	12.50
TSI-Proto 1	17.75
Smoothtalker	27.22
Votrax Type'n'Talk	27.44
Echo	35.56

Table 4.1: The Modified Rhyme Test error rates obtained by (Logan et al. 1989).

final point is discussed further below).

In order that the results obtained should be as comparable as possible with those listed in (Logan et al. 1989), shown in Table 4.1, the MRTs were conducted using a very similar experimental setup. Six speech files were prepared, each of which consisted of a brief introduction to the task, followed by the test words separated by 4 second intervals of silence. The sequences of test words used were exactly as given in the wordlists in (House et al. 1965). Each speech file was prepared digitally, and played directly from memory through the D-to-A converter of a Silicon Graphics Iris R4400 Indigo computer. High quality closed-back headphones were used, and the tests conducted in the CUED speech group's quiet rooms. Two answer sheets were provided, each of which contained 25 groups of 6 words (see Appendix A). The subjects were asked to put a line through whichever of the 6 words on the answer sheet they thought they had heard for each test word played. They were asked to provide only one answer in each case, and to guess if they were not sure.

The major differences between the experimental setup and that used in (Logan et al. 1989), were that Logan et al. recorded their speech onto tape, played it back at exactly 80dB sound pressure level (SPL), mixed the speech with broadband white noise presented at 55dB SPL during playback, used different audio equipment, and used native American English speakers. However, none of these technical differences were thought likely to be very significant. Native British English speakers were used for the tests reported in this thesis, but this was thought appropriate since the systems tested all had British accents, whereas the systems tested by Logan et al. are thought to all have had American accents.

With the exception of the final evaluation of the system (see Section 8.5.4), all of the MRT results reported in this thesis were obtained using groups of just six listeners. The results obtained were useful both diagnostically and comparatively. However, the small number of listeners used meant that statistical significance tests between the MRT scores of different systems were generally not useful. Systems which were clearly of different performance levels on the basis of wider informal listening were often judged as not significantly different from each other, at the 5% level, on the basis of their MRT scores. Significance tests are therefore not used in this thesis.

The sets of listeners used in the initial tests reported in Table 5.2 were inexperienced. In all subsequent tests the six listeners used were experienced, having previously performed at least one test. It was assumed that the tests could be repeated with the same group of listeners because no feedback was given to them about their individual performance. Furthermore, each listener was played a different set of words from the others on each test, thus removing the possibility of the listeners conferring between tests. However, as was discussed in Section 4.1.2, some degree of listener adaptation to synthetic speech can occur despite an absence of feedback. Such adaptation was observed during the course of this work (see Section 8.5.4) and therefore the results using experienced subjects should be treated with some caution. In order to establish a more reliable figure for the final system to enable it to be compared with the results in Table 4.1, a large scale test with inexperienced listeners was conducted with this system (see Section 8.5.4).

Chapter 5

The Basic Synthesis System

This chapter describes the building of a basic speech synthesis system based on a set of state-clustered triphone HMMs trained on a single-speaker speech database. The system used the clustered states of the HMMs as its synthesis units, and generated synthetic speech using linear prediction synthesis. The system was essentially the same as that reported in (Donovan and Woodland 1995a).

5.1 Training Speech

Several speech databases were used during the course of this work, each comprising approximately one hour of speech recorded from a single speaker. The speech was in the form of several hundred sentences, or groups of short sentences, read from prompts. These prompts were also used as word-level transcriptions during system training. Although small numbers of reading errors were likely to be present in all the databases used, the word-level transcriptions were not corrected to reflect these. A detailed description of the databases used, and of the recording procedure, is given in Appendix B. The M1 database was used in the construction of the basic system described in this chapter.

A single-speaker database was used so that the LP parameters representing each state during synthesis (see Section 5.3.4) were well defined, free from the inter-speaker variation which would result from using multi-speaker databases. One hour of speech was thought likely to be the minimum requirement for building a state-clustered HMM system. The reading material used was the Hitch Hiker's Guide to the Galaxy, (Adams 1979), which is fairly conversational in style. It was not thought necessary, or in fact wise, to use a phonetically balanced database, because it was probably beneficial to train the system on the same style of speech that it was likely to synthesise. This automatically ensured that contexts required in synthesis were likely to have appeared in the training data. Furthermore, it also provided a distribution of data which ensured that contexts required most frequently during synthesis were likely to have occurred most frequently in training. These contexts were therefore those in which the models were most finely clustered, and hence the modelling, and resulting synthesis, most accurate.

The speech was coded using frames 25ms long, with a shift between frames of 10ms, hereafter referred to as a *25/10 coding*. The speech in each frame was pre-emphasised, Hamming windowed, and coded into 12 mel frequency cepstral coefficients, to which cepstral mean subtraction was applied. The normalised log-energy of each frame was also calculated. Finally the first and second differentials of these parameters were appended, to create a 39 dimensional feature vector.

5.2 HMM Construction

5.2.1 Dictionary

The dictionary used during this research was the British English Example Pronunciations (BEEP) Dictionary, which was developed at CUED while the work described in this thesis was in progress. The dictionary was based largely on the computer usable version of the Oxford Advanced Learner's Dictionary. At the time of writing, BEEP contained approximately 160,000 words, including the inflected forms of many words, and alternative pronunciations for many words. It also included lexical stress information for many words, though the coverage was somewhat patchy. Various versions of the dictionary were used during the course of this work. The phone sets of these different versions varied slightly; the phone set of the most recent version to be used, BEEP-0.6, is given in Appendix C.

5.2.2 Monophone Training

Initially, estimates were made of the mean and diagonal covariance matrix of the parameter vectors of the whole database, using the parameter vectors of one typical sentence. A set of diagonal covariance matrix monophone HMMs was then created using these global estimates to initialise the output distributions. This set comprised a three-state left-to-right model to represent each phone in the dictionary, a similar model for long silences, and a one-state model for short inter-word silences. A phonetic transcription was generated using BEEP and the word-level transcriptions, selecting between multiple pronunciations at random. The HMMs were then trained using embedded re-estimation. They were then used in a Viterbi alignment, to select between multiple pronunciations, and introduce inter-word silences where appropriate. Using the revised transcriptions, the HMMs were then re-trained again, and the whole process iterated until the log-likelihood of the data being generated by the models had reached a plateau.

5.2.3 Triphone Training

The final monophone models were used to obtain a final monophone transcription, and then this transcription used to obtain a list of all the triphones (i.e. phones in a particular immediate phonetic context) present in the training data. For each of the triphones in the list, a model was created by cloning the appropriate monophone model. The only exceptions were the silence models, which remained monophone models. The new models were then re-trained.

Corresponding states of triphones derived from the same base phone were clustered using the HTK decision tree clustering algorithm described in Section 3.4.2. The question set used was converted from a set developed for use in the HTK large vocabulary speech recognition system, (Odell 1995), to refer to the BEEP phone set. The clustering stopping criteria required a minimum of 25 frames per state, and a minimum increase of 50 in log-likelihood for a node to be split. A total of 99,405 triphones were logically possible, of which only 8,898 occurred in the training data. The 26,694 states of these models were clustered down to only 6,412 clustered states. A single Gaussian was used as the output distribution for each clustered state. A single Gaussian was considered to be adequate since the models were only used for alignment, not recognition, and the speech from only a single speaker was being modelled. Furthermore, the use of Gaussian mixtures would require more training data per state, and it was desirable to keep the amount of training data required to a minimum. Finally, the state-clustered models were re-trained, and used to obtain a time alignment of the clustered state sequence corresponding to the final phonetic transcription of the database.

5.3 Synthesis Parameters

The state alignment was used in conjunction with the original speech database to obtain synthesis parameters for each clustered state.

5.3.1 Duration Parameters

All the occurrences of a particular clustered state in the state alignment were pooled, and the average duration and standard deviation of the duration for that state were found. During synthesis each state was synthesised for a duration given by:

$$Synthesis\ duration = average\ duration + scaling\ factor * std.\ dev.\ of\ duration \tag{5.1}$$

With the basic system described in this chapter, the scaling factor was usually set to 0.5. This slowing of the speech was necessary because the durations extracted from the database were those of fluent natural speech, and synthetic speech produced using them was often too fast to understand due to its poor quality.

Equation 5.3.1 ensured that when the speaking rate was altered during synthesis, those states which were seen to vary most in duration in the database were varied the most, and those states which were seen to have fairly constant durations in the database were varied the least. This approach would be justified if all the duration variation seen for a particular state in the database was due to local variations in speaking rate, since the deviation figure would then simply reflect this variation. In practice, the deviation figure undoubtably also reflected the different durations associated with the different contexts clustered into each state, in which case equation 5.3.1 was perhaps less appropriate. However, despite this caveat, the method seemed to work well. An analysis of the variability of the durations of clustered states is presented in Section 8.3. The actual duration stretch factors corresponding to different scaling factors are discussed in Section 8.5.2.

5.3.2 Energy

All the speech labelled as belonging to a particular clustered state was pooled to calculate the average *s*hort *t*erm *e*nergy *p*er *s*ample (*s.t.e.p.s.*) for that state. During synthesis each state was scaled to have the appropriate s.t.e.p.s. for that state.

5.3.3 Voicing

It was assumed that each clustered state was composed of speech of only one voicing type. Although this assumption was poor for phones of mixed voicing, such as voiced fricatives, it was often quite reasonable (in a temporal sense) for other phones, because the HMM system tended to segment states this way. The type was determined very simply by thresholding the average zero crossing rate of the speech labelled as belonging to each clustered state. In synthesis a pulse train excitation (with zero mean) was used for voiced states, and a Gaussian noise source for unvoiced states.

5.3.4 LP Coefficients

The LP coefficients for each state were calculated by pooling all the speech labelled as belonging to that state into a single autocorrelation vector, and calculating the LP coefficients from this vector using the autocorrelation method. This can be shown to be the way to calculate LP coefficients from multiple segments in order to minimise the total prediction error over all the segments (see Appendix D). Since the speech was sampled at 16kHz, an LP order of 20 was used, (Markel and Gray 1976). An assumption was thus made that each state could be represented by a single set of LP coefficients. This resulted in the spectral quantisation of the synthetic speech into state sized chunks in time, and can clearly be seen in the spectrograms in Figure 5.2.

5.4 Synthesis

During synthesis the words of the utterance to be synthesised were first converted to a phone string by dictionary lookup. Where multiple pronunciations existed one was chosen manually. The only text processing included was to interpret question marks, exclamation marks, full stops, and commas as short durations of silence. The phone string was then converted to a sequence of triphones, and this to a sequence of clustered states using the decision trees. As described in Chapter 3 the decision trees enabled clustered states to be assigned to all possible triphones, whether or not they were seen in the training data.

The clustered state sequence was synthesised using the lattice filter shown in Figure 5.1. The lattice structure enables the speech signal $X(z)$ to be synthesised from the excitation signal $E_p^+(z)$, using the LP parameters in their reflection coefficient form, $k_1 \ldots k_p$. This is useful because, as mentioned in Section 1.4.5, ensuring that $k_n < 1 \;\; \forall n$ guarantees a stable filter. The use of a lattice filter was probably not necessary in the current work, since the LP coefficients were calculated using the autocorrelation approach, which itself ensures filter stability, and were stored to high accuracy as floating point numbers. However, its use did enable instability concerns to be dismissed completely, and also simplified the coefficient smoothing experiments described in Section 5.6.1. For a derivation of the equations implemented in the filter see (Parsons 1986).

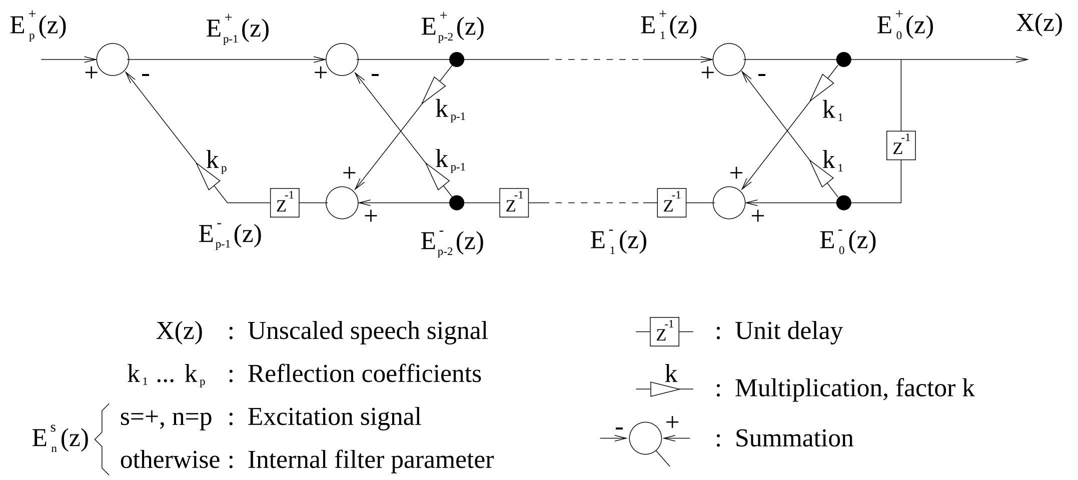


Figure 5.1: The lattice filter used in synthesis.

5.5 Variations on the Basic System

The basic system outlined so far had several parameters which were set at somewhat arbitrary values, and so experiments were conducted to investigate the effects of varying these parameters. The parameters investigated were the clustering stopping criteria, the frame rate, and the number of states in each model.

Alternatives were also sought to the pooling method of estimating the LP coefficients. Whilst this method did produce the set of LP coefficients which most accurately represented the pool as a whole, the coefficients were often blurred by the pooling of dissimilar speech. The result was broadened formant bandwidths, resulting in over-damped and buzzy synthetic speech. A number of ways of selecting a single piece of speech to base the LP coefficients of each state upon were therefore investigated.

5.5.1 Parameter Variation

The three state left-to-right models of the original system combined with the 10ms frame shift effectively enforced a minimum duration of 30ms for each phone. In an effort to determine whether more time resolution was necessary, five state models were created, and in order to keep the minimum duration at 30ms the frame shift was reduced to 6ms. Subsequently three state systems were also ran with a 6ms frame shift. Various stopping criteria were tried with all combinations of parameters.

The results of the various parameter combinations are shown in Table 5.1. The first three columns of the table show the control parameters of the various systems. The next two columns show the resulting number of clustered states per model position (i.e. the total number of states divided by the number of states in each monophone model), and the average amount of speech clustered into each state. Note that the minimum amount of speech clustered into each state is given by multiplying the minimum number of frames per state by the frame shift. The last column shows an informal ranking of the performance of each parameter set. The two systems ranked lowest were substantially worse than all the other systems. This was almost certainly due to an insufficient amount of data in

frame shift (ms)	states per model	Clustering Parameters min. frame occupancy	min. Δ L	Average speech /state (ms)	Clustered states /model position	P rank
10	3	25	50	561	2139	10
6	3	25	50	404	2966	4
6	3	42	0	573	2092	3
6	3	42	84	591	2030	1
6	3	42	168	883	1358	5
6	3	64	128	886	1354	2
6	3	128	256	1879	638	6
6	5	1	1	92	7843	12
6	5	5	10	143	5042	11
6	5	25	25	348	2066	8
6	5	25	50	350	2058	9
6	5	42	84	534	1348	7

Table 5.1: Columns 1-3 give the control parameters used in each experiment, columns 4 & 5 give the resulting system statistics, and column 6 gives the rank order of the performance of the systems using the pooled LPC method.

each clustered state to properly estimate a Gaussian distribution. The audible differences between the other configurations were small, and the rank order assigned to them was not very precise. Highly ranked configurations generally had less artifacts than those with lower ranks.

The principle conclusion of these experiments was that clustering too finely led to poor performance, because too little speech was clustered into each state to robustly estimate the parameters of a Gaussian. A minimum of 25 frames per state appeared to be enough; a minimum of 5 definitely was not. It was also clear that 3 state models performed better than 5 state models, and that a 6ms frame shift was better than a 10ms frame shift. Finally, there was some suggestion that not clustering finely enough was also detrimental to performance.

5.5.2 LP Coefficient Estimation

Four methods of estimating the LP coefficients to represent each state were tried. The first of these was described above, and involved pooling all the speech associated with each state into a single autocorrelation vector. This method will henceforth be referred to as the P-method. The F-method considered every frame of speech within the regions labelled as belonging to each clustered state as possible candidates for selection. The frame with the highest log-likelihood of being associated with the clustered state Gaussian was then selected, and then a 25ms Hamming window centred on the centre of that frame used to calculate the LP coefficients. The C-method was similar to the F-method, except that only the frames at the centre of each occurrence of each clustered state were considered as candidates in an attempt to prevent frames close to the boundary regions, which might not have been representative of that state, from being selected. Finally, the I-method did not use the state Gaussian, but instead compared the speech in a 25ms frame at the centre of each state occurrence to the P-method LP vector using a distance measure similar to the Itakura-Saito distance, described in Appendix D. Note that, at this stage, no attempt was made to use multiple LP vectors to produce each clustered state, and in the three alternative schemes considered the single vector used was estimated from a short segment of speech to reduce the effect of pooling on the formant bandwidths.

The results of the F and C methods were very similar. Whilst the speech was less buzzy than the P-method, it contained numerous artifacts, which made the speech sound poorer overall. The I-method contained less artifacts than the F and C methods, but it was still not clear that it was actually an improvement. Analysis of the artifacts seemed to show that they were either caused by the selection of frames which did not well represent the speech of that particular clustered state, or that a reasonable frame was selected, but that speech synthesised using the LP parameters estimated from that frame did not sound like that frame. The latter could occur particularly with states representing transient signals such as bursts. Here the P-method averaged together many poor sets of coefficients to calculate a very neutral, but audibly acceptable sound, whereas the other methods selected one particular frame, and hence calculated more specific, but poor, LP parameters. The superiority of the I-method can perhaps be explained by considering that the frame selected was the one most likely to have been generated by the LP parameters of the P-method, and hence it's LP parameters were likely to be similar to those of the P-method, which were generated using pooled data. In contrast the F and C-methods selected the frame on the basis of an MFCC coding, and then merely assumed that the LP parameters of the selected frame were both reasonable and representative of the pool, which was not necessarily so. Furthermore, both the parameters used for the MFCC coding, and for the LP parameter estimation, were based on a single frame, which was not always very reliable. From this discussion it may appear that the best solution would have been to use an LP coding to code the original database and build the HMMs, and then select frames on the basis of the log-likelihood. However this approach was not pursued since the use of an LP synthesiser was known to be only a temporary measure.

5.6 Results

Modified Rhyme Tests (MRTs), as described in Chapter 4, were conducted for both the P-method and I-method versions of the top ranked configuration in Table 5.1, and also the P-method version of the top ranked configuration with 5 states per model. Tests were also conducted for natural speech, and for re-synthesised natural speech. The latter was generated by performing standard autocorrelation method LP analysis using 25/6 frames, and re-synthesising the speech at a constant fundamental frequency. Voicing was determined by thresholding the zero crossing rate of each frame, and was used to select one of the excitation signals described in Section 5.3.3 during synthesis. This speech represents the best that the current system could hope to achieve.

The results of the MRTs are shown in Table 5.2. The results showed that the P-method was slightly more intelligible than the I-method, as expected, and that there was no real

System	Mean Error Rate (%)
natural speech	0.7
re-synthesised natural speech	3.3
rank-1 P-method synthesis	33.0
rank-7 P-method synthesis	34.0
rank-1 I-method synthesis	37.3

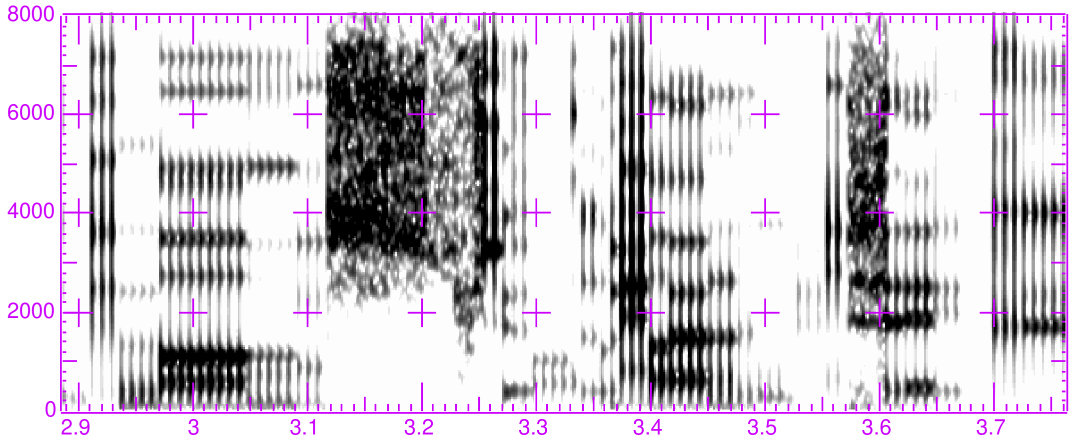
Table 5.2: Modified Rhyme Test results for variations on the basic system, re-synthesised natural speech, and natural speech. The ranks refer to the informal speech quality rankings given in Table 5.1.

advantage in moving to five state models. As discussed in Chapter 4, none of the scores were significantly different in a statistical sense, since this is hard to achieve with such small samples. Analysis of the MRT results show that 74% of the errors with the top ranked P-method system were due to poorly synthesised plosives. The reasons for these errors are discussed in Section 5.7.

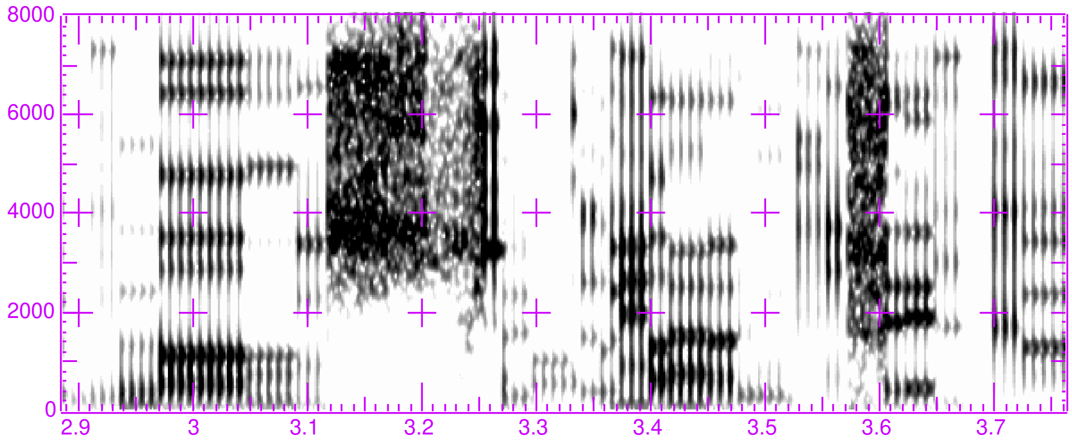
Figure 5.2 shows wideband spectrograms of the sentence fragment "vast Atlantic", taken from the utterance "When a sailor in a small craft faces the might of the vast Atlantic Ocean today...". The speech in Figures 5.2(a) to 5.2(d) were generated using the F, C, I and P method versions of the top ranked system in Table 5.1. The speech in Figure 5.2(e) is re-synthesised natural speech, and that in Figure 5.2(f) is natural speech. All the synthetic speech was synthesised in a monotone at 116Hz, which is approximately the average pitch frequency of the speaker used, using a duration stretch factor of 0.1. This stretch factor resulted in the speech being a little too fast to be easily understood, but was used so that the speech was comparable with later, better quality, speech. The speech that these spectrograms were generated from is available as examples 2-5, 7, and 36, on the accompanying compact disc.

The overall sound of the synthetic speech was surprisingly fluent, and the durations were particularly good. The spectral quantisation resulting from the use of a single set of LP parameters to synthesise each state can clearly be seen in the spectrograms. As expected, in Figure 5.2(a)-(c), where the LP coefficients were based on a single speech segment, the formant bandwidths were slightly narrower, and hence the speech slightly more resonant, than in Figure 5.2(d).

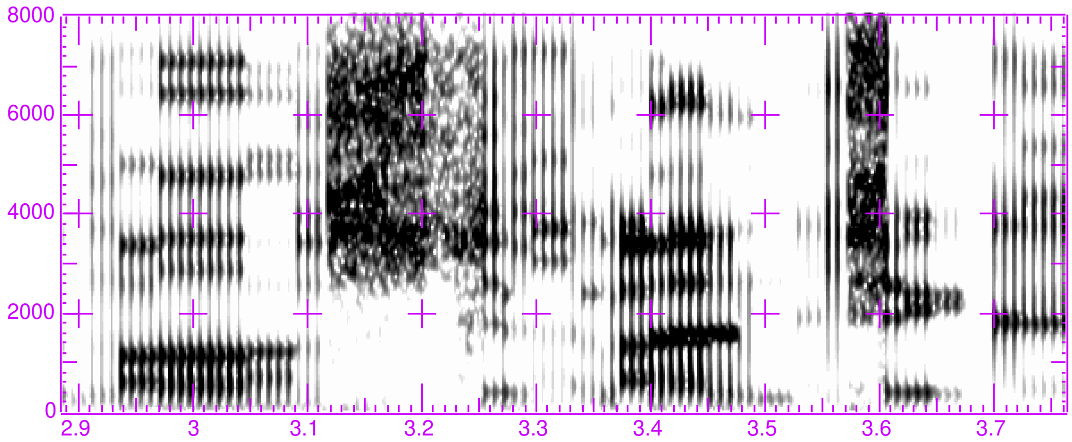
The biggest problem with the speech, as revealed by the MRTs, namely the poor reproduction of plosives, can clearly be seen in the spectrograms. The speech in Figure 5.2(d) makes no real distinction between the closures and the bursts of both the /t/ in "vast" and the first /t/ in "Atlantic", the second of which also suffers a voicing error. The second /t/ of Atlantic was better produced, with only a small amount of noise in the closure, plus a two pulse voicing error just before the burst. The closure and burst of the final /k/ of "Atlantic" were also reasonably distinct, except that the burst was produced with the wrong voicing. The speech in Figures 5.2(a)-(c) suffers similar problems.



(a) Synthesis using the F-method

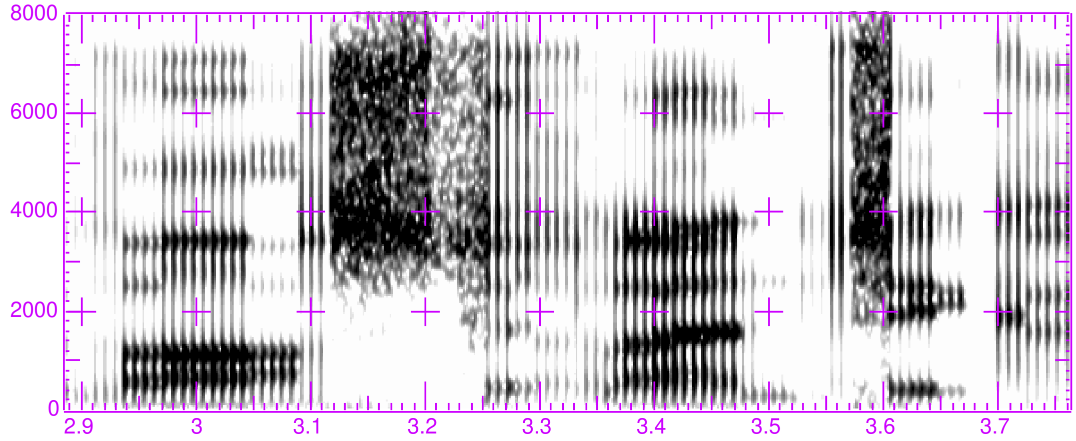


(b) Synthesis using the C-method

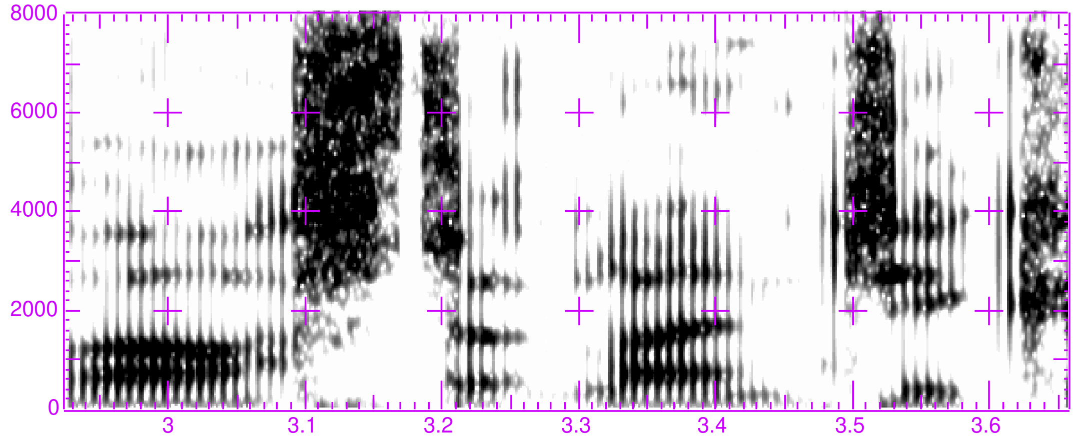


(c) Synthesis using the I-method

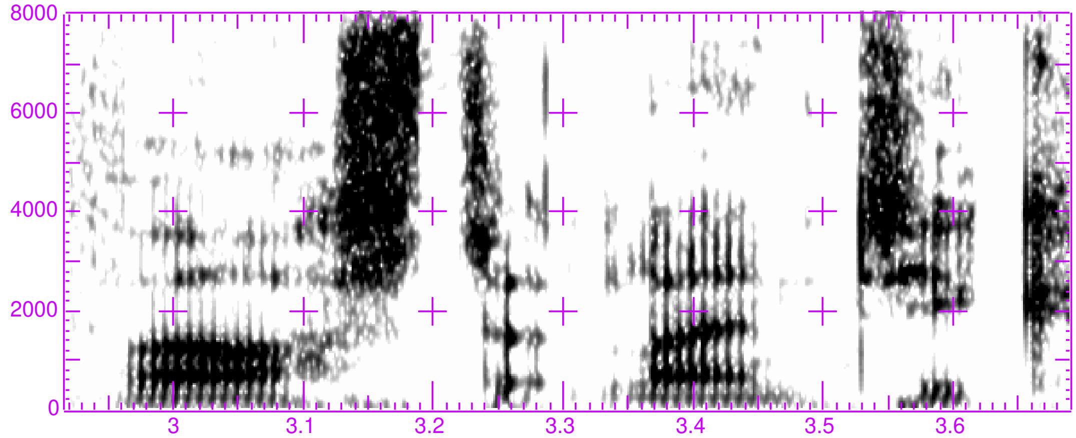
Figure 5.2: Wideband spectrograms of the sentence fragment "...vast Atlantic...". The synthetic speech was produced using the F, C, I and P method versions of the top ranked configuration of the basic system presented in Table 5.1, and by re-synthesising natural speech.



(d) Synthesis using the P-method

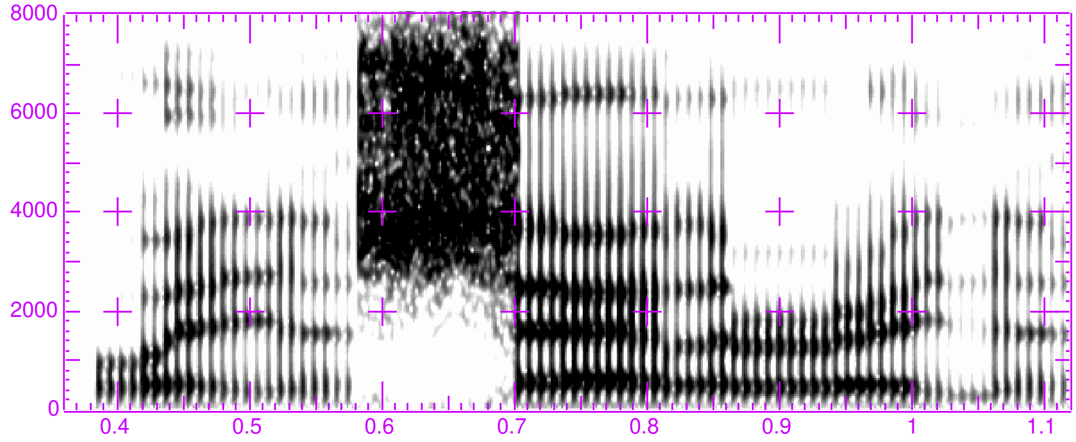


(e) Re-synthesised natural speech

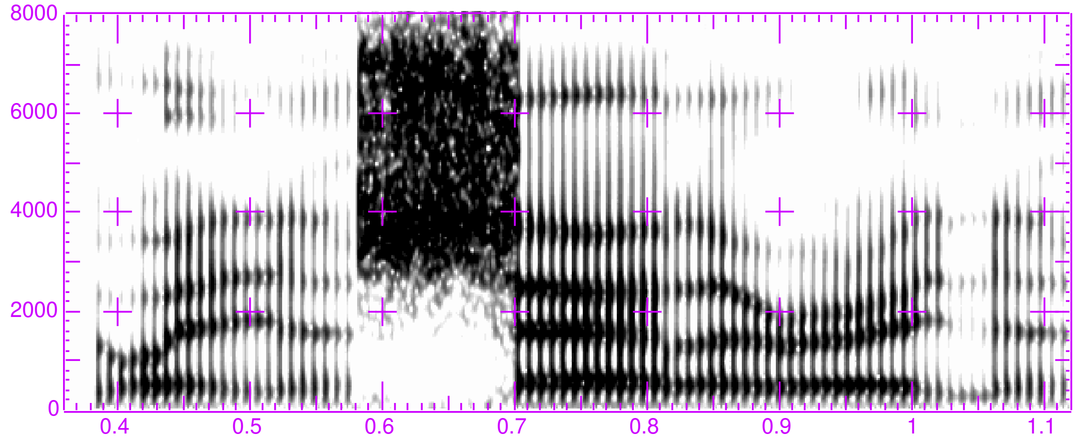


(f) Natural speech

Figure 5.2 (continued): Wideband spectrograms of the sentence fragment "...vast Atlantic...". The synthetic speech was produced using the F, C, I and P method versions of the top ranked configuration of the basic system presented in Table 5.1, and by re-synthesising natural speech.



(a) Synthesis using the P-method



(b) Synthesis using reflection coefficient smoothing

Figure 5.3: Wideband spectrograms of the sentence fragment "When a sailor in a...". The synthetic speech was produced using the standard P-method and by smoothing the P-method reflection coefficients between the centres of consecutive clustered states.

5.6.1 Parameter Smoothing

An experiment was conducted to determine whether the spectral quantisation, and the consequent formant discontinuities present at state boundaries during synthesis, were causing significant degradation to the quality of the synthesised speech. The reflection coefficients used to generate the speech during synthesis were smoothed by using the P-method values at the centre of each clustered state and linearly interpolating them between consecutive centres. Smoothing was only applied to coefficients between clustered states representing voiced, non-silent phones. Although the spectrograms appeared more natural (see Figure 5.3) the resulting synthetic speech, available as example 6 on the accompanying compact disc, was audibly indistinguishable from speech synthesised using the standard P-method. Note that this negative result did not mean that formant discontinuities are not a problem in general, but only that the degradation they introduced into the basic system was insignificant compared to other problems present. Later, in Section 8.5.1, when the system was much improved, it was demonstrated that formant discontinuities are indeed undesirable.

5.7 Discussion

The MRTs made it clear that the system was very poor at synthesising plosives. Examination of the database and the state alignments suggested several possible reasons for this poor performance.

It was found that many plosives specified in the phone level transcriptions were either partially missing (i.e. unreleased), or even completely missing from the speech of the database. The latter was due mainly to lazy, imprecise speech on the part of the speaker used to record the database. The former was also amplified by the laziness of the speech, but it is likely that this effect would still be present to some extent in the fluent speech of even a careful speaker. Completely deleted phones caused problems because the corresponding model was then aligned to the speech of another phone. This degraded the quality of both the model of the phone in question, and the models of neighbouring phones. Unreleased plosives caused less of a problem provided that they were consistently unreleased in the same immediate phonetic context, in which case all three states could model the closure. However, this was not always the case, and occurrences of both the released and unreleased forms of plosives were found in the same immediate phonetic context. In this case the system tried to model both with the same triphone, resulting in blurred model parameters.

The examination also suggested that the time-scale of the speech events present in plosives relative to the frame size was important. The duration of closures in the database was found to be approximately 30-70ms. When present, the following release either took the form of a burst of turbulent noise lasting 20-60ms, followed by a further 10-30ms of aspirated noise, in the case of voiceless plosives (/t/, /k/, & /p/), or of a burst lasting 15-40ms followed by formant transitions into the adjacent vowel, in the case of voiced plosives (/b/, /d/, & /g/). Given these short time-scales, it is clear that the treatment of the speech signal as stationary 25ms frames undoubtably caused some degradation of the synthesis parameters. The high quality of the re-synthesised natural speech shows that the framing effect on its own leads to only a small degradation. However, it is likely that the framing effect combined with the averaging involved in estimating the HMMs amplified the effect, resulting in poor models.

The effects discussed above resulted in poor quality plosive models, which led to poor plosive alignments, and hence poor synthesis parameters.

Chapter 6

Modelling Improvements

This chapter presents the improvements to the basic system made in the areas of automatic phonetic transcription generation, automatic segmentation, and state clustering. Many of these improvements were made whilst still using an LP synthesis scheme, but the need for some of the improvements only became apparent once the transition to a PSOLA synthesis scheme had been made. The implementation of the PSOLA scheme is described in the next chapter.

6.1 Improvements in Transcription & Segmentation

As discussed in Section 5.7, the MRT results of the basic system indicated that major improvements were required in the treatment of plosives. This section describes the improvements made in both transcription and segmentation, which were largely in this area.

6.1.1 Optional Bursts

Both the released and unreleased forms of some plosives were found in the same phonetic context. For example, /n-t+sp/ was seen to be released in instances of "Dent", "descendent" and "opponent", but unreleased in instances of "important" and "don't". When not present the states of the model representing the burst were forced to align to some other part of speech, and so the burst states were degraded.

To solve this problem two possible solutions were considered. The first involved altering the transition matrix of the plosive models, to make it possible to skip the burst states during both re-estimation and alignment. The second was to split the plosive into two separate models, one for the closure and one for the burst, which was made optional. Though both methods were likely to give equally good monophone models, the second method was chosen since it would also aid clustering by allowing a context of a burst to be distinct from a context of a closure. For example, consider the effect on the /iy/ of "eat" in the word pair "don't eat" if the /t/ of "don't" is either released or unreleased. With the first method the /iy/ triphone is always /t-iy+t/, but with the second the distinction can be made between /tbst-iy+tcl/ and /tcl-iy+tcl/.

In addition to the plosives /t/, /p/, /k/, /d/, /b/, and /g/, the models of the affricates /ch/ and /jh/ were also split into separate closure and burst parts. This was done since these phonemes are often analysed phonetically as consisting of the plosive-fricative combinations /t sh/ and /d zh/ respectively, (Klatt 1987).

Initially the original three state models were split such that the closure had one state and the burst had two states. However, on retraining the system (with no bootstrapping) it was found that the first burst state could choose to model the closure instead of the burst. The "burst" model would then always be present, but with a very short alignment for the second state when the burst was not actually there. To rectify this situation all burst models were also made to be one state models. This alteration also greatly improved the system's ability to accurately place boundaries between the closure and burst parts of plosives.

The introduction of separate closure and burst models meant that the left phonetic context of bursts was always the associated closure. This represented a waste of a context label, which could then perform no useful role during clustering. It meant that it was impossible to distinguish, for example, the short unaspirated burst of "stained", from the longer, aspirated burst of "ascertained", since both were /tcl-tbst+ey/. To rectify this situation the left context of bursts were set to the phone before the associated closure.

The introduction of split plosive models and one state burst models, combined with the variable frame size & rate codings discussed in the next section, led to a substantial improvement in the modelling, clustering, and segmentation of plosives. However, the accuracy of the determination of the moment of burst onset was still limited by the resolution imposed by the use of frames. With the PSOLA system, closure and burst waveform segments which were not previously adjacent were concatenated during synthesis, and therefore it was important to ensure consistency in the placing of closure-burst segment boundaries relative to the moment of burst onset during system construction. If this was not done then, for example, a closure segment ending with a burst onset could be synthesised adjacent to a burst segment beginning with a closure, resulting in a double burst. The coding used for plosives was 6ms frames with a 4ms frame shift, and so to avoid the resolution problem all closure-burst state boundaries were shifted 4ms earlier in the final state alignment, ensuring that (nearly) all burst segments began with a moment of closure.

6.1.2 Variable Frame Sizes & Rates

The short time-scale of the speech events in plosives means that relatively long 25ms frames lead to poor models. To understand why, consider a typical burst of, for example, 40ms duration. With a 25/6 coding only three frames lie wholly within the burst region, with approximately six more straddling its boundaries. The state Gaussian, calculated by the BW algorithm, is then likely to be composed of large contributions from the three frames inside the burst, and smaller contributions from the other six frames. The low ratio of frames fully inside the burst to frames straddling its boundaries results in a blurring of the parameters of the state Gaussian.

A system was introduced in which shorter, more numerous frames were used to code plosives, fricatives and silence. A first pass was performed using a 25/6 coding to establish which parts of the database corresponded to which types of speech sounds. The database was then recoded using frames of a different size and rate depending on which class of phone was aligned during the first pass. A 25/6 coding was maintained for voiced sounds, since it was desirable that each frame should average over several pitch periods to avoid any additional variation in parameters due to frame placement effects. For unvoiced speech these considerations did not apply, and the lower limit was that frame size below which so little frequency information could be extracted from each frame that the resulting parameters did not enable different phones to be distinguished. Somewhat arbitrarily a frame size of 6ms was selected, which analysis showed was above this lower limit; i.e.boundaries were accurately placed between adjacent fricatives and bursts. A 4ms frame shift was used for fricatives and silence, and a smaller 2ms frame shift for plosives in an effort to better determine the moment of burst release. The exact sizes used were not extensively investigated, and could perhaps be further refined. Looking again at the 40ms burst discussed above, a 6/2 coding places eighteen frames wholly inside the burst region, and only about four frames straddling its boundaries.

Care was needed to ensure that the parameters obtained when using shorter frames were compatible with those obtained when using longer frames. If this was not the case, then the boundaries between different classes of coding would become effectively fixed after the first recoding, because a model trained on one coding would be very unlikely ever to be aligned to speech coded differently. To ensure energy compatibility, the speech in each frame was scaled to have as much energy as it would have had if it had come from the largest frame being used anywhere in the current coding, that is, the speech samples were scaled by a factor of $\sqrt{maximum\ frame\ size/current\ frame\ size}$.

The steps taken to ensure MFCC compatibility between codings are most easily explained by first describing the method used to calculate these parameters in some detail. The coding was performed using the appropriate tool from HTK Version 1.5. The speech frame was first zero padded up to the next smallest frame with a number of samples equal to a power of two, to enable a radix 2 Fast Fourier Transform (FFT) routine to be used to obtain the Discrete Fourier Transform (DFT) of the signal. It can be shown that zero padding prior to taking a DFT does not change the shape of the spectrum obtained, but merely produces an interpolated spectrum with more points. As shown in Figure 6.1, the magnitude spectrum was then pooled into, typically 24, mel bins using triangular filters. The filters were defined by their centre frequencies, which were equally spaced in mel-scale frequency between 0 and the mel-scale equivalent of half the sampling frequency. The mel frequency scale is a perceptual scale derived from experimental research into how humans perceive pitch, (Parsons 1986). The equation used in HTK to approximate the relationship between linear frequency (Hz) and mel-scale frequency (mels) is given in Figure 6.1. Finally, those mel bins whose contents were less than one were set equal to one, logs were taken, and a cosine transformation performed to calculate the MFCCs.

From Figure 6.1 the effect that using a different FFT size in the analysis would have can be appreciated. Since the sampling frequency is fixed, if the number of mel bins used remained the same, then the only thing which would change would be the number and spacing of the FFT spectral components. However, this new spacing would alter both the number and the weights of the components being pooled into the mel bins, the effect

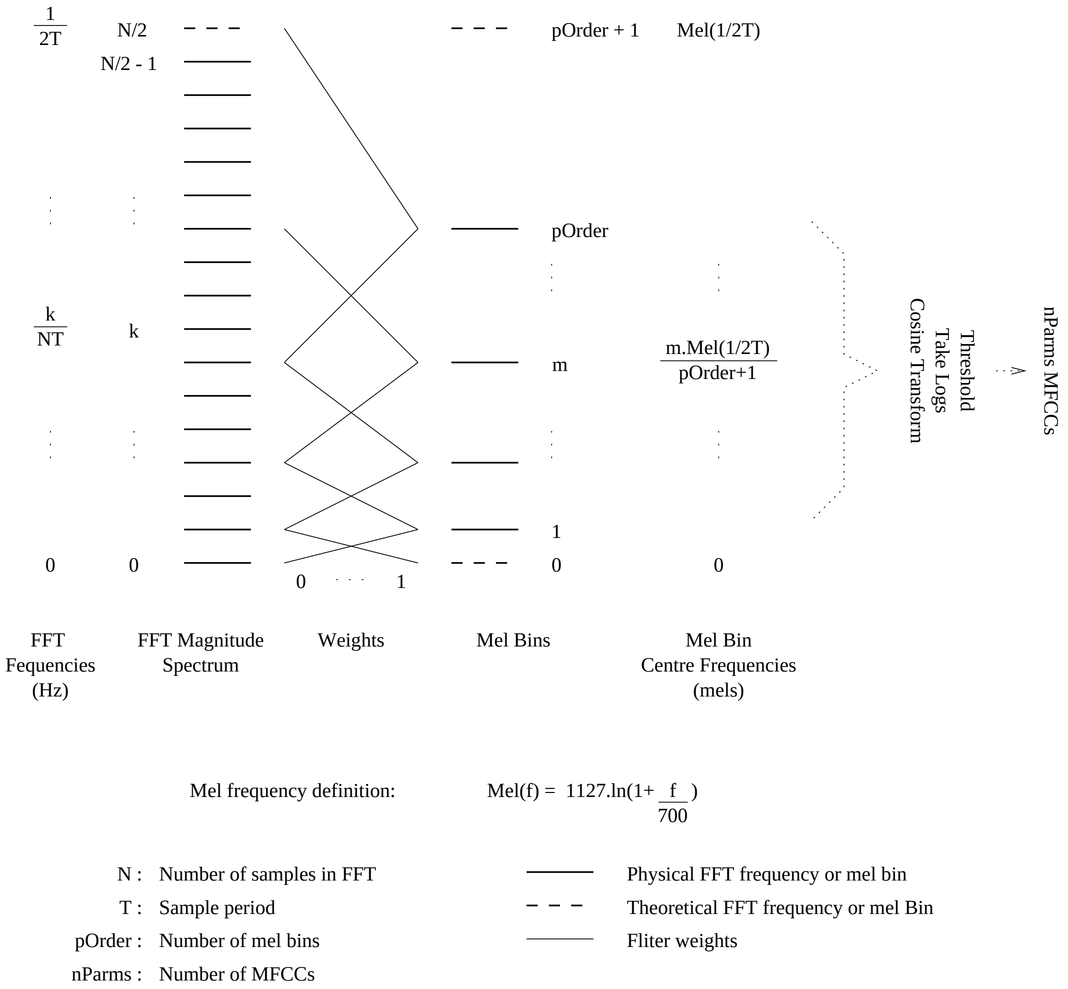


Figure 6.1: The calculation of MFCCs within HTK Version 1.5.

of which could be significant for the lower order bins which only have a few contributing components. Therefore, in order to avoid these quantisation effects, every speech frame was zero padded up to the size of the largest FFT required for any frame in the current coding. The number and weight of the components pooled into each mel bin was then the same for every frame size, with only the amount of interpolation effectively employed to estimate the FFT components varying between frame sizes. By Parseval's Theorem, the above energy scaling also ensured that the average power spectrum of the speech was independent of the frame size, before the magnitude spectrum was pooled into the mel bins. This arrangement was seen to ensure that both MFCCs and energy parameters obtained from the same region of speech were very similar when coded either with 25ms or 6ms frames.

The delta and acceleration parameters were calculated for each frame using itself, and the two nearest frames on either side. No alteration was made to this arrangement when the data was recoded, and so the deltas and accelerations therefore referred to changes on a much shorter time-scale when smaller frame sizes and shifts were used. The new coding was used to re-estimate the HMMs, and obtain new phonetic transcriptions and alignments. The change in frame size and shift was transparent to the HTK re-estimation and alignment tools, because they worked only with sequences of parameter vectors, and did not use any information about the source of these. The only exception to this was the alignment times produced, which needed to be corrected to reflect the underlying framing scheme. The recoding procedure was repeated several times during the monophone training stage of the system's construction.

When the HMMs were re-estimated on the new coding it was found that log likelihoods per frame generally decreased. However, whilst that of silence decreased by 8.0, and those of fricatives by an average of 5.7, those of bursts and closures only decreased by an average of 2.0. (A typical log likelihood was about -70). The fall in log likelihood can be explained by considering that long frames represent an average over more speech than do short frames. Parameters obtained from long frames therefore tend to be more similar to each other than those obtained from short frames. Gaussians constructed from them therefore have smaller variances, and hence higher log likelihoods. The resultant drop in log likelihoods when moving to shorter frames was offset by the fact that the models were improved due to the framing effects discussed above. The relatively small drop seen in the log likelihoods of closures and bursts is therefore thought to be due to substantial improvements in the quality of the models. The fricative and silence models saw larger drops because the improvement in model quality was much smaller, due to the longer average durations of these sounds.

As described above, the use of different coding schemes was determined entirely by the class of phone aligned on a first pass. This was not ideal, since many examples of nominally voiced speech being produced unvoiced, and vice-versa (particularly with /hh/) have been observed. Using a voicing determination algorithm to assign coding schemes would perhaps be preferable. Introducing pitch synchronous frames for voiced speech might also be beneficial.

6.1.3 Silence Modelling

In the basic system two silence models were used, one with three states, and the other with one state. The presence of two silence models was largely historical. In the HTK recognition system the one state model was used for short inter-word silences, where its presence was ignored in describing the context of neighbouring phones. It was in fact a so called 'tee'-model, having a skip transition from entry to exit state, enabling it to be skipped completely during re-estimation. However, in the basic system this feature was largely redundant since the models were inserted optionally by Viterbi alignment, rather than always being present at word boundaries. Furthermore, the one state model was not ignored in describing the context of neighbouring phones. It therefore performed no additional task to the three state model, and so the two silence models were merged into one.

An analysis of the silences present in the M1 database was conducted to establish the most appropriate minimum duration for the silence model, principally to investigate whether a one state model was necessary. The first 10 (approx.) silences aligned to the data with durations of 6ms, 12ms, 18ms, ..., 54ms were examined (using a 25/6 global

Silence Duration (ms)	No. in Database M1	Checked	Present with stated duration	Present, but with longer duration
6	10	10	1	1
12	15	15	2	0
18	40	10	0	2
24	48	10	1	4
30	47	10	0	2
36	28	10	1	3
42	30	6	0	1
48	16	6	1	4
54	20	6	1	2

Table 6.1: Analysis of silences present in the M1 database. Columns show silence duration, total number of such silences, number checked, number of those checked present with stated duration, and number of those checked present, but with longer duration.

coding). The results of the analysis are shown in Table 6.1.

As can be seen from the table, only a very small proportion of the aligned silences which were shorter than 48ms in duration were actually present in the database. Furthermore, most of the silences referred to in the last column of the table, which were actually present but with a longer duration, were above 40ms or so in duration. The analysis therefore demonstrated that a short single state silence model was not required, especially when using a 6/4ms coding, since there were only a handful of genuine silences shorter than 40ms in the entire M1 database. Furthermore, since many of the silence insertion errors occurred precisely because a short silence could be inserted where there was none, a 7 state left-to-right silence model was adopted, which had a minimum duration of 28ms on a 6/4 coding.

An analysis of the locations of the misplaced silences was also instructive, and led to improvements in the quality of the phonetic transcriptions. A very large number of the silences aligned in error were placed either in the initial closures of words beginning with a plosive, in the final closures of words ending in an unreleased plosive, or in the double closure formed by two such words together, such as "ape descended". These extra silences were degrading the transcriptions, since the aligned phone at these points should be the relevant closure(s). The effect of this degradation was to deny appropriate speech to the closure models of some contexts, to lead to an under estimation of closure durations, and to give incorrect contexts to many phones. From Table 6.1 it can be seen that it is likely that there were at least 200 such errors in the database, with possibly many more longer silences being placed in double closures. Although this represented possibly less than one error per sentence, it was thought to be important because it would significantly affect models and durations in some contexts. The following heuristic was therefore employed to remove such silences from the database:

- Remove all silences (sp), from:
  - /∗cl sp ∗cl/ if the combined duration is less than 140ms
  - /∗cl sp/ if the combined duration is less than 100ms
  - /sp ∗cl/ if the combined duration is less than 100ms

where ∗cl represents any closure. The duration at which a closure becomes a pause is something of a subjective decision. It may also vary with speaking rate and perhaps between speakers. The duration thresholds used were determined from examination of the M1 database by one person, and as such were both speaker dependent and subjective. However, they would undoubtably correct many hundreds of errors in the short closures of any speaker, and would be likely to cause only a fraction of this number of rather subjective "errors" in those cases with closure durations near the thresholds.

It was also realised that the durations of closures adjacent to other closures or silence were inherently unreliable, and so these were no longer used to estimate closure durations. This was achieved after converting the monophones models into context dependent models by replacing such closures with a context independent dummy model, /cl/. Performing the conversion before replacing the models ensured that the original contexts were maintained for the neighbouring phones, and that contexts of /cl/ were not introduced. This alteration did mean that the durations of double closures were never properly estimated, but since the system released all bursts during synthesis these were never needed.

6.1.4 Phone Deletion

From an analysis of the phonetic transcriptions generated by the system it became clear that some phones present in the transcriptions were not actually present in the database. This was because the phonetic transcriptions were generated using a dictionary, which only contained citation form pronunciations. Experiments were therefore conducted with the aim of automatically identifying such deleted phones. The obvious experiment, of making all phones optional, was tried but failed completely, with many phones clearly audible in the database not being aligned. Other, less drastic, approaches were therefore sought.

It was reasoned that phones which were absent from the database should have very short alignments if the database was aligned to a transcription in which they were present. The experiments therefore examined those phones with "suspiciously short" alignments. To amplify the effect, the transition matrices of all phones were altered to enable skip transitions prior to producing an alignment. Any phone which was not present could then be aligned to as little as one frame of speech (a minimum enforced by the HTK alignment tool). "Suspiciously short" was defined as closures and fricatives shorter than 6ms, and voiced sounds shorter than 10ms, using the codings described in Section 6.1.2. Bursts were not examined since these were already present optionally. The experiment was repeated using both 2 and 3 Gaussian mixture monophones, and a 3 mixture set which had been re-estimated using the skip transition matrices.

All the experiments yielded very similar results. In all experiments about 3,900 phones were marked as suspiciously short, of which about 1600 were closures, 600 /ax/, 300 /ih/

and 260 /l/. In all cases it was found that, of those phones marked as suspiciously short, only approximately 50% of the sample examined (about the first 18 phones) were not present. Removing the marked phones from the transcriptions would therefore be likely to cause as many problems as it would solve. Making the marked phones optional in future alignments was not attempted due to the failure of the experiment where all phones were made optional.

However, it was noticed that the success rate in identifying missing closures seemed to be higher than for other phones, and so this possibility was investigated. The first 16 closures in the 1 mixture and 3 mixture cases were examined, and in both cases 75% were found to be absent. Since closures were only one state models, another experiment was conducted in which the skip transitions were not introduced, and still 75% of the marked closures were found to be absent. Removing these closures from the transcriptions would therefore have solved three times as many errors as it would have introduced, and so this was thought to be worthwhile. Rather than removing them, the closures were at first made optional, similar to the way in which all bursts had previously been made optional, which had worked quite well. However, problems arose with the PSOLA system when closures were incorrectly omitted and the adjacent bursts selected for use in synthesis. An alternative scheme was therefore adopted in which the marked closures were swapped for a dummy model /cl/ which would never be used in synthesis, or to estimate synthesis durations. During the conversion to context dependent models, /cl/ remained a monophone, and since the closure was likely to be absent, all right and left contexts of other phones which were specified to be /cl/ were altered to refer to the first adjacent phone which was not /cl/.

Many of the identification errors with /ax/, /ih/, & /l/ were found to be with very short occurrences, of only 2 to 4 pitch pulses in duration. This indicated that a very short alignment did not necessarily mean that a phone was absent. It also suggested that perhaps some phones other than plosives should have fewer than three states, or different model structures, and that a pitch synchronous coding might be more successful.

6.1.5 Stressed Vowels

The work in the literature referred to in the introduction by (Nakajima 1993) and (Wang et al. 1993) showed that stress can be an important clustering feature. Although stress information for the database was not available, the pronunciation dictionary used did contain lexical stress information. Lexical stresses were not necessarily realised in the speech of the database, and furthermore, other non-lexical stresses may have been present. However, it was likely that using a label based on lexical stress was better than making no stress distinction at all, since the clustering algorithm would only use the label when the presence and absence of the label corresponded to a significant difference in the acoustic realisation of the phone in question.

Vowels with primary stress, secondary stress and no stress were distinguished in the dictionary. In order that pronunciations which differed only in stress level could be assigned to the database by the system, distinct monophones were required to model different stress levels. For each vowel, where sufficient data existed, deemed to be more than 50 occurrences of each stress level, all three monophones were distinct. Where only the combined occurrences of primary and secondary stresses exceeded 50 the stressed phones were tied, and there were only two distinct models, and finally where there was insufficient data even for this, all three models were tied. The models were used to determine the phonetic transcription of the database, and then untied before the conversion to context dependent models.

6.1.6 Syllable Effects

The determination of syllable boundaries is a difficult, and somewhat imprecise subject. The Longman Pronunciation Dictionary, (Wells 1990), describes three methods for performing syllabification which are currently in use, each of which gives slightly different results. Some variation also seems to occur in deciding which phones are truly syllabic, and which are associated with an adjacent schwa. The phonetic pronunciations comprising the BEEP dictionary were compiled from many sources, and unfortunately pronunciations were not always consistent regarding the latter point (BEEP gives no explicit syllable boundary information). For example, the pronunciation of "button" was given as /b ah t n/, but that of "Aston" as /ae s t ax n/ and that of "bottom" as /b oh t ax m/. A degree of normalisation was eventually introduced in an effort to improve this situation, but the following discussion refers to unaltered BEEP pronunciations and phones.

Problems were experienced with the PSOLA system with the synthesis of /l/s. In English the phoneme /l/ exists in at least two, and possibly three, different allophones, being the "light" syllable initial /l/ of "left", the "dark" syllable final /l/ of "small", and the syllabic /l/ of "bottle". The different allophones can occasionally occur in exactly the same immediate phonetic context, for example the dark /l/ in "militant", and the light /l/ in "belittle", which are both in the same triphone /ih-l+ih/. Note that not even the word initial / word final labelling discussed in Section 6.2.4 would enable these two cases to be distinguished. The lack of an appropriate label to enable the separation of the different allophones led to problems in synthesis with the PSOLA system. Different allophones with similar phonetic contexts were pooled into the same clustered state, and then during synthesis the segments concatenated to construct the /l/ required were selected from both allophones. This resulted in formant discontinuities between adjacent segments, which were heard as glitches in the synthetic speech. The errors were compounded in the synthesis of words containing the phone sequence /ow l/, since in modern British English an /ow/ with a following dark /l/, as in "sold", is pronounced very differently to an /ow/ with a following light /l/, as in "solo". Note that again, the same triphone /s-ow+l/ is present in both words, and that some form of syllable marking would enable the two cases to be distinguished.

Similar cases can be made for the syllabic and non-syllabic forms of /l/, /n/ and /m/. For example, "evilest" usually contains a syllabic /l/ and "livelihood" does not, but both contain the /v-l+ih/ triphone. In "buttoning" the /n/ is syllabic, and in "witness" it is not, but both contain the /t-n+ih/ triphone. Syllabic /m/s were usually represented in BEEP as /ax m/, but the same problem could still occur, as demonstrated by the examples "intermix" and "welcoming", both of which contain /ax-m+ih/, but only the second of which is syllabic in nature.

Although the cases described above are perhaps the most important occasions on which a knowledge of syllable boundaries would be useful, it is possible that other smaller gains would also be achieved if syllable information was widely available.

It is clear that, in addition to a consistent transcription of syllabics (with regard to schwa), a general marking of syllable boundaries would be very useful as a clustering feature. However, due to the inconsistencies between the various approaches described in (Wells 1990), the difficulties experienced by (Jones 1994) in implementing an accurate automatic syllabification system (11.5% of the words in his lexicon were deemed to have an incorrect *number* of syllables), and a lack of time, the simpler task of trying to decide only when /l/, /m/, and /n/ were dark, light, or syllabic was attempted.

The dictionary constructed by merging the Dragon Wall Street Journal Pronunciation Lexicon and the Dragon Resource Management Lexicon, was used to automate the procedure as far as possible. This dictionary, which is for American English, has both separate phones for syllabics, and marks to indicate syllable position. It was assumed that, if the number of /l/s, /n/s and /m/s in a word was the same in American English as in British English, then the syllabic nature of these phonemes was unlikely to change between languages. Syllabics in the Dragon dictionary, marked /ul/, /un/, and /um/, were used to update the corresponding phonemes in the BEEP pronunciations to /l1/, /n1/, and /m1/. Where a schwa was present in the BEEP pronunciation to the left of the phoneme now marked as syllabic, it was removed. Syllable final /l/s in the Dragon pronunciations were used to update the corresponding /l/s in the BEEP pronunciations to /l2/. Since BEEP was much bigger than the Dragon dictionary, not all words in BEEP were updated. Therefore all the words occurring in the training data containing /l/s, /n/s, or /m/s, which had not been updated by this procedure (several hundred), were checked by hand. An attempt was also made to correct words containing dark /l/s in a non syllable final position, such as "gold", which had all been missed by the automatic procedure. Correcting only the words in the training data did mean that new words could be synthesised with the wrong type of sound, but they were at least synthesised without serious formant discontinuities. The new pronunciations were used to train separate monophone models where sufficient data existed, which was determined using a similar method to that described in Section 6.1.5 for stressed vowels.

6.1.7 Phone Substitution

Phone substitution has not been found to be nearly as widespread as either phone insertion (mainly extra silences), or phone deletion (mainly bursts and closures). However it is a well known phenomenon, and so undoubtably occurred in the database. For example, vowels have a tendency to be reduced to schwa, particularly in lazy speech. There are also many cross word effects which can result in a different phone sequence to that expected by simply concatenating individual word pronunciations. A list of such cross word effects is given in (Giachin et al. 1991), an example of which is the frequent conversion of a word final /d/ followed by a word initial /y/ to a single /jh/.

Such effects will be handled well by the system provided that within a single context the same substitution is always occurring. For example, if the vast majority of the occurrences of a vowel in a particular context are reduced, then the clustered states representing that vowel will model the schwa sound instead of the labelled vowel sound. In synthesis this will result in vowels in such a context being produced as schwa, without the phone string used having to explicitly state that the vowel is reduced, which is very useful. Indeed, it has been noticed that the clustering question "C_No_Stress", meaning "Is the central phone unstressed?" was often asked very high up the tree when clustering vowels. It is suggested that this was because there was a significant acoustical difference between the stressed and unstressed occurrences of the vowel because it was often reduced, to some degree, when it was unstressed.

However, in other cases, phone substitution may not be consistent within a context, and then, unless one form is clearly dominant, problems can be expected. One possible solution is that rules, such as those given in (Giachin et al. 1991), could be included both during system construction, to provide alternative pronunciations to the transcription system, and then also during synthesis, to allow a choice between the different pronunciations to be made. Another possibility would be to introduce some new clustering feature(s) to enable the different substitutions to be distinguished, in much the same way that "unstressed" may be acting as a label for vowel reduction. However, it is not clear what these feature(s) might be.

No solutions to the problem of phone substitution have been incorporated into the system described in this thesis. This is principally because no problem has yet been found to be caused by phone substitution, and hence the problem is thought to be a relatively minor one.

6.1.8 Coding Alterations

After all the improvements described in the previous sections had been implemented it was still found that the system was making some errors with bursts, closures, and silences. Specifically, it appeared that the system had difficulty distinguishing closures and silences from neighbouring sounds. These errors were easily seen by examining the labelled waveform files, due to the large difference in energy between the parts of the speech signal which were wrongly labelled. It was thought that such errors were possible because the single Gaussian representing energy information in the models was dominated by the 12 MFCC Gaussians encoding the frequency information. Several alternative coding systems were therefore tried, principally to examine whether giving the energy term more weight relative to the MFCC terms was beneficial.

The first 13 such errors in the M2 database (contained in the first three and a half utterances), were listed, and the effect upon them of alternative codings examined. The results of the experiments are presented in Table 6.2. Mean subtraction (*m.s.*) involved subtracting the mean value of a parameter over each utterance from each value of the parameter within that utterance. The term *normalisation* is used to describe the process of making each sentence's maximum log energy be the same value.

Both experiments 1 and 3 introduced only one new error into the speech under examination. If the performance on the sentences analysed was typical of the whole database, then it is likely that over 800 errors were corrected in the M2 database using the codings

Expt. No.	MFCC Parameters	Log Energy Parameters	Errors Corrected (/13)
0	12 m.s.	1 absolute	0
1	12 m.s.	1 absolute, 1 m.s.	10
2	12 absolute	2 absolute	5
3	12 m.s.	2 normalised	9
4	12 m.s.	1 normalised	1

Table 6.2: Coding experiments conducted to enhance closure, burst, and silence identification. For definitions of m.s. (mean subtraction) and normalisation see text.

of experiments 1 and 3. The coding scheme of experiment 3 was selected for use, since the result was not significantly worse than that of experiment 1, and it was more consistent with previous work and normal practice with HTK. Note that all the errors still occurring in experiments 1 and 3 were problems associated with /ch/ and /jh/. These phones had previously been seen to be difficult to segment, and this suggests that perhaps it was not appropriate to treat them as plosives for this purpose.

In fact, although the coding of experiment 3 was used to determine the phonetic transcription of the database, the synthetic speech was judged to be slightly inferior to that of experiment 0. Therefore, the second energy term was removed from both the coded speech data and the models after the phonetic transcriptions had been fixed, before the conversion to context dependent models. The synthetic speech then produced was judged to be better than that in experiment 3. Although these judgements were only carried out by one person, and were probably highly subjective, the alterations carried out have introduced the notion that perhaps *different* codings are appropriate for the transcription and clustering stages of the system's construction. That is, at the transcription stage the coding is required to provide the maximum level of discrimination between different phones, whereas at the clustering stage the coding is required to group those states within a phone which are audibly the most similar, and that the optimum codings in these two cases are not necessarily the same.

6.2 Improvements in Clustering

This section describes the improvements made to the clustering procedure used in the system. The improvements were introduced either to fix problems observed in the synthetic speech produced by the system, or to include clustering criteria which were reported to be useful in the literature.

6.2.1 Stopping Criteria

As described in Section 5.2.3, the basic system used two stopping thresholds to terminate the clustering procedure. One of these specified the minimum number of frames of speech required to be in each leaf node, and the other the minimum increase in log-likelihood required to cause a node to be split. In order to model the speech as accurately as

No. Occurrences Threshold	No. Clustered States
10	6034
12	5200
16	4129
20	3435
40	1854
60	1272

Table 6.3: The number of clustered states resulting from using different number of occurrences thresholds during clustering, with a one hour speech database.

possible the thresholds were set to produce a large number of clustered states, typically about 5000. However, this resulted in problems, due to the limited amount of training data available. Even with the minimum number of frames threshold set as high as 64, some states were seen to occur only 1 or 2 times in the database. This low number of occurrences was undesirable for several reasons. States which only occurred 1 or 2 times were not subject to the usual constraint that state parameters must reflect a large number of different occurrences. As a result such states could model sounds which did not wholly correspond to the phone label attached to them, resulting in problems when occurrences of these states were used in synthesis. A low number of state occurrences could also give unreliable state duration information, especially state duration variances. Finally, a low number of state occurrences did not offer many candidates for selection for use in the PSOLA system.

The clustering algorithm was therefore altered to use a minimum number of state occurrences threshold instead of a minimum number of frames. Note that the new system automatically enforced the old, since at least once frame had to be associated with each state occurrence. The minimum change in log-likelihood threshold was set to zero, since once the minimum number of occurrences of each state had been enforced it was thought desirable to have as many states as possible, in order to obtain the maximum possible number of independent durations. Several occupancy thresholds were tried, and the number of clustered states resulting is shown in Table 6.3.

As can be seen from the table, the number of clustered states produced dropped off rapidly as the threshold increased. The desire for a large number of states was offset by a requirement that there be enough speech frames associated with each state to properly estimate a Gaussian distribution. Therefore the threshold was set at 12 occurrences in order to produce a similar number of states as before, whilst still ensuring a reasonable minimum number of frames were associated with each state. Note that, since each state occurrence usually contained many frames, the number of frames associated with each state was usually many more than this minimum figure.

As discussed in Section 2.3.2, the work done by (Wang et al. 1993) demonstrated that the over-fitting of decision trees to data can occur if simple minimum occupancy stopping criteria are used. It is therefore possible, and indeed quite likely, that the above stopping criteria do lead to over-fitting of the trees to the data. Such over-fitting might, for example, split a node into two sub nodes on the basis of contexts which were seen during training, neither of which were then appropriate to synthesise a new context which was not seen in training, for which using the more neutral unsplit node would have been better. A backing off arrangement, which used higher nodes for synthesis if the exact context to be synthesised had not been seen in training might at first seem an attractive solution to this problem. However, given that there were eventually over 23 million possible context labels, and that only about 13,000 of these were present in the training data, such a solution might mean that descending a tree during synthesis often never moved off the root node. A more advanced stopping criterion, such as the cross validation techniques used by (Wang et al. 1993), is probably the best solution to the problem. No such scheme was incorporated into the system described in this thesis, principally because, although over-training may well be occurring, no problem in the synthetic speech produced by the system has yet been found to be caused by it.

6.2.2 Stress Level

As described in Section 6.1.5, where sufficient data was available, three separate monophone models were trained for each vowel, to represent vowels with primary, secondary and no stress. The models were tied at the monophone stage if there was insufficient data for them to be distinct, but then untied at the transition to context dependent models. After further re-estimation, the models were clustered. A single tree was built to cluster corresponding states of the context dependent models of each *base* phone. Base phones were defined as the central phones of each context dependent label, with stress information (and later syllable information and position in word information) removed. Additional clustering questions were added to the question list used in clustering to ask about the stress levels of both central phones and adjacent phones. In this way separate clustered states existed for the different stress levels of a phone only when such a split caused the biggest increase in the log-likelihood of the acoustic data fitting the tree at that point; when the acoustic difference between stress levels was relatively unimportant, such splits were not made, and separate states did not exist.

An analysis of the questions actually selected by the system to build the trees showed that stress often was an important clustering label, sometimes even being the first question to be asked when constructing a tree. As discussed in Section 6.1.7 it is suspected that the "unstressed" label may be acting as a marker for some degree of vowel reduction.

6.2.3 Syllable Effects

As described in Section 6.1.6, where sufficient data existed, different monophones were trained for the light, dark, and syllabic allophones of /l/, and for the normal and syllabic allophones of /n/ and /m/. At the conversion to context dependent models the monophones were untied if necessary, and then re-estimated prior to clustering, in the same way as with stressed vowels. Again, a single tree was built for each base phone, and clustering questions were added to enable the different allophones to be distinguished if required. Specifically, questions were asked to establish whether the central phone was a dark/syllabic /l/, or any other specific allophone, as well as whether adjacent phones were syllabic, or a dark/syllabic /l/, or a syllabic nasal, or any of the individual allophones. Again, the new questions were often found very high up the trees, both of the /l/, /m/ and /n/ phones themselves, and also of many vowels.

The lack of agreement on a general syllabification scheme, as discussed in Section 6.1.6, suggests that perhaps the very notion that speech can be broken into distinct syllables is fundamentally flawed. Although syllable nuclei undoubtably exist, and their location can be agreed upon, it is not clear that the notion of precise syllable boundaries is always justified. This suggests that perhaps human imposed syllable boundaries are a non-ideal solution to the problems presented by syllable dependent allophones, and that perhaps some form of data-driven approach would offer more flexibility. For example, it is possible that clustering on the basis of wider phonetic context could enable different syllable dependent allophones to be separated, without the need for explicit syllable boundary specification. These ideas have not been explored further with the system described in this thesis, but do remain an area for possible future work.

6.2.4 Word Level Effects

A labelling scheme was introduced which caused word final, word initial, and word internal phones to be labelled differently. This was done partly because there was no mechanism in the basic system to enable a cross word triphone to be distinguished from the same triphone in a word internal position, and partly because the research of (Nakajima 1993) showed that word boundary information, particularly the ability to distinguish word final vowels, was useful in clustering. The existing monophone models were cloned into word final, word initial, and word internal monophones just prior to the conversion to context dependent models. Including the other changes discussed above, this led to 286 monophones, 284 of which could form any part of the subsequent triphone models, one of which (silence) remained a monophone, but could appear as a context of other phones, and one of which (/cl/) remained a monophone and was not allowed to appear as a context of any other phone. After re-estimation the clustering was carried out as before. Note that the expanded monophone set meant that over 23 million different context labels were logically possible. Additional questions were added to the clustering question list to ask about the position of phones within words. Specifically, questions were added to ask whether the central phone was in a word initial, mid, or final position, and whether the adjacent phones were in initial, mid, or final positions. In addition *every* previous question about all forms of context was repeated in four forms, one in which the phone position within a word was not an issue, and three in which the set of contexts the question referred to was formed by the intersection of the original question and one of the possible phone positions within a word. This resulted in a question set numbering 1928 questions, though some of these could undoubtably have been expressed more compactly.

Again, an analysis of the questions actually selected by the system to build the trees showed that questions either wholly or partially regarding the position of phones within words were being asked, and were occasionally very high up trees. For a more detailed analysis of which questions were found to be important during tree construction see Section 8.2.

Chapter 7

Incorporating TD-PSOLA

The TD-PSOLA algorithm was incorporated into the system in order to improve the quality of the synthetic speech produced. For a discussion of the limitations of the LP synthesis scheme, as used in the basic system, and a justification for using an alternative scheme, see Section 8.4. TD-PSOLA was selected in preference to either multi-pulse LP techniques, residual excited LP techniques, or formant synthesis, since it appeared to offer the possibility of very high quality synthetic speech, and to be the simplest scheme to implement. The only difficulties in implementing the algorithm were to select which waveform segments to use in synthesis, and to determine the moments of principle excitation of the vocal tract within these segments. The solutions which were adopted to these problems are described below, followed by a detailed description of the TD-PSOLA implementation used in this work, including a demonstration of the performance of the implementation when re-synthesising natural speech.

7.1 Segment Selection

As described in Section 1.4.6, TD-PSOLA does not synthesise speech in itself, but merely enables waveform segments to be smoothly concatenated, whilst altering their pitch and durations. The system described in this thesis had at least 12 occurrences of each clustered state in the database, and therefore in order to incorporate the PSOLA algorithm it was necessary to select one of these occurrences to represent each state during synthesis. Note that although one particular occurrence must be selected each time a state is synthesised, this does not mean that the same occurrence must always be selected. It would be possible, for example, to select occurrences dynamically during synthesis to satisfy some other criterion, such as to reduce segment concatenation discontinuities, the presence of which can seriously degrade the quality of all concatenation synthesis systems. Such approaches were not implemented during the course of this work, but are discussed as possible future work in Section 9.1.3. The algorithm developed in the course of this work selected only one state occurrence to represent each state whenever that state was synthesised. The algorithm consisted of three stages:

Stage (i): Discard all the occurrences of a state which have a duration shorter than 80% of the average duration for that state.

This was beneficial for two reasons. As discussed in Section 6.1.4, phones in the phonetic transcriptions which were aligned with very short durations were often not actually present in the speech database. Including a duration threshold in the selection criteria therefore helped to ensure that the states of such deleted phones were never selected for use in synthesis. The threshold was also useful because, although TD-PSOLA enables the duration of segments to be altered, it is wise to attempt to keep any duration lengthening to a minimum, and certainly duration stretches above a factor of two are to be avoided. As described in Section 8.5.2, each state was synthesised for a duration typically averaging either 1.04 or 1.22 times its average duration, when synthesising continuous speech or isolated words respectively. An 80% threshold therefore meant that even when synthesising isolated words, stretching factors averaged less than 1.52. The figure of 80% therefore represents a compromise between the desire to keep the average stretching factor low, and the desire to let as many occurrences as possible remain available to the later stages of the selection algorithm. Note that this figure also affects the size of the segment inventory which needs to be stored for use in synthesis.

Stage (ii): Discard all the occurrences of a state which have an average short term energy per sample (s.t.e.p.s.) lower than 80% of the average s.t.e.p.s. of the speech from all the occurrences still under consideration.

The occurrences selected to represent each state were selected from all parts of the database, and so were not necessarily similar in energy. As a result, energy levels could fluctuate wildly from one segment to the next during synthesis. Each segment was therefore scaled to the average s.t.e.p.s. of the state it represented during synthesis, in order to reduce these fluctuations. This effectively ensured a degree of energy smoothing, and improved the quality of the synthetic speech. However, problems arose when energies were scaled up by large factors, since both speech irregularities and non-speech sounds could be scaled to energies far higher than those at which they were normally produced, introducing glitches into the synthesised speech. The 80% threshold was therefore introduced to ensure that segments were generally attenuated during synthesis. Again, the desire to set a high threshold, to ensure that segments were always attenuated during synthesis, was offset by the desire to let as many occurrences as possible progress to the next stage of the selection algorithm. The figure of 80% was thus another compromise, though its precise value was not thoroughly investigated.

As described in Section 1.9, energy scales with fundamental frequency (F_0), and therefore the s.t.e.p.s. figure of each state occurrence was dependent on its F_0. The average s.t.e.p.s. of all the speech aligned to a state in the database was (probably) an average over many values of F_0. It might therefore have been more appropriate to use the s.t.e.p.s. that each state occurrence would have had if synthesised at an average F_0, in place of the simple occurrence s.t.e.p.s. The energy scaling factor calculated during synthesis would then have resulted in a segment being synthesised at its state's average s.t.e.p.s. if it were synthesised at an average F_0, which would have been more consistent. However, this idea was not pursued in the current work.

Stage (iii): Select the occurrence still under consideration which has the highest average log-likelihood per frame in the state alignments.

This ensured that the segment selected was that most likely to be observed using the state Gaussian. (Note that this was not necessarily the one most representative of the segment distribution defined by the state occurrences.) This was probably beneficial, since it was likely to prevent very unrepresentative segments from being selected. However, it was not necessarily the best segment to select. As suggested above, concatenation continuity considerations probably also have an important role to play in segment selection. These considerations were not investigated in the current work, but are discussed as possible future work in Section 9.1.3.

7.2 Pitch-Mark Identification

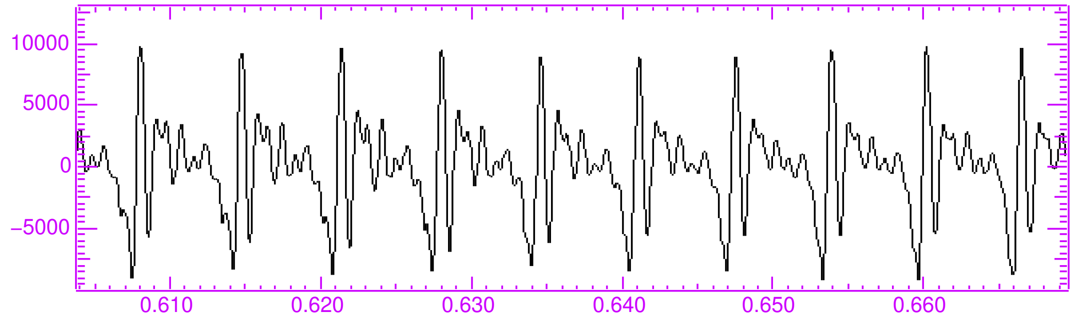
As discussed in Section 7.3.1, the TD-PSOLA implementation used in this work used short Hanning windows, of twice the synthesis pitch period in duration. For good performance these windows must be centred on the moments of principle excitation of the vocal tract. When this relationship is not maintained the quality of the synthetic speech changes and eventually becomes hoarse when the windows are misplaced by more than 30% of a pitch period (Moulines and Charpentier 1990). The determination of the moments of principle excitation, to produce what are termed the *analysis pitch-marks*, was therefore an important part of the TD-PSOLA implementation. Two methods were used to perform this determination in the current work, and these are discussed in Sections 7.2.1 and 7.2.2.

7.2.1 LP Residual Based Methods

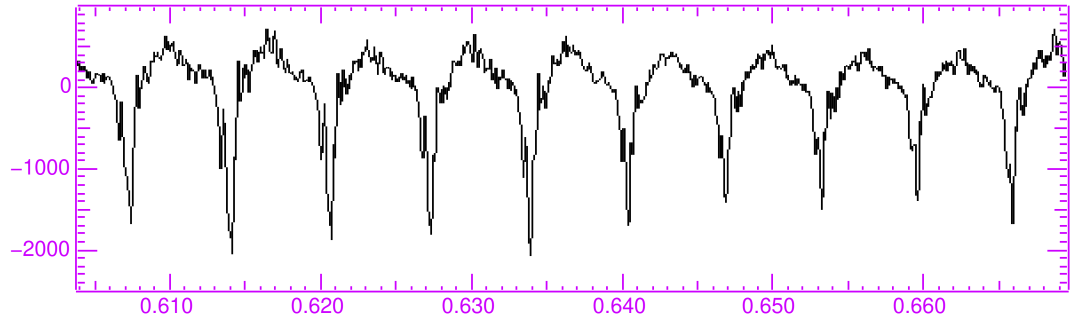
As described in Section 1.4.5, and more thoroughly in Appendix D, linear prediction (LP) analysis assumes that each sample $s(n)$ of a speech frame can be calculated as a weighted sum of the previous P samples, plus a small error term, $e(n)$. The structure of the error signal, $e(n)$, termed the *LP residual*, gives information about the predictability of the speech signal at each point in time. It is usually found, during voiced speech, that the LP residual is small for most of each pitch period, with one concentrated burst of activity corresponding to the moment at which the speech signal begins what is, to the eye, clearly the start of another pitch period. As an example, some speech and its LP residual are shown in Figure 7.1.

One method of locating the pitch-marks of voiced segments is therefore based on analysing their LP residuals. However, the structure of the LP residual is often considerably more complex than that in Figure 7.1, especially with fricatives and voiced fricatives (see Figure 7.2), and bursts. Although pitch-marks can often be easily identified by eye, performing the analysis automatically is non-trivial. An automatic procedure must ensure that only one peak is chosen within each pitch period, that pitch periods are deemed to have realistic separations, and that pitch periods are not assigned to regions of unvoiced speech. In short, the problem is to implement an LP residual based pitch tracking algorithm. Many of the problems involved are therefore problems common to all pitch tracking algorithms, into which a great deal of research has been directed in the past, (Hess 1983), but to which perfect solutions have not yet been found.

The method used in the current work was to analyse the LP residual of each sentence in the database using a computer program called *epochs*, which is commercially available



(a) Speech signal



(b) LP residual

Figure 7.1: Part of the speech and LP residual of the phone /aa/, taken from the word "far" in the M2 database.

from Entropic Research Laboratory, Inc. The program implements an algorithm similar to that described in (Secrest and Doddington 1983). It uses dynamic programming to find the most likely path through a sequence of possible candidates, deemed to be those samples of the residual above a certain fraction of the local r.m.s. value. The optimisation is carried out with respect to a set of rewards and costs for desirable and undesirable behaviour. For example, there is a heavy cost associated with frequency doubling or halving, the false detection of which is a problem often suffered by pitch tracking algorithms.

The algorithm successfully found most of the true pitch-marks for the speech of the M1 database, but, using the default cost settings, also assigned pitch-marks through many regions of unvoiced speech. This problem also occurred with the female speech of the F1 database. For this speech many choices of the cost settings were tried, and although the situation was improved, no choice of settings could be found which solved the problem completely. The problem could be largely overcome by incorporating a voicing determination algorithm (VDA) into the procedure, although this was not done in the current work. In fact, such a solution has been demonstrated, (Talkin 1995), although difficulties were still experienced with regions of creak (vocal fry), where the VDA, which was based on periodicity, made errors.

7.2.2 Laryngograph Based Methods

During voiced speech the vocal tract is excited by the vocal cords, with the moment of principle excitation usually corresponding to the moment of glottal closure. A *laryngo-*

graph enables vocal cord activity to be measured more directly than by observing its effect on the speech signal. Two electrodes are placed on the outside of the throat either side of the windpipe, held in place by a velcro strap around the neck. The impedance between the electrodes is measured, which gives information about the status of the vocal cords. When the vocal cords are open, the impedance is high, and vice-versa. The differential of the laryngograph signal therefore reflects the rate of change in the status of the vocal cords, and the maxima of this signal therefore correspond to the moments of most glottal excitation. An example of some speech, its laryngograph signal and corresponding differential are shown in Figure 7.2, along with the LP residual of the speech for comparison. The advantages of the laryngograph (differential) approach over the LP residual approach are that the pitch-mark candidates are usually much more prominent, and that voicing determination is usually trivial.

Laryngograph signals were recorded as part of the M2, M3 and F2 databases, and with some test sentences recorded by the female speaker used in the F1 database. The signals were processed by first differentiating them, and then removing all samples smaller in amplitude than 1.6-2.0 times the r.m.s. value of the signal over the sentence. This was done to remove noise from the signal, to reduce the number of candidate peaks passed on to the next stage of processing. The exact value of the threshold was found to be important only for the female speech of the F2 database, where the glottal signal was sometimes quite weak, and the signal to noise ratio in the laryngograph differential therefore sometimes quite small. The *epochs* program was then run using the remaining samples as the pitch-mark candidates, to find a set of pitch-marks at reasonable spacings. The result was a very accurate determination of pitch-marks with speech from the speakers used in the M2, M3 and F1 databases, and respectable results with speech from the F2 database. Errors typically numbered a handful per sentence for speech from the first three speakers, but rather more for speech from the F2 database. With the former, the errors were usually associated with the first, weak, pitch pulse of voicing onsets, though with the latter they were distributed more widely.

The pitch-mark times obtained using this method had to be corrected for the delay between the laryngograph signal measured at the vocal cords, and the speech signal measured at the microphone. This delay was due to the sound propagation time from the vocal cords to the microphone. The use of a head mounted microphone meant that this distance was constant, and therefore the delay was also approximately constant for a particular speaker. The delay could therefore be accounted for by adding a constant correction term to the pitch-mark times. The delay was measured as varying between $560\mu s$ and $690\mu s$ for the male speaker used to record the M2 database, and between $500\mu s$ and $560\mu s$ for the female speaker used to record the F1 database. The shorter delay for the female speech was expected since women have shorter vocal tracts on average than men. These delays were significant enough to require correction, since they represented as much as 8% and 12% of a pitch period at the speakers' average fundamental frequencies. At higher frequencies, the percentage error could therefore have come close to the 30% level at which (Moulines and Charpentier 1990) report that the performance of the PSOLA algorithm becomes seriously degraded. Since the corrections necessary were small and fairly similar, a single correction of $562.5\mu s$ (9 samples at 16kHz) was used in all cases.

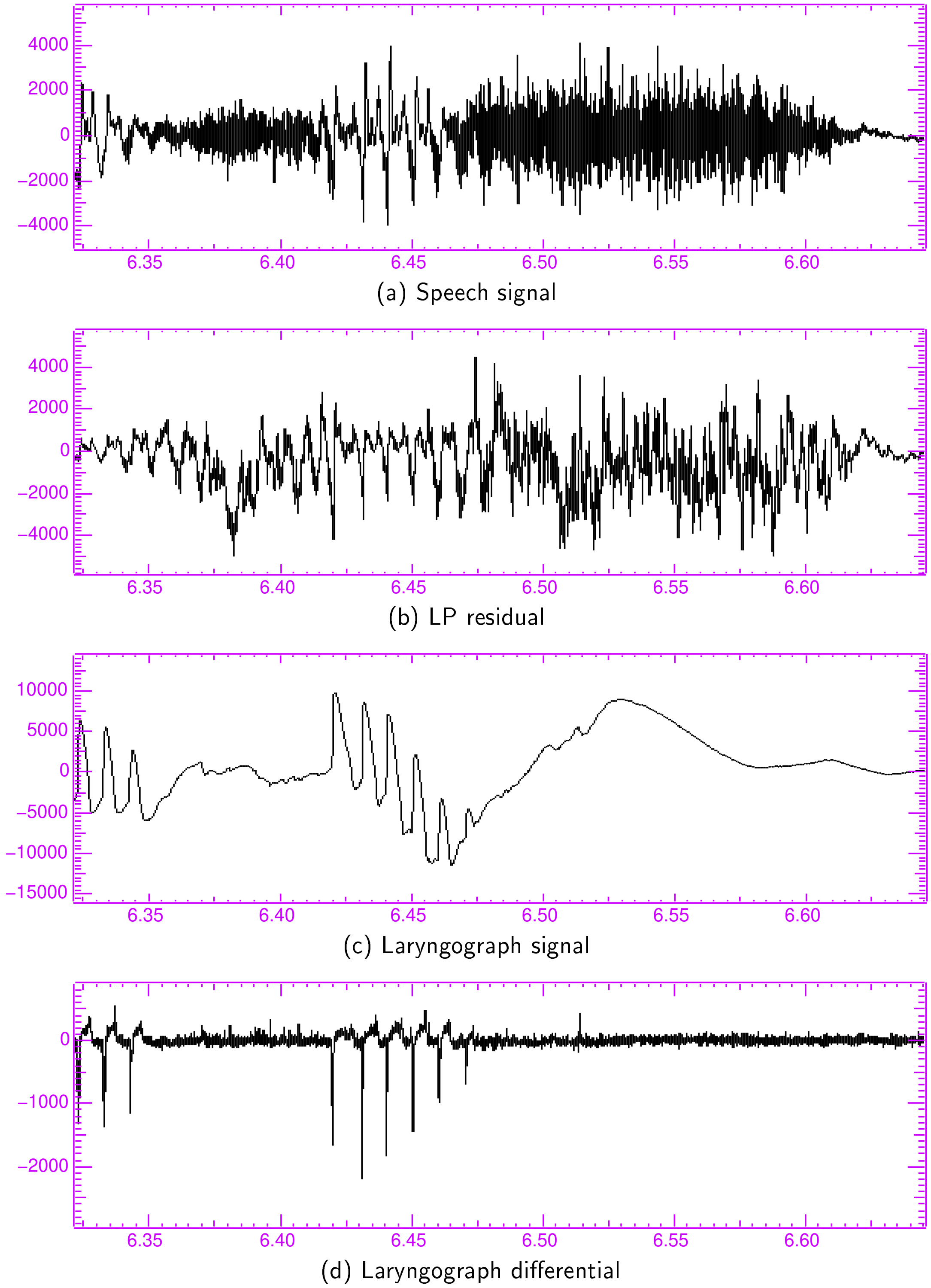


Figure 7.2: The speech, LP residual, laryngograph, and laryngograph differential signals of the phone sequence /ay z ax s/ from the words "lies a small" in the M2 database.

7.3 TD-PSOLA Implementation

This section describes the details of the implementation of the TD-PSOLA algorithm. The results of experiments to re-synthesise natural utterances with different fundamental frequencies and durations are then presented in order to demonstrate its performance.

7.3.1 TD-PSOLA Implementation Details

As explained in Section 1.4.6, TD-PSOLA works by breaking the speech segments to be concatenated into short-term (ST) signals and then recombining these signals during synthesis to obtain the required F_0 and duration in the synthetic speech. The following discussion initially assumes that the speech to be processed is all voiced, leaving the complications introduced by unvoiced speech and segments with multiple voicing transitions until later.

At the heart of the PSOLA algorithm is a mapping between the analysis pitch-marks in the original segments and the synthesis pitch-marks in the synthetic segments. In this system, the original segments were those selected by the algorithm described in Section 7.1 to represent each clustered state, and the analysis pitch-marks were located using one of the methods described in Section 7.2. The synthetic segments are the desired reproductions of the original segments in the synthetic speech, as defined by the duration and F_0 generation components of the ASS system concerned. In this system, synthetic segment durations were specified using the equation described in Section 5.3.1, and F_0 was set to the average for the speaker concerned.

Figure 7.3 shows the pitch-mark mapping between an original segment, with an F_0 which rises from left to right, to a synthetic segment of longer duration, with an F_0 which falls from left to right. The ratio of the durations of the synthetic and original segments is used to warp each synthesis pitch-mark time into a corresponding analysis time, as shown by the dotted lines. The analysis pitch-mark closest to each warped synthesis pitch-mark time is then chosen as the analysis pitch-mark associated with that synthesis pitch-mark, as shown by the dashed lines. An ST-signal is obtained for each analysis pitch-mark by multiplying the original speech segment by a Hanning window centred on that pitch-mark. The synthetic segment is then constructed by adding together all the ST-signals associated, via the mapping, with each of the synthesis pitch-marks, centring each ST-signal on the appropriate synthesis pitch-mark. In Figure 7.3 this procedure results in the repetition of some of the ST-signals in the synthetic speech. If the synthetic segment had been shorter, or had a lower F_0, then it is likely that some of the ST-signals would have been deleted from the synthetic speech.

The effects of Hanning window size on TD-PSOLA synthesis were discussed in Section 1.4.6. In this work short windows (twice the synthetic pitch period) were used, in order to reduce the effect of mis-matches between the synthesis F_0 and the inherent F_0 of each ST-signal. This was possible because the moments of principle excitation of the vocal tract were known from the methods of Section 7.2. The resulting formant broadening was not a major concern, since (Moulines and Charpentier 1990) had shown it to have very little audible effect.

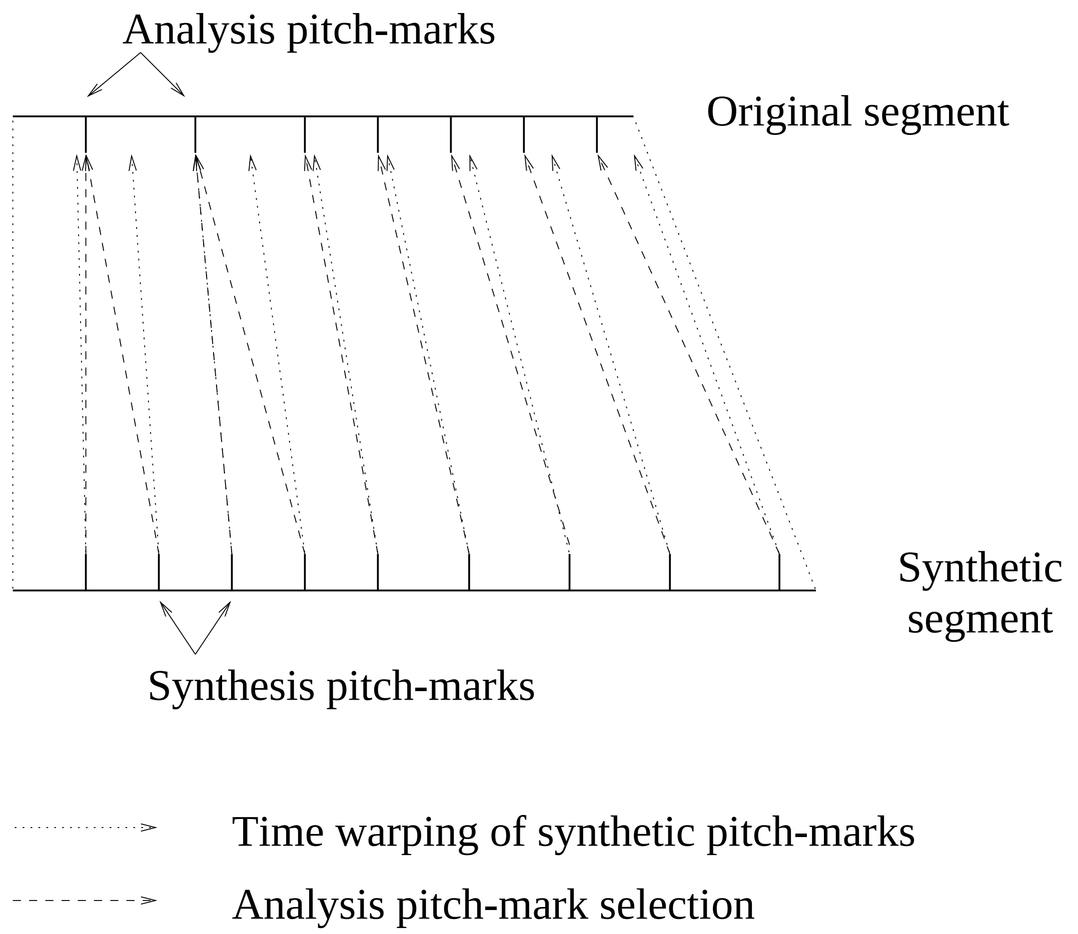


Figure 7.3: Original to synthetic segment pitch-mark mapping in the PSOLA algorithm.

As described in Section 1.4.6, a number of adding schemes are available to re-combine the ST-signals to form the synthetic speech. In this work the ST-signals were simply added together at their new spacings. This scheme accounted for the number and weight of the various contributions to the synthetic speech signal at each point in time because the Hanning window sizes used were always twice the synthetic pitch period. An energy scaling factor was applied equal to the ratio of the average s.t.e.p.s. of the state concerned to that of the *original segment*. Energy therefore scaled with F_0 in the synthetic speech, which was desirable (see Section 1.9). However, as discussed in Section 7.1, it would have been more consistent to use a scaling factor equal to the ratio of the average s.t.e.p.s. of the state to that of the *synthetic segment* synthesised at an average F_0.

The discussion so far has assumed that speech is all voiced, which of course it is not. Although the F_0 modification capability of the PSOLA algorithm is not required for unvoiced speech, the algorithm is still used in order that durations can be modified during synthesis. The algorithm requires that some form of "pitch-marks" are specified for the regions of unvoiced speech to be processed. These pitch-marks (*U-marks*) have traditionally been uniformly spaced through regions of unvoiced speech, although this does not necessarily have to be the case. There may be some grounds for spacing U-marks at the pitch period of any adjacent voiced speech, both in the original and synthetic segments. Such a scheme would be likely to correct small errors made in determining the exact moment of voicing transition. However, such a procedure would also quantise unvoiced sounds on the time-scale of pitch periods, which for low-pitched voices could result in unwanted side effects, such as, for example, repeating the release of a burst ten milliseconds or more after it was first released. For this reason a small (4ms) uniform U-mark spacing was used in this work.

Unvoiced speech was identified as any length of speech which had no voicing pitch-marks (*V-marks*) assigned to it for 20ms or longer. This figure was a compromise between trying to select a duration longer than the largest spacing likely to occur between genuine V-marks, but as short as the shortest likely duration of genuine unvoiced speech. Using this figure, inappropriate assignment of U-marks could therefore occur with speech with an F_0 below 50Hz, or in regions of low-pitched creak. Although creak could occur, the result of it being marked as a voiced-unvoiced-voiced transition in the original segment would be to produce creak in the synthetic segment, which was not necessarily a bad thing. The occurrence of speech with an F_0 below 50Hz was thought to be very unlikely for the speakers used in this work. However, for very low-pitched speakers a longer threshold, or a different method, might be necessary.

Given that both voiced and unvoiced pitch-marks could occur in the same original speech segment, some mechanism was required to determine which sections of the synthesis segment should be voiced, and which unvoiced. The method used is shown in Figure 7.4. The method involved the introduction of voicing transition markers, called *T-marks*, which were placed equidistant between the last pitch-mark of one voicing type, and the first pitch-mark of the following voicing type. During synthesis, these marks were used as follows. Firstly, the overall duration ratio of the synthetic and original segments was found as before. This was then used to map the times of the T-marks in the original segment to times in the synthetic segment, to produce the synthesis T-marks. The synthesis pitch-marks were then generated using the appropriate pitch periods, calculated from the desired pitch track, for voiced regions, and at 4ms intervals in unvoiced regions, with the voicing regions defined by the synthesis T-marks.

The quantisation of the original speech segments into ST-signals meant that they would very rarely be synthesised for exactly their required durations if proper pitch-mark separations were to be maintained. Therefore, in order that the F_0 of synthetic utterances could be computed in advance with reference to the required segment durations, it was necessary to introduce a mechanism by which the synthetic speech could stay in step with these durations during synthesis.

The mechanism introduced involved two time-scales, *ideal time* and *actual time*. Ideal time was that computed from the desired segment durations, and the mapping of original segment T-marks into synthesis time. It formed the framework into which the pitch-marks were placed, the location of which was computed using actual-time. The process worked as follows. The voicing type of the first pitch-mark in a segment was determined. Synthesis pitch-marks of the appropriate voicing type were written out in a left to right fashion, until the next pitch-mark would occur beyond the next T-mark time, computed as the segment start time plus the synthesis T-mark offset into the synthetic segment. The voicing type

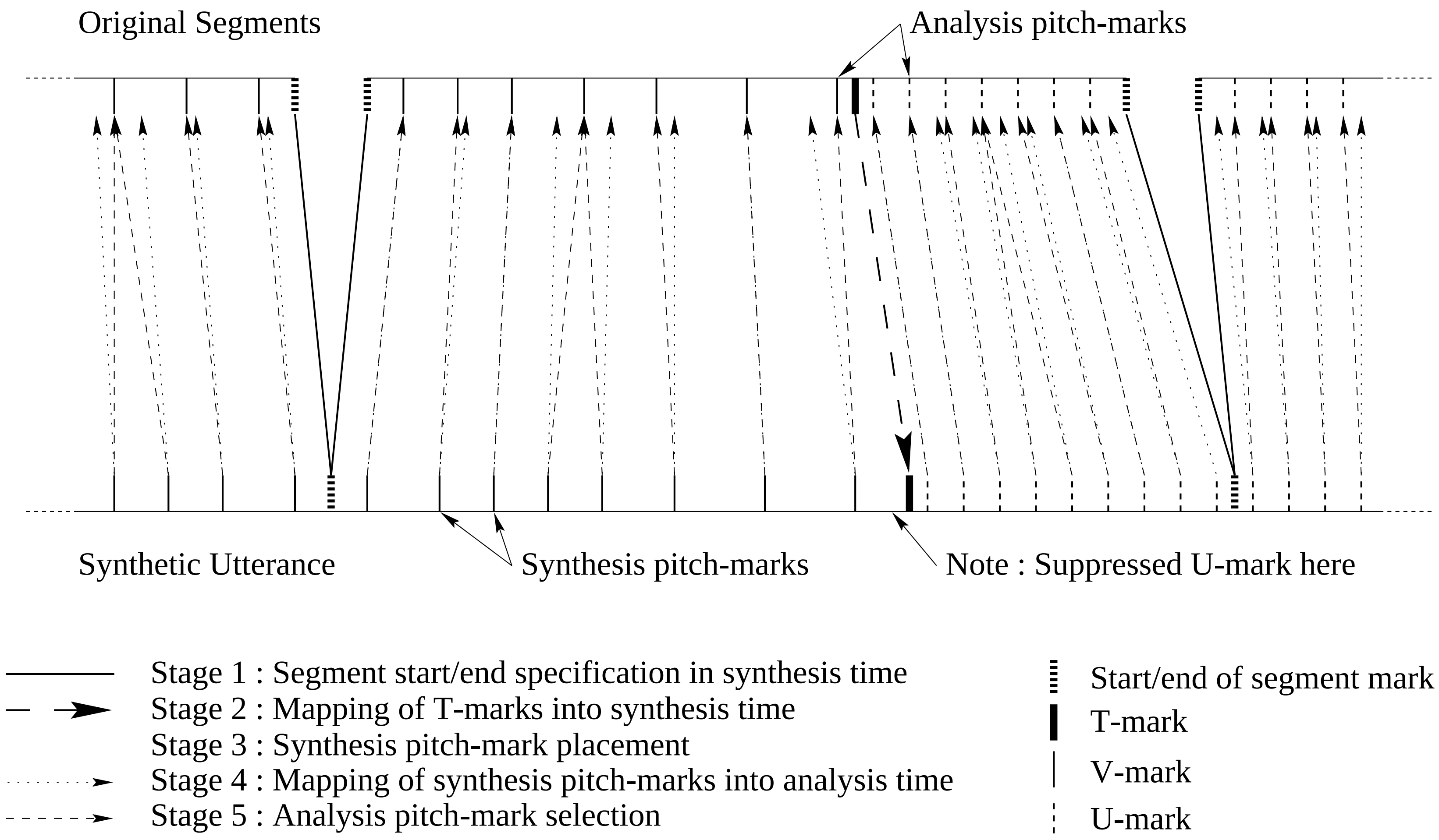


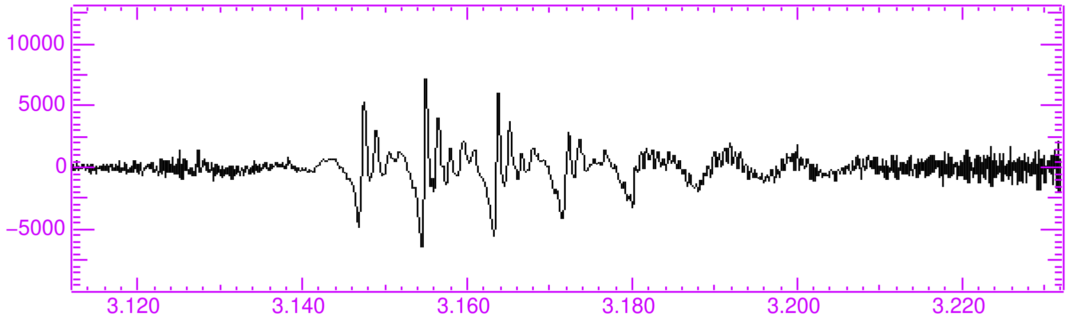
Figure 7.4: The detailed structure of the TD-PSOLA implementation.

of the pitch-marks being written was then changed, and pitch-marks written out at times continuing on from the actual time reached with the previous voicing. Importantly, no pitch-marks in the new voicing type were actually written until their actual time had exceeded the T-mark time. This was to prevent the end of a low-pitched voiced section, into which another V-mark would not fit, being filled with multiple U-marks, which would then be mapped to V-marks in the original segment, producing an artifact in the synthetic speech. This method continued left to right until the time of the next pitch-mark to be written would exceed the ideal segment end time, computed as the segment start time plus the desired segment duration. The next segment would then be begun, with the pitch-mark times again carrying on from the actual time reached in the previous segment. Again, no pitch-marks were actually written until their actual time exceeded the new ideal segment start time, because this could result in multiple synthesis pitch-marks being mapped to the first analysis pitch-mark of a segment, again causing an artifact in the synthetic speech. This process is also shown in Figure 7.4.

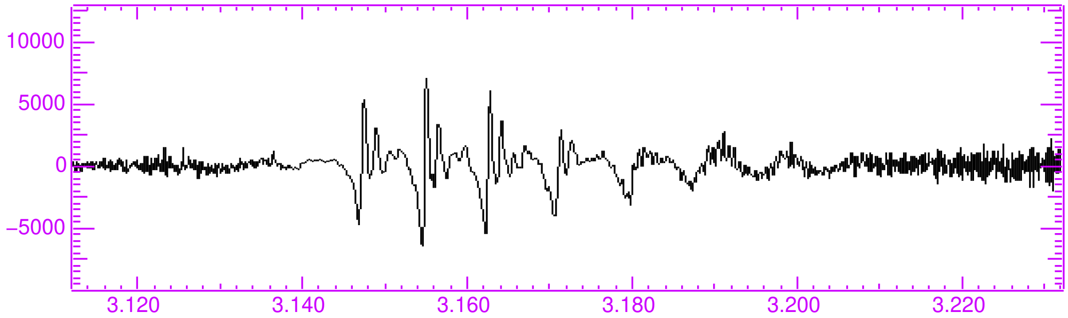
7.3.2 Implementation Demonstration

The TD-PSOLA implementation was intended to be used to concatenate segments of speech from different parts of a database. However, by treating an entire utterance as a single segment, containing many voicing transitions, it was possible to use the implementation to perform analysis-synthesis. This was useful in demonstrating that the algorithm worked properly, and in determining the degree of degradation introduced by F_0 and duration modifications alone. This speech represented the best that the concatenation system could hope to achieve.

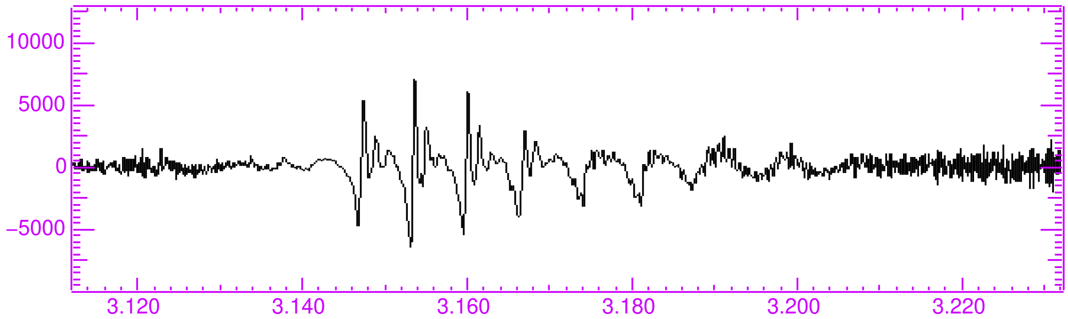
Figure 7.5 shows the waveforms of an original speech segment from the M2 database, and the results of performing TD-PSOLA analysis-synthesis on it. The segment shown corresponds to the phone sequence /f ae sh/, of the word "unfashionable", taken from the first sentence of the training data. The speech from which the waveforms were taken, plus other similar examples, is available as examples 8-19 on the accompanying compact disc (see Appendix E). Figure 7.5(a) shows the original speech waveform, taken from the M2 database. Figure 7.5(b) shows the waveform of the speech after analysis-synthesis, with no change in F_0 or duration. As can be seen, the waveform looks extremely similar to the original, and in fact the speech is audibly indistinguishable from it. Figures 7.5(c)-(f) show the results of re-synthesising with various F_0 modification factors. By listening to the speech examples, it can be heard that raising F_0 is generally more successful than lowering it. This is because the duration of the Hanning windows used was set to twice the local synthesis pitch period, and therefore when lowering F_0 the windows began to have a significant inherent F_0 of their own. This is most clearly heard with the speech shown in Figure 7.5(f), where F_0 was halved during re-synthesis. In this case the inherent F_0s of the ST-signals have combined to recreate the original pitch of the speech, which can both be heard in the speech, and seen in Figure 7.5(f). This indicates that perhaps a procedure which ensured that ST-signals were never larger than two analysis pitch periods would have been useful. Nevertheless, the implementation does work very well for F_0 changes of up to a factor of 1.2, and reasonably well up to factors of 1.5, certainly for F_0 rises.



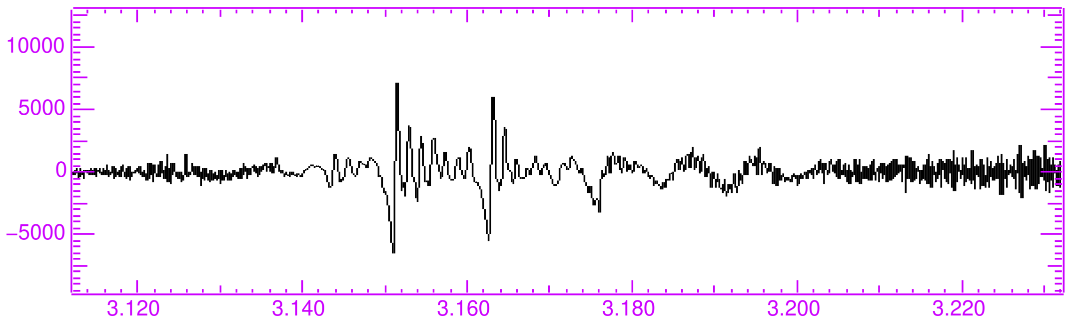
(a) Original speech



(b) TD-PSOLA synthesis at original F_0 and duration

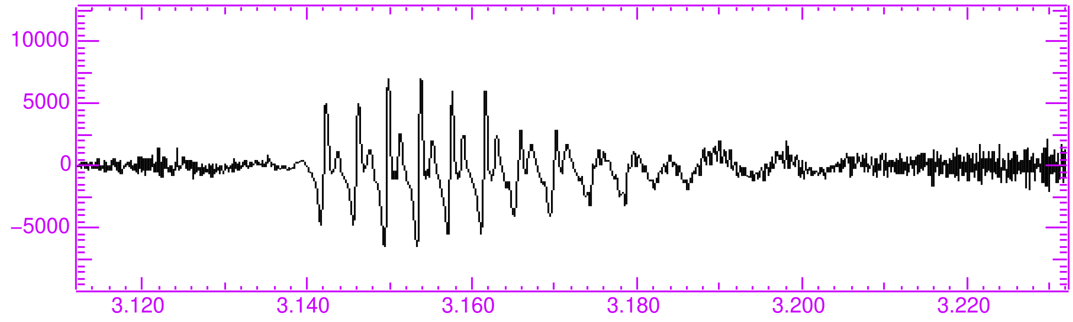


(c) TD-PSOLA synthesis, F_0 scaled up by 1.2, original duration

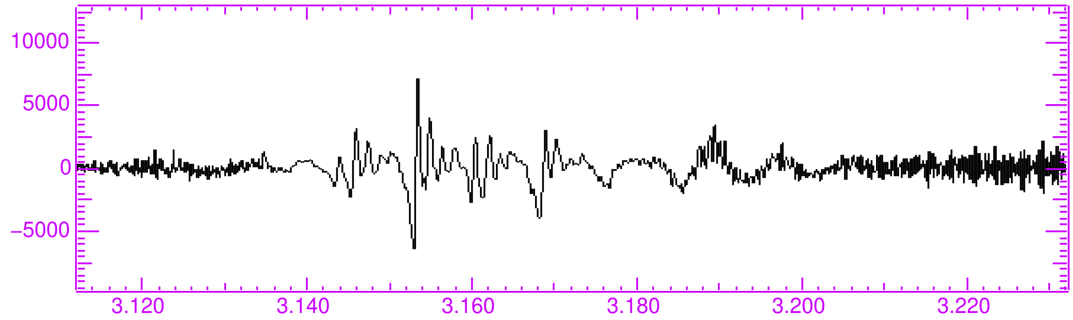


(d) TD-PSOLA synthesis, F_0 scaled down by 1.5, original duration

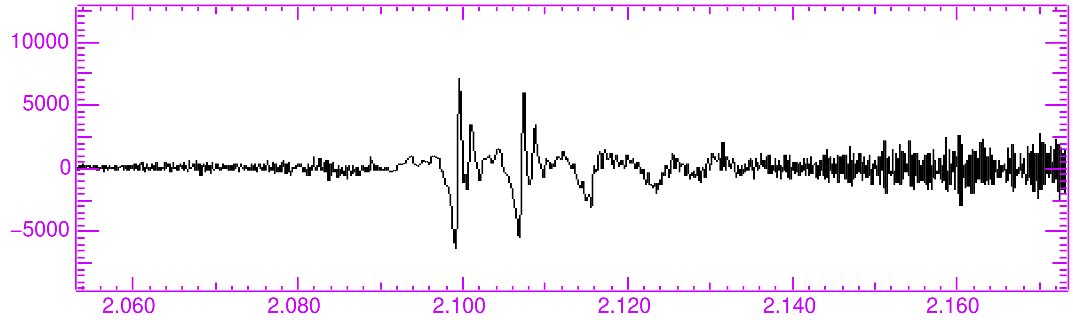
Figure 7.5: TD-PSOLA analysis-synthesis of the speech of the phone sequence /f ae sh/ from the word "fashionable" from the first sentence of the M2 database.



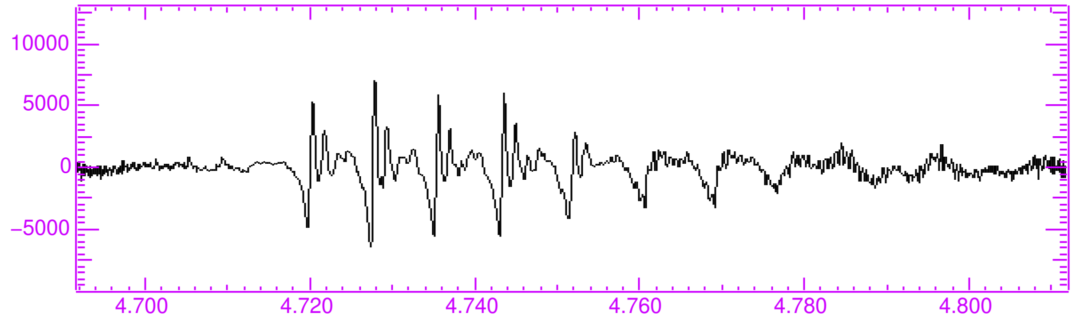
(e) TD-PSOLA synthesis, F_0 scaled up by a factor 2.0, original duration



(f) TD-PSOLA synthesis, F_0 scaled down by a factor 2.0, original duration



(g) Duration scaled down by a factor 1.5, original F_0



(h) Duration scaled up by a factor 1.5, original F_0

Figure 7.5 (continued): TD-PSOLA analysis-synthesis of the speech of the phone sequence /f ae sh/ from the word "fashionable" from the first sentence of the M2 database.

Figures 7.5(g) & (h) show the result of using TD-PSOLA analysis-synthesis to lengthen or shorten the duration of the original speech by factors of 1.5 whilst maintaining the original F_0. The speech can be heard to be of a high quality with a slight degradation audible in the lengthened speech. More extreme examples, with duration factors of 2.0, are also available on the compact disc, and here the lengthened speech can be heard to be considerably inferior. The degradation is due to the artificial periodicity introduced into the unvoiced regions of the synthetic speech, and its effect is to make the speech sound as if it were being heard through a large plastic tube. However, this degradation was expected, and such large increases in duration were not likely to be required by the synthesis system.

Speech examples 20-31 similarly demonstrate the performance with the same sentence taken from the F2 database (see Appendix E). Note that the hoarseness heard in the word "uncharted" is due to the non-identification of a single V-mark.

In summary, the TD-PSOLA implementation has been shown to work very well for F_0 changes of factors of 1.2 or less, and for duration compression up to a factor of at least 2.0. It also works reasonably well up to a factor of 2.0 when raising F_0 up to a factor of about 1.5 when lowering F_0 and up to a factor of about 1.5 when lengthening durations. An analysis was conducted to determine how much prosodic alteration of the original waveform segments was required during synthesis, when the system was trained on the M2 database. The results of the analysis, and an assessment of the implementation's ability to perform the required alterations, is presented in Section 8.5.2.

Chapter 8

Results, Analysis, & Discussion

This chapter begins with a discussion of the effects of the modifications to the basic system introduced in Chapters 6 and Chapters 7. It goes on to present an analysis of the results of the tree clustering and duration estimation procedures used in the system (Sections 8.2 & 8.3). It then presents examples of the synthetic speech from the final LP version of the system (Section 8.4). The results of incorporating the TD-PSOLA synthesis technique are then presented in detail (Section 8.5), before a discussion of the strengths and weaknesses of the PSOLA version of the final system (Section 8.6).

8.1 Analysis of Synthesis Improvements

The work described in Chapters 6 and 7 led to significant improvements in the quality of the synthetic speech produced by the system. Modified Rhyme Tests (see Chapter 4) were conducted periodically to monitor the changes introduced to the system. These tests were used primarily to direct research effort towards the aspects of the system which were most in need of improvement. The results also enabled some assessment to be made of how much improvement resulted from each modification (or group of modifications). A summary of the results is presented in Table 8.1.

As can be seen from Table 8.1, the largest relative improvements in the MRT scores came from the introduction of variable frame size and rate coding, the transition to a TD-PSOLA system, and the introduction of word final/initial clustering and/or the coding alterations of Section 6.1.8. It is likely that the splitting of plosives into separate closure and burst models and the improved silence transcriptions contributed in part to the size of the gain achieved by the use of variable rate coding. The large improvement resulting from the introduction of the early TD-PSOLA scheme was due to the improved synthesis quality obtained, particularly with bursts where the bipolar voicing decision used in the LP system was inadequate. The final large improvement was probably mostly due to the coding alterations, rather than the introduction of word final/initial clustering. It shows the importance of accurate transcription and segmentation in the final system. However, the size of this improvement should probably be viewed with some caution, since the number of error words was very small by this stage.

The stability of the MRT error rate following the implementation of the full TD-PSOLA system and the introduction of syllabic clustering is perhaps surprising. As can be heard

MRT Error Rate (%)	System/Modifications
33.0	The P-method version of the Basic System.
28.7	After splitting plosives into separate closures and two-state burst models, removing silences aligned in the initial or final closures of words, and moving to a single (30ms minimum duration) silence model.
14.7	After introducing variable frame size and rate coding.
11.7	After the methods of Section 6.1.4 were applied to remove suspiciously short closures.
9.7	After the voicing decision made during synthesis was improved (see Section 8.4), and the bursts of voiced plosives made into one state models.
11.7	After introducing stressed vowel clustering, and raising the minimum number of frames per state clustering threshold in an attempt to solve the problems discussed in Section 6.2.1.
4.7	After the change to an early TD-PSOLA system (see Section 8.5.1).
4.7	After the implementation of the full TD-PSOLA system, the laryngograph methods for locating pitch marks, and the syllabic clustering described in Section 6.2.3.
2.7	After the introduction of word final/initial clustering, and the transcription and segmentation improvements resulting from the coding alterations of Section 6.1.8

Table 8.1: Modified Rhyme Test scores during system development.

by listening to the synthetic speech presented in Sections 8.5.1 and 8.5.3, these changes did lead to considerable improvements in the overall quality of the synthetic speech produced by the system. However, these improvements were mostly with the detailed quality of voiced regions of speech, which were generally already very intelligible, and so they did not affect the MRT scores. Finally, the earlier rise in error rate from 9.7% to 11.7% was probably caused by the change in the minimum number of frames per state clustering threshold, which resulted in a 15% drop in the number of clustered states in the system.

As discussed in Section 4.2, all the tests in Table 8.1 (except for that with the basic system) were conducted using listeners who had performed the tests before. The listeners undoubtably adapted to the synthetic speech produced by the system over the course of the tests (see Section 8.5.4). The figures in Table 8.1 therefore do not represent absolute performance, and some allowance for adaptation should probably be made when interpreting them.

8.2 Tree Analysis

An analysis of which questions were the most important during the tree-clustering process is given in Table 8.2. The first question, and the first five questions, asked during the

First question		First five questions	
% of questions asked	question	% of questions asked	question
11.6	R_Vowel	5.3	R_Vowel
9.1	R_Nasal	4.3	R_sp
6.6	L_Vowel	3.3	R_Nasal
5.8	R_sp	3.1	L_Nasal
5.8	R_Unrounded	2.9	L_UnFortLenis
5.0	L_UnFortLenis	2.4	R_Unrounded
5.0	L_Strident	2.4	R_High
2.5	R_UnFortLenis	2.4	L_Vowel
2.5	R_High	2.4	L_Unrounded
2.5	L_Voiced-cons	2.1	L_Front

Table 8.2: The ten most frequently occurring questions in the sets formed by pooling the first questions and the first five questions asked in each tree, using the final system and the M2 database.

construction of each tree were pooled to form two sets. The ten most frequently occurring questions in each set are listed in Table 8.2.

Although not appearing in Table 8.2, questions referring to both stress level, and contexts with specific within-word positions did occur as the first question asked during the construction of some trees. A question about the syllabic nature of /l/ was asked as the first question of the tree built for the leftmost /l/ state. Interestingly, the first question asked in several trees was about a left or right context of one particular phone, rather than about a broader class context. This was sometimes due to a lack of training data for the phone concerned, but this was not always the case. It meant that the trees lost generality very quickly, but it was not clear if it was actually detrimental to performance.

8.3 Duration Analysis

The durations used by the system during synthesis were essentially obtained as a by-product of the acoustic clustering. However, they did lead to remarkably natural sounding synthetic speech. It would be useful to measure the accuracy of the durations predicted by the system, relative to some standard. In the literature human segmentations have usually been used as this standard, although as discussed in Section 2.2.1 such segmentations are in themselves somewhat subjective. Unfortunately, no human segmentations of the databases used in this research, nor even of any sentences recorded by the speakers used, existed to provide this reference. The accuracy of the duration model used by the system was therefore assessed relative to the segmental durations derived automatically during system construction.

If the duration model was perfect, accounting for every possible form of context including speaking rate, then the r.m.s. deviation of the database durations predicted by the model from the database durations obtained using the HMMs (upon which it was trained) would be zero. Therefore, the size of the r.m.s. deviation is an indication of the quality of the duration model. The use of this figure was not ideal, since it is generally preferable to use new data as test data to prevent an over-fitted model being judged as superior. However, the result nevertheless gives some indication of the performance level reached. The r.m.s. deviation between the HMM derived state durations and those predicted by the duration model was 14.8ms. This was calculated over all segments except silences and dummy /cl/s, using the results of the final system trained on the M2 database. Since most phones were composed of three states, this represented an r.m.s. deviation of 25.6ms per three-state phone, assuming that the deviations for each state were independent random variables. This is slightly worse than Riley's result of 23ms per phone discussed in Section 1.9.2. Furthermore, Riley used cross-validation techniques to prevent over-fitting, which was not done in the present case. However, it must be remembered that the present results were produced as a by-product of the acoustic clustering, and not via duration clustering. The possible future use of duration clustering is discussed in Section 9.1.2.

The duration alteration algorithm described in Section 5.3.1 was designed to alter the most variable state durations the most, and the least variable the least. Some analysis was conducted to determine whether any phonetic patterns could be seen amongst states with different variabilities. The results of the analysis are presented in terms of *percentage deviation figures*, which were calculated using the results of the final system trained on the M2 database, as follows. For each state, the r.m.s. deviation between the HMM derived state durations and those predicted by the duration model was expressed as a percentage of that state's mean duration. The average of these percentages was then calculated both globally (again excluding silences and dummy /cl/s), and for each individual phone, weighting each state's figure by the number of occurrences of that state in the database. The global percentage deviation figure was 44%, and with the exception of /ua/ (for which there were very few occurrences), the dummy /cl/, and silence, the phone-based percentage deviation figures varied from 23% to 63%. As expected, bursts in general had smaller percentage deviations than vowels, nasals, and liquids, although there was some overlap. The different fricatives spanned most of the range of values, and closures were in general very variable.

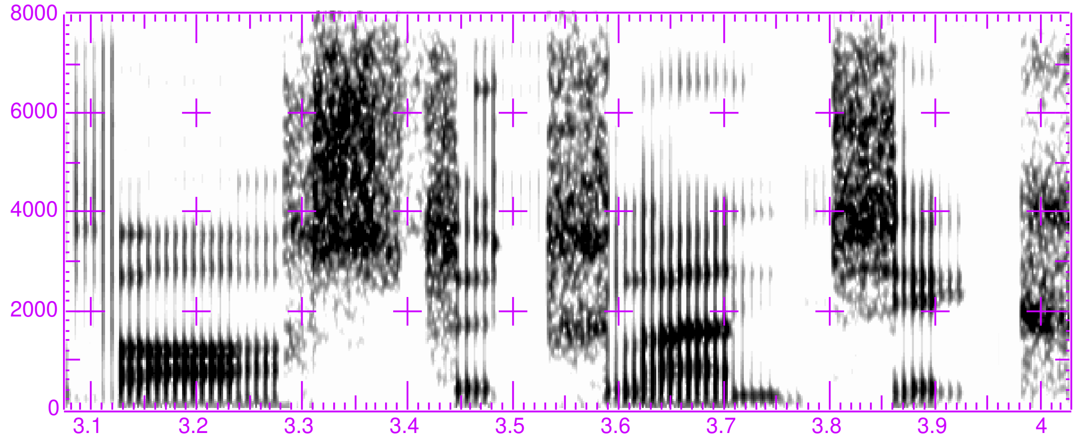
Recalling the discussion in Section 5.3.1, the above analysis shows one of two things. Either the duration clustering resulting from the acoustic clustering was better for bursts than for vowels, nasals, and liquids, or, bursts in a given context simply have less variable durations than other phones at different speaking rates. Both of these may be true; however, from an articulatory point of view, the latter seems unlikely to be unimportant, and therefore the duration alteration procedure used has at least some justification.

8.4 LP Synthesis Results

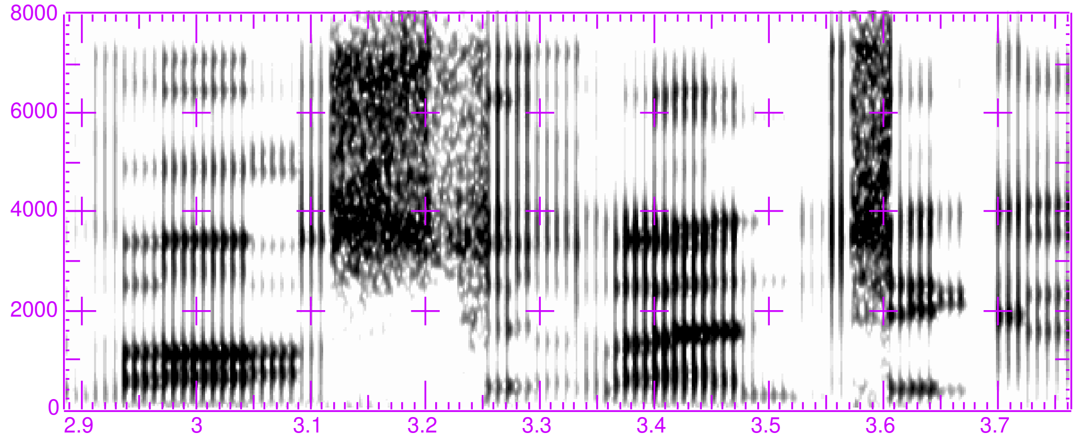
Figure 8.1(a) shows a wideband spectrogram of the sentence fragment "...vast Atlantic..." produced from speech example 32 on the accompanying compact disc. This speech was generated using the final system trained on the M2 database, incorporating all of the improvements described in Chapter 6, but using an LP synthesiser with the LP coefficients estimated using the P-method. The algorithm used to determine the voicing of each segment during synthesis was improved over that used in the basic system. The new algorithm forced the states of the phones /tbst, kbst, jhbst, chbst, pbst, hh, th, s, sh, f/ to be produced unvoiced, and the states of the phones /aa, ae, ah, ao, aw, ax, ay, ea, eh, er, ey, ia, ih, iy, oh, ow, oy, ua, uh, uw/ to be produced voiced, using the zero crossing rate to determine the voicing of states of other phones as before. In order to unambiguously demonstrate the large improvement in the system's ability to model closures and bursts, in this example each closure state was produced at its average s.t.e.p.s. figure. Usually however, closures were produced as periods of silence, since this removed any residual buzz or noise, and improved the quality of the speech. Figure 8.1(b) is a repeat of Figure 5.2(d), i.e. a spectrogram of the same sentence fragment produced by the P-method version of the top ranked configuration of the basic system listed in Table 5.1 trained on the M1 database. It is repeated to enable a direct comparison to be made with that produced from the speech synthesised by the final system. Finally, Figure 8.1(c) is a repeat of Figure 5.2(f), and was produced from a natural version of the utterance spoken by the speaker used to record both the M1 and M2 databases.

As can be seen from the figure, the improvements of Chapter 6 substantially improved the system's ability to synthesise closures and bursts. For example, those in the /t/ of "vast" and the first /t/ of "Atlantic" are much more distinct in the spectrogram obtained from speech synthesised using the final system than they are in that produced from the speech of the basic system. These improvements resulted in the speech sounding much more precisely articulated than that produced with the basic system. Indeed, the final system, which releases all bursts during synthesis, can sound hyper-articulated at times. Another improvement which can be seen from the figure is that the speech produced from the final system did not have as much excess energy present at high frequencies as that produced from the basic system. A possible explanation for this phenomenon is that the superior clustering used in the final system resulted in clustered states comprised of more self-similar speech, which therefore had more accurate LP coefficients. Small changes were present in the quality of some vowels, but this was probably due to improvements made to the pronunciation dictionary, rather than any improvements made to the synthesis system. The synthesis of some fricative to voiced speech transitions was also improved in the speech from the final system, but otherwise the synthetic speech was quite similar to that of the basic system.

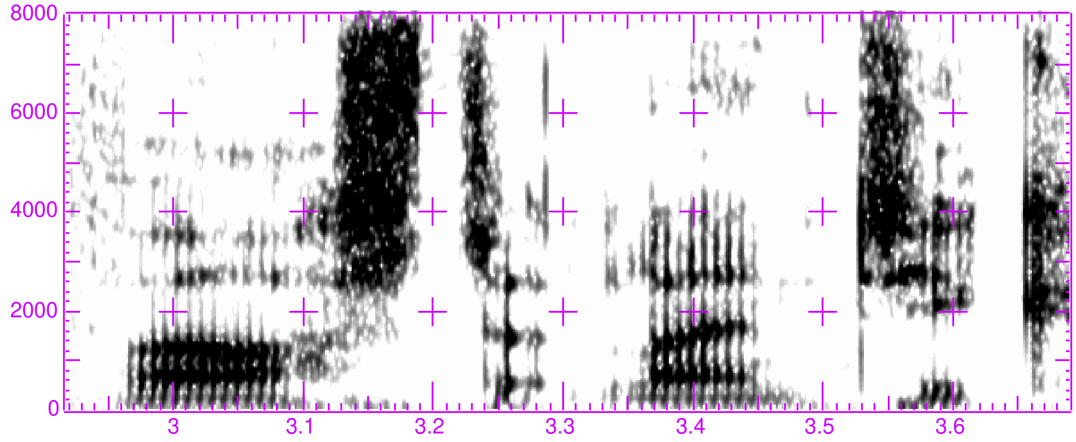
Modified Rhyme Tests were conducted on speech generated from a P-method LP version of the system trained on the M1 database, just prior to the transition to the TD-PSOLA synthesis scheme. This system was very similar to the final system, but did not include word final/initial clustering, or the transcription and segmentation improvements resulting from the coding alterations of Section 6.1.8. MRTs conducted with an LP version of the final system are discussed in Section 8.5.4. The test was performed using six experienced listeners, and the error rate obtained was 11.7%. An analysis of the errors occurring in the tests showed that 63% of them were with the phones /b, d, g, z, v, n, m & ng/. These errors could all be largely accounted for by failings in the synthesis scheme used. The bursts of /b, d & g/ were often synthesised badly because the burst required was really a transient signal, which could not be synthesised using either of the excitation signals available. The voiced fricatives /z & v/ required a mixed excitation, which was



(a) Synthesis using the P-method version of the final system



(b) Synthesis using the P-method version of the basic system



(c) Natural speech

Figure 8.1: Wideband spectrograms of the sentence fragment "...vast Atlantic...". The synthetic speech was produced using the P-method versions of the final system and the top ranked configuration of the basic system listed in Table 5.1. The same speaker was used to record both training databases used, and the natural speech shown.

not implemented, and the nasals, /n, m, & ng/, had spectra which contained zeros, and so were poorly reproduced by LP synthesis.

The poor performance of the LP based system cannot be blamed on the use of LP synthesis per se; the LP re-synthesised natural speech discussed in Section 5.6 also used LP synthesis, and yet obtained a much lower MRT error rate of 3.3%. There are several explanations for this difference in performance. These explanations fall into two broad categories, one concerned with the concatenative nature of the wholly synthetic speech, and the other with the additional deficiencies of the LP scheme used with the wholly synthetic speech.

The wholly synthetic speech was constructed from a sequence of clustered state-size synthetic segments, as defined by the HMMs. If the re-synthesised speech is regarded as being constructed from a similar sequence of segments, then it is clear that this sequence is the most ideal sequence possible, since the segments are based on exactly the speech required. Similarly, the segment durations used in the re-synthesised speech were the most ideal possible. The HMM clustering system and the duration model used to construct the wholly synthetic speech aimed for these ideals, but did not achieve them. As a result, the re-synthesised speech contained more perceptual clues than the wholly synthetic speech, and performed better in the MRTs.

The LP scheme used for the wholly synthetic speech was similar to that used for the re-synthesised speech, but was in fact inferior to it in several ways. The LP coefficients used to synthesise the wholly synthetic speech were estimated by pooling many segments of database speech. Since this speech was in practice not all identical, either within segments or between segments, this resulted in imprecise LP coefficients with broad formant bandwidths, and hence poorly-defined and non-resonant speech. Furthermore, the wholly synthetic speech could use only one excitation type for each state, whereas the re-synthesised speech could use more than one in a corresponding segment of speech since voicing was determined on a frame by frame basis. This meant that more reasonable approximations to the transient bursts of voiced plosives, and the mixed excitation required when synthesising voiced fricatives were possible in the re-synthesised case.

The explanations in the first category above represent problems inherent to wholly synthetic speech, and could not be solved by using an alternative synthesis scheme. However, the problems in the second category could be solved by such a change, or even by improving the LP scheme used. For example, a scheme in which a single segment was used to estimate a sequence of LP coefficients for each state, allowing multiple voicing transitions within each state, would solve many of the problems. However, this approach was not pursued, partly because such ideas had already been partially explored with little success (see Section 5.5.2), but principally because the performance of such a system would still have been limited to that of LP re-synthesised speech, less the additional degradation due to the concatenative nature of the system. A fundamentally superior synthesis scheme was therefore sought in order to enable both the MRT score and the general speech quality to improve beyond the limits imposed by standard LP synthesis. As discussed in Chapter 7 TD-PSOLA was selected to be this alternative.

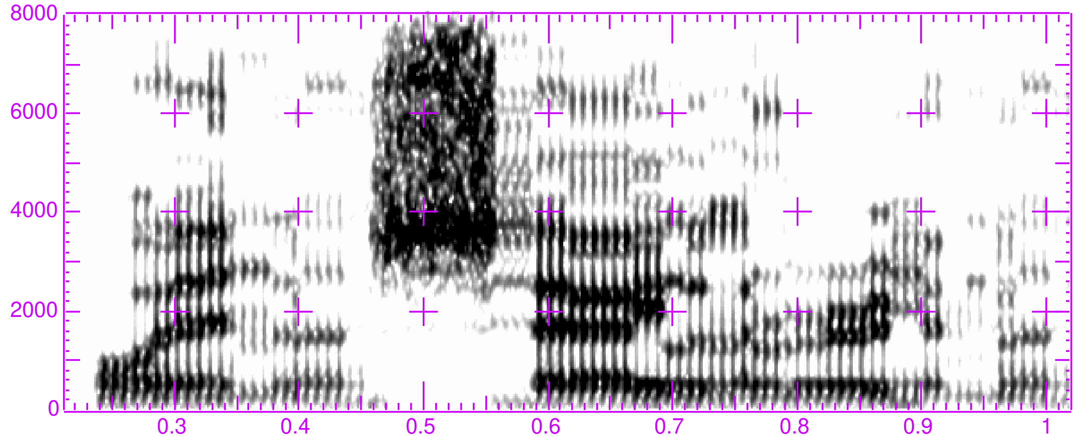
8.5 TD-PSOLA Results

This section begins by presenting the results of an early TD-PSOLA implementation in order to demonstrate the necessity of using entire segments to reproduce each clustered state, and the effect of widespread formant discontinuities. In Section 8.5.2 it presents an analysis of the results of the segment selection algorithm used. Section 8.5.3 presents examples of the speech produced by the TD-PSOLA version of the final system, and Section 8.5.4 the results of the large-scale MRTs conducted with this system. The size of the waveform inventory which must be stored for the final TD-PSOLA system is then examined in Section 8.5.5, and the processing times required by this system discussed in Section 8.5.6. Finally some examples of prosody transplantation and voice transformation using the final TD-PSOLA system are presented in Section 8.5.7.

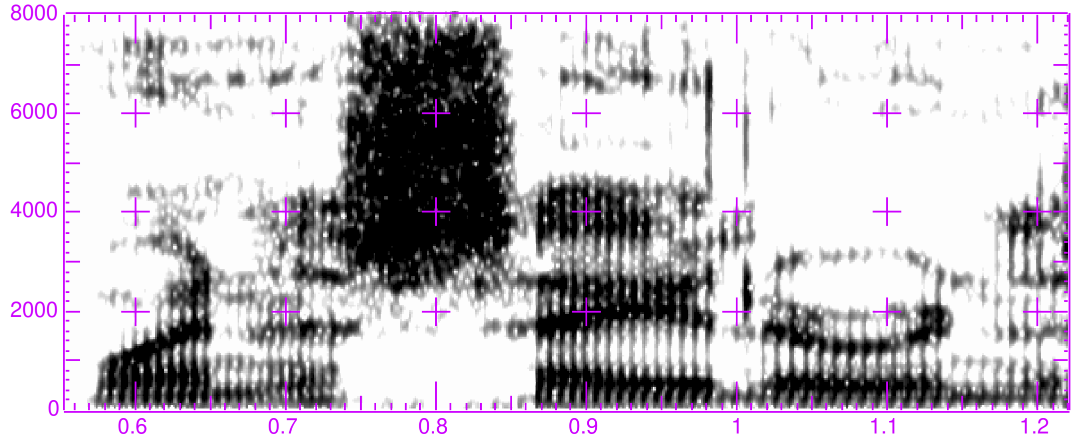
8.5.1 Early TD-PSOLA Results

Figure 8.2(a) shows a wideband spectrogram of the sentence fragment "When a sailor in a..." produced from speech example 33 on the accompanying compact disc. Figure 8.2(b) was obtained from speech example 36, which is a natural version of the utterance spoken by the speaker used in the M1 database. The synthetic speech was produced using a system incorporating an early TD-PSOLA implementation, trained on the M1 database. In this implementation each state was synthesised either as all voiced speech or all unvoiced speech. The original database segments used in synthesis were selected as described in Section 7.1, and so were not necessarily each of a single voicing type, although in practice they often were because the HMMs tended to segment the speech this way. In synthesis unvoiced states were produced as described in Section 7.3.1, but voiced states were produced by repeating only a single ST-signal selected from the original segment. This ST-signal was defined to be centred on the largest sample in the LP residual of the original segment. The voicing decision used in the LP system was found to be insufficiently accurate for this system, and a better algorithm, based on analysis of the log-power spectrum of speech, was developed. This algorithm is not described further here since it was used only temporarily, and did not form part of the final system.

This implementation was made as an intermediate step, before the research into methods of marking the moments of glottal closure discussed in Section 7.2 was carried out. This was before the syllabic clustering, position in word clustering, and coding alterations of Section 6.1.8 were added to the system. The speech is the same as that reported in (Donovan and Woodland 1995b), and had an MRT score of 4.7% when tested on six experienced listeners. It is shown here to demonstrate the effect of synthesising each voiced state by repeating the same pitch pulse, to show why it is necessary to use all of each original segment during synthesis. The effect can clearly be seen in Figure 8.2(a), where formant discontinuities between adjacent voiced sections can be observed which are not present in Figure 8.2(b). The audible effect is an overall artificial quality to the synthetic speech, which is perhaps best appreciated by listening to the speech in conjunction with speech example 35, which is the same utterance synthesised by the full TD-PSOLA implementation, as discussed in Section 8.5.3. It is interesting to compare the audible



(a) Synthesis using an early TD-PSOLA implementation



(b) Natural speech

Figure 8.2: Wideband spectrograms of the sentence fragment "When a sailor in a...". The synthetic speech was produced using a system incorporating an early TD-PSOLA implementation, trained on the M1 database. The natural speech was spoken by the speaker used in the M1 database.

differences in this case with the result of a similar experiment in Section 5.6.1, where no difference could be heard between LP synthesised speech with and without similar formant discontinuities. The difference was thought to be audible in the present case but not in the LP case because in the latter the effect was probably masked by the general poor quality of the LP synthesis scheme used.

Speech example 33 is also useful because it demonstrates the problem which was being experienced with the synthesis of /l/s before the introduction of the syllabic clustering described in Section 6.2.3. A burbling sound can be heard through the /l/ of "...sailor..." as the formants jump about. The exact location of the worst discontinuity is difficult to pinpoint in Figure 8.2(a), because the formant structure in the second half of the spectrogram is not very smooth in general. However, by careful listening the location can be identified as occurring at about the 0.76s time-mark in the spectrogram, with another discontinuity between the /ih/ and the /n/ of "sailor in...", at about the 0.91s time-mark, compounding the audible effect.

Speech example 34 was synthesised using essentially the same system trained on the F1 database. This speech should be compared to that in speech example 43, which was generated using the full TD-PSOLA version of the final system.

8.5.2 Segment Analysis

An analysis of the speech segments selected for use in synthesis by the algorithm described in Section 7.1 was conducted, in order to determine how much modification the TD-PSOLA algorithm was required to impose on them during synthesis.

With the final TD-PSOLA system the duration scaling factor in equation 5.3.1 was set to 0.1 when synthesising continuous speech, and 0.5 when synthesising isolated words. Analysis of the actual durations present when the system was trained on the M2 database revealed that this corresponded to each state being synthesised for an average of 1.04 times and 1.22 times its average duration when synthesising continuous speech and isolated words respectively. The 80% duration threshold used in the segment selection algorithm described in Section 7.1 therefore meant that stretching factors averaged less than 1.30 and 1.52 in the two cases. Note that both these stretching factors are within the acceptable range for the TD-PSOLA implementation used (see Section 7.3.2).

The segments selected from the M2 database by the final system were examined to determine the local F_0 between every pair of adjacent V-marks within a segment. The mean F_0 of the speech in the segments and the standard deviation from this mean were then calculated. It was found that the mean F_0 was 112.3 Hz, with a standard deviation of 14.3 Hz. Assuming a normal distribution, over 95% of the F_0 values will lie within two standard deviations of the mean. Thus, if the speech is transformed to the mean frequency during synthesis, then with over 95% of the speech this will involve scaling F_0 up by factors of 1.34 or less and down by factors of 1.25 or less. It is encouraging to note that both of these factors are well within the ranges for which the TD-PSOLA implementation has been shown to give reasonable performance (see Section 7.3.2). Note that the speech in this thesis based on the M2 database was synthesised at 116Hz, this being an earlier estimate of the average F_0 of this speaker.

When synthesising using a variable pitch track, larger pitch alteration factors might be required if a segment at one pitch extreme has to be synthesised at the other pitch extreme. Assuming the above distribution again, such pitch alteration factors would be unlikely to exceed 4 standard deviations, and this corresponds to a pitch raising or lowering factor of 1.68. Thus, in such cases the required pitch transformation factor may exceed the acceptable range for the current TD-PSOLA implementation. The solution to such problems is probably either to improve the TD-PSOLA implementation, to use some other form of signal processing in place of TD-PSOLA, or to store multiple segments for use during synthesis with different inherent pitches. This last idea is explored further in Section 9.1.3.

8.5.3 Final TD-PSOLA Results

This section presents speech waveforms, wideband spectrograms, and audio examples synthesised using the full TD-PSOLA version of the final system, trained on the M2 and F1 databases. The system trained on the M2 database used the associated laryngograph signals to find the moments of principle excitation of the vocal tract, and the system trained on the F1 database used the LP residuals of the speech.

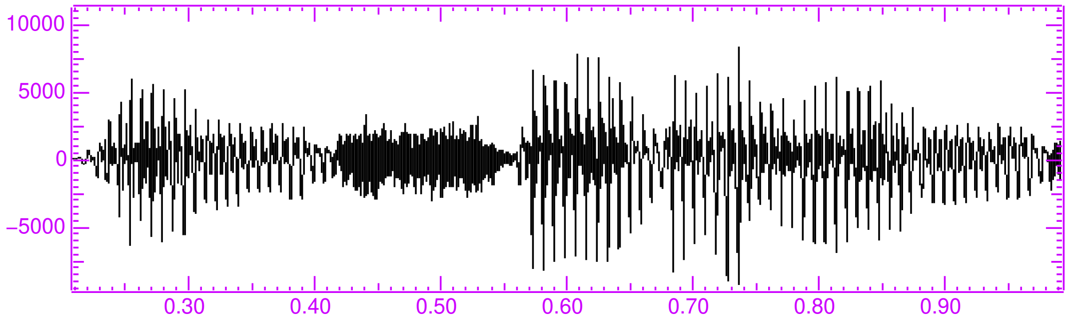
Figures 8.3(a) & (b) show the speech waveform and wideband spectrogram of the sentence fragment "When a sailor in a...", taken from speech example 35. The speech was synthesised using the TD-PSOLA version of the final system, trained on the M2 database. Figure 8.3(c) is a repeat of Figure 8.2(b), which was obtained from speech example 36, a natural version of the same utterance spoken by the speaker used in both the M1 and M2 databases. Figures 8.3(d) & (e) show the speech waveform and wideband spectrogram of the sentence fragment "...vast Atlantic...", taken from later in the same synthetic utterance. Figure 8.3(f) is a repeat of Figure 5.2(f), which was also generated from the natural speech of speech example 36.

With the exception of the second half of Figure 8.3(b), the synthetic speech spectrograms in Figure 8.3 can be seen to be quite similar to those of the equivalent natural speech. The most noticeable differences are that there are more pitch pulses in the synthetic speech spectrograms, because the synthetic speech is slower than the natural speech, and that the first /t/ of "Atlantic" is released in Figure 8.3(e). Formant discontinuities at state boundaries are usually small, with the result that state boundaries are often difficult to identify from the spectrograms.

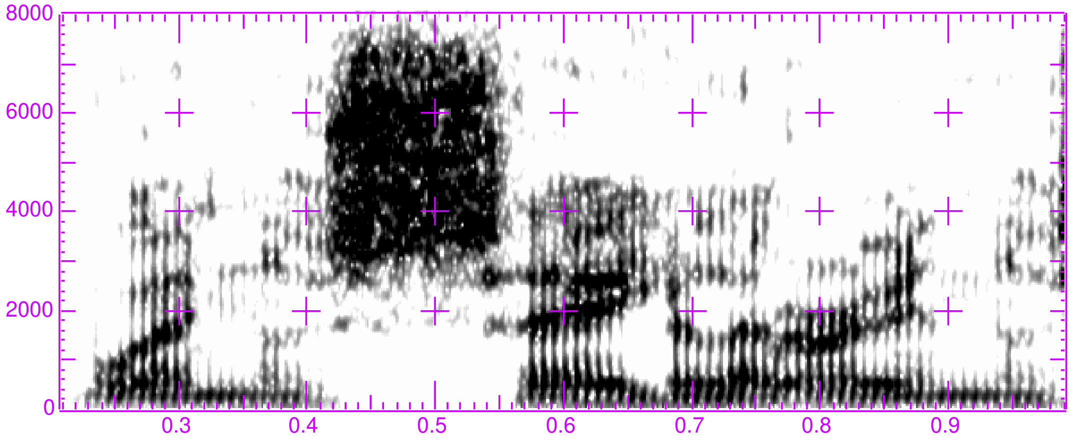
In the second half of Figure 8.3(b), however, the formant continuity is less good. Although the introduction of syllabic clustering did improve the synthesis of many /l/s, the main defect in the speech corresponding to this region of the spectrogram seems to be a burbling sound through the /l/ of "sailor". When looking at the formant structure in the spectrogram it is perhaps surprising that the synthetic speech sounds as good as it does. This phenomena has also been observed with other synthetic speech.

Figure 8.4 shows the waveform and a wideband spectrogram of the sentence fragment "When a sailor in a...", taken from speech example 43. This speech was generated by training the final TD-PSOLA system on the F1 database. The natural speech spectrogram was generated from speech example 44, a recording of the same utterance by the speaker used in the F1 database. As with the male speech of example 35, problems can be heard in the synthetic speech around the /l/ of "sailor". However, in the female speech the problems are more serious, because the formant discontinuities are accompanied by numerous errors resulting from pitch-mark identification errors in the original database. These errors occurred because the F1 database did not include a laryngograph signal, and the alternative pitch-mark identification algorithm based on the LP residual of the speech was far from perfect. During synthesis these errors resulted in voiced speech being treated as if it were unvoiced, which can be heard as hoarseness in the synthetic speech, particularly around the /l/ of "sailor", but also more generally.

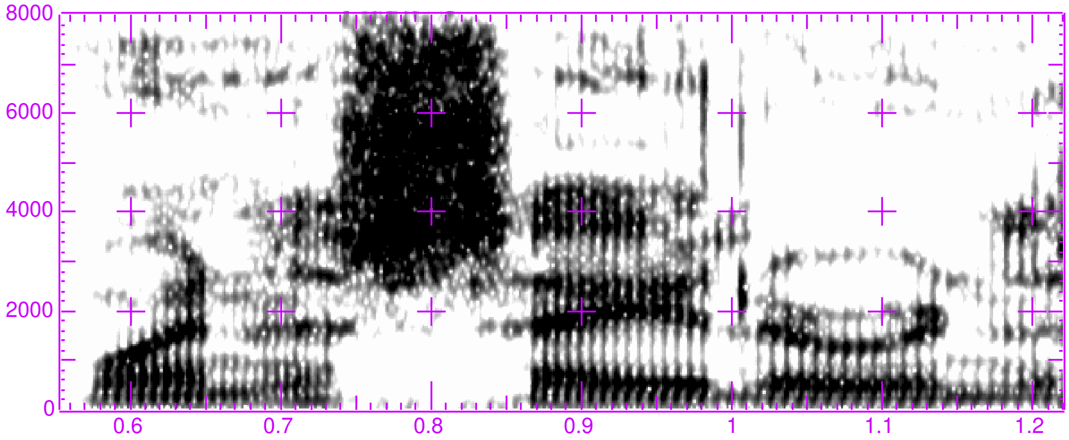
Speech examples 37-42, 45, 46, 48-51, 53, and 54, are also available as further demonstrations of the performance of the final TD-PSOLA system, trained on the M2, F1, F2, and M3 databases. Speech example 41 demonstrates and describes some of the problems which still occur with speech synthesised by the system. Speech example 42 was synthesised using a stylised pitch track, originally taken from a natural version of the utterance, but then altered by hand to match the synthetic durations.



(a) Synthetic speech waveform

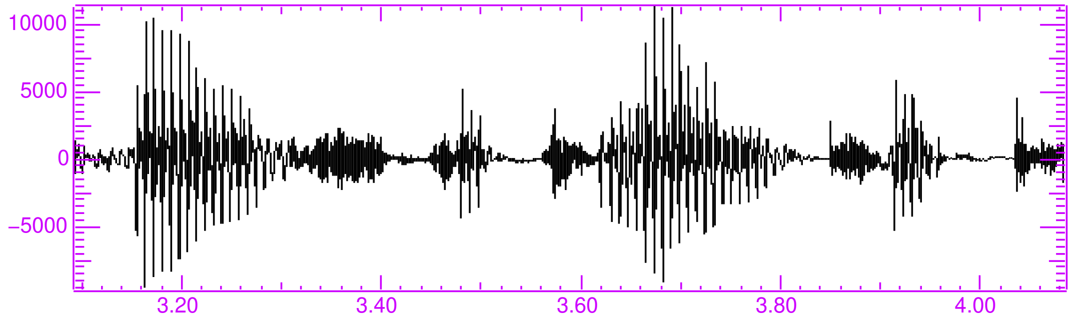


(b) Wideband spectrogram of synthetic speech

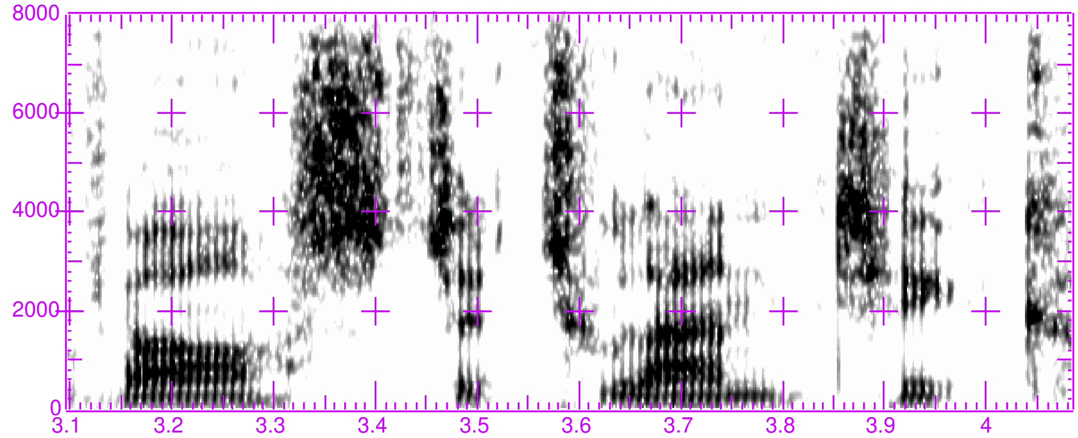


(c) Wideband spectrogram of natural speech

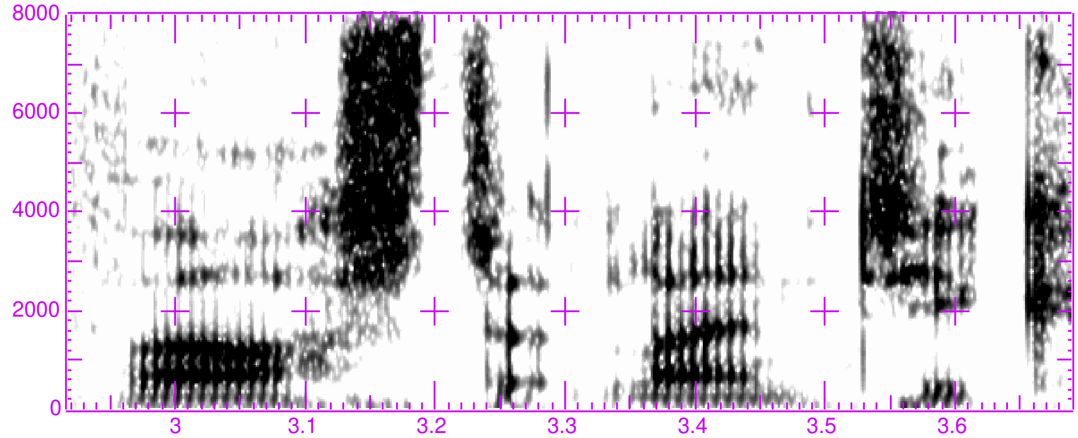
Figure 8.3: A synthetic speech waveform and wideband spectrograms of the sentence fragment "When a sailor in a...". The synthetic speech was produced using the TD-PSOLA version of the final system, trained on the M2 database. The natural speech was spoken by the speaker used in the M2 database.



(d) Synthetic speech waveform

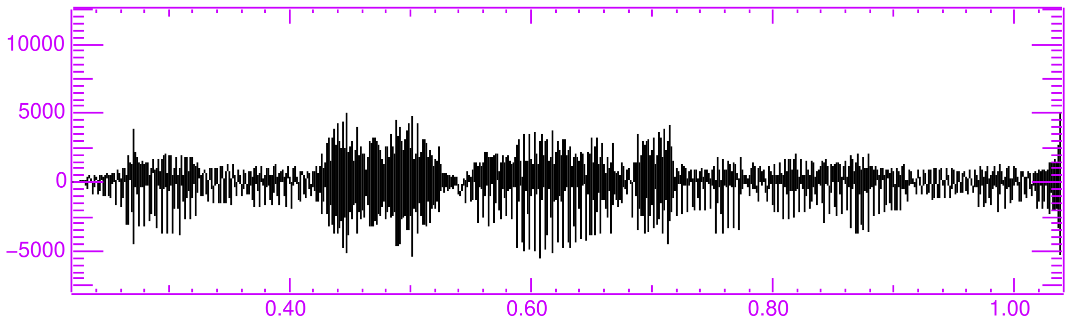


(e) Wideband spectrogram of synthetic speech

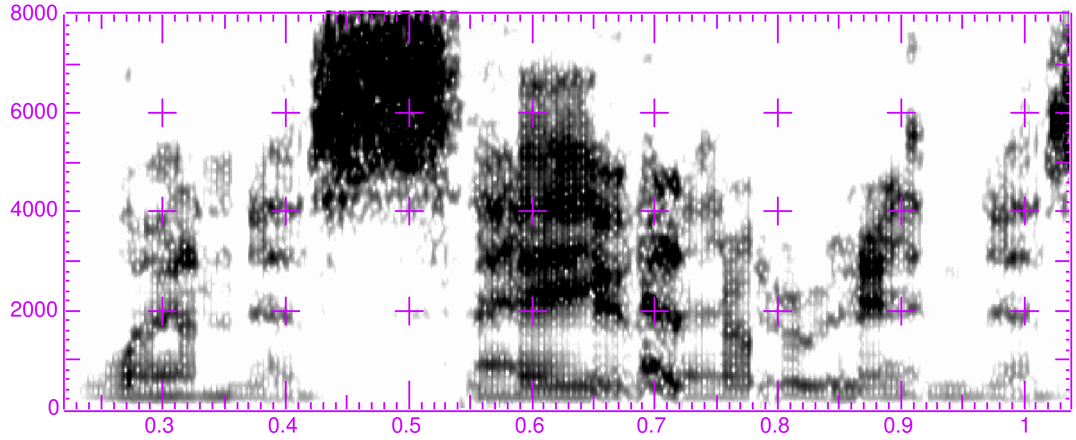


(f) Natural speech

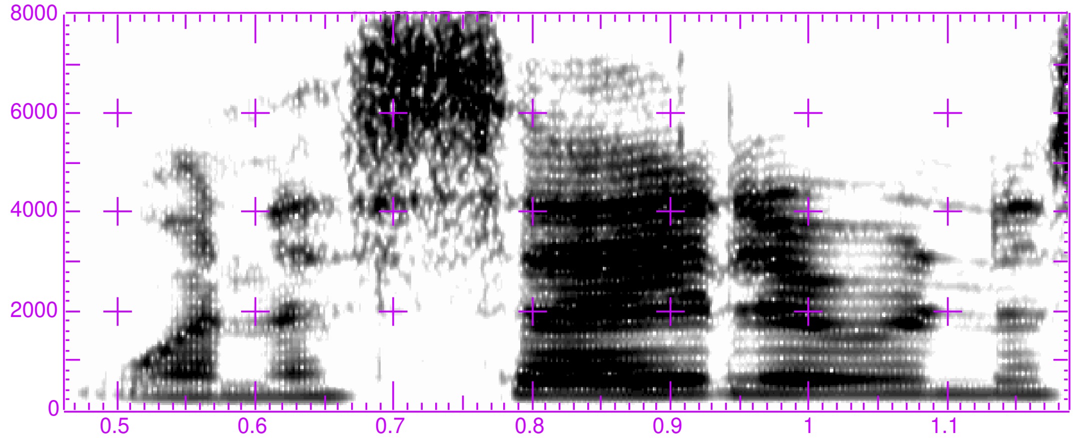
Figure 8.3 (continued): A synthetic speech waveform and wideband spectrograms of the sentence fragment "...vast Atlantic...". The synthetic speech was produced using the TD-PSOLA version of the final system, trained on the M2 database. The natural speech was spoken by the speaker used in the M2 database.



(a) Synthetic speech waveform



(b) Wideband spectrogram of synthetic speech



(c) Wideband spectrogram of natural speech

Figure 8.4: A synthetic speech waveform and wideband spectrograms of the sentence fragment "When a sailor in a...". The synthetic speech was produced using the TD-PSOLA version of the final system, trained on the F1 database. The natural speech was spoken by the speaker used in the F1 database.

Word Final Phones		Word Initial Phones	
phone	% of errors	phone	% of errors
/d/	29	/t/	29
/n/	18	/d/	17
/m/	14	/l/	17
/g/	12	/h/	10
/ng/	6	/b/	7
		/k/	7

Table 8.3: The most frequently occurring errors in the Modified Rhyme Tests conducted on the TD-PSOLA version of the final system.

8.5.4 Final MRT Results

MRTs were conducted to evaluate the speech synthesised by the TD-PSOLA version of the final system, trained on the M2 database. The aim of these tests was to produce an MRT score which was comparable with those for other systems listed in (Logan et al. 1989). As described in Section 4.2, most of the experimental procedures used in all the MRT tests conducted during the course of this work were very similar to those used by (Logan et al. 1989). However, most of the tests were conducted using only six experienced listeners. The final tests were therefore conducted using thirty-six inexperienced listeners, in order to provide a more reliable figure for comparison. The listeners used were gathered by advertising for native British English speakers, with no history of hearing problems, and no previous experience of working with synthetic speech. These conditions were imposed in order to obtain listeners equivalent to those used by (Logan et al. 1989). Each listener was informed that they would be paid £1 for performing a test, plus an additional 50 pence if he or she got 5 or fewer words wrong. Each listener was told that the bonus was easy to obtain if he or she concentrated. A brief description of the test was given by the author, and then the test performed as described in Section 4.2. The bonus was included to encourage the listeners to concentrate on the test. The intent was that every listener should receive the bonus, and in fact every listener did.

The MRT score obtained was 5.00%, with a standard error of 0.47%. The most frequently occurring errors are shown in Table 8.3. As can be seen, the voiced plosives, /d & g/, and nasal errors accounted for over 79% of the word final errors occurring. The word initial errors were more distributed, with errors in the identification of both voiced and unvoiced plosives, /l/ and /h/ all being significant.

Many of the word final errors with plosives occurred between words which differed only in voicing. For example "bead" was often recognised as "beat", and "pig" as "pick", etc. This was because although the final plosive was often well produced, the length of the vowel was often inappropriate; an unvoiced plosive following a vowel should shorten that vowel, (Klatt 1987). The clustering questions "R_Voiced-Closure" and "R_Unvoiced-Closure" did exist in the question list, but were often not used in these cases to split nodes during tree-building. The problem was perhaps that, with the speech coding used, the acoustic difference between such vowels was small, despite the fact that their durations differed considerably. Examples of the problem can be heard in speech example 41. A possible solution to this problem would be the use of an improved duration model (see Section 9.1.2).

It is possible that the large number of errors with word final nasals was due to the system being trained on continuous speech. It is likely that in the continuous speech of humans, nasals are less clearly produced than in isolated words, since their identity is often obvious due to context. In isolated words however, humans are aware of the possible ambiguity, and so deliberately produce them more clearly, whereas the synthesis system merely slows down nasals extracted from continuous speech. These ideas suggest that the system will always have problems with synthesising isolated words, unless it is specifically trained on an isolated word database.

The word initial errors are more diverse, and less easy to explain. Many of the errors with word initial /l/s occurred because the words sounded as if they began with the sequence /b l/. Furthermore, many of the word initial /t/ errors occurred in words beginning with the sequence /t oh/, which sounded like /k l oh/. These errors suggest that the modelling and segmentation of /l/s with the final system was still not perfect, a result consistent with the synthetic speech examples presented in Section 8.5.3. Many more word initial /t/ errors occurred with the word "told", which was frequently mistaken for "sold". Most word initial /d/ errors occurred with "dip", which sounded like "lip". The exact reasons for these errors, and the word initial /h/ errors, are difficult to specify, but appear to be problems more with duration estimation, or unit selection, than with segmentation.

The score of 5.00% was believed to be fairly directly comparable with those obtained by (Logan et al. 1989), as listed in Table 4.1. As described in Section 4.2, the experimental setup used was extremely similar to that used by (Logan et al. 1989), and the differences in arrangements not thought likely to affect the results. It has been suggested that a local test of the DECtalk system could be made to establish the comparability of the results, but this was not done since only British subjects were available, and it was thought that the significant mismatch in accents would render the results so unreliable that it was not worth the significant effort that would be involved.

Finally, note that the final TD-PSOLA system was also tested on six experienced listeners, and that the result of this test was an error rate of only 2.7%. This demonstrates how much the listeners had adapted to the system over many tests, despite never being given any feedback on their individual performances. It also shows how necessary it was to use inexperienced listeners to obtain results which could be quoted in absolute terms. This point is also illustrated by the result of an MRT conducted on the LP version of the final system, again using six experienced listeners. The error rate obtained was 17.0%, which was much higher than that obtained using an almost identical, but probably inferior system, as reported in Section 8.4. However, in the light of the above result using inexperienced listeners, the most likely explanation for this poor performance is simply that over 3 months had elapsed since the listeners used had done their last test, with over 6 months time elapsed since their last test on LP synthesised speech. In short, the listeners were out of training, and hence performed badly.

8.5.5 Inventory Size

The size of the waveform inventory which must be stored for use during synthesis is an important factor for possible real world applications of the synthesiser. For each segment stored, an extra few milliseconds of speech to the left and right of the leftmost and rightmost pitch-marks respectively also had to be stored, in order that the Hanning windows used by the TD-PSOLA algorithm were applied to a continuous piece of speech. The TD-PSOLA implementation used Hanning windows whose size was set by the synthesis pitch, and therefore the exact amount of storage required depended on the lowest pitch required in synthesis. For the final version of the system, trained on the M2 database, the storage requirement to enable synthesis down to a F_0 of 70Hz was 293 seconds of speech, which required about 9.4MB of disc-space since it was sampled at 16kHz with 16 bit resolution.

For Personal Computer based applications 9MB represents a substantial memory requirement at the time of writing. Accessing the inventory from disc would probably not be possible, because of speed constraints. Whilst supplying waveform inventories loaded onto relatively inexpensive ROM chips would be one possibility, a software only solution would generally be more flexible, and thus more desirable. Methods to reduce the required storage size, preferably to less than 1MB, must therefore be sought.

Preliminary experiments were conducted to investigate the effect of using fewer clustered states on both the speech quality produced by the system, and the size of the waveform inventory to be stored. Two methods of achieving this were examined. In one case, the final system was trained on the whole M2 database, but with the minimum number of occurrences per leaf node clustering threshold increased to 30. This produced a synthesis system with 2523 states, the waveform inventory of which comprised 129 seconds of speech, and required 4.1MB of storage. In the other case the system was simply trained on only the first half of the M2 database. This produced a synthesis system with 3088 states, corresponding to 159 seconds of speech, or 5.1MB of storage. Synthetic speech from both synthesis systems is available on the accompanying compact disc as speech examples 55 and 56 respectively. As can be heard, the results in both cases compare well with the same speech as synthesised by the standard system, available as speech example 35. The initial section of speech example 56 can be heard to burble more than that of speech example 55, which in turn sounds perhaps slightly worse than that of speech example 35. However, the rest of the three speech examples are all very similar in quality. Spectrograms of the speech agree with this analysis; the formants of the early part of speech example 56 are much more broken than those of the other two examples, which are fairly similar in smoothness and continuity. Thus it appears that there may be considerable scope for reducing the inventory size required using either of these methods, with at the moment some indication that clustering the available data less finely is a superior method to using less training data. Indeed, if this approach was coupled with a scheme to ensure formant continuity during segment selection, very large reductions in inventory size could be achieved. This idea, and other possible methods for reducing the inventory size, are discussed further in Section 9.2.

8.5.6 Processing times

The final system can be retrained on a new voice in less than 48 hours. A one hour speech database takes about 4.5 hours to record. The processing is performed by two scripts, the first of which finds the moments of glottal closure in the speech, and the second of which constructs the synthesis system. If a laryngograph signal is being used, the former takes approximately 40 minutes on an otherwise unloaded SGI R4400 Indigo. If the moments of glottal closure are to be found using the LP residual, then this may take anything up to a week of CPU time, depending on the values of the thresholds used when running *epochs*. The second script, which constructs the synthesis system, takes 40 hours to run on an otherwise unloaded HP735-99. The first 27 hours of the this time is spent working with monophone models, iteratively re-estimating, aligning, and using the methods of Section 6.1 to obtain the best possible phonetic transcription of the database. The structure of the monophone section of the second script evolved over the course of this work, and is undoubtably very inefficient, and could probably be speeded up considerably.

Synthesis was conducted using a script which ran ten different binaries, and many lines of shell script, writing the output of each stage of the synthesis process to disc. There was an initial overhead for each utterance to be synthesised while the pronunciation dictionary was loaded into memory, and the manual selection between optional pronunciations made. The script then ran with no further human intervention. There was a 33 second overhead associated with synthesising an empty utterance, when run on an otherwise unloaded HP735-99. There was then an additional cost of approximately 14 times the duration of the utterance to be synthesised, when running on the same machine, with most of this additional time being used by the TD-PSOLA synthesiser. Note that the speech segments concatenated by the synthesiser were not actually cut from the original database in advance. The TD-PSOLA synthesiser was therefore extremely inefficient, loading a whole database sentence from disc, across a local network, to extract only one clustered state sized segment from it, and then discarding that sentence. There was therefore a huge scope for improvement in the amount of time required for synthesis. Ideally, the script would be replaced by a single binary which performed all aspects of the synthesis, with the dictionary and waveform segment inventory pre-loaded into memory. It is likely that the system could then synthesise speech in real time.

8.5.7 Voice Transformation

If both the durations and the F_0 for a synthetic utterance are obtained from a natural version of the utterance, then the quality of the underlying synthesis system can be heard, free from the degradations introduced by using synthetic durations, and stylised pitch tracks. Such a process could be called *prosody transplantation*. If the speaker used to provide the prosody is not the same as the speaker mimicked by the synthesis system, then the process also enables a degree of *voice transformation* to be performed.

A system was implemented to enable voice transformation to be performed between any two voices for which a synthesis system had been constructed, provided the words of the utterance were known. The models of the source speaker were used to transcribe and segment the natural version of the utterance. The pitch track, obtained by processing a

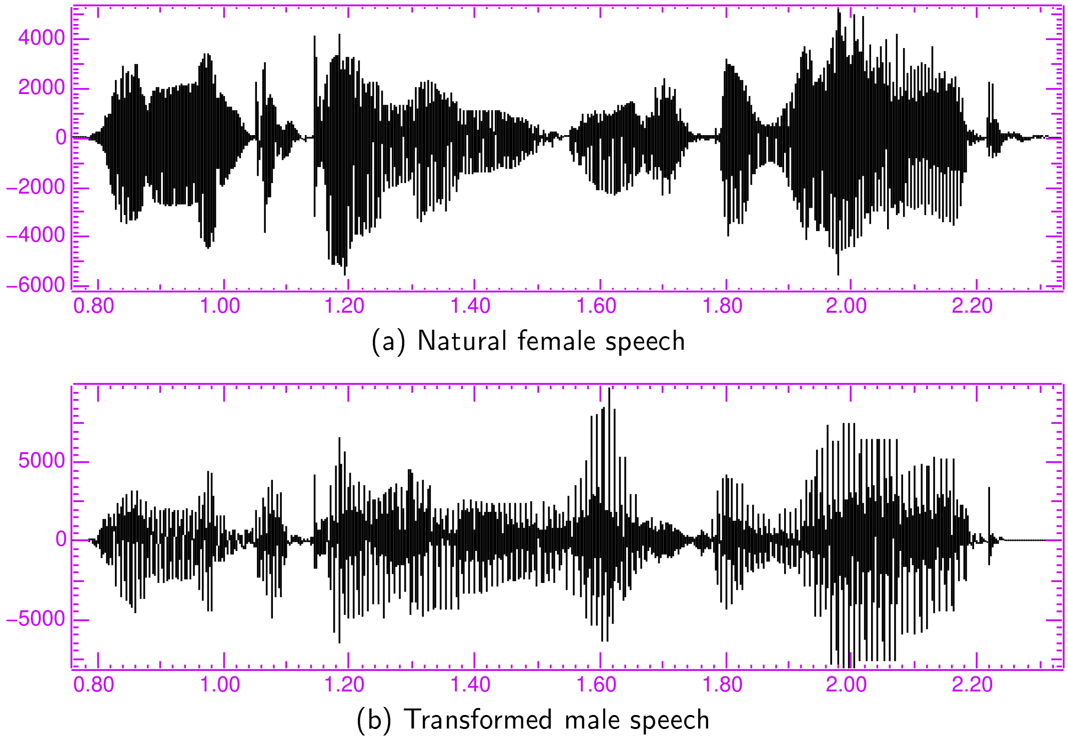


Figure 8.5: Waveforms of the sentence fragment "In the beginning was the word", taken from natural speech spoken by the female speaker used in the F1 database, and speech transformed from this to have the voice of the male speaker used in the M1 and M2 databases.

laryngograph signal recorded at the same time as the natural version of the utterance, the phone sequence and the *phone* durations of the natural utterance, were then used to construct the synthetic speech. The durations of each state of the synthetic speech were found in the normal way, and then scaled such that the sum of the durations of the consecutive states representing a single phone was the same as the equivalent phone's duration in the natural utterance. This was done in case the distribution of durations between the states within a given phone was different for the source and target speakers. If the source and target speakers were different, then the pitch was linearly scaled by a factor equal to the mean pitch of the source speaker divided by the mean pitch of the target speaker.

Figures 8.5(a) and (b) show waveforms of the sentence fragment "In the beginning was the word,...", taken from speech examples 61 and 62 respectively. The former was natural speech from the speaker used in the F1 database, and the latter was the result of transforming this into the voice of the male speaker used in the M1 and M2 databases. The pitch was scaled by a factor of 116/210, being the approximate average pitches of the two speakers in Hz. As can be heard, the synthetic speech is of very high quality, with perhaps the only problems being that it failed to reproduce the phrase boundary after the word "beginning", and that the speech does not sound quite resonant enough.

Speech examples 57-60, 63, and 64, demonstrate further examples of prosody transplantation. Examples 57 and 58 demonstrate the result of transplanting prosody onto synthetic speech generated from a system trained on the same voice. Although some allowance must be made for the inadequacies of the automatic transcription system used, which was probably inferior to that used during synthesis system construction, the fact that the two examples do not sound identical is an indication of the limitations of the final system. Note that the transformations described here require the specification of the words of the original utterance. In the future it may be possible to determine the words of the utterance automatically, or alternatively, as is discussed in Section 9.3.1, to do without the information altogether.

8.6 Discussion

The previous sections have demonstrated that the final system produces high quality synthetic speech, with a very respectable MRT score, and a high degree of naturalness. It can also be retrained very rapidly on easily obtainable training data, and is not voice dependent. Note that theoretically, with a new dictionary and set of clustering questions, it is not language dependent either. The performance is due to both the use of a high quality synthesis scheme, and the automatic selection of a good set of sub-word units. An explanation as to why the HMM state based approach selects such a good set of sub-word units is attempted below. Following this is a discussion of the problems still present with the final system.

The clustering process used by the system is similar to those of other speech synthesis systems developed in recent years discussed in Section 2.3.2. The major difference here is that the clustering is state-based instead of phone-based. State-based clustering has been shown to out-perform phone-based clustering for speech recognition, (Young et al. 1994), (Odell et al. 1994), and similar gains can therefore be expected to occur in speech synthesis applications. HMM states are represented by a single feature vector, and thus lend themselves to clustering more directly than longer segments. Furthermore, the individual HMM states within a phone-model can be clustered independently, which enables better use to be made of a given amount of training data. The smoothness of the formants of the synthetic speech is undoubtably also due in part to the use of 1st and 2nd order differential parameters in the feature vector representing each state. This means that states can be characterised by their dynamic features in addition to their static features. During clustering the relative importance of the dynamic and static features, and of individual dimensions within these vectors, is determined automatically using a log-likelihood distance measure calculated using the variances of the parameters, which were themselves obtained through training.

The segmentation process, using HMMs, is similar to the previous attempts discussed in Section 2.2.1. In this system, however, not only is the HMM system more sophisticated than those discussed in Section 2.2.1, but more importantly the units are segmented using the models created by the clustering process, and themselves become the synthesis units. Using the clustered state models to perform the segmentation enforces a large degree of consistency throughout the entire segmentation process. Boundaries between states will be well defined when those states are very different, when accurate segmentation matters most for synthesis, and less well defined when the states are more similar, which is when it matters least for synthesis. Furthermore, even if boundaries are located inaccurately, as judged by a human observer, provided the boundaries are consistently placed that way the "error" will be undone in synthesis. That is, segments which are likely to appear adjacent to each other during synthesis are likely to come from states which were often adjacent during construction. This means that the segments used in synthesis are likely to concatenate smoothly, because the boundaries of the states involved are likely to have been segmented consistently with each other. Thus to some extent, consistency is more important than accuracy. However, new state sequences can occur during synthesis which were not present in the original database, and so accuracy is also important.

The largest inadequacies of the final system are the lack of a proper text to phoneme conversion system, a phrase boundary identification system, and a pitch estimation system, the large size of the waveform inventory required for synthesis, and the slow speed of the synthesis code. However, none of these were aims of the current research. The remaining problems which were within the aims are discussed below.

The largest problem which occurs during synthesis is undoubtably the occasional serious formant discontinuity, which is heard as a burbling in the synthetic speech. Such discontinuities have often been seen to be the result of clustering inadequacies, which result in too much variability amongst the segments pooled into the same clustered state. Adjacent states may then be synthesised using segments with widely differing formants, resulting in discontinuities. Such a problem did, and to some extent still does, occur with /l/s in particular. Another important problem during synthesis is the occasional unrealistic duration, a few examples of which can be heard in speech example 41. A related effect is the failure of the system to distinguish between vowels followed by voiced and unvoiced consonants. Both are a result of the durations in the system being effectively a by-product of the acoustic clustering. Also important are segmentation errors in the training database, which occasionally cause artifacts in the synthetic speech.

Less prominent problems also remain. One is the lack of resonance that the synthetic speech seems to have, even when everything else is correct, as in speech example 62. It is not clear what is causing this effect, but the tiny formant discontinuities between adjacent segments, and the imperfections introduced by the TD-PSOLA implementation, especially when reducing the pitch, are the prime suspects. Another is the slightly artificial quality of the synthetic speech based on the F2 database. This appears to be largely due to the "plastic tube" effect (see Section 7.3.2) when synthesising some fricatives, but can also be heard to a lesser degree with some voiced speech. Another problem, also demonstrated in speech example 41, is that bursts are always released during synthesis, which can lead to the synthetic speech sounding hyper-articulated. Finally, the system is likely to always have some problems when synthesising isolated words, since it does so by slowing down continuous speech, without adding emphasis which may be necessary with isolated words due to the lack of context information available to the listener. Possible solutions to many of these problems are discussed as possible future work in the next chapter.

Chapter 9

Conclusions & Future Work

This chapter begins with a discussion of possible solutions to many of the remaining problems with the speech synthesised by the final system which were within the aims of the current research. Methods to reduce the size of the waveform inventory which must be stored for use in synthesis are discussed in Section 9.2, and other future possibilities connected with the current research in Section 9.3. Finally, the conclusions of the current work are presented in Section 9.4.

9.1 Improving the Speech Quality

This section presents possible solutions to many of the problems remaining with the synthetic speech which were within the aims of the current research, as discussed in Section 8.6.

9.1.1 Improved Clustering

Many major formant discontinuities, as with /l/ for example, are caused by inadequate clustering, which results in groups of very different segments of speech being present in the same clustered state. During the synthesis of multi-state phones, segments from the different groups may then be concatenated causing formant discontinuities. One possible solution to this problem is therefore to provide new clustering questions, and associated database labelling, to enable such different groups of segments to be separated. For example, labelling syllable boundaries, and asking questions about phone position within syllables might enable many distinctions to be made. Alternatively, as discussed in Section 6.2.3, clustering on the basis of wider phonetic context might perform a similar role, without the difficulties associated with assigning syllable boundaries.

9.1.2 Duration Trees

The durations used by the final system were essentially a by-product of the acoustic clustering process. It is very likely that a better set of durations could be obtained using a separate set of decision trees clustered on the basis of duration data, instead of acoustic data. This clustering could be performed at either the state or the model level. Wider context questions, such as position in phrase, position in sentence, etc., could be asked during the trees' construction, in order to reduce the number and severity of the occasional unrealistic durations which occur with the current system, as mentioned in Section 8.6.

The system just described could result in a long duration being specified for a context for which a relatively short waveform segment had been selected to represent the appropriate acoustic state. With the current TD-PSOLA synthesiser, the large duration stretches required would be likely to cause problems during synthesis, particularly for unvoiced speech. One possible solution to this problem is therefore to adopt an alternative synthesis technique. Another is to simply store more than one waveform segment for any state comprising segments of widely differing durations.

9.1.3 Segment Selection

The segment selection algorithm contained in the final system, described in Section 7.1, works reasonably well, but is actually quite simplistic. As has already been suggested in the previous section, the storage of multiple segments for a state comprising segments of widely differing durations would be likely to reduce the burden on the synthesis signal processing. Similarly, multiple segments could also be stored for states comprising segments of widely differing pitches, or acoustics. Selection of appropriate segments during synthesis could then reduce both the amount of pitch and duration modification necessary, and the size of discontinuities at concatenation boundaries. This would reduce the demands on the signal processing used in synthesis, which, as discussed in Section 1.7, is always likely to be beneficial.

There are two problems associated with the dynamic selection system just described. Firstly, for many applications (at the time of writing) it is impractical to store the entire training database for use in synthesis, and therefore some form of pre-selection must be used to determine which segments are worth storing. Secondly, an algorithm must be devised to select between the segments available for each state during synthesis, to obtain the set of segments which most accurately produces the desired utterance. This second problem could be solved by using a dynamic programming algorithm to perform the selection, minimising some predetermined cost function. Ideally, this cost function would specify the relative importance of concatenation discontinuities, and the selected segment deviating from the required pitch, energy, duration, and state acoustic mean vector. It might take the form of a set of cost curves specifying, for each factor, and for a particular synthesis technique, what cost was associated with what degree of deviation. For example, when using TD-PSOLA synthesis, the duration cost curve would rise sharply as the segment duration dropped below half the required duration. The first problem could then be solved by using these cost curves to cluster the segments comprising each state into a number of self-similar sub-groups, and then pre-selecting only one segment to represent each sub-group. This segment could be chosen by, for example, synthesising a quantity of test speech, and selecting the most frequently used segment from each sub-group. The number of sub-groups per state could be chosen on the basis of the inventory size required. However, determining these cost functions is likely to be very difficult, and therefore a simpler solution, using thresholds, is also attractive. Recent research by (Black and Campbell 1995) seems to indicate that human listeners prefer to hear speech which is smooth at the expense of acoustic accuracy, rather than vice-versa. Therefore, an alternative solution to the selection problem would be to use duration, energy and pitch thresholds to exclude very inappropriate segments during synthesis, and then use a dynamic programming algorithm to select between candidates solely on the basis of their concatenation smoothness. The number of segments to store for each state could be reduced using global energy and duration thresholds, as with the current system. It could be further reduced using methods similar to those described above, clustering only on the basis of the acoustic distance between segment endpoints. Again, the number of sub-groups per state could be chosen to give the required inventory size.

The segment selection procedures discussed in this section would encourage the use of a larger number of adjacent segments in synthesis than the current system, because no segments are likely to concatenate more smoothly than those which were originally adjacent, and the selection procedures encourage concatenation smoothness. Thus, the result would be a system which effectively concatenated variable length units, using longer units wherever they were available and it was advantageous to do so, whilst keeping the underlying state-based approach when longer units were not available. The availability of longer units would be reduced by pre-selection, but this itself aims to provide the best possible set of segments for each state for a given inventory size. Such a system would therefore effectively achieve the ideal described in Section 1.7.

9.1.4 Alternative Synthesis Schemes

The adoption of a parametric synthesis scheme in place of TD-PSOLA could help solve the problem of concatenation discontinuities, by enabling them to be smoothed away. FD-PSOLA, LP-PSOLA, and residual excited LP techniques all offer this capability, and would be able to achieve a high synthesis quality similar to that of TD-PSOLA. Clearly formant synthesis also offers the potential of high quality speech, and could perform formant smoothing at concatenation boundaries, but, as discussed in Section 1.4.4, automatically estimating formant parameters from speech is problematic.

9.1.5 Optional Burst Release

It would be useful to devise some mechanism by which the system could determine, through training, whether or not bursts in different contexts should be released during synthesis. One possibility would be to incorporate such a mechanism into the existing decision trees, such that descending a burst tree could, in a particular context, terminate in a node associated with an unreleased burst. However, this extension would not be straightforward, since no labelling exists (or could exist) in the dictionary to enable questions to be asked about whether bursts are released or unreleased, and therefore both cases will always be able to occur in the same labelled context during synthesis. A possible solution might be the use of two Gaussians at every node in burst trees, to model both released and unreleased bursts, with the leaf nodes characterised during synthesis by the dominant Gaussian.

9.2 Inventory Size Reduction

As discussed in Section 8.5.5, the final system requires about 9MB of storage for the waveform inventory to be used during synthesis, which represents a large memory requirement for Personal Computers at the time of writing. This section discusses methods which could be used to reduce this requirement.

Section 8.5.5 also presented the results of some preliminary experiments to determine the effect on the quality of the synthetic speech of reducing the number of clustered states in the system. The results demonstrated that whilst large reductions in this number might be possible, such reductions would lead to the presence of a larger number of serious formant discontinuities during synthesis. However, these discontinuities could be avoided by adopting the segment selection algorithms described in Section 9.1.3, although this would require a larger number of segments to be stored for each state. The hope is that the superiority of the new system would enable a net saving in inventory size to be made, without degrading the quality of the synthetic speech. Note that the new system would be likely to achieve further savings due to the increased efficiency of storing adjacent segments, the use of which it would encourage. The increased efficiency arises because the extra speech which must be stored to the left and right of each segment, as described in Section 8.5.5, does not have to be stored between adjacent segments. However, the maximum possible saving, if all segments were adjacent, has been estimated at 31% for the M2 database, and therefore in practice savings would be quite small.

The waveform inventory size could, of course, be reduced by compressing the waveform to be stored. Lossless compression of the 16-bit 16kHz signal could be performed using standard compression routines, such as GNU's gzip, or better still an algorithm such as *Shorten*, (Robinson 1994), which exploits the redundancies of the speech signal to achieve larger compression factors. However, even with *Shorten*, lossless compression is not likely to exceed compression factors of about 2.5. Lossy coding offers higher compression factors, although some degradation of the synthetic speech may be noticeable. For example, a compression factor of 4 could be achieved very simply, by down-sampling to 8kHz, and using simple mulaw compression down to 8 bits per sample. Alternatively, higher quality lossy compression is available using *Shorten*, which can achieve compression factors of up to 5.3, by using as few as 3 bits per sample, with no perceptual degradation, (Robinson 1994).

Finally, using an alternative synthesis technique which does not require the storage of actual pieces of waveform could reduce the inventory size. However, this may result in poorer quality speech, since in general there is a tradeoff between the amount of storage required by a synthesis technique and the quality of the synthetic speech produced. The results obtained by Holmes described in Section 1.4.4 show that this does not necessarily have to be the case, but it probably is the case for most current automatic methods of encoding and re-synthesising speech. Nevertheless, slightly poorer speech quality may be acceptable in many applications, and then codebook excited, or multi-pulse, LP synthesis could be used to obtain substantial savings in inventory sizes. Simple LP synthesis could be used to obtain further savings if a larger reduction in speech quality was acceptable.

9.3 Other Future Possibilities

This section discusses other future possibilities connected with the work presented in this thesis, which have not been covered above.

9.3.1 Voice Transformation

As described in Section 8.5.7, the final system can be used to transform speech uttered by one person into speech with the same prosody which mimics the voice of another. Currently, knowledge of the words of the utterance (and their pronunciations) is required, to determine the clustered state sequence of the new speech. The words are currently specified manually, and the pronunciations determined automatically by using an HMM system trained on the source speaker to select pronunciations from a dictionary. In the future, speech recognition technology may improve sufficiently that the word sequence could be determined automatically. However, with current systems, an unacceptable number of recognition errors would be likely to occur, which would result in entire incorrect words being synthesised. An alternative approach, in which the words of the utterance are not required, is therefore currently more attractive. In this approach, a single set of decision trees would be used for both the source and target speakers. A direct one-to-one mapping would then exist between the clustered state sequence of the source speaker, and that required in synthesis for the target speaker. A recognition system would be used to determine the source state sequence. Although the accuracy of current recognition systems on such tasks is much lower than for word recognition, this approach has the advantage that entire incorrect words would not be synthesised. In fact, provided that recognition errors occurred principally with phonetically indistinct segments for which the equivalent segments in the target voice were equally phonetically indistinct, many recognition errors might have only a minor effect on the synthetic speech, especially since they would be produced with the correct prosody. Incorporating the ideas of Section 9.1.3 to encourage concatenation smoothness during synthesis would be likely to improve the synthetic speech quality. The advantage of this method is that any speech produced by the source speaker, including highly ungrammatical sentences, out-of-vocabulary words, and even non-words, could be converted into the new voice.

9.3.2 Voice Adaptation

The experiments described in Section 8.5.5 suggest that it might be possible to reduce the amount of training data used by the system to much less than the one hour used currently, particularly if the methods of Section 9.1.3 were used to encourage concatenation smoothness during synthesis. However, for the system to be able to construct reasonable models of phones in at least a few contexts, it seems unlikely that using less than a few minutes of training data will be possible using the current approach. To enable the system to acquire new voices with less data requires a different approach.

Modern speech recognition systems incorporate speaker adaptation systems which enable them to significantly improve their performance with a new speaker after exposure to only a few utterances of new speech. These adaptation systems either use speaker normalisation techniques to map the parameters of the new speaker towards those of the speaker(s) that the system was trained upon, or adjust the existing model parameters to better model the new speaker, (Leggetter and Woodland 1995). In synthesis it is desirable to adapt the speech produced by the system towards that of a new speaker, and therefore the model adaptation techniques are appropriate. With the LP version of the current synthesis system, adaptation techniques could be used to transform the LP parameters associated with each clustered state towards those appropriate for the new speaker. With the TD-PSOLA version of the system, each state is associated with a waveform segment, and in this case adaptation is therefore less straightforward. One possible solution which has been suggested, (Maes 1995), is to decompose each waveform segment into a set of LP parameters and an associated residual, adapt the LP parameters as just described, and then synthesise a new waveform segment using the new LP parameters and the old residual.

9.3.3 Speech Recognition

Chapter 6 described a number of methods were which were developed during the course of this work to improve the transcription, segmentation, and clustering of the training database. This section briefly discusses whether some of these methods could be usefully incorporated into automatic speech recognition systems.

Some of the largest improvements in synthesis obtained during the course of this work were due to the improved modelling of plosives. Both the use of multiple frame sizes and rates, and optionally released plosive models, considerably improved plosive transcription and segmentation, and hence synthesis performance. However, it is not clear that the improved models would necessarily bring similar improvements to speech recognition systems. The identities of many bursts are contained more in the formant transitions of neighbouring vowels than they are in the bursts themselves. Therefore, whilst better burst models might be of some use in unambiguously determining the presence of *some* burst, and hence constraining the search during recognition, they may not be very helpful in establishing burst identity. Furthermore, recognition systems usually use Gaussian mixture distributions, and can therefore model both released and unreleased plosives with the same model anyway.

The additional clustering questions introduced in Chapter 6 would be likely to bring some benefits to speech recognition systems, by enabling new acoustically important distinctions to be made during clustering. In fact, the use of word boundary information, and phonetic context two phones distant, has already been incorporated into the HTK large vocabulary speech recognition system, (Odell 1995).

9.4 Conclusion

A concatenative speech synthesis system has been developed which uses waveform segments representing the clustered states of a set of decision-tree state-clustered HMMs as its synthesis units. The system selects and segments these waveform segments entirely automatically from a single speaker continuous speech database. Duration and energy parameters are also estimated automatically from the database. The system can synthesise fluent, natural sounding, highly intelligible speech, in a monotone, from a word string specification of known pronunciation. The synthetic speech produced mimics the voice of the speaker used to record the training database. The segmental intelligibility has been measured using large scale Modified Rhyme Tests, and a very respectable error rate, of only 5.0% obtained. The system can be retrained on a new voice in less than 48 hours, and has been successfully trained on four voices.

The system developed achieves very respectable results, and thus demonstrates the validity of an HMM-based approach to speech synthesis. Many possible avenues exist for further development, and the approach therefore holds considerable potential for the future.

Appendix A

Modified Rhyme Test Answer Sheets

Figure A.1 shows the answer sheets used to perform the Modified Rhyme Tests described in Chapter 4. The word sequences used were obtained from (House et al. 1965) and formed six random paths through the words displayed on the answer sheets. Note that the words in each group on sheet 1 differ only in a final consonant, and those on sheet 2 only in an initial consonant.

Please listen to each word carefully and then put a line through
the word on the answer sheet which you think you heard.
If you are not sure, please guess.

1.	bass	bad	bath
	ban	back	bat
2.	bean	beat	beach
	beak	beam	bead
3.	buff	buck	bun
	bus	but	bug
4.	cake	cane	cave
	case	cape	came
5.	cub	cuss	cud
	cuff	cut	cup
6.	dig	din	dip
	did	dim	dill
7.	duck	dung	dug
	dud	dun	dub
8.	fin	fib	fit
	fig	fill	fizz
9.	heat	heath	heap
	heal	hear	heave
10.	kit	king	kid
	kin	kick	kill
11.	lace	lame	lay
	lane	late	lake
12.	mat	mass	man
	math	map	mad
13.	page	pace	pay
	pane	pave	pale
14.	pat	pass	path
	pack	pan	pad
15.	peal	peak	peace
	peat	peas	peach
16.	pit	pill	pin
	pig	pick	pip
17.	puck	pun	pus
	pup	pub	puff
18.	raze	ray	rave
	race	rake	rate
19.	save	sale	sane
	same	sake	safe
20.	sad	sap	sat
	sack	sag	sass
21.	seep	seek	seem
	seed	seethe	seen
22.	sing	sip	sill
	sick	sin	sit
23.	sun	sup	sub
	sung	sud	sum
24.	tap	tang	tan
	tab	tam	tack
25.	teach	teak	team
	teal	tear	tease

Figure A.1: Modified Rhyme Test answer sheet no. 1.

Please listen to each word carefully and then put a line through
the word on the answer sheet which you think you heard.
If you are not sure, please guess.

26.	led shed	red bed	wed fed	38.	win sin	tin pin	din fin
27.	fold cold	gold told	hold sold	39.	run bun	fun gun	nun sun
28.	fig dig	big rig	wig pig	40.	gang rang	bang hang	sang fang
29.	lick kick	sick pick	wick tick	41.	bent went	dent tent	rent sent
30.	took hook	book look	shook cook	42.	sip tip	hip rip	dip lip
31.	hark bark	dark mark	lark park	43.	shop top	mop cop	pop hop
32.	bale gale	sale tale	pale male	44.	beat meat	neat feat	seat heat
33.	reel peel	feel keel	heel eel	45.	bit wit	fit hit	kit sit
34.	bill hill	till will	kill fill	46.	lot not	hot pot	got tot
35.	soil foil	toil oil	boil coil	47.	test rest	vest nest	best west
36.	name same	game fame	tame came	48.	must rust	just bust	dust gust
37.	ten men	hen then	pen den	49.	raw jaw	saw thaw	law paw
				50.	day way	gay pay	may say

Figure A.1 (continued): Modified Rhyme Test answer sheet no. 2.

Appendix B

Speech Databases

Each of the speech databases used in this work comprised 592 sentences, or groups of short sentences, read from the first 43 pages of the novel *The Hitch Hiker's Guide to the Galaxy*, (Adams 1979). The sentences were spoken with natural prosody, at a normal read speech speaking rate. The data typically took about 4 hours to record, and this was usually done over 3 days. A Sennheiser HMD 414 head mounted microphone and a Symetrix SX202 Dual Mic Preamp were used to record the speech signal, and a Portable Laryngograph from Laryngograph Ltd. U.K. used to record the laryngograph trace. The signals were fed directly into the left and right line in sockets of a Silicon Graphics Iris R4400 Indigo computer, in which they were digitised by sampling at 16kHz and quantising into 16 bits per sample. The details of the databases are given below, together with an estimate of the average fundamental frequency of the speaker used in each.

M1 Recorded by the author, in the old CUED Speech Vision and Robotics (SVR) group quiet room, as used for the WSJ CAM 0 database. A speech only database occupying 118MB of disc-space. Average F_0 approximately 116Hz.

M2 Recorded by the author, in the new CUED SVR group quiet room, before the application of acoustic tiles, and thus in a somewhat reverberant, though otherwise quiet, environment. A speech and laryngograph database occupying 262MB of disc-space. Average F_0 approximately 116Hz.

M3 Recorded by Phil, in the new CUED SVR group quiet room, after the application of acoustic tiles. A speech and laryngograph database occupying 265MB of disc-space. Average F_0 approximately 99Hz.

F1 Recorded by Tina, in the old CUED SVR group quiet room, as above. A speech only database occupying 134MB of disc-space. Average F_0 approximately 210Hz.

F2 Recorded by Patricia, in the new CUED SVR group quiet room, after the application of acoustic tiles. A speech and laryngograph database occupying 271MB of disc-space. Average F_0 approximately 189Hz.

Appendix C

BEEP-0.6 Phone Set

As described in Section 5.2.1, various versions of the British English Example Pronunciations dictionary were used during the course of this work. The phone set of the most recent version to be used, BEEP-0.6, is shown in Table C.1.

Phone	Example	Phone	Example
aa	**a**fter	k	**c**at
ae	s**a**ck	l	**l**eg
ah	b**u**g	m	**m**ouse
ao	b**al**l	n	**n**est
aw	all**ow**	ng	ki**ng**
ax	**a**gain	oh	g**o**d
ay	**eye**	ow	wind**ow**
b	**b**ack	oy	t**oy**
ch	**ch**urch	p	**p**ine
d	**d**og	r	**r**ake
dh	**th**en	s	**s**ea
ea	**air**	sh	**sh**ell
eh	g**e**m	t	**t**able
er	b**ir**d	th	**th**eatre
ey	pr**ey**	ua	sumpt**uou**s
f	**f**ire	uh	f**oo**t
g	**g**old	uw	tr**ue**
hh	**h**ouse	v	**v**an
ia	am**ia**ble	w	**w**indow
ih	p**i**g	y	**y**ak
iy	**ee**l	z	**z**oo
jh	**j**udge	zh	lei**s**ure

Table C.1: The BEEP-0.6 phone set.

Appendix D

Linear Prediction Theory

This appendix presents the underlying mathematics of Linear Prediction (LP) theory, the justification for the method used to estimate LP coefficients from multiple segments in Section 5.3.4, and the derivation of the distance measure used for unit selection in the I-method of Section 5.5.2. It does not describe the details of the algorithms used to solve for the LP coefficients, the relationship between LP coefficients and other related parameters, such as reflection coefficients or log-area coefficients, lattice filters, or the spectral interpretation of LP coefficient estimation. For a description of these aspects of LP theory see (Markel and Gray 1976), or any good speech processing textbook, for example (Parsons 1986).

D.1 Basic LP Theory

Let the speech samples in a frame be represented by $y(n)$, where $1 \leq n \leq N$. The basis of Linear Prediction theory is then to assume that each speech sample $y(n)$ can be approximately predicted as a linear combination of the previous P samples of speech,

$$\hat{y}(n) = -\sum_{i=1}^{P} a(i) y(n-i), \tag{D.1}$$

where $a(i)$ are the *Linear Prediction Coefficients* of the frame, and P the *Linear Prediction Order*. Let the difference between the predicted sample and the actual sample be $\sigma e(n)$, where σ is a scaling factor introduced to make the error signal $e(n)$ have an r.m.s. value of 1. Thus,

$$\begin{aligned} \sigma e(n) &= y(n) - \hat{y}(n) && \text{(D.2)} \\ &= y(n) + \sum_{i=1}^{P} a(i) y(n-i) && \text{(D.3)} \\ &= \sum_{i=0}^{P} a(i) y(n-i) \qquad \text{where } a(0) = 1. && \text{(D.4)} \end{aligned}$$

The LP coefficients are found by minimising the sum of the squared error terms, over some range of n. This is the point at which LP theory diverges into two slightly different approaches, known as the *covariance approach*, and the *autocorrelation approach*. In the former, the minimisation is conducted only over those values of $e(n)$ which can be properly calculated using the values of $y(n)$ in the frame, for which $1 \leq n \leq N$. In the latter, the minimisation is conducted over all values of $e(n)$ over all time. This can be done if the speech signal is multiplied by a smooth windowing function, such as a Hamming window, so that $y(n)$ is non-zero only for $1 \leq n \leq N$, since then the error signal $e(n)$ is non-zero only for $1 \leq n \leq N+P$. This will be the approach taken in this appendix; the covariance approach will not be considered further.

The autocorrelation approach therefore minimises E, defined by

$$E = \sum_{n=-\infty}^{\infty} [e(n)\sigma]^2 \tag{D.5}$$

$$= \sum_{n=1}^{N+P} [e(n)\sigma]^2 \qquad \text{which, substituting equation D.4 is,} \tag{D.6}$$

$$= \sum_{n=1}^{N+P} [\sum_{i=0}^{P} a(i)y(n-i)]^2. \tag{D.7}$$

The minimum can be found by setting the differentials of E with respect to each $a(j)$,

$$\frac{\partial E}{\partial a(j)} = \sum_{n=1}^{N+P} 2[\sum_{i=0}^{P} a(i)y(n-i)]y(n-j) \qquad 1 \leq j \leq P, \tag{D.8}$$

to zero. Paying careful attention to indices, this becomes

$$\sum_{i=0}^{P} a(i)r_y(i-j) = 0 \qquad 1 \leq j \leq P, \tag{D.9}$$

where the *autocorrelation function* $r_y(i)$ is defined by

$$r_y(i) = \sum_{n=1}^{N-i} y(n)y(n+i). \tag{D.10}$$

Equation D.9 represents a set of P equations in P unknowns, and can therefore be solved using normal linear equation methods, or more efficiently by an algorithm known as Levinson's or Durbin's recursion. The details of these methods are not described in this Appendix; for further information see the references mentioned above.

By expanding equation D.7 to

$$E = \sum_{i=0}^{P} a(i) \sum_{j=0}^{P} a(j)r_y(i-j). \tag{D.11}$$

and then substituting in equation D.9, E can be calculated as

$$E = \sum_{i=0}^{P} a(i)r_y(i). \tag{D.12}$$

The only remaining unknown is σ. This quantity is defined to be such that the r.m.s. value of $e(n)$ is 1. However, it is not clear over which range of n the r.m.s. calculation should be performed. In this work the range was taken to be $1 \leq n \leq N+P$ to enable the calculation of σ to be integrated with the maths presented so far. The definition is therefore

$$\sqrt{\frac{\sum_{n=1}^{N+P}[e(n)]^2}{N+P}} = 1, \tag{D.13}$$

which, by re-arranging and substituting equation D.6, becomes

$$\sigma^2 = \frac{E}{N+P}. \tag{D.14}$$

Furthermore, note that the mean of the error signal, $\bar{e}$ is given by

$$\bar{e} = \frac{\sum_{n=1}^{N+P} e(n)}{N+P} \tag{D.15}$$

$$= \frac{\sum_{n=1}^{N+P} \sum_{i=0}^{P} a(i)y(n-i)}{\sigma(N+P)} \tag{D.16}$$

$$= \frac{\sum_{i=0}^{P} a(i) \sum_{n=1}^{N} y(n)}{\sigma(N+P)} \tag{D.17}$$

which is zero if the mean of the speech signal over the frame is zero. Thus, zero-meaning the speech frame after Hamming windowing leads to an error signal with zero mean and unity variance.

Finally, the all-pole nature of the LP model can be demonstrated by re-arranging the Z-Transform of equation D.3 to obtain the model's transfer function,

$$\frac{Y(z)}{E(z)} = \frac{\sigma}{1 + \sum_{i=1}^{P} a(i) z^{-i}}. \tag{D.18}$$

D.2 Estimation from Multiple Segments

When the single speech frame used above is replaced by multiple frames, a single set of LP coefficients can be found which best represents the pooled speech. Let the multiple frames be identified by the superscript (f), where $1 \leq f \leq F$. Equation D.5 then becomes

$$E = \sum_{f=1}^{F} \sum_{n=-\infty}^{\infty} [e^{(f)}(n)\sigma^{(f)}]^2 \tag{D.19}$$

$$= \sum_{f=1}^{F} \sum_{n=1}^{N^{(f)}+P} [e^{(f)}(n)\sigma^{(f)}]^2 \tag{D.20}$$

$$= \sum_{f=1}^{F} \sum_{n=1}^{N^{(f)}+P} [\sum_{i=0}^{P} a(i)y^{(f)}(n-i)]^2. \tag{D.21}$$

Differentiating as before, gives

$$\frac{\partial E}{\partial a(j)} = \sum_{f=1}^{F} \sum_{n=1}^{N^{(f)}+P} 2[\sum_{i=0}^{P} a(i)y^{(f)}(n-i)]y^{(f)}(n-j) \qquad 1 \leq j \leq P, \tag{D.22}$$

which, when set to zero, can be written as

$$\sum_{i=0}^{P} a(i) r_y^{(f)}(i-j) = 0 \qquad 1 \leq j \leq P, \tag{D.23}$$

which is analogous to equation D.9, and can be solved for $a(i)$. The $r_y^{(f)}(i)$ term is defined by

$$r_y^{(f)}(i) = \sum_{f=1}^{F} \sum_{n=1}^{N^{(f)}-i} y^{(f)}(n) y^{(f)}(n+i), \tag{D.24}$$

and is the autocorrelation function of equation D.10 extended over multiple frames. Thus, in the multiple frame case, the LP coefficients are calculated in the same way as for the single frame case, but using autocorrelation coefficients calculated from all the frames involved.

D.3 LP Distance Measure for Unit Selection

It has been shown that the error signal $e(n)$ has zero mean and unity variance, provided the speech frame it is associated with also has a zero mean. If it is *assumed* that each sample of the error signal, $e(n)$, is distributed as an independent Gaussian random variable, then a useful log-likelihood based distance measure, between a segment of speech and a given LP vector, can be obtained. The discussion given here is similar to that presented in (Juang 1984).

Let $\mathbf{x} = x(n), 1 \leq n \leq N$ represent the gain normalised speech signal, defined by

$$x(n) = y(n)/\sigma \tag{D.25}$$

where σ is calculated as in equation D.14. The $\mathbf{x}$ signal is used in the following derivation in order that the distance measure derived depends only on the spectral properties of the speech signal, and not on its amplitude. Also, let $\mathbf{m} = m(i), 0 \leq i \leq P$, be some set of LP coefficients not calculated directly from the speech signal. The signal $\mathbf{x}$ could be generated from a model with coefficients $\mathbf{m}$ using the equation

$$x(n) = -\sum_{i=1}^{P} m(i) x(n-i) + e(n) \tag{D.26}$$

if both $e(n), 1 \leq n \leq N$ and $x(n), 1-P \leq n \leq 0$ all took appropriate values. Therefore, the likelihood of the speech signal $\mathbf{x}$ being generated given a model with coefficients $\mathbf{m}$, is the same as the likelihood of these appropriate values occurring, that is,

$$\Pr(\mathbf{x}|\mathbf{m}) = \Pr[e(n), 1 \leq n \leq N; x(n), 1-P \leq n \leq 0]. \tag{D.27}$$

At this point an approximation is made in which one set of end effects is ignored and another set introduced, giving

$$\Pr(\mathbf{x}|\mathbf{m}) \approx \Pr[e(n), 1 \leq n \leq N+P]. \tag{D.28}$$

The approximation can be justified since in general $N \gg P$. If it is *assumed* that each sample of the error signal is distributed as an independent Gaussian random variable, with zero mean and unity variance, then the likelihood of a sample taking a particular value $e(n)$, is given by

$$\Pr[e(n)] = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}[e(n)]^2}. \tag{D.29}$$

The independence assumption also means that the right hand side of equation D.28 can be factorised to give

$$\Pr(\mathbf{x}|\mathbf{m}) \approx \Pr[e(1)] \Pr[e(2)] \ldots \Pr[e(N+P)], \tag{D.30}$$

and hence, by substituting equation D.29,

$$\Pr(\mathbf{x}|\mathbf{m}) \approx (\frac{1}{\sqrt{2\pi}})^{(N+P)} e^{-\frac{1}{2}\sum_{n=1}^{N+P}[e(n)]^2}. \tag{D.31}$$

By taking logs, and rearranging and substituting equation D.26, equation D.31 can be expanded to

$$\ln[\Pr(\mathbf{x}|\mathbf{m})] \approx -\frac{N+P}{2}\ln(2\pi) - \frac{1}{2\sigma^2}\sum_{i=0}^{P} m(i) \sum_{j=0}^{P} m(j) r_y(i-j). \tag{D.32}$$

By defining an autocorrelation function over model parameters, $r_m(i)$, as

$$r_m(i) = \sum_{k=0}^{P-i} m(k) m(k+i), \tag{D.33}$$

it can be shown by algebraic expansion that equation D.32 is equivalent to

$$\ln[\Pr(\mathbf{x}|\mathbf{m})] \approx -\frac{N+P}{2}\ln(2\pi) - \frac{1}{2\sigma^2}[r_y(0) r_m(0) + 2\sum_{i=1}^{P} r_y(i) r_m(i)]. \tag{D.34}$$

Furthermore, by expanding σ^2 using equation D.14 and equation D.11, it can be shown that

$$\sigma^2 = \frac{1}{N+P}[r_y(0) r_a(0) + 2\sum_{i=1}^{P} r_y(i) r_a(i)], \tag{D.35}$$

where $r_a(i)$ is defined similarly to $r_m(i)$.

The distance measure used in the work described in Section 5.5.2 was required to compare many speech segments of equal length to a single LP vector estimated from a pool of segments. The comparison required was purely spectral, since the speech synthesised using the LP coefficients of the selected segment was scaled to the average short term energy per sample of the clustered state in question during synthesis anyway; hence the derivation of $\Pr(\mathbf{x}|\mathbf{m})$ and not $\Pr(\mathbf{y}|\mathbf{m}, \sigma^2)$. With segments of equal length the first term in equation D.34 is always the same, and so the distance measure actually computed was

$$-\frac{1}{2\sigma^2}[r_y(0)r_m(0) + 2\sum_{i=1}^{P} r_y(i)r_m(i)]. \tag{D.36}$$

Finally note that this distance measure is closely related to the Itakura-Saito distance measure, as discussed in (Juang 1984).

Appendix E

Audio Examples

The accompanying compact disc contains numerous speech examples recorded as stereo audio files at a sampling rate of 44.1kHz, which can be played on normal Hi-Fi equipment. The numbers associated with each of the speech examples described in this appendix correspond to track numbers on the compact disc. This appendix first presents examples of the results obtained with the basic synthesis system, as described in Chapter 5, in Section E.1. Section E.2 presents a demonstration of the performance of the TD-PSOLA implementation, as described in Chapter 7. Finally, Section E.3 presents examples of the performance of, and experiments with, both the LP and TD-PSOLA versions of the final system, as described in Chapter 8.

In general, the words used in each speech example are given in the relevant section. The only exception is the sentence fragment

- *When a sailor in a small craft faces the might of the vast Atlantic Ocean today,...*

which is used throughout this appendix, and is referred to simply as the *sailor* fragment. Note that, unless otherwise stated, all synthetic speech (other than re-synthesised natural speech) was produced using a duration scaling factor of 0.1 standard deviations (see Section 5.3.1), and at the average fundamental frequency of the speaker used in the relevant database (see Appendix B). The first track on the compact disc is a brief introduction,

- Example 01 : A natural speech recording of the sentence *This compact disc belongs with a copy of the dissertation submitted by Robert Donovan for consideration for the degree of Doctor of Philosophy at the University of Cambridge*

E.1 Basic System

The speech examples in this section were generated using various versions of the basic system to synthesise the *sailor* fragment. The results of performing LP analysis-synthesis on a natural version of this utterance (available as speech example 36) are also presented here.

- Example 02 : The F-method version of the basic system, trained on the M1 database, synthesising the *sailor* fragment.

- Example 03 : The C-method version of the basic system, trained on the M1 database, synthesising the *sailor* fragment.
- Example 04 : The I-method version of the basic system, trained on the M1 database, synthesising the *sailor* fragment.
- Example 05 : The P-method version of the basic system, trained on the M1 database, synthesising the *sailor* fragment.
- Example 06 : The basic system using interpolated P-method reflection coefficients, trained on the M1 database, synthesising the *sailor* fragment.
- Example 07 : LP analysis-synthesis of Example 36.

E.2 TD-PSOLA Demonstration

The speech examples in this section provide a demonstration of the performance of the TD-PSOLA implementation made during the course of this work. Examples of analysis-synthesis are presented in which either the fundamental frequency or the duration is scaled up or down by a constant factor. The original speech, and otherwise unaltered re-synthesised speech, is also presented. See Section 7.3.2 for further details. The sentence used throughout is

- *Far out in the uncharted backwaters of the unfashionable end of the western spiral arm of the galaxy lies a small unregarded yellow sun.*,

taken from either the M2 or F2 database. In the following, the numbers on the left refer to male speech examples, and the numbers on the right to female speech examples.

- Example 08/20 : Original speech.
- Example 09/21 : Re-synthesised, but otherwise unaltered.
- Example 10/22 : Fundamental frequency raised by a factor of 1.2.
- Example 11/23 : Fundamental frequency lowered by a factor of 1.2.
- Example 12/24 : Fundamental frequency raised by a factor of 1.5.
- Example 13/25 : Fundamental frequency lowered by a factor of 1.5.
- Example 14/26 : Fundamental frequency raised by a factor of 2.0.
- Example 15/27 : Fundamental frequency lowered by a factor of 2.0.
- Example 16/28 : Duration reduced by a factor of 1.5.
- Example 17/29 : Duration raised by a factor of 1.5.
- Example 18/30 : Duration reduced by a factor of 2.0.
- Example 19/31 : Duration raised by a factor of 2.0.

E.3 Final System

This section first presents speech examples generated from the LP version of the final system, in Section E.3.1, and an early version of the TD-PSOLA implementation, in Section E.3.2. It then presents speech examples generated using the full TD-PSOLA version of the final system, in Section E.3.3. Finally, the results of experiments into inventory size reduction and voice transformation are presented in Sections E.3.4, and E.3.5 respectively.

E.3.1 The LP Version of the Final System

The speech example in this section demonstrates the performance of the LP version of the final system, as described in Section 8.4. The resulting speech should be compared with that generated by the P-method version of the basic system, given in speech example 05.

- Example 32 : The LP version (P-method) of the final system, trained on the M2 database, synthesising the *sailor* fragment.

E.3.2 An Early Version of the TD-PSOLA Implementation

This section presents synthetic speech generated using an early version of the TD-PSOLA implementation, in which each voiced state was synthesised as a sequence of identical pitch periods. See Section 8.5.1 for further details. This speech should be compared to speech examples 35 and 43 respectively, which were generated using the full TD-PSOLA implementation.

- Example 33 : An early TD-PSOLA system, trained on the M1 database, synthesising the *sailor* fragment.
- Example 34 : An early TD-PSOLA system, trained on the F1 database, synthesising the *sailor* fragment.

E.3.3 The TD-PSOLA Version of the Final System

This section presents speech examples generated using the full TD-PSOLA version of the final system, trained on the M2, F1, F2, and M3 databases.

M2 database

The M2 database was used extensively during system development, and therefore cannot be considered to be a test database for the system.

- Example 35 : The TD-PSOLA version of the final system, trained on the M2 database, synthesising the *sailor* fragment.
- Example 36 : A natural speech recording of the *sailor* fragment, made by the speaker used in the M1 & M2 databases.

- Example 37 : The TD-PSOLA version of the final system, trained on the M2 database, synthesising *This speech was synthesised by a speech synthesis system developed by Rob Donovan at Cambridge University Engineering Department. It uses the clustered states of a set of decision tree state clustered hidden Markov models, as it's subword units. Speech is synthesised by concatenating representative waveform segments of each of the clustered states. The system can be retrained in less than forty eight hours on a new voice, and could easily be adapted to a new language.*

- Example 38 : The TD-PSOLA version of the final system, trained on the M2 database, synthesising *This is an example of the best speech produced by the system to date. It was trained on text read from a novel, namely the Hitch Hiker's Guide to the Galaxy. The Modified Rhyme Test error for this speech is only five point zero percent. I must apologise for speaking on a monotone, but you see, I have no brain!*

- Example 39 : The TD-PSOLA version of the final system, trained on the M2 database, synthesising *She sells sea shells on the sea shore.*, slowly, with a duration scaling factor of 0.3 standard deviations.

- Example 40 : The TD-PSOLA version of the final system, trained on the M2 database, synthesising *She sells sea shells on the sea shore.*, quickly, with a duration scaling factor of -0.8 standard deviations.

- Example 41 : The TD-PSOLA version of the final system, trained on the M2 database, synthesising *This speech example is included to demonstrate some of the problems still present. Pick. Pig. Sack. Sag. What's the difference? Perfect plosives, pronounced properly, too properly in fact. Waveform segments are concatenated very carefully, but sometimes the durations are a bit strange. Occasionally it's as if someone else is talking in the background. Sometimes it sounds like the speaker is burbling. Sometimes segmentation errors still cause problems.* The isolated words in this example were synthesised using a duration scaling factor of 0.5 standard deviations, with the rest of the example using a factor of 0.1.

- Example 42 : The TD-PSOLA version of the final system, trained on the M2 database, synthesising *My name's Rob Donovan, and I'm from Cambridge University.*, using a stylised pitch track and a duration scaling factor of -0.1 standard deviations.

F1 Database

This database did not include a laryngograph signal. It was used occasionally during the later stages of system development, but only for demonstration purposes. Therefore, it can be viewed as a test database for the system.

- Example 43 : The TD-PSOLA version of the final system, trained on the F1 database, synthesising the *sailor* fragment.

- Example 44 : A natural speech recording of the *sailor* fragment, made by the speaker used in the F1 database.

- Example 45 : The TD-PSOLA version of the final system, trained on the F1 database, synthesising *This speech was produced by training the system on a female database, kindly recorded by Tina Burrows. The system was trained without access to a laryngograph signal, hence the hoarseness.*

F2 Database

This database was not used during system development, and therefore represents a test database for the system.

- Example 46 : The TD-PSOLA version of the final system, trained on the F2 database, synthesising the *sailor* fragment.

- Example 47 : A natural speech recording of the *sailor* fragment, made by the speaker used in the F2 database.

- Example 48 : The TD-PSOLA version of the final system, trained on the F2 database, synthesising *This speech was produced by training the system on a female database, kindly recorded by Patricia. This database did include a laryngograph signal.*

- Example 49 : The TD-PSOLA version of the final system, trained on the F2 database, synthesising *The rain in Spain stays mainly on the plain.*

- Example 50 : The TD-PSOLA version of the final system, trained on the F2 database, synthesising *I don't normally talk in a monotone you know! Rob made me do it!*

M3 Database

This database was not used during system development, and therefore represents a test database for the system.

- Example 51 : The TD-PSOLA version of the final system, trained on the M3 database, synthesising the *sailor* fragment.

- Example 52 : A natural speech recording of the *sailor* fragment, made by the speaker used in the M3 database.

- Example 53 : The TD-PSOLA version of the final system, trained on the M3 database, synthesising *This speech was produced by training the system on a male database, kindly recorded by Phil Woodland. Again, a laryngograph signal was available.*

- Example 54 : The TD-PSOLA version of the final system, trained on the M3 database, synthesising *HTK is the name of a hidden Markov model tool kit developed at Cambridge University Engineering Department. It is now available through Entropic Cambridge Research Laboratory, and Entropic Research Laboratory, Washington.*

E.3.4 Inventory Size Experiments

The speech examples presented in this section were generated using the TD-PSOLA version of the final system, but with a reduced number of states. See Section 8.5.5 for further details.

- Example 55 : The TD-PSOLA version of the final system trained on the whole M2 database, with the minimum number of occurrences per leaf node clustering threshold set to 30, synthesising the *sailor* fragment.
- Example 56 : The TD-PSOLA version of the final system trained on only the first half of the M2 database, with the minimum number of occurrences per leaf node clustering threshold left at 12, synthesising the *sailor* fragment.

E.3.5 Voice Transformation Experiments

The speech examples in this section were synthesised using prosody transplanted from a natural version of the same utterance. The examples are presented in pairs, with the voices used given in brackets as database mnemonics. In each case, the number on the left refers to the natural speech, and the number on the right the synthetic speech, which was synthesised using the TD-PSOLA version of the final system trained on the appropriate database. See Section 8.5.7 for further details.

- Example 57/58 : Natural (M1 & M2) and synthetic (M1 & M2) versions of the poem fragment

  Water, Water, every where,
  And all the boards did shrink;
  Water, water, every where,
  Nor any drop to drink.

- Example 59/60 : Natural (M1 & M2) and synthetic (M3) versions of the poem fragment

  I wandered lonely as a cloud
  That floats on high o'er vales and hills,
  When all at once I saw a crowd,
  A host, of golden daffodils;

- Example 61/62 : Natural (F1) and synthetic (M1 & M2) versions of the sentence *In the beginning was the word, and the word was with God, and the word was God.*
- Example 63/64 : Natural (M1 & M2) and synthetic (F2) versions of the sentence *At last, the end!*

Bibliography

Adams, D. (1979) *The Hitch Hiker's Guide to the Galaxy*, Pan Books, London.

Allen, J., Hunnicutt, M.S., and Klatt, D. (1987) *From Text to Speech: The MITalk System*, Cambridge University Press, Cambridge.

Atal, B.S., and Remde, J.R. (1982) A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates, *Proc. ICASSP'82, Paris*, pp. 614–617.

Bahl, L.R., Bakis, R., Cohen, P.S., Cole, A.G., Jelinek, F., Lewis, B.L., and Mercer, R.L. (1980) Further Results on the Recognition of a Continuously Read Natural Corpus, *Proc. ICASSP'80, Denver*, pp. 872–874.

Bahl, L.R., Bakis, R., Bellegarda, J., Brown, P.F., Burshtein, D., Das, S.K., de Souza, P.V., Gopalakrishnan, P.S., Jelinek, F., Kanevsky, D., Mercer, R.L., Nadas, A.J., Nahamoo, D., Picheny, M.A. (1989) Large Vocabulary Natural Language Continuous Speech Recognition, *Proc. ICASSP'89, Glasgow*, pp. 465–467.

Bahl, L.R., de Souza, P.V., Gopalakrishnan, P.S., Nahamoo, D., Picheny, M.A. (1991) Decision Trees for Phonological Rules in Continuous Speech, *Proc. ICASSP'91, Toronto*, pp. 185–188.

Baum, L.E., Petrie, T., Soules, G., and Weiss, N. (1970) A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains, *Ann. Math. Stat.*, Vol. 41, no. 1, pp. 164–171.

Bellcore (1996) URL: http://www.bellcore.com/demotoo/ORATOR/index.html

Bigorgne, D., Boeffard, O., Cherbonnel, B., Emerard, F., Larreur, D., Le Saint-Milon, J.L., Metayer, I., Sorin, C., and White, S. (1993) Multilingual PSOLA Text-to-Speech System, *Proc. ICASSP'93, Minneapolis*, Vol. 2. pp. 187–190.

Black, A.W., and Campbell, N. (1995) Optimising Selection of Units from Speech Databases for Concatenative Synthesis, *Proc. Eurospeech'95, Madrid*, pp. 581–584.

Boeffard, O., Miclet, S., and White, S. (1992) Automatic Generation of Optimized Unit Dictionaries for Text-to-Speech Synthesis, *Proc. ICSLP'92, Banff*, pp. 1211–1214.

Boeffard, O., Cherbonnel, B., Emerard, F., and White, S. (1993) Automatic Segmentation and Quality Evaluation of Speech Unit Inventories for Concatenation-Based, Multilingual PSOLA Text-to-Speech Systems, *Proc. Eurospeech'93, Berlin*, pp. 1449–1452.

Borden, G.J., and Harris, K.S. (1984) *Speech Science Primer*, 2nd edition, Williams & Wilkins, Baltimore.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984) *Classification and Regression Trees*, Wadsworth & Brooks, Pacific Grove, California.

Browman, C.P. (1980) Rules for Demisyllable Synthesis using Lingua, a Language Interpreter, *Proc. ICASSP'80, Denver*, pp. 561–564.

Bruckert, E., Minow, M., and Tetschner, W. (1983) Three-Tiered Software and VLSI Aid Developmental System to Read Text Aloud, *Electronics* , 56, pp. 133–138.

Brugnara, F., Falavigna D., and Omologo, M. (1992) A HMM-based System for Automatic Segmentation and Labelling of Speech, *Proc. ICSLP'92, Banff*, pp. 803–806.

Campbell, W.N. (1992) Syllable-Based Segmental Duration, *in*, Bailly, G., and Benoit, C., (eds.), *Talking Machines, Theories, Models, and Designs*, North-Holland, Elsevier Science Publishers, Amsterdam, pp. 211–224.

Charpentier, F.J., and Stella, M.G. (1986) Diphone Synthesis Using an Overlap-Add Technique for Speech Waveforms Concatenation, *Proc. ICASSP'86, Tokyo*, pp. 2015–2018.

Coker, C.H. (1976) A Model of Articulatory Dynamics and Control, *Proc. IEEE*, Vol. 64, No. 4, pp. 452–460.

Coker, C.H., Church, K.W., and Liberman, M.Y. (1990) Morphology and Rhyming: Two powerful Alternatives to Letter-to-Sound Rules for Speech Synthesis, *Proc. ESCA Workshop on Speech Synthesis, Autrans, Grenoble*, pp. 83–86.

Courbon, J.L., and Emerard, F. (1982) SPARTE: A Text-to-Speech Machine Using Synthesis by Diphones, *Proc. ICASSP'82, Paris*, pp. 1597–1600.

Cruttenden, A. (1986) *Intonation*, Cambridge University Press, Cambridge.

Deller, J.R., Proakis, J.G., and Hansen, J.H.L. (1993) *Discrete-Time Processing of Speech Signals*, Macmillan, New York.

Dixon, N.R., and Maxey, H.D. (1968) Terminal Analog Synthesis of Continuous Speech Using the Diphone Method of Segment Assembly, *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-16, No. 1, pp. 40–50.

Donovan, R.E., and Woodland, P.C. (1995a) Automatic Speech Synthesiser Parameter Estimation using HMMs, *Proc. ICASSP'95, Detroit*, pp. 640–643.

Donovan, R.E., and Woodland, P.C. (1995b) Improvements in an HMM-Based Speech Synthesiser, *Proc. Eurospeech'95, Madrid*, pp. 573–576.

Downey, S., and Russell, M.J. (1992) A Decision Tree Approach to Task Independent Speech Recognition, *Proc. Institute of Acoustics*, Vol. 14, Part 6, pp. 181–188.

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.

Egan, J.P. (1948) Articulation Testing Methods, *Laryngoscope*, 58, pp. 955–991.

Falaschi, A., Giustiniani, M. and Verola, M. (1989) A Hidden Markov Model Approach to Speech Synthesis, *Proc. Eurospeech'89, Paris*, pp. 187–190.

Fallside, F., and Young, S.J. (1978) Speech Output From a Computer-Controlled Water-Supply Network, *Proc. IEE*, Vol. 125, No. 2, pp. 157–161.

Farges, E.P. and Clements, M.A. Hidden Markov Models Applied to Very Low Bit Rate Speech Coding, *Proc. ICASSP'86, Tokyo*, pp. 433–436.

Giachin, E.P., Rosenberg, A.E., and Lee, C-H. (1991) Word Juncture Modeling Using Phonological Rules for HMM-based Continuous Speech Recognition, *Computer Speech and Language*, Vol. 5, No. 2, pp. 155–168.

Giustiniani, M. and Pierucci, P. (1991) Phonetic Ergodic HMM for Speech Synthesis, *Proc. Eurospeech'91, Genova*, pp. 349–352.

Groner, G.F., Bernstein, J., Ingber, E., Pearlman, J., and Toal, T. (1982) A Real-Time Text-to-Speech Converter, *Speech Technology*, 1, pp. 73–76.

't Hart, J., Collier R., and Cohen, A. (1990) *A Perceptual Study of Intonation*, Cambridge University Press, Cambridge.

Hauptmann A.G. (1993) SpeakEZ: A First Experiment In Concatenation Synthesis From A Large Corpus, *Proc. Eurospeech'93, Berlin*, pp. 1701–1704.

Hess W. (1983) *Pitch Determination of Speech Signals*, Springer-Verlag, Berlin.

Hirose K., and Fujisaki, H. (1982) Analysis and Synthesis of Voice Fundamental Frequency Contours of Spoken Sentences, *Proc. ICASSP'82, Paris*, pp. 950–953.

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. (1973) The Influence of Glottal Waveform on the Naturalness of Speech from a Parallel Formant Synthesizer, *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-21, No. 3, pp. 298–305.

House, A.S., Williams, C.E., Hecker, M.H.L., and Kryter, K.D. (1965) Articulation-Testing Methods: Consonantal Differentiation with a Closed-Response Set, *Journal of the Acoustical Society of America*, Vol. 37, No. 1, pp. 158–166.

Hwang, M-Y., and Huang, X. (1992) Subphonetic Modeling With Markov States — Senone, *Proc. ICASSP'92, San Francisco*, pp. I-33–I-36.

Hwang, M-Y., Huang, X., and Alleva, F. (1993) Predicting Unseen Triphones with Senones, *Proc. ICASSP'93, Minneapolis*, Vol. 2, pp. 311–314.

IEEE (1969) IEEE Recommended Practice for Speech Quality Measurements, *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-17, No. 3, pp. 225–246.

Itoh, K., Nakajima, S., and Hirokawa, T. (1994) A New Waveform Speech Synthesis Approach Based on the COC Speech Spectrum, *Proc. ICASSP'94, Adelaide*, Vol. 1, pp. 577–580.

Jelinek, F., and Mercer, R.L. (1980) Interpolated Estimation of Markov Source Parameters from Sparse Data, *in*, Gelsema, E.S., and Kanal, L.N., (eds.), *Pattern Recognition in Practice*, North-Holland, Elsevier Science Publishers, Amsterdam, pp. 381–397.

Jones, M. (1994) *The Use of Acoustic-Level Prosodics in Large Vocabulary Speech Recognition*, PhD. Thesis, Cambridge University Engineering Department.

Juang, B-H. (1984) On the Hidden Markov Model and Dynamic Time Warping for Speech Recognition — A Unified View, *AT&T Bell Laboratories Technical Journal*, Vol. 63, No. 7, pp. 1213–1243.

Juang, B-H. (1985) Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains, *AT&T Bell Laboratories Technical Journal*, Vol. 64, No. 6, pp. 1235–1249.

Kelly, J., and Gerstman, L. (1961) An Artificial Talker Driven from a Phonetic Input, *Journal of the Acoustical Society of America*, Vol. 33, Suppl. No. 1, pp. 835. (abstract)

Klatt, D.H. (1979) Synthesis by Rule of Segmental Durations in English Sentences, *in*, Lindblom, B., and Ohman, S., (eds.), *Frontiers of Speech Communication Research*, Academic Press, London, pp. 287–299.

Klatt, D.H. (1980) Software for a Cascade/Parallel Formant Synthesiser, *Journal of the Acoustical Society of America*, Vol. 67, No. 3, pp. 971–995.

Klatt, D.H. (1982) The Klattalk Text-to-Speech Conversion System, *Proc. ICASSP'82, Paris*, pp. 1589–1592.

Klatt, D.H. (1987) Review of Text-to-Speech Conversion for English, *Journal of the Acoustical Society of America*, Vol. 82, No. 3, pp. 737–793.

Klavans, J.L., and Tzoukermann, E. (1994) Inducing Concatenative Units from Machine Readable Dictionaries and Corpora for Speech Synthesis, *Proc. ICSLP'94, Yokohama*, pp. 1755–1758.

Lee, K-F. (1990) Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-38, No. 4, pp. 599–609.

Lee, K-F, Hayamizu, S., Hon, H-W., Huang, C., Swartz, J., and Weide, R. (1990) Allophone Clustering for Continuous Speech Recognition, *Proc. ICASSP'90, Albuquerque*, pp. 749–752.

Leggetter, C.J., and Woodland, P.C. (1995) Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models, *Computer Speech and Language*, Vol. 9, No. 2, pp. 171–185.

Lernout & Hauspie (1996) URL: http://www.lhs.com/tts.html

Linggard, R. (1985) *Electronic Synthesis of Speech*, Cambridge University Press, Cambridge.

Liporace, L.A. (1982) Maximum Likelihood Estimation for Multivariate Observations of Markov Sources, *IEEE Transactions on Information Theory*, Vol. IT-28, no. 5, pp. 729–734.

Ljolje, A, and Riley, M.D. (1991) Automatic Segmentation and Labeling of Speech, *Proc. ICASSP'91, Toronto*, pp. 473–476.

Ljolje, A, and Riley, M.D. (1993) Automatic Segmentation of Speech for TTS, *Proc. Eurospeech'93, Berlin*, pp. 1445–1448.

Logan, J.S., Greene, B.G., and Pisoni, D.B. (1989) Segmental Intelligibility of Synthetic Speech Produced by Rule, *Journal of the Acoustical Society of America*, Vol. 86, No. 2, pp. 566–581.

Lovins, J.B., Macchi, M.J., and Fujimura, O. (1979) A Demisyllable Inventory for Speech Synthesis, *Journal of the Acoustical Society of America*, Vol. 65, Suppl. No. 1, pp. S130–131. (abstract)

Lucassen, J.M., and Mercer, R.L. (1984) An Information Theoretic Approach to the Automatic Determination of Phonemic Baseforms, *Proc. ICASSP'84, San Diego*, pp. 42.5.1–42.5.4.

Maes, S. (1995) *Personal Communication*, IBM T.J. Watson Research Center.

Magnusson, L., Blomberg, M., Carlson, R., Elenius, K., and Granstrom, B. (1984) Swedish Speech Researchers Team-up With Electronic Venture Capitalists, *Speech Technology*, 2, pp. 15–24.

Markel, J.D., and Gray, A.H. (1976) *Linear Prediction of Speech*, Springer-Verlag, Berlin.

Moulines, E., and Charpentier, F. (1990) Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones, *Speech Communication*, 9, pp. 453–467.

Murray, I.R. and Arnott J.L. (1993) A Tool For The Rapid Development Of New Synthetic Voice Personalities, *Proc. ESCA Workshop on Speech and Language Technology for Disabled Persons, KTH, Stockholm*, pp. 111–114.

Nakajima, S., and Hamada, H. (1988) Automatic Generation of Synthesis Units Based on Context Oriented Clustering, *Proc. ICASSP'88, New York*, pp. 659–662.

Nakajima, S. (1993) English Speech Synthesis Based on Multi-Layered Context Oriented Clustering, *Proc. Eurospeech'93, Berlin*, pp. 1709–1712.

Nye, P.W., and Gaitenby, J. (1974) The Intelligibility of Synthetic Monosyllable Words in Short, Syntactically Normal Sentences, *Status Report on Speech Research SR-37/38*, Haskins Laboratories, New Haven, Connecticut, pp. 169–190.

O'Connor, J.D. (1973) *Phonetics*, Penguin, Middlesex, England.

Odell, J.J. (1992) *The Use of Decision Trees with Context Sensitive Phoneme Modelling*, MPhil Thesis, Cambridge University Engineering Department.

Odell, J.J., Woodland, P.C., and Young, S.J. (1994) Tree-Based State Clustering for Large Vocabulary Speech Recognition, *Proc. International Symposium on Speech, Image Processing, and Neural Networks, Hong Kong*, pp. 690–693.

Odell, J.J. (1995) *The Use of Context in Large Vocabulary Speech Recognition*, PhD. Thesis, Cambridge University Engineering Department.[1]

Olive, J.P. (1977) Rule Synthesis of Speech from Dyadic Units, *Proc. ICASSP'77, Hartford*, pp. 568–570.

Olive, J.P. (1990) A New Algorithm for a Concatenative Speech Synthesis System Using an Augmented Acoustic Inventory of Speech Sounds, *Proc. ESCA Workshop on Speech Synthesis, Autrans, Grenoble*, pp. 25–29.

O'Malley, M.H. (1990) Text-To-Speech Conversion Technology, *IEEE Computer (Magazine)*, August, pp. 17–23.

Parsons T.W. (1986) *Voice and Speech Processing*, McGraw-Hill, New York.

Paul, D.B., and Martin, E.A. (1988) Speaker Stress-Resistant Continuous Speech Recognition, *Proc. ICASSP'88, New York*, pp. 283–286.

Peterson, G.E., Wang, W.S-Y., and Sivertsen, E. (1958) Segmentation Techniques in Speech Synthesis, *Journal of the Acoustical Society of America*, Vol. 30, No. 8, pp. 739–742.

Pierrehumbert, J. (1981) Synthesizing Intonation, *Journal of the Acoustical Society of America*, Vol. 70, No. 4, pp. 985–995.

Poritz, A.B. (1982) Linear Predictive Hidden Markov Models and the Speech Signal, *Proc. ICASSP'82, Paris*, pp. 1291–1294.

Rabiner, L.R., Schafer, R.W., and Flanagan, J.L. (1971) Computer Synthesis of Speech by Concatenation of Formant-Coded Words, *Bell System Technical Journal*, 50, pp. 1541–1558.

[1]Available by anonymous ftp to svr-ftp.eng.cam.ac.uk

Rabiner, L.R. (1989) A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proc. IEEE*, Vol. 77, No. 2, pp. 257–286.

Rabiner, L.R., and Juang, B-H. (1993) *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, New Jersey.

Riley, M.D. (1990) Tree-Based Modelling for Speech Synthesis, *Proc. ESCA Workshop on Speech Synthesis, Autrans, Grenoble*, pp. 229–232.

Riley, M.D. (1992) Tree-based Modelling of Segmental Durations, *in*, Bailly, G., and Benoit, C., (eds.), *Talking Machines, Theories, Models, and Designs*, North-Holland, Elsevier Science Publishers, Amsterdam, pp. 265–273.

Robinson, A.J. (1994) SHORTEN: Simple Lossless and Near-Lossless Waveform Compression, *Technical report CUED/F-INFENG/TR.156*, Cambridge University Engineering Department.[1]

Ross, K., and Ostendorf, M. (1994) A Dynamical System Model for Generating F_0 for Synthesis, *Proc. 2nd ESCA/IEEE Workshop on Speech Synthesis, New Paltz, New York*, pp. 131–134.

Sagayama, S. (1989) Phoneme Environment Clustering for Speech Recognition, *Proc. ICASSP'89, Glasgow*, pp. 397–400.

Sagisaka, Y. (1988) Speech Synthesis by Rule Using an Optimal Selection of Non-Uniform Synthesis Units, *Proc. ICASSP'88, New York*, pp. 679–682.

Sanders, E., and Taylor, P. (1995) Using Statistical Models to Predict Phrase Boundaries for Speech Synthesis, *Proc. Eurospeech'95, Madrid*, pp. 1811–1814.

van Santen, J.P.H. (1994) Assignment of Segmental Duration in Text-to-Speech Synthesis, *Computer Speech and Language*, Vol. 8, No. 2, pp. 95–128.

Schwartz, R., Chow, Y., Roucos, S., Krasner, M., and Makhoul, J. (1984) Improved Hidden Markov Modeling of Phonemes for Continuous Speech Recognition, *Proc. ICASSP'84, San Diego*, pp. 35.6.1–35.6.4.

Secrest, B.G., and Doddington, G.R. (1983) An Integrated Pitch Tracking Algorithm for Speech Systems, *Proc. ICASSP'83, Boston*, pp. 1352–1355.

Sejnowski, T.J., and Rosenberg, C.R. (1986) NETtalk: A Parallel Network that Learns to Read Aloud, *Elec. Engr. & Comp. Sci. Tech. Report*, JHU/EECS-86/01, John Hopkins University, Baltimore.

Sharman, R.A. (1994) Concatenative Speech Synthesis Using Sub-phoneme Segments, *Proc. Institute of Acoustics*, Vol. 16, Part 5, pp. 367–374.

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, *Proc. ICSLP'92, Banff*, pp. 867–870.

Silverman, K., Kalyanswamy, A., Silverman, J., Basson, S., and Yashchin, D. (1993) Synthesiser Intelligibilty in the Context of a Name-and-Address Information Service, *Proc. Eurospeech'93, Berlin*, pp. 2169–2172.

Sorin, C. (1994) Towards High-Quality Multilingual Text-to-Speech, *Proc. CRIM/FORWISS Workshop on Progress and Prospects of Speech Research and Technology, Munich*, pp. 53–62.

Sorin, C., and Gagnoulet, C. (1995) On the Use of Speech Recognition and Text-to-Speech Technologies in Telecommunication Services Over the French PSN, *Proc. 7th World Telecommunication Forum, TELECOM'95, Geneva*, Vol 1, pp. 33–37.

Sproat, W.S., and Olive, J.P. (1995) Text-to-Speech Synthesis, *AT&T Technical Journal*, March/April, pp. 35–44.

Stella, M.G., and Charpentier, F.J. (1985) Diphone Synthesis Using Multipulse Coding and a Phase Vocoder, *Proc. ICASSP'85, Tampa*, pp. 740–743.

Talkin, D. (1995) *Personal Communication*, Entropic Research Laboratory Inc.

Taylor, P.A., and Isard, S.D. (1991) Automatic Diphone Segmentation, *Proc. Eurospeech'91, Genova*, pp. 709–711.

Tokuda, K., Kobayashi, T., and Imai, S. (1995a) Speech Parameter Generation from HMM Using Dynamic Features, *Proc. ICASSP'95, Detroit*, pp. 660–663.

Tokuda, K., Masuko, T., Yamada, T., Kobayashi, T., and Imai, S. (1995b) An Algorithm for Speech Parameter Generation from Continuous Mixture HMMs with Dynamic Features, *Proc. Eurospeech'95, Madrid*, pp. 757–760.

Varga, A., and Fallside, F. (1987) A Technique for Using Multipulse Linear Predictive Speech Synthesis in Text-to-Speech Type Systems, *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-35, No. 4, pp. 586–587.

Viterbi, A.J. (1967) Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm, *IEEE Transactions on Information Theory*, Vol. IT-13, no. 2, pp. 260–269.

Voiers, W.D. (1968) The Present State of Digital Vocoding Technique: A Diagnostic Evaluation, *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-16, No. 2, pp. 275–279.

Voiers, W.D. (1977) Diagnostic Evaluation of Speech Intelligibility, *in*, M.E. Hawley, (ed.), *Speech Intelligibility and Speaker Recognition*, Dowden, Hutchinson and Ross, Inc., pp. 374–387.

Wang, M., and Hirschberg, J. (1992) Automatic Classification of Intonational Phrase Boundaries, *Computer Speech and Language*, Vol. 6, No. 2, pp. 175–196.

Wang, W.J., Campbell, W.N., Iwahashi, N., and Sagisaka, Y. (1993) Tree-Based Unit Selection for English Speech Synthesis, *Proc. ICASSP'93, Minneapolis*, Vol. 2, pp. 191–194.

Wang, W.S-Y., and Peterson, G.E. (1958) Segment Inventory for Speech Synthesis, *Journal of the Acoustical Society of America*, Vol. 30, No. 8, pp. 743–746.

Wells, J.C. (1990) *Longman Pronunciation dictionary*, Longman, Harlow, England.

Woodland, P.C., Odell, J.J., Valtchev, V., and Young, S.J. (1994) Large Vocabulary Continuous Speech Recognition Using HTK, *Proc. ICASSP'94, Adelaide*, Vol 2, pp. 125–128.

Woodland, P.C., Leggetter, C.J., Odell, J.J., Valtchev, V., and Young, S.J. (1995) The 1994 HTK Large Vocabulary Speech Recognition System, *Proc. ICASSP'95, Detroit*, pp. 73–76.

Woodland, P.C., Gales, M.J.F., and Pye, D. (1996) Improving Environmental Robustness in Large Vocabulary Speech Recognition, *Proc. ICASSP'96, Atlanta*, (to appear).

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*, Vol. 66, No. 3, pp. 685–695.

Young, S.J., Russell, N.H., and Thornton, J.H.S. (1989) Token Passing: a Conceptual Model for Connected Speech Recognition Systems, *Technical report CUED/F-INFENG/TR.38*, Cambridge University Engineering Department.[1]

Young, S.J., and Woodland, P.C. (1993) The Use of State Tying in Continuous Speech Recognition, *Proc. Eurospeech'93, Berlin*, pp. 2203–2206.

Young, S.J., Woodland, P.C., and Byrne, W.J. (1993) *HTK Version 1.5: User, Reference & Programmer Manual*, Cambridge University Engineering Department & Entropic Research Laboratory Inc.

Young, S.J., Odell, J.J., and Woodland, P.C. (1994) Tree-Based State Tying for High Accuracy Acoustic Modelling, *Proc. ARPA Workshop on Human Language Technology, Merrill Lynch Conference Center, Plainsboro, New Jersey*, pp. 307–312.

Young, S.J., Jansen, J., Odell, J.J., Ollason, D., and Woodland, P.C. (1996) *The HTK Book*, Entropic Cambridge Research Laboratory Ltd.