Statistical Trajectory Models for Phonetic Recognition

by

William David Goldenthal

Submitted to the Department of Aeronautics and Astronautics on August 22, 1994
in partial fulfillment of the requirements for the Degree of
Doctor of Philosophy in Aeronautics and Astronautics.

Abstract

The main goal of this work is to develop an alternative methodology for acoustic–phonetic modelling of speech sounds. The approach utilizes a segment–based framework to capture the dynamical behavior and statistical dependencies of the acoustic attributes used to represent the speech waveform. Temporal behavior is modelled explicitly by creating dynamic *tracks* of the acoustic attributes used to represent the waveform, and by estimating the spatio–temporal correlation structure of the resulting errors. The tracks serve as templates from which synthetic segments of the acoustic attributes are generated. Scoring of an hypothesized phonetic segment is then based on the error between the measured acoustic attributes and the synthetic segments generated for each phonetic model.

Phonetic contextual influences are accounted for in two ways. First, context–dependent biphone tracks are created for each phonetic model. These tracks are then *merged* as needed to generate triphone tracks. The error statistics are pooled over all the contexts for each phonetic model. This allows for the creation of a large number of contextual models (e.g., 2,500) without compromising the robustness of the statistical parameter estimates. The resulting triphone coverage is over 99.5%.

The second method of accounting for context involves creating tracks of the transitions *between* phones. By clustering these tracks, complete models are constructed of over 200 "canonical" transitions. The transition models help in two ways. First, the transition scores are incorporated into the scoring framework to help determine the phonetic identity of the two phones involved. Secondly, they are used to determine likely segment boundaries within an utterance. This reduces the search space during phonetic recognition.

Phonetic classification experiments are performed which demonstrate the importance of the temporal correlation information in the speech signal. A complete phonetic recognition system, incorporating all the different model elements, is described. Both context–independent and context–dependent recognition experiments are performed using the TIMIT acoustic–phonetic corpus. The measured phonetic accuracy is virtually identical to the best reported result achieved with hidden Markov models, the most successful speech recognizers developed to this date.

Thesis Supervisor: Dr. James Glass
Title: Research Scientist, Laboratory for Computer Science

‘Twas brillig, and the slithy toves
    Did gyre and gimble in the wabe.

– *Lewis Carroll*

Acknowledgements

Well, the end of a long journey has arrived, and I feel a great sense of gratitude to a large number of people. It is a pleasure to take this opportunity to thank them. First and foremost I'd like to thank my thesis supervisor, Jim Glass, for agreeing to work on this project with me. Jim got me to start examining spectrograms right away and endeavored to teach me as much as he could from his deep understanding of acoustic–phonetics. He showed a great deal of insight and patience throughout this work and was particularly encouraging when things went wrong. I'd also like to express my gratitude to Victor Zue. Victor listened to me explain what I thought I might like to do for a thesis (at a time when I knew virtually nothing whatsoever about speech) and was willing to work things out so I could join his group. He has also provided guidance and wisdom on issues that extend beyond the academic. Together, Victor and Jim changed my career, and thus have had a profound and permanent influence on my life. I happily thank them for this very, very much.

I'd also like to thank Prof. Vander Velde, Prof. Ramnath and Milt Adams for serving on my thesis committee and lending their support to an unusual thesis topic for the Aeronautics and Astronautics Department. Prof. Vander Velde was particularly helpful and I appreciate his continuous guidance, patience, and understanding.

Additionally, thanks go to the Spoken Language Systems Group for making the lab such a fun place to work and for being so helpful over the last few years. Special thanks go to Mike McCandless and Lee Hetherington for their technical assistance on a wide range of coding, Unix, Emacs, and Latex problems. Thanks also to Christine

and Joe for keeping everything up and running. I also had a great time reading spec's with Mike, Jane, Helen, Tim and Giovanni on an almost daily basis for over a year, and with everyone in my spec reading class every Friday. I looked forward to this every week. Speaking of spec's, I'd like to thank Nancy Daly for giving so freely of her time in helping me when I was starting out, and also Jane Chang for taking over the job of organizing and running the weekly class. Jane was also a great office mate who tolerated my idiosyncrasies, such as playing the same cd over and over as I wrote my code. Sharing my work, daily life, and some fun travel with the people in this group has been a great experience. I have really enjoyed myself here and feel extremely fortunate.

Extra thanks also go to Dave Goddeau (dg), my first office mate in the group, and frequent companion at the one and only Miracle of Science. DG listened to me go on about this work at length for several years. He always provided insight, encouragement, and humor when anyone else would have told me to be quiet already and drink my beer. Another person who provided lots of inspiration and interesting ideas over both beers and while working out was Brian Eberman. It was lots of fun trading concepts back and forth on our respective dissertations, which were on essentially the same problem, just in different disciplines. Thanks to John Pazaris and John Wolfe for providing fun, inspiration, humor, and energy both at the lab and on the town.

I'd also like to extend thanks to the crew of people who work at the Miracle of Science, especially Gary, Dana, Tracy, Chad, Justin, Kristen, Christine, Laura, Suzi, and also the owners Matt and Chris.

Special thanks go to the Charles Stark Draper Laboratory which supported this research under the Draper Staff Associate program. I'd like to thank Jake personally for his support of this program under difficult conditions. Thanks also go to Bill Bonnice, Paul Motyka, Peg Conley, and Eli Gai at Draper for their support and friendship over the years.

Thanks to my roommate, Jenn Matheny, for being so good natured when I came home grumbling from the lab, for convincing her employer to connect our apartment to the Internet, and also for being such a lousy poker player. Thanks also go to the rest of the poker crowd, many of whom were undergrads with me in Bexley, especially Dan, Reza, Gene, and Larry (even though he doesn't play).

Finally, my deepest thanks go to those who have provided personal support, encouragement and extra strength throughout this entire ordeal. This includes my family, Dave Parker, Gini Laffey, and Lori Britton. I'd especially like to thank Lori for making the final year of this effort such a wonderful experience.

This thesis is dedicated to my parents.

Contents

List of Figures

Glossary of Common Speech Terms as used in the Thesis

TERM	**MEANING**
phoneme	An abstract linguistic unit that forms the basis for writing down a language. Changing a phoneme changes a word.
phone	Acoustic realization of a phoneme
allophones	The different phonetic variants of a phoneme
frame	Output produced by processing a single window of speech Represented by a vector of acoustic attributes (e.g. spectral or cepstral coefficients). In this work, frames are computed every 5 ms.
segment	A sequence of consecutive frames of speech treated as a single unit.
token	A segment of speech consisting of an acoustic–phonetic unit, such as a phone.
utterance	A sequence of spoken words contained within two #h (silence) symbols.
phonetic transcription	A string of phonetic symbols which represent the acoustic–phonetic units comprising an utterance.
training set	The set of utterances used to train the phonetic models. The speakers, and in some cases the sentences, are distinct from those used in the test set and the development set.
development set	A set of utterances used for evaluation purposes. By experimenting on this set the system parameters can be tuned without unfairly "learning" the test set.
test set	The set of utterances scored by the trained models in a phonetic classification or recognition experiment.
phonetic classification	An experiment where the correct phone boundaries are known, but the phonetic identity must be determined.
phonetic recognition	An experiment where both the boundaries and the phonetic sequence must be determined. The output is an hypothesized phonetic transcription of the input utterance.
context independent	Phonetic models and experiments do not account for phonetic context.
context dependent	Phonetic models and experiments account for phonetic context.
ASR	Automatic Speech Recognition

Chapter 1

Introduction

The task of automatic speech recognition (ASR) consists of decoding a word sequence from a continuous speech signal. In order to achieve reasonable levels of performance, past ASR systems have constrained the permissible speech input in order to simplify the decoding task. Typical constraints are training the system for each individual speaker (speaker–dependent systems), limiting the system vocabulary to a small number of words, requiring input to be isolated words only, permitting only read (as opposed to spontaneous) speech, or some combination of the above. Recently however, state–of–the–art systems have been able to achieve useful performance levels for speaker–independent, continuous/spontaneous speech systems, operating with vocabularies of greater than five thousand words [28, 55, 73].

A block–diagram of the major components of an ASR system is shown in Figure 1-1. Typically, the samples of the continuous speech signal are first processed to form a discrete sequence of observation vectors. This operation is denoted by the *Signal Processing* block in the figure. The resulting components of the observation vectors are the acoustic attributes that have been chosen to represent the speech signal. Examples of commonly chosen attributes are DFT–based spectral coefficients or auditory model parameters [51]. Each observation vector is called a "frame" of speech, and the

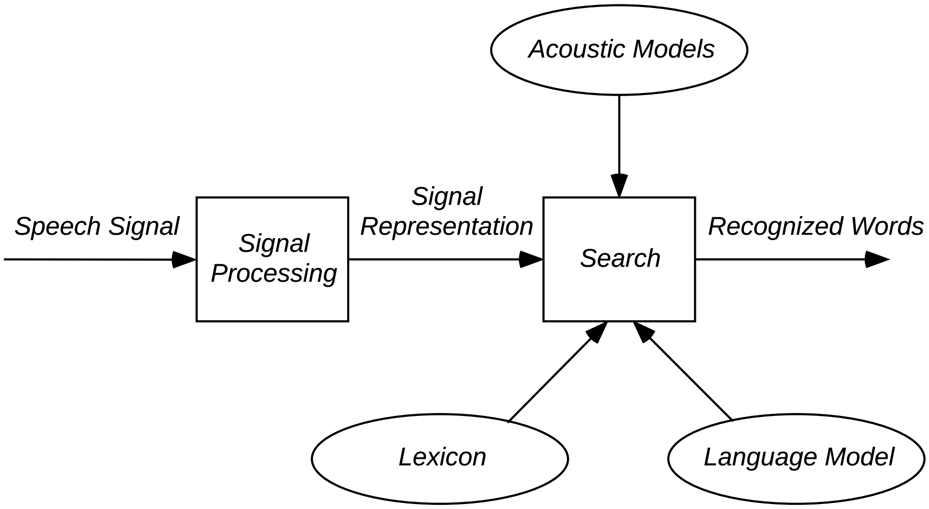

Figure 1-1: Major Components of an Automatic Speech Recognition (ASR) System

sequence of T frames comprises the *Signal Representation*, $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_T\}$.

A search is then conducted over the frame sequence, X, to produce hypothesized word sequences. Acoustic models are used to score the individual frames or multiple frame sequences, known as *segments*. Language models which contain information about allowable sequences of speech units in the lexicon (e.g. phones, words, etc.) are also incorporated into the scoring process. The representation, models, search, and scoring procedures are key design components of the system. The search framework used in this thesis is described in detail in Chapter 8.

As the number of words in the lexicon becomes large, the task of training individual acoustic models for each word becomes prohibitive. Consequently, an intermediate level of representation is generally used. A common representation involves describing the pronunciation of a word in terms of *phonemes*. A phoneme is an abstract fundamental unit of a language. By definition, changing a phoneme changes the meaning of a word. For example, if the phoneme /p/ in the word *pit* is changed to a /b/, the word becomes *bit*. A small number of phonemes can be used to describe all the words in a given language (English consists of roughly forty phonemes). By representing word pronunciations as a sequence of phonemes, the number of acoustic models and the required training data of moderate and large vocabulary systems can be drastically reduced.

Phonemes can be realized in a variety of acoustically distinct manners depending on the phonetic context (e.g. syllable position, neighboring phones), the stress, the speaker, and other factors. The actual acoustic realization of a phoneme is known as a *phone*. This distinction is an important one. The different acoustic realizations of the same phoneme do not affect the meaning of a word. An example of this often occurs in the word *butter*, where the phoneme /t/ is frequently realized in American English as the phone [ɾ] (called a "flap"). The acoustic variability that can occur when realizing the same phoneme is part of what makes the task of identifying a phoneme so challenging.

The acoustic models are generally trained to recognize some set of phones (the exact set being a design decision). The task of decoding a phone sequence is known as *phonetic recognition*, and the resulting output is a sequence of probabilities from which phonetic transcriptions are hypothesized. The phonetic probabilities are of fundamental importance to the ASR task since they are the foundation upon which the word string search is based. All large vocabulary speech systems utilize phonetic models as a component in the speech recognition system. It is the purpose of this thesis to create better acoustic models of the phonetic units, so as to improve the accuracy achieved during the phonetic recognition task, and hence, the performance of the entire ASR system.

1.1 Variability of the Speech Signal

Speech is produced by the coordinated manipulation of a set of articulators, including the tongue, lips, jaw, vocal folds, and velum. The speaker–dependent characteristics of the articulators and the vocal tract can cause a large amount of acoustic variability in the realization of the same phoneme sequence. The speaker's environment, mood, health, and prosody (pitch and emphasis) can all affect the acoustic realization of a phonemic sequence [79].

In addition to these speaker–dependent effects, the phonemic context influences the motion of the articulators and the resulting acoustic output. In many contexts, it is frequently unclear where one phonetic segment ends and the next begins. The overlapping of phonetic segments stems from overlap in adjacent articulatory gestures. This phenomenon is known as co–articulation, and is a significant factor in contributing to the variability in the acoustic realization of a phoneme.

An example spectrogram of the utterance "Two plus seven is less than ten" is shown in Figure 1-2. The spectrogram displays acoustic energy (dark regions) as a function of frequency (y–axis) and time (x–axis). The dark bands in the vocalic regions represent acoustic resonances, or poles of the vocal–tract transfer function, and are known as *formants.* Predictions of the formant frequencies for the vowels, as well as the energy distributions for the other sounds, come under the acoustic theory of speech production [11, 20]. A phonetic transcription and an orthographic transcription are also shown in the figure.

The effects of co–articulation are apparent when comparing the three segments labeled /ɛ/. In the first segment (located at approximately 0.6 seconds), the second formant moves downward from left to right, from 1.8 kHz. to 1.5 kHz. In this case, the second formant is pulled up towards the alveolar locus by the fricative /s/ on the left, and is pulled downward due to the labial fricative /v/ on the right. The target position for this formant is most likely represented by its location in the middle of the /ɛ/ segment. This motion contrasts sharply with the motion in the second occurrence of /ɛ/ which starts just after t = 1.0 seconds. Here the second resonance is moving up from left to right, and again contextual factors are the reason. The dominant factor in this case is the strong downward pull provided by the /l/ in the left position. The third occurrence of /ɛ/ has a relatively stable second formant, however the first

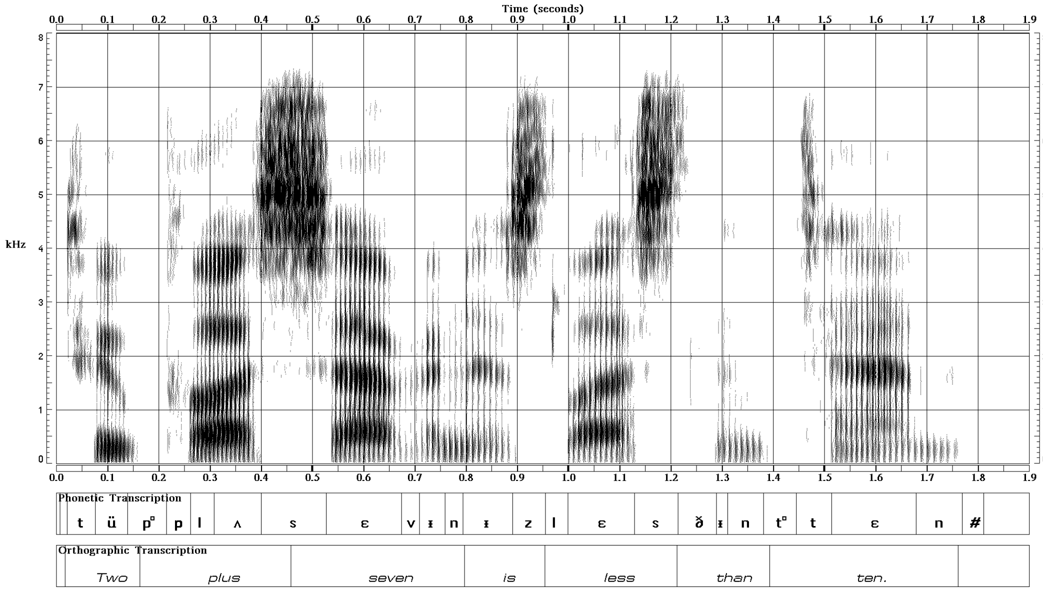

Figure 1-2: Spectrogram of the sentence: "Two plus seven is less than ten." Dark areas show regions of high energy. The x–axis represents time (in seconds) and the y–axis represents frequency (in kHz.) A phonetic transcription and an orthographic transcription of the utterance appear below the spectrogram.

formant has been broadened and made diffuse. This form of co–articulation is known as *nasalization* and occurs because the velum has been lowered in anticipation of the alveolar nasal /n/. A more detailed analysis of this spectrogram can be found in [79].

1.2 Dynamics and Correlation of Acoustic Spectra

Despite the high degree of variability in the speech signal, there exists much that is consistent both within a phonetic unit and across an utterance. This consistency is what makes spoken communication so robust. A given phone generally has a configuration of the articulators, or *target position* [34] associated with it. Whether or not the target position is reached, there tend to exist intervals of speech which are predominantly representative of a particular phone. Although differences exist between various speakers' physical characteristics, the existence of the target position implies that their articulators may share similar *relative* motions when realizing the same phone. This similarity in the dynamics of the articulators should translate into similar dynamics in the acoustic attributes of the phone.

Therefore, the trajectories of the acoustic attributes should share dynamic characteristics for a given sequence of phones as the articulators move through a sequence of gestures. The greater the similarity of the phonetic contexts, the greater the similarity of the motion of the acoustic attributes. An example of this similarity is shown in the spectrogram in Figure 1-3 for the word *white* /w ɑʸ t/ spoken by two different speakers. Although the absolute positions of the formants differ, there is a similar motion throughout the phonetic sequence. A methodology that captures the dynamics of the acoustic attributes of a phone should have an advantage in identifying an unknown phonetic segment of speech. Also note the difficulty in determining precisely when the phone [w] transitions to the phone [ɑʸ].

Statistical models of the phonetic units have historically provided a robust method for dealing with the variability between speakers [1, 3, 64]. These statistical models may capture the correlations between the acoustic attributes at a specific time, and over a specified time interval. The temporal correlation information can be utilized to account for the fact that the same vocal track is producing the entire phonetic sequence in an utterance. These temporal correlations in the speech signal are not modelled directly in most ASR systems. The most popular current algorithm, hidden Markov modelling (HMM), is only able to account for these correlations indirectly. This work will attempt to demonstrate the importance of the temporal correlations, and construct models which utilize them effectively.

1.3 Scope of the Thesis

This thesis attempts to incorporate dynamical models of the acoustic spectra into a phonetic recognition scheme. The approach will be to first determine a means of

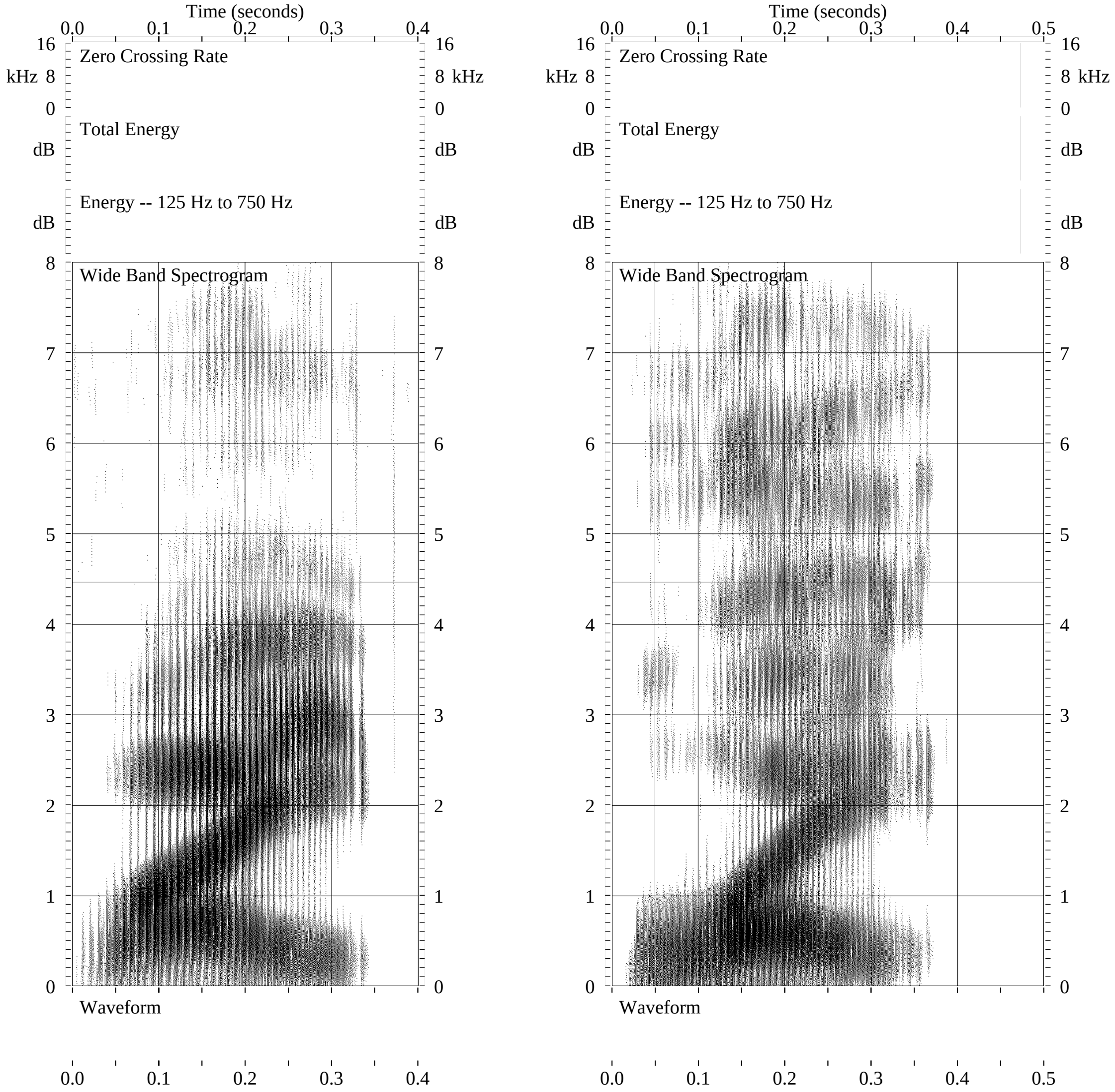

Figure 1-3: Spectrograms of the word “white”.

The word is spoken by two different speakers. Note that although the absolute positions of the formants are slightly different, the relative motions of the first three formants are very similar. Additional acoustic parameters appear above the spectrogram, and the speech waveform appears below it.

mapping a phone's variable duration tokens onto a fixed length *track*. A track is defined to be a trajectory, or temporal evolution of the acoustic attributes over a segment. A track consists of a sequence of M state vectors $T = \{\vec{t}_1, \ldots, \vec{t}_M\}$ which are used as the basis for generating a synthetic segment:

$$G = f(T, N) = \{\vec{g}_1, \ldots, \vec{g}_N\} \tag{1.1}$$

for any number of frames N, where $f()$ is a generation function. The track serves as a template and attempts to capture segment-level spectral dynamics.

After a track is computed from the training tokens for a particular phone, the same tokens are used to generate an error model based on the differences between the track and the training tokens. The objective of the error model is to take advantage of information residing in the error correlations both between acoustic attributes and over time.

The track and its associated statistical error model form a baseline model for each phonetic unit. Although the baseline model provides a robust general characterization of the phonetic unit it represents, details attributable to phonetic context and speaker dependencies tend to be "averaged out." That is, since the track represents the phone in all contexts, it tends not to contain contextual information which is critical to enhancing model accuracy due to co-articulation. One means of addressing this problem is to create context–dependent tracks. Another is to specifically model the transition dynamics between phonemes. Both of these approaches will be addressed in this thesis.

When evaluating a methodology, two types of experiments commonly conducted are phonetic classification and phonetic recognition. In phonetic classification the segmentation boundaries in the utterance are known, and the task is to correctly classify each segment. In phonetic recognition the segment boundaries are not known. As a result, insertion and deletion errors are possible, along with substitution errors (misclassification).

Chapter 2 of this thesis examines the signal processing issues involved in transforming the digitized speech waveform into the acoustic representation used throughout this thesis, the Mel–frequency cepstral coefficients. Chapter 2 then describes the TIMIT acoustic–phonetic corpus, which is the source of all the speech data used in this work. Chapter 3 begins with a description of the current state–of–the–art in automatic speech recognition, and then provides details of other commonly used approaches to the problem.

The development of the statistical trajectory models begins with Chapter 4. This chapter defines the acoustic trajectory models, known as tracks, which are used to capture the spectral dynamics in a phonetic segment. Chapter 5 describes the statistical component of the algorithm, which is based on an error signal derived from the dynamic tracks. Chapter 6 contains baseline classification experiments which provide a preliminary algorithm evaluation and permits some parameter tuning. The statistical trajectory models are initially evaluated on a vowel classification task and then on a complete set of context independent phonetic classification experiments.

The incorporation of context–dependent models and some of the significant advantages of separating the dynamical and statistical model components are explored in Chapter 7. The statistical trajectory model algorithm is then used to create models of the phonetic transitions. The chapter describes how the transitions can be clustered such that models of a reasonable number of “canonical” transitions can be created. The different elements of the thesis are then brought together into a complete system in Chapter 8. This chapter describes the search problem involved in phonetic recognition, and discusses the phonetic recognition experiments. Chapter 9 contains conclusions and suggestions for future avenues of research.

Chapter 2

Background

This chapter provides the details of some of the relevant background for the work conducted in this thesis. A discussion of the signal processing issues involved in computing acoustic features from the digitized speech waveform is presented first. This is followed by a description of the TIMIT acoustic–phonetic speech corpus, and a breakdown of the different data sets that will be referenced later in the thesis.

2.1 Signal Processing and Signal Representation

As discussed in the introduction, the continuous speech signal is digitally sampled and then processed via a temporal and/or spectral analysis into a sequence of observation frames. The signal processing in this work is typical of most ASR systems. The signal representation to be generated consists of the Mel–frequency cepstral coefficients (MFCC's) [53]. The cepstrum is the inverse Fourier transform of the logarithm of the power spectrum of a signal [66]. The MFCC's provide a high degree of data reduction over using values of the power spectral density directly, since the power spectrum at each frame can be represented with relatively few parameters.

There are several other reasons why the MFCC's were chosen as the acoustic attributes. The Mel–frequency warping is motivated by the frequency response characteristics of the inner ear, where the critical bandwidths are known to vary with frequency [21, 72]. The MFCC's are also one of the most common representations used in modern speech research. Therefore direct comparisons can be made between the results obtained here and in other work, without the concern that differences in the results are attributable to a specific acoustic representation. In addition, the MFCC's have been shown to achieve good performance, particularly in an environment where noise is not a critical factor [45, 50, 53].

The key steps involved in producing the MFCC's from the continuous speech waveform are:

1. The signal is sampled at 16 kHz., pre–emphasized, and multiplied by a Hamming window of 25.6 ms. which is advanced at a *frame* rate of 5 ms.

2. A 256 point Discrete Fourier Transform (DFT) is then computed for each frame.

3. The Fourier transform coefficients are squared, and the resulting squared magnitude spectrum is passed through a set of 40 overlapping Mel–frequency triangular filter banks. The log energy output of each of these filters collectively form the 40 Mel–frequency spectral coefficients (MFSC), X_j, j = 0,1,2,...,40.

4. A cosine transform of the MFSC's is then used to generate the 15 MFCC which are the acoustic attributes used in this thesis.

The Mel–frequency filters consist of 13 triangles spread evenly on a linear frequency scale from 130 Hz to 1 kHz, and 27 triangles evenly distributed on a logarithmic scale from 1 kHz to 6.4 kHz. Since the bandwidths of the triangular filters increase with center frequency, the area of each filter is normalized to avoid amplifying the higher frequency coefficients, as in [51]. The cosine transform which yields the MFCC, C_i, i = 0,1,2...,14, from the MFSC is:

$$C_i = \sum_{j=0}^{40} X_j \cos\left[i(j-\frac{1}{2})\frac{\pi}{40}\right] \tag{2.1}$$

Note that the lowest cepstral coefficient, C_0, is a summation of the log energy from each filter. Therefore, it is related to the amount of energy in a frame.

2.1.1 Cepstral Differencing

Cepstral differencing involves computing the rate of change of the MFCC's for use as acoustic attributes [24, 64, 75]. The computation has been widely employed, due to its positive impact on system performance. The performance improvement results from the fact that the frames are not independent, and that derivative information is a means of capturing some of the correlation between frames. The resulting attributes are called the delta cepstrum, and in some instances delta–delta cepstrum has also been used [5, 38].

The derivative of the MFCC's at a given frame is approximated by computing a first–order polynomial over a finite length window, centered about the frame. Specifically, for a given MFCC, C_i, at time t:

$$\Delta C_i(t,k) = \sum_{j=-k}^{k} j C_i(t+j) \tag{2.2}$$

For a given segment, the delta cepstrals near the segment boundary are effectively providing contextual information about the phonetic environment. For this reason, many implementations utilize only the delta cepstrals computed at the segment boundaries [7, 26]. The delta cepstrals will be used as additional acoustic attributes for many of the classification experiments and all of the recognition experiments performed in this thesis.

2.2 TIMIT

The TIMIT acoustic–phonetic speech corpus [36] is used for all training, development, and performance evaluation experiments. This corpus is widely used throughout the world and provides a standard that permits direct comparison of experimental results obtained by different methodologies. The entire corpus consists of 10 sentences recorded from each of 630 speakers of American English. Two of the sentences (*sa*) are identical for all the speakers. Five of the sentences (*sx*) for each speaker were drawn from a set of 450 phonetically compact sentences hand–designed at MIT. The emphasis behind these sentences was on covering a wide range of phonetic pairs. The 450 (*sx*) sentences are each spoken by seven different speakers. The final three sentences (*si*) for each speaker were chosen at random from the Brown Corpus [23] and are unique for all the speakers. The speakers in the corpus are comprised of males and females (at a ratio of roughly two to one) from eight predefined dialect regions of the United States.

Generally, the (*sa*) sentences are not utilized since the phonetic contexts used in these sentences would be over–represented. This has the effect of favorably skewing the performance results. Counting only the (*sx*) and (*si*) utterances, the entire corpus consists of 194,591 phonetic tokens. A list of the TIMIT phone set and their equivalent International Phonetic Association (IPA) symbols are shown in Table 2.1.

Some typical *sx* sentences from the corpus are:

> "Ambidextrous pickpockets accomplish more."
>
> "The bungalow was pleasantly situated near the shore."
>
> "A big goat idly ambled through the farmyard."
>
> "Eating spinach nightly increases strength miraculously."
>
> "At twilight on the twelfth day we'll have Chablis."

For all experiments, the data is divided into distinct units known as the *training set* and the *test set.* The training set is used to estimate the parameters for each of the phonetic models to be used in the experiments. The test set consists of the actual test data for the classification or recognition performance evaluation. Since it is often important to run experiments to determine parameter settings, a third data set is often used. This data set is called a *development set*, and it serves the purpose of allowing algorithms to be refined without inadvertently "tuning" them to a particular test set, which would result in artificially inflated performance.

Regardless of the particular experiment in this thesis, speakers from the training and development/test sets never overlap. This is important to ensure fair experimental conditions. Both sets are generally chosen to reflect a well balanced representation of the speakers in the corpus. Most of the training and test sets utilized in this work were selected specifically because they are identical to training and test sets used in other work. Therefore, the results can be directly compared to those obtained elsewhere and reported in the literature. The different data sets used in this thesis are listed along with some of their statistics in Table 2.2.

IPA	**TIMIT**	**EXAMPLE**	**IPA**	**TIMIT**	**EXAMPLE**
ɑ	aa	f*a*ther	ŋ̍	eng	Wash*ing*ton
æ	ae	b*a*t	f	f	*f*ief
ʌ	ah	b*u*t	v	v	*v*ery
ɔ	ao	b*ou*ght	θ	th	*th*ief
ɑ^{w}	aw	ab*out*	ð	dh	*th*ey
ə	ax	*a*bout	s	s	*s*is
ə^{h}	ax–h	aspirated schwa	z	z	*z*oo
ɚ	axr	din*er*	š	sh	*sh*oe
ɑ^{y}	ay	b*i*te	ž	zh	mea*s*ure
ɛ	eh	b*e*t	p	p	*p*op
ɝ	er	b*ir*d	b	b	*b*ob
e	ey	b*ai*t	t	t	*t*ot
ɪ	ih	b*i*t	d	d	*d*ad
ɨ	ix	ros*es*	k	k	*k*ick
i	iy	b*ea*t	g	g	*g*ag
o	ow	b*oa*t	p$^{\square}$	pcl	p closure
ɔ^{y}	oy	b*oy*	b$^{\square}$	bcl	b closure
ʊ	uh	b*oo*k	t$^{\square}$	tcl	t closure
u	uw	b*oo*t	d$^{\square}$	dcl	d closure
ü	ux	b*eau*ty	k$^{\square}$	kcl	k closure
w	w	*w*et	g$^{\square}$	gcl	g closure
y	y	*y*et	č	ch	*ch*urch
r	r	*r*ed	ǰ	jh	*j*udge
l	l	l*e*d	ɾ	dx	bu*tt*er
l̩	el	bott*le*	ɾ̃	nx	fu*nn*y
m	m	*m*om	h	hh	*h*ay
n	n	*n*o	ɦ	hv	Le*h*eigh
ŋ	ng	si*ng*	▯	epi	epinthetic silence
m̩	em	botto*m*	□	pau	pause
n̩	en	butto*n*	ʔ	q	glottal stop
	h#	Utterance initial and final silence			

Table 2.1: List of the IPA phone symbols, the equivalent TIMIT symbols, and examples.

Data Set Name	**# Speakers**	**Utterances**	**Tokens**
HM Train (Vowels)	499 (357 m/142 f)	2,495 *sx*	20,528
HM Sub–Train (Vowels)	450 (318 m/132 f)	2,250 *sx*	18,450
HM Augment–Train (Vowels)	499 (357 m/142 f)	3,992 *sx, si*	34,576
HM Development (Vowels)	49 (39 m/10 f)	245 *sx*	2,078
MIT Train	567 (397 m/170 f)	4,536 *sx, si*	175,101
MIT Test–V (Vowels)	50 (33 m/17 f)	250 *sx*	1,879
MIT Test	50 (33 m/17 f)	250 *sx*	8,922
MIT Augment–Test	50 (33 m/17 f)	250 *sx, si*	15,027
NIST Train	462 (326 m/136 f)	3,696 *sx, si*	142,910
BG–Dev1	16 (16 m/0 f)	128 *sx, si*	4,810
Dev1	50 (34 m/16 f)	400 *sx, si*	15,334
NIST Core	24 (16 m/8 f)	192 *sx, si*	7,333
BU Train	426 (426 m/0 f)	3,408 *sx, si*	130,906
BU Test	12 (12 m/0 f)	96 *sx, si*	3,731
KFL Train	610 (424 m/186 f)	4,880 *sx, si*	188,435
KFL Test	20 (14 m/6 f)	160 *sx, si*	6,156

Table 2.2: The TIMIT data sets used for training, development, and experimentation. The data sets designated (Vowels) only include tokens of the 16 unreduced vowels of American English. NIST refers to the National Institute of Standards and Technology.

Chapter 3

Current ASR Approaches

Speech recognition approaches tend to fall into two categories, frame based and segment based. Frame based approaches are currently utilized in most ASR systems. In a frame based system, each observation frame in the sequence $O = \{\vec{o}_1, \ldots, \vec{o}_T\}$ receives a score for each phonetic model. There is no pre–segmentation of the signal into larger units. Rather, the segmentation comes about implicitly as a consequence of the frame–by–frame scoring. In a segment based system, start and end times of larger units are hypothesized within the signal as a distinct step in the scoring process. These larger units generally represent individual phonetic units of speech.

Segment based systems offer the potential advantage of being able to accurately capture segment level dynamics, as well as directly modelling temporal correlations within the segment. Also, segment level features, such as segment duration, can be easily incorporated into the system. An advantage of a frame based system is that each frame receives its own score. This permits the scores for different transcription candidates to be directly compared since each alternative transcription has an identical number of scores. The alternative transcriptions share the same probability space, of dimension proportional to the number of frames in the utterance. In a segment based framework it can be difficult to compare utterance likelihoods which hypothesize different numbers of segments since the dimension of the probability space is a function of the number of segments. Finally, a frame based system tends to have a computational advantage since the segmentation step does not have to be explicitly performed.

This chapter presents a brief review of the major approaches used to perform the speech recognition task. The frame based approaches are discussed first, starting with dynamic time–warping (DTW). DTW is a template based approach which achieved some success in smaller vocabulary speaker–dependent tasks, and later some simple speaker–independent tasks [78]. A difficulty with the DTW approach is its inability to generalize, which caused difficulties in speaker–independent systems. DTW has generally been supplanted by approaches which incorporate statistical methods. HMM is the most common technology used in speech recognition systems at this time [78]. HMM is a statistical frame based approach which are able to account for many of the uncertainties in the speech signal, including the temporal variability of phonetic units and speaker variability. More recently, connectionist approaches have been applied to the speech recognition problem [44, 46, 67]. The section on frame based approaches concludes with a discussion of the most successful of these approaches which utilizes a recurrent error propagation neural network (REPN) [67].

This chapter next reviews several segment based approaches. These approaches all generally involve an explicit segmentation of the signal into phonetic units. The section focuses on approaches which attempt to incorporate dynamical information into the segment models, since these approaches are most relevant to the work discussed in this thesis. The final section provides a summary of the current applicable performance levels achieved by these systems.

3.1 Frame Based Approaches

Frame based systems are characterized by approaches which assume some degree of independence between frames, and, do not rely on an explicit segmentation of the speech signal. In general these approaches produce a probability or likelihood score for each observation frame.

3.1.1 Dynamic Time Warping

Dynamic time–warping (DTW) involves the creation of reference templates for each unit to be recognized [31, 65]. Historically, the units used were words or connected words. The key idea is to account for differences in speaking rate while determining which reference pattern is, by some measure, "closest" to the test pattern. Hence, the algorithm involves an alignment component and a distance metric component.

The problem is reduced to finding the best path through a finite grid, subject to constraints imposed by physical limitations (the speaking rate cannot undergo arbitrary variations). The best path is taken to be one which minimizes an accumulated distance metric. This path uses an algorithm known as a Viterbi [22] search which utilizes dynamic programming to find the optimal path. A wide variety of metrics have been used, but the most common [65] are the Euclidean metric and the linear predictor coefficient (LPC) distance metric of Itakura [31]. A major advantage of DTW is that by creating templates at the word level, the variations in the realization of the phones which arise due to context are inherently accounted for. However, regardless of the implementation, statistics on the errors were not generally utilized in the DTW approach. The frame–to–frame errors are considered to be statistically independent.

A major drawback of this method is the lack of any statistical mechanism to account for differences between speakers. This makes it very difficult to generalize from multi–speaker to speaker–independent speech recognition systems. Additionally, DTW does not scale up easily to moderate and large sized speech recognition vocabularies. Once the number of words to be recognized gets into the hundreds, template training becomes difficult due to a lack of training data, and the matching process becomes computationally expensive. If DTW is applied at the phonetic level, its advantages become weakened. The variability of phones due to context is no longer automatically accounted for, and the lack of a statistical mechanism for dealing with this variability is a serious drawback. Also, the machinery which accounts for variations in speaking rate is less powerful at the phonetic level, where the rate is more likely to be constant throughout the duration of a single phone. Therefore, for larger scale speaker–independent tasks, speech recognition systems began to adopt approaches which combine the dynamic programming advantage of DTW with stochastically based techniques.

3.1.2 Hidden Markov Models

Hidden Markov modelling [42, 64] (HMM) is currently the most popular technique for performing phonetic recognition and is utilized in a vast majority of modern speech recognition systems. HMM is a statistical frame based approach consisting of a set of states connected to each other via transition probabilities. While occupying a state, observations are generated randomly from a probability density function (pdf). The transition probabilities and output distributions together constitute an HMM model. The key assumption inherent in an HMM is the Markovian assumption that the observations are independent, given the state sequence up to the current time. In practice, the majority of HMM systems are modelled as first–order Markov processes. Therefore, the observation likelihood depends only on the current state of the system.

A typical topology of an HMM designed to model a phone is depicted in Figure 3-1. This HMM is called a “left–to–right” HMM because state transitions are only permitted in one direction. The state is forced to either stay the same or increase with time. Loosely speaking, the three states are said to model the start, middle,

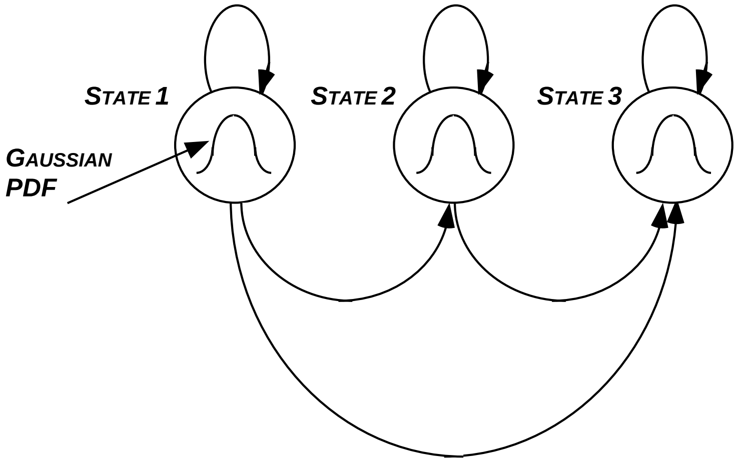

Figure 3-1: A typical left–to-right HMM topology.
Each state contains a pdf (depicted here as Gaussian), from which observations are drawn at random. Given the state sequence, the observations are independent.

and end of a phone. Since the observations, in this case frames of speech, are always moving forward in time, the vast majority of phonetic models contain the left–to–right feature. The output distributions shown are continuous Gaussian densities, although mixtures of densities are more common. The output distribution can also be modelled as discrete observations. The Markov models are called hidden because the "true" state of the system is unknown.

Since each state has a different output distribution, an HMM is equipped to handle the non–stationarity inherent in the speech signal. HMM also manages certain temporal aspects of the speech problem in an elegant manner. The variability of durations over a phone training set is handled automatically by the fact that an individual state can be occupied for a variable length of time. Another advantage of the HMM approach is that it does not require an explicit temporal–alignment, or segmentation, of the speech signal. Since each frame in an utterance receives its own score, the likelihood scores for alternative segmentations can be directly compared to each other. The alignment which results in the best score for the entire utterance is then chosen. Finally, an efficient algorithm, the Baum–Welch reestimation algorithm, exists for training HMM [4].

A disadvantage of HMM is that temporal correlations are not modeled explicitly. Instead, the correlations are represented implicitly through the state transition probabilities. However, it has been shown that the state transition probabilities have much less impact on the observation likelihoods than the output distributions connected to the states [52]. As the dimension of the output distribution increases, this effect becomes more pronounced. The summation over different possible state sequences to compute the observation likelihoods is often not performed. Instead, a Viterbi search determines the most likely state sequence and only this score is used to determine the likelihood of each model [52, 64]. The implication is that the temporal correlation information is not being used efficiently, or that it is not important.

However, it has been demonstrated that significant temporal correlations do exist [15, 26]. Evidence of these correlations will be presented in Chapter 5. Incorporation of this information into an appropriate structure should translate into performance benefits.

There have been attempts to explicitly model the dynamics of the acoustic attributes within an HMM framework. Generally this has been done, with some success, by incorporating first (and possibly second) order differences of the acoustic parameters in the observation vector. Other approaches are segmental HMM, proposed by Russell [71] and also by Marcus [48], and state–conditioned trend functions used by Deng [12]. None of these approaches have gained general acceptance within the community, nor have they been shown to generate results superior to more traditional HMM techniques.

3.1.3 Recurrent Error Propagation Networks

Currently, the phonetic recognition system which has produced the best performance on the TIMIT data base, utilizes a recurrent error propagation network (REPN) to distinguish the phonetic units. A complete description of this system, developed by Robinson and Fallside, along with subsequent improvements, can be found by consulting [67, 68, 69].

The network uses a recurrent multi–layer perceptron architecture, where part of the output is fed back to the input after a time delay of one frame. The network is trained by propagation of the error gradient backwards in time. The network inputs consist of a representation of the power spectrum, additional features such as zero–crossing rate or voicing information, and the previous system output (i.e. feedback). The network outputs are the probabilities that a given frame is part of a segment labeled with a specific phonetic symbol. The network has been constructed so as to function as a component of a hybrid connectionist–HMM system, with the neural net computing the phonetic probabilities and the HMM using these probabilities to recognize words [69].

A key advantage of the connectionist approach is the capability to utilize feedback to potentially incorporate contextual information. While this feedback essentially generates left–context (past) information, a delay in the propagation from input to output allows for right–context (future) information to be incorporated. This feedback also permits the network to indirectly estimate temporal correlations over arbitrarily long time intervals. As stated above, this system has produced the best performance in phonetic recognition on the TIMIT corpus. However, when applied to word recognition, the results thus far have been well below the results achieved by standard HMM based systems [69]. This result is consistent with previous work using hybrid connectionist–HMM systems [54]. Robinson speculates that there are two main factors responsible for poor word level performance: a smaller system parameter size, and an unsophisticated word model (e.g. the system in [69] permitted only a single pronunciation for each word). However, at the time of this writing, no connectionist based system has been able to exceed the performance of the best HMM's at the word level.

3.2 Segment Based Approaches

Recently, there have been a number of segment based approaches to the phonetic recognition problem. This includes MIT's SUMMIT system [60, 80, 81, 82] and the Stochastic Segment Model (SSM) approach of Ostendorf and Roucos [56, 57, 70]. The SSM incorporates segment level dynamics in the modelling process. Other approaches that explicitly attempt to capture segment level dynamics are the dynamical state–space models of Digilakis [14, 15, 16] and fitting acoustic attributes with second order polynomials by Gish and Ng [25]. These approaches will now be described in some detail, since thesis also proposes a segment based dynamic approach. This section will focus on the aspects of these approaches which incorporate dynamic information.

3.2.1 The MIT SUMMIT System

SUMMIT is a speaker–independent, continuous–speech recognition system developed at MIT. SUMMIT explicitly hypothesizes a segmentation structure based on acoustic landmarks in the speech signal. Scoring for classification and recognition is then based on a set of segmental measurements which are determined automatically by a set of *generalized* algorithms [61]. The set of generalized algorithms and the free parameters associated with them form a search space from which the segmental measurements of the acoustic attributes can be optimized. The idea is to extract a set of measurements which will maximize the discriminating power of the system.

The segment level measurements can easily be used to capture different aspects of the dynamics of the acoustic attributes. An example of a measurement which does this is the average change in a spectral peak in a given frequency range, or the average values of an attribute over different time intervals. Statistical models are then created based on mixtures of diagonal Gaussians [60]. More recently, the system has also successfully employed measurements based on the phonetic transitions [62]. The creation and use of statistical trajectory models for these highly dynamic regions will be explored in detail in Chapters 7 and 8 respectively.

3.2.2 Stochastic Segment Model

The description that follows pertains to the more recent SSM implementation as described in [57]. SSM assumes that speech segments are described by a fixed–length sequence of locally time–invariant regions. A deterministic mapping is used to assign each observation vector in a segment to a region. The algorithm theoretically allows for modelling the entire space–time structure of the acoustic attributes, by creating a pdf that accounts for the inter–region correlations (corresponding to the temporal correlations). However, this would result in very high-dimensional pdfs (140 dimensions in [56, 70] and 112 dimensions in [57]), which can not be estimated robustly. As a compromise, the regions are modeled as independent, which is equivalent to assuming that the frames in a segment are conditionally independent given the segmentation. Therefore, separate Gaussian probability density functions are estimated for each region.

This implementation of the SSM is identical to a hidden Markov model with a constrained state sequence. The SSM generally has more regions (eight or ten) than most HMM's used for phonetic recognition (which often use three). In [70] experimental results are reported for implementations which utilize only the spatial correlations (the implementation described above), or alternatively, only the temporal correlations for each acoustic attribute (which assumes independence between attributes). As a baseline, an additional implementation assumed complete independence (diagonal Gaussians for each region). The results for the spatial correlation implementation were significantly superior to the other two experimental conditions. In fact, the implementation utilizing only the temporal correlations performed slightly worse than the complete independence condition. These results provide an interesting contrast to results achieved using the algorithm proposed for this thesis under the same three assumptions.

3.2.3 Dynamic System Model

This work by Digilakis utilizes a traditional state–space dynamical system formalism with standard recursive estimation techniques to create dynamical models of the acoustic attributes. The acoustic attributes are estimated by propagating them over a sequence of stochastic dynamical systems. Each system represents a region of local linearity and stationarity that is assumed to exist in the phone's acoustic trajectory. The regions are considered to be independent of each other. Hence, the trajectory has been assumed to be piecewise linear and stationary. The number of regions for each phone is chosen based on the average duration of the phone over the training set, but in practice was no greater than five. A drawback of this approach is that it can not capture long–term temporal correlations over the course of an entire segment (when more than a single region is used). Also, it should be noted that although this is a segment–based approach, a score is generated for each frame in the utterance.

For each phone, an iterative expectation–maximization (EM) algorithm [10] is used to identify a state–space stochastic model for each region. The driving and measurement noise terms are assumed to be Gaussian and white. A Kalman filter and Rauch–Tung–Striebel smoothing [76] are employed during the estimate step of the EM algorithm. During phonetic classification, the likelihood score for each model is based on the innovations process. Since the innovations are white, no correlation information is lost by assuming independence of the innovations at each frame. This is a major theoretical advantage of this approach. An unfortunate effect of the Kalman filter implementation during classification is that incorrect models generate increasingly accurate estimates of the test trajectory over time. This is due to the inherent stability of the Kalman filters. Consequently, performance in [15, 16] was highly dependent on the accuracy of the initial condition estimates [17]. The results reported in [16] will provide an additional point of comparison with this thesis.

3.2.4 Polynomial Approximations

Gish and Ng [25] have created trajectory models of the acoustic attributes which assume the trajectories can be well represented by first and second–order polynomials. However, the scoring of the error signal (generated by comparing their polynomial templates to actual tokens) does not incorporate any temporal correlations, but assumes the errors at each frame are independent. They generate results for a phonetic classification task on the vowels using the same test set evaluated in parts of this thesis. Hence, their work will provide an additional point of performance comparison at the classification level.

3.3 Previous Performance Results

Many of the approaches described above have been used to generate phonetic classification and/or recognition results using the TIMIT corpus. Therefore, they will provide good benchmarks for comparison as the approaches in this thesis are developed and evaluated. A summary and brief description of some of the previous phonetic recognition results are contained in Table 3.1. It's important to note that the accuracy values in Table 3.1 can not be directly compared to each other. Aside from the fact that different test sets are used, and in one case a different error criterion, the models used in the experiments also reflect different levels of complexity (e.g. some results are based on context–independent phonetic models and others are based on context–dependent phonetic models).

Several comments are required to fully explain the table. The results cited represent the best accuracies reported by the researchers listed. The work by Lamel and Gauvain represents the top phonetic recognition performance for any HMM based approach. They use context–dependent HMM's with ΔMFCC's and ΔΔMFCC's [38]. The work by Robinson represents the best phonetic recognition result on the *NIST Core* data set of the TIMIT corpus in the literature [68] and also includes context–

Approach	**Prin. Researcher(s)**	**Test Set**	**Year**	**Accuracy**
REPN	Robinson	NIST Core	1992	73.9%
HMM	Lamel & Gauvain	NIST Core	1993	69.1%
SUMMIT	Phillips & Glass	NIST Core	1994	68.5%
SSM	Ostendorf & Roucos	BU Test	1990	66.7% $*$
HMM	Lee	KFL Test	1989	66.1%
Dynamical System	Digilakis	BU Test	1992	63%

Table 3.1: Phonetic recognition accuracy results for several approaches using the TIMIT speech corpus. Phonetic accuracy is defined as %Correct - %Insertion Errors. All results are based on a set of 39 phonetic labels. The"$*$" denotes that the SSM algorithm results were computed using a slightly different criterion, to be described later in the thesis. In many instances the results are not directly comparable since different test sets and different levels of model complexity were employed (e.g. some results are based on context–independent phonetic models and others are based on context–dependent phonetic models). The results stated reflect the best accuracies currently reported in the literature for each of the approaches listed.

dependent information.

The implementation of the SUMMIT system included context–independent phonetic models and context–dependent models of the phonetic transitions. The results of Ostendorf and Roucos are based on a test corpus composed entirely of male speakers from the same dialect region of the United States (western dialect) [13, 15]. Also, a slightly different means of calculating errors was used. The SSM results are based on a context–independent phonetic recognition system. No context–dependent phonetic recognition results are currently available. The results of Digilakis, likewise, are based on context–independent results as context–dependent experiments were not conducted. Although Digilakis used the same test set as the SSM work, his method of counting errors was in line with the other non–SSM results. His techniques, experiments, and performance are described completely in his doctoral thesis [15]. Finally, the work by Kai–Fu Lee is a standard early benchmark. Lee used an HMM approach, and conducted both context–independent and context–dependent experiments. The results listed here represent his best context–dependent performance [40].

3.4 Chapter Summary

Speech recognition systems have evolved from temporal based approaches to stochastic approaches. Although much work has been done on segment based approaches, neural networks, and attempts to explicitly incorporate segment level dynamics, HMM's remain the current dominant algorithm for performing the phonetic and word recognition tasks. However, it is widely recognized that their ability to accurately represent spectral level dynamics, or model the within–segment temporal correlations leaves room for improvement. A stochastic, template based approach which is able to capture the dynamic behavior of the acoustic spectra, and the temporal correlations within a segment, is clearly worth consideration as a method of performing speech recognition.

Chapter 4

Acoustic Trajectory Models

This chapter focuses on the *track* component of the statistical trajectory model. The purpose of the track is to accurately capture the dynamic behavior of the acoustic attributes over the duration of a designated unit of speech. The designated unit could be a phone, a sequence of phones, or a specific transition from one phone to another.

Due to the continuous movement of the articulators, the acoustics of the speech signal generally exhibit strongly non–linear dynamic characteristics, both within a phonetic segment and across phonetic boundaries. Studies conducted by Digilakis [15] using a regression analysis indicated that even within a given speech segment, linear models of the spectral dynamics show significant degradation with respect to non–linear models. Intersegmentally, Digilakis concluded that linear models were not adequate. One means of solving this problem is to assume a concatenation of piece–wise linear models. A difficulty with this type of approach is that it artificially constrains the resulting model into a fixed parametric framework. The idea behind the creation of a track is to represent the non–linear dynamics of the acoustic attributes directly by utilizing a non–parametric representation and minimizing the number of potentially constraining assumptions.

The remainder of this chapter will:

1. Define and describe the tracks which will be used to model the spectral motion that occurs within a speech segment.

2. Develop a metric for measuring how accurately a track, or any other dynamic representation, is capturing the phonetic spectral trajectories. This metric will be the mean *distortion*, an average weighted Euclidean distance measure.

3. Examine a variety of algorithms for creating tracks based on different assumptions concerning the variability of a phone's duration. These algorithms will be called generation functions.

4. Select a particular generation function and its associated design parameters based on performance as measured by the distortion metric.

All of the training done in this chapter utilized the *MIT Train* data set of 567 speakers. For the evaluation of the distortion metric, the *MIT Test* set of 50 speakers was used. The 61 phones used in the TIMIT corpus were grouped into 58 models for training. That is, some models combined the data from two or more phones, and no effort was made to distinguish phones whose data were pooled. The 58 models are shown in Table 4.1.

4.1 Dynamic Tracks

A track, $\vec{T}_\alpha$, represents the temporal trajectory of the acoustic attributes in the acoustic space. A track consists of a sequence of *states* which serve as a temporal representation within which a template of a specific unit of speech is stored. In this chapter the focus will be on constructing tracks for individual phonetic units.

The function of the tracks is to account for the dynamics of the phonetic units being modelled. The tracks will form the basis for computing an error between

1. ɑ	10. b	19. n̩	28. ɪ	37. ŋ,ŋ̩	46. š	55. w
2. æ	11. b□	20. ▯	29. ᵻ	38. r̃	47. h#,□	56. y
3. ʌ	12. č	21. ɝ	30. i	39. o	48. t	57. *z*
4. ɔ	13. d	22. e	31. ǰ	40. ɔʸ	49. t□	58. *ž*
5. ɑʷ	14. d□	23. f	32. k	41. p	50. θ	
6. ə	15. ð	24. g	33. k□	42. p□	51. ʊ	
7. əʰ	16. ɾ	25. g□	34. l	43. ʔ	52. u	
8. ɚ	17. ɛ	26. h	35. m,m̩	44. r	53. ü	
9. ɑʸ	18. l̩	27. ɦ	36. n	45. s	54. v	

Table 4.1: Phone models created from training data for all experiments. When more than a single symbol appears, it means that a single model was created by pooling the data for both symbols.

each phonetic model and an hypothesized segment of speech. The error will then be processed to determine the identity of the speech segment. A block diagram of this component of the system is shown in Figure 4-1.

Specifically, the tracks are computed from the training data by mapping the training tokens for each phone to a sequence of M states. The mapping function is known as a *generation function*. When all the tokens in the training set for a particular phone have been mapped, the phone dependent track is calculated from the maximum likelihood estimate of each state.

Once the tracks have been created they will serve as the initial stage in evaluating hypothesized speech segments. To evaluate an N frame speech segment, S, a *synthetic segment,* G is generated. The generation function, f, is used to compute the mapping from the M state track to the N frame synthetic segment. The synthetic segment produced by the generation function is then compared directly to the N frame acoustic segment to form an error sequence:

$$E = S - G = \{\vec{e}_1, \ldots, \vec{e}_N\} \tag{4.1}$$

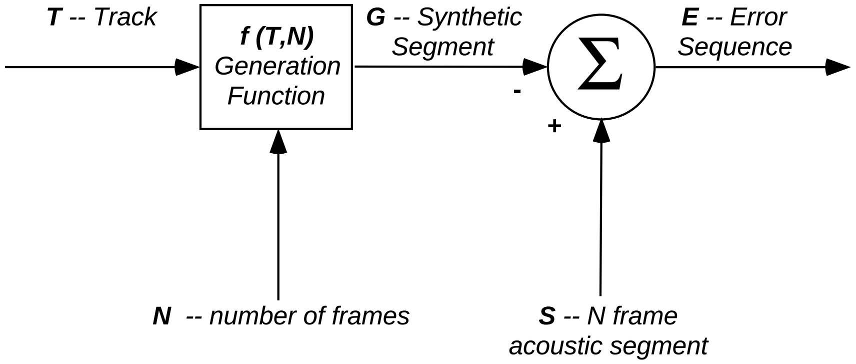

Figure 4-1: Block diagram of the system components associated with the track.

where,

$$\vec{e}_i = \vec{s}_i - \vec{g}_i. \tag{4.2}$$

Note that the generation function used to map the track to an hypothesized number of frames is the same function that is used in the creation of the track. Hence, it is the generation function which determines both the computation of the tracks and their alignment with the speech segments during evaluation. In order to evaluate the accuracy of the generation functions and the associated tracks, it is necessary to develop a means of quantifying the errors which arise from each alternative algorithm. The derivation of a tool for accomplishing this is the topic of the next section.

4.2 Track Evaluation Utilizing a Distortion Metric

Given the variety of methods available to generate the tracks, it is important to develop a metric by which the magnitude of the track error can be measured, independent of the implementation methodology, so that the performance of alternative modelling techniques can be directly compared. Clearly, a track with a more accurate dynamic representation is desired and should lead to superior system performance. The error for a particular token will consist of a temporal sequence of vectors, each computed by taking the difference between a token's vector of acoustic attributes and the track with which it is being compared (see Eq's. 4.1, 4.2). Naturally, a more accurate track will result in error vectors of smaller magnitude than a less accurate track.

To quantify the quality of a track, a *distortion* metric will be utilized. The distortion metric will be based on the Euclidean distance of a token from a synthetic segment. This distance is equivalent to the magnitude of each of the error vectors in the error sequence. However, the magnitude suffers from the fact that the different acoustic attributes have widely diverse mean values. This has the undesired effect of allowing the error in a single dimension (i.e., a particular acoustic attribute) to dominate the distortion metric. Therefore, the variances of the acoustic attributes are used as a means of normalizing each dimension, so that the importance of each component of the error vector is reflected accurately in the distortion metric.

First, the variance of each acoustic attribute is measured indendently using the training set for each phone α for which a model is to be created. That is, given M_α tokens of training data for phone α, and letting $d_\alpha(i)$ be the duration of the i^{th} token (in frames), the total number of frames in training set α is:

$$F_\alpha = \sum_{i=1}^{M_\alpha} d_\alpha(i) \tag{4.3}$$

Thus, the mean value of each acoustic attribute, $C_{\alpha i}$, in training set α is:

$$\overline{C}_{\alpha i} = \frac{\sum_{j=1}^{M_\alpha} \sum_{k=1}^{d_\alpha(j)} C_{\alpha i_{jk}}}{F_\alpha} \tag{4.4}$$

Now the variance of each acoustic attribute in training set α, $\sigma^2_{\alpha i}$, can be calculated:

$$\sigma^2_{\alpha i} = \frac{\sum_{j=1}^{M_\alpha} \sum_{k=1}^{d_\alpha(j)} (C_{\alpha i_{jk}} - \overline{C}_{\alpha i})^2}{F_\alpha} \tag{4.5}$$

The second step is to use these phone dependent variances (which we will need later), to compute the variances for each acoustic attribute over the entire training set. These *pooled* variances will then be used as the normalization weights for the distortion metric. The variances over the entire training set for the i^{th} acoustic attribute are then the weighted mean of the individual $\sigma_{\alpha i}^2$:

$$\sigma_i^2 = \frac{\sum\limits_{\forall\alpha} F_\alpha \sigma_{\alpha i}^2}{\sum\limits_{\forall\alpha} F_\alpha} \tag{4.6}$$

Now the distortion can be defined: *Distortion is the mean value (per frame) of the weighted square magnitude (Euclidean norm) of the error vectors, $\vec{e_i}$. Essentially the distortion is a normalized mean square error metric.* The error vectors, $\vec{e_i}$ are formed by taking the difference between each token and its corresponding synthetic segment for the entire set of data for which the distortion will be measured. Note that all the generation functions will be defined such that the mean error is zero (i.e., $E[\vec{e_i} = \vec{0}]$). The normalization weights are the variances σ_i^2 computed over all the frames in the training data.

The distortion for a particular phone's training set, given that there are P acoustic attributes in each observation vector, is:

$$D_\alpha = \frac{1}{F_\alpha} \sum_{i=1}^{F_\alpha} \sum_{j=1}^{P} \frac{e_{ij}^2}{\sigma_j^2} \tag{4.7}$$

and the distortion over the entire training set is then the weighted mean of the D_α's:

$$D = \frac{\sum\limits_{\forall\alpha} F_\alpha D_\alpha}{\sum\limits_{\forall\alpha} F_\alpha} \tag{4.8}$$

By computing the mean distortion over an evaluation set for each competing algorithm we can determine the relative merits of the different alignment schemes. This will allow us to incorporate the most accurate alignment scheme during subsequent experiments. Furthermore, this metric is valid for comparing completely different types of algorithms, providing each algorithm accounts for the same frames of input speech.

4.3 Track Alignment

Regardless of the method used to generate a dynamic model of the acoustic attributes, a key question that must be answered is how to map tokens of varying duration to a track. The fact that the same phone will have a large variability in its duration, even when spoken by the same speaker in the same context, must be accounted for in a robust manner. In a frame based approach such as HMM's, this durational variability is handled implicitly, since the duration that a phone remains in a state is variable. A segmental approach must deal explicitly with the temporal variability that occurs in the realization of a phone. Different hypotheses which account for this durational variability can serve as a starting point for determining the nature of the dynamic representation.

Clearly, contextual factors and individual speaker characteristics will impact both the articulatory gestures and the resulting spectral trajectories. These issues will be addressed separately. The assumptions which determine the creation of the tracks and their subsequent use will be motivated solely from consideration of the durational variability. Two simple contrasting assumptions that can be made concerning the durational variability of phonetic segments are:

1. The spectral dynamics involved in realizing an acoustic segment are invariant with duration. Differences in duration primarily reflect differences in speaking rate. Therefore, the trajectory followed by the acoustic attributes is the same. Generation functions which utilize this assumption will be defined as *trajectory invariant* generation functions. Trajectory invariant generation functions will rescale the phonetic track in time, until it is of the same duration as the training

or evaluation token. Trajectory invariance, as defined here, need not imply that the articulatory gestures are invariant, only the resulting dynamics of the acoustic attributes.

2. The spectral dynamics involved in realizing an acoustic segment are not invariant with duration. Differences in duration reflect actual differences in the trajectories of the acoustic attributes through the acoustic space. In this case, the assumption is that the dynamics in shorter phones are identical to part of the dynamics expressed in longer phones, such as the initial, central, or final portion. Generation functions based on this assumption will be defined as *time invariant* generation functions. Time invariant generation functions align all tokens about a fixed reference point in time. Therefore, unlike the trajectory invariant functions, there is no temporal warping of the acoustic trajectory. Instead, the trajectory of the acoustic attributes through the space varies with phone duration.

The term trajectory invariance has been borrowed from Digilakis [14], and its use here is very similar to the way it is used in the Dynamic Systems model, allowing for obvious differences in the two approaches. Digilakis' second term was called correlation invariance. This term came about directly from assumptions Digilakis was making concerning the correlation structure of the attributes, namely that the correlation in consecutive states does not change with segment duration and is different from the use of time invariance in this work.

It should be noted that all of the generation functions which follow use a deterministic mapping of the track states to the frames of a synthetic segment. That is, the mapping is independent of the data, or input token. This is in contrast to the DTW approach in which a Viterbi alignment procedure is used to minimize a distance metric between the template and the input token. The advantage of DTW is that it accounts for variations in speaking rate. However, as discussed in Chapter 3, DTW has generally been applied at the word level. While the speaking rate over the course of a word may change significantly, the speaking rate within a phonetic segment is likely to be more constant. Also, performing a Viterbi search while aligning each phonetic model imposes an additional computational burden.

Therefore, although generation functions which are dependent on the input token are not investigated in this work, a DTW type generation function is a potential refinement to the statistical trajectory model approach. This type of generation function would also be fully compatible with the statistical model component which is discussed in the next chapter.

4.3.1 Time Invariant Generation Functions

The tracks are constructed by aligning each token about a reference point in time. Once the training tokens are all aligned, the mean value of each acoustic attribute is computed from the ensemble of tokens which contribute at each point in time. The result is a track for each phonetic model of length equal to the duration of the longest token in the training set for that phone. The tracks tend to be very smooth where many tokens contribute and noisier where fewer tokens contribute (those tokens of unusually long duration for a given phone).

Generation functions which align the training tokens at their mid–point (center), left endpoint (start), and right endpoint (end) are investigated in this section. The algorithm to compute the center aligned tracks is shown in Table 4.2. The algorithms for the start and end time aligned tracks are conceptually identical to the center alignment algorithm, only the initial and final alignment points differ. There is no need to compute a mid–point in these cases, since it is not needed for alignment.

Examination of the resulting tracks gives some insight into the types of dynamical events captured by these generation functions. The cepstral coefficients (without the Mel–scale warping) are defined to be:

$$C_n = \frac{1}{2\pi}\int_0^{2\pi} \log|X(e^{jw})| e^{jwn} dw = \frac{1}{\pi}\int_0^{\pi} \log|X(e^{jw})| \cos{(wn)} dw \qquad (4.9)$$

```
1. $\forall$ phone models, $\alpha$
2. Set all elements of $\vec{T}_\alpha$ and count to zero
3. long($\alpha$) = $\max_{i \in \alpha}$ [duration($i$)]
4. mid_point($\alpha$) = long($\alpha$)/2 (division with truncation)
5. For $1 \leq i \leq M_\alpha$
    (a) first_point = mid_point($\alpha$) – duration($i$)/2
    (b) last_point = mid_point($\alpha$) + duration($i$)/2
    (c) For first_point $\leq j \leq$ last_point
        i. $\vec{T}_\alpha(j) = \vec{T}_\alpha(j) + \vec{S}(j$ – first_point)
        ii. count($j$) = count($j$) + 1
6. For $1 \leq j \leq$ long($\alpha$)
    (a) $\vec{T}_\alpha(j) = \vec{T}_\alpha(j)$/count($j$)
```

Definitions:

- M_α $\equiv$ the number of tokens in the training set for phone model α
- Count $\equiv$ vector whose elements keep track of the number of tokens contributing at each point in time
- Duration $\equiv$ vector of size M_α containing the duration (in frames) of each token

Table 4.2: Time Invariant – Center generation function

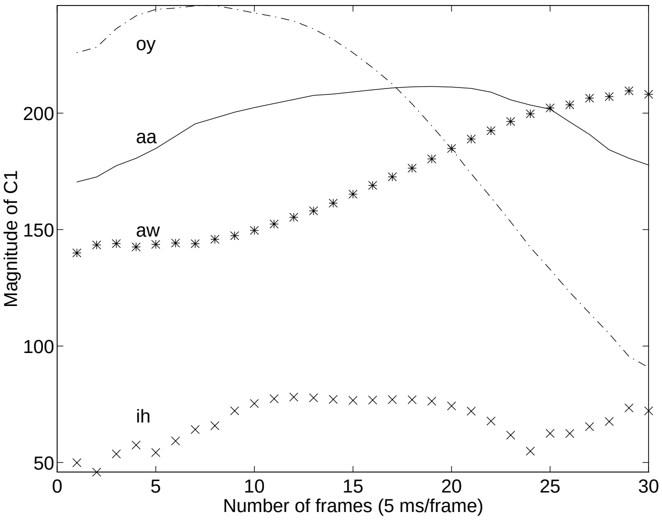

Figure 4-2: C_1 for the four vowels [ɑ] (aa), [ɑʷ] (aw), [ɪ] (ih), and [ɔʸ] (oy) using the Time Invariant – Center generation functions.

where $X(e^{jw})$ represents the Fourier coefficients [66].

Interpreting this equation reveals that C_1 is providing a measure of the spectral balance between high and low frequencies, with low frequencies being weighted positively and high frequencies being weighted negatively. In the case of C_2 the extremes of the frequency range are being weighted positively and the mid–range frequencies are weighted negatively.

Figures 4-2 and 4-3 show the center 30 frames (150 ms) of C_1 and C_2 respectively for four different vowels, using a center based time alignment. The figures reveal that the center alignment algorithm is able to capture some features of the vowel dynamics that are intuitively appealing. For example, in Figure 4-2 the value of C_1 for the vowel [ɔʸ] is initially steady and then drops over time, reflecting the motion of the second and third formants to higher frequencies in the [y] off–glide. In contrast, C_1 rises over time for the vowel [ɑʷ] as the energy falls in frequency due to rounding.

Figure 4-3 also shows interesting dynamical features. In this figure the value of C_2 for [ɔʸ] initially falls as the energy in the vowel moves into mid–ranges, and then

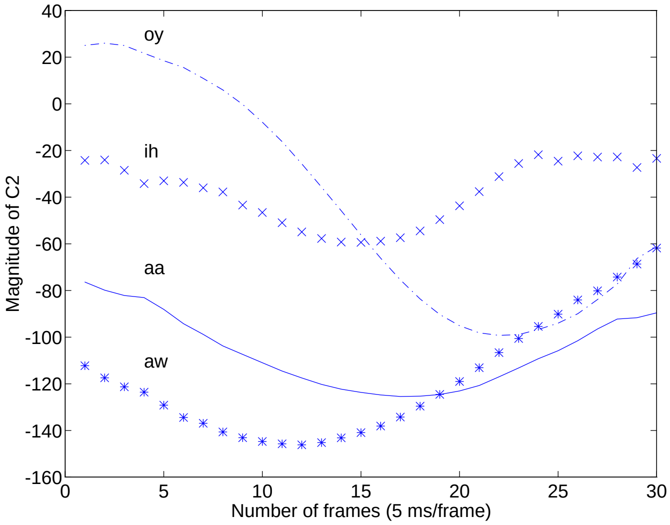

Figure 4-3: C_2 for the four vowels [ɑ] (aa), [ɑʷ] (aw), [ɪ] (ih), and [ɔʸ] (oy) using the Time Invariant – Center generation functions.

begins too rise as the second and third formants continue rising.

The biggest problem with the time invariant techniques is the potential for averaging out significant dynamical events due to improper temporal alignment. For example, this can occur at both the start and end of the track for the center alignment algorithm. The ends of the tracks then tend to incorporate dynamics dominated by durational and contextual factors. Early research on the 16 monothong and diphthong English vowels resulted in slightly superior performance for the center based alignment scheme [26], indicating that for longer duration phonemes, center alignment is the best compromise.

The distortion was then calculated for each of the alignment algorithms over all the phonetic models. To provide a point of comparison, the distortion for a baseline case was also computed. The baseline case uses only the mean value of each of the MFCC's for each phone as a constant, "track." These values are the $\overline{C}_{\alpha i}$ computed earlier in Equation 4.4. The baseline distortion values help to determine the reduction in distortion for each phone achieved by modelling the dynamics. Values for the mean distortions for all sounds using the three time invariant algorithms and the baseline zero'th order MFCC means are shown in Table 4.3. By using the same normalization parameters for all of the phonetic models, it is possible to compare the mean distortion of different phones. It is not surprising to note that [ʔ] (glottal stop) has the highest distortion, since it is so strongly affected by the formants of its contextual environment. The fricatives, most likely due to the consistency of turbulence, and *h#* (silence) have the lowest distortion values.

Table 4.3 also reveals the differences between the three algorithms and the baseline results. For the longer phones with lots of dynamics, such as the diphthongs, the center alignment algorithm is generally superior. The problem with averaging out dynamical events is particularly evident for the voiced and unvoiced stop closures. In these cases, center alignment is a poor idea. The sharp energy transitions which occur at the boundaries will become obscured when the transitions in short duration tokens are averaged with periods of silence in the longer tokens. The reduced distortion values for the start and end alignment algorithms reflect this. The superiority of the end alignment algorithm in these cases is noteworthy. This is due to the fact that although the end alignment algorithm loses information during the initial energy transition, it creates a very accurate model of the final energy transition. The release of a stop is very abrupt and occurs over a shorter period of time than the decrease in energy that occurs during the preceding closure. The closure can often be gradual, thus accounting for the advantage of end point alignment. Figure 4-4 shows the three generation functions for a model created by pooling all of the voiced closure data. Over all the phones, the performance of the center and end based generation functions are close and only slightly better than that of the start generation function.

Two important points regarding these generation functions need to be made. The first is that, although the reduction in distortion over the baseline might appear to be small, it is in fact very significant. In most cases the reductions achieved are roughly an order of magnitude greater than the standard deviation of the estimate of the zero'th order distortion. This finding will also be apparent in the classification experiments presented in the next chapter. Secondly, it will turn out that each of the

phn	*base*	*start*	*center*	*end*	*phn*	*base*	*start*	*center*	*end*
ɑ	18.390	18.221	18.214	18.175	ɪ̵	17.222	17.035	16.995	17.045
æ	17.039	16.927	16.778	16.737	i	16.223	15.830	15.923	16.058
ʌ	18.498	18.452	18.260	18.241	ǰ	10.064	9.644	9.558	9.475
ɔ	21.136	20.943	20.917	20.859	k	14.718	14.296	14.126	14.019
ɑ^{w}	20.446	19.573	19.266	19.628	k$^{□}$	13.558	12.823	12.865	12.264
ə	17.364	17.278	17.209	17.211	l	18.624	18.472	18.531	18.488
ə^{h}	15.645	15.565	15.523	15.544	m	17.516	17.311	17.383	17.307
ɚ	16.477	16.279	16.231	16.245	n	16.916	16.610	16.760	16.796
ɑ^{y}	17.839	16.720	16.061	16.220	ŋ	18.448	18.127	18.275	18.438
b	12.900	12.614	12.197	11.925	ɾ̃	17.279	17.221	17.138	17.133
b$^{□}$	12.476	12.052	12.354	11.952	o	19.868	19.473	19.206	19.263
č	9.636	8.955	8.727	8.719	ɔ^{y}	20.930	19.226	18.524	18.948
d	12.913	12.492	12.488	12.530	p	11.116	10.854	10.642	10.303
d$^{□}$	13.952	13.343	13.560	12.939	□	12.544	12.399	12.410	12.209
ð	13.304	13.268	12.870	12.198	p$^{□}$	9.269	8.462	8.542	8.395
ɾ	15.630	15.496	15.504	15.439	ʔ	24.307	24.153	24.101	23.905
ɛ	17.866	17.712	17.614	17.611	r	18.847	18.797	18.692	18.512
ḷ	15.730	15.348	15.412	15.513	s	9.713	9.416	9.440	9.435
ṃ	16.148	15.846	15.965	16.263	š	9.375	8.985	8.940	8.857
ṇ	15.492	15.291	15.430	15.541	t	11.732	11.034	10.759	10.828
ŋ̣	21.749	21.796	21.910	21.930	t$^{□}$	13.106	12.601	12.530	11.847
□̣	12.137	12.323	12.458	12.356	θ	9.428	8.998	8.882	8.903
ɝ	19.330	19.097	19.013	19.018	ʊ	19.583	19.506	19.517	19.513
e	17.097	16.131	15.928	16.151	u	21.105	20.494	20.470	20.719
f	7.807	7.526	7.320	6.973	ü	17.515	17.192	17.080	17.275
g	15.864	15.572	15.157	14.887	v	11.612	11.099	11.370	11.511
g$^{□}$	16.630	16.107	16.197	15.633	w	17.415	17.329	17.065	16.487
h#	9.267	9.163	9.179	9.188	y	16.473	16.443	16.212	16.134
h	15.335	15.226	15.127	14.837	z	11.759	11.185	11.336	11.596
ɦ	16.284	16.284	16.227	16.154	ž	10.888	10.484	10.356	10.458
ɪ	18.196	18.180	18.001	17.994					
Weighted average over all phones						**14.869**	**14.545**	**14.488**	**14.431**

Table 4.3: Distortions for baseline and time invariance algorithms.

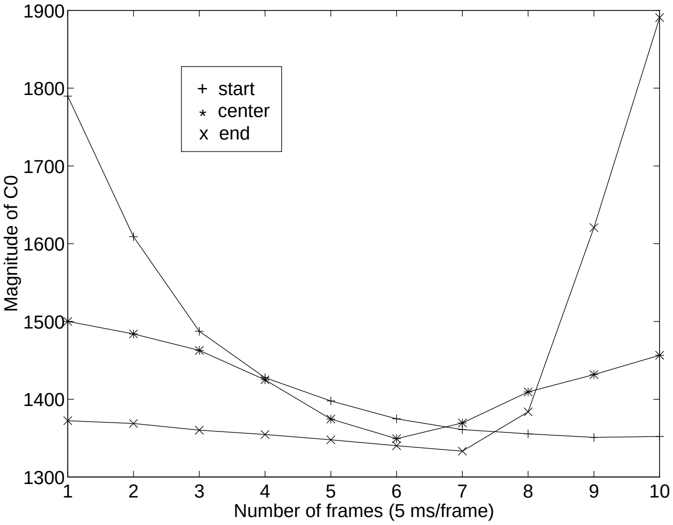

Figure 4-4: C_0 for a Time Invariant voiced closure.
The figure shows the tracks estimated by pooling the three voiced closures. The tracks have 10 states and are shown for each of the three Time Invariant generation functions. The center generation function fails to capture the initial and final transitions, which are averaged out. The time invariance tracks appear to lose important dynamic information for the closures. Note that the transition from silence to the stop is more abrupt than the initial transition to silence.

generation functions is most accurate at their alignment point (start, center, or end). It will be shown that algorithms motivated by a trajectory invariant assumption will capture the accuracy of all three of these algorithms.

4.3.2 Trajectory Invariant Generation Functions

Trajectory invariance assumes that the trajectory through the acoustic–space does not vary with the duration of a specific phonetic unit. Under this assumption, tracks will consist of a fixed sequence of vectors. Each vector can be thought of as a *state*, and hence, the track can be considered to be a sequence of states that the phone is modelled as passing through. Short phones will be aligned to a subset of the track states and long phones will be aligned with the same state more than once. Generation functions may also align observations in between states via interpolation. Variations of each of these approaches will be investigated in this section.

It is relevant to note that many HMM implementations can be interpreted as generating a piecewise constant track consisting of the expected value of the attributes in each state. Accordingly, the track is constant over the duration that a model remains in a particular state. These HMM's can be considered to be a form of trajectory invariance, in that the acoustic attributes pass through a sequence of states in which there is some degree of temporal expansion or compression. Note that the trajectories associated with the HMM's could also be subjected to the same distortion metric defined above.

As in the time invariance case, the trajectory invariant generation function determines the mapping of the track to the input token during both training (when the track is computed) and evaluation. Five alternative mapping algorithms are investigated in this section. In the first four algorithms, each frame of the input token is utilized exactly once, both during track creation and evaluation. The fifth algorithm is distinct in that data in long duration tokens is sub–sampled, and data in short tokens is augmented by interpolation. This allows each token to contribute exactly one data point to each state of the track. A brief description of the five approaches follows:

- *Traj1 — Linear Interpolation with Fixed Endpoints:* This algorithm is based on a linearly interpolated mapping of a token's frames to the frames of the track. The initial and final frames of the token are always aligned with the initial and final frames of the track with intermediate frames falling linearly in between. If the token is longer than the track, the same procedure is followed, but some frames of the track are mapped to more than one frame from the token. This means that multiple frames of the token are averaged into the same track frame for longer tokens.

- *Traj2 — Fractional Linear Interpolation with Fixed Endpoints:* This algorithm preserves the mapping of endpoint to endpoints, but smoothes out the contribution of interior points. Instead of mapping each frame of an input token to a particular frame of the track, the frame contributes its data to adjacent track frames in proportion to how closely it maps to each frame.

- *Traj3 — Linear Interpolation with Fictitious Endpoints:* This algorithm uses linear interpolation, but creates fictitious endpoint frames for both the input token and the track, which always map to each other. The effect of this is that the actual first frame of the input token is mapped to the interior of the track. For tokens of duration longer than the track, this algorithm reverts to Traj1. The idea is to create a slight compromise between this algorithm and the time invariant algorithms.

- *Traj4 — Fractional Linear Interpolation with Fictitious Endpoints:* This algorithm combines the features of Traj2 and Traj3. For longer input tokens, it reverts to the Traj2 mapping algorithm.

- *Traj5 — Fixed Duration Synthetic Segment:* This approach is unique in that data is sometimes created (via interpolation) or ignored (due to subsampling). Rather than map each frame of the input token to a state or combination of states, the input token is stretched or compressed until it has the same duration as the fixed duration track. If the token is shorter than the track duration, it is expanded in time via a linear interpolation with the endpoints mapping to the endpoints of the track. If the token is long, then it is linearly compressed in time, the endpoints are again lined up, and the data is down–sampled. Consequently, each token in the training set contributes exactly one frame to each state of the track.

Analysis of Trajectory Invariant Generation Functions

The Traj1 algorithm is defined in detail in Table 4.4. Examples of mapping the frames of input tokens of different duration into the track are given in Table 4.5. One problem with this algorithm is that, depending on the number of states in the track and the typical durations of the tokens it is representing, consecutive states of the track can receive disproportionate amounts of the training data due to the effects of mapping the frame to the nearest state.

The Traj2 algorithm, which preserves the endpoint mapping and refines the linear interpolation, can resolve this problem. The algorithm, which is defined in Table 4.6, spreads out the contribution of an individual frame to both of the track states that it “falls between.” For example, a token’s frame whose mapping to the track is 3.75 would contribute 25% of its value to the track’s third state and 75% of its value to the track’s fourth state. This results in smoother tracks, particularly for the short duration phones. Figure 4-5 shows the tracks of C_0 for the phone [b] generated with the Traj1 and Traj2 algorithms. For longer phones, such as the vowels, semi-vowels, and strident fricatives, the tracks are indistinguishable. This is shown in Figure 4-6 which depicts C_0 and C_1 for the phone [ɑ^{w}].

The Traj3 approach permits some flexibility in mapping the endpoints of the trajectory. The idea is to create two “fictitious” endpoints of the trajectory, one at the start and one at the end. Each token utilizes these fictitious endpoints, which always map directly to the corresponding states of the track. Then the interior points are linearly interpolated. For tokens of duration equal to or longer than the

```
1. $\forall$ phone models, $\alpha$
2. Set all elements of $\vec{T}_\alpha$ and count to zero
3. num = track_duration – 1
4. For $1 \leq i \leq M_\alpha$
   (a) den = duration($i$) – 1
   (b) For $0 \leq j <$ duration($i$)
       i. track_index = round_to_nearest_integer($j$ * num/den)
       ii. $\vec{T}_\alpha$(track_index) = $\vec{T}_\alpha$(track_index) + $\vec{S}(j)$
       iii. count(track_index) = count(track_index) + 1
5. For $0 \leq j <$ track_duration
   (a) $\vec{T}_\alpha(j) = \vec{T}_\alpha(j)$/count($j$)
```

Definitions:

- Track_duration $\equiv$ pre-specified duration (in frames) to be used for this track
- M_α $\equiv$ number of tokens in the training set for phone model α
- Count $\equiv$ vector whose elements keep track of the number of tokens contributing at each point in time
- Duration $\equiv$ vector of size M_α containing the duration (in frames) of each token

Table 4.4: Trajectory Invariant Generation Function I – Traj1

Track of 10 States	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
token with 2 frames	0									1
token with 4 frames	0			1			2			3
token with 8 frames	0	1		2	3	4	5		6	7
token with 12 frames	0	1	2,3	4	5	6	7	8,9	10	11
token with 16 frames	0	1,2	3,4	5	6,7	8,9	10	11,12	13,14	15

Table 4.5: Trajectory Invariant Generation Function I – Example frame mappings for tokens of different durations to a 10 state track. The frame numbers of the input token are mapped to the indicated states of the track.

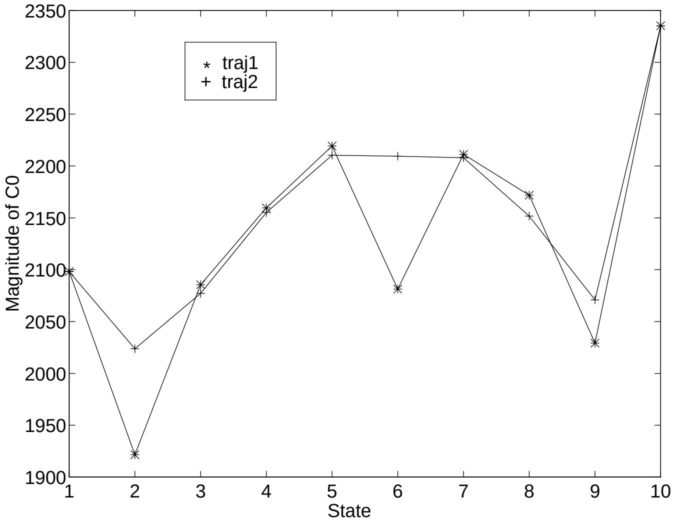

Figure 4-5: C_0 for the labial stop release [b] using Traj1 and Traj2 with a 10 state track.

```
1. $\forall$ phone models, $\alpha$
2. Set all elements of $\vec{T}_\alpha$ and count to zero
3. num = track_duration – 1
4. **For** $1 \leq i \leq M_\alpha$
    (a) den = duration($i$) – 1
    (b) **For** $0 \leq j <$ duration($i$)
        i. track_index = floor($j *$ num/den)
        ii. frac = $j *$ num/den – track_index; omfrac = 1 – frac
        iii. $\vec{T}_\alpha$(track_index) = $\vec{T}_\alpha$(track_index) + omfrac $* \vec{S}(j)$
        iv. count(track_index) = count(track_index) + omfrac

            **If** ($j \neq$ den) Do steps v. and vi.
        v. $\vec{T}_\alpha$(track_index + 1) = $\vec{T}_\alpha$(track_index + 1) + frac $* \vec{S}(j + 1)$
        vi. count(track_index + 1) = count(track_index + 1) + frac
5. **For** $0 \leq j <$ track_duration
    (a) $\vec{T}_\alpha(j) = \vec{T}_\alpha(j)$/count($j$)
```

Definitions:

- Track_duration ≡ pre-specified duration (in frames) to be used for this track
- M_α ≡ number of tokens in the training set for phone model α
- Count ≡ vector whose elements keep track of the number of tokens contributing at each point in time
- Duration ≡ vector of size M_α containing the duration (in frames) of each token

Table 4.6: Trajectory Invariant Generation Function II – Traj2

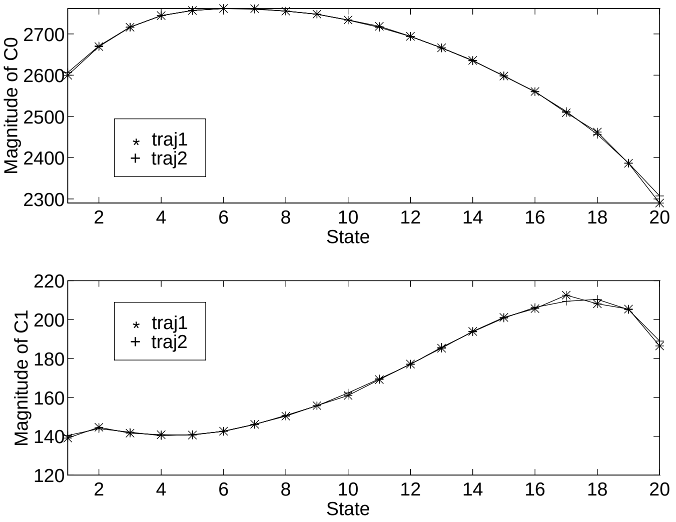

Figure 4-6: C_0 and C_1 for the vowel [ɑʷ] using Traj1 and Traj2 with a 20 state track. The two algorithms virtually coincide.

track, the algorithm reverts to the fixed endpoint approach. The effect on shorter tokens is to map initial and final frames towards the interior of the trajectory. This is a compromise between the previous trajectory invariant algorithms and the time invariant track algorithms. The algorithm is explained in Table 4.7. Examples of the mapping of the input token's frames to the track is shown in Table 4.8.

The Traj4 algorithm is constructed by combining elements of the Traj2 and Traj3 algorithms. The fractional interpolation scheme of Traj2 is used in combination with the fictitious endpoints used in the Traj3 algorithm. The resulting tracks for each of the first four trajectory invariant algorithms are shown for the phone [ð] in Figure 4-7. The Traj4 algorithm effectively smoothes the tracks, but this smoothness will not turn out to generate superior distortion results.

The first four approaches utilize every frame in the input token exactly once. Each frame is mapped to a point in time, or state, in the trajectory, and no data is created or ignored. The fifth and final approach is unique in that data is created by interpolating short tokens and ignored by subsampling long tokens. Rather than

```
1. $\forall$ phone models, $\alpha$
2. Set all elements of $\vec{T}_\alpha$ and count to zero
3. num1 = track_duration – 1
4. num2 = nm1 + 2
5. **For** $1 \leq i \leq M_\alpha$
   (a) den1 = duration($i$) – 1
   (b) den2 = den1 + 2
   (c) **For** $0 \leq j <$ duration($i$)
       i. **if** (duration($i$) < track_duration)
          track_index = round_to_nearest_integer(($j$+1) $*$ num2/den2) – 1
          **else** (revert to Traj1)
          track_index = round_to_nearest_integer($j$ $*$ num1/den1)
       ii. $\vec{T}_\alpha$(track_index) = $\vec{T}_\alpha$(track_index) + $\vec{S}(j)$
       iii. count(track_index) = count(track_index) + 1
6. **For** $0 \leq j <$ track_duration
   (a) $\vec{T}_\alpha(j) = \vec{T}_\alpha(j)$/count($j$)
```

Definitions:

- Track_duration ≡ pre-specified duration (in frames) to be used for this track
- M_α ≡ number of tokens in the training set for phone model α
- Count ≡ vector whose elements keep track of the number of tokens contributing at each point in time
- Duration ≡ vector of size M_α containing the duration (in frames) of each token

Table 4.7: Trajectory Invariant Generation Function III – Traj3

Track of 10 States	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
token with 2 frames				0			1			
token with 4 frames		0		1			2		3	
token with 8 frames	0	1		2	3	4	5		6	7

Table 4.8: Trajectory Invariant Generation Function III – Example frame mappings for tokens of different durations to a 10 state track. The frame numbers of the input token are mapped to the indicated frames of the track.

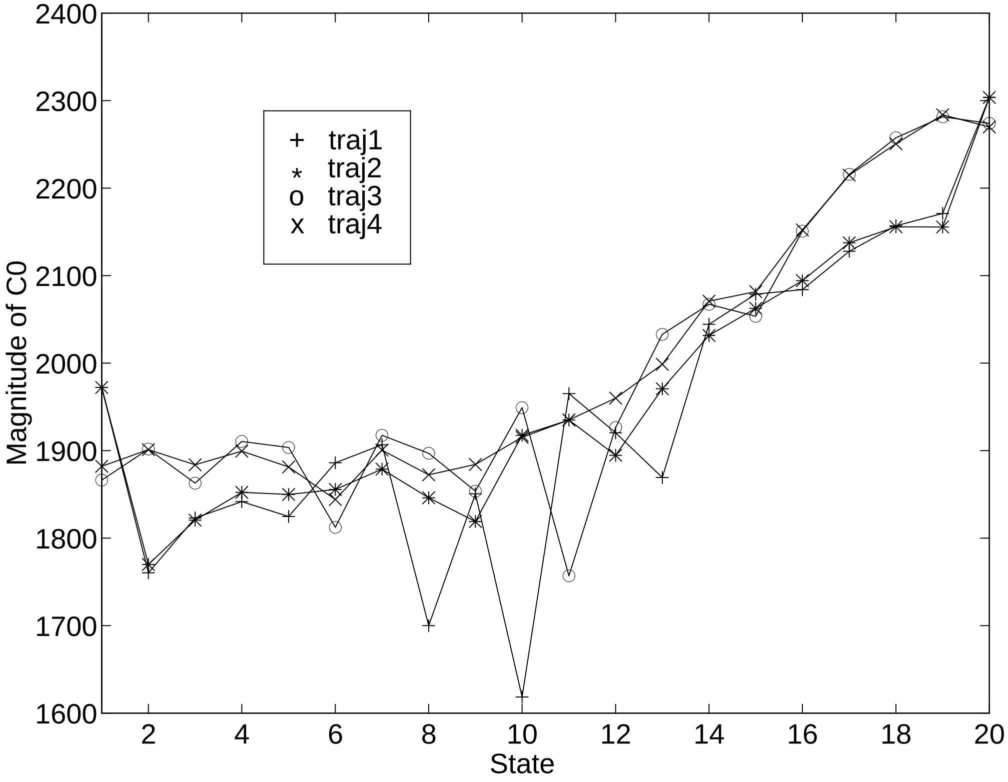

Figure 4-7: Trajectory Invariant tracks for the weak fricative [ð].
This figure shows C_0 for the weak fricative [ð] using generation functions Traj1, Traj2, Traj3 and Traj4 and a 20 state track. The algorithms which have "fractional interpolation" (Traj2 and Traj4) have smoother tracks. However, the algorithms which achieve lower distortions share the "fixed endpoints" feature (Traj1 and Traj2).

map each frame of the input token to the trajectory, the input token is stretched or compressed until it has the same duration as the fixed duration track. If the token has fewer frames than the track has states, it is expanded in time via a linear interpolation with the endpoints mapping to the endpoints of the track. If the token has more frames than the number of states, then it is linearly compressed in time, the endpoints are again lined up, and the data is down–sampled. Hence, each token in the training set contributes exactly one frame to each frame of the track. The algorithm is defined in Table 4.9.

Selecting the Number of Track States

An important issue not yet addressed is choosing the number of states for the trajectory invariant tracks. If too few states are used, then the resulting tracks will not contain all of the relevant dynamical characteristics of the phone being modelled. If too many states are used then unnecessary storage and computation costs are incurred.

To determine the optimal number of states, the mean distortion was computed as a function of the number of states for each phonetic model. The distortions for all the models were then summed, with each model weighted by the number of times it occurred in the training set (the prior probability). Figure 4-8 shows the summed distortion as a function of the number of track states for each of the five trajectory invariance algorithms. It can be seen that the distortion falls off rapidly with the number of states until reaching a steady value at approximately 9 or 10 states. It is also apparent that using too many states does not cause an increase in distortion. The forty–nine speaker development set, *HM Development*, was used for all distortion calculations.

This result was consistent on an individual phone–by–phone basis as well. The mean distortions for 4 individual phones, representing distinctly different phonetic classes, are shown in Figure 4-9. Consistent with the behavior over all the phones, it is apparent that steady–state distortion levels are achieved by using approximately

```
1. $\forall$ phone models, $\alpha$
2. Set all elements of $\vec{T}_\alpha$ and count to zero
3. num = track_duration – 1
4. **For** $1 \leq i \leq M_\alpha$
    (a) den = duration($i$) – 1
    (b) **For** $0 \leq j <$ duration($i$)
        track_index($j$) = $j *$ num/den
    (c) **For** $0 \leq n <$ track_duration
        i. **For** $0 \leq j <$ duration(i)
            A. deltax = $n$ – track_index($j$)
            B. deltay = $n$ – track_index($j$+1)
            C. **If** ((deltax > 0.0) and (deltay < 0.0))
                distance = track_index($j$+1) – track_index($j$)
                ratio = deltax/distance
                $\vec{T}_\alpha$(track_index) = $\vec{T}_\alpha$(track_index) + ratio $* \vec{S}(j+1)$
                + (1 – ratio) $* \vec{S}(j)$
            D. **Else If** (deltax = 0.0)
                $\vec{T}_\alpha$(track_index) = $\vec{T}_\alpha$(track_index) + $\vec{S}(j)$
5. **For** $0 \leq j <$ track_duration
    (a) $\vec{T}_\alpha(j) = \vec{T}_\alpha(j)/M_\alpha$
```

Definitions:

- Track_duration ≡ pre-specified duration (in frames) to be used for this track
- M_α ≡ number of tokens in the training set for phone model α
- Count ≡ vector whose elements keep track of the number of tokens contributing at each point in time
- Duration ≡ vector of size M_α containing the duration (in frames) of each token

Table 4.9: Trajectory Invariant Generation Function V – Traj5

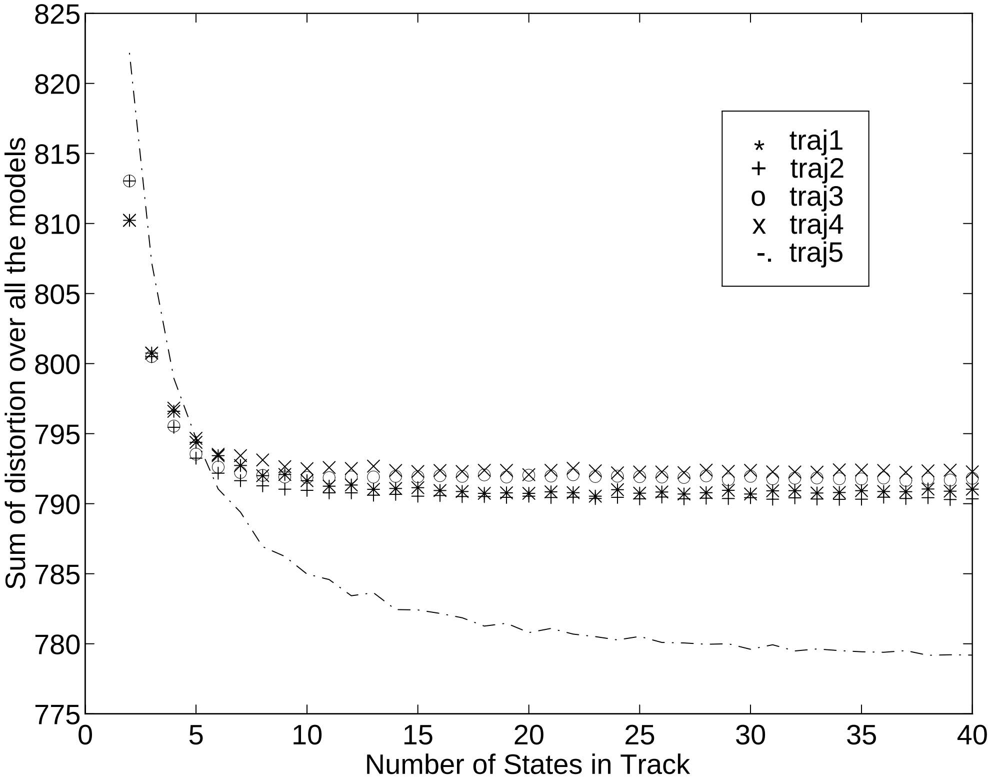

Figure 4-8: Sum of the mean distortion vs. number of states in track. Calculated over all the phones for each of the five trajectory invariance generation functions.

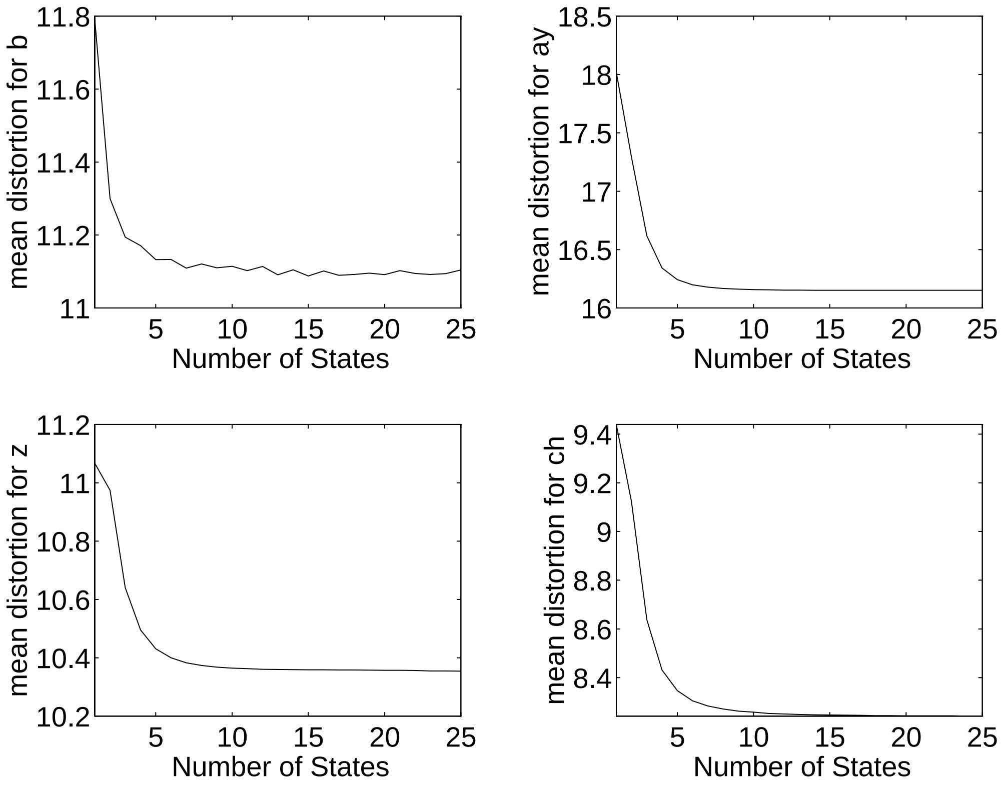

Figure 4-9: Mean distortion vs. number of states for four different phones, computed using the Traj2 algorithm.

10 states in the track. Since even the phone with the shortest mean duration, [b], is not adversely affected by using additional states, it is possible to choose a single track length for all the phonetic models.

An additional issue of importance with respect to phone duration occurs when we examine the initial and final few frames of the trajectory invariant track. Figure 4-10 depicts the synthetic segments generated by the Traj2 algorithm assuming three different input token durations. Although the synthetic segment trajectories look virtually identical, note that the 40 frame synthetic segment assumes a slower transition from the proceeding phone and into the following phone by factors of two and four, when compared to the 20 and 10 frame segments respectively. For example, this means that the transition from a preceding vowel to an [s] is expected to take four times longer if the [s] is 200 ms than if the [s] is 50 ms. This assumption does not reflect intuitive expectations, and could be a potential weakness of the trajectory invariant track generation functions. This problem is addressed later by creating tracks of the transition dynamics directly.

Analysis of Distortion Results

The distortions for the baseline (zeroth order) algorithm and the five trajectory invariant track generation functions, each using a 20 state track, are shown for all the phones in Table 4.10. The first four algorithms are directly comparable, since they all compute the mean distortion over exactly the same set of data points. Direct comparison with the fifth algorithm is not as straightforward, since it is normalized over artificial data, and also eliminates data for the longer phones. Hence, it is not computing comparisons identical to those made by the first four algorithms, all of which deal with identical data. Enumerated below is a summary of some salient points regarding the first four algorithms:

1. For the longer duration segments, such as the vowels and syllabic consonants, there are no statistically significant differences in the distortions. The tracks themselves are virtually indistinguishable. The same is true for six of the eight fricatives, with the exceptions being the two weak voiced (shorter duration) fricatives [ð] and [v].

2. Short duration segments, such as the the voiced and unvoiced stop releases, favor the Traj1 and Traj2 algorithms (fixed endpoints). Traj1 and Traj2 also show very significant performance improvements for the voiced and unvoiced closures, where, as was seen for the time invariant algorithms, endpoint alignment is critical.

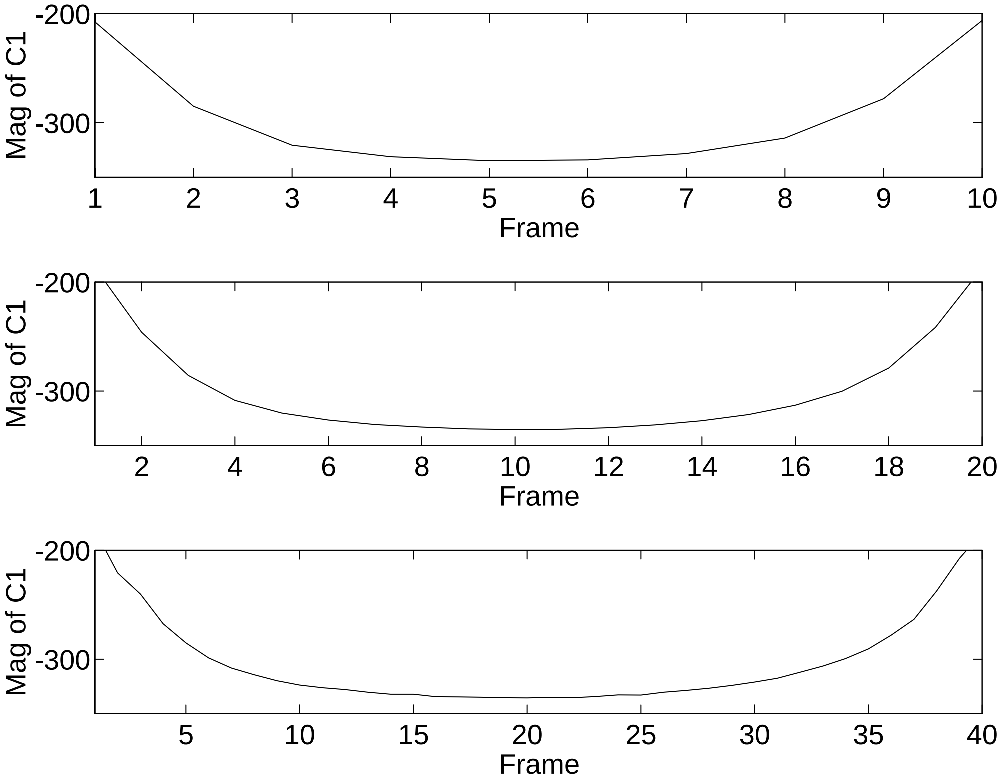

Figure 4-10: Synthetic segments of [s] for different durations.
C_1 is plotted assuming durations of 10, 20, and 40 frames. All synthetic segments were created using the Traj2 generation function. Note that the trajectories, normalized by duration, are virtually identical.

phn	*base*	*Traj1*	*Traj2*	*Traj3*	*Traj4*	*Traj5* *
ɑ	18.390	18.081	18.077	18.079	18.084	18.207
æ	17.039	16.654	16.652	16.651	16.654	16.393
ʌ	18.498	18.104	18.100	18.097	18.106	17.916
ɔ	21.136	20.830	20.828	20.834	20.837	20.995
ɑ^{w}	20.446	19.206	19.206	19.204	19.205	19.078
ə	17.364	17.199	17.194	17.189	17.191	16.773
ə^{h}	15.645	15.488	15.471	15.482	15.520	14.696
ɚ	16.477	16.136	16.134	16.134	16.144	15.694
ɑ^{y}	17.839	15.915	15.903	15.903	15.917	15.938
b	12.900	12.332	12.171	12.237	12.356	11.265
b$^{□}$	12.476	11.563	11.561	11.760	11.764	11.713
č	9.636	8.461	8.450	8.463	8.476	8.080
d	12.913	12.463	12.471	12.558	12.547	11.786
d$^{□}$	13.952	12.466	12.474	12.826	12.813	12.640
ð	13.304	12.519	12.555	12.734	12.704	11.936
ɾ	15.630	15.239	15.259	15.320	15.328	15.070
ɛ	17.866	17.558	17.555	17.546	17.550	17.375
l̩	15.730	15.254	15.254	15.260	15.259	15.345
m̩	16.148	16.243	16.230	16.208	16.235	15.960
n̩	15.492	15.253	15.241	15.268	15.282	14.996
ŋ̩	21.749	21.751	21.751	21.760	21.722	21.614
□	12.137	12.199	12.184	12.267	12.292	11.558
ɝ	19.330	18.867	18.863	18.858	18.861	18.775
e	17.097	15.924	15.918	15.919	15.925	15.683
f	7.807	6.884	6.875	6.880	6.895	6.724
g	15.864	15.062	15.041	15.062	15.095	14.508
g$^{□}$	16.630	15.193	15.205	15.448	15.437	15.156
h#	9.267	9.161	9.153	9.153	9.161	8.943
h	15.335	14.819	14.815	14.872	14.884	14.122
ɦ	16.284	16.146	16.143	16.146	16.162	15.795
ɪ	18.196	17.940	17.935	17.925	17.932	17.846
ɨ	17.222	17.001	17.000	16.987	16.990	16.584
i	16.223	15.849	15.848	15.850	15.851	15.740

phn	*base*	*Traj1*	*Traj2*	*Traj3*	*Traj4*	*Traj5 **
ǰ	10.064	9.252	9.252	9.321	9.336	8.715
k	14.718	13.915	13.917	13.906	13.914	13.520
k□	13.558	11.787	11.785	11.943	11.941	11.554
l	18.624	18.500	18.502	18.530	18.528	18.381
m	17.516	17.148	17.149	17.175	17.172	16.728
n	16.916	16.559	16.560	16.606	16.607	16.499
ŋ	18.448	18.195	18.183	18.220	18.233	18.398
r̃	17.279	16.998	16.977	17.027	17.102	17.162
o	19.868	19.212	19.206	19.202	19.209	19.118
$ɔ^y$	20.930	18.556	18.542	18.540	18.556	18.309
p	11.116	10.330	10.331	10.402	10.402	9.914
□	12.544	12.200	12.206	12.229	12.223	12.995
p□	9.269	7.619	7.624	7.717	7.711	7.386
ʔ	24.307	23.801	23.808	23.835	23.811	22.787
r	18.847	18.577	18.584	18.612	18.609	18.466
s	9.713	9.224	9.219	9.218	9.227	8.967
š	9.375	8.627	8.622	8.622	8.632	8.420
t	11.732	10.629	10.631	10.621	10.618	10.217
t□	13.106	11.587	11.592	11.817	11.819	11.740
θ	9.428	8.658	8.642	8.638	8.653	8.405
ʊ	19.583	19.385	19.384	19.381	19.386	18.950
u	21.105	20.452	20.440	20.440	20.450	20.262
ü	17.515	17.087	17.082	17.084	17.089	16.441
v	11.612	11.041	11.045	11.094	11.078	11.039
w	17.415	16.852	16.865	16.952	16.953	17.403
y	16.473	16.130	16.103	16.098	16.123	15.658
z	11.759	11.013	11.009	11.030	11.036	10.860
ž	10.888	10.073	10.029	10.043	10.081	9.461
	Weighted average over all phones:					
	14.869	**14.276**	**14.273**	**14.303**	**14.308**	**14.096**

Table 4.10: Distortions for baseline and trajectory invariance generation functions. The "∗" denotes that the Traj5 algorithm isn't really directly comparable to the other algorithms listed, and consequently it's superior distortion results may *not* reflect algorithmic superiority.

3. The remaining phones tend to show slight advantages for the Traj1 and Traj2 algorithms. As can be seen, the overall differences are small, with Traj2 coming out on top. This seems to indicate that the endpoint alignment is important, and that the smoothing from "fractional interpolation" is helpful.

4. Note that all the algorithms show significant improvement over the best fixed track results.

5. The results for Traj5, while tantalizing, do not translate into a performance advantage. The reasons for this will be covered when baseline classification experiments are conducted. However, the primary factors appear to be the creation of artificial data for short phones, which creates artificial and erroneous correlation information, and the subsampling of longer phones, when relevant data is ignored. It is surmised that the apparent advantage in distortion is an artifact of this data manipulation.

The advantage of the trajectory invariant approach is made clear when examining Figures 4-11 and 4-12. C_0 is plotted for the start, center, and end fixed track algorithms and for a synthetic segment generated using the Traj2 algorithm. A close examination of the figures reveals that the synthetic segment from the Traj2 algorithm is initially aligned with the start algorithm trajectory, becomes aligned in the middle of the segment with the center trajectory, and over the last several frames is coincident with the end trajectory. This evidence strongly supports the idea that the trajectory invariance assumption is able to capture the more accurate elements of each of the time invariance algorithms.

4.4 Chapter Summary

A distortion evaluation over the development set showed that tracks generated using a trajectory invariance assumption were consistently slightly superior to those generated using a time invariance assumption. The generation function which resulted

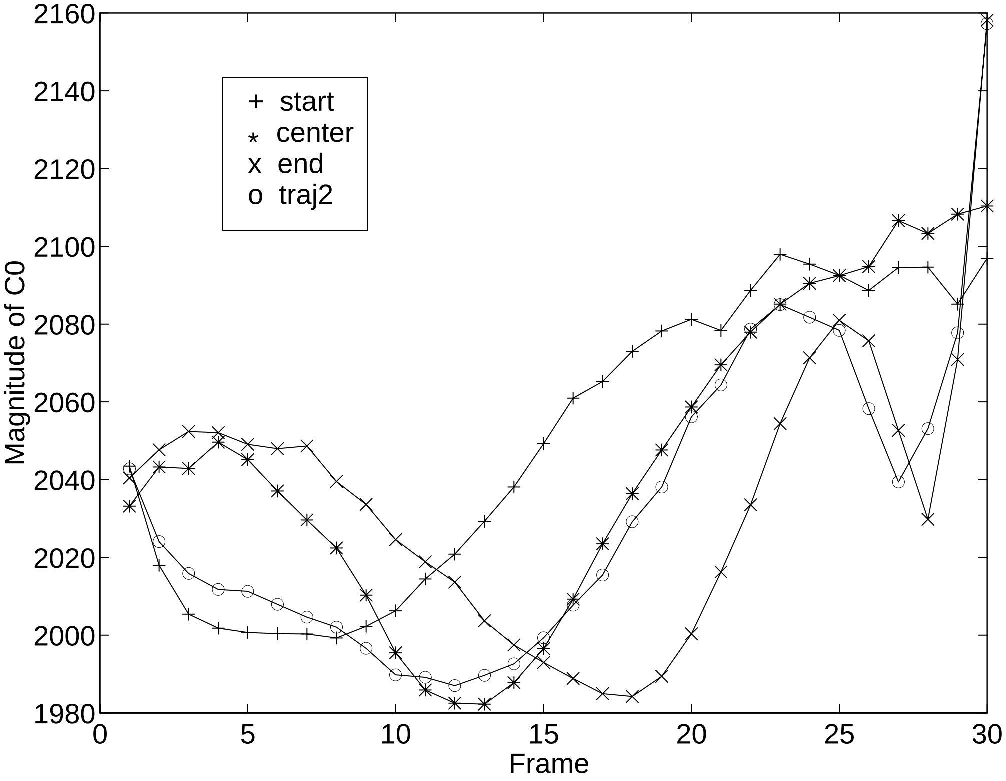

Figure 4-11: C_0 for the three time invariant algorithms and a synthetic segment generated from the Traj2 algorithm of duration 150 ms (30 frames) for the phone [f].

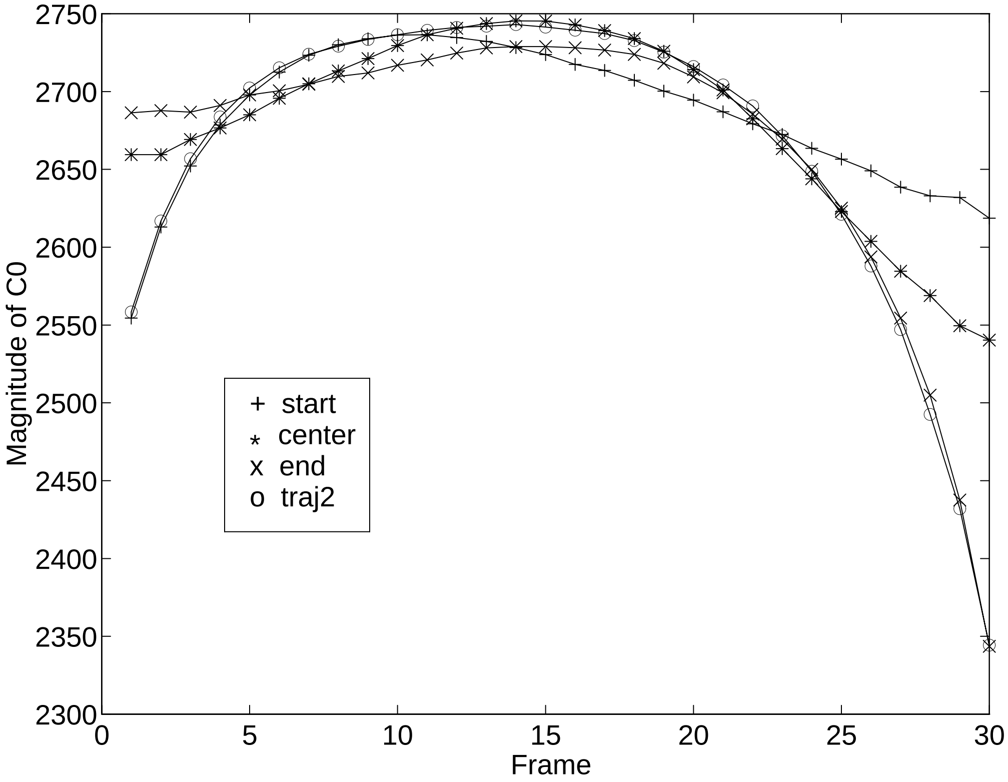

Figure 4-12: C_0 for the three time invariant algorithms and a synthetic segment generated from the Traj2 algorithm of duration 150 ms (30 frames) for the phone [ɑʸ].

in the least distortion for the four algorithms that could be directly compared was *fractional linear interpolation with fixed endpoints.* This generation function is a linearly interpolated mapping of a token's frames to the states of the track. The initial and final frames of the token are always aligned with the initial and final states of the track. This result agrees with work done by Ostendorf, *et al.* where superior performance for SSM was realized with a trajectory invariance type of assumption which included fixing the endpoints [57].

Figure 4-13 shows example trajectories of Mel–frequency cepstral coefficients C_0 through C_3 for a synthetic [ɔ^y] segment and an [ɔ^y] token selected at random from the evaluation set. The synthetic segment accurately captures the dynamic motion of the test token. Note also the temporal correlation of the error over the duration of the segment. Capturing this correlation is a key objective of the error models, which are the topic of the next chapter.

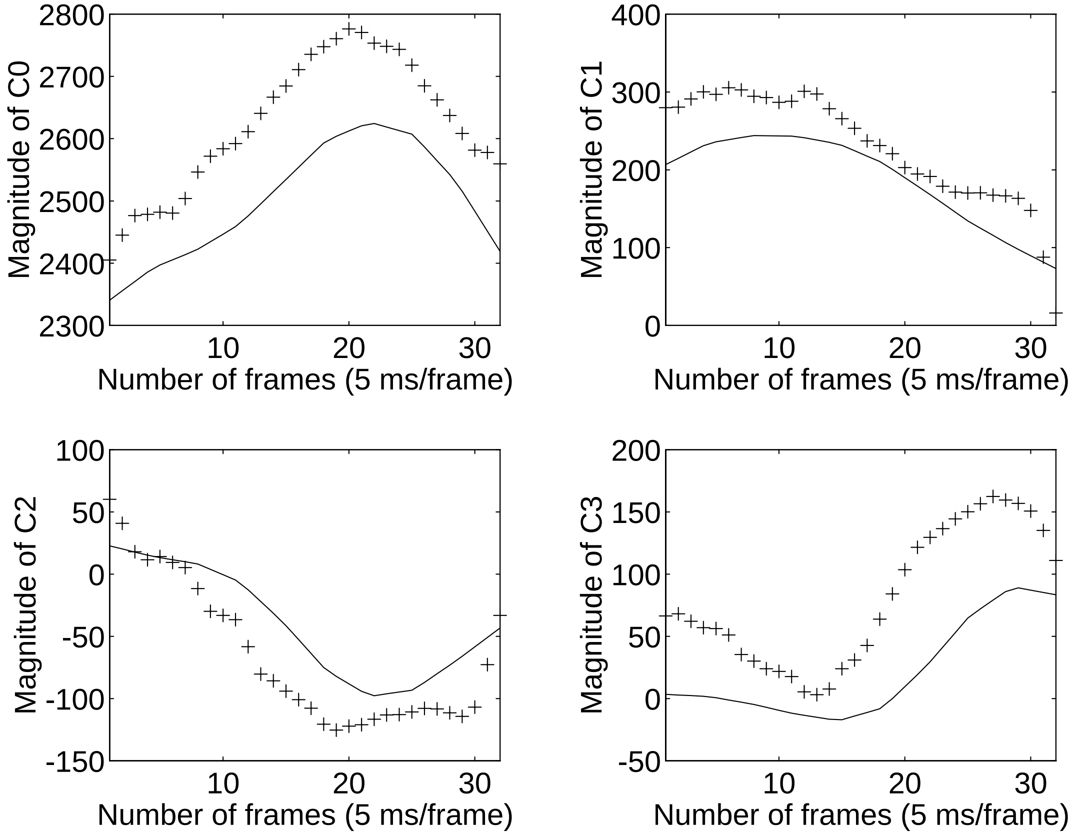

Figure 4-13: Synthetic segment and test token for the phone [ɔʸ].
This figure shows the Mel–frequency cepstral coefficients C_0–C_3 for a synthetic [ɔʸ] segment generated from a ten state track using the Traj2 algorithm (solid), and an [ɔʸ] token (+) randomly selected from the test set. Note that the synthetic segment accurately captures the dynamic motion of the test token. Also note the significant temporal correlations in the error over the duration of the segment.

Chapter 5

Statistical Error Modelling

The speech signal varies slowly enough that successive frames of acoustic attributes are highly correlated in time. Despite this fact, the majority of existing speech recognition systems employ techniques (HMM's) which model the signal as a sequence of conditionally independent observations. The models developed in this chapter will provide a basis for capturing the statistical dependencies of acoustic attributes within a speech segment.

The next section introduces the difficulties inherent in capturing the relevant correlation information and motivates the solution used in this thesis, which involves dividing the error sequence into sub–segments. Then, in section 5.2, an analysis of the key design parameter which determines the number of sub–segments is conducted. The benefits of sub–segmenting the error signal *after* accounting for segmental dynamics are shown by comparing the loss of accuracy to that which would occur if the sub–segmentation had been applied to the original acoustic attributes. This section also provides some analysis of the different types of correlations which are captured. The last section summarizes the results obtained in the chapter.

5.1 Choosing a Statistical Model

The objective of the statistical model is to take advantage of information residing in the correlations both over time and between attributes. Many previous approaches are either not structured to capture all of the relevant correlation information or have been unable to do so in a robust manner. In the work by Digilakis [14, 16] and also by Gish [25], the statistical models are of the errors, which are assumed to be independent. This assumption is appropriate for the stochastic dynamic system model [14, 16] since the Kalman filter produces a white innovations process. For other methodologies however, the simplicity gained by assuming the error sequence is independent is potentially damaging since important correlation information is discarded.

The issue of creating a probability density function which is able to capture correlation information involves two key problems. The first problem is due to the fact that the observation sequence varies in duration. For each segment that is hypothesized, the observation sequence will be N frames long, where N is variable. The second problem arises when the dimension of the distribution becomes large, and the estimate of the covariance matrix parameters becomes difficult due to a lack of training data.

Two approaches can be taken to the first problem. Clearly, if enough training data existed, it would be desirable to create a pdf for each phone, for every possible duration. Since this is not the case, the alternative is to attempt to preserve any acoustical variability that is related to duration by allowing individual sequences to contribute to different parameters of the pdf, or, by normalizing the duration of each sequence so that they all contribute to each element of the pdf parameters.

Some preliminary attempts were made using the first approach early in this thesis work. The idea was to create a single large covariance matrix and to map each contributing sequence to a subset of the mean vector and covariance matrix. The mapping was a function of both the duration and of the generation function used to create the tracks. A principle components analysis was then used to rotate the covariance matrices and extract a lower dimensional representation. Unfortunately this approach never succeeded in achieving performance levels which were competitive with approaches based on normalization. Therefore, the remainder of this chapter will be devoted to investigating an approach based on normalization.

It is interesting to note that all of the segmental approaches discussed in this work have a mechanism for performing duration normalization. The type of normalization used will also directly impact the dimensionality of the pdf and the capability of the approach to estimate it effectively.

The SSM approach [56] provides a framework for capturing temporal and spatial correlations. Most variations of this approach involve performing interpolation on short tokens and sub–sampling or remapping large tokens to produce a fixed length sequence of the acoustic attributes. If the length of this normalized sequence is M frames, then the mapping of the original N frame sequence, S, is similar to a type of generation function:

$$f(S_N) = \tilde{S}_M = \{\vec{\tilde{s}}_1, \ldots, \vec{\tilde{s}}_M\} \tag{5.1}$$

This fixed–length sequence is then concatenated together to form a single high–dimensional vector for estimating a Gaussian pdf

$$\vec{V}_{\tilde{S}} = \{\tilde{s}_{11}, \ldots, \tilde{s}_{1P}, \ldots, \tilde{s}_{MP}\}^T \tag{5.2}$$

The drawback of this approach is that if there are P acoustic attributes, then the resulting dimension of the pdf is PM, which in practice was anywhere from 112 to 140 dimensions. This proved to be too large to allow for robust estimation of the covariance matrix. The approach has been limited to using either only the spatial correlations or spatial correlations in combination with local temporal correlations. However, no implementation utilizing temporal correlations of more than a few frames has been reported in the literature.

A second solution to the problem involves dividing the observation sequence into Q sub–segments of equal duration and averaging the vectors within each sub–segment. For example, for a ten state track with Q equal to three, that part of the error which resulted from comparing the token to the first third of the track (i.e., the first three and a third "states") would be averaged, and so on for each of the other two thirds. The specific operations for a hypothesized segment S_N are:

$$S_N = \{\vec{S}_1, \ldots, \vec{S}_{\frac{N}{Q}} \mid \ldots, \mid \vec{S}_{\frac{N(Q-1)}{Q}+1}, \ldots, \vec{S}_N\}^T \tag{5.3}$$

$$\bar{\vec{S}}_i = \frac{\displaystyle\sum_{j=\frac{N(i-1)}{Q}+1}^{\frac{Ni}{Q}} \vec{S}_j}{N/Q} \qquad i = 1, \ldots, Q \tag{5.4}$$

Concatenation of these vectors is then performed to generate a single vector of dimension QP (where P is again the number of attributes in an observation vector), and it is this vector that forms the basis for estimating the pdf:

$$\vec{V}_{\bar{S}} = \{\bar{s}_{11}, \ldots, \bar{s}_{1P}, \ldots, \bar{s}_{QP}\}^T \tag{5.5}$$

The dimension of the resulting vector used for pdf estimation, QP, is independent of the number of mapping frames or states, M. Note that correlations between the beginning of a segment and the end of a segment will be captured. As Q approaches M, the two approaches become nearly identical, depending on the nature of the mapping function. Therefore, with certain types of mapping functions, the SSM approach can be seen to be a special case of sub–segmenting and averaging the attributes, specifically for the case where $Q = M$.

Normalizing for duration by averaging a sequence of acoustic vectors into a fixed number of parts has also been employed in work by Leung, by Meng, and by Chigier *et al.* [7, 43, 50] who all used a value of $Q = 3$ to achieve a manageable dimensionality. Implicit in this type of normalization is the assumption that the signal can be considered constant over the averaged interval, and that the averaging operation is removing or reducing the impact of a zero–mean additive noise over the averaged interval. Since the normalization is motivated by implementation considerations rather than by theory, it is realistic to view the normalization as a sacrifice of information in order to deal with the durational variabilities of phonetic segments. Therefore, an important consideration is that the loss of information be minimized, so that the piecewise–constant intervals accurately represent the signal prior to sub–segmentation.

5.2 Effects of Error Sub–Segmentation

A distinguishing characteristic of the approach presented in this thesis is that each segment is first compared to a synthetic segment to form an error sequence. Therefore, the assumption is that the error signal, not the observation sequence, is piecewise–constant. This is because the averaging operation is applied to the error, after all the data in the acoustic sequence has been utilized. If the dynamics of the acoustic attributes over the duration of a segment can be modelled accurately, the assumption that the error is piecewise–constant will be less of a simplification than assuming the original signal is piecewise–constant. Thus, less accuracy and information should be lost by sub–segmenting and averaging at this later stage in the processing. An example of the effects of this processing on the observation sequence and the error sequence is shown in Figure 5-1. The following section provides a quantitative analysis and comparison of the effects of sub–segmenting each of these two sequences.

5.2.1 Distortion Analysis

When a signal is sub–segmented into Q pieces, and each piece is averaged, the result can be considered to be a new signal comprised of Q piece–wise constant intervals. One means of measuring the loss of information which occurs due to sub–segmenting and averaging is to examine the energy that remains when the difference is taken between the original signal and its piece–wise constant representation. An experiment was performed which measures this residual energy for each of three sequences and their piecewise–constant approximations, using the same weighting of the components of the observation vector that were used for the distortion calculations described earlier (the σ_i^2's). The acoustic attributes were the first 15 MFCC's (including C_0).

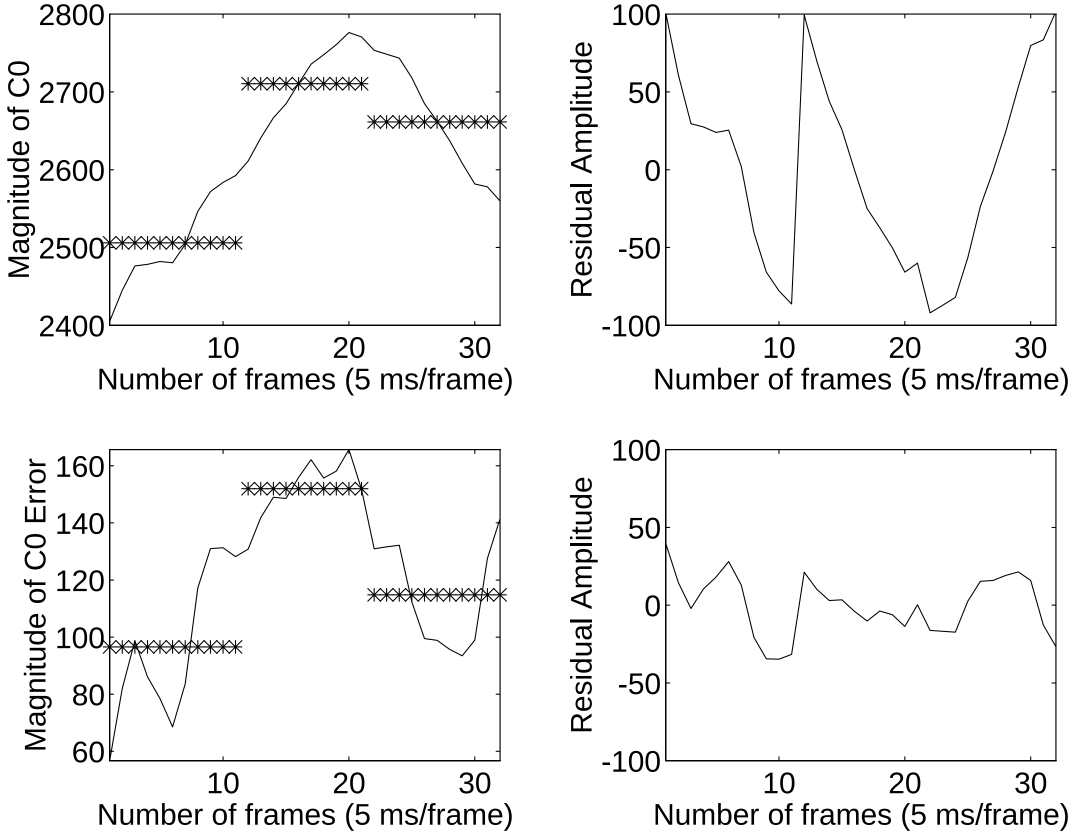

Figure 5-1: Residual energy of the observation and error due to sub–segmentation. The residual energy is computed from the [ɔʸ] token and the synthetic segment shown in Figure 4.15. The top left graph shows the original C_0 sequence (solid) and its sub–segmented approximation (*) using $Q = 3$. The top right is the residual computed from the difference of the two sequences on the left. The bottom left graph shows the error in C_0 between the [ɔʸ] token and its synthetic segment (solid), along with its sub–segmented approximation (*). The bottom right is the residual generated by subtracting the two sequences on the bottom left. Note that the residual generated from the error sequence has significantly less energy than the residual computed using the original value of C_0.

The three sequences are:

1. The original observation sequence of MFCC's.
2. The error sequence formed by taking the difference between the nominal context–independent (CI) synthetic segments (described above) and the MFCC's.
3. The error sequence formed by taking the difference between a set of *context–dependent* (CD) synthetic segments (described in Chapter 7) and the MFCC's.

The expectation is that, if the tracks are capturing the significant dynamic features of the phonetic units they model, the error sequences will involve less approximation, and will have less residual energy. Hence, the residual energy present in the original sequence of acoustic attributes should be reduced in the context–independent and context–dependent cases. These reductions should be greatest for phonetic units with the most abrupt dynamic changes in the acoustic realization, and less for those phones for which the acoustic attributes are relatively constant. For segments with fewer frames than the value of Q, there will be no residual energy.

The concept of gaining accuracy by using an error signal generated from comparing measured data to a model or template, instead of the original measured data, has been successful in other instances. An example of this is the use of linear predictive coding techniques to generate an error signal for the purpose of pitch detection in voiced speech [66]. In this instance, the error signal generated by computing the difference between the original speech waveform and an all–pole model is more robust since the formant frequency information has been removed. Another example in which the error signal provides a better approximation occurs in the theory of differential quantization [66]. In this case the idea is to minimize the loss of information due to quantization by encoding the difference between a signal and it's predicted value instead of the original signal itself.

The residual energy measurements verify that reductions are achieved when using an error sequence instead of the original MFCC's. Using different values of Q, the mean, weighted, residual energy *per utterance* was measured over the *MIT Test* data

Q – # of piecewise constant intervals	1	2	3	4	5	10
Residual Energy Case I: Original MFCC's	87.2	55.7	39.2	29.6	23.5	9.9
Residual Energy Case II: CI Track Errors	79.5	50.6	36.0	27.5	22.0	9.5
Pct. Reduction from Case I	8.8	9.2	8.2	7.1	6.4	4.0
Residual Energy Case III: CD Track Errors	72.7	47.6	34.3	26.3	21.1	9.2
Pct. Reduction from Case I	16.6	14.5	12.5	11.1	10.2	7.1

Table 5.1: Mean utterance **residual energy** due to sub–segmentation of a signal into Q piecewise–constant intervals. Previous approaches performed segmentation based on the original acoustic attribute sequence of a phonetic segment (Case I). The residual energy between the original sequence and the approximation is diminished by first forming an error sequence using track generated synthetic segments (Cases II and III). Increasing the value of Q decreases the effect of the approximation, resulting in a convergence of the three methods. Although large values of Q result in a more accurate representation, they also impose a high dimensionality on the probability density function.

set with tracks created using the Traj2 algorithm and the *MIT Training* data set. The results are summarized in Table 5.1. Although the table shows the benefits of increasing Q, it must be kept in mind that incrementing Q by one adds P (in this case 15) dimensions to the Gaussian probability density function, making a robust estimate of the covariance parameters more difficult.

The mean reduction per utterance obscures the fact that there is a large degree of variability in the reductions achieved for the different phones. The average reductions are also lowered by the initial and final silences. Table 5.2 shows the phones which had the largest reductions in residual energy when sub–segmenting the error sequence produced when using the context–dependent tracks (Case III) instead of the original acoustic sequence (Case I). The reductions achieved using the context–independent tracks are also shown. All of the values in the table were calculated for the case where $Q = 3$.

Table 5.2 shows large reductions for several closures. This is due to the sharp spectral changes which occur at the end of a closure when the burst occurs for the following stop and also at the start when the closure occurs (see Figure 4-4). In the TIMIT corpus, the closure boundaries appear to have a slight overlap with the

Phone	Percent Reduction in Residual Energy	
	Context–Independent	Context–Dependent
[p□]	17.1	21.9
[b□]	17.8	21.5
[k□]	18.2	20.3
[ɾ̃]	18.3	19.7
[ɔʸ]	10.8	19.0
[ɑ]	10.1	18.9
[š]	15.6	18.6
[e]	10.4	18.3
[ɑʸ]	11.5	18.1
[ɝ]	10.0	18.1

Table 5.2: The ten phones which achieved the largest reduction in residual energy when sub–segmenting error sequences generated by the context–independent and context–dependent tracks instead of the observation sequences. The values are for the case where $Q = 3$.

neighboring segments. Another group which shows large distortion reductions are the vowels, particularly those with an attached semi–vowel. The tracks are consistently accounting for some of the spectral motion in the realization of these phones, particularly in the context–dependent case where the reductions are almost 100% greater than in the context–independent case.

Naturally, the phones with a larger degree of dynamic activity will be better represented if a larger number of sub–segments are used. However, for phones which have a relatively flat spectrum over time, the use of a large Q requires the estimation of additional parameters which will be largely redundant. Ideally, the number of sub–segments would depend on the identity of the phone. However, this creates a complication when performing classification and recognition experiments since different phones would have different dimensional pdf's, making it very difficult to directly compare the likelihood scores. For this reason, a single value of Q is selected for all the phones in each set of experiments. The value will be chosen based on the performance achieved on the development set for each task. The value will represent a compromise between capturing additional information in the phones which exhibit a large degree of dynamics and estimating potentially unnecessary parameters in the phones which exhibit minimal spectral variation over time.

5.2.2 Correlation Analysis

Dividing the error sequence for each training token into Q pieces and averaging is identical to using Equations 5.3 and 5.4, except the error sequence E is processed instead of the original segment, S. The resulting vectors, $V_{\bar{E}}$:

$$\vec{V}_{\bar{E}} = \{\bar{e}_{11}, \ldots, \bar{e}_{1P}, \ldots, \bar{e}_{QP}\}^T \tag{5.6}$$

are computed for each training token and then used to estimate a jointly–Gaussian probability density function for each phonetic model. Due to the averaging operation, the error distribution will *not* be zero–mean in practice, although the mean of the error should be very small compared to the standard deviation. The covariance matrices can be analyzed to determine which, if any, correlations are being captured. An interesting way to examine the correlations is to normalize the covariance matrices to produce a matrix of correlation coefficients. This is done by dividing the ij^{th} entry of the matrix by the product of the i^{th} and j^{th} standard deviations. The resulting correlations will range from -1 to 1, where the extremes imply complete linear correlation, and a value of zero means the i^{th} and j^{th} variables are independent.

Figure 5-2 shows the resulting matrix for the phone [e], using a value of $Q = 10$. The absolute value of each element has been taken so that the degree of correlation is displayed, with dark area indicating a high degree of correlation and white areas indicating statistical independence (the diagonal terms will be black since every variable is completely correlated with itself). Each 10x10 sub–block represents the correlations between the two relevant MFCC's, with their correlation at the same instant of time (the Q intervals running down the sub–block diagonal), and the temporal correlations between the two attributes on the off–diagonal. A rich correlation structure is clearly evident and is strongest in a large block extending from C_1 to C_9. Temporal and spatial correlations are also clear between several sets of adjacent

MFCC's. However, a value of $Q = 10$ is too large to be of use in an actual implementation, since the parameters are difficult to estimate robustly, and because of the impact of the computational burden during phonetic recognition. Figure 5-3 shows the matrix for a value of $Q = 4$. Although the correlation structure is more coarsely represented, the majority of the correlation information is retained. Hence, reduced values of Q which produce a covariance matrix of more practical dimension appear to retain useful temporal correlation information.

It is interesting to note that we can eliminate all of the temporal correlations and just leave the spatial correlations. These represent the inter–attribute correlations at each of the Q time intervals (in this case, first quarter, second quarter, etc.). Therefore, although the explicit temporal correlations have been eliminated, there is inherently some temporal information present. For example, the variances at the beginning and end of the segment (first and fourth quarters) will be larger than the variances in the middle of the segment, primarily due to co–articulatory effects. The resulting image of the matrix is shown in Figure 5-4.

Finally, the spatial, or inter–attribute, correlations can be ignored while retaining the temporal correlations. The resulting matrix of correlation coefficients is shown in Figure 5-5. A baseline classification performance experiment will later be conducted to assess the relative impact of these alternative correlation representations. It is also interesting to examine the differences between phonetic units, in terms of which correlations are most important. The matrix of correlation coefficients for [s] and for [f] are shown in Figures 5-6 and 5-7 with $Q = 3$. For [s], the correlation structure is richest in a sub–block bounded by C_3 and C_6, while [f] has a structure with very sparse temporal correlations. The faint strips along the super and sub–diagonals reveal some relevant spatial correlations. Relative to the other phones, [f] has much less inter–attribute and temporal correlation.

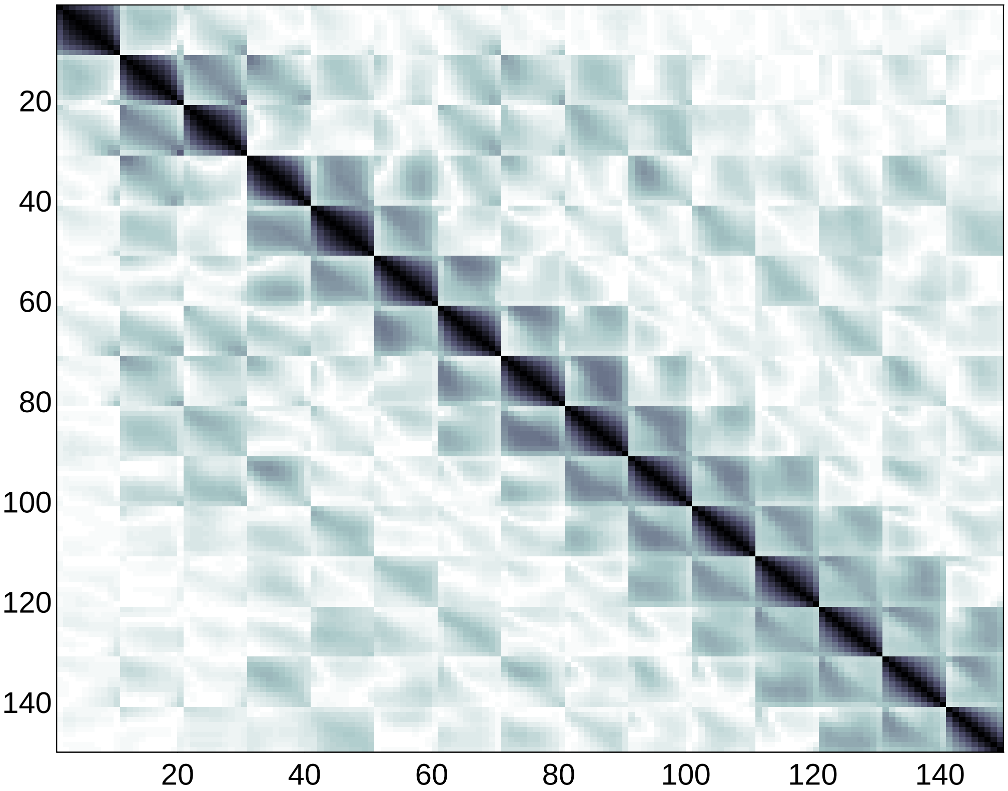

Figure 5-2: Matrix of error correlation coefficients for the phone [e]. The matrix was constructed using 15 Mel-frequency cepstral coefficients (C_0–C_{14}) and $Q = 10$ (10x15 = 150 dimensions). The figure is arranged such that the coefficients of C_0 over time are in the upper left, and the C_{14} coefficients are in the lower right. The absolute value of each element was taken so that large correlations show up dark and areas of little or no correlation show up light. If the errors were independent, the diagonal would have been black and all other elements would be white.

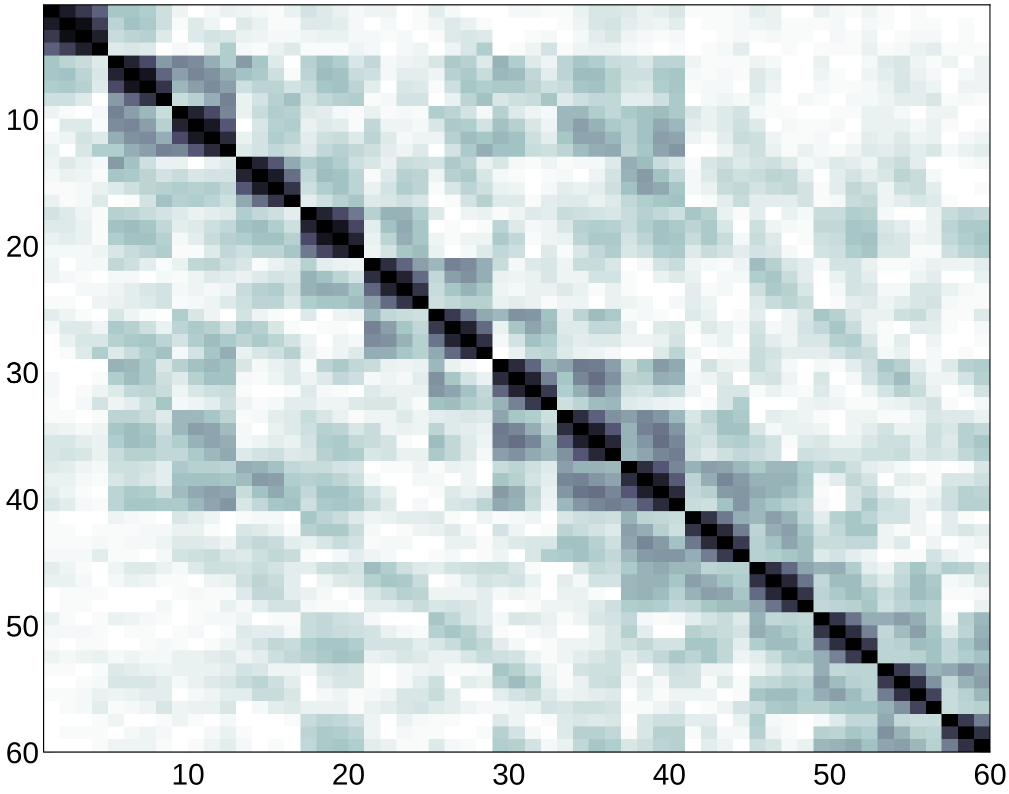

Figure 5-3: Reduced matrix of error correlation coefficients for the phone [e]. The matrix was constructed using 15 Mel-frequency cepstral coefficients (C_0–C_{14}) and $Q = 4$ (4x15 = 60 dimensions). The picture provides a coarser version of Figure 5-2, but the correlation structure is largely maintained.

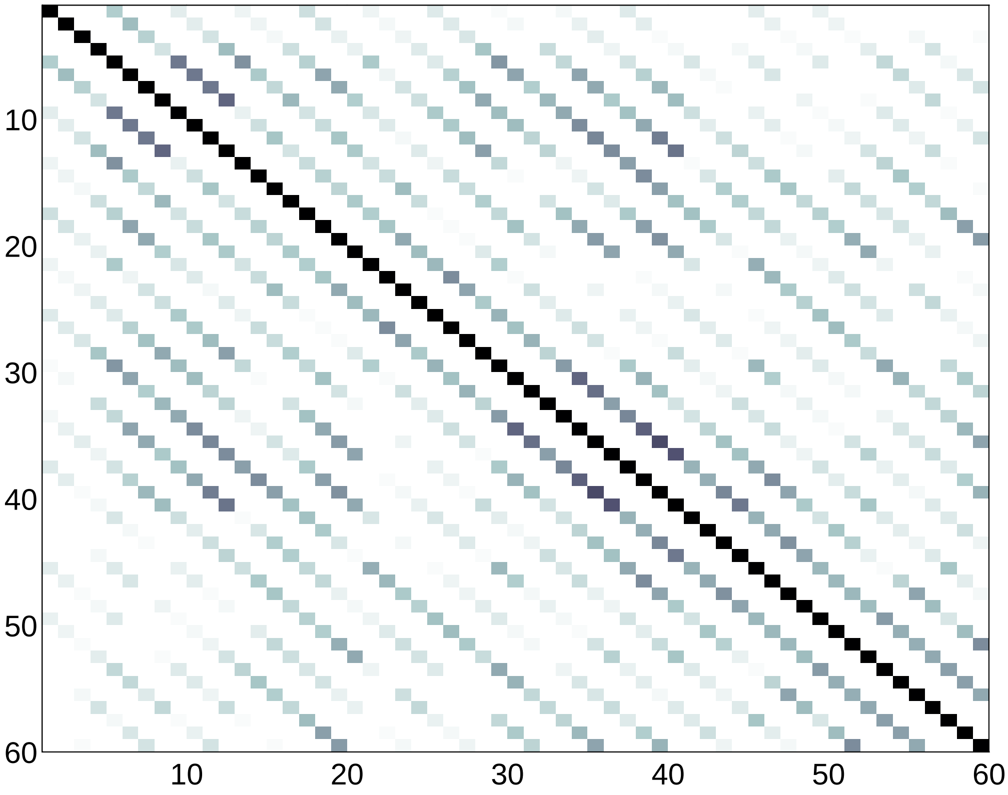

Figure 5-4: Matrix of error spatial correlation coefficients for the phone [e]. The temporal correlations have all been removed, leaving only the inter–attribute correlations at each of the 4 quarters. The figure is otherwise identical to Figure 5.2.

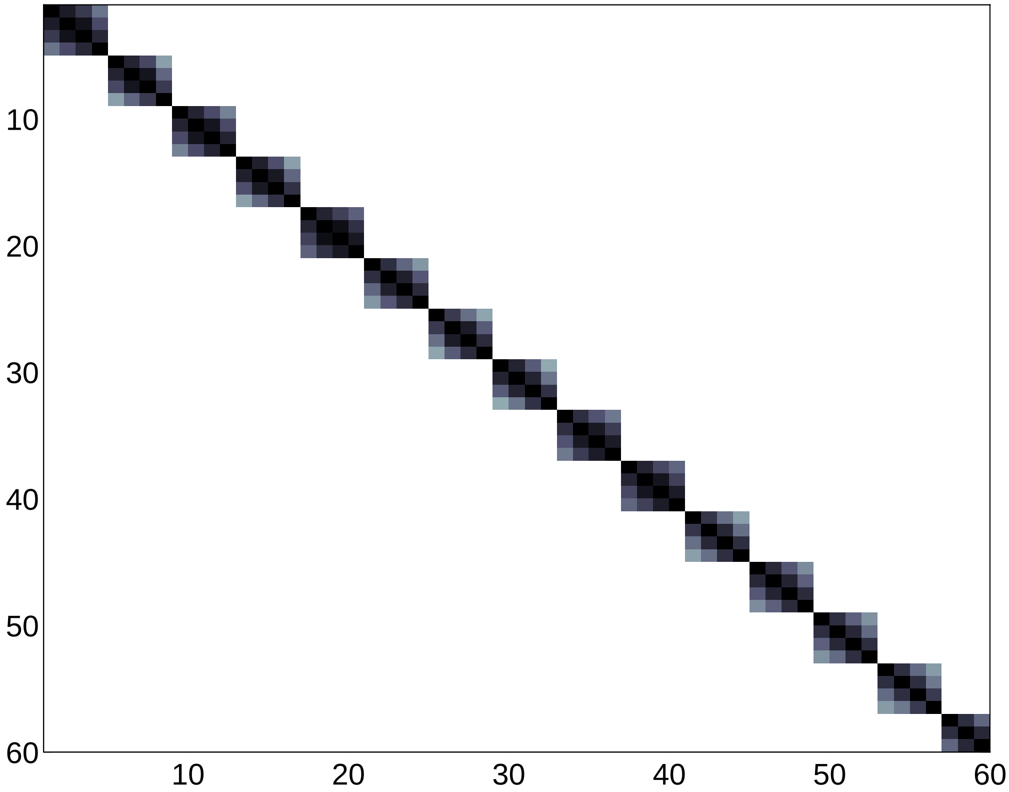

Figure 5-5: Matrix of error temporal correlation coefficients for the phone [e]. The spatial (inter–attribute) correlations have all been removed, leaving only the temporal correlations of each of the MFCC's with itself. The figure is otherwise identical to Figure 5.2.

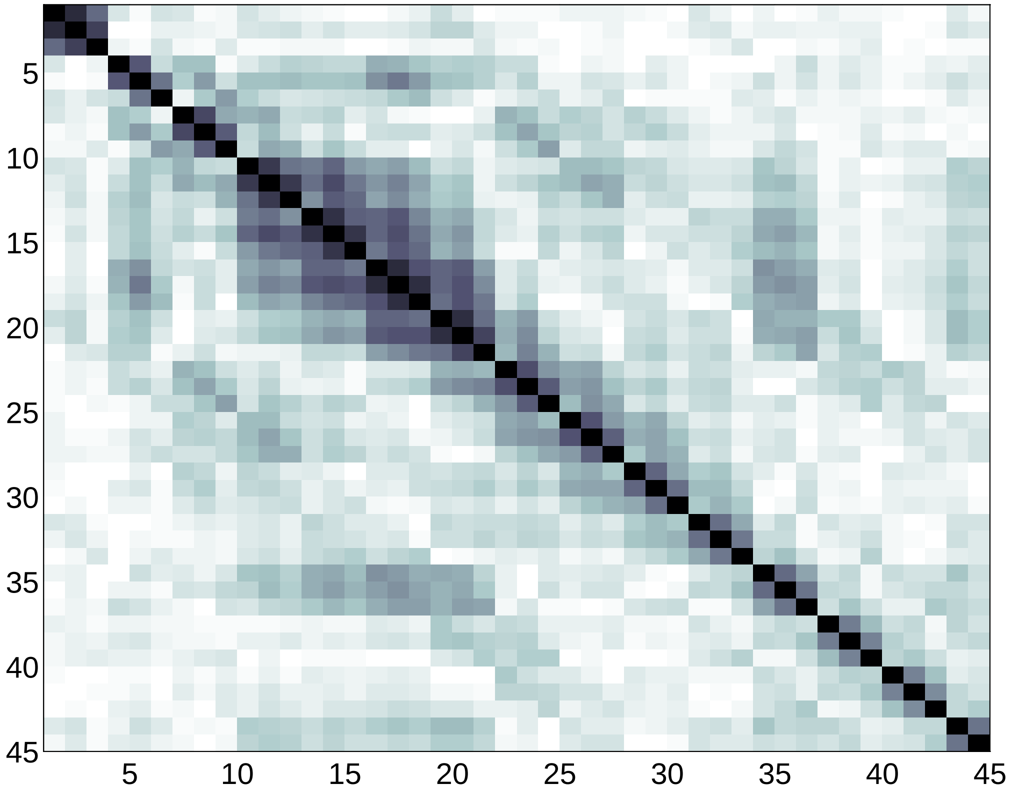

Figure 5-6: Matrix of error correlation coefficients for the phone [s]. The matrix was constructed using 15 MFCC's and a value of $Q = 3$.

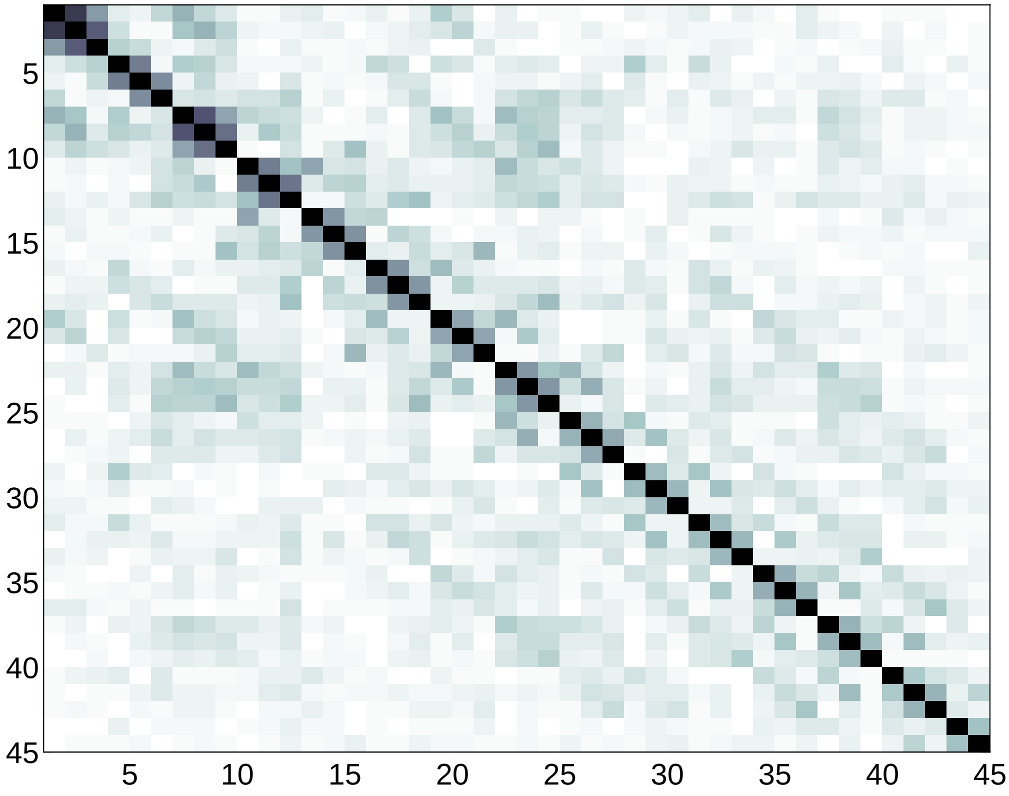

Figure 5-7: Matrix of error correlation coefficients for the phone [f]. The matrix was constructed using 15 MFCC's and a value of $Q = 3$. The matrix reveals that in comparison to other phones, the MFCC's in [f] are relatively independent of each other and have little temporal correlation between attributes.

5.3 Chapter Summary

In this work, the statistical model is a joint-Gaussian probability density function based on the error signal. A method which allows important temporal correlations to be captured while maintaining a dimensionality small enough to robustly estimate the covariance parameters was presented. This method involves creating an error sequence of variable duration with the tracks, and then normalizing this duration by averaging the vectors over each of Q sub–segments. This technique has the advantages of utilizing all of the available data. Thus, the model of the error used in this work is the maximum likelihood estimate of the mean, which is *not zero*, due to the averaging into Q pieces, and full covariance matrix for each phone. For P acoustic attributes, the result is a joint-Gaussian density of dimension PQ. It is important to note that the dimension of the model is independent of the number of states M used to characterize the track. The next chapter will address the issue of the best choice of covariance matrix dimension as affected by the choice of Q.

Chapter 6

Context–Independent Phonetic Classification

This chapter is concerned with developing and tuning the statistical trajectory models and with preliminary evaluation of their performance. To meet this end, baseline phonetic classification experiments will be used to evaluate the impact of different aspects of the system. Following a description of the protocol and framework for conducting classification experiments, the chapter will presents an evaluation of the effects of varying the parameter Q. The value of this parameter involves a trade–off between the ability to capture additional correlation information versus the penalty of increasing the dimension of the covariance matrix used in the pdf. The algorithms Traj2 and Traj5 are then evaluated to determine which generation function is to be used throughout the remainder of the thesis.

The chapter next presents a series of context–independent (CI) phonetic classification experiments. In CI experiments, the phonetic context is not known during either training or testing. The first set of experiments in section 6.3 was conducted on the set of 16 unreduced vowels in American English for preliminary comparison with other results in the literature. The vowels were chosen for this first set of experiments since they exhibit a high degree of dynamic behavior, which is particularly relevant for the trajectory modelling. Additionally, another set of experiments was undertaken to examine the impact of the different correlations in the pdf's on vowel classification performance. These results show the importance of the temporal correlations to classification performance. Classification experiments are also performed over the full set of phones in the TIMIT corpus. These experiments are described in Section 6.4.

6.1 Classification Framework

The classification experiments are conducted by extracting all the segments from each of the utterances in the data set being evaluated and scoring them independently with each of the phonetic units. The segment boundaries are provided with the TIMIT transcriptions. The approach is to first process each segment with the tracks for each phonetic unit to create an error sequence which is then sub–segmented. A maximum likelihood computation is then performed based only on the sub–segmented error, $V_{\bar{E}}$ (defined in Equation 5.6).

Specifically, the probability of a phonetic unit, a_i given the sub–segmented error is:

$$p(a_i|V_{\bar{E}}) = \frac{p(V_{\bar{E}}|a_i)p(a_i)}{p(V_{\bar{E}})} \tag{6.1}$$

The prior probabilities for context–independent experiments are the unigram priors (i.e. the estimate is based on number of times each phone occurs in the training set). The denominator is a normalization constant which does not depend on the phonetic unit and can be ignored. The procedure for selecting the identity of each segment is therefore:

$$a^\star = \underset{i}{\text{argmax}} \;\; p(V_{\bar{E}}|a_i)p(a_i) \tag{6.2}$$

The specific protocol for conducting classification and recognition experiments involves dividing the data into three pieces called the training, development, and test sets. The training set is used to train all of the phonetic models, and also to collect relevant statistics, such as prior probabilities. Therefore, it is important that the data in the training set be representative of the data that the system will be evaluated on. The test set is the data set that determines the system performance. For experiments on systems which are meant to be speaker–independent, it is a generally accepted practice that the speakers in the training and test sets do not overlap. Furthermore, when a system is being tuned and design decisions are being made, it is also standard practice to evaluate the system on another independent set of data known as the development set. The development set serves as a type of "practice" test set. During system tuning, there exists a natural tendency to keep what increases performance and disregard what doesn't. The danger is that a type of hill climbing takes place, wherein it is possible to start "learning" the test set. Decisions may be made based on performance improvements which, rather than actually reflecting a superior implementation, in fact reflect nuances of the speakers in the evaluation set. Using a development set protects the actual test results from being artificially inflated due to this effect.

6.2 Preliminary System Tuning

There are two performance issues to be examined in this section. The first issue examines the impact of different values of Q, and the second issue is the choice of a generation function between the two most promising candidates, Traj2 and Traj5. The early work in this thesis involved experiments focused on the 16 unreduced vowels used in American English. The *HM Train* vowel corpus was split into two pieces for training, *HM Sub–Train* and test *HM Development*. The speakers in the *MIT Test* corpus were set aside for eventual classification experiments. The training set was then used to create tracks and models of their associated error statistics for each of the 16 vowels.

Q – # of piecewise constant intervals	1	2	3	4	5
Dimension of distribution	15	30	45	60	75
Test % Vowels Correct	46.7	57.8	61.0	62.1	57.5
Train % Vowels Correct	46.7	58.0	63.2	64.8	64.5

Table 6.1: Vowel classification on test and training sets for different values of the parameter Q using the Traj2 algorithm. Q determines the number of regions within which the error sequence is averaged. The resulting probability density function is of dimension PQ.

6.2.1 Effects of Varying Q

To examine the effect of varying Q, statistical error models were trained for each of the 16 models for values of Q from one to five. Since there were 15 MFCC's used as acoustic attributes, the dimensions of the resulting pdf's varied from 15 to 75. The results of a series of classification experiments are shown in Table 6.1. Classification performance is shown for the test set and also for the training set used to create the models.

For small values of Q, the performance on the training and test sets is close because the lower dimension of the covariance matrix allows for robust parameter estimation, and there was sufficient training data. However, for the experiment with $Q = 5$, the test performance, which had been rising as Q was increased, suddenly drops almost 5%. This suggests that there was not enough training data to adequately estimate the large number of parameters in the resulting 75 dimension covariance matrix. This hypothesis is supported by the discrepancy between training and test set performance for $Q = 5$.

Based on the data in Table 6.1, a value of $Q = 4$ was used for all of the vowel classification experiments involving the Traj2 algorithm.

6.2.2 Selection of a Generation Function

The Traj5 algorithm showed the least distortion of any of the generation functions explored in Chapter 4 (see Table 4.10). However, due to the unique nature of the Traj5 algorithm its distortion values are not directly comparable to those of the other algorithms, all of which compute distortion using exactly the same MFCC data. Therefore, the best means of comparing the Traj2 and Traj5 algorithms is on an actual classification task.

The Traj5 algorithm was the only algorithm which involved sub–sampling of long tokens and interpolation of short tokens during training and test. To create a track of ten states, this means that all tokens over 50 ms. in duration (ten frames) will be sub–sampled. In the case of the vowels, nearly every token will be greater than 50 ms. in duration. Therefore, to fully understand the effects of the Traj5 algorithm, a vowel classification experiment is insufficient. For this reason, the Traj2 and Traj5 generation functions were compared on a classification task involving all the phones. The data sets used were *MIT Training* and *MIT Augment–Test*. At this stage it was deemed acceptable to use data from the *MIT Test* set. No actual parameter tuning takes place as a result of these experiments, and the trends discovered will be of a very general nature. Additionally, the *MIT Test* set is not the data set which will be used for the phonetic recognition experiments, when the complete algorithm is evaluated.

The Traj2 and Traj5 algorithms were compared directly using an error model built with a value of $Q = 4$. In addition, a series of experiments was conducted for Traj5 in which the error model was constructed without using sub–segment averages of the error sequence. Instead, the full error sequence computed using the Traj5 generation function was used to estimate a full covariance Gaussian distribution. These experiments were conducted for cases where the Traj5 algorithm was set to four, five, and ten states. This means that much data was thrown out due to sub–sampling. For example, in the four state case, a vowel of duration 40 frames (200 ms.) would be sub–sampled every ten frames. The results of these experiments are

Case	Algorithm	Track States	Error Model	Covariance Dim.	% Correct
I.	Traj2	10	Nominal	60	67.29
II.	Traj5	10	Nominal	60	66.61
III.	Traj5	4	Full Seq.	60	66.25
IV.	Traj5	5	Full Seq.	75	64.50
V.	Traj5	10	Full Seq.	150	51.45

Table 6.2: Classification performance comparison between the Traj2 and Traj5 generation functions. The nominal error model refers to the sub–segment and average processing of the error sequence described in the last chapter, using a value of $Q = 4$. The full sequence case means that the entire sequence was used as the basis for creating a high dimensional covariance matrix.

shown in Table 6.2.

Several trends of interest are evident from the table. First, it is once again apparent that there is not enough training data to estimate the high dimensional probability density functions. The second interesting result is the comparison between the Traj5 algorithm using a ten state track and a nominal (sub–segment and average) error model versus a four state track and an error model based on the full error sequence. Although the performance was better for the nominal error model, the performance was close. This means that there is not much difference between averaging the error computed in each quarter of the error sequence and using a single data point from each quarter of the original observations to produce a short error sequence. This means that sub–sampling long tokens can be done without losing important correlation information. However, in no instance was the performance of the Traj5 generation function superior to the performance of the Traj2 generation function. A closer examination of the data reveals the reason.

All of the Traj5 experiments showed a great deal of difficulty with short tokens, in particular, with tokens whose duration is less than the number of states. When a token has fewer frames than the number of states, interpolation is used to fill in the extra parts of the trajectory. An unfortunate side effect of this interpolation is that it artificially creates temporal correlations of the acoustic attributes in the error

Phone	Class	Mean Duration (in Frames)	Case I	Case II	Case III	Case IV	Case V
[b]	stop	3.5	212	539	523	754	1070
[ɾ]	flap	5.7	80	84	96	100	477
[n]	nasal	10.9	377	423	418	476	611
[ə]	vowel	9.9	169	259	266	313	466

Table 6.3: Misclassifications for different phones as a function of the algorithm used during training. The case numbers refer to the experimental conditions described in the previous table. As the number of states becomes greater than the the duration of a significant number of training tokens, the misclassifications caused by the shortest phones within a class rises dramatically. This is due to artificially created correlation information caused by linear interpolation.

model. During test, any short token which is interpolated will also have this artificial correlation structure, which leads to misclassification. This phenomenon is revealed in Table 6.3, which shows the shorter duration phones from each of several acoustic classes. For each phone, the percent correct is shown along with the number of misclassifications caused by this phone, i.e. tokens which were actually other phones but which were classified as the indicated phone. As soon as a large number of training tokens need to be interpolated (mean number of frames near the number of states), the number of misclassifications caused by the shortest phone within a class rises dramatically.

While the artificial correlations caused by linear interpolation to “create” extra data are a problem during classification, their effect on recognition results could cause severe performance degradation. This is because the duration of individual tokens is unknown during recognition and must be hypothesized. Therefore, all the actual segments will at some time have a portion of their data hypothesized as a segment requiring significant interpolation. This will exacerbate the problem of misclassification by the shorter duration models.

Based on the reasons cited above and superior performance in this experiment, the Traj2 algorithm will be used throughout the remainder of the thesis.

IPA							
ɑ	æ	ʌ	ɔ	ɑ^{w}	ɑ^{y}	ɛ	ɝ
e	ɪ	i	o	ɔ^{y}	ʊ	u	ü
TIMIT							
aa	ae	ah	ao	aw	ay	eh	er
ey	ih	iy	ow	oy	uh	uw	ux

Table 6.4: Vowels used for the classification experiments. The vowel symbols are shown using both IPA and TIMIT formats.

6.3 Vowel Classification Experiments

At this point, most of the design parameters have been decided upon, and it is possible to begin to evaluate the statistical trajectory algorithm. The experiments reported in this section pertain to the thirteen monothong and three diphthong vowels of American English, as shown in Table 6.4. The vowels were chosen for the initial set of experiments because they tend to exhibit a large degree of spectral motion, and therefore, provided a good means of exploring the statistical trajectory model concept. There are also several other published vowel classification studies which can be used as a basis of comparison.

The experiments were based on the training sets *HM Train* and *HM Augment–Train*, and the test set *MIT Test–V* described in Table 2.2. The data sets are also shown in Table 6.5 for ease of reference. The training set containing only the (*sx*) utterances is identical to that used in several previous studies [6, 25, 43, 50]. The effects of adding the (*si*) utterances to the training data were also investigated. Some of the experiments also explored the use of ΔMFCC's which were computed at the beginning and end of the vowel segment [7]. The ΔMFCC's used a window extending seven frames (35 ms) in each direction from the boundary. This value will be used in all the experiments in this thesis.

The vowel classification results reported here were all based on a common track configuration of ten states. This choice was determined by the results of the distortion studies in Chapter 4, which showed that the reduction in mean distortion began to

Type	# Speakers	# Sentences	# Vowel Tokens
HM Train (*sx*)	499	2,495	20,528
HM Augment–Train (*sx,si*)	499	3,992	34,576
MIT Test–V (*sx*)	50	250	1,879

Table 6.5: Training and test sets for the vowel classification experiments.

asymptote at larger values. In all cases the resulting error was sub–segmented using a value of four for Q. The choice of these parameters resulted in a 60 dimensional distribution for experiments with the 15 MFCC's and 90 dimensions for experiments that included the two cepstral differences (at the start and end of the segment).

The first set of experiments investigated the relative importance of the temporal and spatial correlations in the error for classification. As shown in Table 6.6 a set of four experiments was performed using a single track and full covariance Gaussian error model for each phone. The *diagonal* condition assumed total independence of all dimensions in the error. The *time* condition retained the temporal correlations of each MFCC across the four sub-segments and assumed independence between MFCC's. The *space* condition assumed independence between each of the four sub-segments. Thus, it modelled the MFCC correlations within each sub-segment and also captured some temporal information as well, since a separate block was trained for each sub-segment. It is important to note that although the *space* condition utilized nearly four times the number of parameters as the *time* condition, the performance was very similar. This result highlights the importance of the temporal correlations. Finally, the *full* condition modelled all correlations and produced the highest accuracy of 62.5%. This last condition was used for all subsequent experiments.

Additional experiments consisted of adding information to enhance classification performance. This information included segment duration, ΔMFCC's, and separate, gender specific models for each vowel. The *(si)* training data was also added to the training set. A vowel classification result of 68.9% was achieved. A summary of the experimental results for the vowels is shown in Table 6.7.

In the table, the baseline configuration refers to the use of a single track for

Condition	**Covariance Parameters**	**Correct (%)**
Diagonal	15 x 4 = 60	51.0
Time	15 x (4x4) = 240	55.0
Space	4 x (15x15) = 900	55.8
Full	(4x4) x (15x15) = 3600	62.5

Table 6.6: Vowel classification performance as a function of information retained in the covariance matrices.

Description	**Baseline (%)**	**Gender (%)**
MFCC's	62.5	62.5
+ Duration	63.8	65.1
+ ΔMFCC's	66.0	67.5
+ *si* Training Data	66.6	**68.9**

Table 6.7: Results for the vowel classification experiments.

each of the 16 vowels and a single Gaussian distribution for each error model. The gender configuration augmented the baseline with tracks and error models which were trained separately on male and female speakers. Hence, three models were produced for each vowel. During testing the gender was unknown, and the top-scoring model determined the classification result.

Duration statistics are collected over the training set for each phone. The best representation of duration was found to be the log duration, which was modelled as jointly–Gaussian with the acoustic parameters. Relevant correlations were discovered between durational errors and errors in the MFCC's. Hence incorporating the durational information into the covariance matrix, rather than treating it as an independent source of error, resulted in better performance results. The choice of a log Gaussian pdf and a more complete study of duration is presented in Appendix A.

The ΔMFCC's help incorporate contextual information, since they are computed using data beyond the segment boundaries. By using only the 2 P dimensional vectors computed at the segment boundaries, the dimension of the covariance matrix is increased by $2P$. Had the ΔMFCC's been processed like the static MFCC's, then the dimension of the covariance matrix would have increased by QP, and the important contextual information obtained at the boundary would have been diluted by the cepstral slope on the interior of the segment. However, this is information already inherent in the track. Additionally, the spectral motion at the boundary is generally greater than the motion within a segment, since the articulators are moving from one phone configuration to another.

Making specific gender models also boosted performance considerably. This was particularly true in distinguishing vowels such as [æ] and [ɛ]. The reason can be seen in Figure 6-1, which shows tracks created for the vowels [æ] and [ɛ] using tokens from both genders along with gender specific tracks. When the data is combined, male [æ] tokens will generally be closer to the [ɛ] track leading to misclassifications. This type of difference in the spectral outputs can be attributed to differences in the vocal tract physiology of the two genders. In particular, males tend to have longer vocal tracts resulting in resonances at lower frequencies.

6.3.1 Vowel Results Discussion

The vowel experiments clearly indicate the advantages of the durations, the ΔMFCC's, and the use of gender specific models. The results obtained on the vowels compare favorably to other results currently in the literature. Meng reports 59.6% on the same task when using 15 MFCC's with an MLP classifier [51]. Her representation is very similar to the static MFCC experiment which achieved 62.5%, although she did not use C_0. Her best result was 65.6% using two auditory model outputs.

Carlson and Glass also reported results on this vowel classification task using an MLP classifier [6]. Their most similar experiment used three average Bark spectral vectors, obtaining 62.5% accuracy. When they included gender information, they obtained 65.8% with a formant–based representation. They found that duration information improved classification performance by around 1.3%, which agrees with the results in this work.

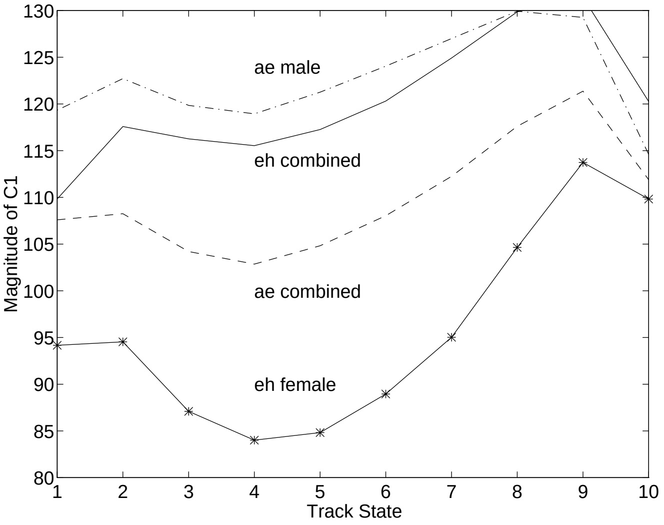

Figure 6-1: Example of track variability due to gender.

C_1 for tracks created using all the [æ] (ae) data and all the [ɛ] (eh) data from both genders (called "combined" in the figure), and also from tracks created using just the male tokens for [æ] and the female tokens for [ɛ]. For clarity, the plots of the [æ] female and [ɛ] male tracks, which would have appeared below and above those shown, have been omitted. The plot reveals why gender specific models can be important. Note that the track from all of the [ɛ] tokens is very close to the track representing the [æ] male tokens. Without gender specific models, male [æ] test tokens would be easily confused with [ɛ].

Type	# Speakers	# Sentences	# Phonetic Tokens
MIT Train (*sx,si*)	567	4,536	175,095
MIT Test (*sx*)	50	250	8,751

Table 6.8: Training and test sets for the phonetic classification experiments.

6.4 Phonetic Classification Experiments

The classification experiments over the full set of phones were also based on a track of ten states. The experiments were based on the training set *MIT Train* and the test set *MIT Test* described in Table 2.2, and also summarized in Table 6.8 for ease of reference. For these experiments, the best results were found when the error was sub–segmented using a value of three for Q. Although the difference was small, some of the phones had insufficient data to support the higher dimensionality that results when Q increases. This choice resulted in a 75 dimensional distribution for experiments with the 15 MFCC's and the ΔMFCC's computed at each segment boundary. The window for the computation of the ΔMFCC's was 7 frames (35 ms) in each direction. For experiments which included log duration information, the Gaussian pdf increased to 76 dimensions.

Fifty–seven (57) phonetic models are created to represent the phones in the TIMIT corpus. Note that in some cases, multiple phones have been combined to form a single model. This was done for [h#, □], [m, m̩], and [ŋ, ŋ̩]. Very little data exists for [m̩] and [ŋ̩], and pauses were deemed similar enough to silences to combine the two. The classification results are computed by mapping the phones to a set of 39 classes defined by Lee [40] and commonly used in the literature [16, 38, 68]. Note that in accordance with Lee, glottal stops are ignored. The phone classes are shown in Table 6.9.

The results over all the phones are roughly equivalent or slightly superior to other results in the literature [7, 16, 45]. The results are summarized in Table 6.10. In the table, “Duration (independent)” means durations were included, but considered statistically independent of the MFCC's. “Duration (correlations)” means the duration

Class	Phones	Class	Phones	Class	Phones
ɑ	ɑ, ɔ	u	u, ü	v	v
æ	æ	r	r	ð	ð
ʌ	ʌ, ə, ə^h	l	l, ḷ	ɾ	ɾ
ɑ^w	ɑ^w	w	w	p	p
ɑ^y	ɑ^y	y	y	b	b
ɝ	ɝ, ɚ	m	m, ṃ	t	t
ɛ	ɛ	n	n, ṇ, ɾ̃	d	d
e	e	ŋ	ŋ, ŋ̣	k	k
ɪ	ɪ, ɨ	s	s	g	g
i	i	z	z	č	č
o	o	š	š, ž	ǰ	ǰ
ɔ^y	ɔ^y	f	f	h	h, ɦ
ʊ	ʊ	θ	θ		
SILENCE	p□, t□, k□, b□, d□, g□ , □, □, h#				

Table 6.9: Phone classes with allowable confusions (confusions scored as correct answers).

information was included in the Gaussian pdf, so as to capture correlations with the other acoustic attributes. *Single Gaussian* means one Gaussian pdf was trained for each phonetic model. *Gender Gaussians* means (as with the vowels), gender based tracks and pdf's were used in combination with a combined model for each phone, and the gender of the test speaker was unknown. *Enforced Gaussian* means gender specific models were trained, but that the gender of the test speaker was assumed to be known. Therefore, only male models were used to test male speakers. For female speakers, the female models were used in conjunction with the combined models. The use of combined models was necessitated by the fact that there are fewer female speakers in the TIMIT corpus, and for some phonetic units training data was sparse.

The assumption that the gender of the speaker is known is not a strong one. Studies by Lamel and Gauvain conducted on the TIMIT corpus showed accurate gender identification performance of better than 97% after 0.4 seconds of speech (about 4 phones) which improved to 99% after 2.0 seconds of speech [37]. When using two utterances by the same speaker, the performance was 100%. The test set consisted

Description	**Correct (%)**
Single Gaussian MFCC's and ΔMFCC's	74.2
Single Gaussian + Duration (independent)	74.4
Single Gaussian + Duration (correlations)	75.2
Gender Gaussians + Duration (correlations)	76.4
Enforced Gender Gaussians + Duration (correlations)	76.8

Table 6.10: Results for the *context independent* classification experiments.

of 168 speakers.

An experiment verifying this consistency was run using the gender specific statistical trajectory models. Each token was scored by all the models for each gender. Duration and prior information was ignored. The gender of the winning token was then compared to the gender of the speaker. Probabilities for each utterance were also obtained by combining the winning likelihood scores for each gender separately. Of the 250 utterances in the test set, all were correctly classified with respect to gender. Further, on a token–by–token basis, the winning model had the correct gender 91.2% of the time. This result includes silences, closures, and glottal stops. In the case of silences and closures, most of the information is assumed to come from the cepstral derivatives which provide some contextual information. Tokens in the silence class (including glottal stops) had the correct gender 80.1% of the time. Excluding silence class tokens, the correct gender was identified on the other tokens 92.9% of the time. This result strongly reinforces the conclusion that the gender of the speaker can be reliably identified.

Therefore, the "enforced gender" results should not suffer significant degradation if the gender must be identified by the system. It should be noted that in those cases where Lamel and Gauvain made gender errors, the results for those speakers were better when using the cross–gender phonetic models.

6.4.1 Classification Results Discussion

The context independent classification experiments show the improvement obtained by including the durational information in the covariance matrix. This is an advantage of modelling duration, or log duration, as a Gaussian variable. The Gaussian assumption makes it possible to directly augment the vector of acoustic errors with the duration error. Again, the benefits of gender specific models are evident, but the 1.6% increase achieved over the full set of phones was not as high as the 2.3% increase which was achieved on the vowels. This is because the acoustic realization of many non–vowel phones, such as those articulated at the lips, are not as dependent on the vocal track, and thus, carry less information about the speaker.

Other significant context–independent classification results currently reported in the literature use different training and test sets from those used in this work. Therefore, performance comparisons are made to illustrate what has been achieved elsewhere as opposed to making a direct comparison. The best result currently reported on a context–independent classification task is 78.0% [7, 45]. This study used the *KFL Train* and *KFL Test* data sets consisting of 610 speakers and 20 speakers respectively. This same work reported results of 77.0% using MFCC's with a multi–layer perceptron (MLP) classifier, and 75.3% was using a Gaussian pdf. These values are close to the result reported here of 76.8%. These results appear to indicate that the flexibility inherent in using a neural network to estimate the pdf provides a performance advantage over assuming a Gaussian distribution.

An additional point of comparison can be made with the Dynamic Systems Models of Digilakis. Using a test set consisting of twelve male western speakers and an acoustic representation based on 18 MFCC's and ΔMFCC's he achieved classification performance of 73.9% [16].

6.5 Chapter Summary

This chapter focused on evaluating the statistical trajectory models at an initial but significant stage in their evolution. Before the classification experiments could be conducted, preliminary experiments allowed values for Q (number of averaged sub–segments in the error model) and a generation function to be selected. The Traj2 algorithm achieved superior performance and is the algorithm which will be used in the remainder of the thesis. Future experiments will also use a value of $Q = 3$ when creating the error models.

The temporal correlations were then shown to be of great importance. Their impact on system performance was equivalent to the impact of the spatial correlations. The statistical trajectory models were shown to achieve high performance on a vowel classification task and an all–phone classification task using full covariance Gaussian distributions of the error. The importance of the ΔMFCC's, duration correlations, and gender specific models was also demonstrated.

Contextual information has not yet been accounted for when creating the tracks. Dynamics due to context are therefore "averaged out" of the track states near the boundaries. Also, many phones exhibit a relatively constant spectral trajectory across the interior of a segment. The dynamic nature of the transitions between phones has not yet been examined. Therefore, areas where the statistical trajectory model may have a significant advantage have not yet been dealt with. These areas are the subject of the next chapter.

Chapter 7

Co–Articulation Modelling

The key idea behind co–articulation modelling is to account for the variability in the acoustic realization of phones that occurs due to the phonetic context. The motion of the articulators is highly influenced by the articulatory configuration of the preceding phonetic segments, referred to as the *left context*, and the following phonetic segments, referred to as the *right context*. While predominantly due to the immediate neighbor, this contextual influence can extend across several phones. An example is shown in Figure 7-1. The spectrogram shows the effect of rounding due to the [o] extending through the preceding phones to the [s] in the word *stroll*. The [s] has a lower cutoff frequency at t = 0.3 seconds than it does at t = 0.1 seconds.

This chapter seeks to attack the problem of co–articulation from two different directions. First, the effects of co–articulation within a segment will be examined. Context–dependent models will be constructed for each of the phones, and the impact on performance will be evaluated. Second, an approach emphasizing the phonetic transitions will be investigated. During transitions, the articulators are in motion, and the acoustic attributes are highly dynamic. By finding a means of accurately modelling the transitions themselves, system performance might be greatly enhanced.

The next section discusses the key problem that makes designing context–dependent models difficult, the problem of sparse training data. The standard approach to solv-

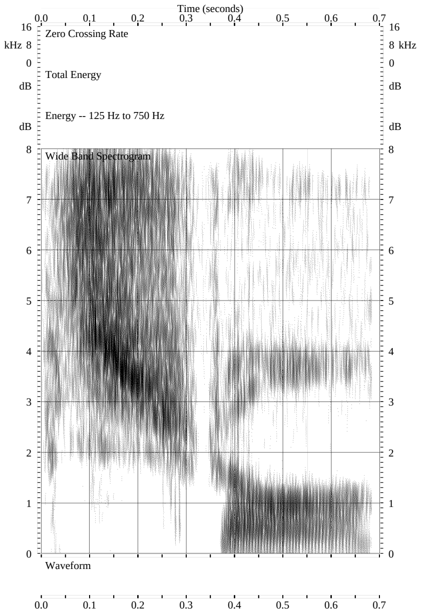

Figure 7-1: Spectrogram of the word "stroll." The cutoff frequency of the [s] falls due to the effect of anticipatory co–articulation with the [o].

ing this problem, clustering of similar contexts, is then discussed. Section 7.2 presents and evaluates the techniques involved in creating context–dependent tracks. The use of a track distortion metric for clustering tracks is presented, and also a technique for merging of tracks to account for both left and right phonetic context. Section 7.3 presents the algorithm used for creating robust tracks of the phonetic transitions. The impact of these transition tracks on performance will be presented in Chapter 8. The chapter is summarized in section 7.4.

7.1 Sparse Data and Clustering

An important consideration in designing context–dependent models is the problem of sparse training data. Ideally, complete models would be constructed for a particular phone in every possible context. If the context includes only an immediate left or right phonetic neighbor it is a *biphone model.* If both left and right contextual dependencies are included, the model is a *triphone* model. As the phonetic model becomes more specific, from a context–independent model (O(N) models), to a biphone model (O(N^2) models) or a triphone model (O(N^3) models), fewer instances of the phone in the specific environment are available for training. Since most models require the estimation of a large number of parameters, the lack of the training data becomes a severe design constraint. Therefore, it is generally only possible to create full triphone models for the most common phonetic combinations. This is also the reason why contextual effects extending beyond the immediate phonetic environment are seldom modelled.

Early work done by Schwartz *et al.* [74] and Chow *et al.* [8] interpolated triphone, biphone, and context–independent models based on their frequency of occurrence. Results from these studies showed error reductions at the word and phone level of approximately 50% compared to their context–independent implementations. An approach employed by Lee and Hon [40] was to create right context models of common biphone combinations. In their work, the context–dependent models are initialized by, and share information with, the context–independent models.

A more recent solution to the sparse data problem which has led to significant performance improvements is to pool, or share data, across contexts via a clustering mechanism. The idea is to let a single model incorporate several different contexts which have a similar effect on the phone's acoustic realization. A typical approach is to perform a clustering operation to determine which contexts should be pooled together.

One approach is to cluster contexts top–down, using linguistic knowledge, such as place–of–articulation [46]. This approach is completely supervised, that is, the clustering categories are based solely on expert knowledge of acoustic–phonetics. Top–down clustering can also be implemented in a data–driven, or partially unsupervised manner. An example is the work of Lee *et al* (generalized triphones) [41]. The method employs an expert generated list of questions about contexts which are generally linguistically motivated, and recursively selects the most appropriate question to split the phone's data. Other top–down algorithms using context decision trees have been successfully applied by Bahl *et al.* [2], and Phillips *et al.* [60]. Ostendorf *et al.* [58] used a linguistically based top–down approach to create initial clusters for triphones. For the SSM, the idea is for triphones in the same cluster to share covariance statistics. The clusters were then re–sorted using a k–means clustering algorithm and a Mahalanobis distance metric.

A second method of clustering the contexts is to use a bottom–up approach. Lee used a greedy bottom–up clustering algorithm and an entropy based distance metric to create generalized HMM triphone models [39]. The decision metric in this case provided a measure of how "close" different HMM models were to each other. The work of Hwang and Huang uses an entropy metric to merge output distributions from different HMM states to create *senones* [29, 30]. Different senonic units share Markov states and output distributions, without sharing an entire model. This can be an advantage over merging entire models, since only parts of the models may be similar. The results in [30] report a 20% improvement over the method of generalized triphones [41] on the 997 word DARPA Resource Management task [63].

Lee *et al.* [41] compared the results obtained using an agglomerative clustering algorithm to a top–down decision–tree algorithm on a word recognition task. In a vocabulary–dependent task where there was full coverage of the test set, the results were comparable. Top–down clustering performed slightly better in this case. However, 20% more contextual models were used in the top–down case because the tree–based structure was able to support more models due to superior smoothing capabilities. Next the two algorithms were compared on a vocabulary–independent task where triphone coverage of the test set was only 90%. In this case the top–down algorithm outperformed the bottom–up algorithm with word error rates being 15.0% and 15.8% respectively. The superior performance of the top–down approach was attributed to two problems with the bottom–up approach:

1. A bottom–up approach has difficulty dealing with the issue of *coverage* of the test set. Coverage has to do with the fact that not every possible phonetic

combination is always seen in the training data. Hence, only a subset of all possible combinations are "covered" during test. With a bottom–up approach, if a context is encountered in the test set that was not seen in the training set, the context–independent model is used. With a tree based top–down approach, it is possible to "back off" a single layer up the tree to find the appropriate context in the hierarchy. For example, a right context for [g] such as [g] [ʊ] may not occur in the training set, but there could be a category [g] [BACK VOWEL]. If [g] [ʊ] is encountered in the test set then this close category could be used, instead of going all the way back to the less useful context–independent [g]. Hence a top–down approach has a more elegant mechanism for dealing with the coverage issue.

2. Clustered biphones and triphones may still have insufficient training data. As in the issue of coverage, a bottom-up approach is forced to move from a triphone to a biphone, or possibly a context–independent model. This is because, unlike a top–down approach, no contextual hierarchy exists which would permit smoothing with a closely related but slightly more general model, which would still include relevant contextual information.

A possible advantage of a bottom–up approach is the fact that the data can fully drive the clustering mechanism. This avoids constraints imposed by *a–priori* assumptions. This advantage can manifest itself in several ways. First, although linguistic knowledge can be useful, expert linguists can not always agree on which contexts are acoustically similar. Second, phonetic combinations which are acoustically close, but which are not intuitively obvious from a linguistic standpoint, can be discovered by the clustering mechanism. Third, linguistic knowledge which is useful for a given phone in a given context, might be less useful in another instance. That is to say, a phonetic environment which has a large impact on one phone may have a different, or reduced, impact on another phone. Therefore, if a means could be found of addressing the two problems cited above, a bottom–up approach might be preferable.

This thesis seeks to utilize a bottom–up approach which permits a high–degree of training data pooling to overcome the sparse data problem, while still modelling a large number of triphone units and covering a high percentage of the possible phonetic combinations. The approach undertaken is inherently less vulnerable to the main difficulties generally encountered using a bottom–up approach. Once the main algorithm is described, experimental results based on the bottom–up clustering mechanism will be presented. The essential point is to show how the dynamic tracks can be utilized to capture segment level contextual effects. The thesis is not attempting to claim that bottom–up clustering is superior to top–down clustering.

7.2 Merging Tracks Based on Segment Dynamics

Although contextual influences can extend across several phonetic boundaries, it is most often the case that the effect of the phone in the left context position is primarily seen near the left boundary of a segment, and the effect of a right context phone is primarily seen near the right boundary. This can be clearly seen if we re–examine the spectrogram shown in Chapter 1. In the phone [ʌ] at 0.35 seconds, the left portion of the phone is primarily affected by the preceding [l] which is pulling down the second formant. This formant can be seen to move gradually towards its target location before being pulled up at the end to the alveolar locus at roughly 1.8 kHz in anticipation of the following [s]. The [ɛ] at 0.6 seconds is pulled slightly up on the left, again by the alveolar locus, and towards the end of the segment all of the formants start to fall in anticipation of the labial fricative [v]. In neither of these instances does the influence of the left or right context extend beyond the center of the segment they are affecting. This is essentially the case with all the segments in this utterance.

Therefore, as a first approximation, it will be assumed that the contextual effects of the phonetic environment can be modelled as influencing primarily the adjacent region of a segment. This simplification should enable us to capture the dominant contextual effects in a novel and efficient way. A method which will take advantage of the dynamic tracks is to *independently* account for the left and right contexts by creating biphone tracks, and then combine them to create triphone tracks as they are needed. Tracks can be estimated and stored for the left and right contexts separately and then *merged* when a synthetic segment is generated to create a triphone based synthetic segment.

Such an implementation would dramatically reduce the magnitude of both the coverage problem and the sparse data problem. For a system with $N = 58$ phonetic models, the maximum number of required context–dependent tracks is reduced from $N^3 = 195{,}112$, to $2 * N^2 = 6{,}728$, not accounting for the large number of transitions that never occur in English. The factor of two occurs because for a given transition we get two possible tracks (e.g. for [r] [ɑ] the [r] data is used for modelling a right context track for [r], and the [ɑ] data is used for modelling a left context track for [ɑ]). This can help to alleviate both the sparse data problem and the coverage problem associated with context–dependent modelling.

Additionally, since the error modelling techniques are independent of the track, the number of statistical error models (which require the majority of parameters) is a design parameter, since the errors can be pooled over different contexts. This means only the tracks themselves will be context–dependent. This pooling of the error matrices, if successful, will alleviate the sparse data problem with respect to the estimation of the statistical parameters.

Hence, the main ideas behind this approach are first, to generate robust biphone tracks, second to merge these tracks to generate triphone synthetic segments, and third to use the errors generated from these triphones to estimate the error covariance parameters. By pooling the errors, tracks can be created for a large number of contexts without compromising the estimates of the parameters used in the Gaussian models. Finally, since the left and right contexts are utilized independently, triphone tracks can be created "on the fly" if needed during test. That is, contexts never seen during training can be *created synthetically* from the left and right biphone tracks. This presents a possible method of greatly increasing the coverage provided by the training set.

The remainder of this section presents an investigation of this approach using data–driven bottom–up clustering. The clustering mechanism uses a metric based on the "distance" between two tracks. In each case, error matrices are pooled over all contexts. Context–dependent classification experiments are then performed with the context assumed to be known.

7.2.1 Bottom–Up Clustering

The key problem inherent in bottom–up clustering involves a lack of access to an intermediate representation when presented with sparse data or a trigram (sequence of three phones) in the test set which was not seen in the training set. This is the problem of lack of coverage. However, the technique of merging biphones could reduce the impact of these difficulties substantially. Fortunately it is possible to directly measure the impact of these two problems. To avoid creating tracks based on sparse data, measurements of distortion reduction on a separate data set will be made. If there are too many tracks created from the training set, they will not be robust, and distortions over an independent set will increase. Hence, the threshold at which clustering should be stopped is a measurable parameter. This is not an issue for the Gaussian distributions, since they are computed for each phone from the entire ensemble of errors computed over the phone's training data. The coverage over the test set can be measured explicitly, by counting the number of instances when either a left or right biphone clustered track is not available.

The algorithm to cluster the phonetic tokens to form context–dependent tracks is performed separately for each phone, once for all left contexts, and again for all right contexts. Given a phone and a left/right context, the algorithm is:

1. Create a separate track for every phonetic context in the training set. These are the "seed" biphone (left or right) tracks. Count the number of contributors to each track state.

2. For biphone tracks with only one or two contributors, merge these tracks with the biphone track nearest them using the *track distance metric* (TDM).

3. Compute the distance between all remaining tracks using the TDM. If the closest tracks have a TDM less than a threshold, merge them and repeat this step. Else stop.

This is a greedy clustering algorithm. The TDM is based on a step–wise optimal computation, but does not guarantee that the final clusters will be optimal [19]. The criterion takes into account the number of tokens in each cluster as well as the track distances from each other. A normalized Euclidean distance between the tracks is weighted by the number of contributors to each track state. In general, this makes the TDM favor adding smaller clusters to larger ones rather than merging two medium size clusters. In addition, each state is also multiplied by the same weight which will be used when the tracks are merged. Hence, the TDM is more heavily influenced by the left states when merging left context tracks, and by the right states when merging right context tracks.

Let P represent the number of acoustic attributes, M represent the number of states, and N represent the count for each state in a track. Then, given the merger weight for state i, w_i, the distance between two tracks for phone α in the two contexts β and γ, is TDM(T_β, T_γ):

$$TDM(T_\beta, T_\gamma) = \sum_{i=1}^{M} w_i \left[\frac{N_{\beta_i} * N_{\gamma_i}}{N_{\beta_i} + N_{\gamma_i}} \right] \sum_{j=1}^{P} \left[\frac{(T_{\beta_{ij}} - T_{\gamma_{ij}})^2}{\sigma_{\alpha j}^2} \right] \tag{7.1}$$

Recall that $\sigma_{\alpha j}^2$ are the phone dependent variances used to normalize the different acoustic attributes.

The threshold used in the clustering algorithm is an important design parameter. If the threshold is set too high, the clustering will continue for too many iterations. This will result in a small number of clusters and contextual resolution will be lost. If the threshold is set to be to low, then there will be many clusters, but there will be two risks. First, there will be too many clusters which were trained on only a few tokens. This will result in non–robust tracks which may not be representative enough to be useful. Secondly, when the error covariance matrix is estimated, the errors will be superficially low for these sparse clusters. This will result in too “tight” an error covariance matrix, and system performance will suffer. We can test the threshold by measuring the distortion on a set of data independent of the training set. If the net distortion on this independent set continues to fall as the number of clusters goes up (threshold goes down) then there is a high degree of confidence in the robustness of the tracks. When this trend begins to reverse itself it means too many clusters are being created and the threshold should be raised to ensure track robustness.

Phones	**Comment**
[r]	retroflexed
[ə] [ʌ] [o]	back vowels
[ɑ^{y}] [ɑ]	back vowels
[ɔ] [ɔ^{y}] [ḷ]	
[ɛ] [æ] [e] [ɑ^{w}]	mid–front vowels
[ə^{h}] [h]	aspiration
[ɚ] [ɝ]	retroflexed vowels
[b$^{□}$] [p$^{□}$] [g$^{□}$] [k$^{□}$] [▯] [m] [f] [v] [š] [n]	closures and labials
[SIL] [ʔ] [ṇ] [t$^{□}$] [ð]	silence and glottal stop
[i] [y] [ɦ] [ŋ]	palatal/velar
[ɪ]	front vowel
[ɨ]	front vowel
[w] [l]	low F2 semi–vowels
[s] [θ]	predominantly [s]
[u] [ʊ]	
[ü]	

Table 7.1: Example of clusters formed using the TDM. The clusters represent the right contexts which formed automatically for the phone [t]. Each cluster appears with the most frequently occurring phones listed first.

An example of 16 clusters which formed for the right context of the phone [t] are shown in Table 7.1. The table reveals that in many instances the tracks for [t] in similar acoustic contexts were clustered together. The third cluster is logical when it is realized that from the perspective of the previous phone [ɑ^{y}] and [ɑ] are often identical. The phone [ḷ] has acoustic characteristics which are very similar to a back vowel and is clustered with [ɔ] and [ɔ^{y}]. The phones [w] and [l] which are often confusable due to their acoustic similarity are also clustered together. The relative consistency with which many phones are clustered with phones that are similar in manner of articulation seems to indicate that the biphone tracks are capturing significant acoustic information.

Merging Biphone Tracks

The basic premise behind merging biphone tracks to create a triphone based synthetic segment is that the left context of a phone has its strongest influence on the left–most portion of a phone, and the right context has its strongest influence on the right. The mergers are computed using the TDM metric.

The following algorithm was used to conduct the mergers:

1. The synthetic segments for the two context–dependent biphone tracks are generated.

2. Each frame in the merged triphone synthetic segment receives contributions from each of the two synthetic segments just generated. These contributions are weighted using linear interpolation. The left context weights begin at a value of 1.0 for the first frame and end at a value of zero for the last frame. The right context weights for each frame are equal to the 1.0 minus the left context weight. Note that the sum of the weights at each frame is one.

A set of context–dependent tracks was created using the *MIT Training* data set. Since the classification experiments use the *MIT Test* set, a development set was not available for optimizing the clustering threshold. Therefore, a value of 20 was chosen for the threshold which resulted in 1,201 left context tracks and 1,348 right context tracks. Thus 2,549 tracks were created in total, an average of 44 biphone tracks for each of the 58 models. To determine if the tracks could be improved by adding a small amount of context–independent information to the center states as a form of "ballast," a similar set of 2,549 tracks was created using the same threshold. Also, to verify that this was not too large a number of tracks, a threshold of 30 was used to create an alternative set of 2,111 tracks. The distortions for the context–independent

Context	CI Track in Center	Threshold	Total # of Tracks	Mean Distortion (per frame)
Context–Independent (combined genders)	—	—	58	14.39
Context–Independent (gender specific)	—	—	116	13.57 (14.01m/12.72f)
Context–Dependent (combined genders)	Yes	30	2111	13.16
Context–Dependent (combined genders)	Yes	20	2549	13.15
Context–Dependent (combined genders)	No	20	2549	13.13
Context–Dependent (gender specific)	No	20	4167 (2398m/1769f)	12.03 (12.39m/11.33f)

Table 7.2: Distortions for context–independent tracks (gender combined and specific) and a variety of context–dependent track conditions.

tracks and all sets of context–dependent tracks, taken over the test set, are shown in Table 7.2. The table also shows the distortion which resulted from using the threshold of 20 to create sets of gender specific tracks.

The distortion results reveal that the threshold of 20 was not too small for maintaining robust tracks, since the distortions were reduced from the case where the threshold value was 30. They also reveal that the addition of information from the context–independent tracks was counter–productive. Therefore, for the experiments described below, the tracks resulting from a cluster threshold of 20 and the biphone merger algorithm detailed above (no context–independent information) will be used.

An example of a merged track is shown in Figure 7-2. The figure shows C_2 for a triphone track synthesized from left and right context biphone tracks. The left and right biphone tracks represent [ɔ] in the contexts [kɔ] and [ɔr] respectively. The triphone track for an [ɔ], in the context [kɔr], is then synthesized by merging the two biphone tracks. Note that the resulting triphone track closely resembles the left biphone track in the initial states and the right biphone track in the later states.

Of all the tokens in the test set, only 18 of them had contexts for which the correct

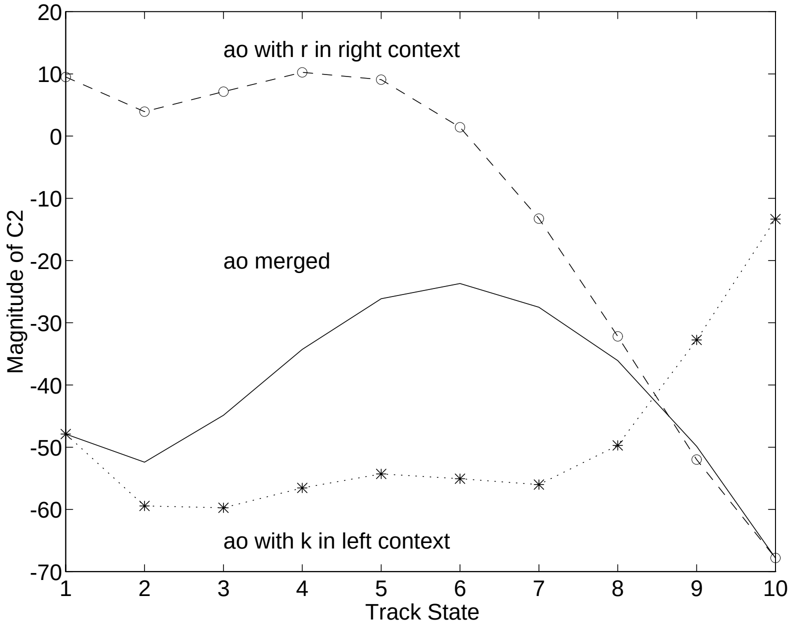

Figure 7-2: C_2 for a triphone track synthesized by merging two biphone tracks. The left and right biphone tracks represent [ɔ] (ao) in the contexts [kɔ] (k ao) and [ɔr] (ao r) respectively. The triphone track for an [ɔ], in the context [kɔr] (k ao r), is then synthesized by merging the two biphone tracks. Note that the resulting triphone track closely resembles the left biphone track in the initial states, and the right biphone track in the later states.

triphone synthetic segment, based on these clusters, could not be synthesized, and the context–independent track was used for the missing context in these cases. Hence, coverage was on the order of 99.8%. To reiterate, this high coverage was possible because contexts not seen in training could be synthetically created during test by merging the appropriate biphone tracks. This result is artificially high, since these training and test sets share some identical utterances. Therefore, this issue must be re–examined in Chapter 8 where the sentences used for training and test are disjoint sets.

The main point of this section is to show the improvement which can be obtained by just creating context–dependent tracks and pooling the errors to create a single covariance matrix for each phone. Therefore, it is not critical to optimize the threshold for clustering at this juncture. When the phonetic recognition experiments are conducted in Chapter 8, a development set will be available for optimizing this parameter. The key point is that a large number of robust tracks can be easily created to account for contextual factors.

7.2.2 Context–Dependent Experiments

A series of experiments was conducted using the 2,549 context–dependent tracks constructed from combining the genders, and the 4,167 gender specific context–dependent tracks. The context–dependent duration statistics are based on the clusters formed during training. The classification results are summarized in Table 7.3. The results are compared to the values achieved under identical conditions in the context–independent case. The first result, “Acoustic Scores” used a uniform distribution for the prior probabilities so as to isolate the effect of only adjusting the acoustic component. For the durations, counts were kept of how often each phone occurred in each tri–cluster context. This was found to be more robust than using the duration statistics for each individual triphone. The context–dependent duration correlations resulted in a relatively large performance improvement compared to the context–independent duration correlations. Under each experimental condition the context–dependent results were approximately 4% higher than the context–independent results.

The results indicate that merging biphones to create synthetic segments which incorporate both left and right contextual information can lead to significant performance improvements. It is important to emphasize that the actual contexts were known during the experiments, hence, this improvement represents an upper bound of what might be expected under more realistic circumstances. However, the results of the phonetic recognition experiments presented in Chapter 8, where the context must be determined, will support the idea that merging of biphone tracks is able to account for significant contextual effects in a way that improves system performance.

Description	**CI (%)**	**CD (%)**
Acoustic Scores	73.2	76.7
+ Duration Correlations	74.1	79.5
+ Priors (unigram)	75.1	79.8
+ Priors (trigram)	80.7	84.9
+ Enforced Gaussians (unigram priors)	76.8	80.6
+ Enforced Gaussians (trigram priors)	81.7	85.6

Table 7.3: Results for the *merger* context–dependent classification experiments. The results are shown for the comparable CI experiment and the CD experiment using bottom–up unsupervised clustering. All the experiments utilized duration correlation modelling in the Gaussian pdf's.

7.3 Transition Dynamics

Another method of using dynamic tracks to enhance system performance is to examine the acoustic information that spans adjacent segments. The idea is to make tracks of the phonetic transitions themselves. This lends itself well to the overall approach, since the transition regions are highly dynamic because the articulators are generally in motion during this interval. During classification/recognition, the transition model scores augment the segment scores to provide contextual information.

The main difficulty which needs to be overcome is the very large number of phonetic transitions which occur. Again, sparse data considerations limit the number of models which can be created. However, many transitions are very similar. While it may be impractical to capture all of the transitions, it may be possible to create a significantly large subset of transition models.

Other approaches have attempted to utilize a method of explicitly scoring the phonetic boundaries. Marteau et al. [49] used HMM models of diphones to classify 9 broad phonetic "macro–classes" of transitions. They concluded that dynamic information was critical for the recognition of high–speed or highly co–articulated segments, when it was often difficult to detect any target configuration. Leung *et* *al.* [27] used an MLP classifier to explicitly detect boundaries in a full recognition framework. As in the work by Marteau, the classifier was trained to recognize broad classes of transitions, which were linguistically motivated. The classification scheme combined the probability that each of two boundary frames were boundaries with the probability that all internal frames were not boundaries. By examining only those boundaries which occurred with a high probability, they were able to reduce the number of hypothesized segments for classification scoring by over an order–of–magnitude. Performance improvements due to the boundary classifier were not reported. More recently, Kimball *et al.* [33] used a similar scheme of combining probabilities for the boundaries with probabilities that internal frames were not boundaries, to determine an explicit segmentation score. In this work, the main classification engine was a Stochastic Segment Model. They report that without the explicit scoring of segment boundaries, system word recognition error increases 18%. Once again, this work used broad phonetic classes (a total of 6) based on manner–of–articulation.

The idea to be explored in this thesis is to use the TDM to cluster together transition tracks to arrive at a group of transitions which are representative of the major classes which such transitions fall into. These major transition classes would essentially be *canonical* transitions. However, rather than using a predetermined set of broad linguistic categories, bottom–up clustering will again be employed. This will allow a large number of unsupervised data–driven transition models to be created.

The transition models can potentially help in two ways. First, the transition scores will be incorporated into the overall scoring framework to help determine the phonetic identity of the two phones involved. Secondly, they can be examined to determine likely segment boundaries within an utterance. This reduces the possible search space when we do phonetic recognition, particularly since the transition likelihoods provide an idea of which phones are involved in the transition.

7.3.1 Transition Track Generation Function

The first issue which must be addressed is the type of generation function best suited for creating the transition tracks. In Section 4.3.2, a potential weakness of the trajectory invariance assumption was noted. When synthetic segments are generated, the rate at which the transition is assumed to take place is directly affected by the duration of the token (see Figure 4-10). This does not match our intuition, which is that the rate at which the articulators move from one configuration to another need not be related to the duration of the phonetic segments. In other words, the transitions to and from a 200 ms. [s] can be made at the same rate as the transitions to and from a 100 ms. [s]. Therefore, when modeling the transitions, it was decided to use a fixed number of frames, centered about the transition boundary, to create the tracks. This is a time invariance assumption with a center alignment point. During recognition experiments, every frame is a potential boundary frame, and can be used as the center frame in hypothesizing a transition. The number of frames to use in the transition tracks is a design parameter which will be chosen based on recognition performance over a development set.

In practice, the strategy will be to preprocess an utterance, by scoring frames every ΔT with each of the canonical transition models. This will effectively create a *segmentation map* of the utterance. At each hypothesized transition frame, the segmentation map will provide likelihood information pertaining to whether a transition took place, and also which of the canonical transitions are the most likely. These segmentation maps will be used during the search process of the phonetic recognition experiments discussed in Chapter 8.

7.3.2 Canonical Transition Design

The goal in creating canonical transition models is to create robust models of as many distinctly different types of transitions as possible. The robustness of each model will be determined by the amount of data we have available to estimate its covariance parameters, and is therefore adversely affected as the total number of models is increased. Therefore, to generate a large number of transition models, the number of phonetic categories was collapsed from 58 to 42. The 42 categories selected were based on the 39 categories designated by Lee [40] and listed in Table 6.9. In addition, the silence class was broken down into silence (*h#*, pause – [□], epinthetic silence – [▯]), voiced and unvoiced closures, and glottal stop [ʔ].

Using 42 classes means there are a total of 1,764 possible phonetic transitions. The data set *NIST Train* was used to analyze the transitions, and 1,275 distinct transitions were found. Tracks of each of these transitions were computed, with each track consisting of 21 states. In this instance the tracks represent the mean MFCC values calculated over an interval of 105 ms., with the center frame aligned using the phonetic boundaries in the TIMIT transcriptions.

The tracks are then clustered, in an unsupervised manner, with the same bottom–up algorithm used in the creation of the context–dependent tracks. The distance between the closest tracks, calculated using the track distance metric, is shown for each merger in Figure 7-3. However, in this instance it is not possible to use the phone dependent variances as the normalization weights in Equation 7.1. Instead, the MFCC variances calculated over all of the phones were used as common normalization weights for each transition model. These are the same σ_i^2 used to calculate distortion in Chapter 4.

To allow for a larger number of canonical transitions, the ΔMFCC's were not used as statistical features. Since duration is not a variable for discriminating between transition models, the total number of dimensions in the Gaussian pdf is 45, resulting in 1,013 independent parameters in each covariance matrix (recall the matrix is symmetric). This allowed for the creation of 200 canonical transition models. The robustness of the covariance matrices was checked by observing the range of eigenvalues and the size of the determinant.

An example of three different transition tracks is shown in Figure 7-4. The first cluster consists of the transitions [in], [en], and [iŋ], the second cluster consists of [ɛn], [æn], [ɑ^{y}n], and [ɔ^{y}n] transitions, and the third cluster consists of only the [ɪn] transitions. Although all three of these transition clusters could be considered as

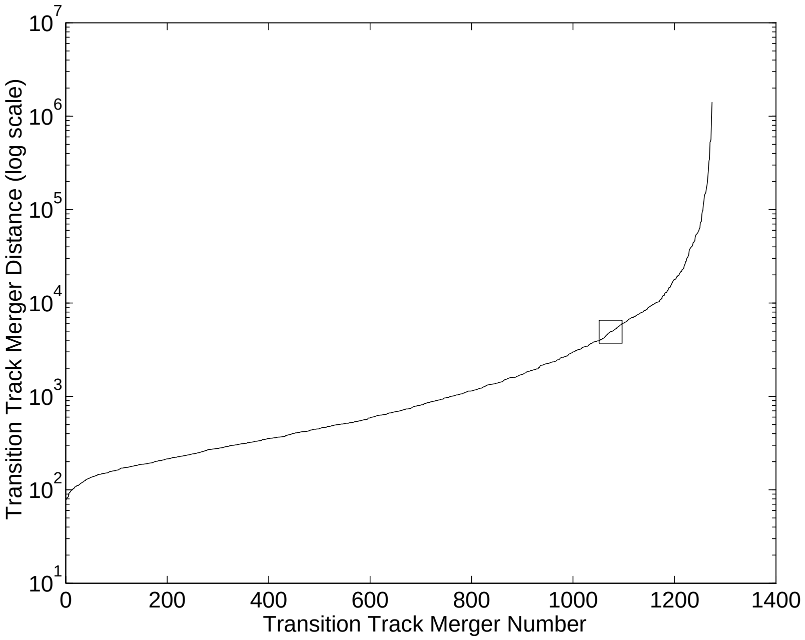

Figure 7-3: The distance between transition tracks at each merger during clustering. There are initially 1275 transition tracks, which are clustered to form 200 canonical transitions. The box marks the point at which the track mergers were halted (200 transition tracks remain). Note that the y–axis uses a log scale.

[front–vowel][n] transitions (with the exception of the [iŋ] component), yet the differences in their trajectory dynamics are apparent in the figure.

7.4 Chapter Summary

This chapter attempted to capture the effects of co–articulation in two ways. The first method involved creating tracks which account for contextual variability based on the acoustics within a phonetic segment. Traditional approaches to this problem have difficulties with sparse training data when attempting to model triphones, and also with lack of coverage of the test set. The technique employed here was to separately cluster biphone tracks which independently accounted for left and right context. The

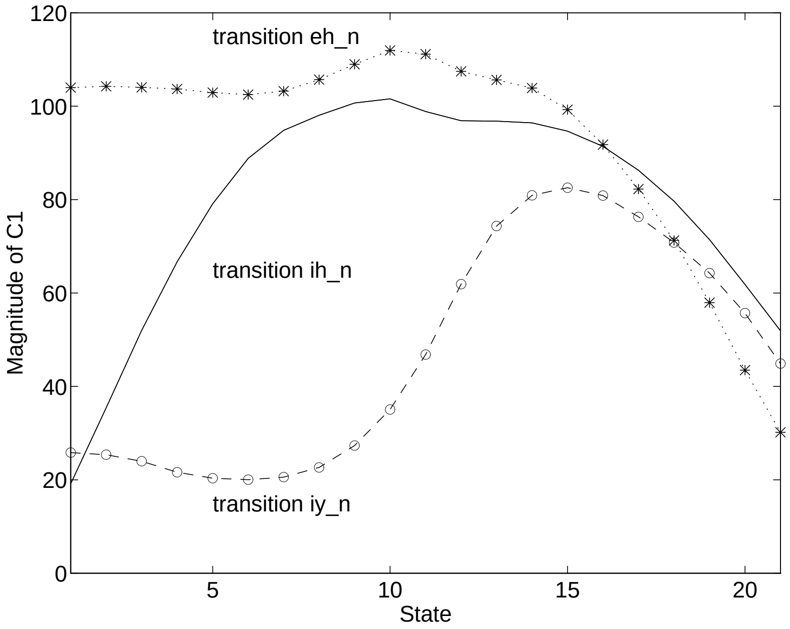

Figure 7-4: C_1 for three different [front–vowel][nasal] transition tracks. The three tracks are taken from a cluster consisting of [in] (iy n), [en] (ey n), and [iŋ] (iy ng) transitions, a cluster consisting of [ɛn] (eh n), [æn] (ae n), [ɑ^{y}n] (ay n), and [ɔ^{y}n] (oy n) transitions, and a cluster of only [ɪn] (ih n) transitions. The standard deviation of C_1 for these tracks is on the order of 55 in the center, and is slightly higher towards the ends.

biphone tracks are then merged as they are needed to create a triphone track. By pooling the errors across all contexts to estimate the statistical parameters, the sparse data problem is significantly reduced. Furthermore, since triphones can be created synthetically for contexts not seen in the training data, the coverage over the test set is near 100%. This result will be verified in the next chapter, when coverage is measured using a test set that does not contain any utterances that overlap with the training set.

Classification experiments showed that using a single Gaussian distribution for each phonetic model and the unigram priors resulted in a performance increase of 4.7% for the context–dependent tracks. This is a decrease in the error rate of nearly

18.9%. When the gender specific models are used, the classification error is reduced from 18.3% to 14.4%. This is a reduction in error rate of 21.3%. Therefore, the reduction from merging the biphone tracks appears to be consistently in the area of 20%. When the more realistic phonetic recognition experiments are conducted, the actual phonetic context will not be known. However, since the error rates will be uniformly higher on this task, there is more room for potential improvement using the triphone synthetic segments. Their impact will be re–evaluated in the next chapter.

The second method used to capture the co–articulatory effects was to model information that occurred across segment boundaries. The acoustical parameters in the transitions exhibit a large degree of dynamic behavior due to the motion of the articulators. To robustly capture the significant dynamic events, a set of 200 *canonical* transitions were created by clustering the tracks created by modelling all of the transitions in the training set. These canonical transitions will be used to create a segmental map of each utterance. The segmental maps will serve to both reduce the search space during recognition experiments, and to provide information about the identity of the phones involved in the transition.

Chapter 8

Search and Phonetic Recognition

The problem of phonetic recognition is to determine the most likely sequence of phonetic units, $A = \{a_1, a_2, \ldots, a_N\}$, by searching a sequence of acoustic observations $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_T\}$. In a segment based system, each phonetic unit will also have an explicitly hypothesized starting and ending boundary associated with it. The sequence of boundaries which partitions the utterance into phonetic units is called a segmentation of the utterance, and is denoted by $S = \{s_1, s_2, \ldots, s_N\}$. Each s_i is a frame number specifying the left phonetic boundary for the following phone. The observations associated with segment, s_i, will be denoted by $X(s_i) = \{\vec{x}_{s_i}, \vec{x}_{s_i+1}, \ldots, \vec{x}_{s_{i+1}-1}\}$.

Therefore, defining S_i as a specific segmentation from the set S of all possible segmentations of X, the problem can be stated as:

$$A^{\star} = \underset{A}{\text{argmax}} \; p(A|X) \; = \; \underset{A}{\text{argmax}} \sum_{S_i \in S} p(AS_i|X) \tag{8.1}$$

As in the case of HMM's, most segment based approaches do not compute the summation on the right hand side of Equation 8.1. HMM's generally are implemented with a Viterbi algorithm that only computes the most likely state sequence for an utterance. Analogously, most segment based approaches [27, 33, 44], use only the most likely joint labeling and segmentation.

$$p(A|X) \;=\; \underset{S}{\text{argmax}} \;\; p(AS|X) \tag{8.2}$$

This approximation greatly simplifies the implementation, and for HMM's has been shown to have no significant effect on performance [52, 64]. Equation 8.1 now becomes:

$$A^{\star} = \underset{AS}{\text{argmax}} \;\; p(AS|X) \tag{8.3}$$

By Bayes Law, the term we are maximizing over in Equation 8.3 can be rearranged as:

$$p(AS|X) \;=\; \frac{p(X|AS)p(AS)}{p(X)} \tag{8.4}$$

The denominator term $p(X)$ is simply a normalization factor, and will not affect the most likely label sequence. Therefore, it can be dropped from the expression. Expanding the joint distribution $p(AS)$ in the numerator now yields:

$$A^{\star} \;=\; \underset{AS}{\text{argmax}} \;\; p(X|AS)p(S|A)p(A) \tag{8.5}$$

The terms in Equation 8.5 can be interpreted as distinct scoring components. The $p(X|AS)$ reflects the acoustic component of the score, given an hypothesized segmentation and a sequence of phonetic models. The term $p(S|A)$ will nominally be approximated by $p(S)$. This term will be modelled simply as a "segment transition weight," it is strictly a function of the number of segments in S. Thus it serves to control the tradeoff between deletion and insertion errors. The $p(A)$ term corresponds to the phonetic grammar of the utterance. This is the *a–priori* probability of the full hypothesized phonetic string. The details of this grammar, known as a language model, are discussed in Section 8.1.2.

Assuming independence between adjacent segments, and ignoring the transition model component of the system for the moment, yields:

$$p(X|AS)p(S|A)p(A) \;=\; p(A)\prod_{i=1}^{N} p(X(s_i)|a_i s_i)p(s_i|a_i) \tag{8.6}$$

To incorporate the transition component of the acoustic score, it is necessary to define some additional notation. Assume that the transition likelihood at frame s_i is being computed. These calculations involve δ frames on each side of s_i. The acoustic observations at this boundary that are used to calculate the transition likelihoods will be denoted $X(\delta_i) = \{\vec{x}_{s_i-\delta}, \ldots, \vec{x}_{s_i+\delta}\}$. Now Equation 8.6 becomes:

$$p(X|AS)p(S|A)p(A) \;=\; p(A)\prod_{i=1}^{N} p(X(s_i)X(\delta_i)|a_i s_i)p(s_i|a_i) \tag{8.7}$$

The relationship between the acoustic observations which comprise a segment and the observations which are involved in the transition dynamics is depicted in Figure 8-1. Note that the acoustic observations within each $X(\delta_i)$ will be used twice, once for the acoustic score of the segment, and again for the transition from the given segment to the following segment. However, it is important to recall that the acoustic likelihoods are calculated not from the observations, but from the *errors* produced when the synthetic segment associated with phone a_i is compared to the observations. Since the transition models and the phone models produce different synthetic segments for comparison to the observation sequences $X(s_i)$ and $X(\delta_i)$ respectively, the errors produced by each of these calculations can be approximated as independent. This yields:

$$p(X(s_i)X(\delta_i)|a_i s_i) \;=\; p_I(X(s_i)|a_i s_i)p_T(X(\delta_i)|a_i s_i) \tag{8.8}$$

where p_I represents the likelihood of the internal component of a_i, and p_T represents the likelihood of a transition from phone a_i to phone a_{i+1}.

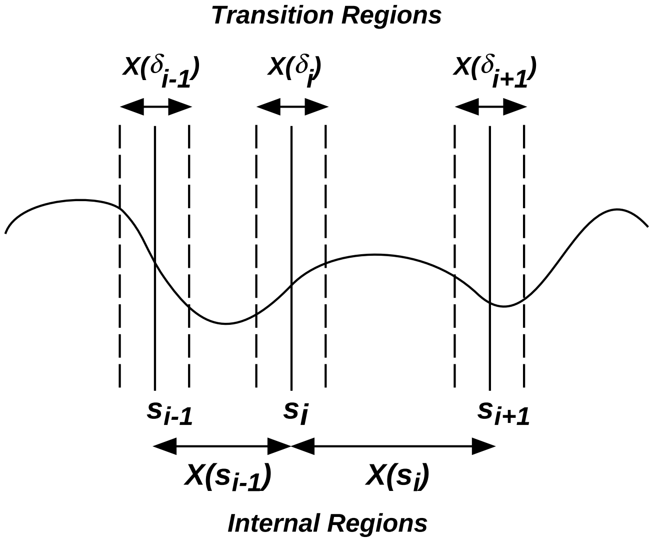

Figure 8-1: A portion of a schematized acoustic attribute, partitioned into segments. Each segment consists of internal and transition regions.

Therefore, the context–independent score for an utterance is:

$$A^\star = \underset{AS}{\operatorname{argmax}} \; p(A)\prod_{i=1}^{N} p_I(X(s_i)|a_i s_i)p_T(X(\delta_i)|a_i s_i)p(s_i|a_i) \tag{8.9}$$

The p_T component is used only when transition models are included.

To incorporate contextual dependencies, define γ_j to be the triphone $\{a_{j-1}, a_j, a_{j+1}\}$. Now Equation 8.9 becomes:

$$A^\star = \underset{AS}{\operatorname{argmax}} \; p(A)\prod_{i=1}^{N} p_I(X(s_i)|\gamma_i s_i)p_T(X(\delta_i)|\gamma_i s_i)p(s_i|\gamma_i) \tag{8.10}$$

8.1 Implementation Issues

This section contains a brief discussion of implementation issues pertaining to the Viterbi search algorithm. More detailed descriptions of this algorithm can be found in [22, 64]. Differences between the context–independent and context–dependent search strategies will be discussed. The section concludes with an examination of the language models used in the search (the $p(A)$ term in the derivation above), as well as the duration model component.

8.1.1 The Viterbi Algorithm

The Viterbi algorithm is a time–synchronous beam search algorithm which utilizes dynamic programming. The search can be envisioned as finding the best path through a lattice. The axes represent time (x–axis) and phonetic model (y–axis). Each (x,y) coordinate represents a potential phonetic boundary at a specific time. Depending on the implementation strategy, the phonetic model coordinate can represent either the beginning or the end of a phonetic segment with the label specified by the coordinate. In this thesis, the search is conducted from left–to–right (increasing time) at an interval of ΔT frames. The value of ΔT determines how frequently a phonetic boundary is hypothesized (e.g. 10 ms). A segment is specified by an arc, connecting two points in the lattice. An example of a partial path through the lattice is shown in Figure 8-2.

The cost associated with a segment is computed according to Equation 8.9 for context–independent recognition and Equation 8.10 for context-dependent recognition. At each point in time at which a segment boundary is hypothesized, the best path to reach each vertex at that time is retained. The best path is the one with the least cost (highest probability) associated with it. The search need only keep track of the best path to reach a vertex because of the dynamic programming aspect of the implementation. Therefore, at each vertex in the search, all that must be stored is the cost up to that time, along with a pointer to the vertex attached to the initial point of the arc.

By conducting an exhaustive search through the lattice, the complete path is constructed. The node with the least cost at the final frame is used to determine the phonetic label sequence along with the segmentation. Exhaustive search is computationally expensive. To compute the segment scores for M models with boundaries

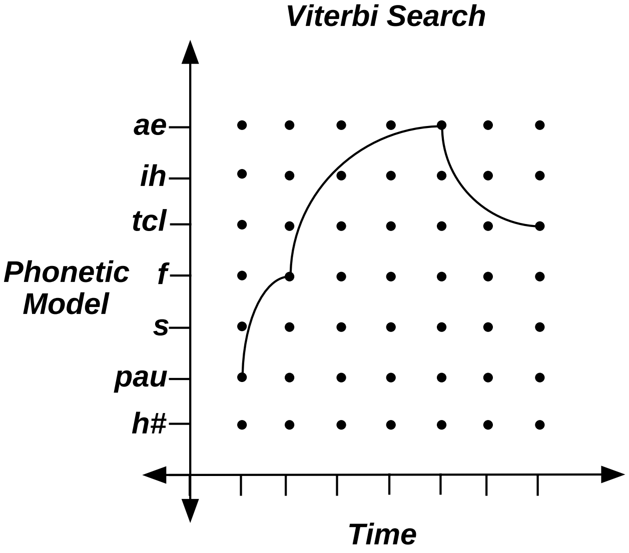

Figure 8-2: A portion of a hypothetical path in a Viterbi search.
The vertices are connected by arcs, which each have a cost associated with them. A vertex can represent either a left or right phonetic boundary, depending on the implementation strategy.

hypothesized at each of N frames requires a total of $MN(N-1)/2$ acoustic likelihood computations. However, this is assuming context–independence of each of the phonetic models. To perform a context–dependent search, the computational requirement becomes heavier. In this instance a single acoustic score connecting a vertex to a previous time is not sufficient. Instead, a context–dependent score must be computed based on the track of the phonetic unit which incorporates the contextual information. This adds an additional factor of M to the search computation for a biphone acoustic representation, and potentially a factor of M^2 for a triphone computation.

The specific implementation strategy becomes a key factor for context–dependent search. The designation of each vertex as a starting or ending boundary for the phonetic model at that coordinate becomes important. If the terminal vertex of each arc is designated as the endpoint for the phone specified by the vertex, then the left context in that path will be known, since it is the departure point for the arc. However, the potential right context associated with that arc is completely unknown. For example, using this type of implementation, the second arc in Figure 8-2 represents an [æ], with an [f] as its left context. But when the acoustic score for the [æ] is computed, no right context can yet be hypothesized for the merger to form a triphone synthetic segment.

The alternative is to designate the terminal vertex of an arc to be the starting boundary of the next phone in the sequence. Under this implementation, the second arc in Figure 8-2 is an hypothesized [f] going to an [æ]. Now, both contexts can be used to create the synthetic segment, since the [f] is already assumed to be coming from a [□] (pause). This will also allow for trigram constraints to be used in other system components. However, this is not a full triphone implementation, since every arc departing from the [f] is assumed to have the same left context. Essentially, the search is being pruned by assuming that the left context for a given vertex is the best context in spite of the fact that the right context has yet to be accounted for. To hypothesize alternative left contexts would require that a pointer be kept to the best path for each possible preceding phone model, potentially adding a factor of M to the computation cost.

8.1.2 Language and Duration Models

Both the language and duration models applied during the search are functions of the amount of contextual information being accounted for. The term context–independence applies to the acoustic–phonetic models, of which duration is a component. It is a common practice to incorporate phonotactic constraints, in the form of a grammar, at the context–independent level.

Language Model

The *a–priori* probability of a phonetic string, $p(A)$, can be written as:

$$p(A) \;=\; p(a_1)p(a_2|a_1)p(a_3|a_2a_1)\ldots p(a_N|a_{N-1}a_{N-2}\ldots a_1) \qquad (8.11)$$

However, due to limitations in the amount of training data which exists, it is not tractable to estimate $p(A)$ accurately for realistic values of N. Therefore, a language model is used which incorporates simplifying assumptions. The assumptions are generally of the form that the *a–priori* probability of a particular phone, a_i, is conditionally dependent only on the identity of it's closest phonetic neighbors. Different levels of complexity include a unigram model in which the phones are assumed to be independent, a bigram model in which the the priors are conditioned on either the left or right phonetic context, or a trigram model which generally conditions the prior probability on either the two preceding or following phones.

The power of a language model is often measured by the reduction in perplexity that it provides. The perplexity, P, is related to the estimate of $p(A)$ [35]:

$$P = p(A)^{-1/N} \tag{8.12}$$

In very loose terms, the perplexity can be interpreted as the average number of phones which can follow a given phone after applying the language model. For this thesis, the language models used are based on the full set of 58 phones shown in Table 4.1. Given the 58 phone lexicon, and using the NIST designated training and core test sets, the perplexity was measured to be 46.1 for a unigram model, 15.7 for a bigram model, and 15.1 for a trigram model. This small reduction in going from a bigram to a trigram is consistent to that measured by Lamel and Gauvain using a 61 phone lexicon (17.7 bigram and 17.0 trigram) [38]. This is attributable to the limited training data available for trigram estimation, and possibly, to the fact that there are no overlapping sentences in the training and test sets.

Duration Model

The duration models used in this chapter are all based on measurements of the statistics of the log duration, as measured in frames. The log duration was chosen based on the fact that this assumption was shown to provide a better explanation of the data (see Appendix A). For both the context–independent and context–dependent experiments, the log duration is included as an attribute in the feature vector, so that its correlations with the MFCC errors can be measured.

For the context–dependent experiments, the mean of the log duration was measured for each cluster–based triphone. Pooling the statistics across the contexts used for clustering the acoustic parameters proved to be more robust than using a triphone model for each individual context. If an unseen context is hypothesized during test, then the context–independent log duration is used.

8.2 Scoring for Phonetic Recognition

Due to the search that must take place during phonetic recognition, scoring recognition experiments is more complex than it is for phonetic classification experiments. Since the number of segments being hypothesized may not be the same as the number of segments in the actual transcription, it is not sufficient to state that a particular segment is either correctly or incorrectly labelled. Instead, errors besides substitution errors can take place. These errors are deletions and insertions, and refer to phones in the utterance which are missed during the search process, and extra phones hypothesized during search which are not present in the provided transcription. The actual segment alignment times are not used in the scoring process.

To score an utterance, the reference transcription (that which is provided with the TIMIT database) is compared to a hypothesized transcription. A NIST designated alignment program is then used to align the two transcriptions such that the total number of errors (substitutions, deletions, and insertions) is minimized [59]. The phonetic *accuracy* is then defined to be one minus the percentage of errors in the utterance:

$$\%Errors = \%Substitutions + \%Deletions + \%Insertions \tag{8.13}$$

$$\%Accuracy = 1.0 - \%Errors = \%Correct - \%Insertions \tag{8.14}$$

$$\%Correct = \frac{number\ of\ phones\ correct}{number\ of\ phones\ in\ utterance} \tag{8.15}$$

Note that it is mathematically possible to have an accuracy of less than zero, due to the presence of insertion errors.

8.3 Context–Independent Recognition

The recognition experiments presented in this section are based on context–independent acoustic models. The context–independent experiments provide a baseline standard upon which subsequent improvements, due to the incorporation of contextual information, can be measured. Also, a large body of research reported in the literature exists which is based on context–independent phonetic recognition. Hence, a basis of comparison is possible between this approach and other phonetic recognition approaches.

To refine the system design parameters, models were trained on the *NIST Train* data set and initial evaluation was performed using *Dev1* data set (see Table 2.2 for details). To save time, the evaluation experiments were run using the *BG–Dev1* data set, a subset of the *Dev1* data set, consisting of twelve male speakers.

All the experiments presented in this section use gender specific, context–independent acoustic models. The models are constructed in exactly the same manner as those used in the context–independent classification experiments described in Chapter 6, including the use of duration and the ΔMFCC's. The grammar is a phone bigram, based on the same 58 phonetic classes used in the classification experiments, and for which acoustic models were constructed. During the Viterbi search phonetic boundaries are proposed every 10 ms.

After the system parameters had been tuned on the development set, models were constructed for evaluation on three distinct test sets. First, the models used in the development set were used in recognition experiments on the *NIST Core Test* data set. This data set has been used many times for reporting context–dependent results [38, 62, 68], but has not generally been used at the context–independent level.

Therefore, the results reported here both provide a baseline for other researchers in the field, and serve to measure improvements when transitional and contextual factors are used. The *NIST Core Test* data set also has the feature, alone among the data sets used in the literature, of having sentences which are completely disjoint from those used in the training set (*NIST Train*). This lack of overlap between training and test set utterances precludes the grammar model from containing a favorable statistical bias towards the overlapping sentences in the test set.

To provide an additional basis of comparison between STM and other approaches, models were also constructed for use with two additional test sets used in the research community. Early successes in phonetic recognition using HMM's were achieved by Lee and Hon [40]. To use their results as a point of comparison, models were trained on the *KFL Train* data set, and tested on the *KFL Test* data set. The test set is identical to that used by Lee and Hon, but the training set is somewhat larger, reflecting the release of additional TIMIT data by NIST. Segment–based phonetic recognition results utilizing representations which incorporate dynamic information have been reported for the Stochastic Segment Model [13] and the Dynamic Systems model [15]. Models were trained using the set of male speakers contained in the *BU Train* data set (426 speakers) and tested on the *BU Test* data set which consisted of twelve male speakers from the western dialect region of the United States. Again, the training set used here was somewhat larger due to the release of additional TIMIT data by NIST.

The results of the context–independent experiments, both with and without the use of the phonetic transition models, are presented in Table 8.1. To utilize the transition models in a context–independent manner, the log likelihoods for all the transitions from the phone being scored were exponentiated and summed. This results in a transition likelihood which is conditioned only on the hypothesized phone. All of the transition models used data from both genders, if available, in the training set.

Table 8.1 shows the accuracies tabulated according to each of three distinct criteria. The *nominal* method is the accuracy using the same allowable confusions used in the classification experiments and defined in Table 6.9, and also includes glottal

Statistical Trajectory Model (STM) Results			
Test Set	Scoring Method to Measure Accuracy		
	Nominal	Remove [ʔ]	Closure/Stop
NIST Core Test	61.9%	61.9%	63.0%
+ transition models	63.9%	64.0%	64.9%
KFL Test	63.9%	63.6%	65.0%
+ transition models	65.6%	65.3%	66.7%
BU Test	66.0%	66.2%	67.2%
+ transition models	68.3%	68.4%	69.5%

Table 8.1: Context–independent recognition results for three data sets using a bigram grammar. The results are presented for each of three alternative methods of scoring the errors.

stops ([ʔ]) which are mapped to silence. The *Remove* [ʔ] condition shows the effect of removing all incidents of [ʔ] from both the reference and hypothesis transcriptions. This is the most common method used to report results in the literature [38, 40, 68]. Finally, the *Closure/Stop* condition shows the results which occur when substitutions of a closure–stop pair by the corresponding single closure or stop are acceptable errors. If a closure–stop pair is hypothesized for a single closure or stop and the result is an insertion error, this insertion error is not counted [15, 18]. In this case, instances of [ʔ] were not removed. The *Closure/Stop* condition seems to make good sense within the context of constructing words, as these types of confusions would then be irrelevant.

As can be seen from Table 8.1, the data sets vary considerably with respect to recognition performance. The superior results for the *KFL Test* and *BU Test* data sets should be partially attributable to the fact that the training and test sets have overlapping sentences, providing a bias for the bigram language model. In addition, the *NIST Core Test* data set is balanced with respect to dialect, with two male speakers and one female speaker from each of eight pre–defined regions. The significant improvements for the *BU Test* data set might also be attributable to the fact that only male speakers from a single dialect region are used. The TIMIT data base provides twice the amount of data for males as it does for females. For the other test sets, the accuracies for the male speakers were anywhere from 1.0% to 2.5% higher than the corresponding accuracies for the females, depending on the use of the transition models (which were constructed from data of both genders).

Finally, the removal of the glottal stops, [ʔ] had little impact on performance. This generally boosts performance by one or two tenths of a percent, but on the *KFL Test* data set it actually causes a drop in the measured performance of 0.3%. The value of keeping or dropping the glottal stops is debatable; however, dropping them has been a common standard adopted in the literature. Allowing closure–stop substitutions by either the closure or the stop boosted performance uniformly by 0.9% to 1.4%. Again, this criterion seems to reflect the fact that, when searching for words, either the correct closure or stop would be beneficial. Some additional confusions at the phonetic level might also be considered to be allowable. This would include permitting confusions of syllabic phones with a schwa–phone counterpart. For example, the distinction between [l̩] and [ə] [l] is often arbitrary, and irrelevant at the word level.

Table 8.2 compares the results from other work reported in the literature to the relevant results obtained in this thesis. The accuracies achieved on the *KFL Test* data set are a considerable improvement to Lee and Hon's accuracy, and are comparable to the accuracy they achieved using context–dependent models (66.1%). However, these results are old (1989) and also the accuracy measure was not used at that time. Lee attempted to maximize percent correct while maintaining insertions under twelve percent. With a lower insertion rate, it is possible his accuracy performance would have improved. Without transition models, the STM results are closest to those of the stochastic segment model (SSM). This is not surprising in light of the fact that the SSM model used to achieve this result included some local modelling of temporal correlations [13].

The accuracy results were consistently better on the smaller test sets than on the *NIST Core Test* data set. This most likely reflects the fact that there is no overlap of sentences in training set and the test set, and also, that the test set is carefully balanced across all eight dialect regions of the TIMIT corpus.

Accuracy Comparison of STM to other Work			
Work	Accuracy	Relevant STM Accuracy	
		w/o transitions	with transitions
KFL KFL Test Set	53.3%	63.6%	65.3%
Dynamic Systems BU Test Set	63%	66.2%	68.4%
SSM BU Test Set	66.7%	67.2%	69.5%

Table 8.2: Comparison of results achieved using statistical trajectory models to other work reported in the literature.

8.4 Context–Dependent Recognition

The recognition experiments reported in this section are based on a set of gender dependent, context–dependent models. Due to the additional computation required for a context–dependent search, the total number of MFCC's and ΔMFCC's was reduced from 15 to 13. This lowered the dimensionality of the Gaussian error distributions from 76 to 66. The models were trained on the *NIST Training* data set. The tracks were clustered bottom–up using the TDM described in Chapter 7. A total of 2,492 gender dependent tracks were generated via bottom–up clustering, using the TDM.

The errors for each phone were pooled into a single covariance matrix for each of the gender models. As was the case for the context–independent experiments, the recognizer design parameters were tuned using the *Dev1* and *BG–Dev1* data sets. Since the context–dependent experiments impose a significantly larger computational burden than the context–independent experiments, the actual experiment tuning was conducted using only the *BG–Dev1* data set. When statistical measures were used to choose design parameters (e.g. the mean track distortion is used to determine the threshold for clustering), the *Dev1* data set was used.

The issue of coverage was discussed in Chapter 7 and can now be re–examined. In Chapter 7, coverage over the data set used in the context–dependent classification experiments was found to be 99.8%. However, the training and test sets used in that chapter contained some overlap of the sentences. As discussed earlier, the NIST designated training and test sets were created specifically to avoid this problem. Triphone coverage was found to be 99.6% for the *Dev1* set, and 99.5% for the *NIST Core Test* data set. This consistently high degree of coverage supports the intuition that merging the biphone tracks to create triphone tracks is an effective solution to the coverage problem. If just the original triphone contexts provided in the training set are used, the coverage on the *Dev1* and *NIST Core Test* drops to 87.6% and 87.1% respectively.

The Viterbi search was conducted by hypothesizing alternative right contexts for each phonetic model at each point in time (i.e., at 10 ms intervals). The left context and boundary for each model were constrained to be those used in the best path achieved up to the current time. Hence, the search was over all possible bigrams. However, since the best left context was "known" for each hypothesized segment, the acoustic models utilized a full triphone track, created by merging the "known" left context biphone track with each allowable right biphone track. Additionally, a trigram phonetic grammar was also employed. Hence, this implementation strategy allowed the use of triphone and trigram information while maintaining a bigram search. Essentially, this constitutes a *constrained trigram* search.

The results of the context–dependent experiments, both with and without the use of the phonetic transition models, are presented in Table 8.3. For these experiments, the transition likelihoods were used to help determine the right context. Two sets of 200 transition models were constructed. One set was composed only of data from the male training speakers and was used on the male test utterances. The other set used all utterances in the training set and was used for the female test speakers.

Once again the improvement due to the transition models is noteworthy. The improvement of 3.0% in recognition accuracy is almost 50% better than that achieved in the context–independent experiments (∼2%). This is not surprising since the contextual information in the transition models was not available in the context–independent experiments. The improvement in the complete context–dependent sys-

Statistical Trajectory Model (STM) Results		
Test Set	Scoring Method to Measure Accuracy	
	Nominal	Remove [ʔ]
Trigram	66.5%	66.5%
+ transition models	69.3%	69.5%

Table 8.3: Context–dependent recognition results for *NIST Core Test* data set using a "constrained" trigram grammar/search. The results are presented with and without glottal stop.

tem was 5.5%. This absolute improvement is slightly higher than was obtained in the context–dependent classification experiments (∼4%). However the relative improvement in terms of reduction of the error is ∼15% in recognition whereas it was ∼21% in classification. As noted in Chapter 7, the fact that the context was known during classification provided a significant advantage.

The best result using statistical trajectory models is compared with other results in the literature in Table 8.4. The results in the table represent the best accuracy achieved on the NIST designated Core Test data set at the current time. The STM result of 69.5% is virtually identical to that achieved by Lamel and Gauvain (69.1%) [38]. Their result represents the state–of–the–art in HMM phonetic recognition, which is currently the dominant speech recognition technology. It is also very close to the best SUMMIT result of 68.5% [62]. The SUMMIT result used context–independent mixture Gaussian pdf's and context–dependent models to determine phonetic transitions. Robinson, using artificial neural networks, achieved an accuracy of 73.9% [68]. This value is significantly higher than any other accuracy reported in the literature on this test set. Clearly, the neural net is capturing important information that is not being modelled well in other approaches. One hypothesis might be that the neural net is making advantageous use of a language model which incorporates high N–gram constraints. However, this hypothesis is not supported by the small decrease in perplexity which occurs as N is increased beyond a value of 2 for the NIST training and core test sets. Also, both Lamel and Gauvain [38], as well as the STM approach show

Comparison of Recognition Results on NIST Core Test Set	
ANN – Robinson	73.9%
STM	69.5%
HMM – Lamel & Gauvain	69.1%
SUMMIT – Phillips & Glass	68.5%

Table 8.4: Best recognition results for the *NIST Core Test* data set reported in the literature. The results are presented using Kai–Fu Lee's 39 confusion classes after removing glottal stops.

only small performance gains in going from a bigram to a trigram. Other possibilities are that the neural network could be estimating a significantly more accurate probability density function, or, incorporating contextual information in a superior way. The possible use of neural nets in an STM framework is discussed briefly in the next chapter.

8.5 Chapter Summary

This chapter described the application of statistical trajectory models to the phonetic recognition task. This task is significant because it is an important component of all modern large vocabulary speech recognition systems. The task requires the use of both acoustic classification and search components. A formalism for computing the search likelihoods for an utterance was described, including the incorporation of phonetic transition likelihoods and contextual dependencies.

A series of context–independent and context–dependent recognition experiments were then performed. Several different scoring methods were utilized so that the impact of each method could be assessed. Finally, a set of context–dependent recognition experiments using the NIST designated training and test sets was performed. The results indicate that the performance of the STM methodology is virtually identical to that achieved by the most successful HMM technology currently available. Since this thesis represents a first attempt at exploiting the potential of the STM approach, it is hoped that further efforts and experimentation will allow significant advances in phonetic recognition performance. Additional avenues of future research using this approach are discussed in the next chapter.

Chapter 9

Conclusions and Future Work

This thesis attempted to establish the importance of capturing the temporal behavior of the speech waveform for use in a speech recognition engine. The temporal behavior is modelled by templates of the dynamics of the acoustic attributes used to represent the waveform, and by estimating their spatio–temporal correlation structure. Models incorporating these two components were created for phonetic units and for phonetic transitions.

This work represents only an initial attempt at utilizing the statistical trajectory model concept. The main findings which resulted from this investigation are summarized in the next section. The thesis then concludes with a discussion of how these results might be expanded upon to produce additional gains in speech recognition performance.

9.1 Conclusions and Thesis Contributions

The implementation of statistical trajectory models encompassed several distinct steps which were of major importance to the eventual task of applying STM to phonetic recognition. In general, each step served to refine the selection and use of each of the two key model components, the tracks and the statistical error models.

The first task involved determining the methodology and design parameters for generating the tracks. It is the tracks which are responsible for capturing the non–linear dynamics of the acoustic attributes. In order to evaluate the relative accuracy of the tracks, it was necessary to define a metric which would provide a meaningful quantitative measure of the "distance" between the tracks and the tokens they are intended to represent. This lead to the use of a *distortion* criterion in Chapter 4. The distortion criterion does not depend on a particular track algorithm, but instead provides a means of evaluating competing algorithms. To apply the criterion fairly, each algorithm involved in the analysis must be compared to the same data. Note that it would also be possible to measure the distortion produced using an HMM by creating a "trajectory" consisting of the mean value of the distribution (or the expectation over the mixture components) used to score each frame.

The distortion criterion provides a useful means of breaking down the design process of the statistical trajectory models. It allows the tracks and generation functions to be evaluated independently of the statistical models. Subsequent track improvements, such as context–dependent modelling, speaker adaptation, noise normalization, etc. can all be evaluated in a simple manner without conducting extensive recognition experiments. This is because it is generally the case that tracks which result in a larger distortion are unlikely to result in performance improvements.

Using distortion measurements, a generation function was selected for use in subsequent experiments. The generation function, "fractional linear interpolation with fixed endpoints," is a linearly interpolated mapping of a token's frames to the states of the track. The initial and final frames of the token are always aligned with the initial and final states of the track. This generation function carries the implication that the trajectory through the acoustic space is not affected by durational variabilities in the realization of a phone. Naturally, when a phone's duration is correlated to the phonetic context, then co–articulation will affect the trajectory. However, the results here indicate that, when averaged over all contexts in the training set, the trajectory invariance assumption is a viable one.

Chapter 5 focused on the second main task, the creation of a statistical error model. The objective of the statistical model is to take advantage of information residing in the correlations both over time and between attributes. This objective involves dealing with two key difficulties. The first difficulty is that the observation sequence varies in duration. For each segment that is hypothesized, the observation sequence will be N frames long, where N is variable. The second difficulty involves the dimension of the distribution, which is limited by the amount of training data available.

A method which produced a good compromise was to create error sequences of variable frames and to normalize the duration by averaging the vector sequence over each of Q pieces. This technique had the advantages of reducing the dimensionality of the Gaussian distribution while utilizing all of the data.

The benefits of sub–segmenting the error signal instead of sub–segmenting the acoustic attributes themselves was demonstrated. This was done by measuring the residual energy caused by the sub–segmentation approximation. This result supports the use of a track to model the dynamics at the phonetic level. By accounting for the dynamics prior to making a statistical model, the accuracy of the signal after sub–segmentation is enhanced.

Chapter 5 concluded with some insights into the structure of the information in the covariance matrix. After determining some initial design parameters, Chapter 6 presented some vowel classification experiments, both as an initial evaluation of STM and as a means of assessing the impact of the different components in the covariance matrix. It was demonstrated in a vowel classification experiment that the temporal correlations had a performance impact at least as great as the spatial (inter–acoustic) correlations. The vowel classification results, which compared favorably to other results reported in the literature, indicated that the STM's were doing a reasonable job of scoring phonetic segments. This conclusion was supported by a full set of context–independent phonetic classification experiments. The result of 76.8% compares favorably to other results reported in the literature using MFCC's and full Gaussian covariance statistics. The importance of the ΔMFCC's, the correlation of duration with the acoustic attributes, and performance increases obtained by training separate models for each gender was also demonstrated.

Chapter 7 addressed the incorporation of the effects of co–articulation into the STM framework. Two types of modelling were attempted, both of which can be utilized in a complete phonetic recognition system. The first technique attempts to account for the dynamic effects of co–articulation in the tracks. This technique is able to handle the two main difficulties which arise when creating context–dependent models, sparse data problems created by splitting the data up into a large number of contexts, and coverage of contexts not seen in the training set. The sparse data problem is resolved by creating context–dependent tracks and pooling the resulting errors over context. Merging biphone tracks to synthesize triphone tracks as they are needed was shown to provide a high degree of coverage of the test set. An important result was that significant performance improvements were shown to be possible by accounting for contextual effects with the track alone. This allows the system to operate with only a single probability density function for each phonetic unit, or two if separate gender models are used. Lamel and Gauvain use 500 full HMM's to model context in their system [38].

The second technique which was applied to the co–articulation problem involved making STM's which spanned the segment boundaries in an attempt to accurately capture the dynamics of the phonetic transitions. Sets of 200 (or more) "canonical" transition models were created using a bottom–up, data–driven, clustering algorithm. The models are highly dynamic, due to the motion of the articulators during phonetic transitions, and are therefore well suited to the STM approach. The transition models were used to augment the acoustic scores and also to help reduce the search space during phonetic recognition.

Finally, in Chapter 8, the different design elements presented throughout this thesis were combined to create a complete phonetic recognizer. Chapter 8 described the search problem and formally developed the scoring framework to be used during the Viterbi search. The language and duration model components were also described. This was followed by a description of several sets of context–independent phonetic recognition experiments. These experiments established that STM's were capable of performance that compared favorably with other context–independent results reported in the literature. Additionally, performance improvements resulting from the use of the transition models, in a context–independent manner, were demonstrated. Hopefully, the benchmarks established in this section will be of use to future researchers as a point of comparison.

The last section in Chapter 8 presented the results when the full context–dependent STM's and transition model likelihoods were used. The transition models increased the recognition accuracy by 3.0%, which is slightly better than the increases achieved for the context–independent results. This is most likely due to the fact that the transitional context information was not used in the context–independent experiments. The context–dependent accuracies were 4.6% and 5.5% higher than the corresponding context–independent accuracies, measured with and without the transition models respectively. The best result of 69.5% is virtually identical to that achieved by Lamel and Gauvain of 69.1% [38]. Their result represents the state–of–the–art in HMM phonetic recognition, which is currently the dominant speech recognition technology.

It appears that the use of both dynamic models of the acoustic attributes and their temporal correlations can be beneficial in determining an accurate phonetic transcription of an utterance. It should be reiterated that this work represents only a first attempt at incorporating these elements into a complete speech recognition system. Additional research should lead to further performance improvements. Some areas where this improvement can possibly be found are the topic of the next section.

9.2 Future Work

The goal in this work has been to ascertain the viability of the concepts behind STM, rather than to exhaustively attempt the promising implementation alternatives. However, each of the model components could easily have been implemented in a different way, and many potential enhancements exist. A few of these possibilities will be briefly described here.

Several generation functions were considered as a basis for forming the tracks. The ones examined here were motivated by two alternative assumptions which attempt to account for the durational variability which occurs in the realization of the same phonetic unit. Certainly other types of assumptions could be made and other generation functions explored. It might also be of value to re–examine the trajectory invariance assumption with the context–dependent tracks. Once the phonetic context has been accounted for, the validity of this assumption could be more readily ascertained.

Generation functions which perform a type of dynamic time–warping may serve as an enhancement. However, it is unclear whether this flexibility is actually advantageous at the phone level. Generation functions which are a function of the phonetic identity might also be employed. Better clustering techniques, and an improved track distance metric are also areas where the tracks could be improved. For example, the track distance metric in this work used a linear weighting of the states when merging tracks. A better weighting might be related to the phone dependent variances of the individual states.

Instead of always merging biphone tracks, the original triphone tracks could also be incorporated. It might be possible to use a procedure similar to that used in language modelling where a combination of the trigram and bigram statistics are used. In this case, the triphone tracks could be weighted and combined with the merged biphone track. The weight which determined the exact combination of triphone track and merged biphone track would depend on the number of examples which were available in the training data.

Another area which is well suited to a dynamically based approach would be to create compound phonetic units, such as [æl], or [ɔr]. It might also be possible to adapt the tracks during recognition to account for speaker–dependent characteristics. The small number of parameters needed in a track makes for a much simpler adaptation scheme than re–estimation of the statistical parameters.

The transition tracks also warrant further investigation. One simple refinement not explored here would be to retain a large number of transition tracks, and to pool the errors across sets of clustered tracks to create a smaller number of statistical models. This approach was effective when applied to the context–dependent biphone tracks.

The statistical error models could also be improved in several ways. The dimensionality problem has a highly constraining effect on the use of the temporal correlation information. A principle components analysis might be one method of getting around this problem. A large value could initially be used for Q, the sub–segmentation parameter, and then the dimension could be reduced via principle components [32]. Mixtures of diagonal Gaussians might also yield a more robust/accurate representation of the distribution in the acoustic space. Alternatively the errors could be fed directly into an artificial neural network. A neural network would be completely unconstrained in estimating the shape of the probability density function.

The search algorithms used to obtain the recognition results could easily be enhanced. The system employed here used a very small set of tunable design parameters (on the order of five). A more sophisticated search engine could be employed in the future. A full triphone $A^\star$ search would also probably boost system performance. It might also be possible to dynamically adjust the relative weighting of the segment and transition acoustic scores depending on the log likelihoods produced for a given hypothesized segment.

Finally, feedback from higher level knowledge sources which take into account super–segmental features of the utterance should be incorporated into the search. In addition, speech recognition engines would also be greatly enhanced if superior methods of rapidly adapting to speaker–dependent characteristics are effectively incorporated.

Appendix A

Modelling Phonetic Durations

An important component of a segment based approach is the information inherent in the durations of different phonetic units. Duration is a segment level feature that can be used to help differentiate phones which are acoustically similar. For example the vowel pair [æ] and [ɛ] are easily confused acoustically, but the mean duration of an [æ] is approximately 27.3 frames, while the mean duration of an [ɛ] is approximately 18.7 frames. The durations vary with context, speaker, placement within a sentence, and other factors.

The duration is measured in frames and is therefore always an integer value. Since the measurements are discretely valued, one method of creating a probability distribution of duration would be to count the number of tokens which occur for each duration for each of the phones. However, this method can lead to inaccuracies at the extremes of the distribution, or at intermediate values due to sparse data or an unusual tendency within the training set. Therefore, a continuous probability density function is generally selected to approximate the actual discrete distribution. The continuous density serves to smooth the data and provides a more robust measure at the extremes.

Three different probability density functions were considered as duration distribution models. They were a gamma distribution, a Gaussian distribution, and a log

Gaussian distribution. Duration distributions, particularly for short phones, have tails which are lengthened at longer durations and are more compact at the shorter durations. This is partially due to the limiting factor that the duration is always positive. The gamma distribution was chosen as a candidate since it shares this attribute. For the same reason. a Gaussian distribution is at a disadvantage, due to it's symmetry about the mean value and the fact that it will assign some component of the probability to negative durations. Therefore, a log Gaussian was also chosen as a possible distribution.

The accuracy of the distributions was measured by computing the distribution parameters from a training set, and then measuring the likelihoods of the training tokens using each of the distributions just created. This method provides a direct measure of the relative accuracy which the distributions are able to achieve.

The data set used for this computation was the *MIT Train* data set, and the results are summarized in Table A.1. The log Gaussian distribution was significantly superior to the Gaussian distribution and slightly superior to the gamma distribution. An additional advantage of the Gaussian and log Gaussian assumptions, is that they allow the durational information to be modelled jointly with the acoustic parameters. This permits relevant correlations between the duration errors and errors in the acoustic attributes to be captured. This combination of advantages lead to the choice of the log Gaussian distribution for modelling the duration of phonetic segments.

phn	*Gauss*	*Log Gauss*	*Gamma*	*phn*	*Gauss*	*Log Gauss*	*Gamma*
ɑ	-0.7613	-0.7432	-0.7442	ǰ	-0.2939	-0.2775	-0.2810
æ	-0.7903	-0.7766	-0.7776	k	-1.0591	-1.0558	-1.0451
ʌ	-0.7142	-0.6809	-0.6880	l	-1.3360	-1.2954	-1.2972
ɔ	-0.7047	-0.6922	-0.6919	m	-1.0624	-1.0622	-1.0512
ɑʷ	-0.2448	-0.2386	-0.2398	n	-1.7437	-1.7026	-1.6979
ə	-0.0811	-0.0729	-0.0751	ŋ	-0.3399	-0.3339	-0.3323
əʰ	-0.9675	-0.9452	-0.9444	ɾ̃	-0.1243	-0.1252	-0.1241
ɚ	-0.8179	-0.7979	-0.7986	o	-0.5683	-0.5524	-0.5550
ɑʸ	-0.7050	-0.6951	-0.6950	ɔʸ	-0.1117	-0.1114	-0.1111
b	-0.3809	-0.3592	-0.3605	p	-0.7373	-0.7315	-0.7242
č	-0.2418	-0.2348	-0.2355	ʔ	-0.8275	-0.8091	-0.8048
d	-0.5542	-0.4985	-0.5102	r	-1.3793	-1.3539	-1.3486
ð	-0.6525	-0.6223	-0.6231	s	-2.0876	-2.0519	-2.0505
ɾ	-0.3445	-0.3378	-0.3373	š	-0.4246	-0.4216	-0.4205
ɛ	-1.0579	-1.0219	-1.0283	silence	-3.9936	-3.4707	-3.6920
l̩	-0.3117	-0.3035	-0.3044	t	-1.1407	-1.1266	-1.1173
n̩	-0.1946	-0.1917	-0.1911	θ	-0.2464	-0.2461	-0.2435
□	-0.2265	-0.1998	-0.2081	ʊ	-0.1715	-0.1664	-0.1670
ɝ	-0.6182	-0.6060	-0.6067	unvcl	-3.7539	-3.6221	-3.6236
e	-0.7840	-0.7602	-0.7646	u	-0.2006	-0.1936	-0.1943
f	-0.7580	-0.7699	-0.7609	ü	-0.4861	-0.4616	-0.4657
g	-0.2641	-0.2552	-0.2544	v	-0.5531	-0.5452	-0.5429
h	-0.2878	-0.2772	-0.2781	vcl	-2.1167	-2.0487	-2.0474
ɦ	-0.2010	-0.2008	-0.1992	w	-0.7167	-0.6957	-0.6943
ɪ	-1.2979	-1.2486	-1.2567	y	-0.2982	-0.2857	-0.2871
ɨ	-1.9564	-1.8879	-1.8943	z	-1.1495	-1.1045	-1.1133
i	-1.6144	-1.5515	-1.5620	ž	-0.0541	-0.0517	-0.0522
Sum over all tokens in data set					**-44.51**	**-42.87**	**-43.11**

Table A.1: Log likelihoods (x10^{-4}) of the tokens in the data set *MIT Train* assuming different probability density functions. The phone models are the same as the 58 models defined in Table 4.1 except that the data for all the voiced closures (b□, d□, g□) have been pooled into the model, *vcl*, and the unvoiced closures (p□, t□, k□) have been pooled into the model *unvcl*.

Bibliography

[1] L. Bahl, F. Jelinek, and R. Mercer, “A Maximum Likelihood Approach to Continuous Speech Recognition,” *IEEE Trans. Pattern Anal. Machine Intell.*, Vol., PAMI–5, pp. 179–190, 1983.

[2] L. R. Bahl, P. V. deSouza, P. S. Gopalakrishnan, D. Nahamoo and M. A. Picheny, “Context–Dependent Modeling of Phones in Continuous Speech Using Decision Trees,” *Proc. DARPA Speech and Natural Language Workshop*, pp. 264–269, Pacific Grove, CA, February, 1991.

[3] J. Baker, “Stochastic Modeling for Automatic Speech Understanding,” *Readings in Speech Recognition*, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.

[4] L. Baum, “An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes,” Inequalities, Vol. 3, pp. 1–8, 1972.

[5] R. Cardin, Y. Normandin, and E. Millien, “Inter–Word Coarticulation Modeling and MMIE Training for Improved Connected Digit Recognition,” *Proc. ICASSP 93*, pp. 243–246, Minneapolis, MN, April, 1993.

[6] R. Carlson and J. Glass, “Vowel Classification Based on Analysis-by-Synthesis,” *Proc. ICSLP 92*, pp. 575–578, Banff, Canada, October, 1992.

[7] B. Chigier and H. Leung, “The Effects of Signal Representations, Phonetic Classification Techniques, and the Telephone Network,” *Proc. ICSLP 92*, pp. 97–100, Banff, Canada, October, 1992.

[8] Y. L. Chow, R. Schwartz, S. Roucos, O. Kimball, P. Price, F. Kubala, M. Dunham, M. Krasner and J. Makhoul, "The Role of Word–Dependent Coarticulatory Effects in a Phoneme Based Speech Recognition System," *Proc. ICASSP 86*, pp. 1593–1596, Tokyo, Japan, April, 1986.

[9] R. Cole and Y. Muthusamy, "Perceptual Studies on Vowels Excised from Continuous Speech," *Proc. ICSLP 92*, pp. 1091–1094, Banff, Canada, October, 1992.

[10] A. Dempster, N. Laird, D. Rubin, "Maximum Likelihood Estimation from Incomplete Data," *Journal of the Royal Statistical Society (B)*, Vol. 39, No. 1, pp. 1–38, 1977.

[11] P. Denes and E. Pinson, *The Speech Chain,* Anchor Press/Doubleday, Garden City, NY, 1963.

[12] L. Deng, "A generalized hidden Markov model with state–conditioned trend functions of time for the speech signal," *Signal Processing* 27, Vol 1, pp. 65–78, April, 1992.

[13] V. Digilakis, M. Ostendorf, and J. Rohlicek, "Fast Algorithms for Phone Classification and Recognition Using Segment–Based Models," *IEEE Trans. Signal Processing*, December, 1992.

[14] V. Digilakis, J. Rohlicek, and M. Ostendorf, "A Dynamical System Approach to Continuous Speech Recognition," *Proc. ICASSP 91*, pp. 289–292, Toronto, Canada, May, 1991.

[15] V. Digilakis, "Segment–Based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition." Ph.D. Thesis, Boston University, 1992.

[16] V. Digilakis, J.R. Rohlicek and M. Ostendorf "ML Estimation of a Stochastic Linear System with the EM Algorithm and Its Application to Speech Recognition." *IEEE Trans. Speech and Audio Processing*, Vol. 1, No. 4, pp. 431–442, October, 1993.

[17] Personal communications with V. Digilakis, Feb., 1993.

[18] Electronic communications with V. Digilakis, Feb. 7, 1994.

[19] R. Duda and P. Hart, *Pattern Classification and Scene Analysis,* John Wiley and Sons, New York, NY, 1973.

[20] G. Fant, *Acoustic Theory of Speech Production,* Mouton and Co., The Hague, Netherlands, 1960.

[21] R. Feldtkeller and E. Zwicker, "Das Ohr als Nachrichtenempfanger," S. Hirzel, Stuttgart, Germany, 1956.

[22] G. Forney, "The Viterbi Algorithm," *Proc. IEEE*, Vol. 61, pp. 268–278, March, 1973.

[23] W. Francis and H. Kucera, *Frequency analysis of English usage: Lexicon and grammar,* Houghton Mifflin, Boston, MA 1982.

[24] S. Furui, "Speaker Independent Isolated Word Recognition based on Dynamics Emphasized Spectrum," *Trans. IECE of Japan*, Vol. 69, No. 12, pp. 1310–1317, December, 1986.

[25] H. Gish and K. Ng, "A Segmental Speech Model with Applications to Word Spotting," *Proc. ICASSP 93*, pp. 447–450, Minneapolis, MN, April, 1993.

[26] W. Goldenthal and J. Glass, "Modelling Spectral Dynamics for Vowel Classification," *Proc. Eurospeech 93*, pp. 289–292 Berlin, Germany, September, 1993.

[27] H. Leung. I. Hetherington, and V. Zue, "Speech Recognition using Stochastic Explicit–Segment Modeling," *Proc. Eurospeech 91*, pp. 931–934, Genova, Italy, September, 1991.

[28] X. Huang, F. Alleva, M. Hwang, and R. Rosenfeld, "An Overview of the SPHINX–II Recognition System," *Proc. ARPA Workshop on Human Language Technology*, pp. 81–86, Plainsboro, NJ, March, 1993.

[29] M. Hwang and X. Huang "Subphonetic Modeling with Markov States – Senone," *Proc. ICASSP 92*, pp. 33–36, San Francisco, CA, March, 1992.

[30] M. Hwang and X. Huang "Shared–Distribution Hidden Markov Models for Speech Recognition," *IEEE Trans. Speech and Audio Processing*, pp. 414–420, Vol. 1, No. 4, October, 1993.

[31] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Trans. ASSP*, Vol. ASSP–23, No. 1, pp. 67–72, February, 1975.

[32] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis,* Prentice–Hall, Englewood Cliffs, NJ, 1982.

[33] O. Kimball, M. Ostendorf, and R. Rohlicek, "Recognition Using Classification and Segmentation Scoring," *Proc. DARPA Speech and Natural Language Workshop*, pp. 197–201, Harriman, N.Y., February, 1992.

[34] P. Ladefoged, *A Course in Phonetics*, Harcourt Brace Jovanovich, Inc., New York, NY second edition 1982.

[35] F. Jelineck, "Self–Organized Language Modeling for Speech Recognition," *Readings in Speech Recognition*, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.

[36] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: design and analysis of the acoustic-phonetic corpus," *Proc. DARPA Speech Recognition Workshop*, Report No. SAIC-86/1546, pp. 100–109, Palo Alto, CA, February, 1986.

[37] L. Lamel and J. L. Gauvain, "Identifying Non–Linguistic Speech Features," *Proc. Eurospeech 93*, pp. 23–30, Berlin, Germany, September, 1993.

[38] L. Lamel and J. L. Gauvain, "High Performance Speaker–Independent Phone Recognition Using CDHMM" *Proc. Eurospeech 93*, pp. 121–124, Berlin, Germany, September, 1993.

[39] K. F. Lee, “Large–Vocabulary Speaker–Independent Continuous Speech Recognition: The SPHINX System,” Ph.D. dissertation, Computer Science Department, Carnegie Mellon Univ., 1988.

[40] K. F. Lee and H. W. Hon, “Speaker–Independent Phone Recognition Using Hidden Markov Models,” *IEEE Trans. ASSP*, Vol. 37, No. 11, pp. 1641–1648, November, 1989.

[41] K. F. Lee, S. Hayamizu, H.–W. Hon, C. Huang, J. Swartz and R. Weide, “Allophone Clustering for Continuous Speech Recognition,” *Proc. ICASSP 90*, pp. 749–752, Albuquerque, NM, April 1990.

[42] S Levinson, L. Rabiner, and M Sondhi, “An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition,” *The Bell System Technical Journal*, Vol. 62, No. 4, pp. 1035–1074, April, 1983.

[43] H. Leung and V. Zue, “Phonetic Classification using Multi–Layer Perceptrons,” *Proc. ICASSP 90*, pp. 525–528, Albuquerque, NM, April, 1990.

[44] H. Leung. I. Hetherington, and V. Zue, “Speech Recognition using Stochastic Segment Neural Networks,” *Proc. ICASSP 92*, pp. 613–616, San Francisco, CA, March, 1992.

[45] H.C. Leung, B. Chigier, and J.R. Glass., “A Comparative Study of Signal Representations and Classification Techniques for Speech Recognition,” *Proc. ICASSP 93*, pp. 680–683, Minneapolis, MN, April, 1993.

[46] D. M. Lubensky, “Generalized Context–Dependent Phone Modeling Using Artificial Neural Networks,” *Proc. Eurospeech 93*, pp. 1477–1480 Berlin, Germany, September, 1993.

[47] S. Maeda, “On Articulatory and Acoustic Variabilities,” *Journal of Phonetics*, Vol. 19, pp. 321–331, 1991.

[48] J. Marcus, “Phonetic Recognition in a Segment–Based HMM,” *Proc. ICASSP 93*, pp. 479–482, Minneapolis, MN, April, 1993.

[49] P. Marteau, G. Bailly, and M. Janot–Giorgetti, “Stochastic Model of Diphone–Like Segments Based on Trajectory Concepts,” *Proc. ICASSP 88*, pp. 615–618, New York, NY, April, 1988.

[50] H. Meng and V. Zue, “A Comparative Study of Acoustic Representations of Speech for Vowel Classification using Multi–Layer Perceptrons,” *Proc. ICSLP 90*, pp. 1053–1056, Kobe, Japan, November, 1990.

[51] H. Meng, “The use of distinctive features for automatic speech recognition,” S.M. Thesis, Massachusetts Institute of Technology, September 1991.

[52] N. Merhav and Y. Ephraim, “Hidden Markov Modeling Using the Most Likely State Sequence,” *Proc. ICASSP 91*, pp. 469–472, Toronto, Canada, May, 1991.

[53] P. Mermelstein and S. Davis, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences,” *IEEE Trans. ASSP*, Vol. 23, No. 1, pp. 67–72, February, 1975.

[54] N. Morgan and H. Bourlard, “Continuous Speech Recognition using Multilayer Perceptrons with Hidden Markov Models,” *Proc. ICASSP 90*, pp. 413–416, Albuquerque, NM, April, 1990.

[55] H. Murveit, J. Butzberger, V. Digilakis, and M. Weintraub, “Progressive–Search Algorithms for Large–Vocabulary Speech Recognition,” *Proc. ARPA Workshop on Human Language Technology*, pp. 87–90, Plainsboro, NJ, March, 1993.

[56] M. Ostendorf and S. Roucos, “A Stochastic Segment Model for Phoneme–Based Continuous Speech Recognition,” *IEEE Trans. ASSP*, Vol. 4, No. 12, pp. 1857–1869, December, 1989.

[57] M. Ostendorf, A. Kannan, O. Kimball, and J.R. Rohlicek, "Continuous Word Recognition Based on the Stochastic Segment Model," *DARPA Proc. on Continuous Speech Recognition Workshop*, September, 1992.

[58] M. Ostendorf, I Bechwati, O. Kimball, "Context Modeling with the Stochastic Segment Model," *Proc. ICASSP 92*, pp. 389–392, San Francisco, CA, March, 1992.

[59] D. Pallett, W. Fisher, and J. Fiscus, "Tools for the Analysis of Benchmark Speech Recognition Tests," *Proc. ICASSP 90*, pp. 97–100, Albuquerque, NM, April, 1990.

[60] M. Phillips, J. Glass and V. Zue, "Automatic Learning of Lexical Representations for Sub–Word Unit Based Speech Recognition Systems," *Proc. Eurospeech 91*, pp. 577–580, Genova, Italy, September, 1991.

[61] M. Phillips and V. Zue, "Automatic Discovery of Acoustic Measurements for Phonetic Classification," *Proc. ICSLP 92*, Banff, Canada, October, 1992.

[62] M. Phillips and J. Glass, "Phonetic Transition Modelling for Continuous Speech Recognition," *J. Acoust. Soc. Amer.*, Vol. 95, No. 5, pp. 2877, June, 1994.

[63] P. Price, W. Fisher, J. Bernstein and D. Pallett, "A Database for Continuous Speech Recognition in the 1000 Word Domain," *Proc. ICASSP 88*, pp. 651–654, New York, NY, April, 1988.

[64] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, February, 1989.

[65] L. Rabiner and S. Levinson, "Isolated and Connected Word Recognition — Theory and Selected Applications," *Readings in Speech Recognition*, Morgan Kaufmann Publishers, Inc., San Mateo, CA 1990.

[66] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals,* Prentice Hall, Inc., Englewood Cliffs, NJ, 1978.

[67] T. Robinson and F. Fallside, “A Recurrent Error Propagation Network Speech Recognition System,” *Computer Speech and Language,* 5(3):259–274, 1991.

[68] T. Robinson, “Several Improvements to a Recurrent Error Propagation Phone Recognition System,” *Technical Report CUED/TINFENG/TR.82*, Cambridge University Engineering Dept., September, 1991.

[69] T. Robinson, “Recurrent Nets for Phone Probability Estimation,” *DARPA Proc. on Continuous Speech Recognition Workshop*, September, 1992.

[70] S. Roucos, M. Ostendorf, H. Gish, and A. Derr, “Stochastic Segment Modelling Using the Estimate–Maximize Algorithm,” *Proc. ICASSP 88*, pp. 127–130, New York, NY, April, 1988.

[71] M. Russell, “A Segmental HMM for Speech Pattern Modelling,” *Proc. ICASSP 93*, pp. 499–502, Minneapolis, MN, April, 1993.

[72] M. Schroeder, “Recognition of complex acoustic signals,” *Life Sci. Res. Rep.,* T. Bullock, Ed., Vol. 55, pp. 323–328, 1977.

[73] R. Schwartz, T. Anastasakos, F. Kubala, J. Makhoul, L. Nguyen, and G. Zavaliagkos, “Comparative Experiments on Large Vocabulary Speech Recognition,” *Proc. ARPA Workshop on Human Language Technology*, pp. 75–80, Plainsboro, NJ, March, 1993.

[74] R. Schwartz, Y. L. Chow, O. Kimball, S. Roucos, M. Krasner and J. Makhoul, “Context–Dependent Modeling for Acoustic–Phonetic Recognition of Continuous Speech,” *Proc. ICASSP 85*, pp. 1205–1208, April 1985.

[75] F. K. Soong and A. Rosenberg, “On the use of Instantaneous and Transitional Spectral Information in Speaker Recognition,” *Proc. ICASSP 86*, pp. 877–880, Tokyo, Japan, April, 1986.

[76] Technical Staff, The Analytic Sciences Corp., *Applied Optimal Estimation,* MIT Press, Cambridge, MA, 1974.

[77] R. Van Son and L. Pols, "Formant Movements of Dutch Vowels in a text, read at normal and fast rate," *Journal of the Acoustical Society of America*, Vol. 92, No. 1, July, 1992.

[78] A. Waibel and K. F. Lee, *Readings in Speech Recognition*, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.

[79] V. Zue, "The Use of Speech Knowledge in Automatic Speech Recognition," *Proc. IEEE*, Vol. 73, No. 11, pp. 1602–1614, November, 1985.

[80] V. Zue, J. Glass, M. Phillips, and S. Seneff, "Acoustic Segmentation and Phonetic Classification in the SUMMIT System," *Proc. ICASSP 89*, Glasgow, Scotland, May, 1989.

[81] V. Zue, J. Glass, D. Goodine, M. Phillips, and S. Seneff, "The SUMMIT Speech Recognition System: Phonological Modelling and Lexical Access," *Proc. ICASSP 90*, Albuquerque, NM, April, 1990.

[82] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff, "Recent Progress on the SUMMIT System," *Proc. Third DARPA Speech and Natural Language Workshop*, Hidden Valley, PA, June, 1990.