UNIVERSITY OF BIRMINGHAM

DOCTORAL THESIS

The Application of Continuous State HMMs to an Automatic Speech Recognition Task

---

Author:
Chloe SEIVWRIGHT

Supervisors:
Prof. Martin RUSSELL
Dr. Steve HOUGHTON

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy in the

Speech Recognition by Synthesis Group
Electronic, Electrical Systems Engineering

July 4th, 2019

Declaration of Authorship

I, Chloe SEIVWRIGHT, declare that this thesis titled, "The Application of Continuous State HMMs to an Automatic Speech Recognition Task" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

Abstract

Hidden Markov Models (HMMs) have been a popular choice for automatic speech recognition (ASR) for several decades due to their mathematical formulation and computational efficiency, which has consistently resulted in a better performance compared to other methods during this period. However, HMMs are based on the assumption of statistical independence among speech frames, which conflicts with the physiological basis of speech production. Consequently, researchers have produced a substantial amount of literature to extend the HMM model assumptions and incorporate dynamic properties of speech into the underlying model. One such approach involves segmental models, which addresses a frame-wise independence assumption. However, the computational inefficiencies associated with segmental models have limited their practical application. In recent years, there has been a shift from HMM-based systems to neural networks (NN) and deep learning approaches, which offer superior performance compared to conventional statistical models. However, as the complexity of neural models increases, so does the number of parameters involved, requiring a greater dependency on training data to optimise model parameters.

This present study extends prior research on segmental HMMs by introducing a Segmental Continuous-State Hidden Markov Model (CSHMM) examining a resolution to the issue of inter-segmental continuity. This is an alternative approach when compared to contemporary speech modelling methods that rely on data-centric NN techniques, with the goal of establishing a statistical model that more accurately reflects the speech production process. The Continuous-State Segmental model offers a flexible mathematical framework which can impose a continuity constraint between adjoining segments addressing a fundamental drawback of conventional HMMs, namely, the independence assumption. Additionally, the CSHMM also benefits from a practical training and decoding algorithm which overcomes the computational inefficiency inherent in conventional decoding algorithms for traditional Segmental HMMs.

This study has formulated four trajectory-based segmental models using a CSHMM

framework. CSHMMs have not been extensively studied for ASR tasks due to the absence of open-source standardised speech tool-kits that enable convenient exploration of CSHMMs. As a result, to perform sufficient experiments in this study, training and decoding software has been developed, which can be accessed in (Seivwright, 2015).

The experiments in this study report baseline phone recognition results for the four distinct Segmental CSHMM systems using the TIMIT database. These baseline results are compared against a simple Hidden Markov Model-Gaussian Mixture Model (HMM-GMM) system. In all experiments, a compact acoustic feature representation in the form of bottleneck features (BNF), is employed, motivated by an investigation into the BNFs and their relationship to articulatory properties. Although the proposed CSHMM systems do not surpass discrete-state HMMs in performance, this research has demonstrated a strong association between inter-segmental continuity and the corresponding phonetic categories being modelled. Furthermore, this thesis presents a method for achieving finer control over continuity between segments, which can be expanded to investigate co-articulation in the context of CSHMMs.

Acknowledgements

I would like to express my gratitude to my supervisor, Professor Martin Russell, for his invaluable guidance, unwavering support, and generous sharing of knowledge and expertise, which inspired this research. I consider myself extremely fortunate to have had such a patient and encouraging mentor who invested in me and my work. I would also like to extend a special thank you to Dr. Steve Houghton for his creativity in helping me conceptualise the models that formed the foundation of this research.

My sincere appreciation goes out to my colleagues in the Speech Group, including Dr. Peter Jancovic, Dr. Philip Weber, Dr. Linxue Bai, Dr. Eva Fringi, Dr Xizi Wei, and Dr Mengjie Qian. Their advice, support, and stimulating discussions over tea and cake helped shape the direction of this research.

I am deeply grateful to my grandparents and mother, Mr. Glyn Williams, Mrs. Brenda Williams, and Ms. Tracey Williams, for their unwavering love and support throughout the years. To my friends, I extend my heartfelt thanks to Nerupa Kidnapillai, Larry Godwin, and Elizabeth Johnstone, who have navigated the world of research with me providing endless encouragement. I am especially grateful to Larry for teaching me about posterior positivity!

Finally, I would like to thank my husband Vernon Caisley, you have run this long race with me with unfaltering patience, an abundance of kindness and consistent faith. Your support has seen me across the finish line - this is a win for us both.

Contents

List of Figures

List of Tables

List of Acronyms

ANN Artificial Neural Network

ASR Automatic Speech Recognition

BNF Bottle-Neck Features

CNN Convolutional Neural Network

CPLTM Continuous Piecewise Linear Trajectory Model

CSHMM Continuous State Hidden Markov Models

DFT Discrete Fourier Transform

DNN Deep Neural Network

DPLTM Dis-continuous Piecewise Linear Trajectory Model

GMM Gaussian Mixture Models

HMM Hidden Markov Models

IPA International Phonetic Alphabet

MAP Maximum A Posteriori

MFCC Mel-Frequency Cepstral Coefficient

MLP Multi Layer Perception

PCTM Piecewise Constant Trajectory Model

pdf Probability Density Function

PER Phone Error Rate

PTSHMM Probabilistic Trajectory Segmental Hidden Markov Model

RNN Recurrent Nueral Network

SHMM Segmental Hidden Markov Models

SC-PLTM Soft-Continuity Piecewise Linear Trajectory Model

WER Word Error Rate

Chapter 1

Introduction

1.1 Automatic Speech Recognition Today

In the past decade, impressive advancements have been made in the field of Automatic Speech Recognition (ASR) technologies. Within a relatively short time span, ASR models have transitioned from utilising Hidden Markov Models (HMM) (Rabiner, 1989) to employing more complex HMM-Artificial Neural Network (ANN) hybrid systems (Trentin and Gori, 2001). At present, state-of-the-art ASR solutions rely on Deep Neural Network (DNN) techniques. The application of neural networks (NN) to the ASR problem was initially explored in the 1960s and executed in the 1980s, however, the computational limitations of that era rendered a NN solution impractical for real-world scenarios.

In recent times, speech modelling tasks using NNs have been revitalised by the considerable advancements in computing hardware and processing power, as well as the availability of vast training corpora. Researchers have dedicated substantial efforts to investigating the practical implementation of DNNs to the speech recognition problem. As documented by (Hinton, 2012: Deng and Li, 2013), impressive enhancements in ASR have been reported early on in the decade. Industry-standard ASR systems conventionally employ DNNs with multiple non-linear hidden layers to model context-dependent states directly. These DNN-based recognisers have achieved phone error rates of 13.8%

on the widely used TIMIT dataset (Ravanelli et al., 2019), which serves as a benchmark in ASR research.

The emergence of NN-based methods for ASR has led to parallel research efforts focused on interpreting these models, which are often regarded as "black-box" systems. Such efforts include model introspection, where activations in different layers of a DNN are interpreted with respect to phonetic features (Nagamine et al., 2015). Additionally, the use of DNNs for feature dimensionality reduction has been studied. For instance, (Bai et al., 2018) examined the phonetic interpretation of a low-dimensional bottleneck feature representation and concluded that DNNs can learn phonetic structures in acoustic features that correlate to broad phone classes. While DNN systems improve ASR accuracy compared to HMM systems, they introduce model abstractions that rely on acoustically rich datasets for neural networks to learn from, which can be impractical for low-resource tasks.

Prior to the recent deep learning era, Hidden Markov Models and statistical methods dominated speech research efforts. The success of HMM-based systems can be attributed to their ASR performance and because they are mathematically well defined, making them computationally efficient to implement. However, HMM models have limitations, as outlined in the foundational paper for HMM-based ASR (Rabiner, 1989), and discussed in Section 3.2. Segmental HMMs (Holmes and Russell, 1999) and Continuous State HMMs (Champion and Houghton, 2016) were developed in response to the limitations of HMM-based speech recognition systems. Although Segmental HMMs are intuitively better models of speech, in practice, they fail to deliver significant performance improvements. Conversely, Continuous State HMMs are relatively understudied for ASR tasks.

In comparison to previous statistical HMM techniques, NN-based techniques used for modern ASR can be considered a data-centric solution to the recognition task. In contrast, the models proposed in this thesis can be regarded as knowledge-driven, offering an alternative solution to data-centrism.

1.2 Formulation of the ASR Problem

This section will describe the formulation of the ASR problem, highlighting the different components of an ASR system, and the associated modelling challenges. The primary objective of an ASR task is to obtain a sequence of discrete speech entities, usually words or phonemes, from a continuous time-varying speech signal. In this work, the discrete units being studied are phonemes. However, for the purposes of problem formulation, they will be assumed to be words. The main goal of the recognition task is to determine the most probable word sequence $W = \{w_1, w_2, \ldots, w_j\}$ given a sequence of speech features $Y = \{y_1, y_2, \ldots, y_T\}$ which can be expressed as a conditional probability $P(W|Y)$. Statistically, the objective is to identify the optimal word sequence W^* given all possible word sequences such that:

$$W^* = \underset{W}{\operatorname{argmax}} \; P(W|Y) \tag{1.1}$$

Bayes rule can be used to express this problem as a combination of probabilities:

$$P(W|Y) = \frac{P(Y|W)P(W)}{P(Y)} \tag{1.2}$$

The probability of the observed speech features $P(Y)$ is independent of the considered word and therefore, can be treated as constant. The prior probability of a word occurring, given a dictionary of candidate words, is denoted by $P(W)$ and can be estimated using a language model. The probability of a sequence of observed features given a word $P(Y|W)$ can be approximated from an acoustic model. For optimal performance, a good ASR system requires $P(Y|W)P(W)$ to be maximised, this implies that a good acoustic model score and a good language model score are both necessary for an optimal ASR system.

For a practical application of an ASR system, the main components defining the task can be broadly categorised into three areas:

Front-End Analysis: A crucial step in the ASR task that involves mapping an input speech signal to a feature vector representation. The output of the front-end analysis is a sequence of T, d-dimensional feature vectors denoted as $Y = y_1, y_2, \ldots, y_T$. The feature extraction technique used should produce a feature vector representation that complements the fundamental characteristics of the acoustic model. Chapter 4 provides an expanded description of the features utilised for experiments in this thesis. For a detailed discussion of common feature extraction techniques, refer to Section 2.2.

Acoustic Modelling: The acoustic model aims to output the likelihood that an observation sequence $Y = y_1, y_2, \ldots, y_T$ was generated by a sequence of trained models $\boldsymbol{\lambda} = \lambda_1, \lambda_2, \ldots, \lambda_k$. A model can be defined in various ways, but this study focuses on a family of models that can be described as Markovian statistical models. Chapter 3 provides the definitions of a number of Markovian models with details of the specific model descriptions.

Inference: The inference step in an ASR task involves searching over all model sequences arising from all possible word sequences to find the most probable sequence of models that could have generated the observed data. The goal is to infer the most likely sequence of words $W = w_1, w_2, \ldots, w_j$ given a set of acoustic model probabilities $P(Y|\lambda)$, lexical model probabilities $P(\lambda|W)$, language model probabilities $P(W)$, and a sequence of observed features $Y = y_1, y_2, \ldots, y_T$. This is achieved by maximising the joint probability of the model sequence, lexical sequence, and language model probability defined as:

$$W^* = \underset{W}{\operatorname{argmax}}; P(Y|\lambda); P(\lambda|W); P(W) \tag{1.3}$$

Here, $P(Y|\lambda)$ is the probability of the observed features given the sequence of models, $P(\lambda|W)$ is the lexical model probability which is unity unless multiple pronunciations are allowable, and $P(W)$ is the language model probability.

The focus of this thesis is to investigate the acoustic modelling component in ASR systems, with a specific focus on Markovian statistical acoustic models. A key contribution of this work is the development of a novel Segmental Continuous State HMM (CSHMM). This model is motivated by the continuous nature of speech production considering both articulation and acoustic continuity. In order to effectively implement and evaluate the proposed models, it is necessary to consider the following practical considerations:

1. **Training the models**: A model structure is defined by the set of parameters, during training, the model parameters are updated given labelled examples of speech data. In general, models with higher number or higher dimension parameters will require more extensive and linguistically diverse datasets during the training stage to optimise the model's performance.

2. **The decoding problem**: Assuming the model is defined by a state-space, the decoding stage aims to approximate the most probable sequence of states with which a sequence of observations are realised. A suitable decoding methodology must be employed to best approximate the underlying sequence of states.

3. **Evaluation/scoring**: A suitable evaluation metric is needed to benchmark and interpret the model's performance accurately. This metric characterises the probability that a particular sequence of speech units is produced by a model. Such a metric is necessary to compare different models and their respective performance.

1.3 Research Motivation and Contributions

The motivation of this thesis is to develop a speech model that more accurately reflects the dynamic nature of speech production. When humans speak, their speech articulators move along continuous trajectories while producing a continuous sound stream. This articulatory and acoustic continuity is a central focus of this research. The study of speech dynamics in this context was previously explored in (Deng, 2006) who states that:

> "A dynamic speech modelling approach can enable us to "put speech science back into speech recognition" instead of treating speech recognition as a generic, loosely constrained pattern recognition problem."

This quote directly aligns with the objectives of the current study, which seeks to demonstrate that a speech model that better represents the dynamic nature of speech production can enhance speech recognition. This thesis aims to investigate this hypothesis and demonstrate the feasibility of a Segmental CSHMM speech model in improving speech recognition performance.

A key contribution of this work is to extend current research of CSHMMs in the context of ASR. The hypothesis underlying this research is that a CSHMM framework offers greater flexibility for investigating different degrees of continuity in speech, compared to traditional HMM and statistical methods that have traditionally dominated speech research. A secondary advantage the proposed CSHMM approach is that it provides a parsimonious solution to acoustic modelling that is computationally efficient for practical applications. The proposed acoustic models in this work provide an alternative approach to data-driven methods that are common in contemporary ASR research. The intuition is that a parsimonious system will require less training data to estimate system parameters accurately, making it an attractive approach for low-resource tasks.

Research Contributions

The research contributions made in this thesis can be summarised as follows:

- The Formalisation of a trajectory-based Segmental CSHMM methodology for acoustic modelling, in Chapter 6.
- Implementation of four distinct trajectory-based Segmental CSHMMs which have been used to analyse the effects of varying continuity constraints across segment boundaries, (Seivwright, 2015).

- Presentation of benchmark results for a Segmental CSHMM system compared to a standard HMM system, Chapter 6.
- Proposal of a probabilistic measure of continuity between adjacent speech segment trajectory end-points, in Chapter 7.
- Presentation of a simplified model notation for HMMs, Segmental HMMs, Continuous State HMMs and Linear Dynamical Systems to remove any notation confusions that arise across literature sources in Chapter 3.

All recognition results in this thesis utilises the TIMIT speech corpus (Garofolo et al., 1993) with a low dimensional bottleneck feature (BNF) representation of acoustic data, which is explained in detail in Chapter 4.

1.4 Thesis Outline

This thesis is structured into eight chapters, the initial four chapters form the theoretical foundation of the study, delving into a family of Markovian statistical models that are relevant for ASR. The subsequent three chapters present the experimental methodology and outcomes of applying a Segmental CSHMM to an acoustic modelling task. Finally, the last chapter offers a summary of the research findings and outlines future directions for further development of this work.

The present introductory chapter outlines the motivation for this research and formally defines the task of ASR. In Chapter 2, a comprehensive literature review of speech production and articulation is presented. The chapter also includes a discussion of phonetic details of speech and speech feature representations that are typically compatible with Markovian statistical models, which are relevant to the development of acoustic models that are more faithful to the speech production process. The selection of an appropriate speech feature representation is a crucial aspect of the model design and is discussed in detail in Chapter 4. This chapter describes the Bottleneck features used in

this study to provide a comprehensive understanding of their relevance and suitability for the acoustic models defined in this work.

Chapter 3 builds upon the literature review by providing a tutorial-style analysis of Markov processes used for ASR. This chapter uses a simplified notation to present the model formulas for various models including HMMs, SHMMs, CSHMMs and Linear Dynamical Models (LDMs). A concluding summary highlights the similarities and shared assumptions among these models. Chapter 4 describes the derivation of the bottleneck speech features (BNFs) used in this study. An empirical analysis of the BNFs is presented, highlighting the characteristics of these features that align with the model assumptions. Chapter 5 outlines the speech corpora used in this research and provides preliminary details for all experiments in this study.

Chapter 6 presents the experimental results obtained from comparing a continuous state system with no continuity constraint between adjacent segments to a continuous state system with continuity enforced at segment endpoints. This chapter is a central part of this study, as it presents a detailed analysis of the novel trajectory-based Segmental CSHMMs, including the training and decoding algorithms used. Furthermore, the benchmark ASR performance of the Segmental CSHMMs is compared to conventional HMM-GMM system.

Chapter 7 extends the initial experimental results presented in Chapter 6 by introducing a soft continuity constraint framework. This chapter presents the experimental results for three different probabilistic soft-continuity methods and concludes with a discussion of the findings. Finally, Chapter 8 concludes the thesis by summarising the significant contributions of this research and proposing potential avenues for future research.

Chapter 2

Literature Review

Automatic speech recognition (ASR) is an applied technology within speech science, which includes the study of speech communication. This broad interdisciplinary field requires knowledge of anatomy, acoustics, signal processing, linguistics, physiology, and psychology. For more than seven decades, research on speech recognition has been a significant focus in human-machine communication and artificial intelligence. The development of ASR models has relied upon human speech and hearing physiology knowledge to improve models and feature representation.

This chapter provides a comprehensive review of relevant literature on the physiology of human speech production. This study focuses on the acoustic and articulatory details of speech production, including phonetic representations, and how such features can reflect the physiology of speech. Section 2.1 presents an overview of the physiological processes involved in human speech production, including the anatomy of the vocal tract and articulation as well as descriptions of the acoustic, articulatory, and phonetic properties of speech that are considered for ASR modelling tasks. Section 2.2 reviews commonly used speech feature representations that serve as input to an ASR model. Additionally, this chapter concludes with a summary of modern speech recognition applications, highlighting the models that have contributed to the advancement and improvement of ASR technology overall. This contextualises the work presented in this thesis and underscores the importance of research on the physiology of human

speech production in developing effective ASR models.

2.1 Speech Production

The human speech production process has been extensively studied across various disciplines, including linguistics, speech science, and communication technologies. Speech production has an acoustic and articulatory component, both of which are discussed in this section.

The primary organs used when producing a sound are the lungs, trachea, larynx, vocal folds, and the oral and nasal tracts, illustrated in Figure 2.1. During speech production, a stream of air is expelled from the lungs. Air travels up the trachea and passes through the larynx, where it is modulated by the vocal folds (Flanagan, 1979). This modulation occurs through the opening and closing of the small opening at the vocal folds known as the glottis in response to changes in air pressure. The positive air pressure from the lungs forces the glottis open momentarily; the increase in airflow when the vocal folds open results in a drop in pressure, consequently drawing the folds back into the closed position. The constant airstream from the lungs causes the vocal folds to open and close cyclically with a typical cycle period of 10ms for human speech which has been determined by empirical observations and physiological measurements (Fry, 1979).

2.1.1 Acoustic Properties

The vibrating of the vocal folds is the primary source of excitation for speech. The *type* of excitation can alter the periodicity of the speech signal; typically, speech signals are usually considered voiced or unvoiced. Voiced sounds are produced when vocal folds are close together and vibrating, whereas unvoiced sounds are produced when the vocal folds are apart and stationary. There are also other categories of excitation, such as "creaky voice", "breathy", and "whisper," each of which are produced by vocal folds vibrating in a particular region of the glottis.

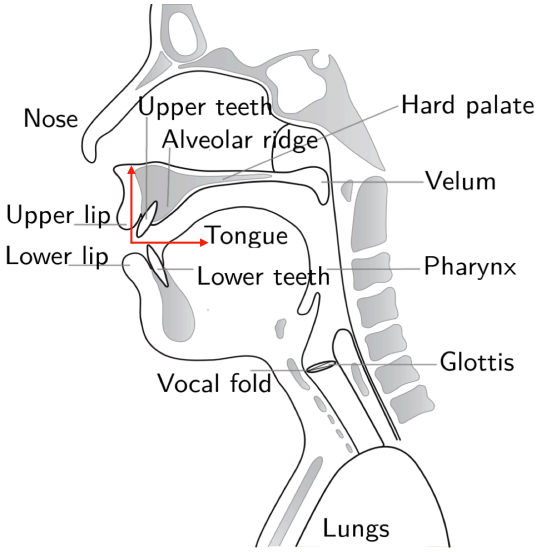


FIGURE 2.1: Anatomical diagram of human head with labelled speech articulators. - Diagram modified to include red directional arrows to illustrate two of the spacial directions with which articulators are loosely constrained to move along.

For voiced sounds, the frequency at which the vocal folds vibrate for a unit of time is known as the fundamental frequency (F0). The rate of vibration varies for males, females, and children due to biological variations in the length and thickness of the vocal tract. The configuration of the vocal tract during speech production relates to formant frequencies, which are concentrations of acoustic energy around specific frequencies in a speech wave. The vocal tract acts as a resonator, and different frequencies can be modulated by the articulators, forming vocal formants (Holmes and Holmes, 2001). Formants are well-formed in voiced regions of speech and have motivated research exploring formant measures as an acoustic feature representation for ASR.

There are four main acoustic properties of speech; amplitude, formants, frequency and time. A spectrogram plot can be used as a visual aid to interpret how the different frequencies that make up a waveform change over time. Figure 2.2 shows the speech waveform and the corresponding spectrogram for the utterance "He will allow a rare lie" generated using PRAAT software (Boersma and Weenink, 2001). The x-axis represents time and the y-axis represents the frequency in Hertz. The amplitude of the frequency is shown as a grey level, where black indicates the most energy, and light grey indicates

regions with the least energy. The spectrogram is a useful way of visualising the time, frequency and amplitude dimensions of speech. The particular utterance in Figure 2.2 has been selected as it is rich in voiced regions, meaning the vowels are well structured, consequently, the amplitude is easily identified from the dark bands that exist through most of the speech.

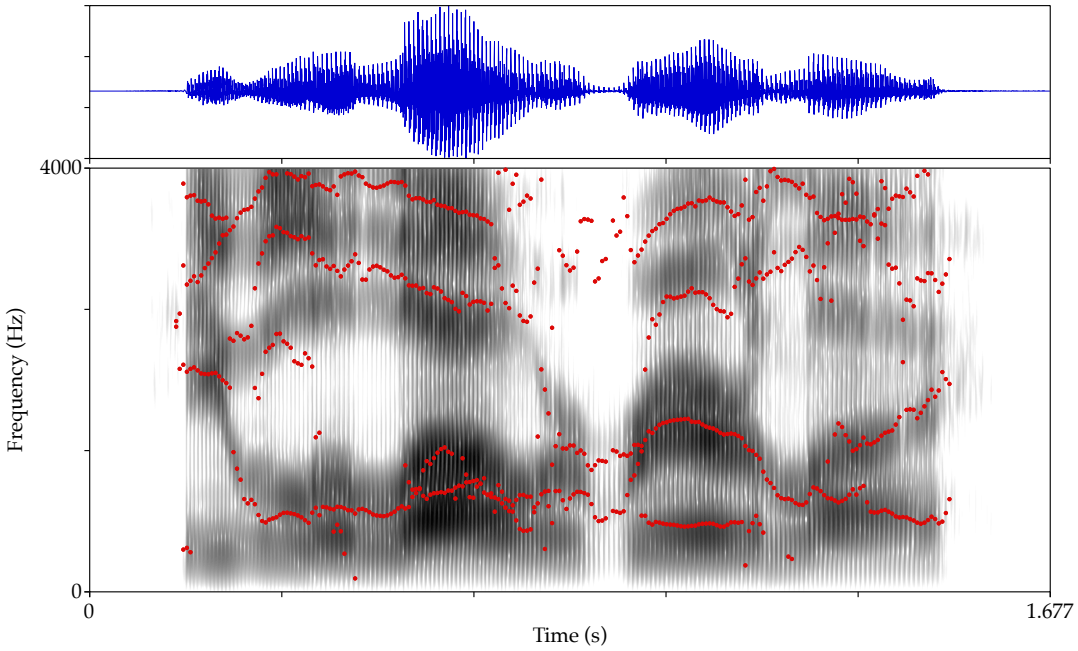


FIGURE 2.2: Plot of waveform and spectrogram, with annotated Formants (red dots) for TIMIT utterance SX11.wav - *"He will allow a rare lie"*.

2.1.2 Articulatory Properties

When humans speak, the position of the tongue, lips, teeth and other articulators move in a continuous motion along trajectories constrained by the jaw's physical structure and the human anatomy, as illustrated by the red arrows overlaid on Figure 2.1. These arrows serve to demonstrate that when humans speak, only a limited set of gestures are available to different articulators.For example, the upper and lower lips can move in an up/down motion along one axis, although they are not limited to only this movement, while the tongue can move in a forward, backward motion along a dimension perpendicular to the teeth.

To illustrate this further, consider the movement of articulators when saying the word "round". The lips move forward with a small opening, the teeth are almost closed, and the tongue moves to the back of the mouth cavity while producing the /r/ sound. This configuration is followed by the opening of the jaw, the lips and teeth move apart in an up-down motion creating a wide opening of the mouth, and the tongue moves forward towards the teeth during the /ou/ sound. When the /n/ sound is produced, the teeth close, and the tip of the tongue is pressed against the upper front region of the mouth called the alveolar ridge. Next, the velum is lowered to let the airstream pass through the nasal cavity. The final /d/ sound has a similar articulatory configuration as the /n/ where the tip of the tongue is pressed against the alveolar ridge. However, in the case of producing the /d/, the velum is raised to build up the air pressure behind the tongue which is then rapidly released.

This example highlights how articulators move smoothly from one configuration to the next in such a way that movements needed for adjacent sounds are often completed in transition. The vocal tract cannot change instantaneously from one configuration to another. The articulatory transition between sounds will influence the sound signal, a phenomenon known as co-articulation (Deng and Ma, 2000). Co-articulation occurs when the acoustic information relative to a particular sound spreads beyond the boundary of that sound. An explicit model of co-articulation, one which provides the flexibility to model systematic variation, could be beneficial to acoustic modelling for ASR (Frankel, 2003).

The application of a soft-continuity model presented in this thesis explicitly models the continuity at adjacent segment boundaries in the context of Continuous State Hidden Markov Models (CSHMMs). This could provide an exciting opportunity to extend the research on co-articulation and is discussed further in Chapter 7.

2.1.3 Phonetic Categories

This section aims to emphasise the fundamental impact of acoustic and articulatory properties in human speech communication and strengthen the hypothesis that an acoustic model, which is a more faithful model of speech production, can enhance speech recognition. A phone is a sound unit used in speech analysis to encode distinct physical or perceptual properties of speech. A description of different phonetic categories in the context of their acoustic and articulatory variability are presented in this section.

Speech sounds can be categorised into groups based on their excitation and articulatory configuration. As previously discussed, sounds can be divided into the categories of voiced and unvoiced excitation. Sounds can be further divided into phonetic categories, which are characterised according to whether they are voiced or unvoiced and relative to the place and manner of articulation. Two broad phone categories are vowels and consonants.

A vowel is a voiced sound realised by the vibrations of the vocal folds when a steady stream of air is expelled from the lungs. The study of the articulator positions when producing vowel sounds has interested linguists, phoneticians, and speech scientists for decades (Johns, 1975). The articulator position is a good descriptor for the type of vowel being produced. These vowel sounds can be mapped to a vowel space trapezium which is a diagram that relates the tongue's position to the realisation of the different vowel sounds. Figure 2.3 is an example of a vowel space diagram for the English language. The phonetic symbols used in this particular diagram are taken from the International Phonetic Alphabet (IPA) set (International, 1975). The absolute positions of phones could differ on different vowel space diagrams due to the variance in human vocal tracts. However, the general relative positions stay in the same pattern.

Consonants are made up of all non-vowel sounds and can also be sub-categorised according to their articulatory configuration. Consonants differ from vowels in that they are formed with an obstruction in the airflow and are grouped according to the

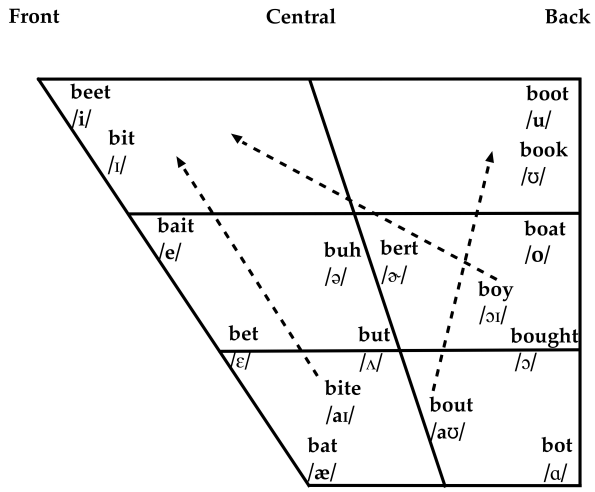


FIGURE 2.3: Vowel space trapezium for British English phonemes. - Diagram shows the relationship between vowel sounds and positions of the tongue during the production of such sounds, dashed arrows represent diphthongs. (Wells, 1982)

manner and place of the obstruction. In (Denes and Pinson, 1993), different categories of consonants are summarised into plosive, fricative, nasal, and approximant.

A plosive sound can be both voiced, for example [b, d, g] or unvoiced [p, t, k]. For this type of sound, air pressure builds up and is then rapidly released. The difference between the realisation of voiced and unvoiced plosive sounds is that the vocal folds vibrate in the case of voiced plosives. A fricative sound [v, z, sh] is produced when the articulators are positioned such that air is squeezed through a narrow channel causing turbulent airflow. Fricatives can be organised according to the articulator configuration, the different sub-groups are out of scope for the work in this thesis. However, a comprehensive description can be found in (Denes and Pinson, 1993). Nasal consonants [m, n, ng] are realised by blocking air from being released from the mouth, the velum is lowered, and air is released through the nose.

For this thesis, the revised ARPABET phonetic symbol set is used (Garofolo et al., 1993). Table 2.1 presents the phone categories used in this work, as defined in (Halberstadt and Glass, 1998). Further details on the phone representation in the speech corpus used for experimentation in this thesis can be found in Chapter 5.

Phone Category	Phone label
Plosive	/g/, /d/, /b/, /k/, /t/, /p/
Strong Fricative	/s/, /z/, /sh/, /zh/, /ch/, /jh/
Weak Fricative	/f/, /v/, /th/, /dh/, /hh/
Nasal/Flap	/m/, /n/, /en/, /ng/, /dx/
Semi-Vowel	/l/, /el/, /r/, /w/, /y/
Short- Vowel	/ih/, /ix/, /ae/, /ah/, /ax/, /eh/, /uh/, /aa/
Long Vowel	/iy/, /uw/, /ao/, /er/, /ey/, /ay/, /oy/, /aw/, /ow/
Silence	/sil/, /epi/, /q/, /vcl/, /cl/

TABLE 2.1: Phone categories with corresponding phone labels used for experimentation in this thesis (Halberstadt and Glass, 1998).

2.2 Speech Feature Representations

The application of an ASR system requires a feature vector representation of a speech signal as input for training and decoding. The feature analysis component of an ASR system plays a crucial role in the system's overall performance. There are several widely-used feature representations for speech recognition, some of which are described in this section. The feature extraction process in this context is fundamentally a signal processing problem where a speech signal is to be converted to a data product containing *useful* information pertaining to the different speech sounds being spoken. Researchers have long explored feature extraction methods to improve ASR, including methods encoding the signal directly and also measures of the articulator movement.

A motivation of this study is to define an acoustic model which is more faithful to the human speech production process. In order to achieve this, the feature representation used for experiments must be complementary to the assumptions of the proposed models. Therefore, an overview of feature representations commonly used for HMM-based models will be presented to provide the relevant context to the design decision to use Bottleneck features in this work.

Each of the presented feature representations has its advantages and disadvantages. The choice of which feature representation to use depends on the application and the assumptions of the acoustic model being used.

2.2.1 Formants

Formant features have been widely studied in the context of speech recognition research as they provide an effective representation of the vocal tract characteristics of speech sounds. Formants are defined as the peaks in the frequency spectrum of a speech signal and are caused by the resonances of the vocal tract. They are particularly useful for representing vowel sounds as the vocal tract configuration during the production of vowel sounds is closely related to the formant frequencies (Flanagan, 1979).

The utilisation of formant features in speech recognition has a long history, dating back to the seminal work of (Fant, 1970), who proposed using formant frequencies as a feature representation for speech recognition. Subsequent research has demonstrated the robustness of formant features in representing speech sounds, particularly in vowel sounds (Wells, 1967). However, it should be noted that while formant features are not as well-structured in unvoiced regions of speech due to the turbulent excitation at the source, they still exist, as the vocal tract still has resonance even when no speech is being produced. Furthermore, research has established a correlation between formant dynamics and the dynamics of speech articulators, as demonstrated by (Deng et al., 2006). For example, a lower fundamental frequency formant value corresponds to a closer position of the tongue to the roof of the mouth.

Formant features are typically calculated by applying a linear predictive coding (LPC) analysis to the speech signal (McCandless, 1974; Schafer and Rabiner, 1975). LPC analysis estimates the vocal tract transfer function, and formant frequencies can be obtained from the poles of the transfer function. The number of formants used in a speech recognition system can vary depending on the application, but utilising three or four formants is common. Despite the research on formant tracking and automatic formant extraction techniques proposed over the years (Deng et al., 2006), accurately extracting reliable formant features remains challenging, particularly in real-time or resource-constrained applications. In recent years, other feature representations have become more prevalent in ASR research, such as Mel-Frequency Cepstral Coefficients

(MFCCs) and Perceptual Linear Prediction (PLP) coefficients. Another reason for this shift is that formant features are sensitive to an individual speaker's vocal tract characteristics and are therefore less robust across different speakers and speaking styles. In contrast, MFCCs and PLP coefficients are designed to be more speaker-independent (Schafer and Rabiner, 1975).

While formant features are a powerful representation of the vocal tract characteristics of speech sounds, they are less robust across different speakers and speaking styles, more sensitive to noise, and more computationally intensive to extract, which have led to the use of other feature representations in ASR.

2.2.2 Mel Frequency Cepstral Coefficients

Mel-Frequency Cepstral Coefficients (MFCCs) have been a popular and extensively studied feature representation in ASR systems for several decades (Davis and Mermelstein, 1980). MFCCs have demonstrated consistent superior performance over other feature representations in ASR systems. The primary reason for this can be attributed to their ability to model the spectral energy distribution of speech in a perceptually meaningful way. The Mel-frequency scale, which is used to calculate the coefficients, is based on the human auditory system, providing a more natural representation of the spectral characteristics of speech (Stevens et al., 1937). Mel filter banks are an alternative widely used method for representing the spectral characteristics of speech in a perceptually meaningful way. The key difference between Mel filter banks and MFCCs is that a discrete cosine transform (DCT) is applied to the output of the Mel filter banks to remove any correlation between the coefficients, resulting in the final feature representation known as MFCCs.

Figure 2.4 outlines the key processes of extracting MFCCs from a waveform speech signal. The first step is pre-emphasis, which involves applying a high-pass filter to the speech signal to emphasise the higher frequency components of the signal. The speech signal is then divided into overlapping frames, typically using a 25ms window

with a 10ms overlap. Each frame is multiplied by a windowing function to reduce the effects of frame-to-frame discontinuities. The next step is to compute the Mel filter bank coefficients, which is achieved by transforming the frequency axis of the speech signal to a Mel-frequency scale, followed by a logarithmic compression to reduce the dynamic range of the signal. Finally, the DCT is applied to the logarithmically compressed Mel filter bank coefficients to remove the correlation between the coefficients.

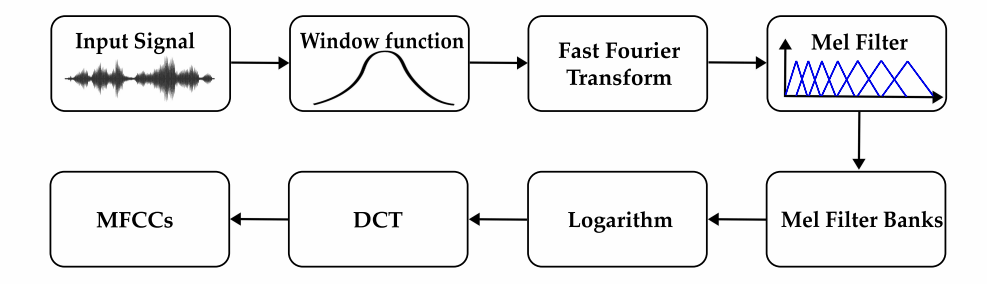


FIGURE 2.4: Mel-Frequency Cepstral Coefficient Extraction Pipeline.

Research has indicated that augmenting the MFCCs with delta-spectral cepstral coefficients (DSCCs) enhances the performance of ASR systems (Furui, 1986). The motivation for using DSCCs is the non-stationarity of the speech signal. Since speech signals are inherently non-stationary, i.e., their statistical properties change over time, researchers have proposed using Δ and $\Delta\Delta$ cepstral coefficients, which are the time derivatives of the static signal, to capture this non-stationarity. These coefficients capture the temporal dynamics of speech signals and provide additional information that is not captured by the static MFCCs.

The process of extracting DSCCs is similar to that of extracting MFCCs. The first step is to compute the static MFCCs, followed by computing the Δ and $\Delta\Delta$ coefficients. These coefficients are obtained by applying a finite difference technique to the static MFCCs. The resulting coefficients are then appended to the static MFCCs, resulting in a feature representation that captures both the spectral and temporal characteristics of speech signals.

In summary, MFCCs have been the dominant feature representation in ASR systems for many years. Their popularity can be attributed to their compatibility with various

acoustic models, particularly the Hidden Markov Model-Gaussian Mixture Model (HMM-GMM) acoustic models, which have been the gold standard for ASR systems until recently. The combination of MFCCs and HMM-GMM acoustic models has proven to be a robust approach for speech recognition, achieving state-of-the-art results in many ASR tasks. However, in recent years, deep neural network (DNN) systems have gained more prominence in ASR, and as a result, other feature representations have been explored. For instance, using Mel filter banks with a DNN system is more feasible than with HMM-GMM systems because DNNs are capable of learning how to use the correlations in the data implicitly. This allows for the use of other feature representations, such as Mel filter banks, which may be more suitable for DNN-based ASR systems.

2.2.3 Perceptual Linear Prediction Coefficients

Perceptual Linear Prediction (PLP) coefficients are another type of feature representation that are commonly used in ASR systems to represent the spectral characteristics of speech sounds. PLP coefficients are designed to capture the perceptual characteristics of speech by incorporating three concepts from psycho-acoustics: spectrum critical band spectral resolution, the equal loudness curve, and intensity power law (Hermansky, 1990).

The process of extracting PLP coefficients is similar to that of MFCCs in that the speech signal is divided into overlapping frames, multiplied by a windowing function to mitigate frame-to-frame distortions, and transformed into a logarithmic frequency scale that is more closely aligned with the human auditory system. However, the linear prediction coefficients for each frame are calculated using the Levinson-Durbin algorithm. Spectral smoothing is implemented using triangular filters to transform the coefficients to a logarithmic frequency scale. Finally, a DCT is applied to transform the logarithmic compression coefficients into the frequency domain, followed by a cepstral mean normalisation step to reduce the effects of variations in speaker and recording conditions. There are variations on how spectral smoothing is implemented during

the calculation of PLPs; the Hidden Markov Model Toolkit (HTK) (Young et al., 2002) employs triangular filters positioned along a Mel-warped scale.

MFCCs and PLP features are both commonly used feature extraction techniques for speech and audio signals, but they are based on different principles and are intended to capture different aspects of the sound. MFCCs are based on the Mel scale and are intended to capture the harmonic structure of the sound, while PLP features are based on linear prediction and are intended to capture the formant structure of the sound.

2.2.4 Articulatory Features

Articulatory data provides information about the movements of the articulators during speech production, it has long been considered a valuable feature representation for ASR systems. This is because it has the potential to overcome some of the limitations of traditional acoustic-based feature representations

Acoustic-based ASR systems rely on the characteristics of the speech signal, such as the frequency and amplitude, to recognise speech. However, these characteristics can be affected by various factors such as the speaker's accent, dialect, and environment, which can lead to low recognition rates. Articulatory data, on the other hand, provides information about the movements of the articulators, which are directly related to the physiological process of speech production. Using articulatory data can help to overcome the problem of co-articulation. Co-articulation is the phenomenon in which the articulatory movements for one speech sound are affected by the context of surrounding sounds. For example, the vowel sound in the word "bat" differs from the vowel sound in "bait". Acoustic-based ASR systems may have difficulty recognising these sounds because they are affected by the context of the surrounding sounds. However, using articulatory data can provide specific information about the movements of the articulators for each speech sound, which can improve the recognition of speech sounds in context (Johns, 1975). For decades, extensive research

has been conducted to investigate techniques to extract articulatory data for ASR tasks, some of which are discussed here.

Acoustic-to-Articulatory Inversion (AAI) involves estimating the movements of the vocal tract (articulatory information) from the speech signal (acoustic information) based on the assumption that the mechanical process of articulation is reflected in the speech signal. It is a challenging problem because the relationship between the articulatory and acoustic signals is highly nonlinear and non-unique (Neiberg et al., 2008; Toda et al., 2008). The goal of acoustic-to-articulatory inversion is to determine the precise location and configuration of the vocal tract articulators such as the lips, tongue, and jaw, that produce a given speech sound. Figure 2.3 illustrates a mapping of an articulatory configuration to an acoustic space relative to vowel speech units. However, the inverse task of estimating the articulatory configuration from the acoustic speech signal remains an unsolved problem. AAI is non-invasive and does not require specialised equipment, but it is still an active area of research. A comprehensive review of research in acoustic-to-articulatory inversion can be found in (Toutios and Margaritis, 2003). There are numerous potential applications of speech technologies that can benefit from solving this mapping problem. One such benefit is the ability to model co-articulation more accurately, in which the articulatory movements for one speech sound are affected by the context of surrounding sounds.

Electromagnetic Articulography (EMA) is a technique that uses electromagnetic sensors to track the movements of the articulators, such as the tongue, jaw, and lips. The sensors are attached to the articulators and generate signals that are used to track their movements in real time. The sensors are typically small coils that can be placed on or in the articulators; they generate electromagnetic fields that are used to track the position and movement of the articulators (Schönle et al., 1987). The EMA outputs the x and y coordinates of each of the coils, from which the position of the articulators is estimated. The MOCHA database (Wrench, 1999) combines the EMA output coordinates with acoustic data. EMA is considered a gold standard in articulatory data collection,

as it provides highly accurate and detailed information about the movements of the articulators. Similar to the EMA, another system that has been developed for measuring the movement of speech articulators is the X-ray microbeam system (Westbury et al., 1994). The X-Ray microbeam system is a technology that outputs the x and y coordinates of the speech articulators obtained by tracking the positions and movements of the speech articulators, specifically the tongue, lips, and other surfaces. Eight small pellets are placed on the articulatory surfaces measuring their position and velocity during speech production. These measures are observed in relation to the speech waveform and spectrogram. The X-ray microbeam system has yet to become a common technique for speech technology and is mostly used for research purposes. It is an invasive method to study speech production and can help to understand the relation between the speech signal and articulatory movement in the mouth. A third invasive system that is less common is the use of a Laryngograph to capture changes in the glottis by measuring variation in the current passed between electrodes placed on either side of a speaker's larynx (Holmes and Holmes, 2001). The EMA, X-Ray microbeam, and Laryngograph all suffer from the limitation of being invasive and expensive. In each case, sensors need to be attached to the articulators, and the data collection process requires specialised equipment and trained operators.

Alternatively, ultrasound imaging is a non-invasive imaging technology that does not require the attachment of sensors to the articulators (Shawker and Sonies, 1985). Instead, ultrasound imaging uses high-frequency sound waves to capture images of the inside of the mouth and track the movements of the articulators. The sound waves are emitted by an ultrasound transducer and are reflected by the different structures inside the mouth. The reflected sound waves are then captured by the transducer and used to create images of the inside of the mouth. The images captured by the ultrasound system can then be used to analyse information about the positions and movements of the vocal tract during speech, such as the shape of the vocal tract during different phoneme production and how it changes over time. However, this method is limited by the quality of the images, which are sensitive to factors such as the position of the

transducer (Cleland et al., 2013).

In summary, articulatory data provides information about the physical process of speech production, which has the potential to improve the robustness of ASR systems. However, there are difficulties associated with the use of articulatory data, including the specialised equipment and experimentation required for invasive methods such as EMA, as well as the limitations associated with acquiring such data in practical settings. Despite these challenges, previous research has explored using articulatory data in conjunction with or as an alternative to traditional acoustic-based feature representations. A useful review of these methods can be found in (Richmond, 2002).

2.2.5 Bottleneck Features

This thesis is motivated by the use of an alternative front-end acoustic parameterisation known as Bottleneck Features (BNFs). BNFs are a compact feature representation that can be used as input to an ASR system. They are derived from a DNN trained to model the speech signal, specifically by extracting the activations of a particular layer, known as the bottleneck layer, located in the middle of the network. In recent years, there has been a growing interest in interpreting the output of neural networks to explore alternative feature representations of speech. Previous studies have proposed methods that extract NN features from various intermediate hidden layers or the output layer as input to an ASR system, such as (Grezl et al., 2007) and (Deng and Chen, 2014) who considers NN features from the output layer as input to an ASR system. Similarly, (Sainath et al., 2012) proposed the extraction of BNFs from deep belief networks instead of multi-layer perceptron networks. Furthermore, (Bai et al., 2015) explored an intermediate layer of a NN, with a notable difference being that the intermediate layer was highly compressed with dimensions as low as three.

Literature has demonstrated the effectiveness of BNFs in ASR, as they capture a compact, high-level representation of the speech signal. For example, (Bai, 2018) showed

that a 9-dimensional BN feature representation provided a comparable phone recognition performance to 39-dimensional MFCCs. Additionally, (Tüske et al., 2013) has shown that BNFs can be used to improve the performance of ASR systems in combination with traditional feature representations such as MFCCs.

This work conducts experiments using BNFs proposed in (Bai et al., 2015). Chapter 4 provides a detailed description of these features and an observational analysis of the specific data used in these experiments. A vital characteristic of the BNFs presented in (Bai et al., 2015) is their compression; these features' dimensions are significantly smaller compared to other layers in the network. This constraint forces the system to project pertinent information into a reduced feature space. These particular features were also explored in (Weber et al., 2016b), which concludes that low dimensional BNFs broadly represent distinct phonetic properties of speech. Furthermore, (Bai et al., 2018) showed that when visualising these low-dimensional BNFs, distinct regions corresponding to different phonetic categories similar to phonetic vowel space diagrams (see Figure 2.3), are evident.

The investigation of an alternative front-end parameterisation that complements the proposed Segmental CSHMMs forms part of the contributions in this study. Chapter 4 details the low dimensional bottleneck features initially presented in (Bai et al., 2015) used in this work for experimentation. The use of BNFs in ASR presents an exciting opportunity for future research. It offers the potential for improved performance and a greater understanding of the underlying representations of speech in neural networks.

2.2.6 Direct Waveform

In recent years, DNN approaches have led to considerable advances in ASR (Hinton, 2012). With these new acoustic model architectures, there has been a reconsideration of appropriate speech features since DNN systems work differently than HMM-GMM systems. Machine learning methods have outperformed model-based methods in most areas of speech and language technology. This raises a research question on whether

parameterisations motivated by human knowledge are the best representations for ASR when using ML techniques.

Previous research has investigated using the raw waveform as a direct input to DNN systems for ASR. Work by (Graves and Jaitly, 2014) has examined the use of the raw waveform as input to a recurrent neural network (RNN) with no explicit phonetic representation. Speaker-independent systems using RNN approaches have achieved impressive results in ASR (Chorowski et al., 2015; Ravanelli et al., 2018), demonstrating that the raw waveform is a powerful feature representation for such systems when using neural methods. DNNs can learn appropriate representations of the speech signal directly from data, making the use of raw waveform advantageous. Raw waveform captures the temporal dynamics of the speech signal, which is important for recognising speech sounds as the temporal structure of speech signals is a crucial cue for speech recognition. In addition, raw waveform eliminates the need for feature engineering, which can be a time-consuming and challenging task.

Although using the speech waveform as a direct input to an ASR system is an active area of research, particularly in the context of DNN methods, the motivation of this study is to consider an improved model-based approach. The models presented in this thesis offer a different perspective on the ASR problem and explore Markov-based methods for solving it. While using raw waveform as a feature representation is relevant to current ASR trends, it is not the primary focus of this research.

2.3 Modern Applied ASR

In the last decade, there has been a significant shift in the field of ASR towards the use of artificial neural networks as opposed to traditional statistical model-based systems. The origins of NNs can be traced back to the 1940s (McCulloch and Pitts, 1943). The fundamental concept of a NN is based on the biological process of neurons passing information in the brain. This concept was later translated into a trainable model by (Rosenblatt, 1958), who introduced the term "perceptron" and defined an architecture

that included a collection of nodes, referred to as layers. As the field of NNs progressed, the term "Deep Neural Networks" (DNNs) was adopted for NNs with a minimum of two hidden layers and the multi-layer perceptron (MLP) was documented (Rumelhart et al., 1988). DNNs consist of multiple layers of interconnected nodes, which enable the model to extract increasingly complex features from the input data. As a result, DNNs have been shown to be highly effective in a wide range of tasks, including ASR outperforming traditional HMM-based systems. The use of DNNs in ASR has led to improvements in speech recognition performance, particularly in adverse conditions such as noise or when the speaker has a speech disorder.

Another factor contributing to the adoption of DNN methods can be attributed to the advancement in computer hardware and processing power, leading to a proliferation of research and applications in the field of DNNs. As a result, DNNs have surpassed traditional statistical techniques in a variety of fields. A comprehensive survey of DNN processing (Sze et al., 2017) notes that *"While DNNs offer state-of-the-art accuracy on a range of artificial intelligence tasks, they come with the trade-off of high computational complexity*. Nevertheless, complex artificial neural networks have become the standard technique in various tasks such as robotics, image processing, and speech recognition. One of the early modern HMM-Deep Belief Network systems (Mohamed et al., 2009) reported a 23% phone error rate[1] (PER) using the TIMIT corpus.

Since then, there has been a significant amount of research exploring different NN architectures, input feature representations, and feature interpretations. Table 2.2 presents a compilation of ASR results obtained using the TIMIT database over the last decade, with different neural network architectures. The table illustrates the range of recognition results that modern DNN systems are capable of achieving. It is important to note that the table is not meant to be used for direct comparison between different architectures, as the ASR systems may have been trained and tested under different conditions. However, it serves as a useful reference for the current state-of-the-art

[1] A measure of the accuracy of speech recognition systems, typically calculated as the proportion of incorrectly recognised phonemes to the total number of phonemes in a test set.

performance on the TIMIT dataset, and can provide insights into the design choices and trade-offs involved in ASR systems. An exploration of deep learning methods for discovering features in speech signals is provided in (Jaitly, 2014). The study cites the use of a conventional GMM-HMM statistical model, which achieved a phone error rate (PER) of 30.16% on the TIMIT database. This result serves loosely as a benchmark comparison for conventional statistical models in the field of automatic speech recognition. The results of the study demonstrate that DNN models are achieving superior performance in comparison to traditional statistical models.

TABLE 2.2: List of reported phone error rates (PER) on the TIMIT speech corpus in the last decade - Including details of model architecture and features used for experimentation, ordered by publication date.

Reference	NN Structure	Feature Representation	PER
Mohamed et al., 2009	Hybrid HMM - Deep Belief Networks (DBN) model which performs a discriminative fine-tuning phase using back-propagation.	MFCCs + energy + $\Delta, \Delta\Delta$	23.00%
Graves et al., 2013	Bidirectional Long Short Term Memory (Bi-LSTM) Recurrent Neural Network (RNN) with a pre-trained Transducer	Mel-Filter-Banks + energy + $\Delta, \Delta\Delta$	17.7%
Tóth, 2014	Convolutional Neural Network (CNN) with time and frequency domain convolutions with 7 filters. CNN trained with a drop out rate of 0.25	Mel-Filter-Banks + frame-level energy + $\Delta, \Delta\Delta$	16.7%
Chorowski et al., 2015	Attention-based Recurrent Sequence Generator (ARSG) incorporating convolutional attention features and a smooth focus	Mel-Filter-Banks + energy + $\Delta, \Delta\Delta$	17.6%
"Wavenet: A generative model for raw audio"	A deep CNN which uses dilated convolutions, allowing it to process a large context of audio samples at once, named WaveNet.	Raw audio	18.8%
Vanek et al., 2017	6-Layer DNN, with 1024 neurons in each layer and a DBN pre-training step.	fMLLR features	16.5%
Nayak et al., 2017	Hybrid DNN-HMM where the likelihoods obtained from the DNN are used as emission probabilities in a HMM	MFCC + IF cosine coefficients (IFCCs) - Smoothed	16.8%
Ravanelli et al., 2018	A Light Gated Recurrent Unit RNN (Li-GRU) with ReLU activations and batch normalisation limited to feed-forward connections only.	fMLLR features	14.9%
Ravanelli et al., 2019	PER obtained by combining multiple NNs and acoustic features: Multi-Layer Perceptron (MLP) + Li-GRU + MLP	MFCC + Filter-Banks + fMLLR	13.8%

2.4 Summary

This chapter reviews the acoustic, articulatory, and phonetic components of the speech production process. The acoustic properties pertain to the physical characteristics of speech sound— the articulatory properties relate to the movement of the speech organs during speech production. Additionally, phonetic categories are essential for understanding the relationship between sounds and their meanings. A comprehensive list of various speech feature representations is presented, with a focus on their suitability for use in ASR systems. The methods examined formants, MFCCs, PLPs, articulatory features, bottleneck features, and direct waveform input features.

The advancements of modern DNN-based ASR systems are introduced with a brief review of ASR experiments achieved over the last decade. The field of speech technology is continuously evolving, and many researchers believe that speech modelling will be the key to significant improvements in ASR performance. This viewpoint is supported by the work in (Roweis, 1999), where the authors suggest that the key to successful ASR is the development of an appropriate model where inference is sought from noisy observations. Such a model should have internal states that reflect the underlying speech production process.

The primary motivation for the research presented in this thesis is the development of a more faithful model of speech production, which is in contrast to the prevalent use of DNN methods in the field. A parsimonious model, which accurately reflects the underlying speech production process, is proposed to be better suited for a broader range of practical ASR applications with limited training data. Furthermore, statistical models provide robust insights and interpretability for the results, unlike the black-box nature of neural networks.

Chapter 3

Markov Processes for Acoustic Modelling

This chapter presents the theory of Markov Processes as a foundation for the development of a novel acoustic modelling approach known as the Continuous-State Segmental Model. This approach was implemented and applied to an Automatic Speech Recognition (ASR) task in this study (see Chapter 6).

Markov Processes have been a popular modelling solution across various domains, including information theory (Shannon, 1953), computer performance evaluation (Scherr, 1965), and web search applications (Brin and Page, 1998). One of the most successful applications of Markov models has been in acoustic modelling for speech recognition, with early works including (Fant, 1970). For over three decades, Markov models have dominated research in the field of ASR. However, a challenge of this extensive body of research and cross-disciplinary descriptions of these models is that the formal notation used can vary significantly across different publications.

This study examines four state-based models: the Hidden Markov Model (HMM) (Rabiner, 1989), Segmental Hidden Markov Models (SHMM) (Russell and Moore, 1985; Yu, 2010), including the probabilistic trajectory-based Segment model (pt-SHMM) (Holmes and Russell, 1999), the Continuous State Hidden Markov Model (CSHMM) (Ainsleigh, 2001; Champion and Houghton, 2016), and the Linear Dynamical Model

(LDM) (Frankel, 2003). The main objective of this work is to present a comprehensive and unified formalisation of these models, using consistent notation across models to integrate disparate bodies of work. This chapter uses a tutorial-style format to highlight the similarities among the different models, all of which are examples of stochastic Markovian models. In addition, the chapter clarifies the distinctions between modelling definitions and algorithmic implementations.

Sections 3.1 through 3.5 provide a detailed examination of each model, including a description of each component and the assumptions made when applying them to speech data. Section 3.6 outlines the training, decoding, and evaluation methods commonly used for these models. Finally, Section 3.7 summarises the key points and theoretical foundations presented throughout the chapter, highlighting the overlapping assumptions of the different models.

To the best of our knowledge, this is the first work to formalise these select models with a single consistent notation. This unified view of these models provides a solid foundation for understanding the intuition behind these methods and provides the theoretical context for the systems applied in this work.

3.1 Introduction to Markov Processes

A Markov process is a specific type of stochastic process that exhibits a characteristic called the Markov property. In probability theory, a stochastic process refers to a sequence of events where the occurrence of each event at any given stage depends on some probability. While the term "event" is commonly used to describe the outcome of an experiment, other interchangeable terms such as "random variable" or "state" may also be employed. For the sake of clarity and consistency, this work pertains to state-space models and thus, the term "state" will be used synonymously with "event".

The Markov property assumes that the probability of a current state depends only on the previous state. Let $S = \{s_1, s_2, ..., s_t\}$ be a sequence of states, where each state can take on a value from a state-space Q, such that $s_t \in Q \; \forall \; t$. The Markov property

implies that the probability of a state at time t having some value is dependent only on the value of the previous state at $t-1$, and not on the full history of the sequence of states. This is expressed mathematically as:

$$\text{Markov Assumption: } P(s_t|s_1, ..., s_{t-1}) = P(s_t|s_{t-1})$$

A state can be represented by various structures, including a vector, a number, a label, a discrete measure, or a probability vector in a given space of all possible outcomes. The state-space can be indexed by a discrete time scale $S = \{s_t, t = 1, 2...N\}$ where the states are countable, or on a continuous time scale $S = \{s_t, 1 \leq t < \infty\}$ in which case the states are not countable. The state-space itself can also be defined as continuous or discrete resulting in four categories of stochastic processes based on the nature of the time scale and state space (discrete or continuous). If the state space of a Markovian stochastic process is discrete and observable it is commonly referred to as a Markov chain, this is the simplest type of Markov model.

Definition 1 *A Markov Process is defined by the following properties:*

- *It has a finite number of states.*
- *The current state is only dependent on the outcome of the previous state. (memory-less)*
- *The state transition probabilities are constant over time.*

A Markov process can be parameterised as $\lambda = \{Q, S, \Pi, A\}$ where Q[1] , S, Π and A denote the state space, a particular sequence of state, the set of initial state probabilities, and a state transition matrix, respectively (Jurafsky and Martin, 2009).

Consider a discrete state space Markov process where it is possible to transition from any state to any other state between times t and $t+1$. This is called an ergodic Markov chain, as illustrated in Figure 3.1, which shows a two-state ergodic Markov chain. An arrow between two states means that the corresponding state transition

[1]Throughout this thesis, a simplified notation uses the index i refers to the i-th internal state from the set $Q = \{q_1...q_N\}$.

Name	Notation	Meaning/Property
State space	$Q = \{q_1 ... q_N\}$	Finite set of N internal states
Sequence of states	$S = \{s_1 \dots s_T\}$	Particular sequence of states
Initial probability	$\Pi = \pi_1, \pi_2, ... \pi_N$	Probability that a Markov process will start in state i, s.t. $\pi_i = P(s_1 = i)$
Transition probability matrix	$A = a_{11} ... a_{ij} ... a_{NN}$	Probability of moving from state i to state j, $a_{ij} = P(s_t = j \mid s_{t-1} = i)$, s.t. $\Sigma_{j=1}^{N} a_{ij} = 1$

probability is non-zero. The transition probability for an ergodic model is indexed as $a_{ij} \neq 0, 1 \leq i, j \leq M$ where a_{ij} is the probability of a transition between states i and j.

$$A = \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{bmatrix}$$

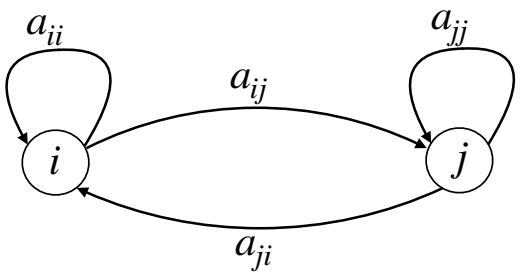


FIGURE 3.1: Diagram showing a two-state ergodic Markov chain with annotated transition probability indexes.

Ergodic Markov chains can be used when predicting a sequence of events where it is possible to transition from every state to every other state with positive probability.

Another type of discrete state-space Markov process is called a left-right Markov Chain. This type of model is structured such that $a_{ij} = 0$ whenever $i > j$ with the probability of staying in the final state equal to 1, as shown in Figure 3.2. The left-right Markov Chain model structure is particularly appropriate for modelling phenomena that evolves through an ordered sequence of states.

$$A = \begin{bmatrix} a_{ii} & a_{ij} \\ 0 & a_{jj} \end{bmatrix}$$

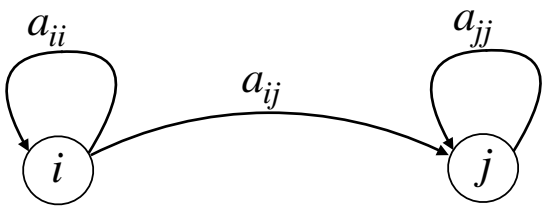


FIGURE 3.2: Diagram showing a two-state left-to-right Markov Chain with annotated transition probability indexes.

A standard ergodic or left-right Markov Chain assumes that each state is linked to a unique outcome, meaning that there is a clear correlation between physical observations and states. However, in many practical scenarios, it's not possible to directly observe a state or event. This means that the one-to-one relationship between a state and observation is not always accurate. The following sections expand on these basic assumptions and broaden the concept of Markov models to encompass cases where the observation is a probabilistic function of the state.

3.2 Hidden Markov Models

A Hidden Markov Model (HMM) is a type of Markov process that involves hidden underlying states that cannot be directly observed. HMMs have proven to be highly effective in pattern recognition tasks such as handwriting recognition and ASR.

In a HMM, the parameterisation takes the form of $\lambda = \{Q, S, \Pi, A, B\}$. This parameterisation extends the structure of a Markov process by introducing the additional parameter B. The sequence of observations $Y = y_1, y_2, ..., y_T$ is omitted as all models presented in this chapter assume that an input sequence of observations Y is given. The relationship between the unobserved states and observations is conditionally independent and is determined by a probability density function, which is often referred to as the emission probability.

The following components define a HMM:

Name	Notation	Meaning/Property
State space	$Q = \{q_1 ... q_N\}$	Finite set of N internal states
Sequence of states	$S = \{s_1 \dots s_T\}$	Particular sequence of states
Initial probability	$\Pi = \pi_1, \pi_2, ... \pi_N$	Probability that a Markov process will start in state i, s.t. $\pi_i = P(s_1 = i)$
Transition probability matrix	$A = a_{11} ... a_{ij} ... a_{NN}$	Probability of moving from state i to state j, $a_{ij} = P(s_t = j \vert s_{t-1} = i)$, s.t. $\Sigma_{j=1}^{N} a_{ij} = 1$
Emission probability	$B = b_i(y_t)$	Probability of observing y_t from an internal state i

In an ASR task, the aim is to identify a word sequence given a set of observations. In this case, the states represent the underlying acoustic properties of speech and the observations, while the observations are represented by speech feature vectors. In many real world systems it is not always possible to observe the underlying state. For example, when humans produce speech, it is very difficult to observe the internal human anatomy with which exists the acoustic environment that generates speech, however it is possible to observe the audio realisation of the underlying acoustic and articulatory configuration. There are some techniques which have considered more intrusive methods to capture or measure the speech articulatory process such as electromagnetic articulography (Schönle et al., 1987). However, ASR systems more often rely on acoustic feature representations. In this context, the simple model paradigm of a HMM has proven to be very successful when applied to acoustic modelling tasks.

The acoustic modelling component of the ASR foundation equation, as defined in Equation 1.3, can be represented as the probability of the observations given a model, $P(Y|\lambda)$[2]. This probability can be calculated by summing over all possible state sequences, S.

$$P(Y|\lambda) = \sum_{S} P(Y, S|\lambda) \tag{3.1}$$

[2]All probabilities are assumed to be dependent on a particular model parameterisation λ and henceforth λ will be excluded from probability definitions after Equation 3.3.

This joint probability can be factored such that,

$$P(Y, S|\lambda) = P(Y|S, \lambda)P(S|\lambda) \tag{3.2}$$

So,

$$P(Y|\lambda) = \sum_{S} P(Y|S, \lambda)P(S|\lambda). \tag{3.3}$$

Two underlying assumptions of a HMM enable the factorisation of probabilities in Equation 3.3. The first assumption is the observation independence, which specifies that an observation y_t depends only on the state that produced the observation and is represented by the emission probability. This observation independence implies a one-to-one mapping between a state and observation vector.

$$\begin{aligned} P(Y|S) &= P(y_1, \ldots, y_t|s_1, \ldots, s_t) \\ &= P(y_1|s_1) \ldots P(y_t|s_t) \\ &= b_{s_1}(y_1) \cdot b_{s_2}(y_2) \ldots b_{s_t}(y_t) \\ &= \prod_{t=1}^{T} b_{s_t}(y_t) \end{aligned} \tag{3.4}$$

The second model assumption is related to the Markov property, which specifies that a state at time t only depends on the previous value for a state s_{t-1}. This allows the probability of a sequence of states, given a model, to be factorised as follows:

$$
\begin{aligned}
P(S) &= P(s_1, \ldots, s_i, \ldots, s_t) \\
&= P(s_t|s_1, \ldots, s_{t-1}) \cdot P(s_{t-1}|s_1, \ldots, s_{t-2}) \ldots P(s_2|s_1) \cdot P(s_1) \\
&= P(s_t|s_{t-1}) \cdot P(s_{t-1}|s_{t-2}) \ldots P(s_2|s_1) \cdot P(s_1) \\
&= \pi_{s_1} . a_{s_1 s_2} \ldots a_{s_{t-1} s_t} \\
&= \pi_{s_1} \prod_{t=2}^{T} a_{s_{t-1} s_t}
\end{aligned} \tag{3.5}
$$

Therefore, the joint probability of a sequence of observation features and a particular hidden state sequence can be calculated using the transition probabilities and emission probabilities, assuming the observation independence and the Markov assumption. Specifically, the joint probability distribution for a HMM can be calculated as follows:

$$
P(Y, S) = P(y_1|s_1)P(s_1) \prod_{t=2}^{T} P(y_t|s_t)P(s_t|s_{t-1}) \tag{3.6}
$$

Calculating the probability of an observation sequence involves summing over all possible hidden state sequences:

$$
P(Y) = \sum_{S} \pi_{s_1} b_{s_1}(y_1) \prod_{t=2}^{T} b_{s_t}(y_t)\, a_{s_{t-1} s_t} \tag{3.7}
$$

An important distinction needs to be made between a particular state index at a time t and the underlying symbol that it represents. Recall that a state-space Q defines a finite set of all possible internal states, the state subscript s_t in the emission and transition probabilities in Equations 3.4, 3.5 and 3.7 refers to the state index and emits the particular internal state symbol. In this thesis, the following shorthand notations are used in different contexts but all represent the same concept: $b_{s_t=i}(y_t) \equiv b_{s_t}(y_t) \equiv b_i(y_t)$ and $a_{s_{t-1}=i\ s_t=j} \equiv a_{s_{t-1} s_t} \equiv a_{ij}$.

Figure 3.3 illustrates a conventional HMM architecture based on the model parameterisation λ. The diagram shows three connected hidden states, along with a specific state sequence $\{s_1, s_2, s_3\}$ and three observations $\{y_1, y_2, y_3\}$. The annotated transition and emission probabilities are also displayed.

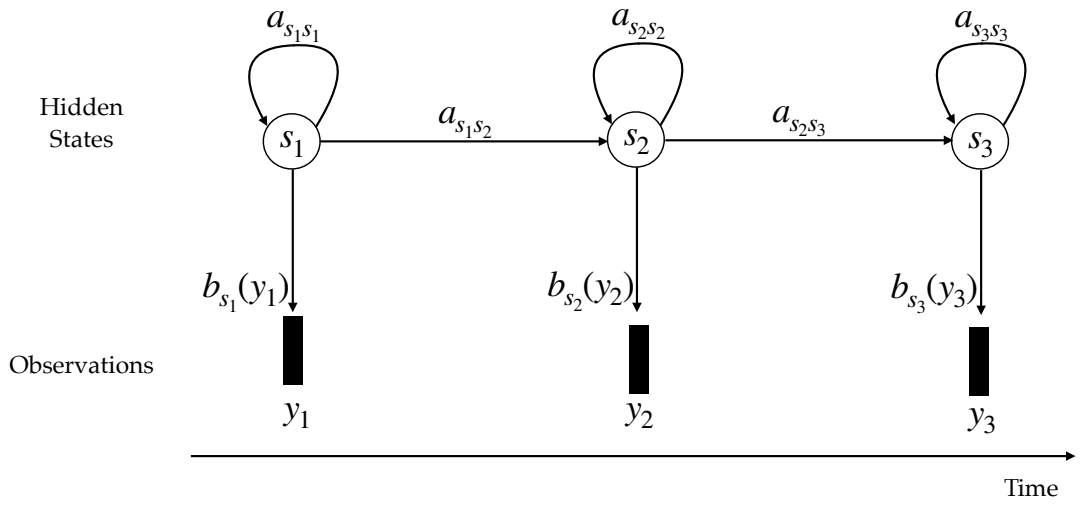


FIGURE 3.3: Conceptual diagram of a three state HMM structure with annotated transition and emission probabilities.

The corresponding transition matrix and initial state vector for this Markov chain example in Figure 3.3 would be:

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \qquad \pi = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

3.2.1 HMM Limitations

The HMM has been successful in various applications, including speech recognition. However, there are some well-known limitations associated with the model assumptions when used for ASR. A notable tutorial (Rabiner, 1989) attributes these limitations to the observation independence assumption, the Markov state-dependency and an implicit geometric duration modelling in a standard model, each of which will be

discussed here.

The Independence Assumption

The assumption that successive observations are independent of each other and depend only on the state is a weakness of this model when applied to speech. In reality, the acoustic properties of a speech signal are highly correlated. Typically, a speech feature vector is encoded at a frames rate of 10ms in standard systems. When sound is produced it can be observed from a inspecting a spectrogram that strong dependencies can occur for durations of 50–100ms. To address correlations between observation vectors, one approach is to incorporate dynamic coefficients such as delta and acceleration coefficients. These coefficients can be appended to the input feature vector, as proposed in (Furui, 1986). Alternatively a more complex modelling paradigm can be considered by extending the HMM to model segments of features. A detailed summary of segmental modelling techniques can be found in (Ostendorf et al., 1995).

Piecewise Stationarity The Markov property itself upholds a strong assumptions that a state is dependent only on the immediate prior state. This assumes that speech is produced by a piece-wise stationary process, with instantaneous transitions between stationary states governed by the transition probability. This is rarely the case for speech sounds where a speech signal is produced by a continuously moving physical system such as the vocal tract (Holmes and Huckvale, 1994). When considering the articulatory movement required to produce speech, the human anatomy has physical limitations which contradicts the hypothesis that articulators can move instantaneously from one configuration to another.

State Duration Distribution

The probability of a model staying in the same state for several frames is determined only by the self-loop transition probability. This implicit state duration model conforms to a geometric distribution due to the Markovian assumption. The notation $p_i(d)$ is used to denote the probability distribution of occupying a state i for exactly d time

units which can be defined as:

$$p_i(d) = a_{ii}^{d-1}(1 - a_{ii})$$

Therefore, the model assigns the highest probability of staying in a state for a duration of 1, with successively smaller probabilities assigned to longer durations, leading to a bias towards shorter durations. This is an inadequate timing model for most speech sounds, to address this, there is research on explicitly modelling the duration of a state such as (Russell and Moore, 1985). The next section on hidden semi-Markov models explains this in further detail.

3.3 Hidden Semi Markov Models

A hidden semi-Markov model (HSMM) is a probabilistic model that extends the HMM to allow for variable duration states. The HSMM was first proposed by (Ferguson, 1980), whose work formalised an explicit state duration model. Since then, HSMMs have been successfully applied to a number of scientific and engineering areas, such as speech recognition and synthesis (Gales and Young, 1993), human activity recognition and prediction (Natarajan and Nevatia, 2007), handwriting recognition (Senior et al., 1996), functional MRI brain mapping (Faisan et al., 2002) and network anomaly detection (Tan and Xi, 2008). Across these different domains, different naming conventions are adopted: Semi-Markov models, generalised Markov models, explicit or variable duration HMM, segmental HMM and segment model. In speech literature, it is common to refer to these models as segmental HMMs (SHMM), and so this naming convention and abbreviation will be used in this work analogous to HSMM.

A SHMM introduces the notion of a segment in a Markov chain by explicitly specifying a state duration distribution D. This extension of the standard HMM framework allows a one-to-many mapping between a state and observation. The number of observations produced while in state is determined by the duration d spent in that particular state (Yu, 2010). This is a fundamental difference from a standard HMM. The observations

produced in a segment are still considered independent given the state that generated them, but this independence assumption is relaxed by specifying that observations are conditionally dependent on the segment properties. This subtle addition of conditional dependence is a better intuitive model of speech, certain attributes of speech such as speaker characteristics, can now be fixed over a whole segment and thus can be factored in by an updated emission probability (Ostendorf et al., 1995).

The parameters for a SHMM are $\lambda = \{Q, S, \Pi, A, B, D\}$:

Name	Notation	Meaning/Property
State space	$Q = \{q_1 ... q_N\}$	Finite set of N internal states
Sequence of states	$S = \{s_1 \dots s_T\}$	Particular sequence of states
Initial probability	$\Pi = \pi_1, \pi_2, ... \pi_N$	Probability that a Markov process will start in state i, s.t. $\pi_i = P(s_1 = i)$
Transition probability matrix	$A = a_{11} ... a_{ij} ... a_{NN}$	Probability of moving from state i to state j, $a_{ij} = P(s_t = j \mid s_{t-1} = i)$, s.t. $\Sigma_{j=1}^{N} a_{ij} = 1$
Emission probability	$B = b_{i,d}(y_t^d)$	Probability of observing a segment y_t^d from an internal state i, s.t. $P(y_t^d \mid s_t = i, d)$ where the length of the sequence is given as d
Duration distribution	$\mathcal{D} = d_1, \dots, d_N$	Probability distribution representing a particular duration of a state i as d_i

Recall that an acoustic model estimates the probability that a sequence of observations is generated by a particular model $P(Y|\lambda)$. For a SHMM, a sequence of observations is represented as $y_t^{d_i} = y_t, \dots, y_{t+d_i-1}$ which corresponds to a segment of length $d \in \mathcal{D}$ where d_i is a random variable and the segment is assumed to be emitted from a state i. Additional notation conventions are used here to describe the more complex SHMM. Specifically, the subscript of a state indicates the time index while the superscript represents the state value such that s_t^i denotes a state at a time t with a value i. This notation is shorthand for the probability $P(s_t = i)$. Given this notation, the probability of an observation sequence given a state can be defined:

$$P(y_t^{d_i}, d_i | s^i) = P(y_t^{d_i} | s^i, d_i).P(d_i | s^i), \tag{3.8}$$

where $P(d|s^i)$ is the duration distribution which specifies the likelihood that a particular state i has a duration d, corresponding to a segment duration. At the segment level, the probability of a sequence of observations given a state and a duration $P(y_t, \ldots, y_{t+d-1}|s^i, d)$ is defined by the emission probability $b_{i,d_i}(y_t^{d_i})$. Equation 3.8 can be written as:

$$P(y_t^{d_i}, d_i|s^i) = b_{i,d_i}(y_t^{d_i}).P(d_i|s^i) \tag{3.9}$$

Figure 3.4 shows the general structure of a SHMM adapted from (Yu, 2010). The initial state and duration are selected according to the initial probability distribution. The length of an observation sequence for each state is determined by the duration variable of each state. For example, in figure 3.4, the first state produces three observations hence the duration equals three and then transitions to the second state. The second state then produces an observation sequence length of five. This can be extended for all remaining states up to a time t.

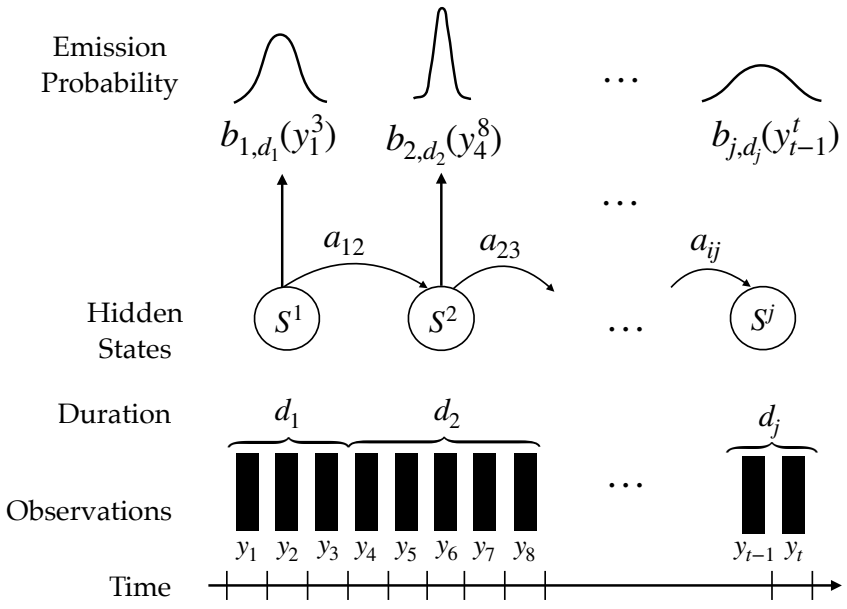


FIGURE 3.4: SHMM structure showing the mapping between observation and hidden states, including their corresponding durations. Additionally, transition and emission probability notation annotated for conceptual understanding.

The generative nature of this model depicted in Figure 3.4 specifies that on entering a state i at a time t, a duration d_i is drawn according to the probability $P(d|s^i)$ which results in d output observations which are generated according to the emission probability $b_{i,d_i}()$. At a time $t + d_i - 1$ a state transitions from a state i to a state j according to the transition probability a_{ij}. Therefore, the overall segmentation of the sequence of observation features is determined by the sequence of segment lengths. Given a state sequence, it is necessary to sum over all possible segment durations to calculate the probability of the observation sequence.

The SHMM assumes that observations are independent given a segment length, which can be expressed as:

$$P(y_t^d|s^i, d) = \prod_{t=1}^{T} b_{i,d}(y_t^d) \tag{3.10}$$

The recognition task for a SHMM is equivalent to that of a HMM, as defined in Equation 3.3. The primary aim is to determine a model that best accounts for the observations. For SHMMs, the joint probability of a sequence of observations and states is formulated as:

$$P(Y, S) = P(y_1^{d_1}|s_1, d_1)P(d_1|s_1)P(s_1)\prod_{t=2}^{T} P(y_{d_{t-1}+1}^{d_t}|s_t, d)\ P(d|s_t) \tag{3.11}$$

To compute the total probability of an observation sequence using the SHMM framework, it is necessary to explore all possible state durations for all sequences of states. As a result, the process may be defined as follows:

$$P(Y) = \sum_{S}\sum_{D} P(Y, S) \tag{3.12}$$

A SHMM is a generalised Markov model. Consider the case where observations within a segment are assumed to be independent and the explicit duration model is assumed to be geometric. The emission probability $b_{i,d}(y_t^d)$ simply becomes the product of the probabilities of each individual observation. Therefore, the segmental model

reduces to a one-state HMM. However, in practice, it is of no benefit to under-utilise a SHMM framework which has the flexibility to provide an explicit duration model. A better duration model can capture temporal correlations between observations within a segment. Additionally, the SHMM framework provides the opportunity to model the output density according to an underlying trajectory, which will be discussed in the following section.

3.3.1 Trajectory Modelling with SHMMs

A SHMM can be thought of as an extension of a traditional HMM, in which a state is associated with a sequence of observations within a segment rather than a single vector. This formulation of the SHMM includes an explicit probability component for segment duration. However, the SHMM framework is general enough that it is possible to extend the model description further by introducing a concept of a trajectory to capture two different types of variability; intra-segmental variation, which captures the variation of observations around a particular underlying trajectory, and extra-segmental variation which models the variation in the underlying trajectory which describe a segment. In early research, these models were described according to the nature of the trajectory such as "static", "linear" or "polynomial". In (Holmes and Russell, 1999), the term "Probabilistic-Trajectory Segmental HMMs" (PT-SHMMs) was introduced to describe these models. The models discussed in this presentation are examples of PT-SHMMs.

In a SHMM, the emission probability b_i represents the probability of observing a sequence of features y_t^d when in state s_i. In this framework, each state is associated with a segment, and a segment is described by its trajectory. As the state is hidden, so too is the trajectory. This notation follows the convention presented in (Holmes, 1997). Let $F = \{f_a\}, a \in \mathcal{A}$ represent a set of trajectories that define a segment. To calculate the emission probability for a sequence of features given a model requires integrating over all possible trajectories such that:

$$P(Y) = b_{i,d}(y_t^d) = \int_{a \in \mathcal{A}} P(y_t^d, f_a) \tag{3.13}$$

A comprehensive study of a segment models with an explicit trajectory parameterisation is documented in (Gales and Young, 1993) which derived the total probability calculations of a segmental model explicitly. Alternatively, a particular segmental model proposed by (Russell, 1993) considers an "optimal trajectory" approximation by utilising a MAP optimisation method which can be defined:

$$P(Y) = b_{i,d}(y_t^d) = \max_{a \in \mathcal{A}} P(y_t^d, f_a) \tag{3.14}$$

With an optimal trajectory defined by:

$$a^*(y_t^d) = \operatorname*{argmax}_{a \in \mathcal{A}} P(y_t^d, f_a) \tag{3.15}$$

The joint probability of observations and a trajectory f_a can be specified by the following equation:

$$P(y, f_a) = P(y|f_a).P(a) \tag{3.16}$$

The extra-segmental variation, described by $P(a)$, captures differences in speakers or pronunciations of a speech sound, which can lead to distinct trajectories for the same speech segment. On the other hand, the intra-segmental variation $P(y|f_a)$ accounts for noisy observations around a given trajectory, thereby capturing correlations on a frame-by-frame basis. The conditional probability of observations y given a trajectory f_a can be calculated by taking the product of individual observations at a time t given the function value at t.

$$P(y|f_a) = \prod_{t=1}^{T} P(y_t|f_a(t)) \tag{3.17}$$

It is worth noting that there has been significant research on various trajectory-based

segmental models, including constant trajectory models (Russell, 1993), linear dynamical trajectory models (Digalakis, 1992), polynomial regression (Deng et al., 1994), various probabilistic trajectories (Holmes, 1997), multiple-level segmental models (Jackson and Russell, 2002), hidden dynamical SHMM (Richards and Bridle, 1999) and others. These works share the common goal of examining the modelling constraints of a speech in order to more accurately capture the acoustic and articulatory continuity of the speech process, which aligns with the focus of the present thesis. The following sections briefly summarise the trajectory constraints that are most pertinent to the present work.

Constant Trajectory Model

A constant trajectory model is the simplest of the trajectory-based SHMMs, also known as a static SHMM (Russell, 1993), or in the case where both the extra and intra-segmental probabilities are assumed to be Gaussian, the naming convention Gaussian Segmental HMM (GS-HMM) is also used. The underlying model trajectory is assumed to be constant over time meaning the observations within a segment are dependent only on the segment mean which is a single target value. To formalise this, consider a single state in a constant trajectory SHMM, the extra-segmental variation can be defined by a Gaussian pdf $\mathcal{N}(c; \mu, \gamma)$ where c is a canonical underlying trajectory. The observations are assumed to be distributed around the target trajectory according to a Gaussian pdf with some variance τ such that the intra-segmental variation can be defined as $\mathcal{N}(y; c, \tau)$, (Gales and Young, 1993).

Therefore, the extra and intra-segmental Gaussian variation can be substituted into Equation 3.13 such that:

$$P(Y) = \int_{-\infty}^{\infty} \mathcal{N}(c; \mu, \gamma) \prod_{t=0}^{T} \mathcal{N}(y_t; c, \tau)\ dc. \tag{3.18}$$

This constant trajectory model can be used as a baseline system when comparing other acoustic models focused on modelling the dynamics of a system with a trajectory

assumption. This specific model is useful when studying long-term variations of a signal and the effect of modelling the as opposed to modelling the extra and intra-segmental variance separately (Holmes and Russell, 1995).

Linear Trajectory Model

A linear SHMM extends the constant trajectory model by assuming the underlying trajectory changes linearly over time leading to the parametrisation of the underlying trajectory on a target, c, and slope parameter, m. This model has been described in detail in (Holmes, 1997). Typically, the formulation of the linear SHMM is expressed in terms of the slope and midpoint of the trajectory, such that $f(y_t; m, c) = c + m(y_t - y_{\frac{(t+d)}{2}})$. Given a sequence of observations, $Y = y_1, y_2, \ldots, y_T$, the model formulation for the linear SHMM can be obtained by substituting the general form of the joint probability, as described in Equation 3.16, into Equation 3.13. This results in the following formulation:

$$P(Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathcal{N}(m; \mu, \gamma)\mathcal{N}(c; \nu, \eta) \prod_{t=0}^{T} \mathcal{N}(y_t; f_{m,c}, \tau)\, dc\, dm. \tag{3.19}$$

There are at least two different configurations of a trajectory that can be used to define a linear SHMM. The first configuration is known as a constrained slope trajectory, where the mean of the slope is fixed to zero, and the variance of the slope is set to an arbitrarily small value, effectively limiting the variability of the slope parameter. The second configuration is referred to as a flexible slope (Holmes and Russell, 1999), where both the mean and variance of the slope parameters can be estimated from the training data.

There are several studies on SHMMs that explore various configurations of PT-SHMMs such as (Gales and Young, 1993; Holmes, 1997; Holmes and Russell, 1999), these include the constant and linear trajectory models with both fixed and flexible trajectory constraints, using optimal and trajectory-independent architectures. A closed-form training and inference solution is presented in each of these works. Experimental results

using the TIMIT dataset (Holmes and Russell, 1999) demonstrate that the linear PT-SHMM with constrained slope performed the best in a recognition task, outperforming a standard HMM system. Additionally, a phonetic classification experiment found that the linear PT-SHMM had improved performance, particularly for dynamically varying sounds such as diphthongs, semivowels, and glides. Overall, this research suggests that PT-SHMMs provide a promising framework for speech acoustic modelling, with the potential to outperform or perform comparably to standard HMM models in ASR and classification tasks.

3.3.2 Limitations

Despite the modelling framework of SHMMs offering more flexibility to better model speech, there are a number of known disadvantages in the segment models. Firstly, the increased number of free parameters in the models, such as the number of segments and the distribution parameters of each segment, require more training data to adequately estimate the model parameters. Secondly, there is still the issue that segments in a SHMM are assumed to be independent, while this independence assumption is greatly improved from the case of a standard HMM, it is still an oversimplification of the speech process, particularly in the context of co-articulation. Thirdly, most of the segment conditional output densities considered only represent unimodal distributions although the observation vector distributions are often more complex due to speaker and environment variability. This can be addressed by introducing more complex output densities such as mixture distributions, but this also increases the number of free parameters in the model. Therefore, despite the advantages of SHMMs, their increased flexibility comes at a cost of increased complexity and computational requirements. Reported ASR results for SHMMs do not provide the significant performance gains which would warrant the increased computation (see Table 3.2).

3.4 Linear Dynamical Models

The Linear Dynamical Model (LDM) is a type of state-space model that is commonly used for predicting the behaviour of time-evolving systems. These models are widely applied in various fields, including target tracking and navigation systems, image processing, traffic control (Delahaye and Puechmorel, 2010), and speech parameter estimation (Frankel, 2003). The LDM belongs to a family of linear Gaussian models that are defined by two equations:

$$s_{t+1} = f(s_1, s_2, ..., s_t, \eta_t) \tag{3.20a}$$

$$y_t = g(s_t, \epsilon_t) \tag{3.20b}$$

Equation 3.20a describes how the state evolves over time. This is determined by a function $f(\cdot)$ and a measure of uncertainty η. Equation 3.20b describes how the observations are generated based on the current state. This is determined by a function $g(\cdot)$ and an observation error ϵ. The functions $f(\cdot)$ and $g(\cdot)$ can be either linear or non-linear, in the context of the LDMs discussed in this work, they are strictly linear functions with Gaussian additive noise. Specifically, η is characterised by a Gaussian distribution, $\eta \sim \mathcal{N}(\mu_t^{(s)}, \Sigma_t^{(s)})$, while ϵ is characterised by $\epsilon \sim \mathcal{N}(\mu_t^{(y)}, \Sigma_t^{(y)})$.

The LDM is a generative model that assumes a continuous hidden state vector s_t evolves according to first-order Markovian dynamics. As such, the current state s_t only depends on the previous state s_{t-1}, making the model structure inherently auto-regressive. In the context of ASR, the hidden state vector can be assumed to capture details of the speech process such as the articulator positions and the dynamic way in which they move from one configuration to another. Although the assumption of linearity in the LDM oversimplifies the inherently non-linear articulatory process (Blackburn, 1997), it nonetheless represents an improvement over the discrete state-space structure of a HMM.

Similarly to the SHMM presented previously, the LDM has the advantage of being able to model intra-segmental dynamics separately from extra-segmental variations. Moreover, observations generated during the evolution of the state process are conditionally dependent on the underlying process until the state is reset. This property makes the LDM an effective tool in the field of acoustic modelling due to its ability to accurately capture the implicit dynamics of speech production.

The state space, also known as the latent space, is defined by a transformation function that incorporates rotations and stretches applied to the state's dimensions, representing the dynamic component of the model. The state evolution process in an LDM can be based on either a discrete finite state machine, as in the case of HMMs, or a linear first-order Gauss-Markov process, as is typical in traditional linear dynamical systems (Rosti and Gales, 2001). In (Frankel, 2003), various LDM configurations are employed to address an acoustic modelling problem, with proposed techniques for composing a state evolution function that reflects articulatory dynamics.

The observation space in an LDM is mapped to a state space according a linear transformation function. The observation or measurement "error" is modelled as an additive Gaussian noise term, representing the approximate displacement of observations around a canonical articulatory process.External factors of speech such as speaker differences or environmental noise are captured by this measurement error. These variations are typically described as the extra-segmental variation. Together, the state and observation process of an LDM can be defined as:

$$s_t = As_{t-1} + \eta_t \tag{3.21a}$$

$$y_t = Hs_t + \epsilon_t \tag{3.21b}$$

The state process defined in Equation 3.21a with A being the state transition matrix, while the observation process is formulated in Equation 3.21b with H being a linear transformation matrix. It is not uncommon for the state and observation spaces to

have different dimensions, in which case the matrix H can be understood as a method for dimensionality reduction (Rosti and Gales, 2001). To summarise, an LDM can be parameterised as $\lambda = \{S, A, \eta, H, \epsilon, \Pi, \}$ where the additive Gaussian errors in the system η and ϵ are assumed to be uncorrelated.

Name	Notation	Meaning/Property
Sequence of states	$S = \{s_1 \dots s_N\}$	A sequence of N hidden states
Initial probability	$\Pi = \mathcal{N}(\mu_i, \Sigma_i)$	An initial state density s.t π_i represents the probability of starting in state i according to Gaussian parameters μ and Σ.
State evolution transformation	$A, \eta \sim \mathcal{N}(\mu_t^{(s_i)}, \Sigma_t^{(s_i)})$	A distribution defining the system dynamics and a state-evolution uncertainty defined by a Gaussian mean and variance $\mu_t^{(s_i)}, \Sigma_t^{(s_i)}$
Observation transformation matrix	$H, \epsilon \sim \mathcal{N}(\mu_t^{(y)}, \Sigma_t^{(y)})$	A mapping function from state to observation space and the measurement error defined by a Gaussian mean and variance $\mu_t^{(y)}, \Sigma_t^{(y)}$

3.4.1 Application of LDMs to ASR

Similar to the HMMs discussed earlier in this chapter, the LDM also upholds the Markov property. Figure 3.5 illustrates the state-observation dependency structure of an LDM. This figure shares a similar graphical layout to that of a HMM in Figure 3.3 with the only difference being the use of a continuous hidden variables rather than a discrete variables.

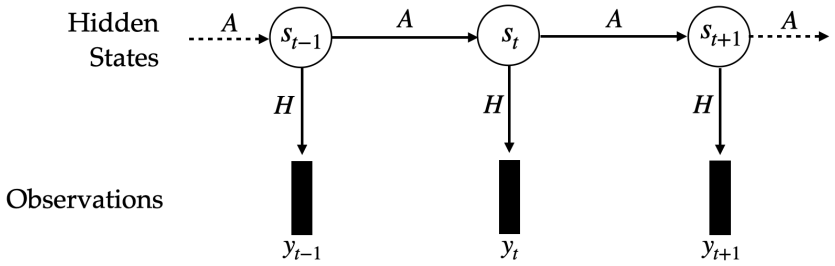


FIGURE 3.5: Diagram showing the general system architecture for a Linear Dynamic Model.

The joint probability of a sequence of states and observations can be factored such that:

$$P(Y,S) = P(s_1)P(y_1|s_1)\prod_{t=2}^{T}(s_t|s_{t-1})P(y_t|s_t), \tag{3.22}$$

where:

$$P(s_1) = \pi_1 = \mathcal{N}(\mu_{s_1}, \Sigma_{s_1}),$$

and the conditional densities for the state and output are:

$$P(s_t|s_{t-1}) = \mathcal{N}(As_{t-1} + \mu_t^{(s)}, \Sigma_t^{(s)}) \tag{3.23}$$

$$P(y_t|s_t) = \mathcal{N}(Hs_t + \mu_t^{(y)}, \Sigma_t^{(y)}). \tag{3.24}$$

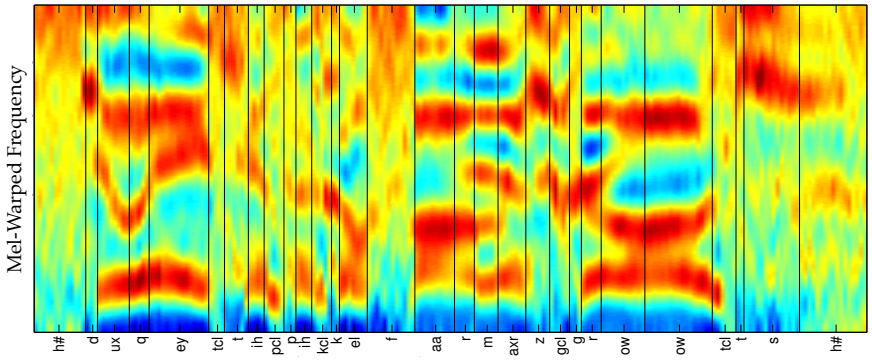
The likelihood of an observation sequence can be determined by integrating over all possible hidden sequences, as follows:

$$P(Y) = \int_S P(s_1)\ P(y_1|s_1)\prod_{t=2}^{T}(s_t|s_{t-1})P(y_t|s_t) \tag{3.25}$$

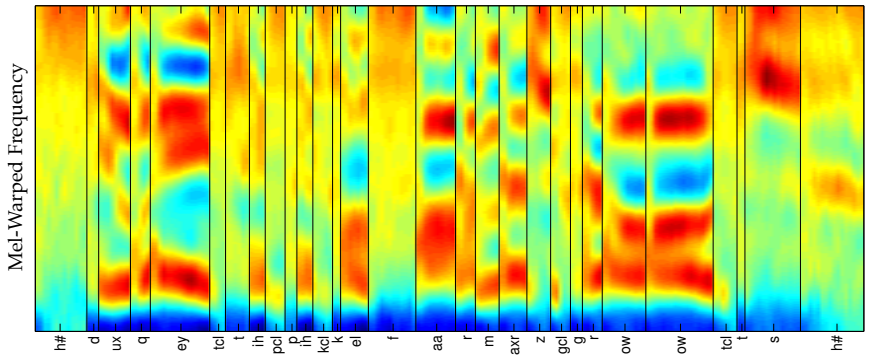
The use of LDMs for acoustic modelling has been explored in research studies demonstrating the effectiveness of LDMs applied to an ASR task. Additionally, the similarities between LDMs and HMMs are well documented in (Frankel, 2003; Rosti and Gales, 2001) with a key advantage of an LDM being attributed to the flexibility in modelling the state evolution.

Extensive experimentation detailed in (Frankel, 2003)reports that using an LDM can result in an overall 2.2% increase in accuracy compared to a traditional single-state HMM. This work also explores whether a linear system can effectively capture the systematic variation in speech data and whether the added complexity of a non-linear model yields improved performance. The results suggest that, under certain conditions, a linear system can perform similarly or better than its non-linear counterpart, particularly for the reconstruction of speech segments. An interesting experiment result

from (Frankel, 2003) is shown in Figure 3.6, which demonstrates the LDM's ability to capture the essence of acoustic data in a reconstruction utility task. Figure 3.6a displays the original Mel cepstra for the spoken phrase "Do atypical farmers grow oats?", while Figure3.6b presents the cepstra predictions generated by a set of trained LDMs. Despite the visible impact of resetting the state statistics at segment boundaries, many of the spectral features present in the original spectrogram can be discerned in the reconstruction.



(A) Spectrogram generated from Mel-cepstra.



(B) Spectrogram generated from LDM.

FIGURE 3.6: Comparative spectrograms taken from Frankel (2003) showing an LDM reconstruction of a spectrogram. -Red regions correspond to high energy, and blue to low.

Kalman Filtering

The role of Kalman Filtering is crucial in the context of linear Gaussian models and LDMs. Kalman Filtering is often used interchangeably in conjunction with terminology of LDMs (Quillen, 2000).A detailed explanation of this technique is presented in Section 3.6.3, however, a brief introduction to the principles of Kalman filtering in the context of Linear Gaussian models and LDMs will be detailed here. Specifically, The Kalman filter algorithm calculates the conditional expectation of a likelihood function, by minimising the mean square error of the estimated parameters in a system where all noise measures are Gaussian, which is the case with LDMs. This well-established recursive method is used to estimate the Gaussian mean and covariance parameters in the equations of LDMs (Equations 3.23 and 3.24).

Despite being a well-known inference algorithm in signal tracking literature, Kalman Filtering is less commonly used in the literature of automatic speech recognition. This has led to inconsistencies in naming conventions, and in some cases, modelling methods have been defined solely by their algorithmic recursion equations. To address this issue, this research aims to distinguish modelling definitions from algorithmic implementations, with all algorithmic definitions provided in Section 3.6.

3.5 Continuous State HMMs

The Continuous State HMM (CSHMM), is a generalised type of Markov model where the state space is continuous rather than discrete. While CSHMMs have been found useful in many fields, they are often referred to by other names such as Linear Gaussian models, Kalman-filter models, or Hidden Gauss-Markov Models; it is generally unknown that such models are in fact instances of a more generalised CSHMMs. A foundational theoretical study (Ainsleigh, 2001) shows that CSHMMs serve as an overarching general class of models to which these other named models can be considered special cases under certain constraints. Despite the potential utility of CSHMMs, they

have not been extensively studied in the context of signal processing and speech literature. This thesis aims to address this lack of attention given to CSHMMs by thoroughly examining their properties and capabilities. The experiments in this thesis are based on the CSHMM definition presented in (Champion and Houghton, 2016), which is a particular instance of a CSHMMs. The details of these models and experiments are outlines in Chapter 6.

The CSHMM is a first-order Markov process that generates data through an initial state probability density, a state-transition probability, and a state output probability distribution. Although the structure of a CSHMM is similar to that of a discrete state space HMM, the difference lies in the use of density functions to parameterise its distributions rather than discrete probabilities. A CSHMM can be parameterised as $\lambda = \{S, \Theta_1, \Theta_s, \Theta_y\}$, where Θ represents a model density.

Name	Notation	Meaning/Property
Sequence of states	$S = \{s_1 \dots s_N\}$	A sequence of N hidden states
Initial probability distribution	$P(s_1\|\Theta_1)$	The probability that a Markov process will start in state s_1
Transition probability	$P(s^j\|s^i, \Theta_s)$	The probability of moving from a state i to state j in a single time step.
Emission probability	$P(y_t\|s_t, \Theta_y)$	the probability of obtaining a particular measurement y_t, given a state s_t.

An informative table taken from Ainsleigh (2001) offers a useful comparison of the notation used in a standard HMM and a CSHMM. This table conveys the similarities and differences between the two models in a concise and informative manner. For clarity and ease of reference, this table is included here.

	HMM	CSHMM
Observation	y_t	y_t
State	$s_t = i$	$P(s_t^i; \Theta_i)$
Initial state probability	π_1	$P(s_1\|\Theta_1)$
Transition Probability	a_{ij}	$P(s^j\|s^i, \Theta_s)$
Emission probability	$b_i(y_t)$	$P(y_t\|s_t^i, \Theta_y)$

TABLE 3.1: Parameter notation for HMM and CSHMM from (Ainsleigh, 2001) adapted with updated notation.

The parameter Θ can represent any probability density function in its general form. However, it typically takes the form of well-defined density distributions such as Gaussian, Rayleigh, or gamma distributions. Regardless of the specific form of Θ, the parameters Θ_1, Θ_s, and Θ_y must be estimated from training data. As a result, Gaussian densities are the most commonly used probability density function in the literature, as demonstrated in the following studies (Ainsleigh, 2001; Ainsleigh et al., 2002; Champion and Houghton, 2016).

Let $Y = y_1, \ldots, y_T$ be an observation sequence and $S = s_1, \ldots, s'_T$ be a state sequence. It is important to note that there can be a many-to-one relationship between a state and observation, meaning that the state sequence may not be the same length as the observation sequence. Specifically, $1 \leq T' \leq T$. The state evolution is defined by a joint probability distribution of a sequence of observations and states, which can be expressed as:

$$P(Y, S) = P(y_1|s_1)\ P(s_1) \prod_{t=2}^{T} P(y_t|s_t)\ P(s_t|s_{t-1}) \tag{3.26}$$

Equation 3.26 is equivalent to the joint likelihood of the HMM in Equation 3.6. To calculate the total probability of a sequence of observations, the joint probability must be marginalised over all possible state sequences, as shown below:

$$\begin{aligned} P(Y) &= \int_S P(Y, S)\ dS \\ &= \int_S P(y_1|s_1)\ P(s_1) \prod_{t=2}^{T} P(y_t|s_t)\ P(s_t|s_{t-1})\ dS \end{aligned} \tag{3.27}$$

Equation 3.27 has the same form as the probability equation for the Linear Dynamical Model in Equation 3.25. This equivalence arises because the LDM is in fact a type of CSHMM, and can be factorised as such.

3.5.1 Application of Continuous State HMMs to ASR

The theoretical foundation of a CSHMM within the context of signal processing was established in earlier works such as (Ainsleigh, 2001; Ainsleigh et al., 2002). Recent research such as (Champion and Houghton, 2016) has explored the application of CSHMM to an ASR task using a particular naming convention and model notation. The proposed CSHMM system in (Champion and Houghton, 2016) draws inspiration from the Holmes-Mattingly-Shearme (HMS) speech synthesis model (Holmes et al., 1964) and seeks to provide a more dynamic and parsimonious system. The proposed models in this thesis build upon the HMS-inspired CSHMM and address a limitation of standard HMMs by addressing the independence assumptions in the modelling framework.

ASR and Text-to-Speech (TTS) tasks are differentiated based on their modelling characteristics and goals in applied speech technologies. ASR involves analysing human speech to transcribe it into written text, while TTS focuses on generating human-like speech from written text or other linguistic input. These tasks are conceptually distinct, yet both involve processing spoken language by computational systems. There are recommendations in (Paliwal and Rao, 1982) and (Holmes and Holmes, 2001) which advocate for applying the dynamic parametric models found in speech synthesis to a speech recognition task. This is the case for the CSHMM system proposed in (Champion and Houghton, 2016), which is primarily inspired by the Holmes-Mattingly-Shearme (HMS) speech synthesis model.

The HMS model assumes that speech can be modelled as a sequence of constant phonetic targets, with transitions between targets governed by linear trajectories in a suitable space. This hypothesis was explored based on the analysis of Formant frequencies in (Holmes et al., 1964), which demonstrated that the realisation of Formant frequencies in speech could be approximated as linear, providing the foundational basis for the HMS model. The application of a simplified HMS model to an acoustic modelling task using Formants was introduced in (Champion and Houghton, 2016)

and extended to alternative feature representations in (Weber et al., 2016a; Weber et al., 2016b). The work in this thesis further extends the CSHMM dwell-transition modelling framework to a segmental model context and addresses the problematic inter-segmental independence assumption of SHMMs.

Interpretation of the CSHMM Framework

The CSHMM model is designed based on the concept of a trajectory that is realised according to a canonical phonetic target. Figure 3.7 idealises the underlying modelling assumptions of a CSHMM. A canonical phonetic target is depicted as a dashed orange line with a realised trajectory represented as a blue line. Observations are marked as crosses and are distributed around the realised trajectory (blue line) based on a measurement error, known as the intra-segmental variation in both SHMM and LDM. The uncertainty between the realised trajectory and canonical trajectory is known as the extra-segmental variation in the context of a SHMM or the state-evolution measure in reference to an LDM. In either case, this error accounts for factors such as variation in a speaker's physiology, vocal tract length, or speaking style and is most commonly Gaussian.

A trajectory in the HMS-inspired CSHMM model can assume two forms, a piecewise-constant state which is referred to as a "dwell" state, or a piecewise-linear state referenced as a "transition" in (Champion and Houghton, 2016). Note, the term state in this context is not the same "state" that defines the Markovian model, but rather the structural constraint of the trajectory.

The dwell-transition framework for modelling a speech signal is, by design, an entirely continuous process with no discontinuities in the trajectories. This approach aligns with the physical constraints of speech articulators, which cannot instantaneously transition from one static position to another. Instead, this framework assumes that speech articulators move smoothly between configurations. From an acoustic perspective, a large part of speech consists of smooth transitions of acoustic features from one sound

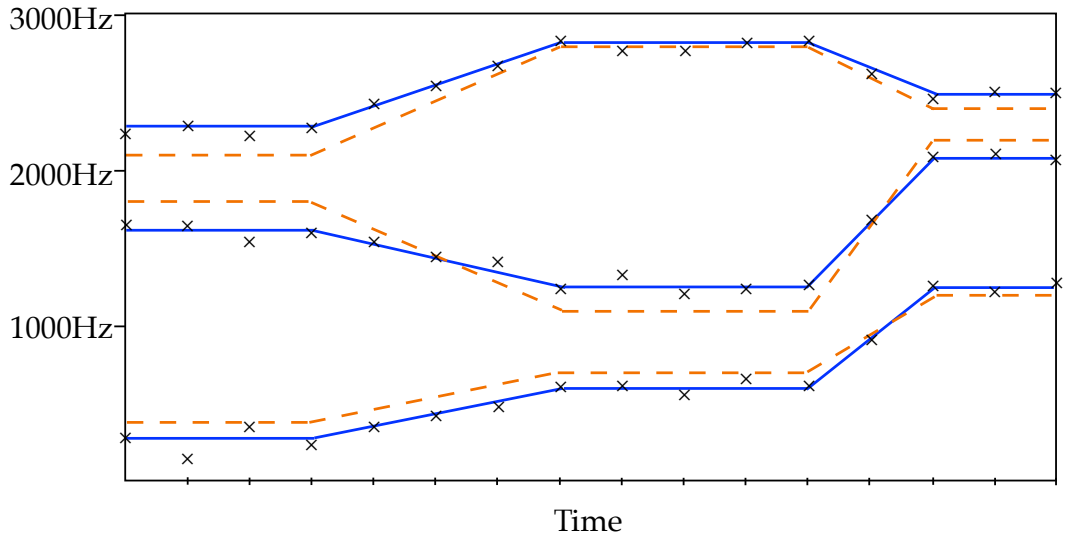


FIGURE 3.7: Idealised model of CSHMM formant tracks adapted from (Champion and Houghton, 2016) - Canonical trajectories are shown as dashed orange lines, realised trajectories shown as blue lines. Observations are black crosses distributed according to the realised trajectory.

to another as part of the source signal generation process. For example, smooth formant tracks can be observed in a spectrogram as discussed in Section 2.2.1.

Many factors can affect the performance of an ASR system, the models of interest in this work focus on the implicit continuity of the speech process, however, it is of equal importance that an appropriate feature representation complements the structure of the CSHMMs investigated. Previous research has explored different feature representations for CSHMMs, such as formant tracks (Champion and Houghton, 2016), bottleneck neural network features (Weber et al., 2014), and pseudo-time-dependent representations (Quillen, 2000). For consonant recognition, (Weber et al., 2015) used perceptually motivated spectral energy features as the feature representation. When training and decoding using a CSHMM, it has been established in (Ainsleigh, 2001) that the appropriate algorithms for a CSHMM closely resemble those of the Kalman Filter Algorithms, as described in Section 3.6.3.

The effectiveness of the dwell-transition CSHMM framework is attributed to both the flexibility of the modelling framework and the computationally efficient training and

inference algorithms which are extensions of the Kalman Filter equations. In addition, the CSHMM enables the representation of speech through piecewise-linear trajectories without the need for pre-segmenting the data, as is required for a SHMM. Therefore, the CSHMM is an attractive solution to model the dynamics of speech and address the independence assumptions of standard HMMs, similar to the SHMM and LDMs that have been discussed previously. The explicit derivation of the segmental CSHMMs and benchmark experimental results are presented in detail in Chapter 6.

3.6 Training and Decoding Algorithms

A well-cited benefit of using HMM-based models for an ASR task is the existence of clear training and decoding procedures. This section presents the Baum-Welch and Viterbi algorithms, followed by the A* Decoding algorithms. These techniques are often applied to the state space models described in this chapter and are also implemented in the experimental work presented in Chapter 6.

As previously stated in Chapter 1, the goal of an ASR task is to output the most likely sequence of words (W) given an input speech feature representation (Y) and a trained model (λ). This objective can be formalised as follows:

$$W^* = \underset{W}{\operatorname{argmax}}; P(y|\lambda); P(\lambda|W); P(W) \tag{3.28}$$

The training problem for an ASR task seeks to optimise the parameters of a model that will maximise the probability of a given observation sequence. This is a maximum likelihood optimisation task. For example, consider a standard HMM model $\lambda = \{A, B, \Pi\}$, the acoustic model parameters A and B can be efficiently estimated from a set of training utterances using an Expectation-Maximisation (EM) algorithm such as the Baum Welch algorithm (Baum et al., 1970) also known as the forward–backward algorithm or using Viterbi training process also known as forced alignment (Young et al., 2002).

3.6.1 Baum Welch Algorithm

The Baum-Welch algorithm is an Expectation Maximisation technique that consists of a forward-backward pass through the model states to approximate and optimise the model parameters. In order to construct the Baum-Welch algorithm, it is first necessary to define the forward and backward probability calculations. These calculations are detailed across many works, for ASR, most notably in (Rabiner, 1989).

Forward Algorithm

The forward algorithm is used in HMMs to calculate the joint probability of a partial observation sequence given a HMM model. It calculates the probability of being in a particular state at a given time, given the previous observations. The joint probability is represented as $\alpha_t(j)$, which is the probability of being in state s_j[3] at time t and having observed the first $t-1$ observations. This probability can be computed by summing over all possible hidden state paths in the model that could generate the observed sequence.

The $\alpha_t(j)$ probability is defined as:

$$\alpha_t(j) = P(y_1, y_2, \ldots, y_t, s_j|\lambda) \tag{3.29}$$

The formula for computing $\alpha_t(j)$ is:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(y_t) \tag{3.30}$$

This formula sums over all possible previous states i, multiplies the probability of transitioning from state i to state j by the probability of emitting observation y_t in state j, and then sums over all possible previous states i to obtain the total probability of being in state j at time t.

[3]The state subscript notation s_j in this section refers to a state symbol.

Figure 3.8 illustrates the general forward computations of being in state j at a time t after seeing the first $t-1$ observations. Transitions between all states may not be defined according to the Markov chain and so will not contribute to the forward probability of the current state, this will be the case when $a_{ij} = 0$.

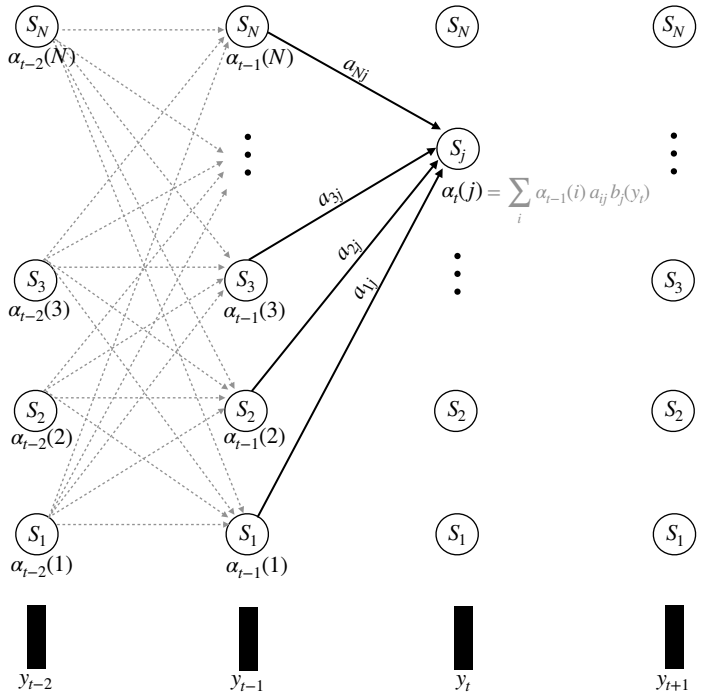


FIGURE 3.8: Diagram showing the forward algorithm computation trellis with the annotated calculation operations for an element $\alpha_t(j)$.

The forward algorithm is a recursive process that can be solved by induction, consisting of three steps:

1. Initialisation:

$$\alpha_1(j) = \pi_j \, b_j(y_1) \qquad 1 \leq j \leq N$$

2. Induction:

$$\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} b_j(y_{t+1}) \qquad 1 \leq j \leq N, 1 \leq t \leq T$$

3. Termination:

$$P(Y, S|\lambda) = \sum_{i=1}^{N} \alpha_t(i)$$

Calculating α_t involves extending the probability of the previous time step, α_{t-1}, and incorporating the transition and emissions probabilities for the specific state that is occupied. This recursive process is computationally efficient. However, computing the joint probability $P(Y, S)$ requires summing over all potential state pathways, resulting in a computation complexity of $\mathcal{O}(N^2T)$ where N is the number of state symbols in a state space and T is the length of the observation sequence.

Backward Algorithm

The backward algorithm calculates the probability of a partial observation sequence from a time $t + 1$ to the end of the sequence, given a state s_i at time t.

The $\beta_t(i)$ probability is defined as:

$$\beta_t(i) = P(y_{t+1}, y_{t+2}, \ldots, y_t | s_i, \lambda) \tag{3.31}$$

This probability can be calculated recursively by:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(y_{t+1}) \beta_{t+1}(j) \tag{3.32}$$

Figure 3.9 illustrates the computation operations required to calculate the backward procedure.

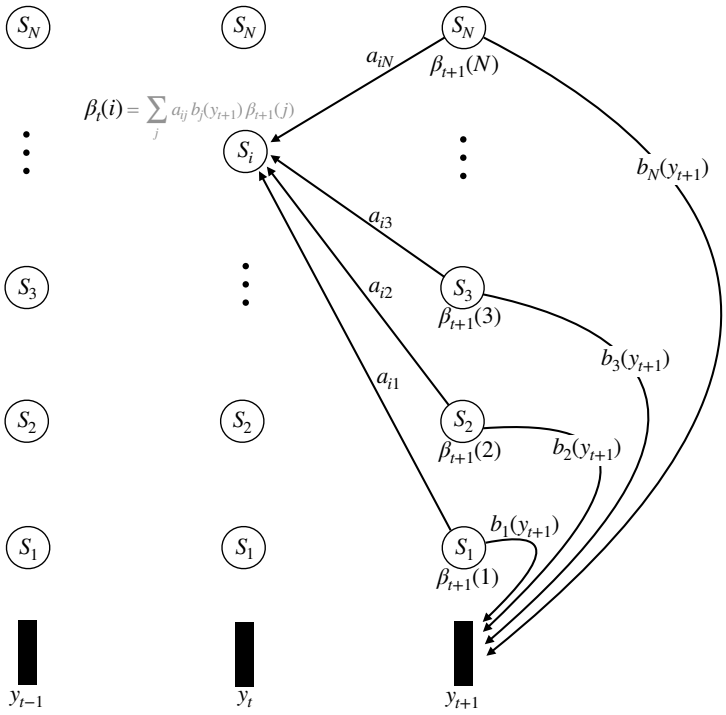


FIGURE 3.9: Diagram showing the backward algorithm computation with the annotated calculation operations for an element $\beta_t(i)$.

The backward probability calculation is similar to the forward algorithm, it can be calculated recursively considering all possible state pathways for the remaining observations from each state. The steps to calculate this recursion are as follows:

1. Initialisation:

$$\beta_T(i) = 1 \qquad 1 \leq i \leq N$$

2. Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(y_{t+1}) \beta_{t+1}(j) \qquad 1 \leq i \leq N, 1 \leq t \leq T$$

3. Termination:

$$P(Y) = \sum_{j=1}^{N} \pi_j b_j(y_1) \beta_1(j)$$

The backward and forward algorithms are required for the Baum Welch Training procedure, which updates the model parameters of a HMM.

Baum Welch Re-estimation

The Baum-Welch algorithm is a method for estimating the parameters of a HMM by maximising the likelihood of the observed data. The algorithm uses the forward and backward probability calculations to update the model parameters A and B. This Expectation-Maximisation (EM) algorithm is used to approximate these parameters. To define the EM steps of the Baum-Welch algorithm, two additional variables are introduced.

The first variable ξ_t represents the probability of being in state s_i at time t and state s_j at time $t+1$, given an observation sequence Y and a model λ:

$$\begin{aligned}\xi_t(i,j) &= P(s_t = i, s_{t+1} = j|Y) \\ &= \frac{\alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}\end{aligned} \tag{3.33}$$

The second variable $\gamma_t(j)$ represents the probability of being in state s_j at time t, given an observation sequence Y and a model λ. This variable is defined as:

$$\begin{aligned}\gamma_t(j) &= P(s_t = j|Y) \\ &= \frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)} \\ &= \sum_{j=1}^{N}\xi_t(i,j)\end{aligned} \tag{3.34}$$

Finally, during model training, the parameter update equations for the initial state probability $\hat{\pi}_i$, transition probability $\hat{a}_{ij}$ and a symbol emission probability $\hat{b}_j(k)$ can be computed such that:

$$\hat{\pi}_i = \gamma_1(i) \qquad 1 \leq i \leq N \tag{3.35}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \qquad 1 \leq i \leq N, 1 \leq j \leq N \tag{3.36}$$

$$\hat{b}_j(k) = \frac{\sum_{t=1, y_t=\phi_k}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)} \qquad 1 \leq i \leq N, 1 \leq j \leq N \tag{3.37}$$

The notation $\sum_{t=1, y_t=\phi_k}^{T-1}$ refers to the sum over all values of t for which the observation at time t is equal to the specific label represented by ϕ_k. In the context of an ASR task, ϕ_k may represent a particular phoneme label.

3.6.2 Viterbi Algorithm

The Viterbi algorithm is a dynamic programming algorithm that is widely used in ASR and other applications for finding the most likely sequence of hidden states given a sequence of observed events for a probabilistic model denoted as λ, (Viterbi, 1967). The algorithm computes the joint probability of an observation sequence and a state sequence $P(Y, S)$. The optimisation criterion for the Viterbi algorithm uses the Maximum A Posteriori (MAP) to estimate an optimal state sequence s^* such that:

$$s^* = \arg\max_s P(Y, S) \tag{3.38}$$

The Viterbi algorithm constructs a computation trellis that represents all possible sequences of hidden states. At each time step, the algorithm computes the probability of transitioning from one hidden state to another, based on the likelihood of the observation which is the emission probability and a transition probability. The path with the highest overall probability is extended to the next time step, which results in the most likely sequence of hidden states at time T. Figure 3.10 provides a visual representation of the Viterbi trellis.

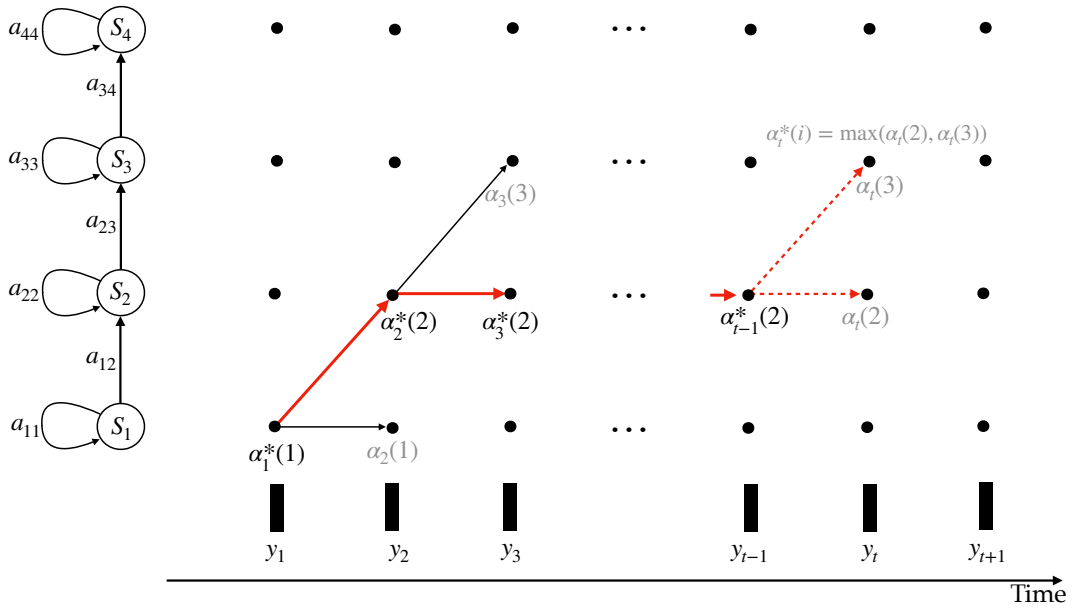


FIGURE 3.10: Visualisation of a Viterbi computation trellis for efficient decoding.

Similar to the Baum Welch forward-algorithm, the Viterbi algorithm operates left-to-right iteratively calculating the probability of being in a state s_i after observing the first $t-1$ observations. Figure 3.10 shows each cell of the trellis is represented by $\alpha_t(i)$ and the most likely path $\alpha^*(i)$ is extended at each time step. This $\alpha^*(i)$ probability can be expressed as:

$$\alpha_t^*(i) = \max_s P(y_1, \ldots, y_t, s_1, \ldots, s_i) \tag{3.39}$$

This can be solved recursively by computing:

$$\alpha_t^*(i) = \max_s \alpha_{t-1}^*(i)\, a_{ji}\, b_i(y_t) \tag{3.40}$$

It is important to note that α^* is not the same as the forward algorithm α probability. It can however be considered as an approximation of the forward probability α variable. The difference being the forward algorithm computes the actual probability of being in any given state at any given time by summing across all possible paths that pass through that state at time t. Whereas the Viterbi probability calculation extends only the single most likely path and consequently only calculates an approximate state

occupancy probability.

The Viterbi algorithm can also be used as a training procedure, in which case it is referred to as forced alignment. An advantage of Viterbi training is the computational efficiency of the algorithm; works such as (Bahl et al., 1983) highlight several of these advantages of Viterbi training and decoding when compared to other MLE and decoding algorithms. In the context of ASR, it has been shown that in some cases, Viterbi training performs the same or only slightly worse than a Baum Welch training procedure when applied to speech recognition systems while reducing the number of computations which is advantageous (Rodríguez and Torres, 2003). Despite its utility for training HMM model parameters, the Viterbi algorithm is most commonly used as a decoding method.

The Viterbi decoder computes the most likely sequence of states based on trained models and a set of unseen observations. It is a breadth-first search scheme in which all candidate hypotheses are evaluated at a time t before extending the search to time $t+1$. Formally, the actual likelihood of an observation sequence is calculated by summing over all possible state pathways and is computed by the forward algorithm:

$$P(Y) = \sum_{S} P(Y, S) \tag{3.41}$$

The Viterbi algorithm approximates the total probability by extending the probability of a best path state sequence such that:

$$P(Y) = \max_{S} P(Y, S) \tag{3.42}$$

The Viterbi recursion is the same as the forward-algorithm recursion formula from Section 3.6.1, except that instead of the sum component $\sum_S$, it employs a $\max_S$ assignment.

Viterbi decoding, while efficient, has the limitation of being bound to the Viterbi criterion, which dictates that when two paths occupy the same state at a given time, the

path with the locally lower likelihood will never supersede the other. This can present challenges in cases where a word has multiple possible pronunciations, as only the most likely pronunciation will be considered, potentially resulting in a sub-optimal global performance.

3.6.3 Kalman Filtering

The Kalman filter is a popular algorithm used in a variety of fields such as control systems, navigation systems, and signal processing. It is a stochastic optimal estimator method that aims to minimise the mean squared error by finding the best estimate of an unknown parameter or variable (Kalman, 1960). The Kalman filter algorithm uses a sequence of measurements observed over time that contain noise or other inaccuracies and produces estimates of the unknown hidden variables. These estimates tend to be more precise than those based on a single measurement alone.

A Kalman Filter can be used to calculate the forward-backward probabilities of the Baum Welch Algorithm (Section 3.6.1) for a continuous hidden state space model. It has been applied to a LDM (Digalakis et al., 1993) discussed in Section 3.4 and the lesser-known CSHMM (Ainsleigh et al., 2002) Section 3.5. The Kalman filter algorithm is a type of expectation-maximisation strategy (EM) that involves two steps: prediction and correction. In the prediction step, the filter estimates the state posteriors given the observations up to the current time index, this is related to the forward Baum Welch α calculation. In the correction step, the filter uses the entire measurement sequence to refine the estimate of the system's current state.

To formalise the prediction step, consider a sequence of observations $Y = y_1, \ldots, y_t$ and a current state s_j followed by a prior state s_i. The forward probability computes the probability of a partial sequence of observations given a state s_j. Let the forward

component, $\alpha_t(j)$, be defined as follows:

$$\alpha_t(j) = P(y_1, y_2 \dots y_t, \dots, s_i, s_j) \tag{3.43}$$

$$= P(y_t|s_j)\ P(y_1, y_2 \dots y_{t-1}, \dots, s_i) \tag{3.44}$$

$$= P(y_t|s_j) \int_{s_i} P(y_1, y_2 \dots y_{t-1}, \dots, s_i)\ ds_i \tag{3.45}$$

$$= P(y_t|s_j) \int_{s_i} P(s_j|s_i) P(s_i, y_1, y_2 \dots y_{t-1})\ ds_i \tag{3.46}$$

$$= P(y_t|s_j) \int_{s_i} P(s_j|s_i)\ \alpha_{t-1}(i)\ ds_i \tag{3.47}$$

The correction step of a Kalman filter is related to the Baum Welch backward calculation β. This calculation is sometimes referred to as the Kalman smoother or the Rauch-Tung-Streibel smoother (Rauch et al., 1965). The backward step calculates the best estimate for a state s_j using all of the observed data so far while also utilising all the data from a time $t+1$ to the end of the observation sequence. The calculation of $\beta_{t-1}(i)$ is presented as:

$$\beta_{t-1}(i) = P(y_t, y_{t+1}, \dots, y_T, s_{t+1}, \dots, s_T|s_i) \tag{3.48}$$

$$= \int_{s_j} P(y_t, y_{t+1}, \dots, y_T, s_t = j|s_i)\ ds_j \tag{3.49}$$

$$= \int_{s_j} P(s_j|s_i)\ P(y_t|s_j) P(y_{t+1}, \dots, y_T, s_{t+1}, \dots, s_T|s_j)\ ds_j \tag{3.50}$$

$$= \int_{s_j} P(s_j|s_i)\ P(y_t|s_j) \beta_t(j)\ ds_j \tag{3.51}$$

With the initial components of the recursion being:

$$\alpha_1(s_1) = P(s_1)\ P(y_1|s_1) \tag{3.52}$$

$$\beta_T(s_T) = 1 \tag{3.53}$$

When the state variables are discrete, the integrals in Equations 3.47 and 3.51 correspond to Equations 3.30 and 3.32, respectively, and can be computed exactly by calculating

the summation. However, when the state variable is continuous, the state parameters must be stored, and the integrals must be solved analytically. The added complexity in computation imposes constraints on the allowable transition and emission densities that can be used in practice. While, in theory, any density adhering to the properties of the exponential family can be used, in practice, the model distribution is assumed to be Gaussian for both the LDM and the CSHMM.

Given a standard Gaussian distribution,

$$\mathcal{N}(y;\mu,\Sigma) = \frac{1}{|2\pi\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right), \tag{3.54}$$

the conditional state and observation variables s_i, s_j and y_t are also Gaussian such that:

$$P(s_j|s_i) \sim \mathcal{N}(As_i,\Gamma) \tag{3.55}$$

$$P(y_t|s_j) \sim \mathcal{N}(Hs_j,\Delta) \tag{3.56}$$

$$P(s_1) \sim \mathcal{N}(\mu_1,\Sigma_1), \tag{3.57}$$

Where A and H represent a transition and state-observation mapping matrix as discussed in Section 3.4. The state and observation variables can be written more conveniently as a set of linear equations driven by noise, documented in (Ghahramani and Hinton, 1996) as:

$$s_j = As_i + \eta \tag{3.58}$$

$$\eta \sim \mathcal{N}(0,\Gamma) \tag{3.59}$$

$$y_t = Hs_j + \epsilon \tag{3.60}$$

$$\epsilon \sim \mathcal{N}(0,\Delta) \tag{3.61}$$

The Kalman filter is initialised according to Equations 3.52 and 3.53. The filter can then be solved recursively by substituting Equations 3.55 and 3.56 into the $\alpha_t(j)$ and

$\beta_{t-1}(i)$ probability expressions in Equations 3.47 and 3.51 respectively. The state and observation Gaussian parameters can be updated according to a Gaussian factorisation lemma defined in Appendix B. A number of previous studies have explicitly derived these recursions in the context of LDMs and CSHMMs, such as (Ainsleigh et al., 2002; Rosti and Gales, 2001; Frankel, 2003). In particular, (Ainsleigh et al., 2002) derives the generalised Kalman filter equations in the context of a CSHMM and further highlights the relationship to the Baum Welch re-estimation parameters. These studies have been fundamental in deriving an efficient solution for testing and evaluating the novel Segmental CSHMM proposed in this thesis.

3.6.4 A* Decoding

The A* algorithm is a graph search method designed to determine the shortest path between two nodes in a graph, as originally introduced in (Hart et al., 1968). Its application to speech recognition was explored in (Jelinek, 1969). Unlike the Viterbi decoder, the A* search algorithm is an asynchronous best-first search algorithm, meaning that the evaluation of potential pathways through a sequence of states does not need to occur concurrently or in a predetermined order. Instead, multiple path hypotheses can be considered at different indexes as long as the required data is available.

The A* search algorithm's primary component is an evaluation function f^*. This evaluation function is a combination of likelihoods describing the cost of a partial path from the starting point to a state g^* and a heuristic estimated cost from a state to the end goal h^*.

$$f^* = g^* + h^* \tag{3.62}$$

In the context of acoustic modelling, the cost function corresponds to the likelihood of observations given the acoustic, lexical, and language models:

$$g^* = P(y|\lambda)\ P(\lambda|w)\ P(w) \tag{3.63}$$

Where $P(\lambda|w)$ is given by a lexical model and $P(w)$ by a language model. See Section 1.2 for details of these model components. The heuristic cost function represents the acoustic likelihood of the remainder of the observations given a model λ:

$$h^* = P(y_{t+1}, \ldots, y_T|\lambda) \tag{3.64}$$

The A* algorithm can be referred to as a Stack Decoder, however, it does not correspond to the computer science definition of a last-in-first-out data structure. Instead, it references the implementation methodology of "stacking" hypotheses in a priority queue (Sturtevant, 1989). A* stack decoding is an iterative process where, for each iteration, all hypotheses on the stack are extended by computing the evaluation function according to the new index and adding each hypothesis to the queue. The A* algorithm terminates when the end of the speech data is reached, the first hypothesis popped from the stack is then considered the most promising hypothesis.

The decoder maintains a stack of hypotheses, where each hypothesis includes information regarding the partial state sequence (path history), the current language model state, an end-of-sentence pointer, and the evaluation function based on a state start and end index. The basic computation operations of the A* stack decoder are summarised in the following pseudocode.

Pseudocode A* Stack Decoder

```
1: Initialise the stack with a null state for N hypotheses.
2: Pop the best (highest scoring) hypothesis off the stack reducing the stack size by 1.
3: if End-of-sentence == True then
4:     Output the state sequence and terminate.
5: end if
6: for Each word on the candidate list do
7:     Perform acoustic and language-model probability update calculations to compute the new hypothesis output log-likelihood.
8:     if End-of-sentence then
9:         Insert into the stack with end-of-sentence flag = TRUE.
10:    end if
11:    if Not end-of-sentence then
12:        Insert into the stack with updated state label.
13:    end if
14: end for
15: Go to 2.
```

A limitation of the A* algorithm is that the evaluation function is a product of probabilities shown in Equations 3.63 and 3.64. As probability values lie within the interval [0,1], the product of probabilities will result in the evaluation function (score) getting progressively smaller with each iteration, causing the algorithm to always favour shorter hypotheses. To address this issue, the score of a hypothesis can be normalised based on the number of frames of data it spans. However, despite this normalisation, the A* stack decoder may still encounter difficulties with issues such as speed, size, accuracy, and robustness when applied to large vocabulary speech recognition tasks (Paul, 1991; Frankel, 2003).

3.7 Summary

This chapter discusses several statistical approaches explored in the context of acoustic modelling for a speech recognition task. Three different acoustic modelling approaches are discussed: frame-based hidden Markov models (HMMs), segment-based acoustic models (SHMMs), and Continuous state hidden Markov models (CSHMMs), which include Linear Dynamical models (LDMs).

In the frame-based acoustic model, which historically has been the dominant approach,

each state in a HMM is mapped to a single observation, with a state emission distribution typically described by a Gaussian mixture model (GMM). While this approach has been highly successful, the standard HMM acoustic model is limited by its core modelling assumptions, including the hypothesis that speech features are temporally uncorrelated. These assumptions lead to several fundamental limitations in the modelling framework, such as the independence assumption, the implicit piecewise stationarity of a state, and a default geometric duration distribution. Although most HMM and hybrid systems attempt to mitigate these limitations by incorporating delta and delta-delta features, the underlying dynamic process is highly complex and cannot be fully captured through simple feature enhancements. Therefore, more sophisticated modelling techniques are necessary to capture the complex dynamics of speech signals effectively.

Segment-based HMMs were proposed to improve the underlying HMM assumptions by incorporating frame-to-frame correlations from the speech signal to improve recognition performance, with active research dating back to the 1990s. Despite the progress made with SHMMs, the success of these models has been primarily limited to small-scale tasks. The segmental model can be seen as an expansion of traditional HMMs, allowing variable-length segments to be represented as states, thereby relaxing the assumption of conditional independence between observations within a segment. Applying segmental models to large-scale ASR poses several challenges, some of which have been explored in works such as (Holmes, 1997; Russell and Jackson, 2005; Russell, 2005), and (Gales and Young, 1993). A survey article (Deng and Li, 2013) provides an overview of various types of segmental models, identifying two recommended research opportunities: incorporating scientific knowledge about the underlying articulatory speech dynamics and developing efficient computation methods for training and decoding in ASR applications. This thesis addresses both of these challenges.

A Linear Dynamical Model is an alternative approach to addressing the frame-wise independence assumption of HMMs. This approach formulates the ASR problem as a

state-space model, where the speech signal is represented as a linear combination of hidden states that evolve over time according to a first-order Markov process. Hidden states are assumed to be Gaussian distributed, with observations assumed to be linearly related to the hidden states with additional noise. The LDM can capture long-range temporal dependencies in the speech signal which is essential for accurate recognition. Despite their effectiveness, LDMs are computationally expensive to train and require careful tuning of hyper-parameters, as pointed out by previous works such as (Frankel, 2003) and (Rosti and Gales, 2001).

In other works such as (Champion and Houghton, 2016), a continuous state-space model is defined according to a dwell-transition modelling assumption motivated by the Holmes-Mattingly-Shearme model Holmes et al., 1964. This CSHMM system is described as a "similar" method to a Kalman Filter, however, it is of the opinion in this work that as formulated by (Ainsleigh, 2001), a Continuous State HMM is a generalised system with which the dwell-transition underlying trajectory, and Kalman filter formulation has the same underlying trajectory assumption.

In the dwell-transition system which has inspired the applied models in this work, the trajectory constraints are described as static (dwell) regions connected by linear transitions, consequently enforcing continuity throughout the entirety of an utterance. The research presented in this thesis extends these trajectory constraints using a CSHMM framework, thereby extending the contribution of applied CSHMMs to the field of ASR.

This analysis of Markovian-based statistical models is presented in a tutorial-style format to provide the foundational knowledge and intuition necessary to understand the Segmental CSHMMs proposed in this work in Chapter 6. The literature surveyed in this section covers several decades, and a simplified and consistent notation is adopted to facilitate the comparison of concepts across different model descriptions. It is not possible to directly compare phone recognition results across the different models due to the variations in the experimental setup, such as the feature representation,

the inclusion of language models, and the train/test data split, which impacts the evaluation dataset. However, Table 3.2 reports a range of baseline performance metrics on a TIMIT phone recognition task relevant to the models presented in this chapter. This table serves as a reference for future research involving Markovian-based statistical models.

TABLE 3.2: List of reported phone error rates (PER) on the TIMIT speech corpus in the last decade - Including details of model architecture and features used for experimentation, ordered by publication date.

Year	Reference	System Description	Feature Representation	% Acc	TIMIT Test Set
1989	Lee and Hon, 1989	HMM	MFCC + $\Delta, \Delta\Delta$	60.08	160 utterances from Test set (TID7)
1992	Young, 1992	HMM	MFCC + $\Delta, \Delta\Delta$	59.9	160 utterances from Test set (TID7)
1993	Lamel and Gauvain, 1993	Continuous Density HMM	MFCC + $\Delta, \Delta\Delta$	72.9	Full Test
1993	Gales and Young, 1993	SHMM	MFCC + $\Delta, \Delta\Delta$	53.51	Dialect Region 2 utterances (DR2)
1997	Holmes and Russell, 1999	Static Trajectory SHMM Linear Fixed Trajectory SHMMs Linear Probabilistic Trajectory SHMMs	MFCC + $\Delta, \Delta\Delta$	67.8 72.6 73.2	All Male speakers of the TIMIT Core Test set
2003	Frankel, 2003	LDM	MFCC + $\Delta, \Delta\Delta$	60.3	Core Test
2015	Weber et al., 2015	CSHMM	MFCC	55.6	Consonant segments from full test set excluding 'SA' utterances
2015	Houghton et al., 2015	CSHMM	Formants + Log-Energy	48.2	Vowel segments from single speaker "MWEW0" from DR2
2017	Bai et al., 2015	HMM	BNF + $\Delta, \Delta\Delta$	73.07	Core Test

Chapter 4

Acoustic Feature Representation

This chapter presents a comprehensive analysis of bottleneck features (BNFs), introduced in Section 2.2.5 and serves as the basis for the experimental investigations conducted in this thesis. The primary objective of this research is to explore a parsimonious yet reliable model of the speech production process, as previously stated. The properties of speech features must align with the model designed to interpret and classify them. An optimal speech feature representation should possess a robust extraction methodology and stability across different speech sounds, regardless of variations in speaker or environmental characteristics. To this end, this thesis surveys existing research on BNFs followed by the specific extraction method employed for the data and experiments utilised in this study. Finally, a visual analysis of the specific BNFs used in the experiments described in Chapter 6 and 7 is provided. This analysis serves to support the feature design decisions made in this work.

4.1 Bottleneck Feature Representation

There has been a significant effort to investigate alternative feature representations and models of speech that more accurately capture the dynamical properties of speech in recent years. Several techniques are discussed in Section 2.2. However, the emergence of neural networks (NNs) has led to alternative approaches to feature extraction, such

as those proposed by (Grezl et al., 2007), who considers features derived from intermediate hidden layers of NN. The layer of the NN which the features are extracted is typically much smaller than the input and output layers, hence the name "bottleneck", they have been shown to be effective in reducing the dimensionality of speech features while maintaining relevant information for ASR. In (Deng and Chen, 2014), a comparative study is presented of different static DNN architectures for extracting high-level acoustic features. The results of this study show that the extracted features are temporally more smooth than the raw acoustic data sequence. Furthermore, the study found that BNFs yielded better phone recognition accuracy across all of the DNNs explored when compared to input raw filter-bank features. Similarly, in (Jiang et al., 2014), the authors investigated the use of low-dimensional features extracted from a Deep Belief Network (DBN). The study found that the low-dimensional compact representation obtained from the DBN has "powerful, descriptive, and discriminative capabilities". Furthermore, the experimental results of the ASR task using the DBN representation showed improved performance for short-duration utterances.

The range of research conducted on BNFs has looked at systems with layers having tens to hundreds of neurons. For example, (Doddipatla, 2016) considered a 75-dimensional BNF layer, (Petridis and Pantic, 2016) extracted 500-dimensional BNFs, whereas (Bai et al., 2015) described a very low-dimensional BNF layer with dimensions ranging from 3 to 32. While there is no obvious optimisation of BNF dimension, there is evidence that a BNF feature representation improves overall recognition across a broad number of tasks and datasets when DNN is carefully constructed and trained. This thesis aims to contribute to parsimonious speech modelling research by extending the use case of the very low-dimensional BNFs derived in (Bai et al., 2015), applied to a Segmental CSHMM framework.

The very low dimensional features derived in (Bai et al., 2015) can be interpreted as a compressed representation of speech. One of the key findings of this work is that, when

comparing the BNF representation to 39-dimensional MFCCs in a standard HMM-GMM based system, the BNFs outperform the corresponding MFCC system with respect to recognition accuracy, with improvements between 2% to 4%, depending on the size of the bottleneck. An interesting finding of the study was that the monophone BNFs consistently exceeded the performance of the triphone BNFs for varying network configurations. The authors suggested that the bottleneck features capture both spectral and temporal properties of speech and that contextual information may be compressed due to the very low feature dimensionality. These findings motivate the current study, to explore the use of low-dimensional BNFs in the Segmental CSHMM framework.

In a follow-up study by (Weber et al., 2016b), these low dimensional bottleneck features were explored with respect to a phonetic, articulatory, and acoustic analysis and primarily compared to Formants. The qualitative conclusions from this work highlight similarities between the spacial representation of BNFs in vowel phonetic regions and the F1:F2 vowel quadrilateral. For other phonetic categories such as fricatives, this work presents a strong correspondence to the findings in (Choo and Huckvale, 1997), which projects fricatives such as /s/, /sh/, /f/, /th/ and /hh/ in a two-dimensional space where the axes can be loosely interpreted as sibilance and place of articulation. A conclusion of (Weber et al., 2016b) suggests that a unified feature space for both voiced and unvoiced speech regions can be achieved using low dimensional BNFs. A final remark in the study states that "the networks generating the features are in some sense learning to recognise speech, rather than spurious characteristics of the data".

Furthermore, (Bai et al., 2018) extends their original feature extraction research and explores visualisation techniques to interpret the 9-dimensional bottleneck features and draw parallels to known structural patterns occurring in human speech. Two techniques, Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbour Embedding (t-SNE) are employed to interpret the 9-dimensional BNFs. These techniques are applied to a set of recommended broad phone categories from (Halberstadt and Glass, 1998) presented in Table 2.1. Figure 4.1, taken from (Bai et al., 2018), shows a

2D mapping of BNFs for each phone label.

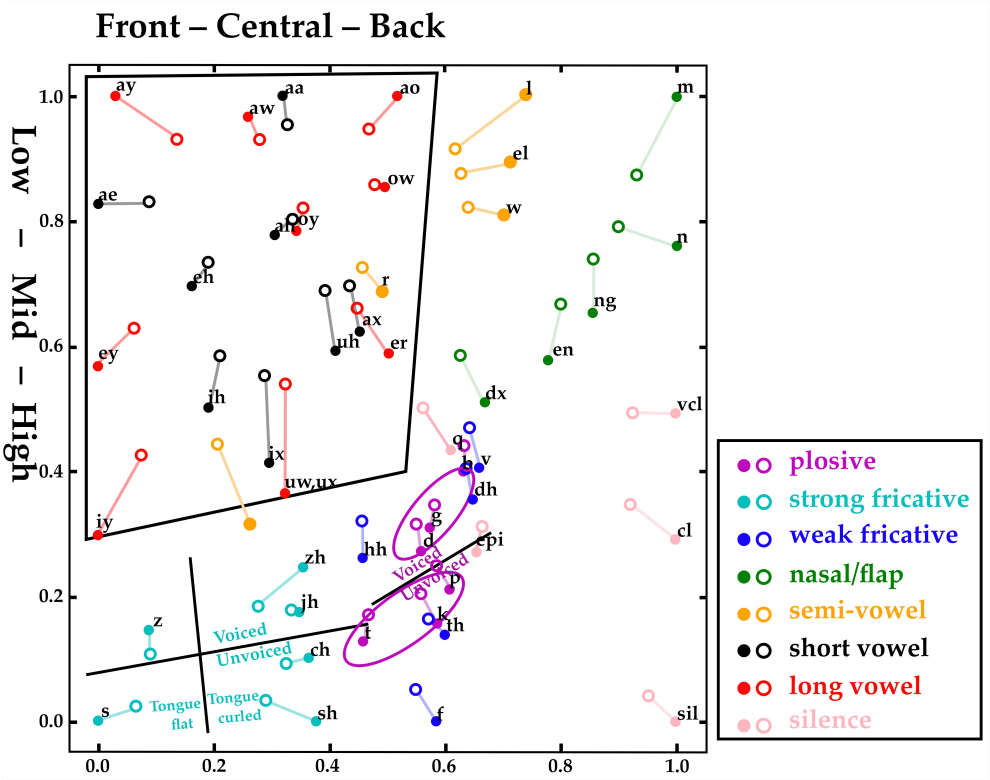


FIGURE 4.1: Optimised 2D BNFs (dots) and feature means of 2D BNFs (circles) for each phone for a phone classification DNN. - Original figure in (Bai et al., 2018)

The shaded region in the upper left corner of the graph under examination reveals a resemblance to a traditional F1:F2 vowel space diagram in terms of the arrangement of long and short vowels. An analysis of this projected space suggests that the positions of the vowels roughly correspond to places of articulation, with the phones /ay/, /ey/, /iy/ located on the left side and corresponding to low tongue positions, while the phones /ao/, /uh/, /uw/ are located on the right side and correspond to high tongue positions. Furthermore, a horizontal examination from left to right demonstrates that the phones /ey/, /ah/, /ow/ correspond to a progression in tongue position from front to back.

Now consider the strong fricatives, phones which are produced with the tip of the tongue curled up, such as /zh/, /jh/, /ch/ and /sh/. This set of phones possesses

good separation from phones produced with a flat tongue, such as /s/ and /z/. Furthermore, it is possible to separate these strong fricatives according to voicing properties. The same holds true for plosives, with a clear boundary separating voiced plosives, such as /d/, /g/, /b/, and unvoiced plosives, such as /t/, /k/, /p/. For voiced and unvoiced plosives, phones are placed horizontally in an order that reflects their place of articulation from left to right: teeth, soft palate and lips. It is important to note that while the axes Figure 4.1 are not universal for interpreting all phonetic categories in unison, localised regions can be analysed, highlighting phonetic structures that correspond to articulatory configurations. This conclusion suggests that the BNF network is capable of compressing all phones into a single feature space and that each phone category lies within nicely separable sub-spaces. Within these sub-spaces, the organisation of phones appears to correspond to phone production mechanisms.

4.2 Experimental Setup: Bottleneck Neural Network Structure

The architecture of the bottleneck NN used in these experiments is a five-layer system that comprises an input layer of 256 units, a hidden layer of 512 units, a bottleneck layer of 9 units, another hidden layer of 512 units, and an output layer of a size relative to the output target. The input data to the network uses Mel-scaled filter-bank energies. The TIMIT corpus sampled at 16 kHz was analysed using a 25*ms* Hamming window with a 10*ms* frame rate using the phone mapping documented in Appendix A. The architecture of the network is illustrated in Figure 4.2, which was adapted from the description provided in (Bai, 2018).

Two experimental frameworks are presented in (Bai, 2018). The first experimental framework produces a BNF dataset referred to as Dataset 1, which is derived from a reconstruction experiment. The objective of this experiment was to reconstruct an input spectrum signal using a NN. In this particular case, the target output of the NN was of the same dimension as the input signal, specifically 256 units. The second experimental framework produces a different BNF dataset named Dataset 2, which is derived from a

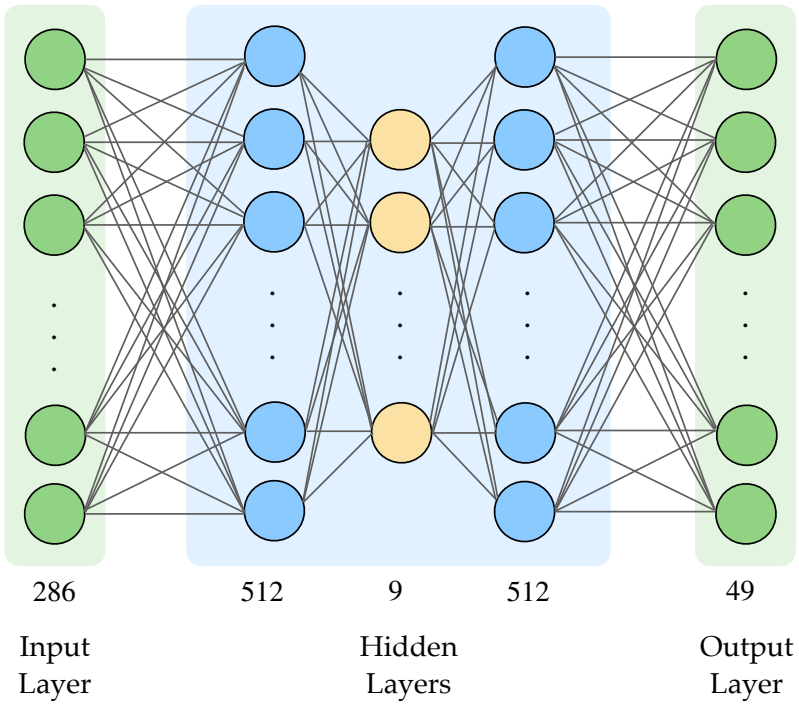


FIGURE 4.2: Five-layer neural network architecture used to extract low-dimensional bottleneck features. Adapted from description in (Bai, 2018)

phone discrimination experiment. This experiment aimed to use a NN to discriminate between different phones. In this case, the output targets of the NN can be interpreted as the posterior probabilities of the 49 phonemes. The datasets used in this work are identical to those used in (Bai, 2018), with no additional manipulation or alteration.

The experiments presented in (Bai, 2018) show remarkable results, demonstrating that a very low-dimensional BNF representation could produce results comparable to those obtained using MFCC data with an HMM-GMM ASR system. A condensed summary of the findings, as documented in (Bai, 2018), is provided in Table 4.1, including the experimental results obtained using Formants. The use of bottleneck-based features leads to a notable improvement in phone recognition accuracy when compared to formant-based features of the same dimension. Specifically, when comparing BNFs to 3-dimensional formant-based features, a comparative improvement of 20.23% in phone recognition accuracy is observed. The same trend is observed when comparing

BNFs to MFCC features. For example, using 9-dimensional BNFs results in a phone recognition accuracy of only 0.38% lower than that obtained using 39-dimensional MFCCs. Furthermore, when the first and second-order derivatives, Δ and $\Delta\Delta$ are included, the BNF system outperforms the MFCC system by 2.12% while maintaining a smaller feature dimension.

Feature Representation	Dim	% Corr	% Acc
MFCC $+\Delta + \Delta\Delta$	39	76.23	70.95
Formants	3	49.30	40.71
Formants $+\Delta + \Delta\Delta$	9	56.32	51.12
BNF	3	65.02	60.94
BNF	9	74.37	70.57
BNF $+\Delta + \Delta\Delta$	27	76.77	73.07

TABLE 4.1: Recognition performance of an HMM-based ASR systems utilising formant and bottleneck feature representations as documented in (Bai, 2018).

All experimental results presented in this thesis utilise dataset 2, BNFs derived from the phone discrimination utility task. Preliminary experiments testing the proposed CSHMM models on both dataset 1 and 2 are documented in Table 6.3 in Chapter 6.

4.3 Visual Analysis of Bottleneck Features

In order to further investigate the suitability of using BNFs for the present study, a visual analysis of TIMIT utterances was conducted. The plots of each of the 9-dimensional features with the original TIMIT segmentation labels displayed as vertical lines and phone labels appended to the x-axis are shown in Figures 4.3 - 4.8. The BNFs are in the range [0, 1], to enhance clarity, each dimension is plotted on a separate stacked axis, with each phone label offset for readability and to reduce transcription overlap in regions of small segments.

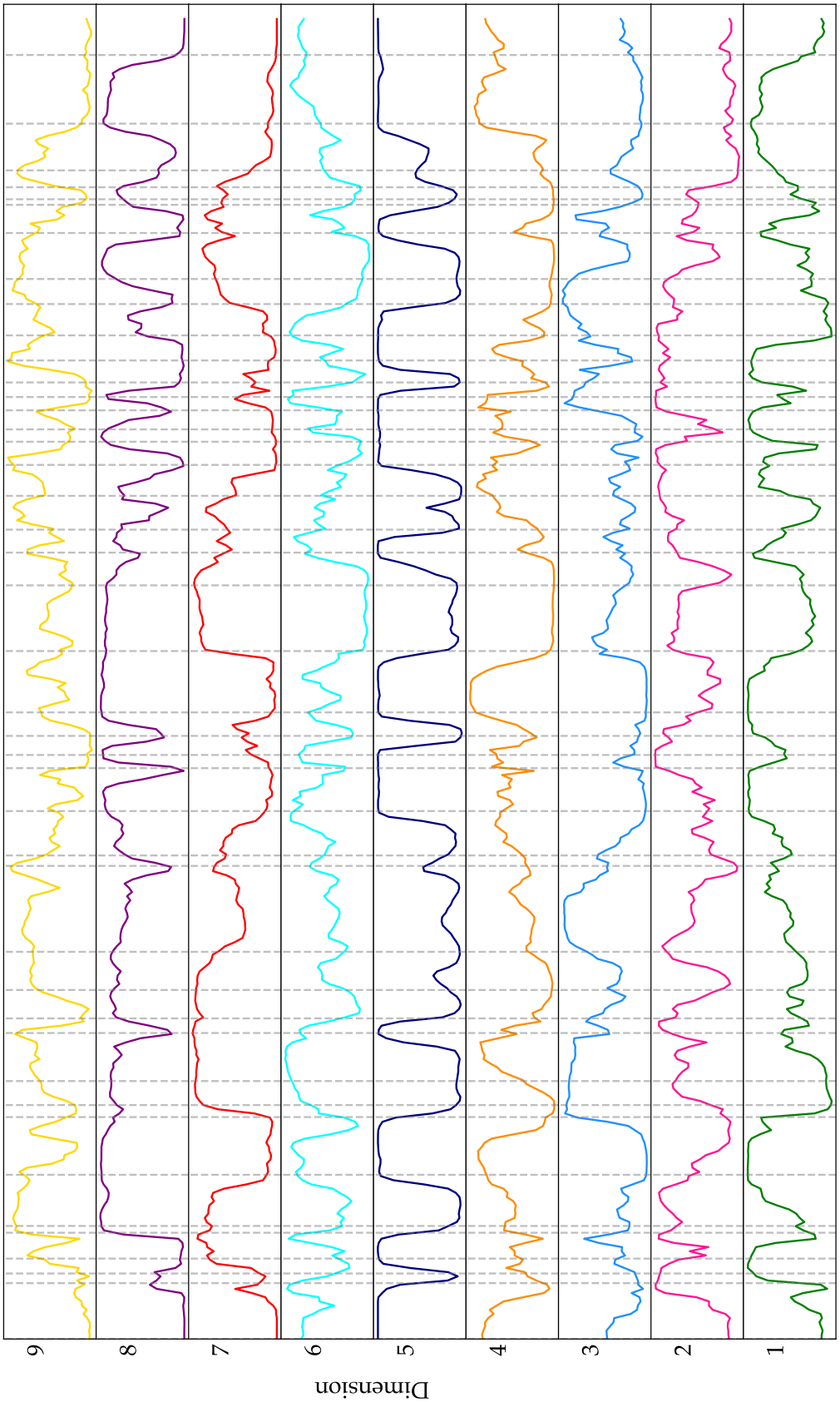


FIGURE 4.3: Visualisation of 9 dimensional BNFs with the TIMIT segmentation (vertical lines) for the utterance: TEST/DR8/MJTC0/SX110 - containing the spoken sentence "The best way to learn is to solve extra problems".

FIGURE 4.4: Visualisation of 9 dimensional BNFs with the TIMIT segmentation (vertical lines) for the utterance: TEST/DR8/MJLN0/SX9 - containing the spoken sentence "Where were you while we were away".

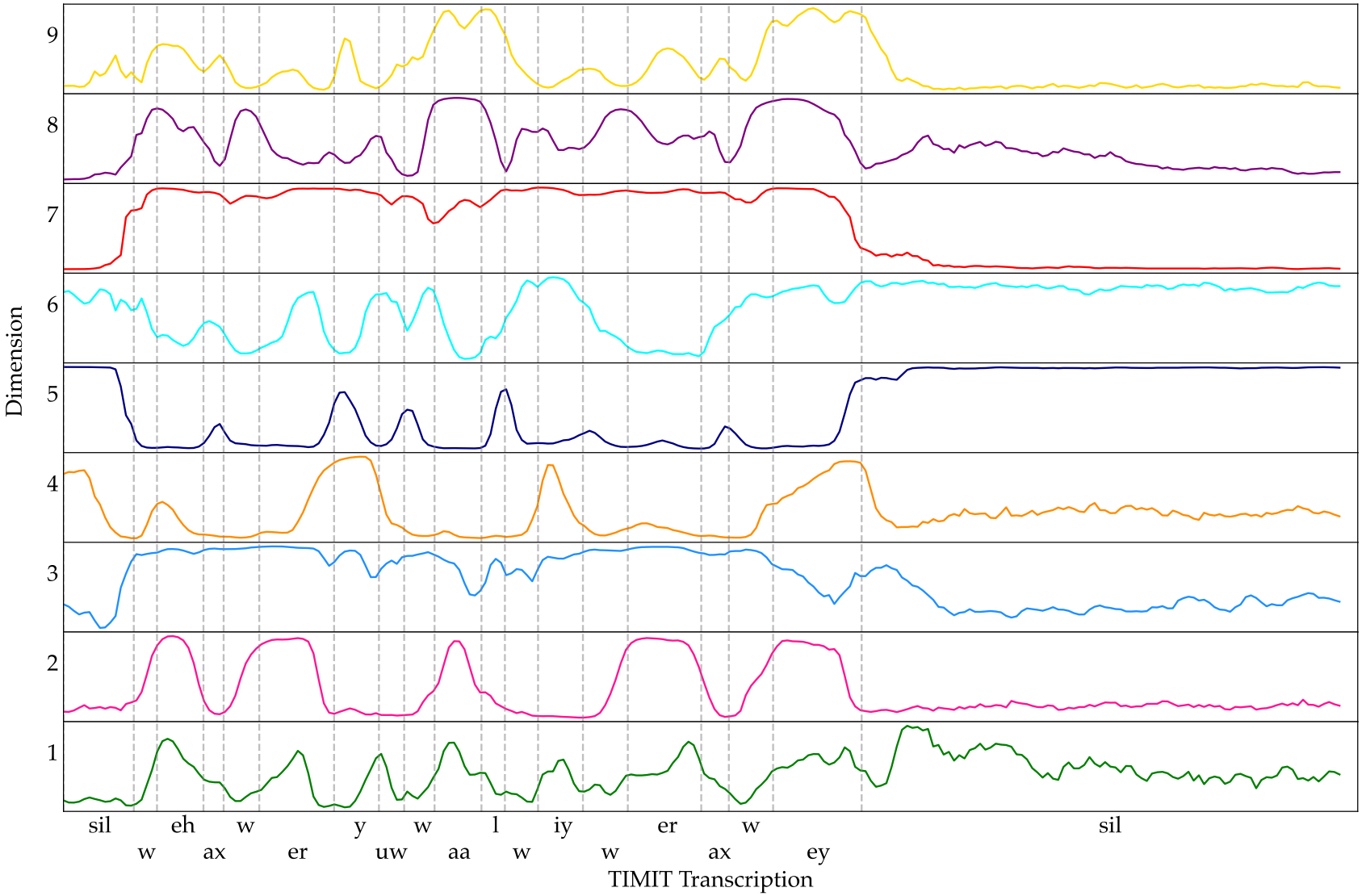


FIGURE 4.5: Visualisation of 9 dimensional BNFs with the TIMIT segmentation (vertical lines) for the utterance: TRAIN/DR6/MRMB0/SX231 - containing the spoken sentence "I honor my mom".

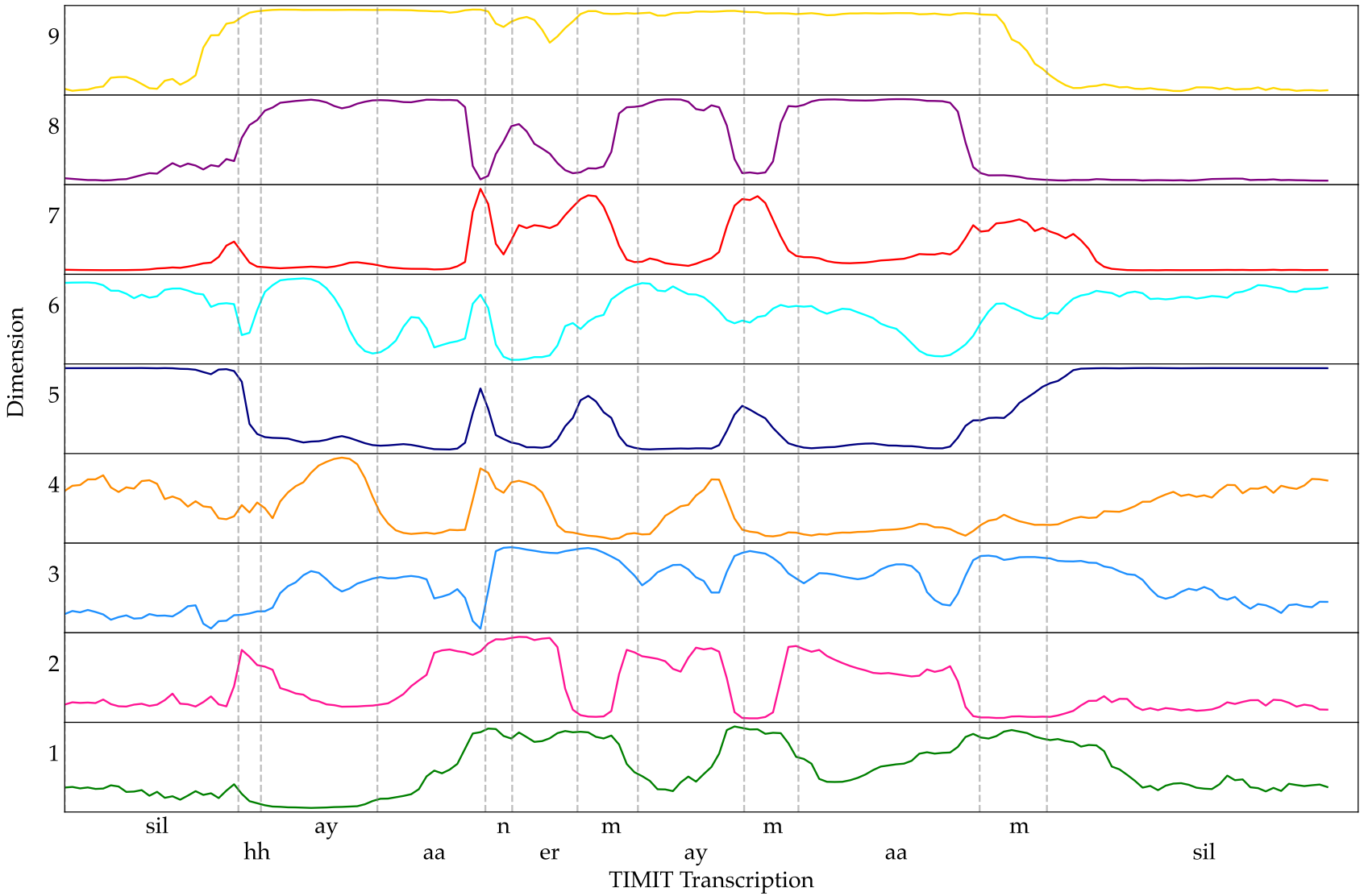
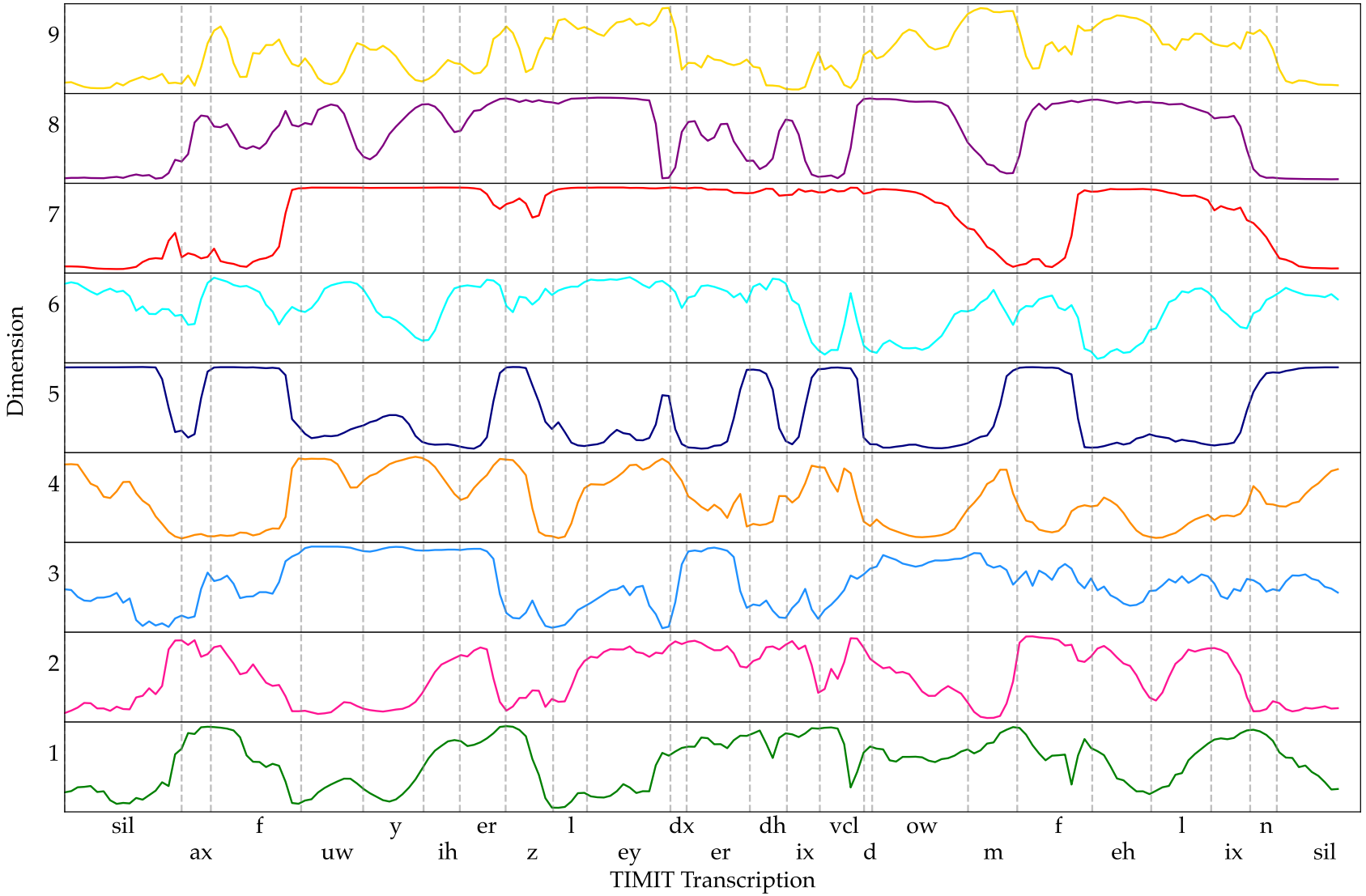


FIGURE 4.6: Visualisation of 9 dimensional BNFs with the TIMIT segmentation (vertical lines) for the utterance: TRAIN/DR2/MWEW0/SI731 - containing the spoken sentence "A few years later the dome fell in".



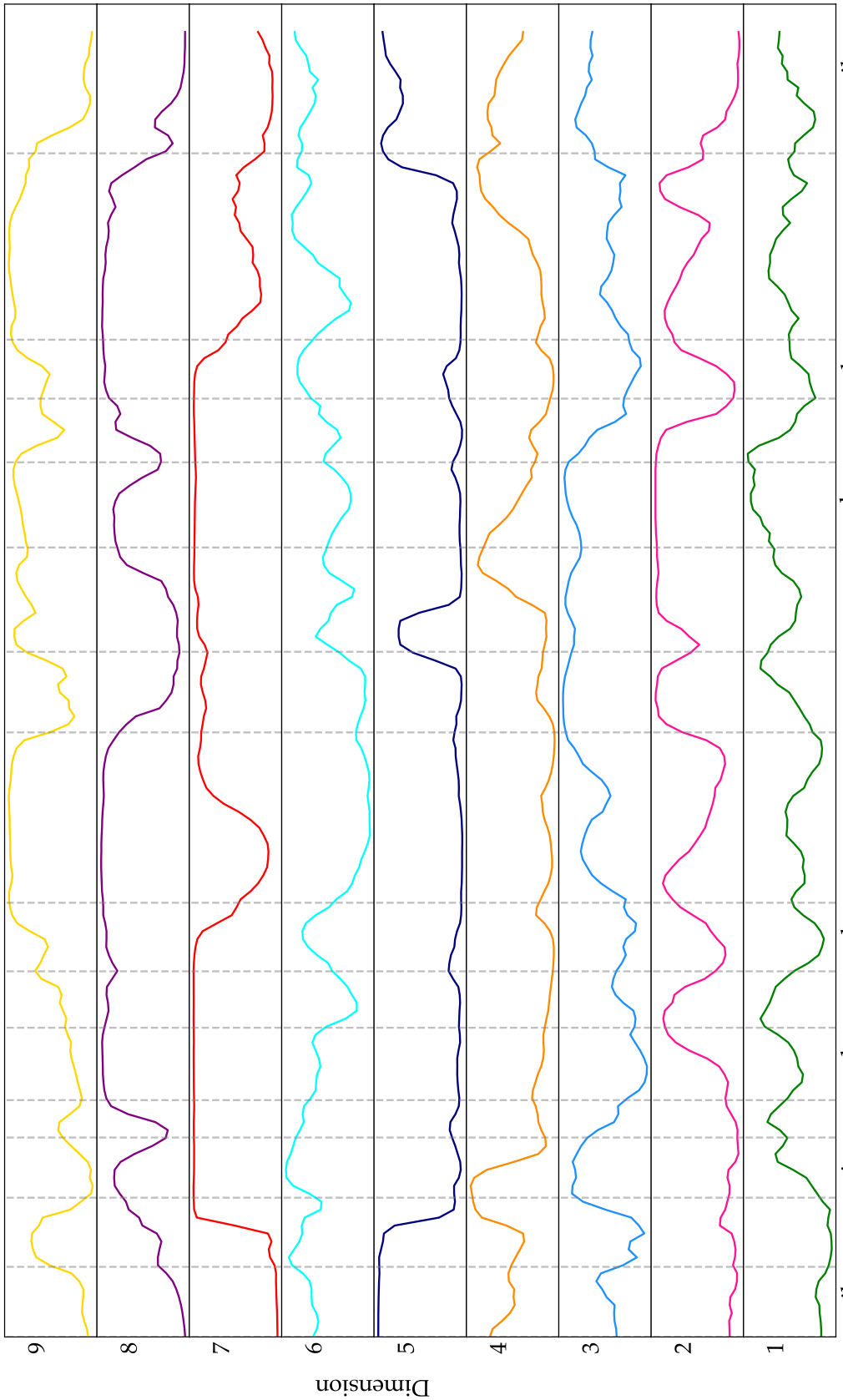
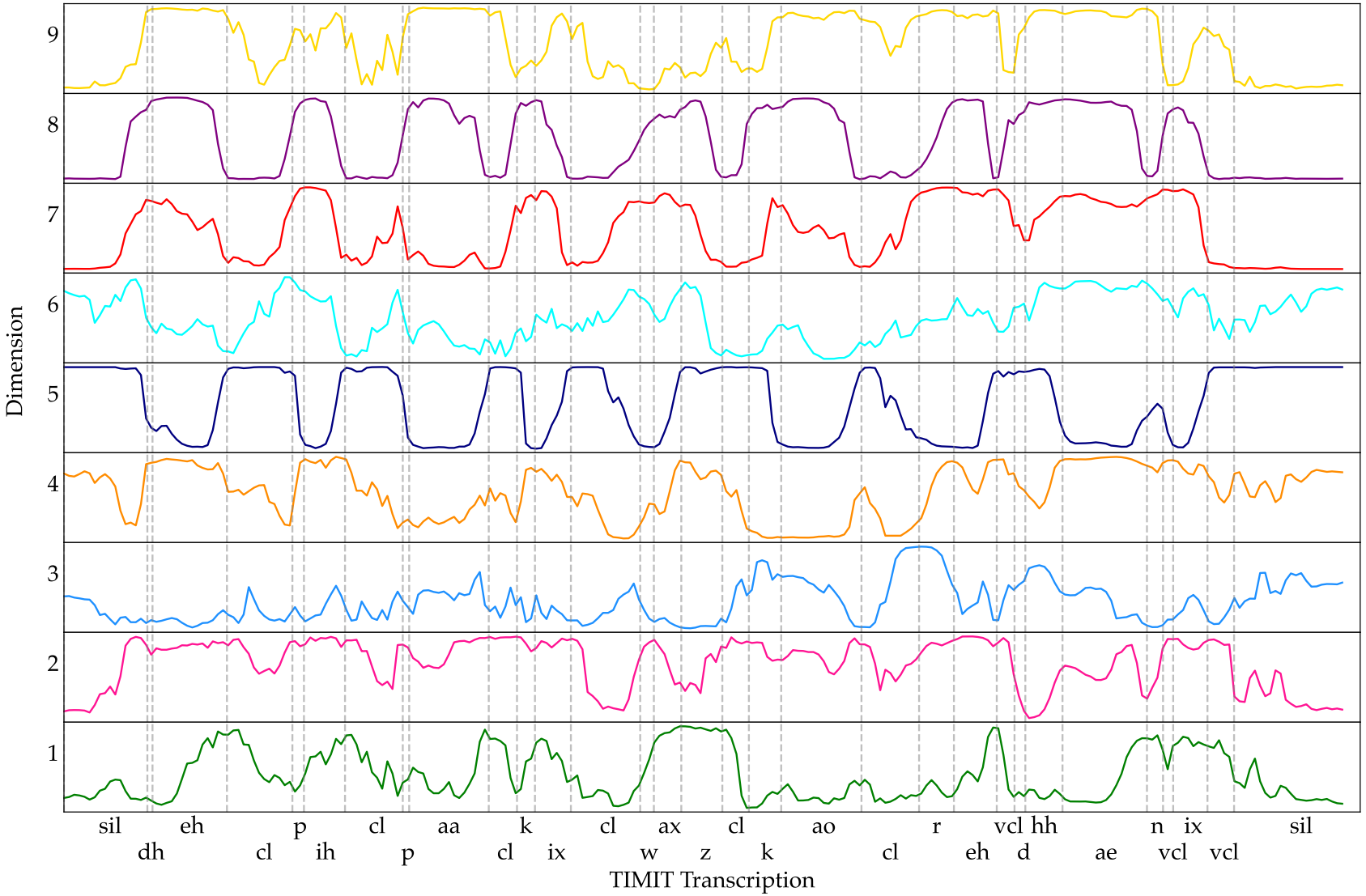


FIGURE 4.7: Visualisation of 9 dimensional BNFs with the TIMIT segmentation (vertical lines) for the utterance: TEST/DR2/MWEW0/SX11 - containing the spoken sentence "he will allow a rare lie".

FIGURE 4.8: Visualisation of 9 dimensional BNFs with the TIMIT segmentation (vertical lines) for the utterance: TRAIN/DR2/MDLB0/SX46 - containing the spoken sentence "That pickpocket was caught red-handed".



The first noteworthy characteristic of these trajectories is that each dimension consists of time-varying trajectories that exhibit distinct properties. While there is no definitive interpretation of these trajectories, it is possible to speculate that dimensions 3 and 5 relate to the voicing or overall energy of the signal. Across all Figures 4.3 - 4.8, dimension 5 exhibits regions where the signal remains relatively stationary within a segment with steep transitions between these stationary regions. These peaks correspond to mostly voiceless obstruents, with the exception of the voiced obstruent /z/ phoneme which has the same trajectory characteristic. The production of obstruent phonemes involves creating a significant constriction or obstruction in the airflow through the vocal tract, which can result in high energy and turbulence in the acoustic signal for such sounds. In contrast, the trajectory of dimension 3 appears to exhibit the opposite characteristics of dimension 5, having peaks corresponding to voiced, sonorous regions of the utterance.

Another interesting observation pertains to dimension 2, which demonstrates similar trajectory patterns for categories of sounds. Specifically, the trajectory of dimension 2 exhibits a distinctive downward-sloping pattern for the vowel sounds /ax/,/aw/, and /ay/, while it displays an upward slope for the vowel sound /aa/. The vowel-rich utterances depicted in Figures 4.4, 4.5, and 4.7 show the trajectory to have a very smooth structure compared to the other utterances which contain a more diverse set of phone categories and have many small fluctuations in the trajectory structure. The stable regions of this trajectory also correspond to other vowels, such as /ey/, /ih/, /eh/, /er/, and the semi-vowel /r/, suggesting that this dimension may capture a latent representation associated with the stability of the formant structure.

In each plot, it is noticeable that the silence regions at the beginning and end of the spoken utterance are primarily static, with bottleneck features falling within a similar range. Specifically, dimensions 1, 2, 7, 8, and 9 exhibits a BNF value close to 0, while dimensions 4, 5, and 6 exhibit values closer to 1, while dimension 3 has a mid-range value. The segmentation of silence can be challenging due to various factors associated

with audio quality, particularly when the initial phoneme is a plosive or has attributes of aspiration.

In Figures 4.7 and 4.5, the voiceless glottal fricative sound /hh/ has minimal impact on the trajectory in the initial silence segment, possibly due to the aspiration of this voiceless sound. Aspiration is produced by forcing air through a narrow space between the vocal cords, causing them to vibrate rapidly, accompanied by a puff of air when pronounced. This is in contrast to Figures 4.3, 4.4, and 4.8, which contain initial phonemes that require more complex articulation. Although the /dh/ and /w/ phonemes have different articulations, they are both voiced and sonorous sounds, similar to the opening /ax/ phoneme in Figures 4.5. In these cases, the co-articulatory effect on the opening /sil/ trajectory is more noticeable than in the sentences beginning with the aspirated /hh/.

There has been limited investigation into the visualisation and interpretation of NN structures for speech recognition. However, previous research by (Nagamine et al., 2015) suggests that DNNs can learn phonetic structures from acoustic features and that distinct neural representations coincide with different broad phone classes. Work by (Tan et al., 2015) argues the opposite, stating that DNNs must be stimulated to learn proper phonetic structures. (Bai, 2018) further delves into the interpretation of very-low dimensional features providing a comprehensive analysis that concludes that the bottleneck layer of DNNs efficiently compresses speech signals into a low-dimensional representation that exhibits distinctive phonetic attributes across its dimensions. The findings demonstrate that the bottleneck layer of DNNs captures critical phonetic features in speech signals that are useful for accurately discriminating phonetic units. It is important to highlight that bottleneck features vary depending on the initialisation procedure. However, (Bai et al., 2018) demonstrates that the same trajectory structure of low dimensional bottleneck features is consistent regardless of initialisation. Additionally, (Weber et al., 2016b) shows that low-dimensional BNFs obtained from NNs with different initialisations have minimal impact on the final ASR result.

4.4 Summary

This chapter presents the motivations for using BNFs in this research and reviews BNFs and their applications in speech and audio processing tasks. The review highlights several studies (Bai et al., 2015, Houghton et al., 2015, Weber et al., 2016b, Weber et al., 2016a, Bai et al., 2018) that share a similar motivation with this research. The reviewed studies demonstrate that BNFs can capture both phonetic and articulatory properties.

Additionally, prior research on low dimensional BNFs demonstrate that the features can produce results comparable to a more conventional MFCC representation for ASR, as evidenced by the strong recognition performance presented in Table 6.3. This supports the utility of BNFs as a compressed feature representation and motivates further investigation into their use in CSHMM modelling. The interpretability of BNFs in relation to phonetic and articulatory properties, as illustrated in Figure 4.1, makes them an appealing representation for the acoustic models that this thesis explores. The visual analysis of the data used in this research provides insight into the characteristics and nature of the BNFs, revealing that the extracted features have a smooth trajectory over time and speculates regions of the output features that may relate to acoustic properties of speech.

In the context of the current findings, it is a compelling option to experiment using a compact, low-dimensional BNF representation in the study of CSHMMs, particularly in exploring a parsimonious acoustic modelling system. This approach is consistent with previous studies, such as (Weber et al., 2016b), which have also demonstrated that BNFs are a promising representation for CSHMMs.

Chapter 5

Experiment Preliminary Details

This chapter introduces the TIMIT speech corpus (Garofolo et al., 1993), which is used as the primary data source for all ASR experiments in this thesis. The TIMIT corpus is a widely-used and well-established resource in ASR research, consisting of a diverse set of transcribed speech samples from various speakers with different accents and speaking styles. In addition to describing the TIMIT corpus, this chapter also outlines the standard evaluation metrics used to assess the performance of an ASR system. These metrics, such as word error rate and character error rate, are commonly used in the field to compare the performance of different ASR systems. By thoroughly examining both the TIMIT corpus and the evaluation metrics used in this thesis, this chapter sets the foundation for the experimental results and analysis presented in the subsequent chapters.

5.1 Speech Corpus

The TIMIT (Texas Instruments/Massachusetts Institute of Technology) speech corpus includes roughly 5.4 hours of near-field recorded speech from 630 speakers, with each speaker recording 10 phonetically rich sentences, making a total of 6,300 sentences. The speaker population is composed of 70% males and 30% females and represents a diverse range of 8 major American English dialects. The speech samples include a wide range of phonetic phenomena, making them a valuable resource for training and

testing ASR systems and for studying American English phonetics. The corpus is also phonetically balanced making it suitable for research purposes. The TIMIT database is widely used in speech and linguistics research and is considered a standard benchmark for evaluating ASR systems.

The TIMIT corpus is composed of recordings of read speech that are based on three different text types, which are labelled as (SA) dialectal variant, (SX) phonetically compact, and (SI) phonetically diverse sentences. The distribution of these sentence types within the corpus is detailed in Table 5.1, as cited from the original TIMIT source (Garofolo et al., 1993).

Sentence Type	**# Sentences**	**# Speakers**	**Total**	**# Sentences\Speaker**
SA	2	630	1260	2
SX	450	7	3150	5
SI	1890	1	1890	1
Total	2342		6300	10

TABLE 5.1: Summary of the speech material in TIMIT corpus.

The sentences in this corpus have been designed to ensure that all possible phone pairs are represented, especially phonetic contexts that are rare or particularly challenging. Special attention has been paid to the inclusion of dialectal variations between speakers, as evidenced by the presence of SA sentences. However, for the purposes of this thesis, all SA utterances have been excluded as the identification of dialectal features between speakers is not relevant to this work. The inclusion of these sentences would only serve to increase the complexity of the recognition experiment.

As outlined in the TIMIT documentation, the suggested subdivision of the speech files has been adopted for all experiments in this thesis unless otherwise stated. These divisions split the remaining 2340 sentences (excluding the SA prompts) into the following sets: train, development, core test, and full test sets. Table 5.2 outlines the data division of these sets.

The TIMIT database is a popular resource because it has manually labelled data at the phone level and has been widely used in ASR research. This speech corpus is

Set	Speakers	Utterances	Hours	Tokens
Train	462	3696	3.14	142910
Development	50	400	0.34	15334
Core Test	24	192	0.16	7333
Full Test	168	1344	0.81	51680

TABLE 5.2: Speech statistics for TIMIT sets (training, full test, development, and core test) excluding SA sentences.

particularly useful for understanding how ASR systems behave at the phone level due to the verified word and phonetic segmentations of the data. Consequently TIMIT continues to be a good source for baseline experiments with new techniques including HMM and DNN based research.

5.1.1 Phone Mappings

The TIMIT database utilises a 61-symbol phone set in the manual labelling of its data. However, for recognition experiments, a reduced phone set of 49 symbols is utilised to build models, and a 41-symbol set is used to calculate phone accuracy (Lee and Hon, 1989). This condensed phone mapping is a widely accepted convention for recognition baseline experiments as it eliminates distinctions between very similar sounds, which are not deemed to have a significant impact on ASR errors. The complete mapping between the phone sets and broad phone classes from (Garofolo et al., 1993) are presented in Appendix A.

Phone examples can fall into broad phone categories following the recommendations made by (Halberstadt and Glass, 1998) and (Reynolds and Antoniou, 2003), and are summarised in Table 5.3. The distribution of these categories can be determined by the number of sampling frames in the test and training data as visualised in Figure 5.1. The classification of long and short vowels aligns with the conventional definition of long and short vowels as stated in (Oller, 1973). Figure 5.1 highlights a well-balanced distribution of phones in both the training and testing data, further reinforcing the credibility of the recommended test/train data split of the TIMIT corpus.

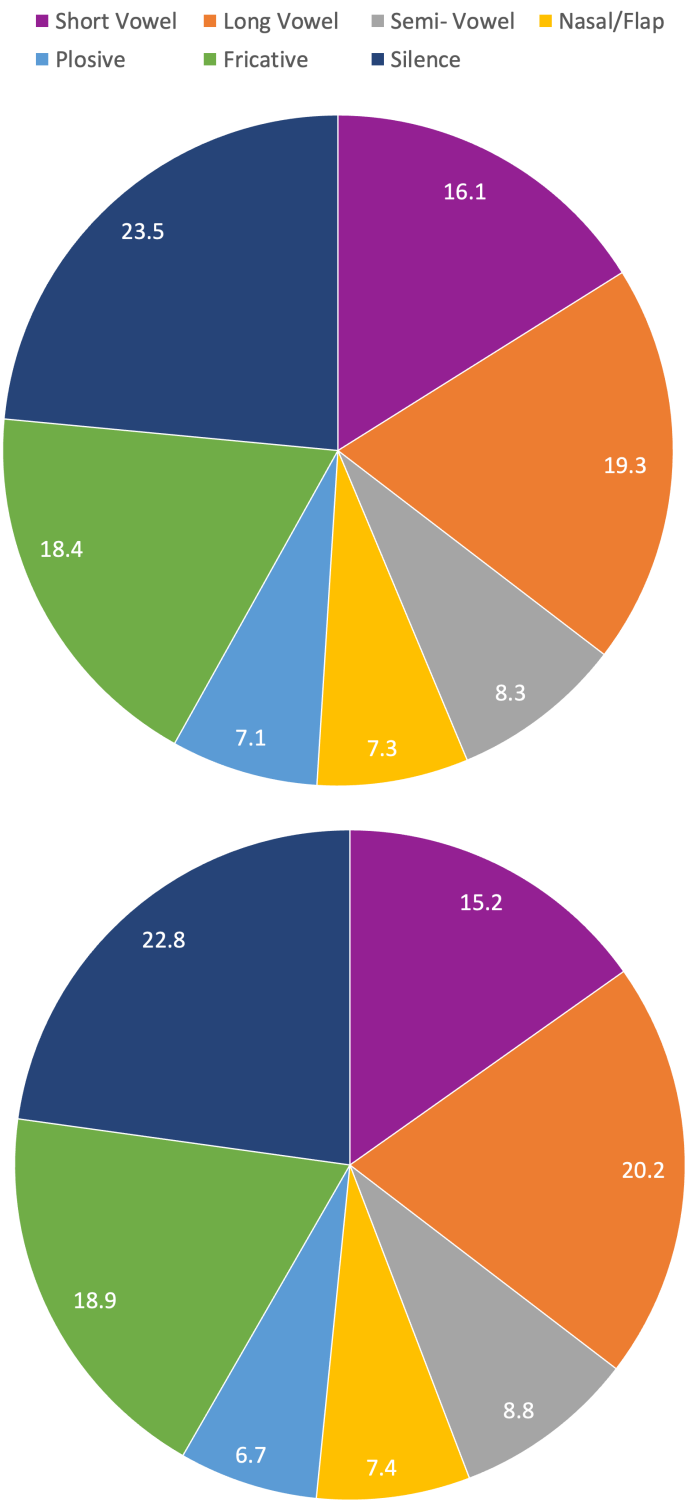


FIGURE 5.1: Distribution of broad phone categories in the recommended training (*top*) and test *(bottom)* dataset split for TIMIT (Garofolo et al., 1993).

Phone Class	TIMIT Labels
Short Vowel	ij ih eh ix ax ah uw uh
Long Vowel	ae aa ao ey ay oy aw ow er
Semi- Vowel	l r el w y
Nasal/Flap	em en eng m n ng nx dx
Plosive	b d g p t k jh ch
Fricative	s z sh zh v f dh th hh hv
Silence	pau epi h# bcl dcl gcl pcl tcl kcl q d

TABLE 5.3: Recommended broad phone categories.

5.2 Evaluation Metrics

The assessment of ASR experiments involves the use of several standard evaluation metrics. Among these metrics is the Word Error Rate (WER) or Phone Error Rate (PER), which compares the recognition hypothesis with the reference transcription. The computation of WER and PER uses a Dynamic Programming-based string alignment method, which quantifies the number of substitutions, insertions, and deletions required to match the reference transcription. Alternatively, word or phone level accuracy may be used to measure the percentage of words or phones that are correctly recognised in the recognition hypothesis. These metrics are provided by the HTK toolkit and are computed using the HResults function, as specified in (Young et al., 2002).

The experiments presented in this work focus on the phone error rate and accuracy rate as a model performance metric which can be calculated as follows:

$$\text{Correct} = \frac{H}{N} \times 100 \tag{5.1}$$

$$\text{Accuracy} = \frac{H - I}{N} \times 100 \tag{5.2}$$

In the calculation of the recognition performance metrics in HResults, the variables **H**, **I**, and **N** are defined as follows: **H** represents the number of tokens that have been correctly recognised, where a token refers to either a phone label when calculating the phone error rate or a word label when calculating the word error rate. **I** represents

the number of insertions that occur when a token is present in the recognition hypothesis but not in the reference transcript. **N** is the total number of recognised tokens. The HResults function also produces two other outputs, deletions and substitutions, which are not directly used to calculate the metrics but provide valuable information. Deletions occur when a token is present in the reference transcription but not in the recognition hypothesis, while substitutions occur when a token in the recognition hypothesis differs from the corresponding token in the reference transcription.

Chapter 6

Continuous State Segmental Models for Speech Recognition

The chapter presents a description of two trajectory-based models: the constant trajectory CSHMM and the linear trajectory CSHMM. The CSHMM framework is introduced in Section 3.5 as a Markovian-based model that models speech units as constant dwell regions of variable durations. Smooth transitions connect one dwell to the next, as shown in Figure 3.7. The CSHMM has been explored in other domains using naming conventions such as Linear Gauss Models and Kalman Filters. However, there has been insufficient experimentation applied on speech data. The study builds upon previous work (Champion and Houghton, 2016) by evaluating the CSHMM in a segmental model setting. In addition, it includes details on the training and inference algorithms used to benchmark the proposed models using the TIMIT dataset.

This study evaluates whether a trajectory-based model can capture essential correlations in speech, similar to a Segmental Hidden Markov Model (SHMM) described in Section 3.3. The hypothesis is that by utilising a CSHMM framework, a segmental trajectory-based system can also include flexible constraints at segment boundaries. This would reduce the impact of inter-segmental independence, a known limitation of the SHMM. Additionally, this research highlights the advantages of the CSHMM in addressing the issue of pre-segmentation. The SHMM requires pre-segmentation, which can lead

to computational inefficiencies because all segment durations need to be considered for all state sequences which is unscalable. However, pre-segmentation is no longer necessary by using the training and decoding algorithms implemented for a CSHMM, making the CSHMM a more computationally efficient and effective solution.

As defined in Section 3.5, a CSHMM can be parameterised according to a sequence of states $S = s_1, s_2, ...s_N$, an initial state probability $P(s_1|\Theta_1)$, a transition probability $P(s_j|s_i, \Theta_s)$, and an emission probability $P(y_t|s_t, \Theta_y)$ where Θ represents a d-dimensional Gaussian probability distribution of the following form:

$$\Theta = (2\pi)^{-\frac{d}{2}} |P|^{\frac{1}{2}} \exp\left(-\frac{1}{2}x^{\mathrm{T}}Px\right) \tag{6.1}$$

To simplify the calculations in this chapter, the normal distribution is defined in terms of a precision, which is the inverse of the covariance, such that $P = \Sigma^{-1}$.

Markov models use hidden variables, known as *states*, to represent the underlying system that generates observable data. The implementation of a CSHMM in this work integrates the concept of a state into the algorithmic definition such that a state is defined by both discrete and continuous components. This representation is largely informed by the CSHMM formulation described in (Champion and Houghton, 2016) and (Houghton et al., 2015), which define a CSHMM state based on the Baum Welch forward probability variable α. The components of a state in this work include:

ϕ_i	A phoneme label of a state s_i.
Φ	The full phonetic history vector for a hypothesis.
h	The amount of time spent in the current state.
μ_ϕ	A phoneme-specific canonical mean of a Gaussian realisation distribution.
E	A measurement precision matrix.
A	A realisation precision matrix.
K_t	A state normalisation factor that is to be interpreted as a *score* to rank the systems hypotheses.

6.1 Constant Trajectory CSHMM

The constant trajectory model is a generalised variant of the static Gaussian segmental Model as described in (Russell, 1993). This model aligns with the dwell trajectory concept in the CSHMM (Champion and Houghton, 2016) and assumes that the underlying trajectory remains constant over time for segments of variable lengths.

In this model, the trajectory is regarded as a constant with a state value (s) drawn from a Gaussian distribution with a phoneme-specific mean μ_ϕ and precision A:

$$s \sim \mathcal{N}(\mu_\phi, A) \tag{6.2}$$

The parameters of the canonical distribution for each phoneme in the inventory are estimated during the training process. The observations y_t are assumed to be distributed around the constant trajectory according to a Gaussian pdf:

$$y \sim \mathcal{N}(y_t - s, E) \tag{6.3}$$

Where E is the measurement precision.

A conceptual diagram of the underlying model assumption for the constant trajectory model (PCTM) is depicted in Figure 6.1. The x-axis represents time with black dotted vertical lines signalling segment boundaries, crosses represent observations. The first segment displays two competing hypotheses for two distinct phonemes ϕ_1 and ϕ_3. Subsequent segments demonstrate an idealised example of a complete hypothesis for a constant trajectory. The canonical trajectories are dashed orange lines, while the realised trajectories are solid blue lines. The realised trajectory deviates from a canonical trajectory according to a realisation distribution depicted for illustrative purposes along the y-axis. The observations are shown to be distributed around the realised targets.

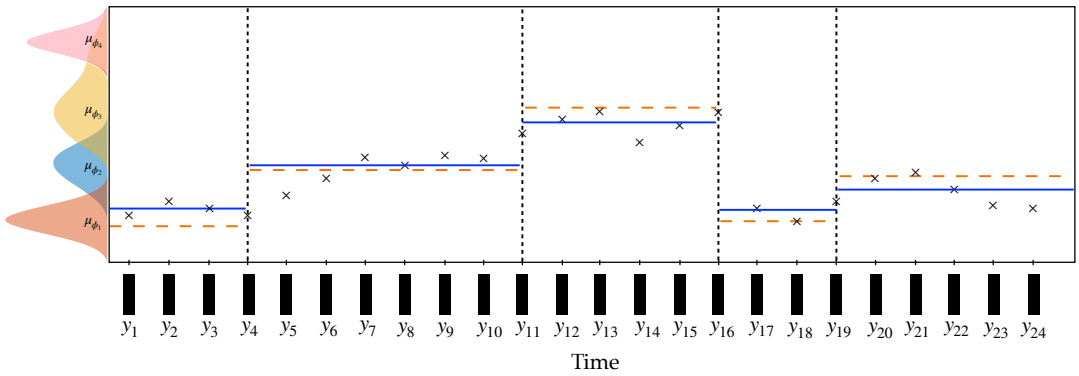


FIGURE 6.1: A conceptual diagram of the Piecewise Constant Trajectory Model. - Black crosses are observation points, the blue line is the realised trajectory. Orange dashed lines show the canonical target from an inventory which is defined by a canonical distribution illustrated along the y-axis.

The trajectory parameters in a static Gaussian segmental model (Russell, 1993) remain fixed for the entire duration of a segment in a single forward pass through the data. In contrast, a constant trajectory CSHMM adjusts the trajectory parameters as new observations are seen. This is possible in a CSHMM because the parameters are probability distributions as opposed to sampled values from a distribution. Figure 6.2 illustrates how a trajectory position may update with each observed data. As each observation is seen, a hypothesised trajectory mean is updated, allowing the underlying trajectory to shift, optimising its position to best fit the observations seen up to time t.

The PCTM shares a similar structure to a standard HMM, assuming piecewise linearity that remains constant over time. However, the two models differ fundamentally in their assumptions about the relationship between states and observations. Whereas a standard HMM assumes a one-to-one mapping between states and observations, the PCTM permits a one-to-many mapping between the state and observations through an explicit duration distribution.

The PCTM is defined by a parametric representation of the Baum-Welch Alpha $\alpha_t(s)$, where s is a vector of continuous state variables at a specific time t. The alpha value corresponds to a forward probability representing the joint probability of a sequence of states and observations. To initialise the model, before any data is observed, the initial

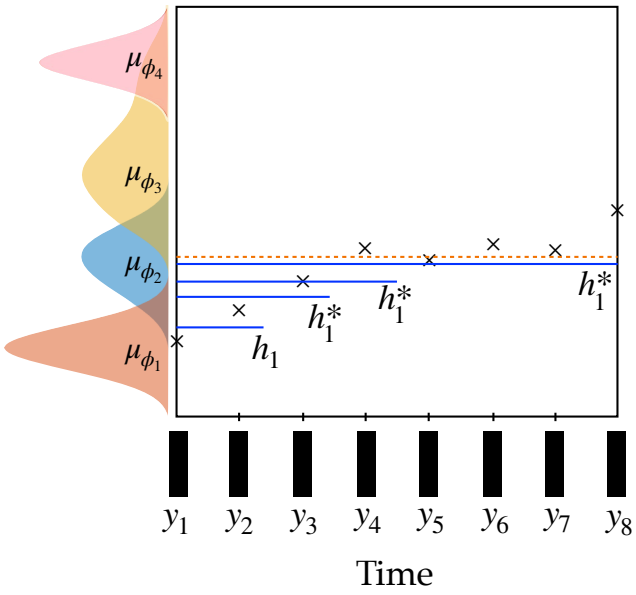


FIGURE 6.2: A conceptual diagram showing a hypothesised trajectory update as new features are observed in a single segment. - Black crosses are observations, the blue line is the realised trajectory, and the orange dashed line is the canonical target trajectory.

state s in the Markov process has a probability of $\alpha_0^{\phi_i}(s)$ for a specific state value. For example, in the context of acoustic modelling let ϕ_i represent a particular phoneme from a set of M phonemes in an inventory. This initialisation means that before any data is observed, M hypotheses are generated for the M phonemes in the phoneme inventory.

1. Initialisation

$$\begin{aligned} \alpha_0^{\phi_i}(s) &= P(s)\ P(\phi_i) \\ &= \mathcal{N}(s - \mu_{\phi_i}, A_{\phi_i})\ P(\phi_i) \end{aligned} \tag{6.4}$$

The probability of a particular phoneme ϕ_i can be determined by a language model, denoted as $P(\phi_i)$. The probability of a state is defined as a probability density function centered on an estimated canonical mean μ_{ϕ_i} with a precision A that can be learned from data. There is no slope dependency in a PCTM, and so a simple Gaussian distribution

of a state is formulated as:

$$\begin{aligned} s &\sim \mathcal{N}(\mu, A) \\ &= (2\pi)^{-\frac{d}{2}} |A|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\mu^{\mathrm{T}} A\, \mu\right) \end{aligned} \tag{6.5}$$

As data is observed, the alpha-value parameters can be updated by extending a hypothesis in its current state or through a hypothesis extension involving a state change. The generalised alpha-recursion is established by induction.

2. Induction

$$\begin{aligned} \alpha_t^{\phi}(s) &= P(y_t|s_t)\; P(s_1, \ldots, s_{t-1}, y_1, y_2, \ldots, y_{t-1}) \\ &= P(y_t|s_t)\; \alpha_{t-1}^{\phi}(s) \\ &= \mathcal{N}(y_t - s_t, E_t)\; \alpha_{t-1}^{\phi}(s) \\ &= K_t^*\, \mathcal{N}(s - \mu_t^*, A_t^*) \end{aligned} \tag{6.6}$$

In the case where a hypothesis is extended in its current state, the alpha-value can be calculated as:

$$\alpha_{t+1}^{\phi}(s) = K_t\, \mathcal{N}(s - \mu_t, A_t)\, \mathcal{N}(y_{t+1} - s_{t+1}, E) \tag{6.7}$$

$$= K_{t+1}^*\, \mathcal{N}(s - \mu_{t+1}^*, A_{t+1}^*) \tag{6.8}$$

Where K_t is a Gaussian normalisation scale factor which can be loosely interpreted as the sum of probabilities of all paths consistent with a given hypothesis. Parameters K_t^*, μ_t^*, and A_t^* represent the updated normalisation factor, state distribution mean, and

state precision, respectively. They can be calculated by the following update equations:

$$A_t = A_{t-1} + E \tag{6.9}$$

$$\mu_t = A_t^{-1}(A_{t-1}\ \mu_{t-1} + Ey_t) \tag{6.10}$$

$$K_t = K_{t-1}\ \mathcal{N}\left(y_t - \mu_{t-1},\ (A_{t-1}^{-1} + E^{-1})^{-1}\right) \tag{6.11}$$

These update equations have been derived according to a Gaussian refactorisation lemma (Appendix B) which specifies that the product of two Gaussian distributions results in another unnormalised scaled Gaussian. Expanding all terms of a Gaussian product and completing the square results in the parameter update formulas. Update equations of the same form have been derived in (Weber et al., 2014) and (Ainsleigh et al., 2002). Additionally, the scaled Gaussian updates are equivalent to the Kalman-Filter time update, Kalman gain, and Kalman Filter measurement updates as demonstrated in (Ainsleigh et al., 2002).

After updating the parameters of the alpha-value on seeing a new observation, it is necessary also to factor in two additional probabilities; the prior probability of a phoneme $P(\phi)$ and a transition probability specifying the hypothesis extension for the same state $P(s_{t+1} = \phi_i | s_t = \phi_i)$. Alternatively, a hypothesis extension can branch to consider a different state at time $t+1$, in which case equation 6.8 is updated such that:

$$\alpha_{t+1}^{\phi}(s) = K_{t+1}\ \mathcal{N}(s - \mu_{t+1}, A_{t+1})\mathcal{N}(s - \mu_{\phi_j}, A_{\phi_j})\ P(\phi_j)\ P(s_{t+1}^{\phi_j} | s_t^{\phi_i}) \tag{6.12}$$

After processing all observations in a single forward pass through the data and extending all hypotheses to cover all possible state sequences, the hypothesis with the highest score is considered the most probable pathway through the data. The Gaussian scale factor K_t serves as the hypothesis score and is employed to rank, prune, and threshold the list of hypotheses. One of the discrete components of the PCTM is the phonetic history vector, which can be extracted at the end of an utterance to reveal the most likely sequence of phonemes that is most pertinent for an ASR evaluation.

6.2 Linear Trajectory CSHMM

The previous section introduced a piecewise constant model that models speech according to a constant trajectory constraint. The PCTM has the same limitations as a standard HMM by assuming that a piecewise constant underlying trajectory is an accurate assumption to capture the dynamics of speech production. To address this limitation, this work evaluates a piecewise linear CSHMM. Two model forms are considered: the discontinuous piecewise linear trajectory model (DPLTM) and the continuous piecewise linear trajectory model (CPLTM). The critical distinction between these models lies in the way in which the segment trajectory functions during state change-points. Subsequent sections will elaborate on the underlying trajectory assumptions of each model.

6.2.1 Discontinuous Piecewise Linear Trajectory model (DPLTM)

The discontinuous piecewise linear trajectory model (DPLTM) assumes that observations within a segment can be adequately modelled according to a linear trajectory described by a straight line equation $s' = hx + s$ where s' is the trajectory value after h steps through the data. A trajectory start-point is denoted as s, and x is the realised slope. This assumption is similar to the trajectory definition in the linear Gaussian SHMM (Holmes and Russell, 1999) and is structured in accordance with the transition trajectory definition of the CSHMM as detailed in (Champion and Houghton, 2016).

Figure 6.3 presents a conceptual representation of the underlying trajectory in a DPLTM system. The observations are as black crosses, while the canonical trajectory is the orange dashed line, and the realised trajectory is the solid blue line. Dotted vertical lines along the x-axis indicate the segment boundaries. A key detail of the DPLTM is that at segment boundaries, when a state transitions from s_i to s_j, there is a discontinuity at the segment boundary.

The DPLTM is parameterised based on the mean and slope parameters making the state dimension double the size of the corresponding distributions in the piecewise constant

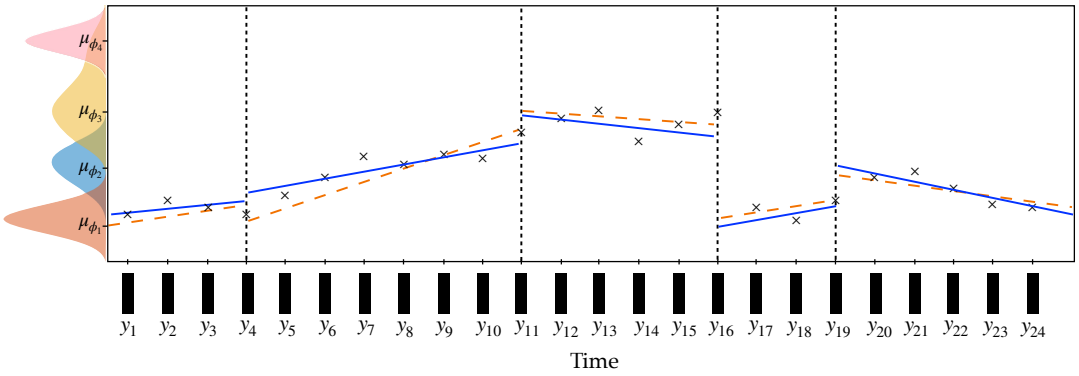


FIGURE 6.3: A visual representation of a Discontinuous Piecewise Linear Trajectory Model (DPLTM). - Black crosses are observations, the blue line is the realised trajectory, and the orange dashed line is the canonical target trajectory.

system. The Gaussian probability density function for the DPLTM is formulated as follows:

$$\mathcal{N}(s, A) = (2\pi)^{-\frac{d}{2}} |A|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\begin{pmatrix}\mu \\ x\end{pmatrix}^{\mathrm{T}} A \begin{pmatrix}\mu \\ x\end{pmatrix}\right), \tag{6.13}$$

Where the slope values x are appended to the state mean vector. The precision matrix A is defined as:

$$A = \begin{pmatrix} A_s + E & E \\ E & A_x + E \end{pmatrix} \tag{6.14}$$

The mean of the trajectory start-point and the slope distribution parameters are μ_s and μ_x, respectively, while A_s and A_x represent the realisation precision for the start-point and the slope. The recursion calculation for the DPLTM can be formalised by the following: **1. Initialisation**

$$\begin{aligned} \alpha_0^{\phi_i}(s) &= P(s)\ P(\phi_i) \\ &= \mathcal{N}\left(\begin{pmatrix} s - \mu_s^{\phi_i} \\ x - \mu_x^{\phi_i} \end{pmatrix}, A\right)\ P(\phi_i) \end{aligned} \tag{6.15}$$

For h steps into a state hypothesis, the alpha recursion is specified as:

2. Induction

$$\begin{aligned}\alpha_t^{\phi}(s) &= \alpha_{t-1}^{\phi}(s_{t-1})\,\mathcal{N}(y_t - s_t, E_t) \\ &= K_{t-1}\,\mathcal{N}\left(\begin{pmatrix} s-\mu_s \\ x-\mu_x \end{pmatrix}, A_{t-1}\right)\,\mathcal{N}(y_t - (s+hx), E) \\ &= K_t^*\,\mathcal{N}\left(\begin{pmatrix} s-\mu_s^* \\ x-\mu_x^* \end{pmatrix}, A_t^*\right) \end{aligned} \tag{6.16}$$

The updated components of the alpha-value identified by (*) can be computed as follows:

$$A_t = A_{t-1} + \begin{pmatrix} E & hE \\ hE & h^2E \end{pmatrix}, \tag{6.17}$$

$$\mu_t = A_t^{-1}\left(A_{t-1}\,\mu_{t-1} + \begin{pmatrix} Ey_t \\ hEy_t \end{pmatrix}\right), \tag{6.18}$$

$$K_t = K_{t-1}\sqrt{\frac{|A_{t-1}||E|}{|A_T|(2\pi)^3}} \times \exp\left(-\frac{1}{2}(\mu_t^{\mathrm{T}} A_t \mu_t - \mu_{t-1}^{\mathrm{T}} A_{t-1}\mu_{t-1} - y_t^{\mathrm{T}} E y_t)\right). \tag{6.19}$$

A hypothesis in a DPLTM can be extended in the same manner as described for a constant trajectory model. If a hypothesis is extended in its current state, the alpha-value is updated such that:

$$\alpha_t^{\phi}(s) = \alpha_{t-1}^{\phi}(s_{t-1})\,\mathcal{N}(y_t - s_t, E_t)\;P(\phi_i)\;P(s_t = \phi_i | s_{t-1} = \phi_i) \tag{6.20}$$

For the case where a hypothesis branches to consider a transition to a different state, the alpha-value update assumes the form:

$$\alpha_t^{\phi}(s) = \alpha_{t-1}^{\phi}(s_{t-1})\,\mathcal{N}(y_t - s_t, E_t)\mathcal{N}(s - \mu_{\phi_j}, A_{\phi_j})\;P(\phi_j)\;P(s_t = \phi_j | s_{t-1} = \phi_i) \tag{6.21}$$

The trajectory definition used in the DPLTM shares similarities with the optimal trajectory model proposed in (Holmes, 1997) and the linear Gaussian SHMM explained in

(Gales and Young, 1993). However, there is a difference in the way the forward trajectory update equations are formulated in this study, which defines the trajectory based on its starting point, while in (Holmes, 1997) and (Gales and Young, 1993), it is based on the midpoint. Defining the update equations according to the starting point enables the complete phonetic history of a hypothesis to be output in a single forward pass of the data. This is unlike the midpoint definition, which requires both forward and backward dynamic programming passes during the inference stage. By reformulating this model using the CSHMM framework, it is possible to address the computational inefficiency of the same structured model defined by an SHMM framework.

6.2.2 Continuous Piecewise Linear Trajectory model (CPLTM)

The continuous piecewise linear trajectory model extends the former DPLTM trajectory assumptions by incorporating a strict continuity constraint between consecutive segments. This constraint requires the endpoint of a segment's trajectory to have the same value as the start point of the subsequent segment's trajectory. This research aims to investigate the hypothesis that a system that can capture the correlations throughout a complete speech utterance is a more faithful model of the articulatory speech process. Extending continuity across segment boundaries to implicitly capture the speech production process within the model definition has been considered in previous works such as (Frankel, 2003), (Bridle et al., 1998) and (Champion and Houghton, 2016). Co-articulation is an interesting attribute of speech that could benefit from an improved model capturing the correlation across an entire utterance.

The goal of accurately modelling co-articulation is to enhance the overall accuracy of speech recognition systems. Several previous studies have examined the effects of co-articulation and how phonetic context influences the acoustic realisation of a phoneme. For instance, (Wieworka, 1997) proposed a novel approach to speech recognition using exponential interpolation to model the gradual changes in speech production due to co-articulation. The system utilises a set of HMMs to model each phoneme and dynamically updates the parameters of the HMM through exponential interpolation.

(Deng and Ma, 2000) proposed a statistical approach to modelling the effects of co-articulation using an Extended Kalman Filter algorithm to smooth across segments. The authors referred to this method as "a 'super-segment' model where the correlation structure in the model extends over an entire speech utterance". This model was designed to capture the dynamic changes in the vocal-tract-resonance that occur due to co-articulation and presented promising results when applied to a spontaneous speech database.

This present study continues this line of research and extends the DPLTM by incorporating the continuity constraint through formulating the CPLTM. The joint probability of a sequence of states and observations can be calculated in the same manner as in a DPLTM in Equations 6.15-6.16 , however, an additional step is necessary at a state change-point in the CPLTM. When branching to a new state, the hypothesis probability must be redefined based on the trajectory endpoint by marginalising the slope through integration. The reparameterisation can then be defined as:

$$\alpha'(\hat{s}) = \int_x \alpha_t(s + hx, x)dx \tag{6.22}$$

The trajectory endpoint is defined as $\hat{s} = s + hx$, and $\alpha'(\hat{s})$ represents the sum of probabilities of paths arriving at the realised endpoint target. The notation α' is used to distinguish the re-parameterised alpha value from the $\alpha(s)$ calculation based on the trajectory start-point. By integrating the slopes associated with the linear trajectory, an updated trajectory value can be determined to specify the start of the preceding trajectory. This integration step allows for a continuous trajectory hypothesis to be modelled throughout the entire speech utterance. Figure 6.4 shows an idealised representation of the underlying model trajectories in a CPLTM.

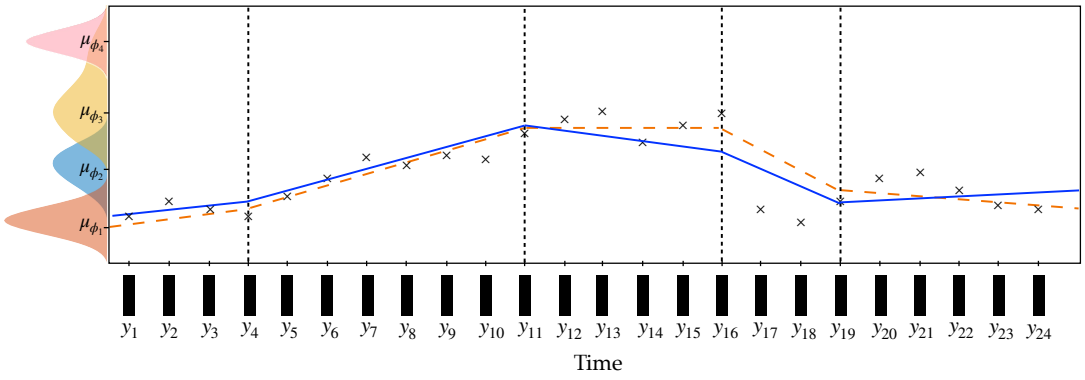


FIGURE 6.4: A visual representation of the underlying model structure of a Continuous Piecewise Linear Trajectory Model (CPLTM). - Black crosses are observations, the blue line is the realised trajectory, and the orange dashed line is the canonical target trajectory.

The following section presents the training and decoding procedures implemented to evaluate the performance of the described piecewise constant trajectory model (PCTM), discontinuous piecewise linear trajectory model (DPLTM), and the continuous piecewise linear trajectory model (CPLTM) using the TIMIT database comparing the results with traditional speech recognition systems.

6.3 Training and Decoding Algorithms

The segmental CSHMM incorporates a forward A* Search algorithm for the training and decoding phases. This section provides an overview of the training and decoding procedures, experimental optimisations, and benchmark ASR outcomes. The ASR results are based on a bottleneck feature representation of the TIMIT dataset, which is described in Chapter 4.

6.3.1 Training Procedure

The CSHMMs used in this study are trained using a supervised approach, with training examples and corresponding class labels provided as input from the TIMIT transcriptions. This training process aims to estimate the set of parameters that maximise the

joint probability of states and observations. The TIMIT database, as described in Chapter 5, was used as the training corpus in these experiments. Its advantage lies in the availability of segmentation labels for the training data.

The Gaussian parameters that describe the state realisations $\mathcal{N}(\mu_s^{\phi}, A_s)$, the slope parameters $\mathcal{N}(\mu_x^{\phi}, A_x)$, are estimated during the training process. The training dataset is also used to estimate a language model, which includes an initial phoneme probability distribution $P(\phi_i)$, and a duration model.

The training data contains M instances of a specific phoneme ϕ_i, where a sequence of data with length h corresponding to that phoneme is denoted as $[y_t, \ldots, y_{t+h}]$. The mean and precision of the canonical realisations are calculated by taking the average and variance of the estimated start points for all M instances of the phoneme in the data:

$$\mu_s^{\phi_i} = \frac{1}{M} \sum_{m=1}^{M} y_t^m \tag{6.23}$$

$$A_s^{\phi_i} = \left(\frac{1}{M} \sum_{m=1}^{M} (y_t^m - \mu_s^{\phi_i})^2 \right)^{-1} \tag{6.24}$$

Similarly, the canonical slope parameters can be estimated as the average mean and precision of a segment slope calculated as:

$$\mu_x^{\phi_i} = \frac{1}{M} \sum_{m=1}^{M} \frac{y_{t+h}^m - y_t^m}{(t+h) - t} \tag{6.25}$$

$$A_x^{\phi_i} = \left(\frac{1}{M} \sum_{m=1}^{M} \left(\left(\frac{y_{t+h}^m - y_t^m}{(t+h) - t} \right) - \mu_x^{\phi_i} \right)^2 \right)^{-1} \tag{6.26}$$

The measurement error can be approximated by calculating the difference between the estimated underlying trajectory of the system and the particular observation or by performing an empirical grid search evaluating the recognition performance for

various parameter values to optimise the system numerically.

Timing Model

A timing model can be introduced in the forward calculations of a CSHMM, and any generalised distribution can be incorporated. An interesting area to consider for future work to build on the findings in this thesis is to specify a timing model to complement the data. The experiments reported in this thesis use a global static uniform distribution, represented as Unif$[t_{min}, t_{max}]$, where t_{min} is the shortest allowable segmentation, and t_{max} is the longest allowable segmentation. While it is possible to consider a phoneme-specific duration model by extracting the maximum and minimum segment lengths for each phoneme category in the training data, this scenario was not explored in this work. Using a static global uniform distribution for the duration model provides a baseline for comparison against systems that employ phoneme-specific duration distributions.

Language Model

The language model is a crucial component in ASR, as it utilises a statistical approach to estimate the probability of a word in a sequence given the previous words. The language model can be incorporated into the CSHMM algorithm as $P(\phi_i)$ and, in theory, can be generalised to any distribution. One widely utilised language model is the N-gram model, which defines the probability of each word given the preceding $N-1$ words as $P(w_i|w_{i-1}, \ldots, w_{i-N+1})$.

In this work, phone recognition experiments were conducted on the TIMIT corpus and evaluated using a bi-gram language model where $N = 2$. The parameters of the bi-gram model were learned from the training data. Initial experiments were performed using a "flat" language model, which assumes equal probabilities for all phonemes following any other phoneme $1/M$, meaning when a hypothesis changes state M, state extensions are considered for each of the phonemes in the inventory. The results of the initial exploratory experiments showed that the bi-gram model performed better than the flat language model.

6.3.2 Viterbi Training

An initial phoneme inventory is constructed based on the assumption that the hand-labelled TIMIT phoneme boundaries accurately segment the data. The CSHMM parameters are estimated according to this segmentation. While the TIMIT labels may be sufficient for a single-state phoneme model parameter estimation, they may not be optimal for tri-phone model training. This work considers both single-state and three-state models with initial benchmark experiments utilising the TIMIT segmentation.

A subsequent experiment implements a Viterbi Alignment method to further optimise the data segmentation and improve the accuracy of parameter estimation. The Viterbi Algorithm, described in Section 3.6.2, is a breadth-first search through a probability-weighted lattice calculating the MAP maximum *a posteriori* and shortest path through the lattice. This algorithm is highly efficient and is useful for simple HMM automatic speech recognition tasks, with a computational cost of $\mathcal{O}(T)$ (Young et al., 2002).

The iterative Viterbi Alignment process aims to find the data's optimal segmentation, which will consequently optimise the parameter estimation. The Viterbi criterion states that when two competing hypotheses exit the same state at the same time with the same phonetic history, the hypothesis with the highest probability will be considered the optimal hypothesis. This criterion reduces the computational load by condensing the hypotheses heap during state transitions, enabling efficient Viterbi training and decoding. When running a Viterbi alignment, the forward algorithm, as described previously for a DPLTM or CPLTM system, is implemented accompanied by a strict language model. The language model only allows transitions between phonemes according to the training data transcription. This forces the model to find an optimal pathway through the data by considering all segmentations for the transcribed sequence of phoneme models. A maximum heap size needed for a full decode to be present on the heap at the end of the forced alignment experiment can be calculated by $N \times t_{max}$ where N is the number of phonemes in the inventory and t_{max} is the maximum allowed duration governed by the timing model.

Experimental results using a development dataset reveal small fluctuations in the CSHMM "score", which can be loosely interpreted as the likelihood probability as the number of iterations increases. Furthermore, after training on the full TIMIT training data, a decoding experiment resulted in an increase in phone error rate (PER) for each additional forced alignment run, as presented in Table 6.1. These findings suggest that while the Viterbi Alignment is an effective method for optimising the segmentation of speech data in a standard HMM, the additional trajectory constraints may impact the benefit of Viterbi Training.

# Iterations	%Corr	%Acc	ins	del	sub
0	66.97	56.48	10.49	7.82	25.21
1	68.57	51.15	17.42	5.25	26.18
2	68.19	49.92	18.27	5.18	26.64
3	67.03	50.59	16.06	5.96	27.01
4	66.92	50.91	16.01	5.97	27.11
5	66.52	50.71	15.95	6.13	27.35

TABLE 6.1: Recognition performance after iterations of the Viterbi alignment procedure for a single state DPLTM.

The number of insertions present in the experiment with no forced alignment training compared to a single iteration of forced alignment shows a 7% increase in insertion errors after alignment. Upon further examination, it was determined that the A* Search Algorithm is susceptible to the same problems faced in the Viterbi SHMM described in (Holmes, 1997). Specifically, the measurement error decreases with each forced alignment iteration and consequently biases the model towards weighting smaller segments as more probable than longer segments. This implicit bias results in an increased number of insertions of phonemes with very small segments.

To mitigate the effects of this implicit bias, a duration-dependent correction scaling factor was employed in conjunction with a system-wide measurement variance floor value of $1e^{-9}$ during the training process. This approach aimed to overcome the limitations of the Viterbi alignment and improve the accuracy of the model.

Visual Analysis of Viterbi Training

To further investigate the recognition performance results in Table 6.1, which shows that for model training, the TIMIT segmentation is optimal, and there are no additional benefits for running a forced alignment Viterbi training procedure. A visual analysis is considered to observe if there are notable trends in how the segmentation alters after running a forced alignment process for both the DPLTM and the CPLTM.

A number of randomly selected TIMIT utterances from the test data are visualised in Figures 6.5 - 6.10. These figures plot the same utterance and segmentation according to the two models of interest, the DPLTM and CPLTM. The top spectrogram shows the segmentation from the DPLTM system with the blue vertical lines against the TIMIT transcribed segmentation indicated by the red lines. The bottom spectrogram shows the segmentation from the CPLTM system, with green vertical lines against the TIMIT segmentation again shown by the red lines. The phoneme transcriptions for the DPLTM, TIMIT, and CPLTM systems are aligned with the segmentation in the spectrograms above and below.

FIGURE 6.5: Spectrogram displaying approximated segmentation for TIMIT utterance: "The oasis was a mirage". - The segmentation corresponds to the TIMIT labels (red) and the approximated segmentation output for the DPLTM (blue) and the CPLTM (green) systems.

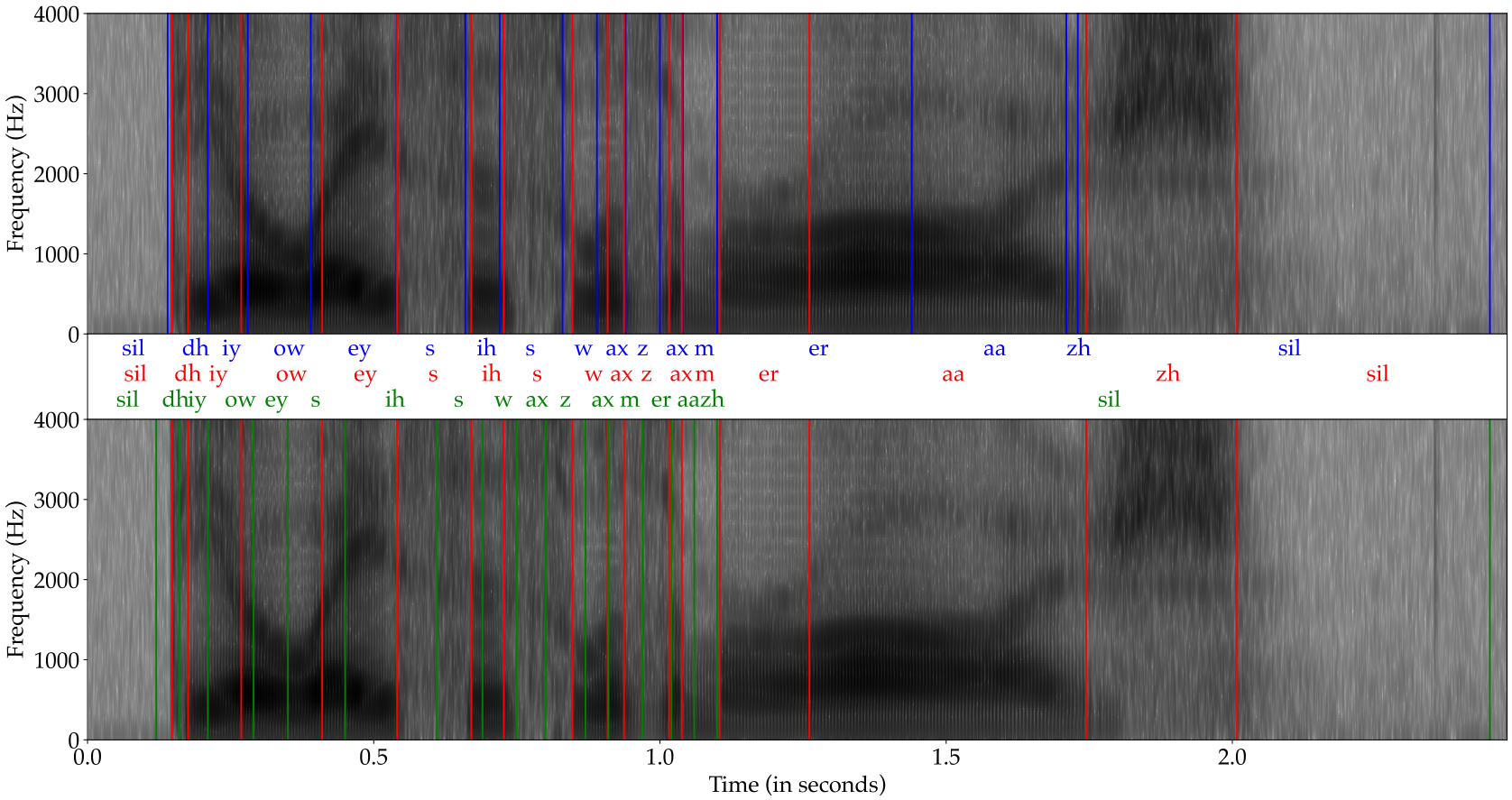


FIGURE 6.6: Spectrogram displaying approximated segmentation for TIMIT utterance: "Elderly people are often excluded".- The segmentation corresponds to the TIMIT labels (red) and the approximated segmentation output for the DPLTM (blue) and the CPLTM (green) systems.

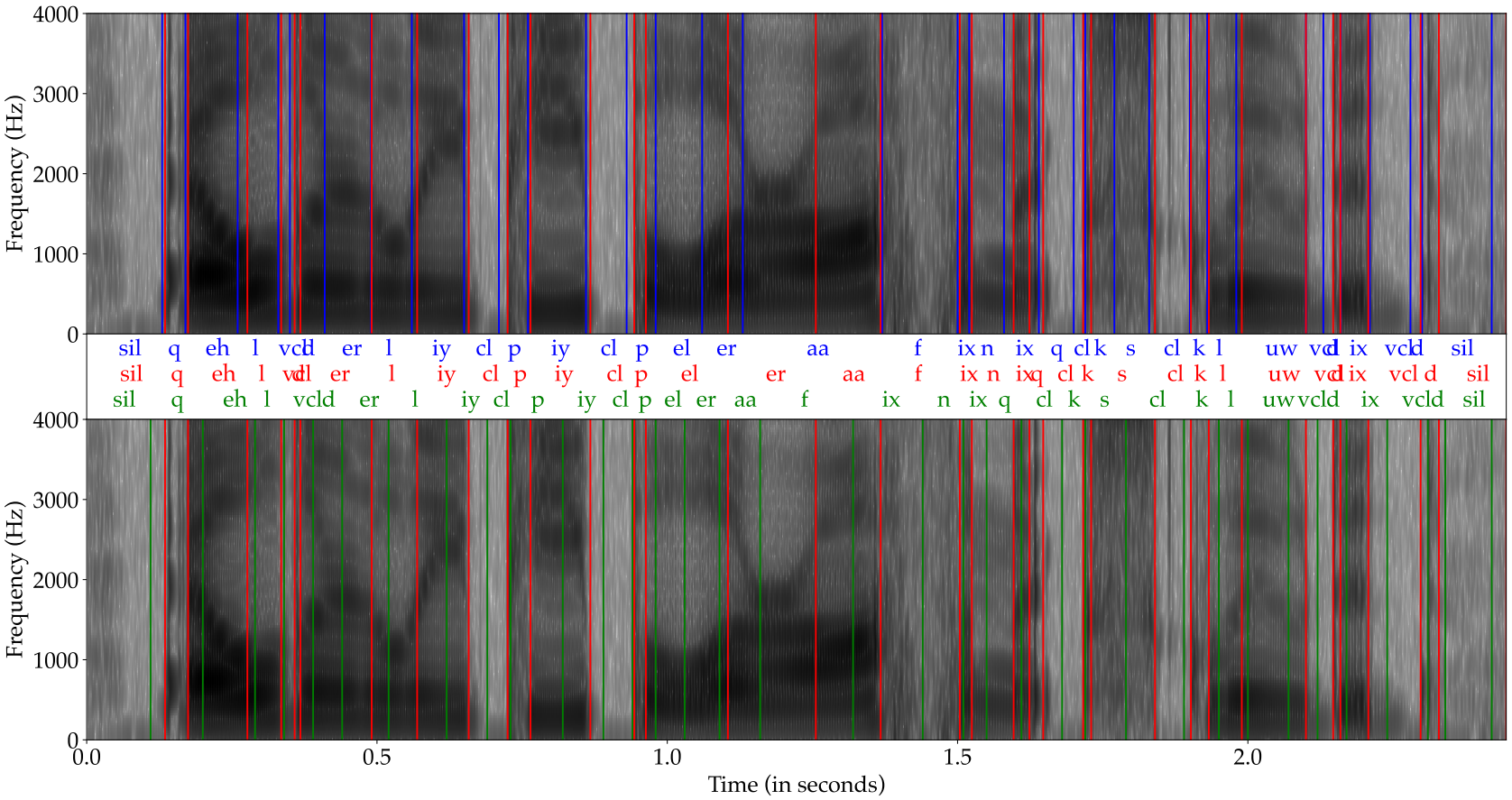


FIGURE 6.7: Spectrogram displaying approximated segmentation for TIMIT utterance: "It provides a frame for the sampling ceremony".- The segmentation corresponds to the TIMIT labels (red) and the approximated segmentation output for the DPLTM (blue) and the CPLTM (green) systems.

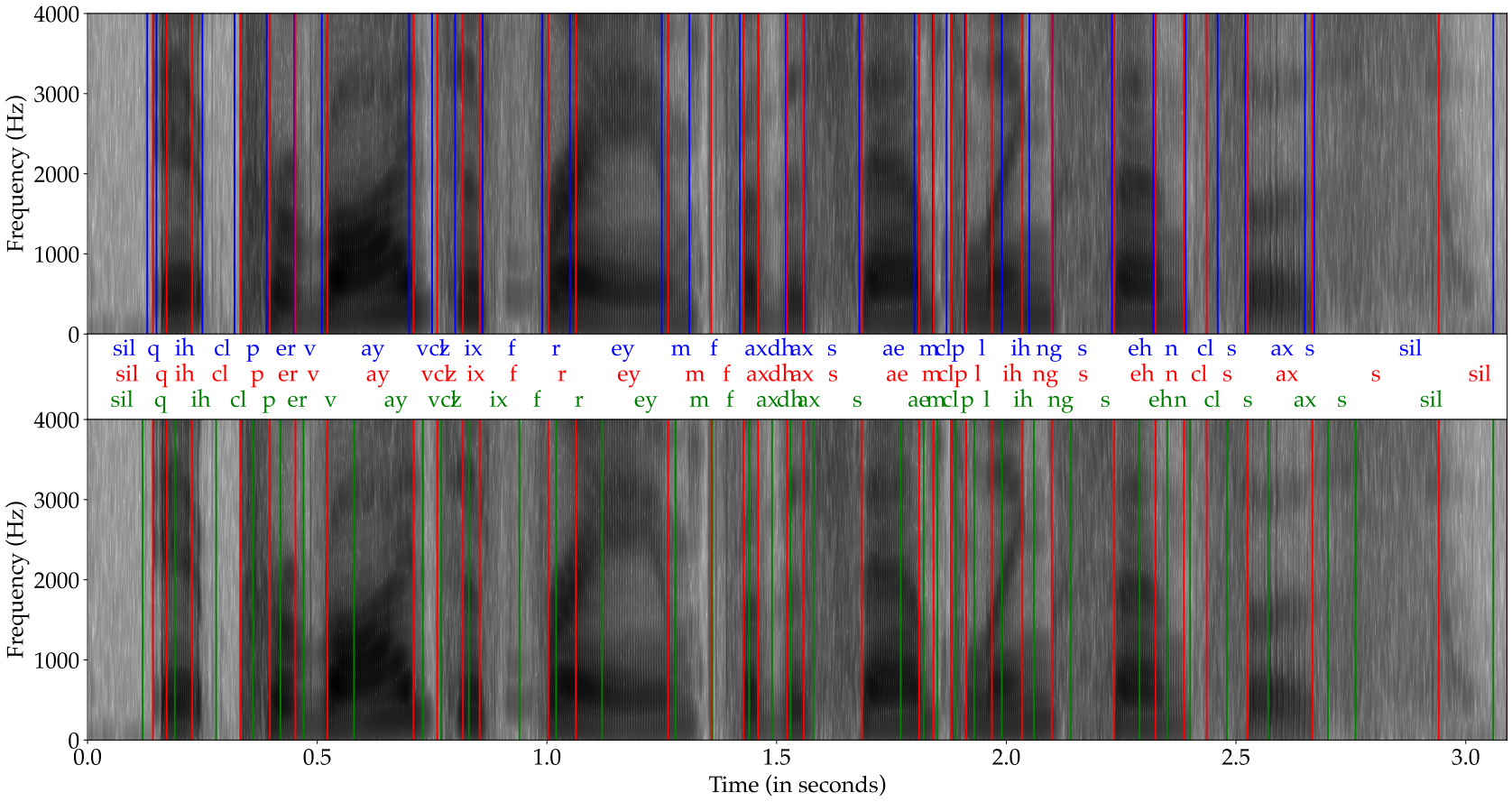


FIGURE 6.8: Spectrogram displaying approximated segmentation for TIMIT utterance: "He will allow a rare lie".- The segmentation corresponds to the TIMIT labels (red) and the approximated segmentation output for the DPLTM (blue) and the CPLTM (green) systems.

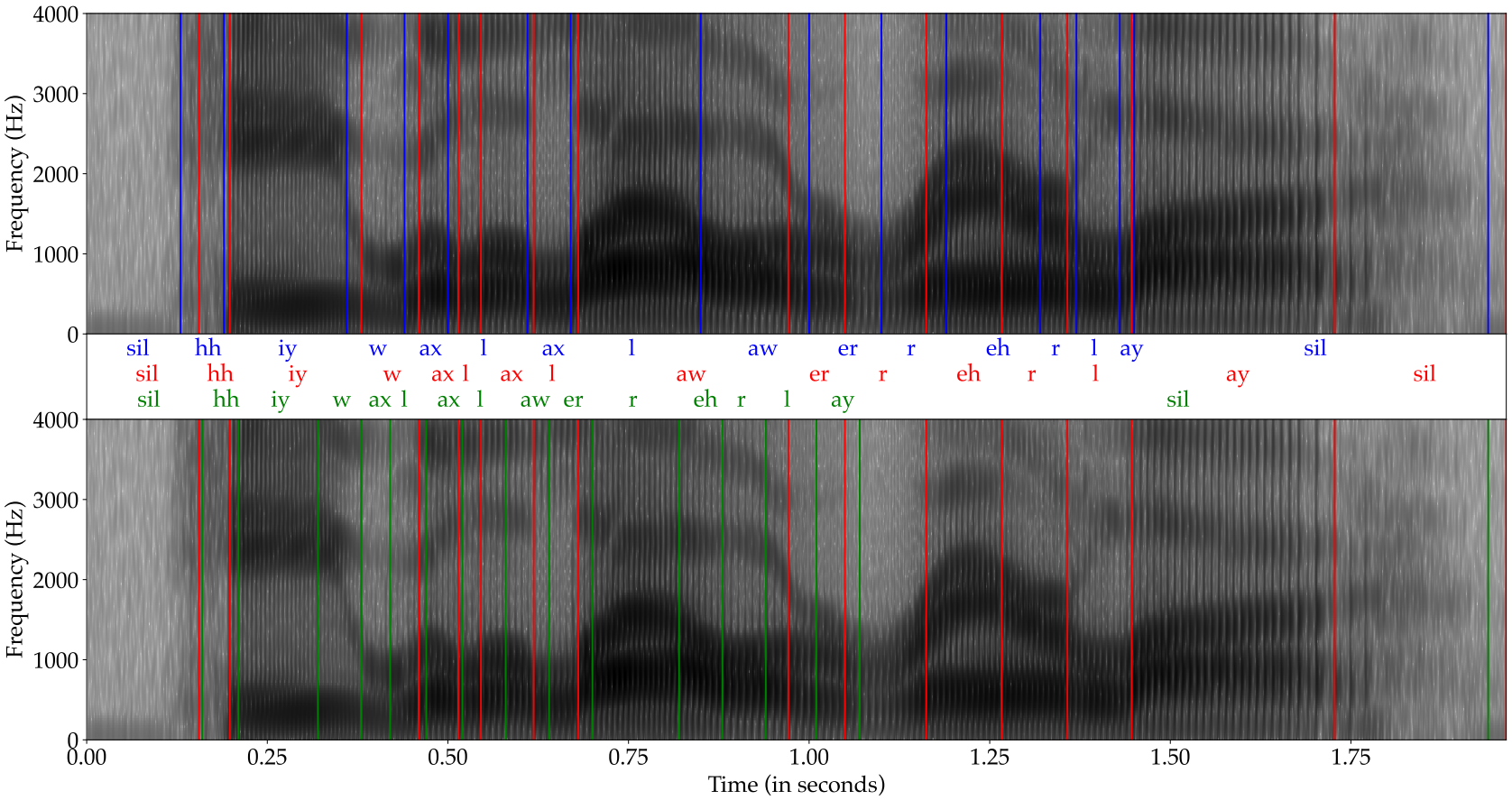


FIGURE 6.9: Spectrogram displaying approximated segmentation for TIMIT utterance: "Only lawyers love millionaires".- The segmentation corresponds to the TIMIT labels (red) and the approximated segmentation output for the DPLTM (blue) and the CPLTM (green) systems.

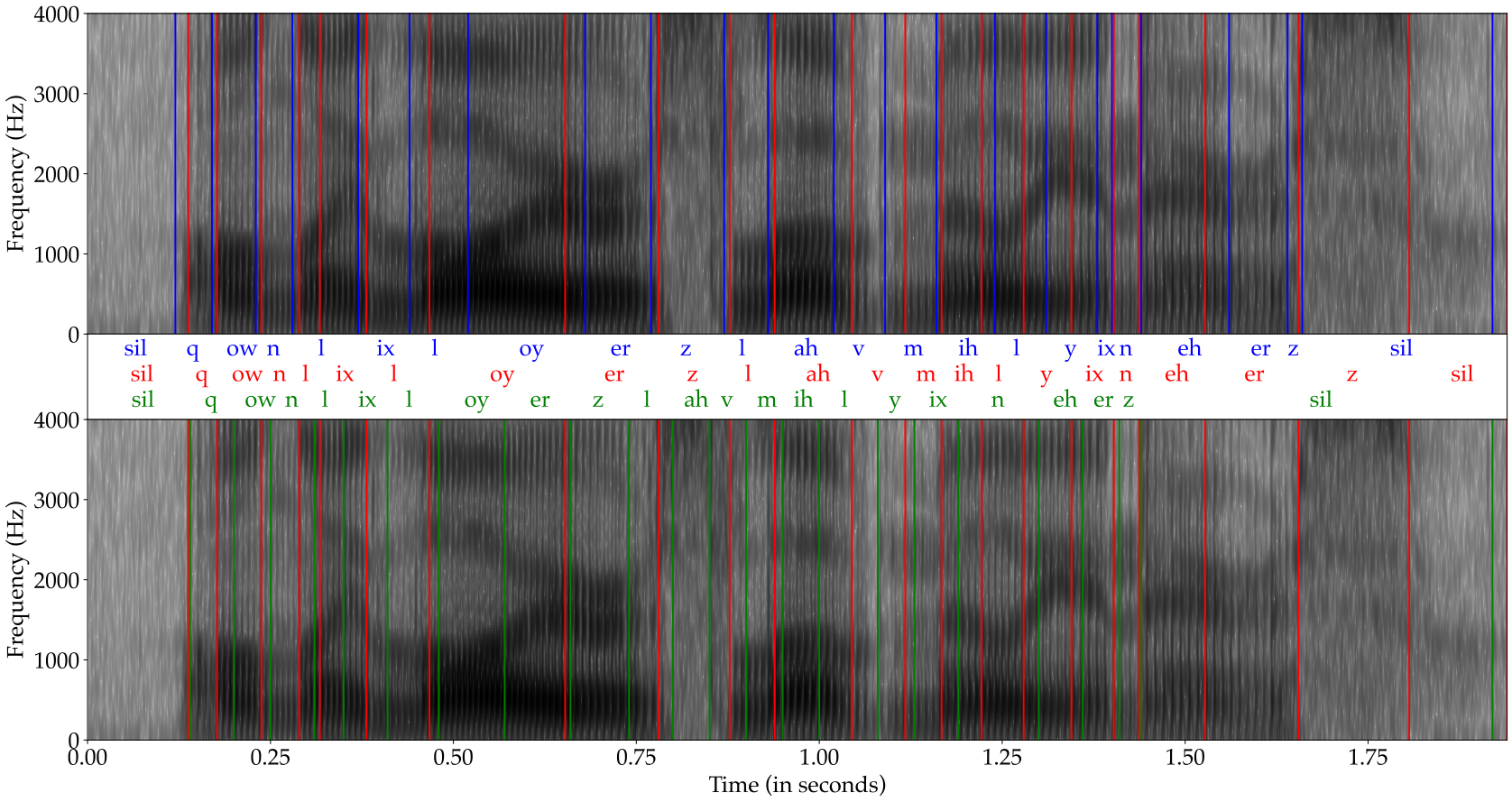
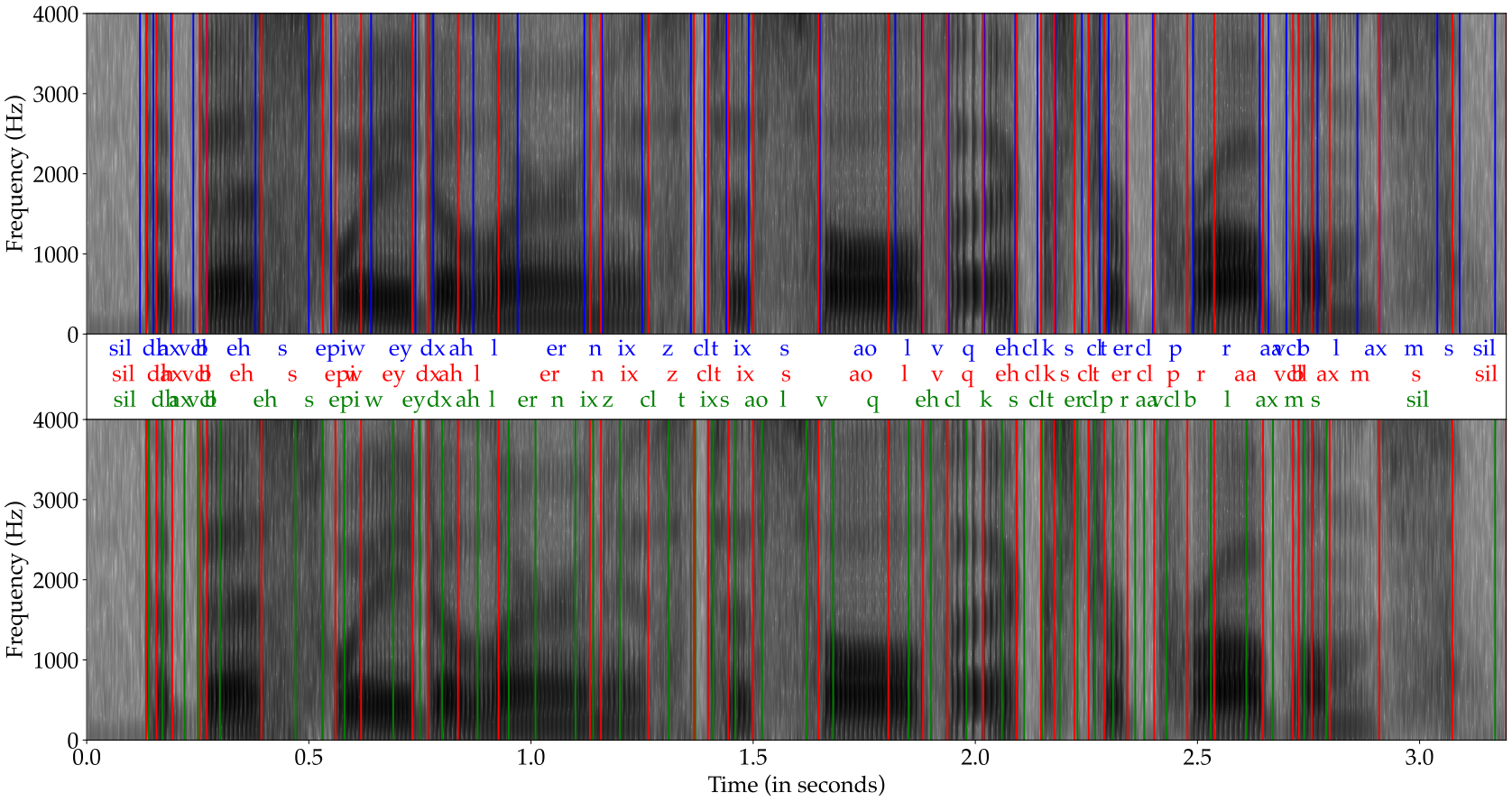


FIGURE 6.10: Spectrogram displaying approximated segmentation for TIMIT utterance: "The best way to learn is to solve extra problems".- The segmentation corresponds to the TIMIT labels (red) and the approximated segmentation output for the DPLTM (blue) and the CPLTM (green) systems.



There are a number of observable trends given an analysis of different utterances. Firstly, the DPLTM system is generally more consistent with the TIMIT segmentation. The DPLTM system was found to frequently hypothesise the same segmentation point as the manually labelled segmentation in TIMIT and where it differs, the system generally hypothesises a change point just before the TIMIT segmentation (to the left). In contrast, the CPLTM system was found to diverge from the TIMIT labelling, with a tendency to underestimate the segment length, resulting in change points to the right of the TIMIT segmentation. This trend was particularly evident in Figure 6.7, which shows that the CPLTM system performs poorly in segmenting the fricative phoneme /s/.

The number of frames by which the DPLTM and CPLTM systems deviate from the TIMIT segmentation varies. When the DPLTM system proposes different segmentation, it is shifted on average by +/- 3 frames. For the CPLTM system, this shift is +/- 7 frames. Furthermore, in the CPLTM system, the shift tends to be consistent throughout the whole utterance, with the deviation never recovering to match the TIMIT segmentations. Another noticeable attribute of the CPLTM is that the system hypothesises a single long segment at the end of the utterance, with all other phone segments comprising of small-medium segments. This is most evident in Figures 6.8, 6.9 and 6.10. In contrast, the DPLTM system was found to often recover, demonstrating greater consistency with the TIMIT segmentation. These findings provide valuable insights into the deviation patterns of the DPLTM and CPLTM systems, highlighting the strengths and weaknesses of each system.

The difference in behaviour observed may be attributed to the influence of the continuity constraint on the segmentation process. The system's efforts to match the start and end points of adjacent segments lead to the probability influence of the continuity constraint, out-weighing the impact of the segmentation. As a result, the realisation probability becomes more influential than the measurement probability in regions surrounding state transitions. Additionally, the decreasing measurement error value may bias the model towards inferring smaller segmentations, reparameterising the

trajectory distribution at the end of a segment results in a quicker convergence of the measurement variance to the floor value.

6.3.3 Decoding Procedure

Decoding is a crucial step in any ASR system, as stated previously, the goal of decoding is to find the most likely sequence of states that generated the observed speech signal. The foundation ASR equation to be solved when decoding is:

$$W^* = \underset{w}{\operatorname{argmax}} \; P(y|\lambda) \; P(\lambda|W) \; P(W) \tag{6.27}$$

Where W^* is the most likely sequence of words given an acoustic model probability $P(y|\lambda)$, a lexical model probability $P(\lambda|W)$, and a language model $P(W)$. The decoding problem can be understood as a search problem, which necessitates an efficient search through the state space considering all possible state sequences. A favourable attribute of the CSHMM framework is that the CSHMM algorithm can be implemented in a computationally efficient manner. A challenge of extending speech research using segmental HMMs is that the Viterbi training and decoding algorithms for SHMMs are computationally expensive. Studies such as (Russell, 2005) investigate optimisation strategies to reduce the computational load of SHMMs, however, a 95% reduction in probability calculations yielded a 3% increase in ASR errors. With the proposal of the CSHMM in the context of ASR in (Champion and Houghton, 2016), which has inspired this work, it is now possible to re-visit the viability of segmental models using the CSHMM and compact computational framework. Despite this, there has been insufficient applied experimentation to this technique on speech data. This may be because of a lack of software available to implement a CSHMM for experimental purposes, such models do not exist as standard in the widely available HTK and Kaldi toolkits. Another complexity of applying a CSHMM to speech data lies in the fact that the feature representation must compliment the model constraints. This

thesis addressed both of these issues by implementing the CSHMM and exploring an alternative speech feature representation for experimentation.

The source code for the DPLTM and CPLTM systems in this work is documented in (Seivwright, 2015). In addition, the decoding algorithms are described here.

The CSHMM algorithm is a sequential branching process that recovers a phoneme sequence and the corresponding times during which each phoneme is realised. It is implemented as an A^* stack decoding strategy that maintains a sorted heap of hypotheses ordered according to the CSHMM K_t "score" parameter. As observations are made, a CSHMM hypothesis can be extended in its current state or branch to a different state. The branching factor is governed by a language model, making it possible to assign zero-probability transitions between unlikely state transitions, this method can regulate the branching factor. However, for the experiments in this work, and general ASR, it is more realistic to allow all phoneme transitions to exist with unlikely phoneme pairs having a low probability rather than a zero probability.

If we assume that all pairwise phoneme transitions are allowable, the CSHMM exhibits an exponential branching factor. Each hypothesis extension yields M alternative hypotheses, where M is the number of phonemes in the global inventory. Maintaining a heap data structure with M^L hypotheses for an observation sequence of length L is computationally challenging. To provide perspective, the average length of a TIMIT utterance in the training dataset is 3.06 seconds, or 306 $10/$ms frames. Using a standard 49 phoneme set would require a heap size of approximately 1.58^{306}, which is intractable. However, there are various pruning and thresholding techniques that can be used to implement this algorithm practically.

- **Viterbi Criterion:** While the Viterbi criterion is not integral to the search at inference time, it can be used to reduce the number of maintained hypotheses. Suppose two competing hypotheses branch to the same phoneme at the same time and also contain the same phonetic transcription with different segmentation. In that case, it is sufficient to assume the highest-scoring hypothesis after branching

will be the highest-scoring hypothesis globally of the two. In such cases, the lower-scoring hypothesis can be dropped from consideration.

- **Threshold Factor:** A threshold factor τ, defined as $logK_t > log\kappa - \tau$ where κ is the largest likelihood, can be used to determine the minimum score a hypothesis must have in order to be considered for further processing. If the difference between the current hypothesis score and the highest-scoring hypothesis on the heap is greater than the threshold then it qualifies to be put onto the heap for further extension. Experiments in this work use a threshold value $\tau = 100$
- **Pruning Factor:** A pruning factor is a control variable that limits the overall size of a heap, consequently reducing the number of elements that need to be processed by the decoder, which improves its efficiency. In this work, a maximum heap size of 250 is maintained.

6.4 Experiments and Results

This section presents the baseline ASR experimental results for variants of the CSHMM using 9-dimensional bottleneck features.

6.4.1 Preliminary Experimental Optimisations

Decoding involves the combination of likelihoods from the acoustic, language and duration models. To balance the contribution of each of these values, a language model scaling factors (LMSF) and phone insertion penalty (IP) can be factored into the decoding computation. In practice the log-likelihood of Equation 6.27 is implemented such that the calculation is of the form:

$$W^* = \underset{w \in \mathcal{L}}{\operatorname{argmax}} \ \log P(y|\lambda) + \log P(\lambda|w) + LMSF \times \log P(w) + M \times IP \qquad (6.28)$$

The LMSF and IP parameters are determined heuristically using a subset of the TIMIT development data. This subset consists of 16 speakers, one male and one female speaker

from each dialect region, each speaking a unique utterance from the development data, resulting in a total of 16 utterances. The development data used for optimising these parameters is excluded from the model training and language model estimation. Tuning these scaling factors resulted in a 4.78% improvement for the DPLTM single-state system and a 2.74% improvement in accuracy for the CPLTM single-state system. Table 6.2 presents the specific parameters used for the baseline systems in this study.

Decoding System	WIP	LMSF
HTK HMM GMM	0	5
1-State		
PCTM	-8	1
DPLTM	-3	0.5
CPLTM	-20	3
3-State		
PCTM	-25	0.5
DPLTM	-7	0.5
CPLTM	0	3

TABLE 6.2: Optimal hyper-parameters for developed systems determined from an empirical grid search.

Bottleneck Feature Dataset

All experiments presented in this work use a bottleneck feature representation (BNF), the details of which are described in Section 4. Two neural network configurations were used to produce the compact feature representation (Bai et al., 2015). Dataset 1, derived from a phone reconstruction experiment, and Dataset 2, derived from a phone discrimination experiment, the specific details of which are described in Section 4.2. To determine which dataset to use for this work, an initial ASR experiment was conducted using both BNF datasets on the TIMIT development test set. Table 6.3 presents the phone recognition results to determine which BNF dataset would be used for further experimentation.

CSHMM Model	Dataset 1		Dataset 2	
	%Corr	%Acc	%Corr	%Acc
PC	67.06	57.00	65.99	60.18
DPLTM	61.66	55.09	63.62	55.52
CPLTM	63.14	52.72	62.88	54.77

TABLE 6.3: Phone recognition results for two BNF datasets as described in Chapter 4 tested on a single state baseline system.

The dataset generated by the phone discrimination network demonstrated consistently superior recognition accuracy across all baseline models tested in this study, the same result was also presented in (Bai, 2018) for a different acoustic model architecture. Although the reason behind the improved accuracy for this set of BNFs applied to an ASR task remains unclear, it can be hypothesised that the utilisation of phone discrimination as a utility may serve as a more intuitive approach for the ASR task, as it requires the encoding of latent features that enhance the distinction between phonemes. In the process of decoding speech utterances, a model's ability to differentiate between phoneme units plays a crucial role in determining its overall performance. While further examination is necessary to validate this hypothesis, the underlying intuition remains compelling. The baseline experimentation in this thesis presents the results obtained from using the BNF Dataset 2.

6.4.2 Baseline TIMIT Experiments for Segmental CSHMM

All experiments in this section present the results of a phone classification utility using the train/test TIMIT divisions as described in Chapter 5. The CSHMM-based models use 9-dimensional bottleneck features as an input speech representation and are compared to a HMM-GMM system that uses a 39-dimensional MFCC feature representation as input, including Δ and $\Delta\Delta$ features corresponding to the time derivative and acceleration, as described in Section 2.2.2.

The segmental CSHMM systems have been optimised through hyper-parameters tuning using a dedicated held-out development dataset. These experiments aim to examine and evaluate the performance of the Segmental CSHMM on the complete TIMIT dataset.

The initial application of a CSHMM to speech was reported in (Weber et al., 2014), and the models in this study employ a variation of the original dwell-transition model, incorporating the CSHMM framework with varying trajectory assumptions. The types of CSHMM systems explored in this work are a piecewise constant system (PCTM), a discontinuous piecewise-linear trajectory model (DPLTM), and a continuous piecewise linear trajectory model (CPLTM). These systems are compared to a conventional HMM-GMM model implemented using a standard HTK recipe. The benchmark results obtained on the widely-adopted TIMIT dataset in the field of automatic speech recognition offer a basis for evaluating the comparative viability of this model against various existing works in the literature.

It is worth noting that this thesis is not focused on producing a state-of-the-art recognition system but on investigating the consequence of implicitly modelling the speech production process and thus addressing the inter-segmental independence assumption. Therefore, we use this simple baseline experiment as a benchmark to compare our other more complex models.

	Model	%Corr	%Acc	*ins*	*del*	*sub*
1 - state	HMM GMM	70.82	62.99	7.69	7.44	21.88
	PCTM	65.99	60.18	5.81	10.52	22.50
	DPLTM	74.76	61.19	9.59	9.93	22.83
	CPLTM	68.83	57.51	5.39	16.89	26.46
3 - state	HMM GMM	66.22	62.72	3.47	12.70	21.10
	DPLTM	68.61	58.41	14.57	5.25	21.77
	CPLTM	66.06	56.22	9.84	7.95	25.99

TABLE 6.4: Performance comparison of three segmental CSHMMs and a standard HMM-GMM model on the TIMIT core test dataset for automatic speech recognition.

The results in Table 6.4 indicate that the PCTM and the HMM system perform similarly, and for simple models, they achieve relatively good recognition results. This reinforces previous findings from (Bai, 2018) that the BNF data is a suitable feature representation

for CSHMM-based systems. Moreover, the results suggest that the PCTM could perform well in an optimised ASR system with more complex language and timing models. The highest cited accuracy of a comparable single state HMM with a single Gaussian Mixture Model (GMM) is 73.80% Correct and 66.08% accuracy on a TIMIT dataset comprising 160 utterances (Lee and Hon, 1989). This benchmark serves as valuable context for interpreting the HMM-GMM implementation results and predicting the accuracy gains of the PCTM if more complex language and timing models are considered. The PCTM and the HMM-GMM system share the same underlying trajectory assumption, which makes for a good baseline system comparison. The PCTM under-performs when compared to the HMM-GMM model by -2.81%, which is satisfactory given the simplicity of the model and reduction in the input feature dimension.

The recognition results for the two piecewise linear systems implemented in this work vary. The DPLTM system is the highest overall performing system, according to the percentage correct metric, it also outperforms the HMM-GMM system, however, the accuracy metric falls just below the HMM-GMM benchmark. The CPLTM system performs worse than the DPLTM and the PCTM in terms of accuracy, which is surprising, and this case holds when considering both a single-state and three-state system.

Using triphone models leads to lower recognition accuracies than monophone models, particularly for very low-dimensional BNFs, which is unusual. This is an interesting result because, in standard HMM-GMM systems, it is typical to observe an improvement in recognition performance when using a triphone model. The discrepancy in the present study may be attributed to the potential loss of contextual phonetic information during neural network training due to the low feature dimensionality of the BNFs. The classification network employed to generate the BNFs involves classifying MFCC vectors as phones based on the TIMIT labelling, which requires mapping all MFCC vectors representing the same phone to the same output. As a result, differences between MFCC vectors that correspond to the same phone are reduced by the bottleneck layer. This is one possibility as to why the same trend is observed across several decoding

systems.

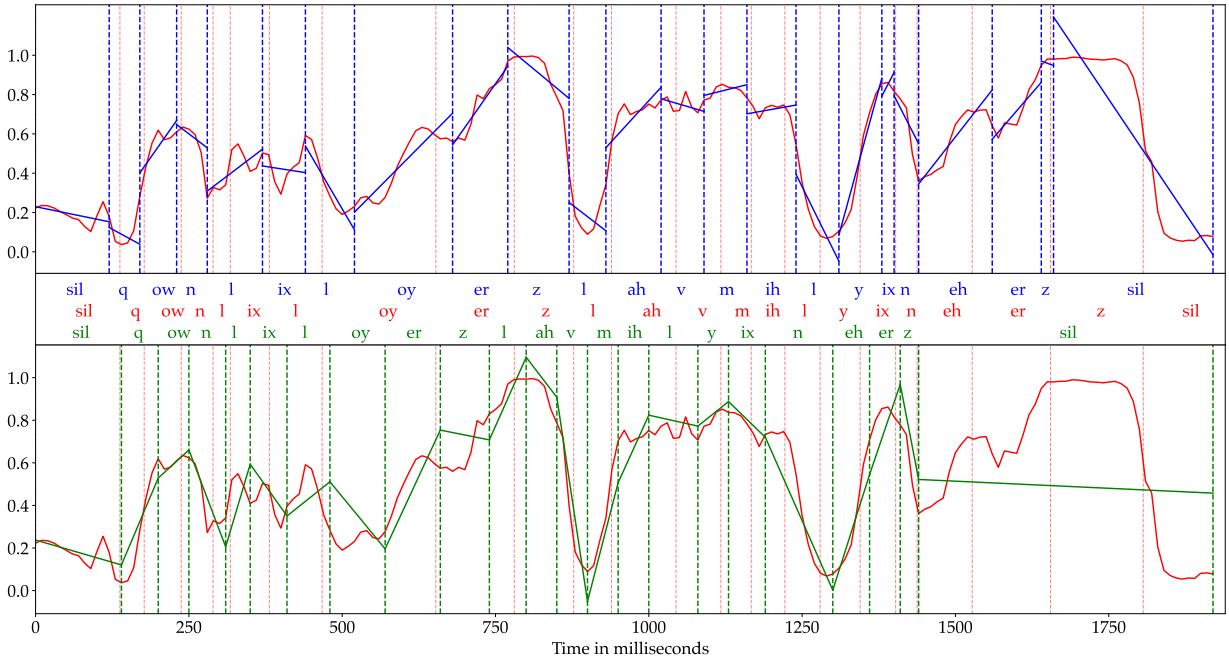
The DPLTM system outperforms the CPLTM system by a statistically significant margin of 3.68% in recognition accuracy. The CPLTM system's inter-segmental dependency modelling approach preserves the natural production process of speech, which is discussed in Section 6.2.2. The study's initial hypothesis was that a more accurate speech production model could potentially enhance an ASR system's overall performance. The observed performance reduction with the CPLTM model motivated further investigation and comparison between the DPLTM and CPLTM systems. A visual analysis was conducted on their output trajectories to further explore the observed performance discrepancy between the two systems.

Like the PCTM, the accuracy rates obtained for these experiments are considerably high and competitive compared to a baseline HMM system.

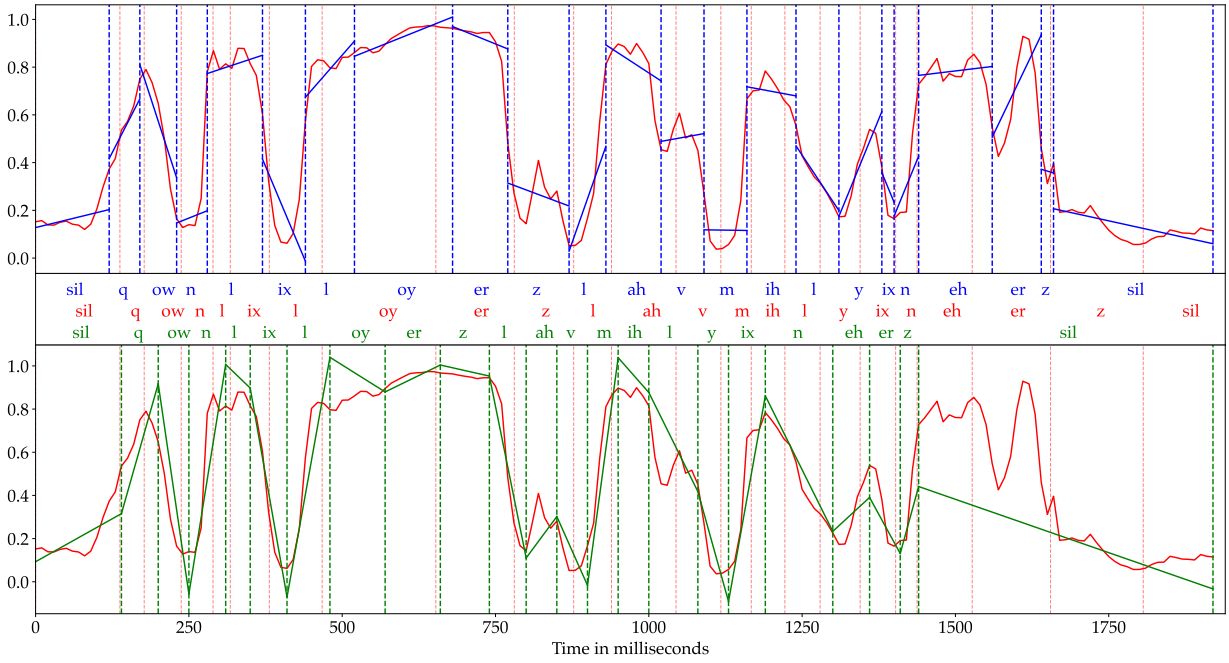
Visual Comparison of DPLTM and CPLTM

The implementation of the CSHMM system in this work stores both the discrete components of the model, namely the complete phonetic history and state segmentation, as well as the continuous details represented by the distribution parameters. This feature enables the extraction of a full phonetic decode from a single forward pass through the data. As new data is observed, the state parameters such as the realisation mean, variance, phonetic history, current time, and current state, are recorded. Based on this log, the approximate output trajectories for the DPLTM and CPLTM systems have been visualised and presented for analysis.

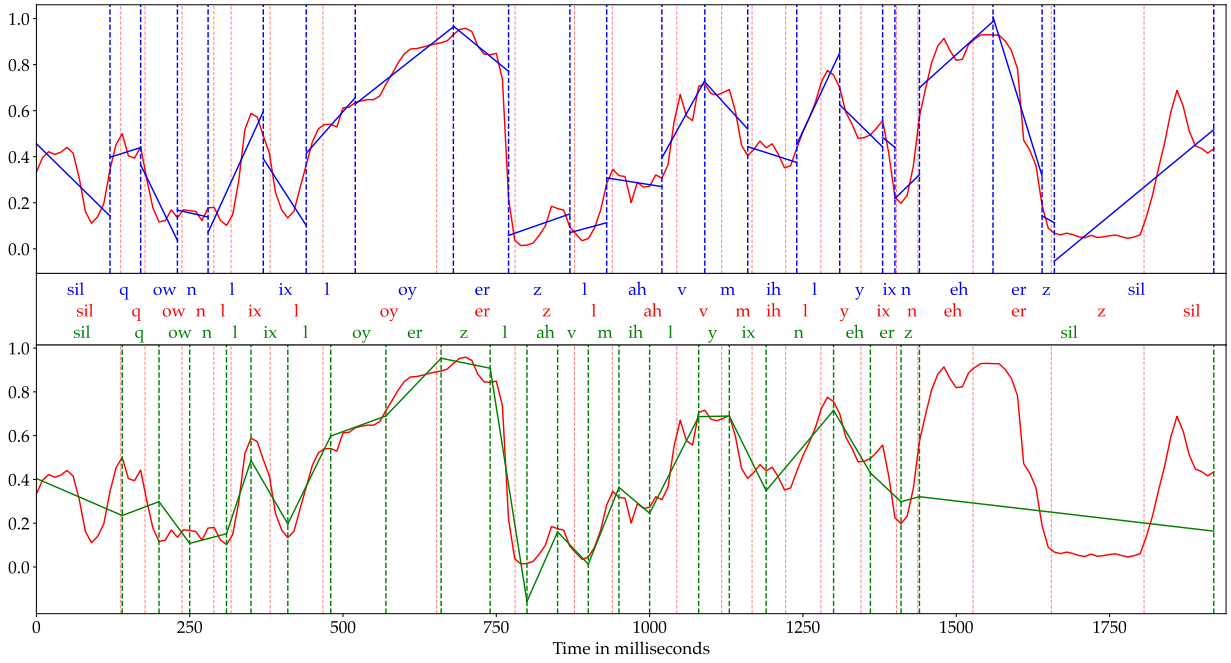
The subsequent figures show the outcome of a forced alignment experiment conducted for each system, with the DPLTM system shown on the top plot in blue and the CPLTM system displayed on the bottom plot in green. The original TIMIT segmentation and feature are illustrated in red. The forced alignment uses a strict language model to ensure a complete utterance transcription is present at the end of the decode.



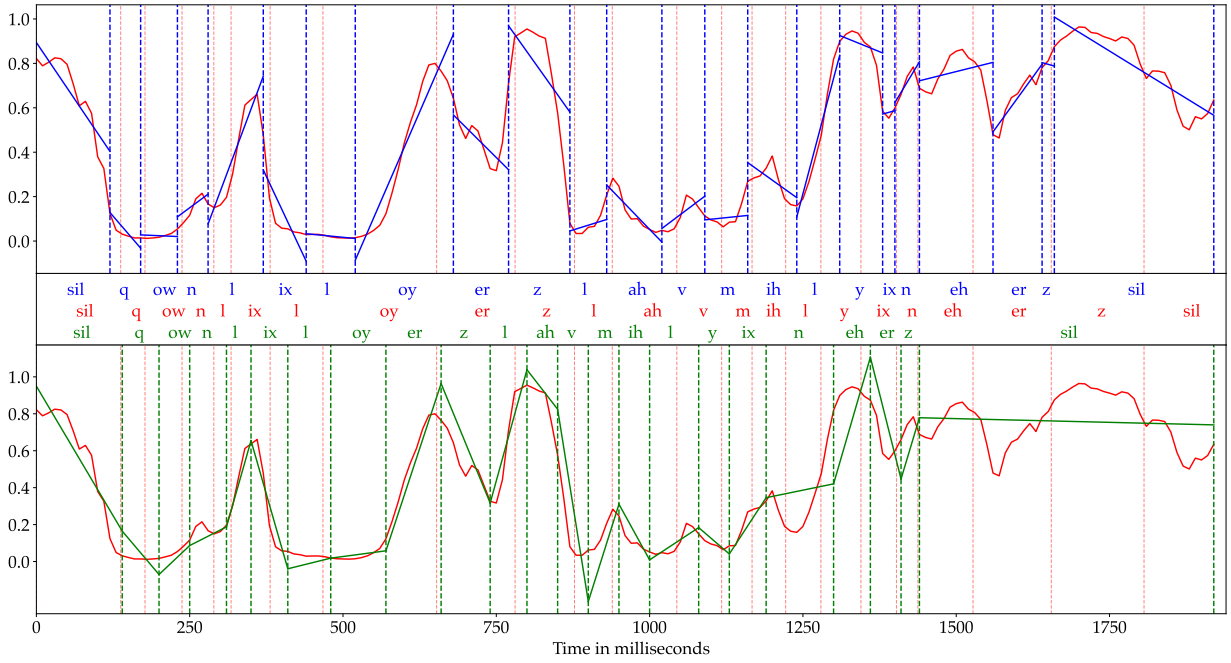
(A) BNF dimension 1



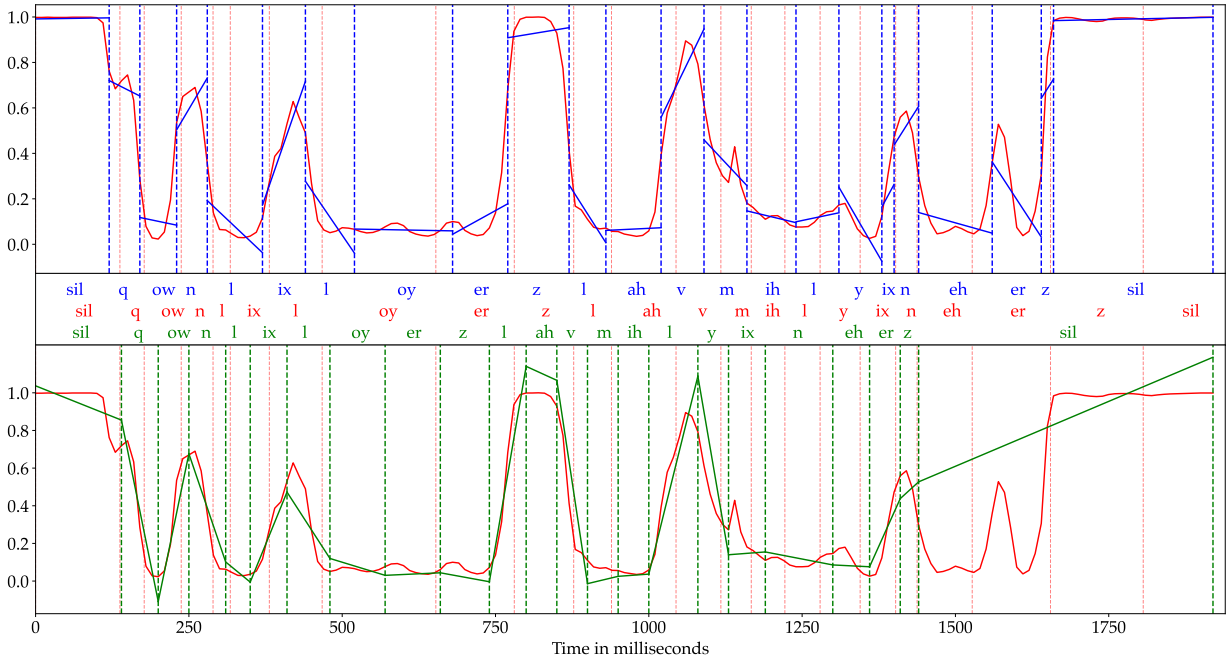
(B) BNF dimension 2



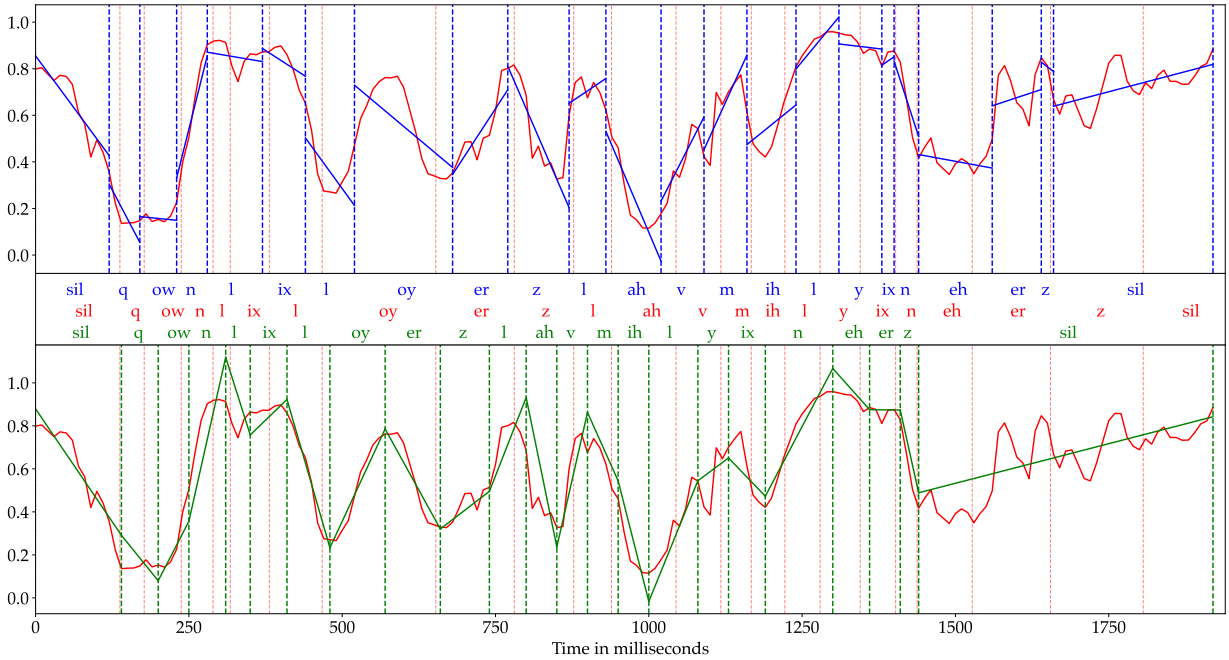
(C) BNF dimension 3



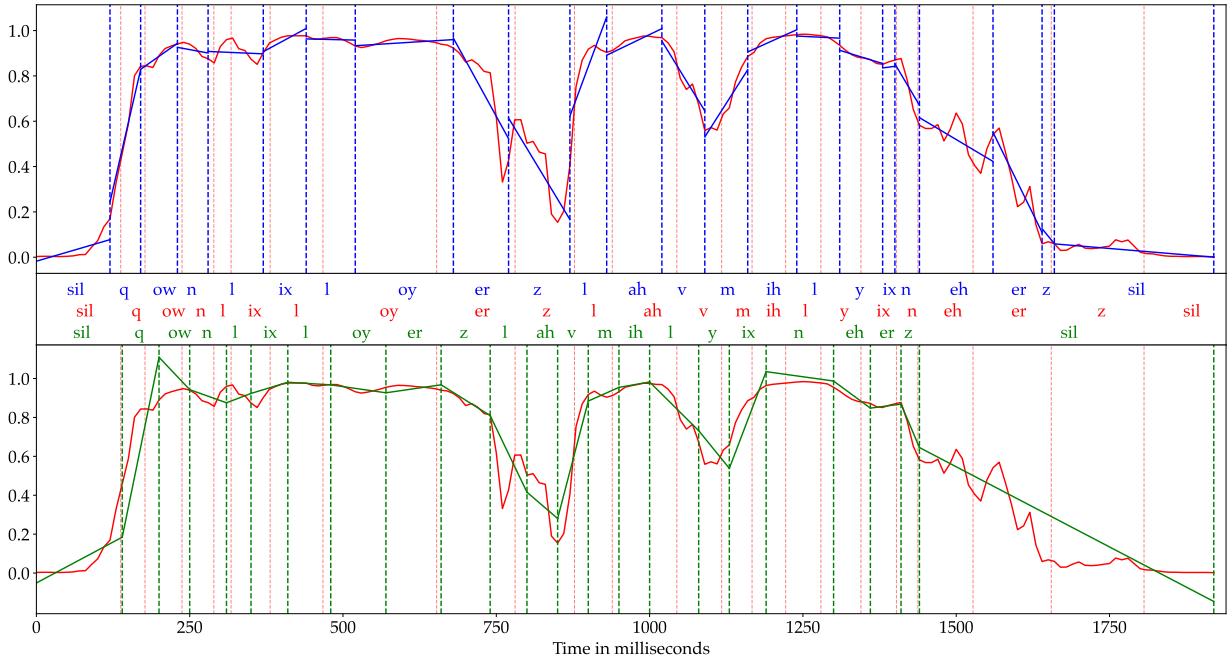
(D) BNF dimension 4



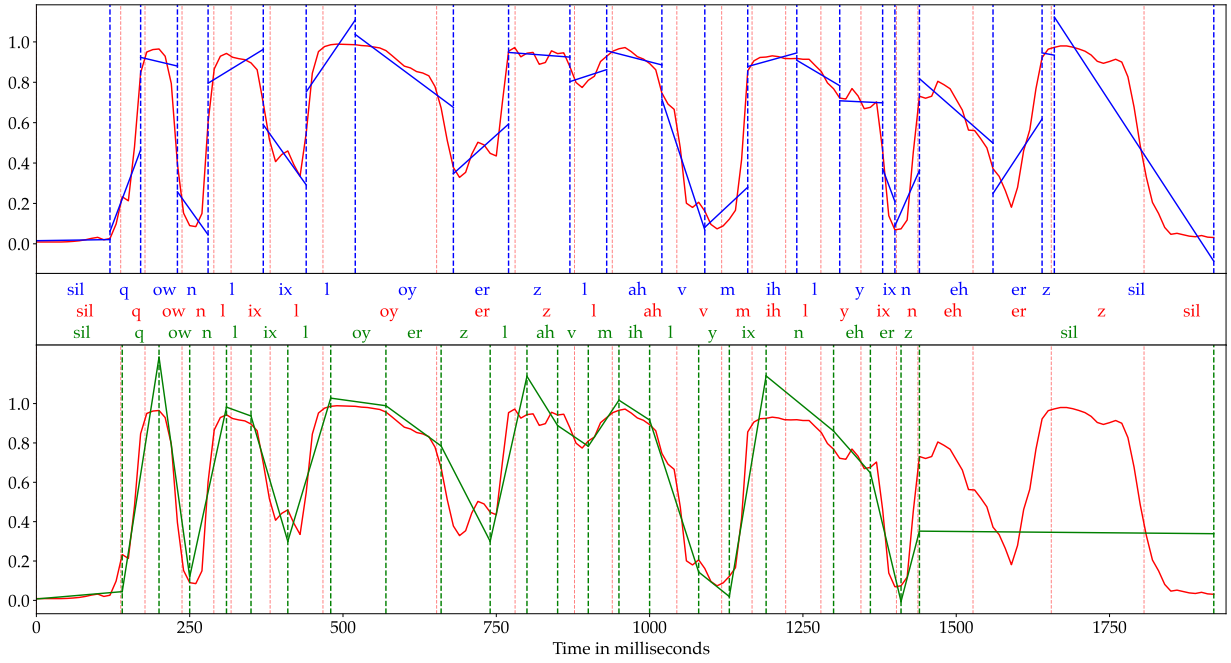
(E) BNF dimension 5



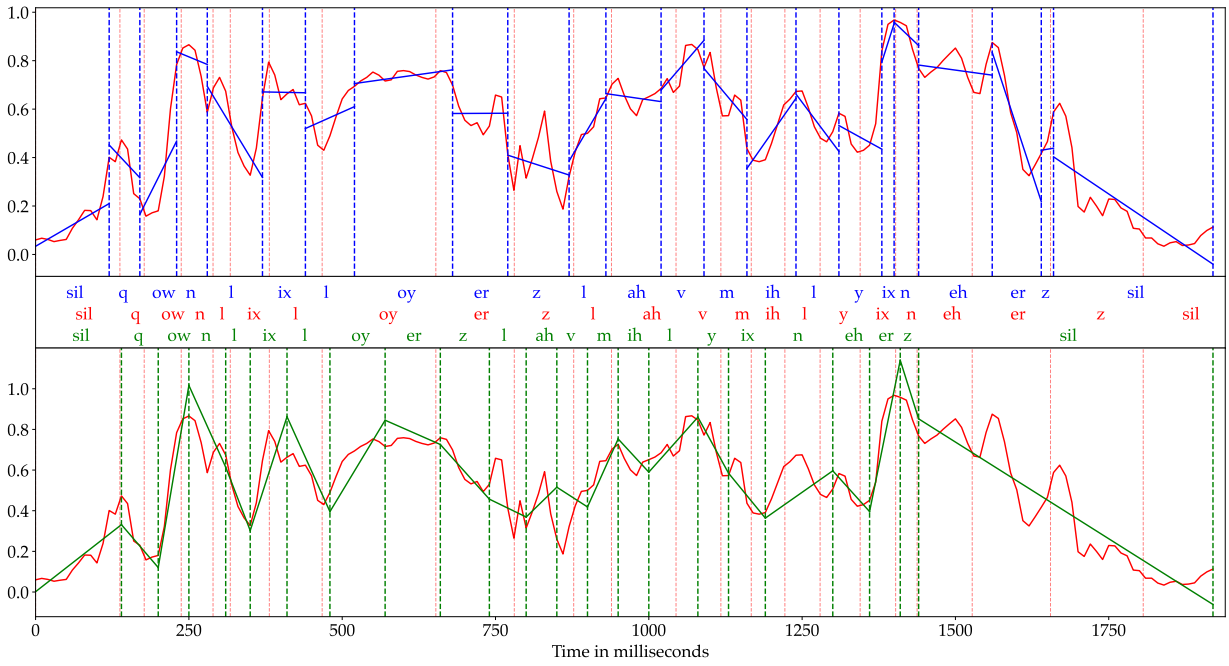
(F) BNF dimension 6



(G) BNF dimension 7



(H) BNF dimension 8



(I) BNF dimension 9

FIGURE 6.11: TIMIT sentence SX110 Bottleneck feature data with approximated system trajectories DPLTM (blue) CPLTM (green) with segmentation (dotted vertical)

In order to aid in the analysis of the bottleneck dimensions, each dimension has been presented in a separate visualisation. However, the overall likelihood of a hypothesis is determined by a weighted combination of all 9 bottleneck feature dimensions. The purpose of examining each dimension individually is to observe how well the trajectory fits over each of the distinct dimension trajectories, as these trajectories have unique contours, as previously discussed in Section 4.3.

6.4.3 System Evaluation and Test of Significance

The main objective of the experiments in this thesis is to evaluate various segmental CSHMM acoustic models with different trajectory constraints, inspired by the robustness and flexibility of Markov Models, as discussed in Chapter 3. A binomial

significance test is employed to assess the significance of the results. This test is used to compare the performance of two competing models, namely DPLTM and CPLTM, and determine if the errors produced by one of the models can be attributed to random chance or if the model is statistically weaker in recognising specific phone classes associated with the errors compared to its counterpart. To achieve this, a null hypothesis is established, which assumes that any differences in errors generated by the two systems can be attributed to random errors inherent in ASR systems.

A comparison of phone confusions between each system is conducted to test the null hypothesis. It is assumed that phone classification is governed by a multinomial distribution with N=41 output probabilities; 41 because of the evaluation phone set mapping described in Appendix A. Given a set of K examples of the i^{th} phone ϕ_i, the output probabilities which govern the multinomial distribution $p_{i,1}, p_{i,2}, ..., p_{i,N}$ correspond to the i^{th} row of a confusion matrix. If $|\phi_i \rightarrow \phi_j|$ represents the number of occurrences of phone substitution $\phi_i \rightarrow \phi_j$, then $|\phi_i \rightarrow \phi_i|$ can be evaluated, which corresponds to the diagonal element of the confusion matrix. The marginal distribution can then be calculated as a binomial distribution with parameters p_i, i, and K:

$$P(|\phi_i \rightarrow \phi_i| = k) = \frac{K!}{k!(K-k)!} p_{i,i}^{k} (1 - p_{i,i})^{K-k} \tag{6.29}$$

Where k is the number of classifications of a particular phone ϕ_i.

The aim is to determine the probability $P(|\phi_i \rightarrow \phi_i| = k)$ to assess whether the misclassifications result from random ASR errors or if they are significantly different between the two systems. A threshold of $j = 0.05$ is employed such that if the cumulative probability $P(|\phi_i \rightarrow \phi_i| \geq k) \leq j$, it can be concluded that the error pattern between systems is statistically significant and, therefore, likely to result from differences in the underlying models. This test for statistical significance considers the prevalence of phone substitutions in ASR experiments. It only considers a substitution to be significant if it occurs more frequently than would be expected due to random

variations in the reference data.

	% Correct			% Correct	
Phone	DPLTM	CPLTM	Phone	DPLTM	CPLTM
aa	64.1	68	l	62	68
ae	54.1	58.7	m	69.5	72.3
ah	48.5	50.6	n	67	56.4
aw	57.5	69.3	ng	77.5	71.2
b	77.8	42.3	ow	44	49.2
ch	88.4	82.3	oy	68.6	73.7
d	50.9	17.6	p	75.6	73.9
dh	39.8	62.7	r	62.2	58.4
dx	78.7	66.6	s	83.8	90.8
eh	48.1	48.5	sh	81.3	83.9
er	67.4	74.2	sil	88.3	83
ey	67.4	74.2	t	71.5	68.9
f	77.6	78.4	th	49.4	47
g	59	45.8	uh	30.4	19.9
hh	71.6	74.3	uw	59.5	60.2
ih	40.2	38.3	v	57.4	36.8
ix	50.4	51.8	w	69.2	75.3
iy	74.2	80.3	y	62	50.5
jh	61.5	53.3	z	67.5	65.4
k	77.2	73.7	zh	61.5	55.4

TABLE 6.5: Binomial significance test results showing the percentage correct for DPLTM and CPLTM system. Highlighted cells signify a system performance to be statistically significantly better than the comparative system.

Table 6.5 presents the results of the significance test conducted to evaluate the segmental CSHMM models. The test aimed to compare the performance of two competing models, namely the DPLTM and CPLTM systems. The results of the test indicate that the CPLTM system outperforms the DPLTM system in terms of recognising vowel phonemes. Specifically, the CPLTM system statistically significantly recognised 60% of the vowels better than the DPLTM system. Furthermore, the overall recognition rate for vowel phonemes was better with the CPLTM system, which accurately recognised 85% of the vowel sounds. On the other hand, the DPLTM system was found to better recognise 100% of the fricative and plosive sounds.

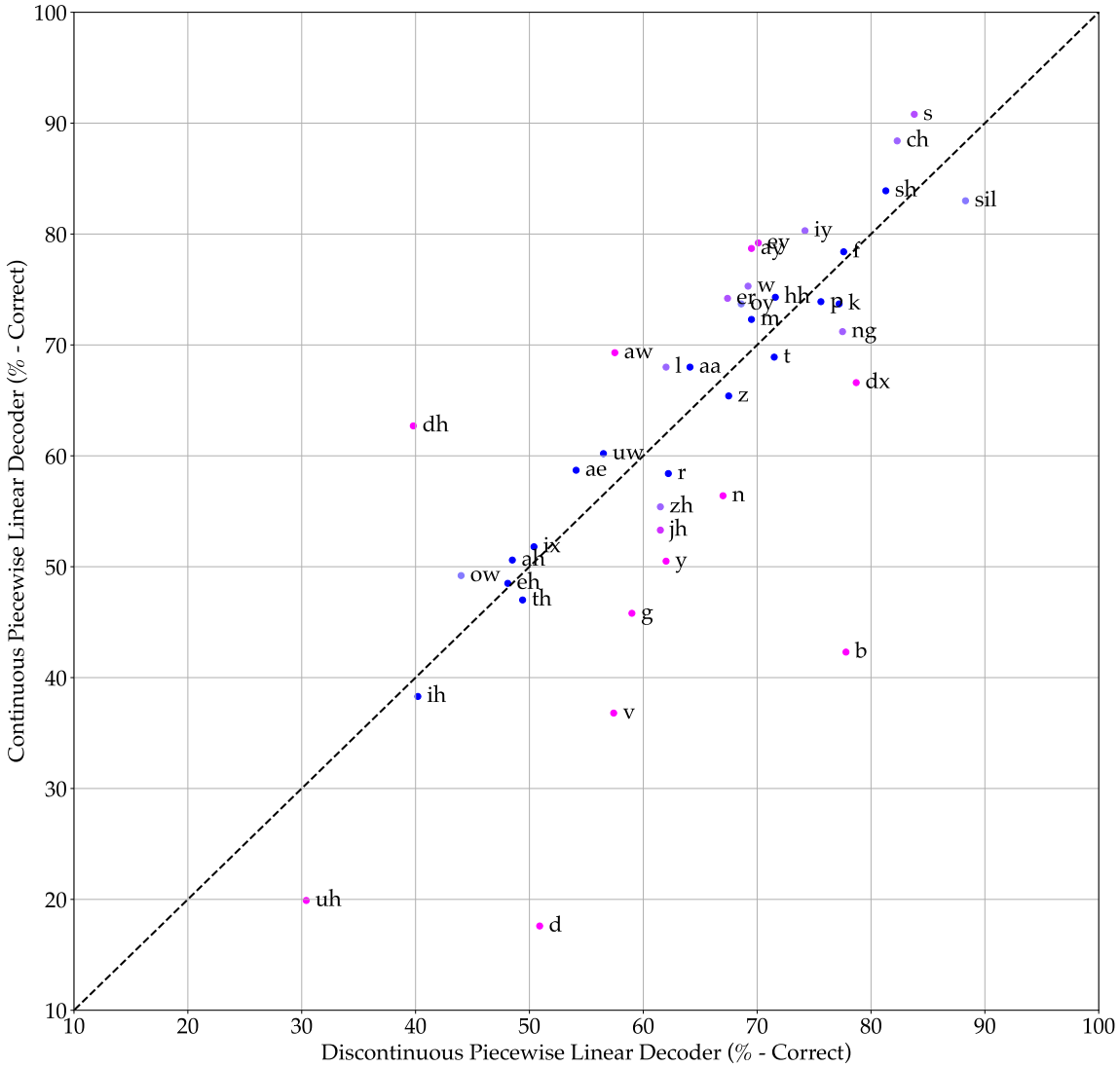


FIGURE 6.12: Visualisation of phone recognition performance (% Correct) showing significant differences between DPLTM and CPLTM systems - Colour gradient used to show scale of significance with blue points as least significant and pink points most statistically significant.

Figure 6.12 displays the plot of the phone confusions between the DPLTM and CPLTM systems. The x-axis and y-axis correspond to the DPLTM and CPLTM systems, respectively, with the axis values representing the phone recognition percentage correct. The pink data points denote statistically significant confusions between the two systems. Purple data points show system preference with less significance, and the blue data

points show no statistical significance between the systems. The dotted diagonal line indicates the equal performance of the two systems, with data points close to this line signifying similar performance regardless of the modelling system. Conversely, the data points indicating the DPLTM system performs better appear in the lower right triangle, and phonemes better recognised by the CPLTM system exist in the upper left triangle.

These results have prompted the research question of whether a soft-continuity measure can be defined to influence the behaviour of discontinuities at the end of a segment. This research question is explored in Chapter 7.

6.4.4 Summary

This chapter presents three CSHMMs that incorporate different and distinct trajectory assumptions. These models are the piecewise constant trajectory model (PCTM), the Discontinuous Piecewise Linear Trajectory Model (DPLTM), and the Continuous Piecewise Linear Trajectory Model (CPLTM). The PCTM model assumes that a static constant trajectory is sufficient to model a speech segment and is directly comparable to an HMM system. The DPLTM model assumes a linear trajectory characterises a segment with no trajectory constraints at a segment boundary and is directly comparable to the probabilistic trajectory models as presented in (Holmes and Russell, 1999). The CPLTM model also assumes piecewise linearity within a segment with an additional constraint at segment boundaries that enforces a trajectory to be continuous throughout the entire speech utterance, this model is most similar to the dwell-transition models presented in (Champion and Houghton, 2016).

Previous work on CSHMMs favoured Formant features as a preferred feature representation that complements the underlying model assumptions. Formants directly capture the resonant frequencies of the vocal tract when speech is produced, however, they are limited to only being well structured in voiced regions of speech. Additionally, extracting formants is notoriously difficult, with very little open-source formant data

for experimentation. This work proposes a bottleneck feature representation developed in (Bai et al., 2015) to benchmark the CSHMMs using the TIMIT corpus, which is widely used in ASR research.

Compared to a standard HMM-GMM model, the segmental CSHMMs performed competitively in the phone recognition experiments. The DPLTM system was found to be superior. However, a visual analysis of the state trajectories for the DPLTM and CPLTM systems suggested that each model fit the data better in different regions. A binomial significance test was conducted to further explore this finding and deduce whether phone confusions were significant or a consequence of random errors. It was found that the DPLTM system performed significantly better for fricative and plosive phone groups, while the strength of the CPLTM system was in regions corresponding to vowels. This finding motivated the exploration of whether a soft-continuity measure could influence the behaviour of the trajectory's discontinuity at the segment's endpoint, which is the focus of the next chapter.

Chapter 7

Soft Continuity Measure for Continuous State Segmental Model

The current study aims to investigate the problem of inter-segmental continuity by proposing a segmental CSHMM system that employs a probabilistic approach to specify the continuity of a state trajectory at a change point. This research builds upon the experimental findings presented in Chapter 6. Previous studies, including works by (Bridle, 2004) and (Deng and Ma, 2000), have explored the problem of inter-segmental continuity by investigating various smoothing techniques at segment boundaries, often within the context of co-articulation, as discussed in Section 6.2.2. This study introduces novel techniques to address the problem of inter-segmental continuity within a CSHMM framework, expanding the range of models and experimental research presented in the preceding chapter.

Figures 6.3 and 6.4 depict the schematic diagrams of the DPLTM and CPLTM, respectively. To fully capture the continuous nature of speech production, an ideal soft continuity model would combine the characteristics of both models, allowing for a switch between modelling criteria. This approach would enable appropriate modelling of the continuous nature of the speech production process and the discontinuities in the acoustic feature space, as illustrated in Figure 7.1. The present study aims to introduce a SC-PLTM which has the flexibility to uphold continuity and discontinuity assumptions

at segment boundaries. In Figure 7.1, segments 1-3 exhibit full continuity, consistent with the observations in the CPLTM system, which would be expected in sonorous regions of the signal. However, segments 3, 4, and 5 display discontinuities at the segment boundaries with a constant trajectory in segment 4.

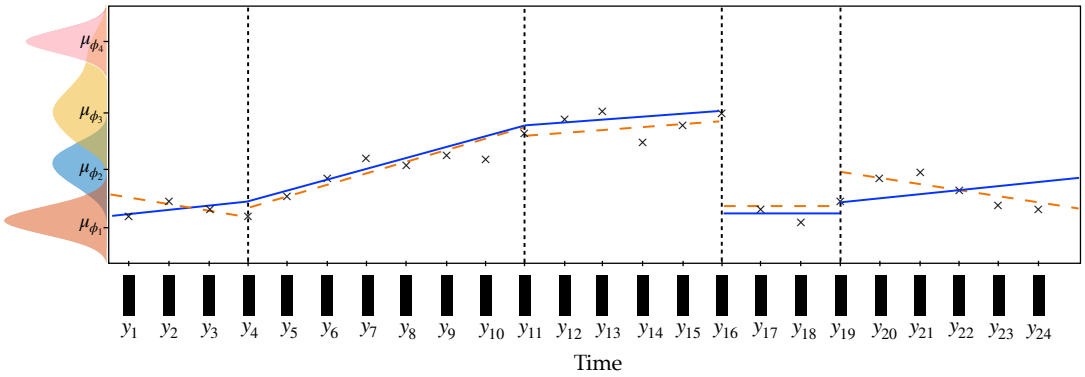


FIGURE 7.1: A schematic idealisation of a Soft-Continuity Piecewise Linear Trajectory Model (SC-PLTM).- Black crosses are observations, the blue line is the realised trajectory, and the orange dashed line is the canonical target trajectory.

The previous experiments discussed in Section 6.4 have highlighted the need to investigate a system that relaxes the strict continuity constraint imposed by the CPLTM. This will enable the existence of trajectory discontinuities in various regions of the bottleneck feature data. The present investigation seeks to evaluate the effect of a soft continuity measure by implementing and evaluating the following approaches.

7.1 Binary Switching Between DPLTM and CPLTM Decoders

The first approach that was considered used a binary switching method to alternate between the DPLTM and CPLTM systems. During the forced alignment training, the average Euclidean distance between the endpoints of a segment trajectory and the canonical start point of the new state was computed at a state change point when a hypothesis branches. This average distance value was subsequently utilised as a threshold to activate the binary switch. However, the resulting hybrid-switch system exhibited a preference towards the DPLTM system, which was attributed to the scaling

of the CSHMM score, represented as K_t, between the two systems. Specifically, the CPLTM system includes a prior measure on the trajectory slope, which is updated at each step throughout the data, leading to a different scaling factor in comparison to the DPLTM system. As a result, it was concluded that enforcing the soft continuity constraint through switching and alternating between the two systems is not feasible unless an appropriate normalisation factor is identified.

7.2 Convolutional Scaling with Constant Global Parameters

Another approach for a soft-continuity model involves using a convolution function at a segment endpoint to model the discontinuity between adjacent segments. A state change-point marks the end of a segment trajectory. In the CPLTM, the state is reparameterised based on the segment endpoint by integrating the slope, the trajectory endpoint value is then used as the subsequent trajectory start-point. A convolution function can be used to model the discontinuity between the segment endpoints. Specifically, a Gaussian function denoted as $g = \mathcal{N}(0, \sigma^2)$ with a global variance parameter σ^2 is convolved with the state trajectory function parameterised on the segment endpoint, denoted as f. This convolution results in a scaled Gaussian distribution $f' \sim \mathcal{N}(\mu', \sigma^{2\prime})$, as defined in Appendix B, where:

$$\mu' = \mu_f + \mu_g$$

and

$$\sigma^{2\prime} = \sigma_f^2 + \sigma_g^2$$

The scaled Gaussian probability density function f' becomes steeper or flatter depending on the value of σ^2. By scaling the endpoint distribution, the region around the current trajectory endpoint with which the next trajectory start-point will have a high probability can be probabilistically constrained.

To evaluate the convolutional scaling method, an empirical investigation is conducted

by exploring a range of variance values for a zero-mean scaling Gaussian distribution. The use of a fixed zero-mean constrains the trajectory endpoint from any further shift, while variance scaling can broaden or narrow the distribution at an endpoint, adjusting the region within which a high probability start-point can occur. As the variance in the scaling Gaussian probability density function increases, the endpoint distribution becomes flatter, allowing for a greater degree of permissible discontinuity. Reducing the variance of the Gaussian convolution to zero would result in the system imposing a hard continuity constraint consistent with the CPLTM.

A comparative analysis was conducted to assess the effectiveness of a SC-PLTM with a global variance in comparison to a DPLTM and a CPLTM baseline system. The sub-development test set, comprising of 16 spoken utterances, one by a male and one by a female speaker from each dialect region, as described in 6.4.1, was used for the experiment. A phone recognition task was conducted using a standard bi-gram language model and a range of global variance values, spanning from 0.7 to $1e^{-6}$. The results of this experiment are presented in Table 7.1.

Decoding System	Variance	% Correct	% Accuracy
DPLTM *(baseline)*	-	58.71	48.43
CPLTM *(baseline)*	-	63.29	50.14
SC-PLTM	0.7	56.14	49.00
	0.5	57.86	49.14
	0.2	63.29	50.57
	0.1	61.00	50.86
	0.01	61.57	52.00
	0.001	64.14	51.71
	0.0001	63.86	50.71
	0.000001	63.71	50.43

TABLE 7.1: Performance of a SC-PTLM with scaled global variance compared to DPLTM and CPLTM baselines on TIMIT development dataset.

The experiment outcomes indicate that a very low global variance leads to the convergence of the system likelihood value to that of the CPLTM system, whereas a higher variance leads to convergence towards the DPLTM system. These results support the hypothesis that the integration of a probabilistic soft continuity measure can alter the

distribution of end-of-segment values. This alteration, in turn, can either increase or decrease the variance around the trajectory endpoint, ultimately constraining the extent of discontinuity. Building on the insights from Table 7.1, a phone recognition experiment was conducted on the TIMIT core test set. A fixed global continuity variance of $\sigma = 0.01$ was used, which yields the highest accuracy result on the sub-development set. This variance value offers the system adequate flexibility to accommodate reasonable levels of discontinuity at the trajectory endpoints while imposing restrictions to prevent the occurrence of larger jumps.

Decoding System	% Correct	% Accuracy
DPLTM	63.62	55.52
CPLTM	63.14	52.72
SC-PLTM	61.68	55.57

TABLE 7.2: Results of phone recognition experiment for a SC-PTLM compared to a DPLTM and a CPLTM using the TIMIT test dataset.

Table 7.2 presents phone recognition results, showing that the SC-PLTM system exhibits a slight drop in recognition performance with an average reduction of 1.7% Correct compared to DPLTM and CPLTM systems. However, the SC-PLTM performs the best in terms of ASR accuracy. One limitation of the SC-PLTM is that the use of a global variance value assumes a uniform allowable discontinuity at a segment boundary for all phoneme combinations, which may not be the case. The discrepancy between percentage correct and accuracy may stem from the system's proficiency in recognising certain phonemes in specific contexts, resulting in higher accuracy for those phonemes.

Acoustic discontinuities can occur at the boundaries between phonemes, and the nature of the discontinuity varies depending on the specific phonemes and their context within the word. For example, stop consonants like /p/, /t/, and /k/ have a brief period of silence followed by a release burst characterised by a sudden increase in energy in the higher frequencies. In fricatives like /s/, /f/, and /sh/, the boundary between phonemes is characterised by a change in the nature of the noise produced by the turbulent airflow through the vocal tract. In vowel sounds, the discontinuity may

be less pronounced, but there may still be subtle changes in the formants or spectral characteristics of the speech signal.

To address the context-dependent nature of discontinuity, we consider a convolution scaling method that incorporates contextual variance parameters, as detailed in the subsequent section.

7.3 Convolution Scaling with Context Dependent Parameters

The previous section has shown that more refined control over inter-segmental continuity can be achieved by scaling the distribution at a trajectory endpoint. The effectiveness of this approach is supported by the results presented in Table 7.1, which show that adjusting the variance of a scaling distribution can lead to convergence towards both the DPLTM and CPLTM system ASR results for a single system. In a subsequent experiment, an optimised variance parameter is utilised in a phone recognition experiment and compared to a baseline DPLTM and CPLTM system. According to the results in Table 7.2, the ASR accuracy for the SC-PLTM system has slightly improved compared to the other systems. Nevertheless, there was a slight drop in the percentage of correctly identified speech sounds for the SC-PLTM system, and this could be due to different pairwise phoneme transitions requiring different levels of discontinuity. Consequently, a context-dependent measure of continuity at segment endpoints is examined. In this method, the parameters of a Gaussian function $g = \mathcal{N}(\mu, \sigma^2)$ are discovered such that the average distance between the trajectory endpoint from state i to the trajectory start-point from state j can be estimated as follows:

$$\mu_{i \to j} = \frac{1}{K} \sum_{k=1}^{K} d_k(f_\omega(i), f_\alpha(j)) \tag{7.1}$$

In this case, $\mu_{i \to j}$ denotes the average distance between the trajectory endpoint from state i to the trajectory start-point for a state j. To calculate this distance for each particular transition, a distance function d can be defined assuming K examples of

transitions from a state i to a state j. The respective trajectory endpoint and start-point are represented by f_ω and f_α. The variance can be estimated as follows:

$$\sigma^2_{i\to j} = \frac{1}{K}\sum_{k=1}^{K}(d_k(f_\omega(i), f_\alpha(j)) - \mu_{i\to j}) \tag{7.2}$$

Introducing inter-segmental continuity in a context-dependent scenario provides the advantage of more accurately considering the effects of co-articulation. However, the number of parameters in the SC-PLTM system increases by $2N(N-1)$, where N represents the number of phones in the lexicon.

For a phone recognition experiment, the context-dependent mean and variance parameters were estimated using the TIMIT train dataset. The average endpoint discontinuity was calculated using a Euclidean distance function, and decoding was performed using the 49 phone set with scoring on the 40 phone set symbols. Table 7.3 provides an overview of the results for the SC-PLTM system, as well as all other developed models in this thesis for reference.

Decoding System	% Correct	% Accuracy	*ins*	*del*	*sub*
PCTM	65.99	60.18	5.81	10.52	23.5
DPLTM	74.76	61.19	9.59	9.93	22.83
CPLTM	68.83	57.51	5.39	16.89	26.46
SC-PLTM (μ, σ^2)	56.41	46.08	10.32	14.80	28.79
SC-PLTM (σ^2)	62.19	52.30	9.90	7.24	30.57

TABLE 7.3: Summary of the baseline phone recognition results obtained for the four optimised Segmental CSHMMs developed in this work.

The results of the baseline SC-PLTM experiment did not meet the expected outcome and exhibited a lower recognition performance compared to the DPLTM and CPLTM systems. Two decoding experiments were conducted. In the first experiment, the estimated mean and variance parameters for the context-dependent scaling Gaussian were used, while the second experiment only utilised the estimated variance parameter. The results indicate a 6% absolute increase in accuracy when the endpoint distribution centroid was fixed by setting the scaling Gaussian mean to zero.

One possible reason for the drop in performance of the SC-PLTM model may be the insufficient training data. The sparsity of training data can affect the estimation of pair-wise phoneme transition parameters and, in turn, impact the overall ASR performance. Despite the experiment's failure to meet projected outcomes, the context-dependent SC-PLTM model remains a promising idea that aligns with previous research findings, as supported by (Richards and Bridle, 1999).

7.4 Summary

This chapter aims to evaluate the hypothesis that incorporating a probabilistic measure of continuity at a segment boundary would lead to improved ASR performance. The experimental results in Table 6.5 support this avenue of exploration, demonstrating that the DPLTM and CPLTM systems each exhibit better recognition performance for distinct phone categories. An initial soft-continuity proposal considered a binary switching system which ultimately exhibited a bias towards the DPLTM system, mainly due to different normalisation factors in the CSHMM "score" value. As a future work suggestion, it may be worthwhile to investigate re-scaling the normalisation constant to enable a switching system. Within the CSHMM framework, a soft-continuity approach can be formalised by performing a convolution at a state change point, employing a scaling Gaussian distribution. This method is computationally efficient, as Gaussian distributions have properties such that the convolution of two Gaussian probability density functions yields a scaled Gaussian probability density function, thus allowing the probabilistic soft-continuity measure to be incorporated into the CSHMM update equations.

The soft-continuity system produced an interesting outcome, demonstrating that adjusting the scaling of the trajectory endpoint distribution can lead to finer control over continuity at segment boundaries. Specifically, when a static global variance of 0.7 was used, the SC-PLTM model converged to the DPLTM ASR outcomes, whereas setting the global variance parameter to $1e^{-}6$ caused the SC-PLTM to converge to the CPLTM

ASR results. Despite this promising result, there was a small degradation in phone recognition performance by the SC-PLTM when compared to the baseline systems tested on the core test set of TIMIT data.

The SC-PLTM system is extended to allow for different degrees of continuity at a segment boundary depending on the context of the recognised phoneme. A context-dependent convolution method was tested, which resulted in a drop in performance compared to the baseline DPLTM and CPLTM systems. It was found that including further shifts in a trajectory endpoint degraded the performance more than when the scaling Gaussian distribution had a zero mean. However, to validate this initial finding, further analysis would be necessary.

In summary, this study investigated the modelling enhancements needed to incorporate a probabilistic measure of trajectory continuity at a segment boundary within a CSHMM framework. There has been limited application of CSHMM systems to speech data. The methods proposed in this work constitute a new contribution to the research on CSHMMs, providing a foundation for future research focused on enhancing statistical speech models for ASR. The two methods proposed and evaluated include a binary-switch system and a probabilistic soft-continuity measure. The soft-continuity approach demonstrated the potential for finer controllability of continuity at a segment boundary. Additional analysis of pairwise confusions from the SC-PLTM and statistical assessments of phoneme pairs in the TIMIT data can be completed to determine whether data sparsity constituted a loss in performance.

Chapter 8

Conclusions

The main research question addressed by this thesis is whether a more faithful statistical model of the dynamical speech production process, when applied to speech data, can offer significant improvements over existing methods for ASR tasks. This motivation diverges from recent trends in ASR research that emphasise data-centric deep learning approaches. Instead, this work proposes a practical solution for ASR tasks that face limited availability of training data, which is a valuable consideration for ASR research. The core assumption underlying this research is that a simple and parsimonious system will require less training data and, therefore, can be applied to low-resource tasks.

Prior work in this area has explored various aspects of speech production, such as the impact of co-articulation on the acoustic space, the construction of linear dynamical systems for modelling the speech production process, and the study of articulatory features. However, capturing the dynamic properties of speech is a nontrivial task that requires careful consideration of both the feature representation and the modelling paradigm. The foundation of this work lies in a critical analysis of a family of Markovian acoustic models, including Hidden Markov Models (HMMs), Segmental Hidden Markov models (SHMMs), Linear Dynamical Models (LDMs), and Continuous State Hidden Markov Models (CSHMMs). The primary goal of this research is to extend the study of CSHMMs by benchmarking this method against standardised ASR methods while addressing the limiting inter-segmental independence assumption in speech

models.

Chapters 2 and 3 present a comprehensive review of relevant literature. Chapter 2 focuses on the physiological aspects of speech production and emphasises the importance of selecting a suitable feature representation that complements the underlying model structure. This chapter critically explains the speech production process and its related components. Chapter 3 takes on a tutorial-style approach to review the Markovian models discussed in this work, introducing their modelling assumptions, notation, and limitations. These chapters provide the foundational intuition and modelling assumptions that motivated the study of the CSHMMs in this work. Furthermore, the presentation of the Markovian models using intuitive diagrams and simplified notation removes any notation inconsistencies that arise from the application of these models in various fields. This work agrees with the study by (Ainsleigh, 2001), which categorises the four Markovian models in this work as constrained examples of a more general Continuous state HMM.

Chapter 4 explores an alternative feature representation as input for the models proposed in this thesis. There have been extensive efforts in recent decades to explore alternative feature representations and speech models that more accurately capture the dynamic properties of speech. The very low dimensional bottleneck features (BNFs) used in this work were first introduced in (Bai et al., 2015) and have been explored in several studies (Bai, 2018; Weber et al., 2014; Houghton et al., 2015). In each of these studies, BNFs are claimed to be a promising representation when exploring CSHMMs. A visual analysis and a discussion of the BNFs show them to be consistently smooth trajectories with distinct characteristics. Although there is no intuitive interpretation of the features, it has been possible to map them to phonetically meaningful regions that correlate to articulatory positions. Furthermore, Chapter 6 presents a further visual analysis of the CSHMM trajectory recovery of BNFs, demonstrating that for the DPLTM system, the transition regions have good alignment with the TIMIT phoneme boundaries. The decision to use bottleneck features in the experimental design can

be attributed to several factors, including the ease of extracting stable features for experimentation, the visual analysis of the features which suggests they would be a complimentary representation to the underlying model proposed, the articulatory properties present in this particular BNF set, the low dimensionality of the input which allows for a parsimonious solution to acoustic modelling, and finally, the competitive recognition performance when compared to a more standard MFCC representation.

To address the more prominent research agenda of applying trajectory-based CSHMMs to an ASR task, this work presents the modelling assumptions and benchmark results of four systems named: Piecewise Constant Trajectory Model (PCTM), Discontinuous Piecewise Linear Trajectory Model (DPLTM), Continuous Piecewise Linear Trajectory Model (CPLTM) and the Soft Continuity Piecewise Linear Trajectory Model (SC-PLTM). These models are implemented, and baseline results are reported using the widely adopted speech corpus TIMIT. This work provides the first presentation of Segmental CSHMMs benchmarked using the full TIMIT test data, which contributes to the broader field of speech research by reporting the recognition performance of such models compared to other widely known techniques. The overall recognition results for the PCTM, DPLTM, and CPLTM show a promising comparative result against a standard HMM-GMM system. The ASR results demonstrate that the DPLTM is a superior system outperforming both the PCTM and the CPLTM. However, analysing the significance of recognition errors reveals an interesting result. Both the DPLTM and CPLTM are better models for different categories of phonemes. The CPLTM performs significantly better than the DPLTM in 85% of the vowel regions of speech, whereas the DPLTM outperforms the former model 100% of the time in fricative regions. This finding motivated the development of a system that can take advantage of the trajectory assumptions of both DPLTM and CPLTM, enabling finer controllability of continuity at a segment boundary.

Another research contribution of this work is developing and applying a soft-continuity system that extends the CSHMM trajectory formulation to allow for probabilistic

scaling of a trajectory endpoint. The study explores three approaches to employing a soft continuity measure and provides baseline results using the TIMIT test data. It was discovered that using a binary switching method between the two systems is not mathematically tractable due to the normalisation constants of the model update equations. This method is highly biased towards the DPLTM system, and it was found that a hypothesis could switch from the CPLTM to the DPLTM but not vice versa due to the difference in system normalisation constants. The integral that marginalises the slope parameter in the CPLTM system scales the score in a way that it is much smaller than the DPLTM, making this method unreliable. To address this issue, a second methodology is explored that leverages the properties of Gaussian distributions to perform a convolution with a scaling Gaussian distribution, thereby stretching or flattening the distribution of a trajectory endpoint. An empirical experiment validated the intuition that the flatter the trajectory endpoint distribution, the more discontinuity is permitted as there is a broader region of high-probability start points. When this endpoint distribution is flat, the DPLTM is achieved, whereas squashing the trajectory endpoint to narrow the distribution to a single peak produces the CPLTM. However, the recognition performance of the SC-PLTM system failed to yield improvements, possibly due to the homogeneity of the scaling distribution across all phoneme contexts. Therefore, this study explored a context-dependent scaling for the SC-PLTM system, which introduced more parameters but again failed to improve upon the DPLTM and CPLTM systems. Despite this, the soft continuity system is an interesting theoretical solution that holds promise in addressing the problem of co-articulation in the context of CSHMMs. Further investigations to better understand the nuance of co-articulation in this setting is a compelling extension of this work.

8.1 Future Work

The work presented in this thesis on trajectory-based CSHMMs for ASR has opened up several avenues for further research. Based on the findings and limitations of the

present study, the following directions for future work are suggested.

First, the soft-continuity framework introduced in this study has shown promise in tackling the problem of inter-segmental independence. However, more work is needed to understand the optimal way to incorporate context-dependent scaling. A valid research question to explore is whether single-state model units are the correct resolution for this method. A diphone may be a more appropriate modelling unit, as it would allow for the context-dependent information to be encoded at a state level which may improve the model's accuracy. A future experiment could investigate the performance of the soft-continuity system using diphones as the modelling unit and compare it against the system's performance using single-state model units presented here. The current implementation of the soft-continuity system was trained and evaluated using the TIMIT corpus. However, this speech corpus may not contain sufficient examples of phoneme pairs to adequately train the context-dependent system. Investigating how data sparsity in the training stage affects the output recognition performance would be valuable. To test this, a reduced evaluation set containing phoneme pairs that are well represented in the training examples could be used to validate the effectiveness of the proposed methodology.

Second, the current work focuses on the TIMIT corpus, a widely adopted research speech corpus. However, it would be valuable to extend this study to other corpora and to evaluate the performance of the trajectory-based CSHMMs on more diverse datasets. In particular, it would be interesting to evaluate the soft-continuity system's performance on corpora with more complex co-articulation patterns, such as non-native speaker datasets.

Third, the experiments in this work have utilised an input bottleneck feature representation to validate the suitability of BNFs for ASR. However, to further strengthen this hypothesis, it would be of great interest to conduct comparative experiments on an articulatory-based dataset, such as the MOCHA-TIMIT database (Wrench, 1999). By

comparing the recognition results of CSHMMs using both acoustic and articulatory features, this research can contribute significantly to the argument that CSHMMs provide a more faithful model of the physiological aspects of speech production. As speech production is a complex phenomenon that involves both articulatory and acoustic aspects, the results of such experiments could provide valuable insights into the effectiveness of CSHMMs in capturing the complexities of speech production and guide future research in this area.

Finally, a more theoretical direction for future research could be to further explore the mathematical properties of CSHMMs in more detail. For example, investigating the mathematical structure of CSHMMs could lead to a deeper understanding of how these models work and solve the problem of the score imbalance in the binary switching soft continuity system by mapping the scoring criteria of the DPLTM and CPLTM systems to the same scale. Other avenues of exploration include investigating the properties of CSHMMs in non-linear or non-Gaussian environments.

Appendix A

TIMIT Phone Set Mappings

TABLE A.1: Recommended phoneme mappings from 61 to 49 and 41 phone symbols (Lee and Hon, 1989).

	61- Phone set	49- Phone set	41- Phone set
Stops	p	p	p
	t	t	t
	k	k	k
	b	b	b
	d	d	d
	dx	dx	dx
	g	g	g
	q	q	sil
Closures	pcl	cl	
	tcl		
	kcl		
	bcl	vcl	
	dcl		
	gcl		
Other	pau	sil	
	h#		
	epi	epi	

	61- Phone set	49- Phone set	41- Phone set
Semivowels/Glides	l	l	l
	el	el	
	r	r	r
	w	w	w
	y	y	y
	hh	hh	hh
	hv		
Vowels	iy	iy	iy
	ih	ih	ih
	ix	ix	
	eh	eh	eh
	ey	ey	ey
	ae	ae	ae
	aa	aa	aa
	ao	ao	
	aw	aw	aw
	ay	ay	ay
	ah	ah	ah
	ax-h		
	ax	ax	
	oy	oy	oy
	ow	ow	ow
	uh	uh	uh
	uw	uw	uw
	ux		
	er	er	er
	axr		

	61- Phone set	49- Phone set	41- Phone set
Nasals	m	m	m
	em		
	n	n	n
	nx		
	en	en	
	ng	ng	ng
	eng		
(Af-) Fricatives	f	f	f
	v	v	v
	s	s	s
	z	z	z
	sh	sh	sh
	zh	zh	zh
	th	th	th
	dh	dh	dh
	ch	ch	ch
	jh	jh	jh

End of Table A.1.

Appendix B

Gaussian Identities

This work relies on two particular Gaussian identities documented here for completeness.

B.0.1 Product of Two Gaussian PDFs

The product of two Gaussians gives another unnormalised Gaussian.

The multivariate Gaussian distribution has a joint pdf given by:

$$\mathcal{N}(x|\mu, P) = (2\pi)^{-\frac{d}{2}} ||P||^{\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T P(x-\mu)\right) \tag{B.1}$$

where μ is the mean vector (length D) and P is the $(D \times D)$ precision matrix such that $P = \Sigma^{-1}$ with Σ being the positive, definite covariance matrix. A short hand notation can be adopted such that $x \sim \mathcal{N}(\mu, P)$.

Given two Gaussians $\mathcal{N}(x|a, A)$ and $\mathcal{N}(x|b, B)$, it can be shown:

$$\mathcal{N}(x|a, A)\ \mathcal{N}(x|b, B) = Z\mathcal{N}(x|c, C) \tag{B.2}$$

where,

$$C = (A + B) \tag{B.3a}$$

$$c = C^{-1}(Aa + Bb) \tag{B.3b}$$

$$Z = (2\pi)^{-\frac{d}{2}} |A+B|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(a-b)^T(A+B)(a-b)\right) \tag{B.3c}$$

The resulting precision C is the sum of precisions and resultant mean c is the convex sum of the means weighted by the precisions. The normalisation constant neatly simplifies to a Gaussian in a or b. To prove this identity substitute eq. (B.1) into eq. (B.2) such that:

$$\begin{aligned}(2\pi)^{-\frac{d}{2}} |A|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(x-a)^T A(x-a)\right) (2\pi)^{-\frac{d}{2}} |B|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(x-b)^T B(x-b)\right) \\ = Z\ (2\pi)^{-\frac{d}{2}} |C|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(x-c)^T C(x-c)\right)\end{aligned} \tag{B.4}$$

To simplify and to improve readability, take the log of eq. (B.4).

$$\begin{aligned}-d\log(2\pi) + \log(|A|) - (x-a)^T A(x-a)\ + \log(|B|) - (x-b)^T B(x-b) \\ = 2\log Z + \log(|C|) - (x-c)^T C(x-c)\end{aligned} \tag{B.5}$$

Expand all terms from eq. (B.5)

$$\begin{aligned}-d\log(2\pi) + \log(|A|) + \log(|B|) - x^TAx - x^TAa - a^TAx + a^TAa - x^TBx - \\ x^TBb - b^TBx + b^TBb \\ = 2\log Z + \log(|C|)\ - x^TCx - x^TCc - c^TCx + c^TCc\end{aligned} \tag{B.6}$$

Equating coefficients of the form $x^T.\ x$:

$$x^TAx + x^TBx = x^TCx \quad \rightarrow \quad A + B = C \tag{B.7}$$

Equating coefficients of the form $x^T.\ -$:

$$\begin{aligned}x^TAa + x^TBb &= x^TCc \\ Aa + Bb = Cc \quad &\rightarrow \quad c = C^{-1}(Aa + Bb)\end{aligned} \tag{B.8}$$

Equating all other coefficients:

$$\begin{aligned} -d\log(2\pi) + \log(|A|) + \log(|B|) - a^T Aa - b^T Bb &= 2\log Z + \log(|C|) - c^T Cc \\ 2\log Z &= -d\log(2\pi) + \log(|A|) + \log(|B|) - \log(|C|) - a^T Aa - b^T Bb + c^T Cc \end{aligned} \tag{B.9}$$

Substitute terms from eq. (B.3a) and eq. (B.3b):

$$\begin{aligned} 2\log Z = -d\log(2\pi) + \log(|A|) + \log(|B|) - \log(|(A+B)|) \\ -a^T Aa - b^T Bb + C^{-1}(Aa+Bb)CC^{-1}(Aa+Bb) \end{aligned} \tag{B.10}$$

$$\begin{aligned} = -d\log(2\pi) + \log(|A|) + \log(|B|) - \log(|(A+B)|) \\ -a^T(A - AC^{-1}A)a - b^T(B - BC^{-1}B)b + a^T(AC^{-1}B)b + b^T(BC^{-1}A)a \end{aligned} \tag{B.11}$$

Eq. (B.11) can be simplified using the matrix inversion lemma that states:

$$C = (A+B)^{-1} = A - A(A+B)^{-1}A = B - B(A+B)^{-1}B$$

The log terms in eq. (B.11) simplified and other terms resemble the expanded exponential term of a Gaussian and can be re-written such that:

$$\log Z = -\frac{d}{2}\log(2\pi) + \frac{1}{2}\log(|C|) - \frac{1}{2}\left((a-b)^T C(a-b)\right). \tag{B.12}$$

Let x and y be jointly Gaussian random vectors such that:

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix} \right) \tag{B.13}$$

The product of two Gaussians identity holds and following steps used from eq. (B.4) to eq. (B.12) the following can be achieved:

$$\mathcal{N}\left(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix} \right) . \mathcal{N}\left(\begin{bmatrix} e \\ f \end{bmatrix}, \begin{bmatrix} E & D \\ D^T & F \end{bmatrix} \right) \propto \mathcal{N}\left(\begin{bmatrix} p \\ q \end{bmatrix}, \begin{bmatrix} P & W \\ W^T & Q \end{bmatrix} \right) \tag{B.14}$$

where:

$$\begin{bmatrix} P & W \\ W^T & Q \end{bmatrix} = \begin{bmatrix} A+E & C+D \\ C^T+D^T & B+F \end{bmatrix} \tag{B.15}$$

$$\begin{bmatrix} p \\ q \end{bmatrix} = \left(\begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} A & C \\ C^T & B \end{bmatrix} + \begin{bmatrix} E & D \\ D^T & F \end{bmatrix} \begin{bmatrix} e \\ f \end{bmatrix} . \left(\begin{bmatrix} P & W \\ W^T & Q \end{bmatrix} \right)^{-1} \right) \tag{B.16}$$

B.0.2 Convolution of Two Gaussian PDFs

The convolution of two Gaussians gives another unnormalised Gaussian.

Given two Gaussian functions $f(x) \sim \mathcal{N}(a, \Sigma_A)$ and $g(x) \sim \mathcal{N}(b, \Sigma_B)$, where a, b and Σ_A, Σ_B are the mean and variance respectively, it can be shown:

$$\mathcal{N}(a, \Sigma_A) * \mathcal{N}(b, \Sigma_B) \propto \mathcal{N}(a+b, \Sigma_A + \Sigma_B) \tag{B.17}$$

In general, resultant mean is equal to the sum of each function mean $\mu_{f(x)*g(x)} = \mu_{f(x)} + \mu_{g(x)}$. The resultant variance is equal to the sum of each function variance $\Sigma_{f(x)*g(x)} = \Sigma_{f(x)} + \Sigma_{g(x)}$.

Bibliography

Ainsleigh, P. L. (2001). *Theory of continuous-state hidden Markov models and hidden Gauss-Markov models*. Tech. rep. Naval Undersea Warfare Center Div. Newport.

Ainsleigh, P. L., N. Kehtarnavaz, and R. L. Streit (2002). Hidden Gauss-Markov models for signal classification. In: *IEEE Transactions on Signal Processing* 50.6, pp. 1355–1367.

Bahl, L. R., F. Jelinek, and R. L. Mercer (1983). A maximum likelihood approach to continuous speech recognition. In: *IEEE transactions on pattern analysis and machine intelligence* 2, pp. 179–190.

Bai, L. (2018). Speech analysis using very low-dimensional bottleneck features and phone-class dependent neural networks. PhD thesis. University of Birmingham, UK.

Bai, L., P. Jančovič, M. J. Russell, and P. Weber (2015). Analysis of a low-dimensional bottleneck neural network representation of speech for modelling speech dynamics. In: *Proc. Interspeech,* Dresden, Germany, pp. 583–587.

Bai, L., P. Weber, P. Jančovič, and M. J. Russell (2018). Exploring How Phone Classification Neural Networks Learn Phonetic Information by Visualising and Interpreting Bottleneck Features. In: *Proc. Interspeech,* Hyderabad, India, pp. 1472–1476.

Baum, L. E., T. Petrie, G. Soules, and N. Weiss (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. In: *The annals of mathematical statistics* 41.1, pp. 164–171.

Blackburn, C. S. (1997). Articulatory methods for speech production and recognition. PhD thesis. PhD Thesis, Cambridge University, Engineering Department.

Boersma, P. and D. Weenink (2001). *Praat: doing phonetics by computer [Computer program].* Version 6.3.03. Retrieved 17 December 2022 from: `http://www.praat.org/`.

Bridle, J., L. Deng, J. Picone, H. B. Richards, J. Ma, T. Kamm, M. Schuster, S. Pike, and R. Regan (1998). *An Investigation of Segmental Hidden Dynamical Models of Speech Coarticulation of Automatic Speech Recognition*. Tech. rep. The John Hopkins Univesity.

Bridle, J. S. (2004). Towards better understanding of the model implied by the use of dynamic features in HMMs. In: *Proc. Int. Conf. on Spoken Lang. Proc.*, Jeju Island, Korea.

Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual web search engine. In: *Computer networks and ISDN systems* 30.1-7, pp. 107–117.

Champion, C. and S.M. Houghton (2016). Application of Continuous State Hidden Markov Models to a Classical Problem in Speech Recognition. In: *Computer Speech and Language* 36.C, pp. 347–364. ISSN: 0885-2308.

Choo, W. and M. Huckvale (1997). Spatial relationships in fricative perception. In: *Speech Hearing and Language: work in progress* 10.

Chorowski, J. K., D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio (2015). Attention-based models for speech recognition. In: *Advances in neural information processing systems* 28.

Cleland, J., C. McCron, and J. M. Scobbie (2013). Tongue reading: Comparing the interpretation of visual information from inside the mouth, from electropalatographic and ultrasound displays of speech sounds. In: *Clinical Linguistics & Phonetics* 27.4, pp. 299–311.

Davis, S. and P. Mermelstein (1980). Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 28.4, pp. 357–366.

Delahaye, D. and S. Puechmorel (2010). Air traffic complexity based on dynamical systems. In: *49th IEEE Conference on Decision and Control (CDC)*, pp. 2069–2074.

Denes, P. and E. Pinson (1993). *The speech chain: the physics and biology of spoken language*. (2nd ed). New York, W.H. Freeman and Company.

Deng, L. (2006). Dynamic speech models: theory, algorithms, and applications. In: *Synthesis Lectures on Speech and Audio Processing* 2.1, pp. 1–118.

Deng, L., M. Aksmanovic, X. Sun, and J. Wu (1994). Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states. In: *Speech and Audio Processing, IEEE Transactions on* 2.4, pp. 507–520.

Deng, L. and J. Chen (2014). Sequence classification using the high-level features extracted from deep neural networks. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6844–6848.

Deng, L., X. Cui, R. Pruvenok, J. Huang, S. Momen, Y. Chen, and A. Alwan (2006). A Database of Vocal Tract Resonance Trajectories for Research in Speech Processing. In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 1, pp. I–I.

Deng, L. and X. Li (2013). Machine learning paradigms for speech recognition: An overview. In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.5, pp. 1060–1089.

Deng, L. and J. Ma (2000). Spontaneous speech recognition using a statistical coarticulatory model for the vocal-tract-resonance dynamics. In: *The Journal of the Acoustical Society of America* 108.6, pp. 3036–3048.

Digalakis, V. (1992). Segment-based stochastic models of spectral dynamics for continuous speech recognition. PhD thesis. MA: Boston University.

Digalakis, V., J. R. Rohlicek, and M. Ostendorf (1993). ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. In: *IEEE Transactions on speech and audio processing* 1.4, pp. 431–442.

Doddipatla, R. (2016). Speaker adaptive training in deep neural networks using speaker dependent bottleneck features. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5290–5294.

Faisan, S., L. Thoraval, J. P. Armspach, and F. Heitz (2002). Hidden semi-Markov event sequence models: Application to brain functional MRI sequence analysis. In: *IEEE Proceedings. International Conference on Image Processing*. Vol. 1, pp. I–I.

Fant, G. (1970). *Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations*. 2. Walter de Gruyter.

Ferguson, J. D. (1980). Hidden Markov Analysis. In: *in Hidden Markov Models for Speech, Institute for Defense Analysis, Princeton, NJ*.

Flanagan, J.L. (1979). *Speech Analysis Synthesis and Perception*. Vol 1, 2nd ed. Springer Berlin Heidelberg.

Frankel, J. (2003). Linear dynamic models for automatic speech recognition. PhD thesis. Centre for Speech Technology Research, Edingurgh University.

Fry, D. B. (1979). *The Physics of Speech*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Furui, S. (1986). Speaker-independent isolated word recognition using dynamic features of speech spectrum. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34.1, pp. 52–59.

Gales, M. J. F. and S. J. Young (1993). Segmental Hidden Markov Models. In: *Proc. Eurospeech '93,* Berlin, Germany, pp. 1579–1582.

Garofolo et al., J. S. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Linguistic Data Consortium. Univ. Pennsylvania, Philadelphia, PA.

Ghahramani, Z. and G. E. Hinton (1996). Parameter estimation for linear dynamical systems. In: *Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science*.

Graves, A. and N. Jaitly (2014). Towards end-to-end speech recognition with recurrent neural networks. In: *International conference on machine learning*, pp. 1764–1772.

Graves, A., A. Mohamed, and G. Hinton (2013). Speech Recognition with Deep Recurrent Neural Networks. In: *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649.

Grezl, F., M. Karafiat, S. Kontar, and J. Cernocky (2007). Probabilistic and Bottle-Neck Features for LVCSR of Meetings. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing*. Vol. 4, pp. IV–757–IV–760.

Halberstadt, A. and J. Glass (1998). Heterogeneous measurements and multiple classifiers for speech recognition. In: *Proc. Int. Conf. on Spoken Lang. Proc.,* Sydney, Australia, pp. 995–998.

Hart, P. E., N. J. Nilsson, and B. Raphael (1968). A formal basis for the heuristic determination of minimum cost paths. In: *IEEE transactions on Systems Science and Cybernetics* 4.2, pp. 100–107.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. In: *the Journal of the Acoustical Society of America* 87.4, pp. 1738–1752.

Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In: *Neural networks: Tricks of the trade*. Springer, pp. 599–619.

Holmes, J. N., I. G. Mattingly, and J. N. Shearme (1964). Speech synthesis by rule. In: *Language & Speech* 7, pp. 127–143.

Holmes, J.N. and W.J. Holmes (2001). *Speech synthesis and recognition*. 2nd. London and New York: Taylor and Francis.

Holmes, W. J. (1997). Modelling Segmental Variability for Automatic Speech Recognition. PhD thesis. University College London.

Holmes, W. J. and M. Huckvale (1994). Why have HMMs been so successful for automatic speech recognition and how might they be improved. In: *Speech, Hearing and Language, UCL Work in Progress* 8, pp. 207–219.

Holmes, W. J. and M. J. Russell (1995). Experimental evaluation of segmental HMMs. In: *1995 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1, pp. 536–539.

Holmes, W. J. and M. J. Russell (1999). Probablistic-trajectory segmental HMMs. In: *Computer Speech and Language* 13.1, pp. 3–37.

Houghton, S. M., C. J. Champion, and P. Weber (2015). Recognition of voiced sounds with a continuous state HMM. In: *Sixteenth Annual Conference of the International Speech Communication Association*.

International, Phonetic Association (1975). *Journal of the International Phonetic Association*. v. 5.

Jackson, P. J. B. and M. J. Russell (2002). Models of speech dynamics in a segmental-HMM recogniser using intermediate linear representations. In: *Proc. Int. Conf. on Spoken Lang. Proc.*, Denver, CO, pp. 1253–1256.

Jaitly, N. (2014). *Exploring Deep Learning Methods for discovering features in speech signals*. University of Toronto (Canada).

Jelinek, F. (1969). Fast Sequential Decoding Algorithm Using a Stack. In: *IBM Journal of Research and Development* 13.6, pp. 675–685. ISSN: 0018-8646.

Jiang, B., Y. Song, S. Wei, J. Liu, I. V. McLoughlin, and L. Dai (2014). Deep bottleneck features for spoken language identification. In: *Public Library of Science San Francisco, USA one* 9.7.

Johns, D. (1975). *An outline of English phonetics*. 9th Edition. Cambridge University Press. ISBN: 978-0521290418.

Jurafsky, D. and J.H. Martin (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. In: *Journal of Basic Engineering* 82.1, pp. 35–45.

Lamel, L. F. and J. L. Gauvain (1993). High Performance Speaker-Independent Phone Recognition Using CDHMM. In: *Proc. Eurospeech '93,* Berlin, Germany, pp. 121–124.

Lee, K. and H. Hon (1989). Speaker-independent phone recognition using hidden Markov models. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.11, pp. 1641–1648. ISSN: 0096-3518.

McCandless, S. (1974). An algorithm for automatic formant extraction using linear prediction spectra. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 22.2, pp. 135–141.

McCulloch, W. S. and W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133.

Mohamed, A., G. Dahl, and G. Hinton (2009). Deep belief networks for phone recognition. In: *Nips workshop on deep learning for speech recognition and related applications*. Vol. 1. 9, p. 39.

Nagamine, T., M. L. Seltzer, and N. Mesgarani (2015). Exploring how deep neural networks form phonemic categories. In: *Sixteenth Annual Conference of the International Speech Communication Association*.

Natarajan, P. and R. Nevatia (2007). Coupled hidden semi markov models for activity recognition. In: *2007 IEEE Workshop on Motion and Video Computing (WMVC'07)*, pp. 10–10.

Nayak, S., S. Bhati, and K. S. Murty (2017). An Investigation into Instantaneous Frequency Estimation Methods for Improved Speech Recognition Features. In: *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 363–367.

Neiberg, D., G. Ananthakrishnan, and O. Engwall (2008). The acoustic to articulation mapping: Non-linear or non-unique? In: *Ninth Annual Conference of the International Speech Communication Association*.

Oller, D. K. (1973). The effect of position in utterance on speech segment duration in English. In: *The journal of the Acoustical Society of America* 54.5, pp. 1235–1247.

Oord, A., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In: *9th ISCA Speech Synthesis Workshop*, pp. 125–125.

Ostendorf, M., V. Digalakis, and O. A. Kimball (1995). From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition. In: *IEEE Transactions on Speech and Audio Processing* 4, pp. 360–378.

Paliwal, K. K. and P. V. S. Rao (1982). Synthesis-based recognition of continuous speech. In: *The Journal of the Acoustical Society of America* 71.4, pp. 1016–1024.

Paul, D. B. (1991). *An Efficient A* Stack Decoder Algorithm for Continuous Speech Recognition with a Stochastic Language Model.* Tech. rep. Massachusetts Inst Of Tech Lexington Lincoln Lab.

Petridis, S. and M. Pantic (2016). Deep complementary bottleneck features for visual speech recognition. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2304–2308.

Quillen, C. (2000). Adjacent Node Continuous-State HMM's. In: *Sixth International Conference on Spoken Language Processing*.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: *Proceedings of the IEEE*. Vol. 77. 2, pp. 257–286.

Rauch, H. E., C. T. Striebel, and F Tung (1965). Maximum likelihood estimates of linear dynamic systems. In: *AIAA journal* 3.8, pp. 1445–1450.

Ravanelli, M., P. Brakel, M. Omologo, and Y. Bengio (2018). Light Gated Recurrent Units for Speech Recognition. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 2.2, pp. 92–102.

Ravanelli, M., T. Parcollet, and Y. Bengio (2019). The PyTorch-Kaldi Speech Recognition Toolkit. In: pp. 6465–6469.

Reynolds, T. and C. Antoniou (2003). Experiments in speech recognition using a modular MLP architecture for acoustic modelling. In: *Information Sciences* 156.1-2, pp. 39–54.

Richards, H. B. and J. S. Bridle (1999). The HDM: a segmental Hidden Dynamic Model of coarticulation. In: *Proc. IEEE-ICASSP,* Phoenix, AZ, pp. 357–360.

Richmond, K. (2002). Estimating articulatory parameters from the acoustic speech signal. PhD thesis.

Rodríguez, L. J. and I. Torres (2003). Comparative study of the baum-welch and viterbi training algorithms applied to read and spontaneous speech recognition. In: *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, pp. 847–857.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. In: *Psychological review* 65.6, p. 386.

Rosti, A. V. I. and M. J. F. Gales (2001). *Generalised linear Gaussian models*. Cambridge University, Engineering Department.

Roweis, S. T. (1999). *Data-driven production models for speech processing*. California Institute of Technology.

Rumelhart, D. E., G. E. Hinton, R. J. Williams, et al. (1988). Learning representations by back-propagating errors. In: *Cognitive modeling* 5.3, p. 1.

Russell, M. J. (1993). A segmental HMM for speech pattern modelling. In: *Proc. IEEE-ICASSP,* Minneapolis, MN, pp. 499–502.

Russell, M. J. (2005). Reducing computational load in segmental HMM decoding for speech recognition. In: *Electronics Letters* 41.25, p. 1.

Russell, M. J. and P. J. B. Jackson (2005). A multiple-level linear segmental HMM with a formant-based intermediate layer. In: *Comp. Speech & Lang.* 19.2, pp. 205–225.

Russell, M. J. and R. K. Moore (1985). Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 10, pp. 5–8.

Sainath, T., B. Kingsbury, and B. Ramabhadran (2012). Auto-encoder bottleneck features using deep belief networks. In: *2012 IEEE international conference on acoustics, speech and signal processing*, pp. 4153–4156.

Schafer, R. W. and L. R. Rabiner (1975). Digital representations of speech signals. In: *Proceedings of the IEEE* 63.4, pp. 662–677.

Scherr, A. L. (1965). *An analysis of time-shared computer systems*. Vol. 535. Massachusetts Institute of Technology.

Schönle, P. W., K. Gräbe, P. Wenig, J. Höhne, J. Schrader, and B. Conrad (1987). Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract. In: *Brain and Language* 31.1, pp. 26–35.

Seivwright, C. (2015). *Repository: segmental-cshmm-public*. `https://bitbucket.org/uobsrbs/segmental-cshmm-public/src/master/`.

Senior, A., J. Subrahmonia, and K. Nathan (1996). Duration modeling results for an on-line handwriting recognizer. In: *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. Vol. 6, 3482–3485 vol. 6.

Shannon, C. (1953). The lattice theory of information. In: *Transactions of the IRE Professional Group on Information Theory* 1.1, pp. 105–107.

Shawker, T. H. and B. C. Sonies (1985). Ultrasound biofeedback for speech training: Instrumentation and preliminary results. In: *Investigative Radiology* 20.1, pp. 90–93.

Stevens, S. S., J. Volkmann, and E. B Newman (1937). A Scale for the Measurement of the Psychological Magnitude Pitch. In: *Acoustical Society of America Journal* 8, p. 208.

Sturtevant, D. (1989). A stack decoder for continous speech recognition. In: *Speech and Natural Language: Proceedings of a Workshop Held at Cape Cod, Massachusetts 1989*.

Sze, V., Y.H. Chen, T.J. Yang, and J.S. Emer (2017). Efficient processing of deep neural networks: A tutorial and survey. In: *Proceedings of the IEEE* 105.12, pp. 2295–2329.

Tan, S., K. C. Sim, and M. Gales (2015). Improving the interpretability of deep neural networks with stimulated learning. In: *2015 IEEE workshop on automatic speech recognition and understanding (asru)*, pp. 617–623.

Tan, X. and H. Xi (2008). Hidden semi-Markov model for anomaly detection. In: *Applied Mathematics and Computation* 205.2, pp. 562–567.

Toda, T., A. W. Black, and K. Tokuda (2008). Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model. In: *Speech communication* 50.3, pp. 215–227.

Tóth, L. (2014). Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 190–194.

Toutios, A. and K. Margaritis (2003). Acoustic-to-articulatory inversion of speech: A review. In: *Proceedings of the International 12th TAINN*.

Trentin, E. and M. Gori (2001). A survey of hybrid ANN/HMM models for automatic speech recognition. In: *Neurocomputing* 37.1-4, pp. 91–126.

Tüske, Z., R. Schlüter, and H. Ney (2013). Deep hierarchical bottleneck MRASTA features for LVCSR. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6970–6974.

Vanek, J., J. Zelinka, D. Soutner, and J. Psutka (2017). In: *Statistical Language and Speech Processing: 5th International Conference, Le Mans, France*. Springer, pp. 204–214.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In: *IEEE transactions on Information Theory* 13.2, pp. 260–269.

Weber, P., L. Bai, S.M. Houghton, P. Jančovič, and M.J. Russell (2016a). Progress on Phoneme Recognition with a Continuous State HMM. In: *Proc. IEEE-ICASSP,* Shanghai, China.

Weber, P., L. Bai, M. Russell, P. Jančovič, and S. Houghton (2016b). Interpretation of low dimensional neural network bottleneck features in terms of human perception and production. In: *Proc. Interspeech,* San Francisco, CA, USA, 3384–3388.

Weber, P., C. J. Champion, S. M. Houghton, P. Jančovič, and M. J. Russell (2015). Consonant recognition with continuous-state Hidden Markov Models and perceptually-motivated features. In: *Sixteenth Annual Conference of the International Speech Communication Association*.

Weber, P., S. M. Houghton, C. J. Champion, M. J. Russell, and .P Jančovič (2014). Trajectory analysis of speech using continuous state hidden Markov models. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3042–3046.

Wells, J. C. (1982). *Accents of English: Beyond the British Isles*. Vol. 3. Cambridge University Press.

Wells, J.C. (1967). A study of the formants of the pure vowels of British English. MA thesis. University of London.

Westbury, J. R., G. Turner, and J. Dembowski (1994). X-ray microbeam speech production database user's handbook. In: *University of Wisconsin*.

Wieworka, A. (1997). Speech recognition using Hidden Markov Models with exponential interpolation of state parameters. PhD thesis. Imperial College London (University of London).

Wrench, A. (1999). Mocha-timit. In: *Department of Speech and Language Sciences, Queen Margaret University College, Edinburgh, speech database*.

Young, S., G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, et al. (2002). The HTK book. In: *Cambridge university engineering department* 3.175, p. 12.

Young, S. J. (1992). The general use of tying in phoneme-based HMM speech recognisers. In: *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1, pp. 569–572.

Yu, S. (2010). Hidden semi-Markov models. In: *Artif. Intell.* 174, pp. 215–243.