1989

Vowel recognition in continuous speech

Darrell C. Stam

R.I.T.

Rochester Institute of Technology

School of Computer Science

Vowel Recognition in Continuous Speech

By

Darrell C. Stam

A thesis, submitted to the Faculty of the School of Computer Science, in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science.

Approved by:

9/22/89

Dr. James Hillenbrand

John A. Biles

Dr. Peter Anderson

I hereby *waive* all copyrights permission to the Wallace Memorial Library at the Rochester Institute of Technology.

17 Nov 89

Table of Contents

Abstract

Keywords:
Artificial Intelligence, Speech Recognition, Signal Processing,
Speech Analysis, Vowel Classification, Neural Networks,
Hidden Markov Model, Maximum Likelihood Distance Measure.

In continuous speech, the identification of phonemes requires the ability to extract features that are capable of characterizing the acoustic signal. Previous work has shown that relatively high classification accuracy can be obtained from a single spectrum taken during the steady-state portion of the phoneme, assuming that the phonetic environment is held constant. The present study represents an attempt to extend this work to variable phonetic contexts by using dynamic rather than static spectral information. This thesis has four aims: 1) Classify vowels in continuous speech; 2) Find the optimal set of features that best describe the vowel regions; 3) Compare the classification results using a multivariate maximum likelihood distance measure with those of a neural network using the back-propagation model; 4) Examine the classification performance of a Hidden Markov Model given a pathway through phonetic space.

Chapter 1

Speech Understanding

The Speech Recognition Problem

Designing a computer that understands spoken words is something speech scientists and artificial intelligence researchers have been working on for several decades. Already there are some systems that are capable of transforming acoustic input into an action desired by the speaker. Unfortunately, most of these systems are limited in their performance of tasks and understanding of speech. Several factors that contribute to the complexity of this speech recognition problem are: speaker dependence versus independence, discrete word versus continuous speech, and vocabulary size.

Speaker Related Systems

Speaker dependent recognition systems are simpler to develop than systems that can recognize speech independent of the speaker. Most unsophisticated speech recognition systems use a form of template matching to isolate words. Using a unique template for every word is possible, but as Figure 1-1 shows, speakers differ from one another in the way they say the same word. Notice how the word "shoe" spoken by two different people can differ in its spectrographic appearance. Some variations can be found in every word spoken by different people, thus making it nearly impossible to recognize the word with a single template. Realizing that speaker dependent recognition systems are limited, due to the number of speakers given access to the system and the search space required to find the matching template, work is being conducted to transform these systems to ones that are more independent of the speaker. Programming problems are inherent when multiple users have access to speech recognition systems. These range from the various acoustically diverse accents of speakers

down to the sex of the speaker, with each person producing different spectral patterns based upon the shape of the vocal tract.

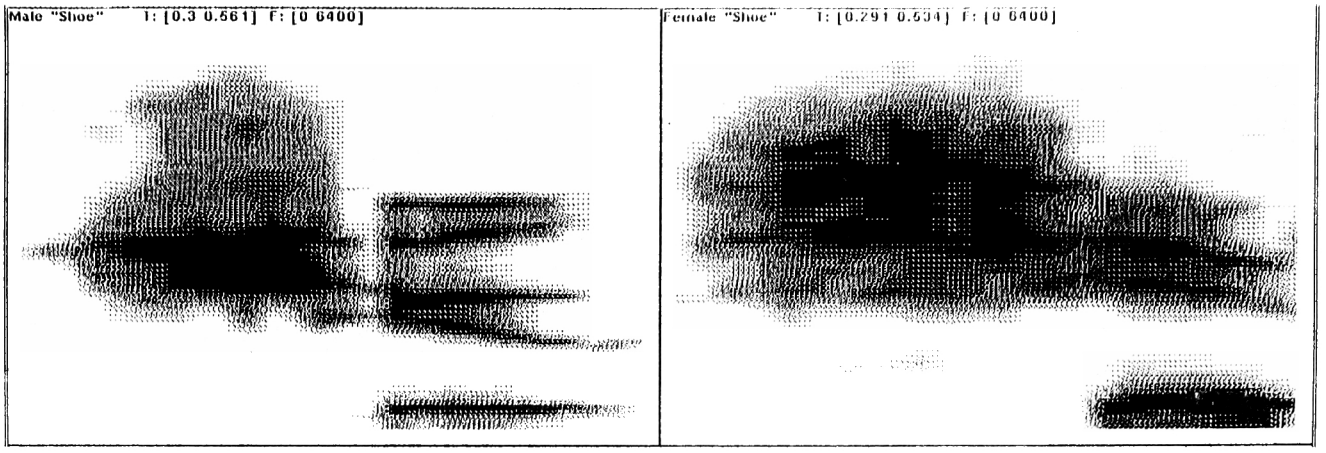


Figure 1-1 The word shoe spoken by two different speakers.

Continuous Speech

Seldom is there clear acoustic evidence of phonetic boundaries. Displayed in Figure 1-2 are the continuous utterances "Athletic events jam the tv on Saturday afternoons" and "Toast and jam tastes good for breakfast", labeled A and B respectively. Notice in both of the spectrograms how hard it is to locate the end of one word and the beginning of the next. An aid in determining word boundaries can be found with both the time aligned waveform and phonetic transcriptions; though note that the transcriptions were produced by using the spectrogram and being familiar with the utterance. In spectrogram A it is not obvious where the final voicing of /ey/ in Saturday finishes and the initial /ae/ of afternoon starts. Observe how the formant transitions are continuous from one vowel to the next without any distinguishing pauses or breaks. The acoustic waveform does not offer much help either. Another problem found in using continuous speech is the amount of time and stress placed on individual words. In both of the utterances the word 'jam' appears between two timed markers specifying the starting and stopping interval for each word. Even though the underlying formant transitions

are similar, it is difficult to find the exact correlation. One reason stems from the amount of power (loudness) used by the speaker to articulate the utterance. Another can be found in the duration of the word. Thus to alleviate the variety of temporal problems found in segmentation, most current speech understanding systems use discrete words rather than continuous speech when attempting to interpret the intent of the speaker.

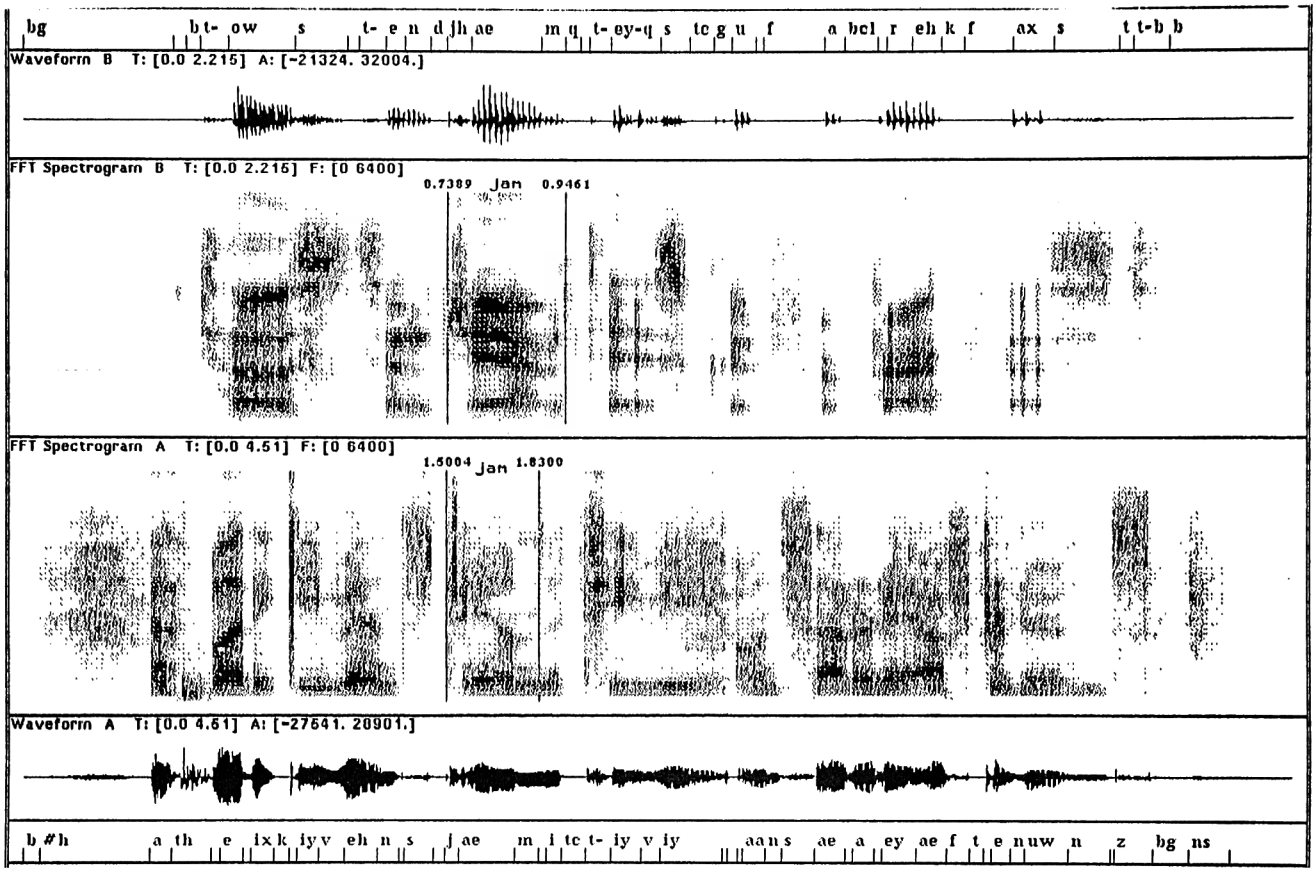


Figure 1-2 Continuous utterances used to demonstrate segmentation problems.
Sentence A is "Toast and jam tastes good for breakfast".
Sentence B is "Athletic events jam the TV on Saturday afternoons".

Vocabulary Size

Most speech recognition systems use a limited vocabulary for several reasons. Mainly, a small vocabulary is easier to manage than a larger one because there are fewer chances of similar spectral patterns occurring. Given that the vocabulary of an average person is in the range of hundreds of thousands of words, especially if proper names or technical terms are included, a large vocabulary would cause a search space problem as well as a discriminability problem. These problems would require a costly search through the entire dictionary for such homonyms.

RITRC Speech Understanding Architecture

The Rochester Institute of Technology Research Corporation (RITRC) is taking a different approach to solving these problems than normal speech understanding systems. Instead of representing each word by its acoustic symbol words in their system take on a phonetic semblance. At a high level of representation, a phonetic dictionary is cheaper to maintain than an acoustic dictionary which models words by vector quantized symbols in lower level representations. A significant difference is that a phonetic representation would require a smaller search space. Thus, this approach results in a system that is capable of supporting a large vocabulary. However, this method requires a system for translating the speech signal into a string of discrete phonetic symbols. The RIT Research Corporation is in the process of developing a speech recognition system that will be capable of working within the large vocabulary, continuous speech, speaker-independent domain. The ultimate goal is to demonstrate an end-to-end system starting from the acoustic waveform and ending with a knowledge representation of an utterance. This architecture provides a framework both for demonstrating speech understanding techniques and comparing them with more traditional methods of speech and signal recognition. The software architecture incorporates a knowledge-based system that attempts to capture the knowledge that experts use when reading and interpreting spectrograms (Figure 1-3).

The speech understanding process starts at the lowest level with a digitized sample of speech, which is processed by using standard signal processing

algorithms, such as a fast Fourier transform (FFT) and linear predictive coding (LPC), to obtain low-level features such as formant and pitch tracks, energy measures, zero-crossing rates, etc. These features then are used as input into a series of classification modules that attempt to segment and recognize phonemes from the speech sample [NAIC88].

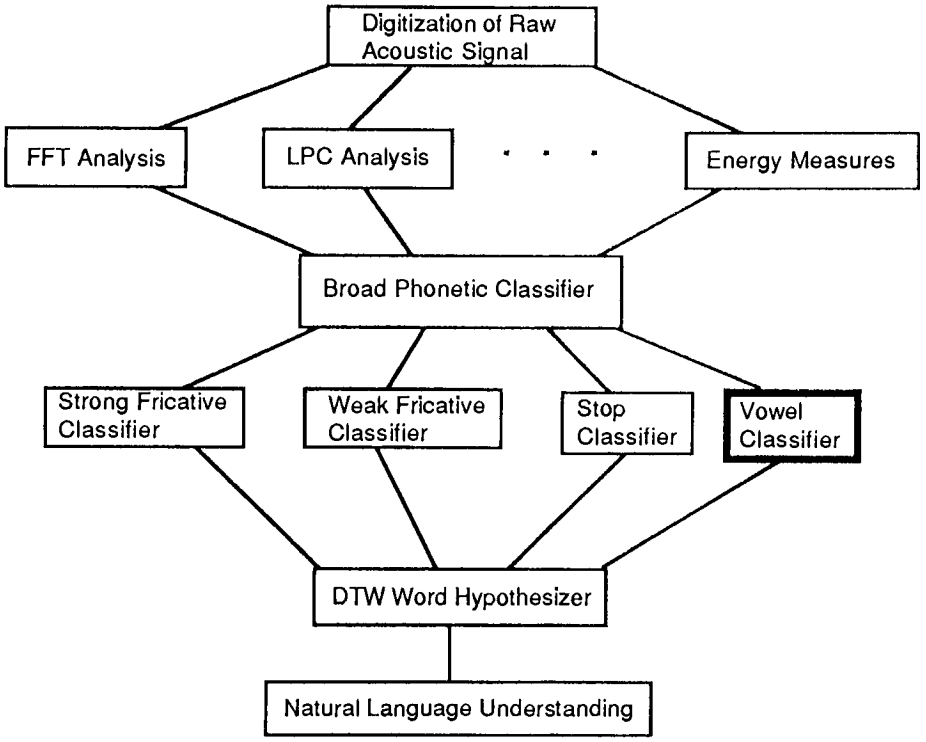


Figure 1-3 RITRC Speech Understanding Architecture

Once the features have been extracted, the broad phonetic classification module divides the signal into discrete segments based on the broad categories "vowel-like", "strong fricative", "weak fricative" and silence. "These segments can be thought of as regions of the utterance that are roughly homogeneous" [NAIC88 p4]. These broad phonetic segments then are analyzed further up in the architecture by modules that attempt to assign phonetic labels to these broad segments.

Once the lattice of phonetic labels has been generated, a high-level module hypothesizes word candidates from a phonetically based lexicon. Currently, the Research Corporation has investigated the use of Dynamic Time Warping (DTW), incorporated with knowledge of the English language to hypothesize words.

Probabilistic values of common phonetic errors such as insertion, deletion, and substitution also have been taken into account at this stage [NAIC88].

Located at the most advanced level in this architecture is a natural language understanding system. This module attempts to build a semantic representation of the input utterance using high level knowledge sources including domain knowledge, syntactic and semantic information, and acoustic cues such as pauses and inflection. The final output of the system will be a model that captures the intent of the spoken utterance and not necessarily a system that recognizes each individual word.

The Vowel Classifier

Once the broad phonetic categories have been determined, the goal of this thesis is to analyze and assign phonetic labels to each vowel-like segment based upon the information extracted from the acoustic features (see dark box Figure 1-3). This assignment of phonetic labels is the result of several stages of processing. The first stage requires performing several signal processing routines upon the acoustic signal. The specific routines, formant and pitch trackers and four spectral moments, among others, have been selected based upon their ability to characterize the various vowel regions. The vowel regions along with the subsequent characteristic features then are extracted from the acoustic signal and fed into a pre-classifier. The performance of two separate pre-classifiers, a neural network running the back-propagation model and a maximum likelihood distance measure, are compared. Their objective is to produce a pathway through phonetic space so that at the final stage a hidden Markov model can make the assignment of phonetic labels to the vowel region. The intent of using pre-classifiers is to reduce the time involved in running the hidden Markov model since a Markov model is computationally expensive, and it is not cost effective to present every feature to the model for training purposes. The output of this thesis is a vowel phoneme that is the closest match to the spoken vowel segment based upon the acoustic characteristics that were presented to the system.

Chapter 2

Phonetics

Phoneme Variability

An individual's speech production system has the ability to produce a near infinite number of sounds. However, in the English language there are approximately 42 basic sound units called phonemes. Believing that each phoneme possess certain characteristics that make it distinguishable from other phonemes, speech scientists have set out to identify these traits. Being able to identify these characteristics, segmenting speech, and then extracting a specific phoneme from a speech signal are key issues in the process of recognizing the spoken phoneme. Shankweiler et. al. [SHAN75] point out that there are no apparent bounding segments in the acoustic waveform that would single out a specific phoneme. Even if boundaries are arbitrarily imposed upon the signal, segments corresponding to a particular phoneme often vary considerably in composition. Variability arises from the differences found among every speaker; just as in handwriting where no two signatures are identical, no two utterances are acoustically the same [SHAN75]. This unique property of speech results from the frequency range of the speaker as well as the rate of speech, volume, accent, gender, and the size and configuration of the vocal tract.

Coarticulation

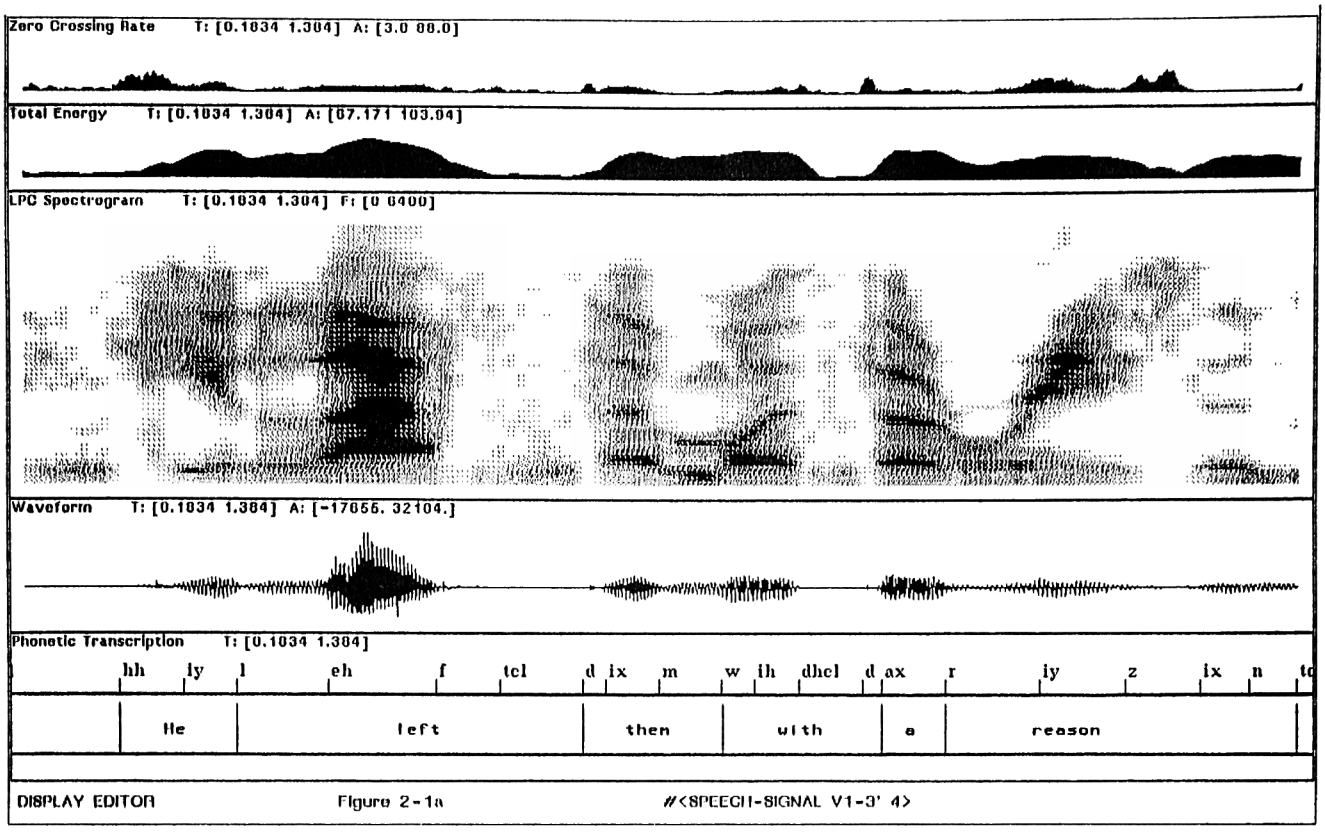
"Phonemes are not merely joined acoustically; they overlap so that two or more phonemes can be represented simultaneously on the same stretch of sound" [SHAN75 p123]. This source of variability is called coarticulation. Coarticulation occurs when the characteristics of one phoneme change as a result of the surrounding phonemes. The best way to explain this effect is to view these

differences in a spectrogram. Figure 2-1 shows a female speaker's recording of the sentence "He left them with a reason to believe in themselves," with the time-aligned orthographic and phonetic transcription below. A spectrogram provides a display of the relevant temporal and spectral characteristics of a waveform by plotting the frequency along the vertical axis, time across the horizontal axis and the intensity or amplitude of the speech waveform represented by darkness.

Vowels are visually identifiable in the spectrogram by their formant frequencies which appear as dark bands of energy. "The frequency locations of the formants, especially the first two, labeled F_1 and F_2, are closely tied to the shape of the vocal tract as the lips, tongue, pharynx and jaw move to articulate the consonants and vowels" [PICK80 p46].

The illustrated sentence contains several examples of how local context can influence the acoustic properties of the underlying phonemes. Most noticeable is the semivowel /l/ and back-vowel /iy/ as in the word "believe". /l/ typically has a low F_2 value, but when the /iy/ precedes it as in the words "He left", F_2 has a noticeably higher than normal value. Notice how this transposition is different from the behavior of both /l/ and /iy/ when the former proceeds the latter (Figure 2-1b). The location of stress placed on a word also influences the spectral behavior of various phonemes. In the two occurrences of /eh/ in the word "themselves", especially note how F_2 in the first occurrence is extremely weaker to that of the second /eh/. The overall formant transitions are somewhat similar but definitely not identical to the /eh/ in the word "left", where the formants tend to be smeared across the segment boundaries.

These examples illustrate how even with the same phoneme it is difficult to identify similarities between two or more occurrences in continuous speech. Although the number of phonemes that comprise the English language is relatively small, identifying them is not a simple job of matching a small set of well defined patterns, or templates. It involves a much deeper understanding of the acoustic properties of phonemes and their interaction with each other [ZUE 85].



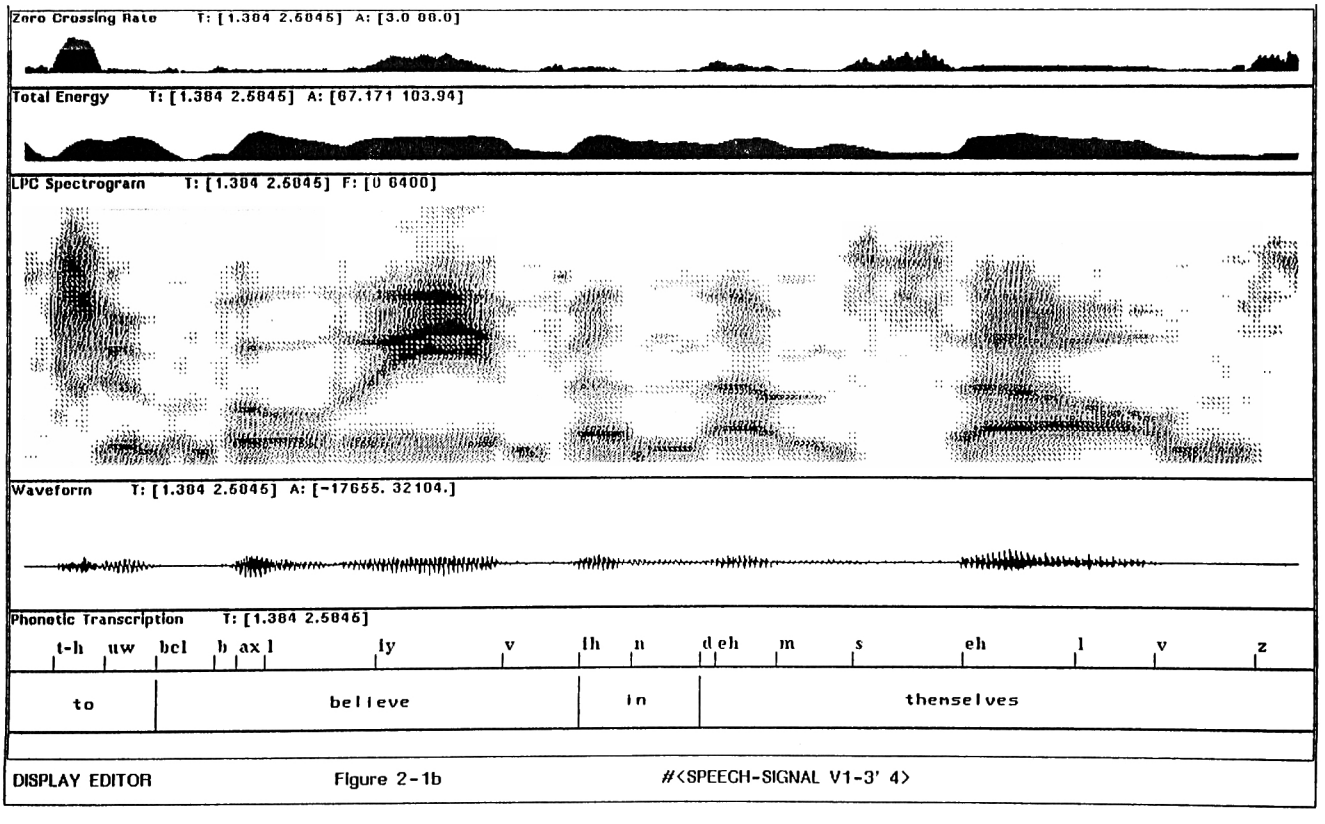


Figure 2-1 Influence of local context on the phonemes

Spectrogram Reading

Cole et. al. [COLE80] attempted to determine whether a spectrogram could be read with a high degree of accuracy. The spectrogram reader was Victor Zue, who had trained himself to read spectrograms over a period of many years. Zue was asked to read spectrograms from three categories consisting of isolated words, normal sentences (e.g. "Go paddle your own canoe."), and phonetically and semantically anomalous utterances, (e.g. "Give mine yadiya of his hate raret.") These utterances were unknown to Zue and were produced by unknown speakers. The experiment involved comparing Zue's phonetic transcriptions based upon a visual representation to those of three other phoneticians who listened to the same utterances.

In reading spectrograms, Zue's first task was to determine the phonetic boundaries within each utterance. This is known as segmentation. Secondly he labeled each segment with the phonetic symbol that best matched the spectrogram segment based upon the characteristic features that were displayed. For isolated words, Zue's segmentation matched that of the three phoneticians 100% of the time, and he was 97% accurate for continuous speech. In labeling consonants, Zue was 89% accurate, with 79% accuracy for labeling vowels. It is interesting to note that Zue's performance did not decrease with the semantically anomalous utterances, and in fact his performance improved with the use of a carrier phrase such as "Say _____again." His success rate proves that phonemes can be recognized from speech spectrograms since enough information can be obtained from the acoustic signal.

When presented a continuous utterance, Zue was reasonably adept in finding the characteristic features that enabled him to identify the various vowel regions. This achievement is quite remarkable since coarticulated phonemes, especially vowels as seen in Figure 2-1, are not distinct in their spectrographic appearance. In order to appreciate how Zue was capable of identifying vowels, it is appropriate to investigate how different vowel sounds are produced from the combination of the vocal tract and glottal source.

Chapter 3

Production, Acoustics and Perception of Vowels

Source-Filter Theory

"The production of vowels may be discussed in terms of the acoustical properties of the sound source [produced at the larynx] and the modification of that source by the acoustical filtering which takes place within the vocal tract" [MINI73 p237]. The source-filter theory of speech production depends on three factors: a source of energy, a vibrating body, and a resonator (Figure 3-1).

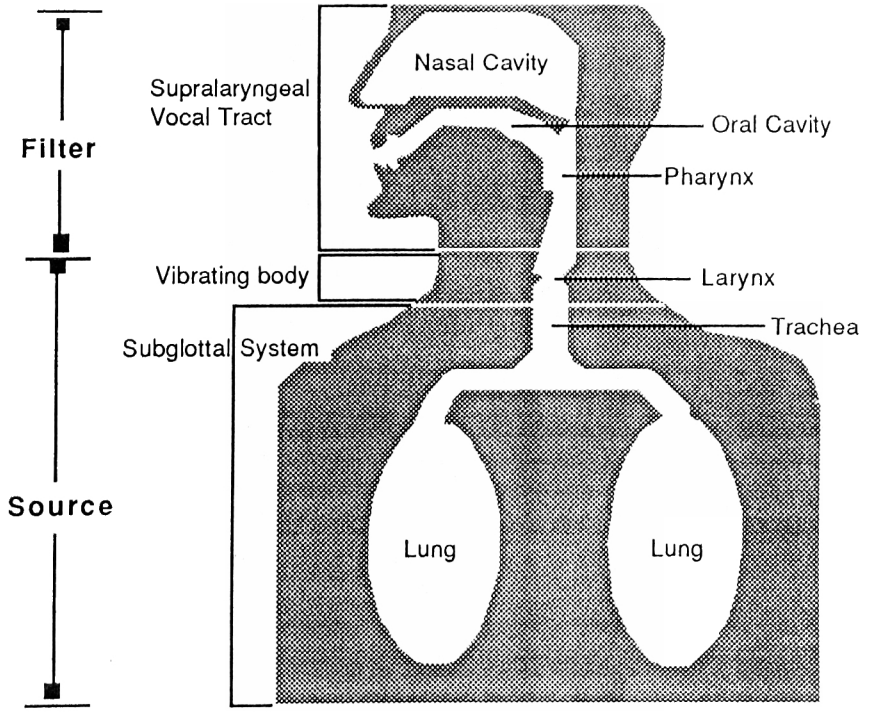


Figure 3-1 Physiological Components of Human Speech Production
(Adapted from [LIEB77]).

The Source

In speech, vowel sounds are produced by repeated acoustic excitation of the vocal tract air column with a series of glottal pulses. Air flows up from the lungs through the vocal folds, into the pharyngeal and oral tracts and out the mouth. This action usually occurs in pulsation and "acts as an outward airflow disturbance and so it is propagated back toward the vocal fold surfaces" [PICK80 p23]. The vocal folds act like the bottom of a bottle and reflect the pulse back upward again in a periodic fashion. These folds can be opened for normal breathing, completely closed, or partially open (Figure 3-2). The opening between the vocal folds is called the glottis. When the glottis is partially open, air is allowed to escape from the lungs, but the output of air causes the folds to vibrate [WILD75]. This phenomenon occurs during voicing.

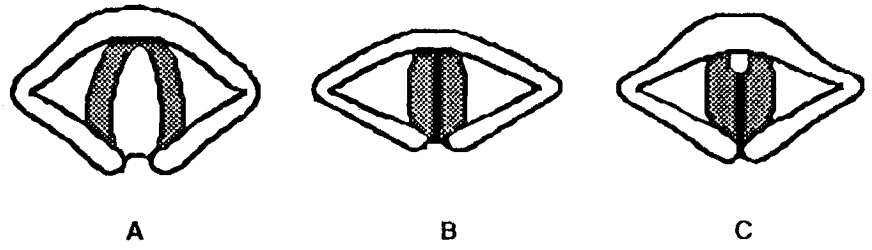


Figure 3-2 Position of the vocal folds in (a) normal breathing, (b) complete closure, and (c) voicing *(Adapted from [WILD75]).*

The degree of tension applied to the vocal folds controls the frequency of vibration [WILD75]. Along with the subglottal air pressure, the glottal tension affects the characteristics of the sound source, especially the pulsing rate, but it also affects the source amplitude and source spectrum [PICK80]. By increasing the subglottal air pressure, the rate of repetition of airflow pulses increases; that is, the fundamental frequency (pitch) increases. Pitch is also affected by the amount of stress placed upon certain syllables. For expressing a question, the pitch contour rises during the utterance due to the increased vocal fold tension. In the same fashion the stressed syllable then will have a higher pitch than the other syllables. By increasing the subglottal pressure, the intensity of the sound from the glottal source increases and consequently raises the high frequency range of the source spectrum relative to the low frequency range. The changes in tension and pressure for stressed vowels produce a higher second formant amplitude and

overall higher formant frequency values than for unstressed vowels. This observation must be noted when attempting to recognize vowels in various contexts.

The Filter

The filter's shape, determined by the length of the pharyngeal and oral passages, determines the resulting vowel sound. When a vowel is spoken, the vocal-tract has a tubular shape, and the sound produced contains certain resonating patterns. As displayed in Figure 3-3, when applying a specific filter response of the vocal tract to the glottal sound spectrum, a particular vowel spectrum is formed based upon the filter. It is important to note "that the spectrum of the glottal source is the same for all the different vowels and this sound spectrum is then changed by the filtering of the vocal tract to produce vowel sounds that have different spectral patterns" [PICK80 p67]. The shape of the spectrum envelope of the vowel spectra stems from both the glottal spectrum and the resonant peaks of the vocal tract. Thus, when some other vocal tract shape is formed, a different pattern of vowel sounds are formed as can be seen in the changing spectrum envelopes.

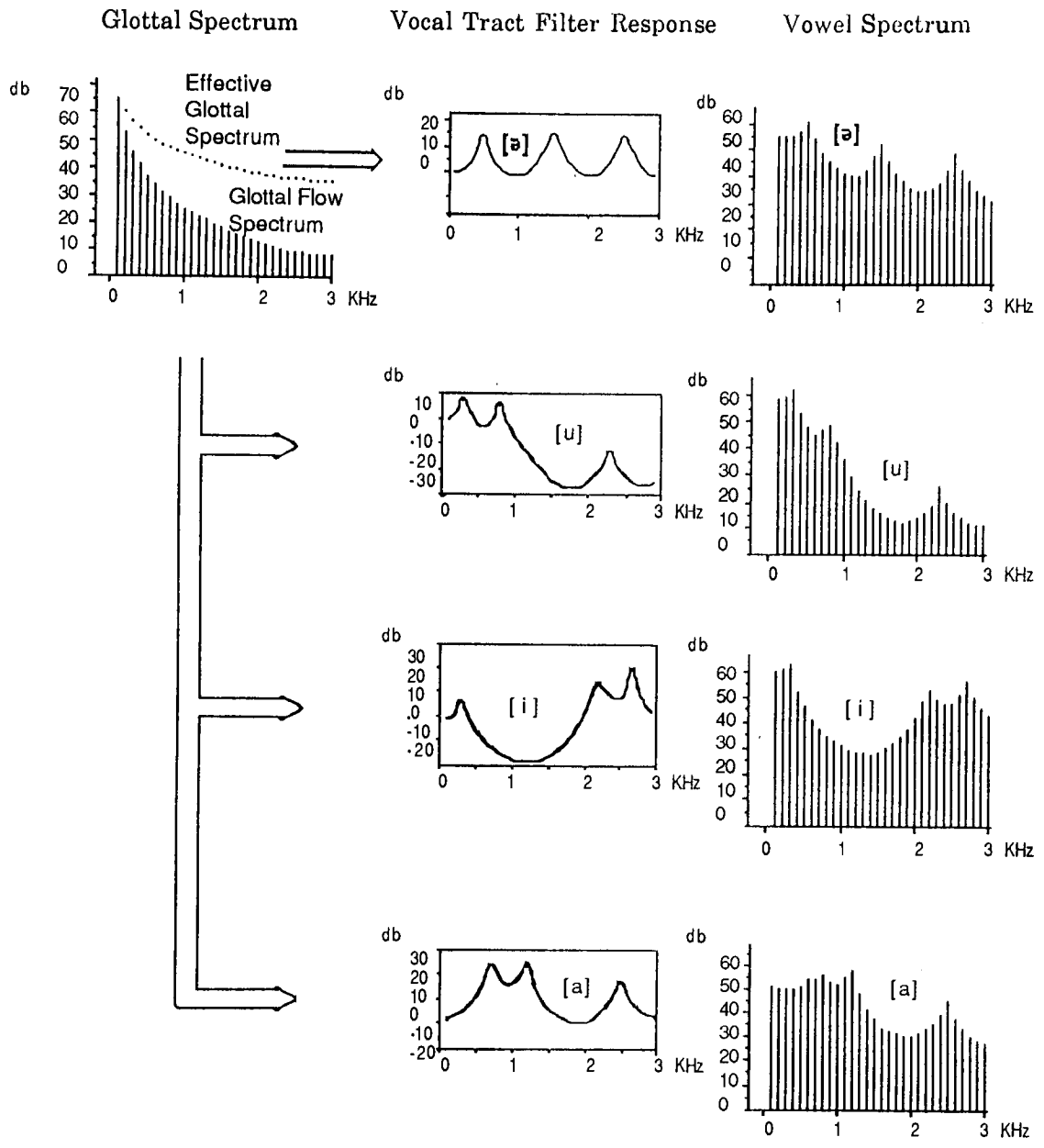


Figure 3-3 Filter Responses of the vocal tract to produce various vowel spectra *(Adapted from [PICK80]).*

In the average male, the vocal tract "tube" has a length of 17.5 cm with the glottis located at one end and the lips at the other (Figure 3-4). The voicing sound that resonates during the sound transmission through the vocal tract is called formants. "The effects of these vocal resonances are apparent in the spectrum of a speech sound, and the spectrum peaks may be called the *formants* of the speech sound, but this is not completely correct because it is not the sound that has formants or resonances, it is the vocal tract " [PICK80 p45].

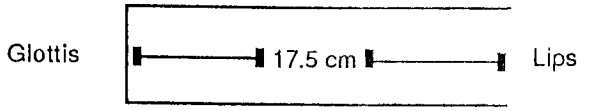


Figure 3-4 Vocal Tract 'Tube'

"As the size and shape of the resonance cavities are altered during speech production, formant frequencies are also changed so that every configuration of the vocal tract has its own characteristic formant frequencies " [WILD75 p37].

Vowel Production

Vowels are produced by modifying the source by the acoustical filter. This event takes place within the vocal tract and is caused by the various combinations of the tongue, lips and jaw. It is the location of the tongue that enables us to classify different vowel sounds based upon the point of constriction in regards to its position (front, back, central) and height (high, mid, low). As seen in Figure 3-5, the point of maximum tongue constriction is located near the alveolar ridge for the front vowels /iy,ih,ey,eh,ae/. (Table 1, in Appendix A, contains the full phonetic listing of vowels in both the International Phonetic Alphabet (IPA) and in arpabet symbols.) For the back vowels /uw,uh,ow,ao,aa/ the highest point of the tongue is near the velum. The remaining vowels sounds /ix,ax,ah/ are produced when the tongue is in a more central position. The degree of constriction is determined by the height of the tongue. A case in point occurs with high vowels such as /iy/ and /uw/ which are highly constricted, whereas low vowels such as /ae/ and /aa/ are not. Figure 3-6 shows the locations of various vowels based upon these positions in what is known as a vowel triangle.

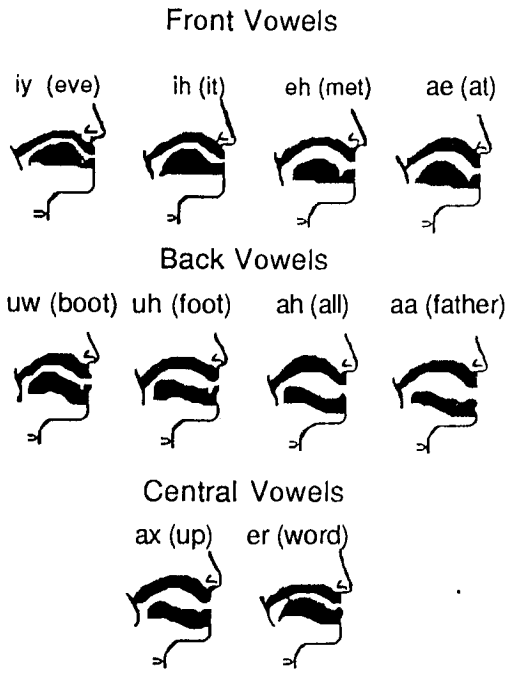


Figure 3-5 Mouth Profiles *(Adapted from [WILD75])*

In addition to the effect of tongue position, the relative roundness (rounded or unrounded) of the lips and tension of the tongue muscles contribute to the characteristics of each vowel. Rounded vowels /uw,uh,ow/ are not only constricted in the mouth, but also at the lips.

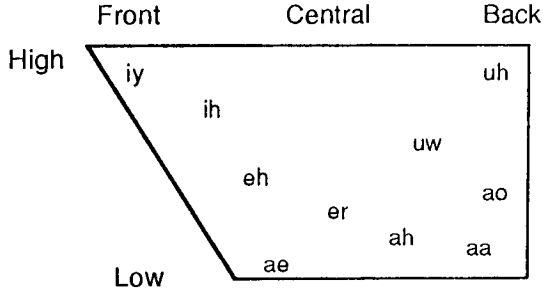


Figure 3-6 Vowel Triangle

Diphthongs

A diphthong is produced when the speaker glides the tongue continuously from one vowel position to another, as in the word *'boy'*. When the articulatory posture of the tongue changes, so does the position of the formants. Diphthongs contain two types of ascending and descending glides making neither element phonetically identifiable with any stressed monophthong [LEHI61]. Figure 3-7 displays the transitions of the first two formant frequencies for the diphthong /oy/. The second formant transition is characteristic of the diphthongs.

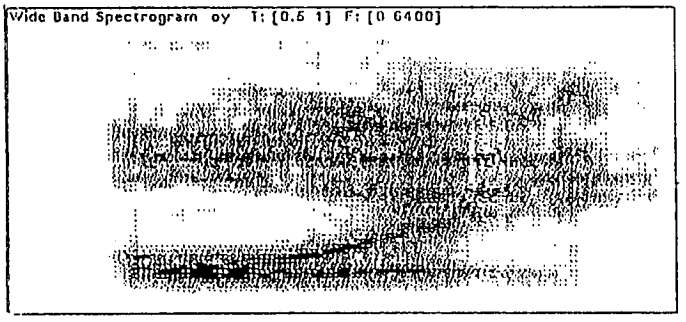


Figure 3-7 Diphthong formant movements

Semi-Vowels

Semi-vowels /l,r,y,w/ are produced with a greater degree of constriction than vowels but are less constricted than consonants. The production is characterized by the vibration of the vocal folds giving them a voiced sound, and can be recognized by the transitional cues of the second and third formants (Figure 3-8). Semi-vowels can be divided into glides /y,w/ and liquids /l,r/.

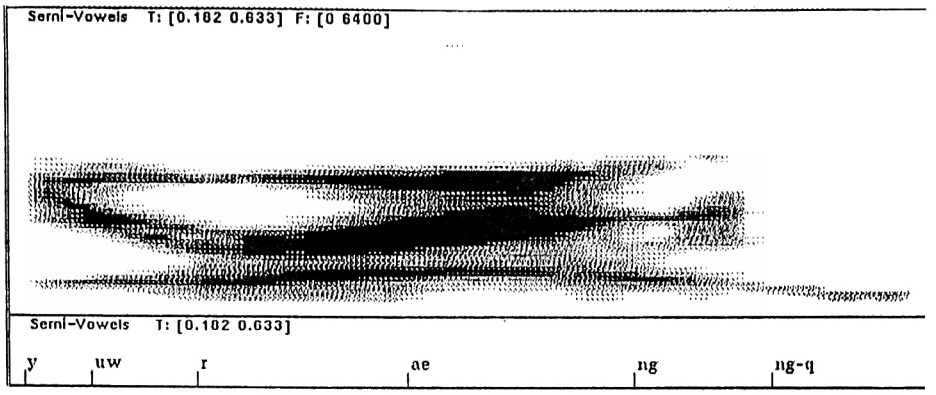


Figure 3-8 Semi-Vowel formant movements

The "production of glides requires the movement of the tongue and lips to change the vocal tract shape from the starting position to the next vowel position" [BORD80 p109]. A /y/ starts with a high front tongue, and /w/ starts with a high back tongue with protruded lips.

> The liquids /l,r/ are produced in syllable initial position by raising the tongue toward the alveolar ridge while voicing. Differences in the tongue tip configuration and position create the distinctions between the two sounds. For /l/, the tip [of the tongue] is resting lightly against the alveolar ridge dividing the pressure waves into two streams which emerge at each side. For /r/, the tongue is grooved and does not contact the aveolar ridge, so the acoustic energy emerges centrally.... Many speakers retroflex the /r/ which means the tongue tip is pulled back further and tensed [BORD80 p110].

The main effect of retroflexion is an unusually low third formant value.

Vowel Nasalization

"Vowel sounds of English are [generally] spoken with the velum raised against the walls and back of the pharynx to shut off the nasal passages completely from the pharynx and oral tract" [PICK80 p73]. By lowering the velum, and leaving the entrance to the nasal cavities open, a nasalized sound can be produced. The addition of the nasal branches to the vocal tract creates a larger, longer resonator. The effects of the longer resonator is seen by lower formant frequency values. "Nasalization also occurs in vowels that are adjacent to nasal consonants, especially in the portions of the vowel immediately next to the

consonant" [PICK80 p73]. This causes an attenuation of upper formants relative to those of the neighboring vowels. The damping of the resonances is partially a result of the broader band frequency response set up in the elongated tract. Another reason why nasals suffer a loss in intensity is that sound is absorbed by the soft walls and convolutions within the nasal cavities. "This causes a reduction in F_1 amplitude and insertion of antiresonances and extra formants in the transmission of the pharyngeal-oral-tract, thus altering the normal vowel spectrum" [PICK80 p77].

Vowel Acoustics

As discussed above, the frequency location of the formants are closely tied to the shape of the vocal tract as the filter is transformed to articulate vowels and consonants [PICK80]. The most thorough and widely cited study of vowel formant frequency patterns was conducted by Peterson and Barney [PETE52]. The first part of their study investigated the relationship between the phoneme intended by a speaker and that identified by the listener. Peterson and Barney made a list of ten monosyllabic words each beginning with /h/ and ending with /d/ and differing only in the vowel. The words used were heed, hid, head, had, hod, hawed, hood, who'd, hud, and heard. A group of 76 speakers, which included 33 men, 28 women, and 15 children, each recorded two lists of ten words to produce a total of 1520 recorded words. These words were then randomized and presented to a group of 70 listeners, 32 of whom were from the original set of 76 speakers. Listeners were asked to classify the words into one of the ten possible categories based upon the vowel sound they heard. The overall classification accuracy by the listeners was 94.4%, which implies that vowels are highly identifiable by human listeners.

In addition to the classification performed by listeners, the second half of their study involved analyzing the words by means of the sound spectrograph. Peterson and Barney made measurements of both the frequency and amplitude of the formants for the 20 words recorded by each of the 76 speakers. These measurements were made during the steady state part of the vowel "following the influence of the /h/ and preceding the influence of the /d/" [PETE52 p177]. By plotting F_1 against F_2 (Figure 3-9), it is discouraging to see that the 10 English

vowels are not capable of being strongly separated. By examining the formant frequency diagram it is unclear how a listener is capable of successfully classifying vowels with an error rate as low as 5 1/2 percent based upon the labeled "hoops". Each of these hoops encompass about 90 percent of the data points for each phonetic category. Notice that some of the hoops overlap and not all of the vowels are clustered within their correct vowel category. This finding makes their study so widely cited since they demonstrated that human listeners must use more information than what is present in the first two formant frequencies.

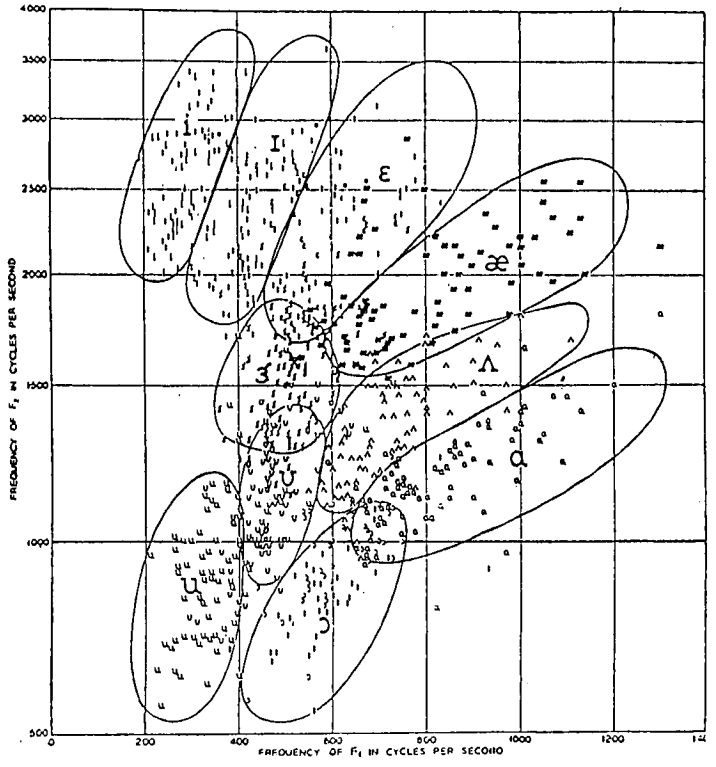


Figure 3-9 Frequency of second formant versus frequency of first formant *(Adapted from [PETE52]).*

Effects of Coarticulation

Although Peterson and Barney's measurement data showed a relatively strong relationship between vowel identity and formant frequency patterns, it is important to recognize that all vowels were produced in single words and in the same phonetic environment. It is well known that factors such as speaking rate, linguistic stress and phonetic environment can have very strong effects on the formant frequency patterns of vowels [HILL89]. In a controlled state, such as in

Peterson and Barney's study, each vowel is characterized by its target formant values within an acoustic "vowel space".

> Thus, for example, if a vocal-tract configuration for a vowel is realized by instructing the tongue mass to move in a particular direction, the actual displacement of the tongue mass will lag behind the instruction, and will not approach the specified position until a certain time has elapsed. During the production of a syllable, therefore, when forces are applied to a given component of the system in order to cause a maneuver from a consonant configuration to a vowel configuration and back, ... the displacement that results from this combination of forces may fall short of the displacement associated with the ideal target vowel configuration, ... [STEV63].

Stevens and House [STEV63] conducted a study on the perturbation of vowels in 14 different consonantal contexts. Using only three male speakers, they were able to show that systematic shifts in the vowel formant frequencies depend upon the particular vowel, the voicing characteristics and the manner and place of production of the adjacent consonants. The effects of place of articulation of the consonantal context on vowels is shown in Figure 3-10.

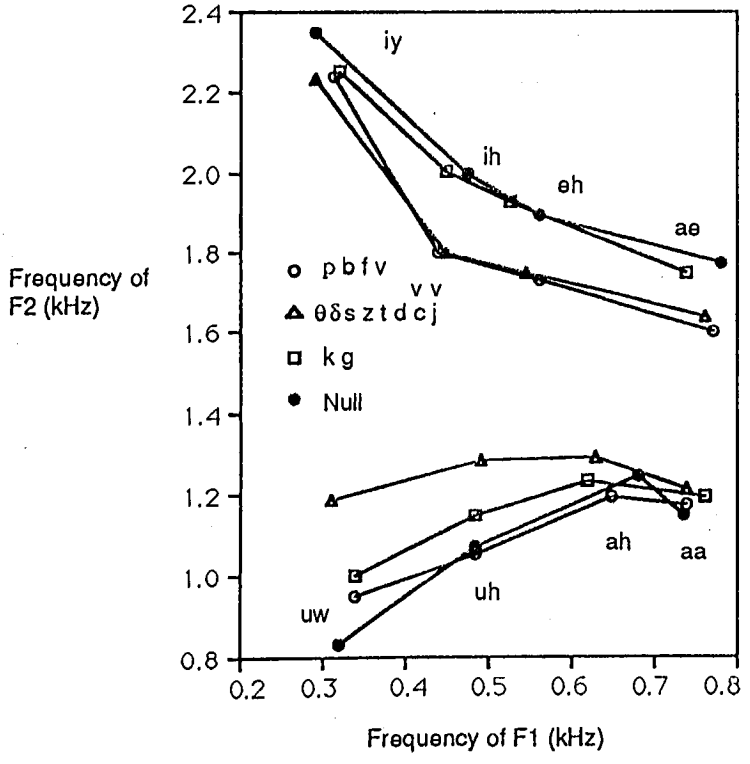


Figure 3-10 The effects of place of articulation on consonantal context *(Adapted from [STEV63]).*

In this figure, the values of the first two formants for eight different vowels are averaged for the 3 speakers and then are plotted to demonstrate this effect. The consonantal contexts included are velars, postdentals, labials and the null environment (an average of the isolated vowel as well as /h-V-d/ contexts), which are represented by squares, triangles, hollow circles and filled circles, respectively. As this figure shows, the average shift in F_1 is small for different places of articulation of the consonant. On the other hand the changes in F_2 are clearly apparent. The amount of deviation for F_2 from the null environment depends on the particular vowel being examined. The amount of downward F_2 shifting is least noticeable for velar consonants spoken with front vowels, and labials spoken with the back vowels. Conversely the greatest amount of variation in F_2 occurs when front vowels are joined with the labials and postdentals. Deviation of F_2 also occurs when back vowels are joined with the post dentals.

By examining vowel formant frequencies according to the voicing characteristic and manner of production of the consonantal context, the effects shown in Figure 3-11 demonstrates the displacement of F_2 toward a more central position. Stevens and House showed that when the consonant is voiced, the F_1 values for a vowel are generally lower than those for a vowel following a voiceless consonant. It can be noted that F_2 remains relatively unchanged for back vowels and is slightly higher for front vowels. Low F_2 values can be thought of as undershooting the ideal target position.

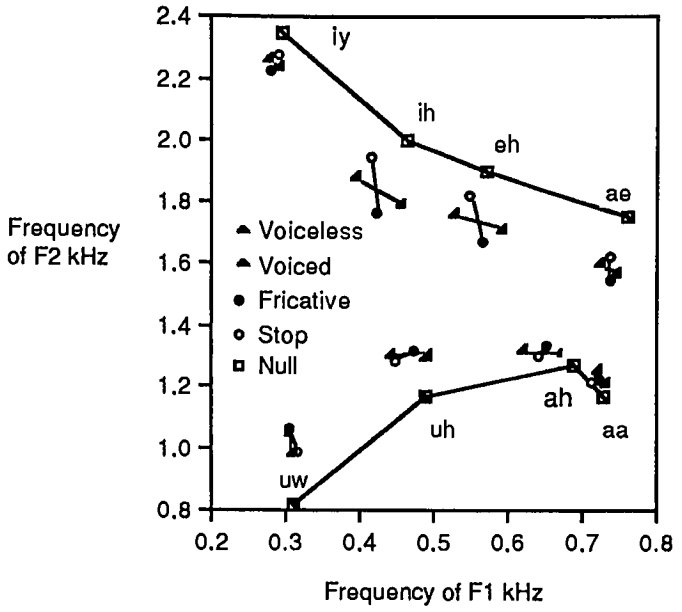


Figure 3-11 The effect of manner of production and voicing characteristic of the consonantal context *(Adapted from [STEV63]).*

Öhmann [ÖHMA66] continued this study of the effects of undershooting the target vowel in a vowel-consonant-vowel (VCV) environment, where the consonants in question were the stop consonants /b,d,g/. His results showed that a VCV utterance can not be regarded as a linear sequence of three successive gestures. Öhmann states that, "we have clear evidence that the stop-consonant gestures are actually superimposed on a consonant-dependent vowel substrate that is present during all of the consonantal gesture" [ÖHMA66 p165]. Given the variability of the formant transition in VCV sequences, his paper suggests that traces of the final vowel are observable already in the transition from the initial vowel to the consonant. By using x-rays of the mouth during speech production, Öhmann was able to illustrate that the tongue is able to make a distorted vowel gesture, while it is executing the stop consonant. This implies that our muscles are freely able to anticipate a following vowel, and moves toward such a position. So "as a result of coarticulation, vowels are encoded into the structure of a full syllable. The imprint of the vowel is not localized, but is smeared throughout the entire temporal course of the syllable" [SHAN75 p124]. For this reason, the study of vowels is complex since the vowel information is available not only in the steady-state portion of the vowel, but also in the transition from phoneme to phoneme.

Vowel Perception

By listening to isolated words the listeners in Peterson and Barney's study [PETE52] were able to correctly identify the vowels spoken approximately 94% of the time. Several studies have been performed both by using human listeners and with the aid of a computer in hopes of duplicating or even improving this performance.

In all of the studies mentioned above, vowel classification has been conducted in a static fashion by looking at a single spectral cross-section from the steady-state portion of the particular vowel. This assumption that the essential acoustic information lies in a specific location fails especially in the case of coarticulated vowels. Within coarticulated vowels "there is no steady-state portion that can be easily identified as being the most representative of the vowel target" [STRA87 p4]. Consequently, "no single spectral cross-section adequately captures the perceptually relevant information; rather, the acoustic information for vowel identity resides in the *changing* spectral structure" [STRA87 p9-10].

Strange [STRA87,89] suggests that important cues to vowel identity can be found in the formant trajectories into and out of the syllable nucleus. She examined the effects of these 'onglides and offglides' [LEHI61] with the use of four different vowel environments, labeled as: silent-center, variable-center, fixed-center, and the complement of the silent-center case. Silent-center syllables were created by attenuating the syllable nucleus to silence while leaving the initial and final transitional portions in their original temporal location. This way, the intrinsic duration and trajectory shape is saved while the target information is removed. Variable-centers were constructed by removing from the syllable the initial and final transitions while keeping intact the duration and target information. In the fixed-center syllables the "vocalic nuclei were all trimmed to the length of the shortest original nucleus" [STRA87 p6]. The last environment contained a vowel nucleus with the initial and final transitions removed. Ten vowels in the /b-V-b/ context were used along with a carrier sentence "I say the word ___ somemore."

Strange noticed that "when the duration information was removed, the error rates increased significantly for both the silent-center and fixed-center

stimuli" [STRA87 p7]. She also stated that vowel identity was not well maintained when the initial or final transitional components were presented alone, which suggests that vowels have an underlying intrinsic timing characteristic. The result of this study suggests that "normalizing for differences in vocal tract size and shape and perceptual compensation for "target undershoot" may not be necessary" [STRA87 p10] after all; rather, temporal information must be examined in order to accurately identify a vowel.

By using only the static spectral properties, Hillenbrand and Gayvert [HILL88] tried to duplicate Peterson and Barney's high rate of vowel identification. Based upon the pitch and three formant values in the Peterson and Barney database, they used a formant synthesizer to generate a 300 msec steady-state version of all 1,520 vowels in order to present these stimuli to 12 listeners. The listeners were capable of achieving an identification rate of approximately 73%. Hillenbrand and Gayvert [HILL88] then added to the static stimuli a dynamic pitch contour in hopes that a more natural sounding vowel would increase the identification rate. Unfortunately, the listeners also achieved an error rate of nearly 25% - more than four times greater than the error rate originally reported by Peterson and Barney. The result of this study suggests that "the dynamic information which was missing from the resynthesized steady-state signals plays a strong role in vowel identification" and that the fundamental frequency value may not be important at all [HILL89].

These results also support the findings of Rakerd et. al. [RAKE84]. Rakerd et. al. showed that vowels are better judged in CVC context than in isolation. They suggest the reason for this is the vowel-consonant combination tends to make up words already represented in a subject's lexicon, or at least portions of such words. This may be because in English, vowels in context tend to be less ambiguous then those of isolated words [RAKE84].

Automatic Vowel Recognition

Now that several studies have been conducted in which human listeners attempt to classify vowels based upon auditory cues, work has been shifted to the development of vowel recognition algorithms. Using the results of previously successful studies: Peterson and Barney [PETE52], Stevens and House

[STEV55,61,63], Öhmann [Öhma66] and Zue [COLE80] to name a few, speech researchers are trying to automate vowel recognition using features that were proven beneficial in accurately classifying vowels.

Miller [MILL87,89] believes that an auditory-perceptual interpretation of the vowel captures the important features required for vowel identification. "In the auditory-perceptual theory it is assumed that the listener's auditory system derives a sensory-spectral envelope" [MILL87 p6]. This envelope then acts as a pointer into a relevant Auditory-Perceptual Space (APS) where the vowel spectrum is located as a sensory point. The coordinates of his perceptual space are determined by the distance between the various sensory formants and a sensory reference (SR) "The sensory reference is believed to depend on the talker's average vocal characteristics and on appropriately filtered pitch modulations" [MILL87 p6]. The three prominent peaks in the spectrum correlate to the sensory formants SF1, SF2, and SF3. In the A-P space, the x axis corresponds to the logarithmetic distance from SF2 to SF3; log(SF3/SF2). The y axis is represented by the logarithmetic distance from SR to SF1; log(SF1/SR). The z axis is defined as the logarithmetic distance from SF1 to SF2; log(SF2/SF1). Miller suggests that by using either a ratio of the formant frequency values or mel ratios, talker differences can be eliminated. In this A-P space, it was noticed that vowels form a 'slab'. The presence of the slab is thought to be associated to the vocal tract size, pitch and third formant value. By simply rotating the slab among the various axes, the more familiar articulatory descriptions such as: the high-low, front-back dimension that characterize the tongue position; the opened-closed dimension that represents the jaw opening associated with the vowel; and the grave-acute, and compact-diffuse dimension of the first two formants can be found. In using the auditory-perceptual interpretation of vowel perception, Miller was able to demonstrate a classification of American English vowels with 93% accuracy.

In an earlier study, Syrdal and Gopal proposed "a quantitative perceptual model of vowel recognition based upon the spatial patterns of auditory excitation produced by American English vowels" [SYRD86 p1099]. They claimed that by transforming the physical formant frequency measures to a critical band scale (barks), their system is capable of classifying vowels into phonetic features better than a linear system that only used unnormalized frequency values. Their results

showed that the discriminant analysis of bark differences was more accurate than a linear system (86% vs 82%).

In order to compare a variety of different vowel classification algorithms proposed in the literature, a very thorough study was performed by Hillenbrand and Gayvert [HILL87]. In order to make a valid comparison, they performed each of their tests on the Peterson and Barney database. The features used in this process were the bark scale test found in [SYRD86] and a spectral distance test as in [MILL87], as well as a form of Gerstman's [GERS68] self-normalization algorithm. Their results, based upon the fundamental and formant frequencies, demonstrated that no one particular approach is significantly better than the other. As table 3-1 shows, each of the methods on average were able to achieve classification accuracy of approximately 87% [HILL87]. These findings show that the linear frequency values for the pitch and first three formants were comparable to the formant differences used in the bark or mel scales.

Parameter Set	Linear	Log	Bark	Mel
F_1, F_2	74.9	76.8	75.9	75.2
F_1, F_2, F_3	84.2	84.2	84.4	84.3
F_0, F_1, F_2, F_3	87.7	87.3	87.5	87.3
F_1-F_0,F_2-F_1,F_3-F_2	86.3	86.9	87.3	86.1

Table 3-1 Classification Accuracy for various formant parameters
(Adapted from [HILL87]).

By averaging each individuals' formant and fundamental frequency values, their classification accuracy improved significantly (Table 3-2). The symbols mf_0, mF_1, mF_2, and mF_3 represent the mean frequency values for the speaker; 'mF_1' is the mean first formant frequency, and so on. "The most attractive parameter set in this table is the first entry in the bottom set -- F_1, F_2, mF_1, mF_2. The performance is reasonably good, and you can do everything by measuring just two parameters -- F_1 and F_2" [HILL87 p3]. The modestly higher classification accuracy of other combinations are offset by potential formant tracking errors.

Parameter Set	Classification Accuracy
F_1, F_2, mf_0	85.8
F_1, F_2, mF_1	88.0
F_1, F_2, mF_2	88.3
F1, F2, mF3	85.0
F_1, F_2, F_3, mf_0	87.8
F_1, F_2, F_3, mF_1	89.5
F_1, F_2, F_3, mF_2	89.5
F_1, F_2, F_3, mF_3	88.0
F_1, F_2, mF_1, mF_2	90.8
$F_1, F_2, mf_0, mF_1, mF_2$	91.0
$F_1, F_2, mf_0, mF_1, mF_2, mF_3$	91.1
$F_1, F_2, F_3, mF_1, mF_2, mF_3$	91.5
$F_1, F_2, F_3, mf_0, mF_1, mF_2, mF_3$	91.8

Table 3-2 Internal Information combined with generic speaker information *(Adapted from [HILL87])*

More recently an automatic vowel recognition experiment compared classification based on overall spectral shape versus formant frequencies. Zahorian and Jagharghi [ZAHO87] examined 11 vowels spoken in 4 contexts by 17 female speakers and 12 male speakers. From the four contexts, /h-V-d/, /b-V-b/, isolated vowel and typical word, 140 msec's was extracted from the steady-state portion of each vowel. "Both the formants and spectral shape values were based on a single 25.6 msec frame located at the center of the vowel segment" [ZAHO87 p2]. They used a neural network, a maximum likelihood classifier, and linear discriminant analysis to categorize the signals. Input to these models consisted of the formant values measured on a linear frequency scale as well as on a mel frequency scale, along with using spectral shape values. "The spectral shape was encoded using three methods: as the outputs of a 16 channel one-third octave filter bank, as the outputs of a mel spaced filter bank, and as the discrete cosine transform of the FFT-computed magnitude spectrum (DCT)" [ZAHO87 p2]. For the 11 vowels used in their study, they were able to achieve about 69% accuracy in classification by using the formant values, versus approximately 85% correct based on spectral shape. As Figure 3-12 shows, there was very little difference between linear and mel formants, and only a small difference was noted for all three spectral shape methods.

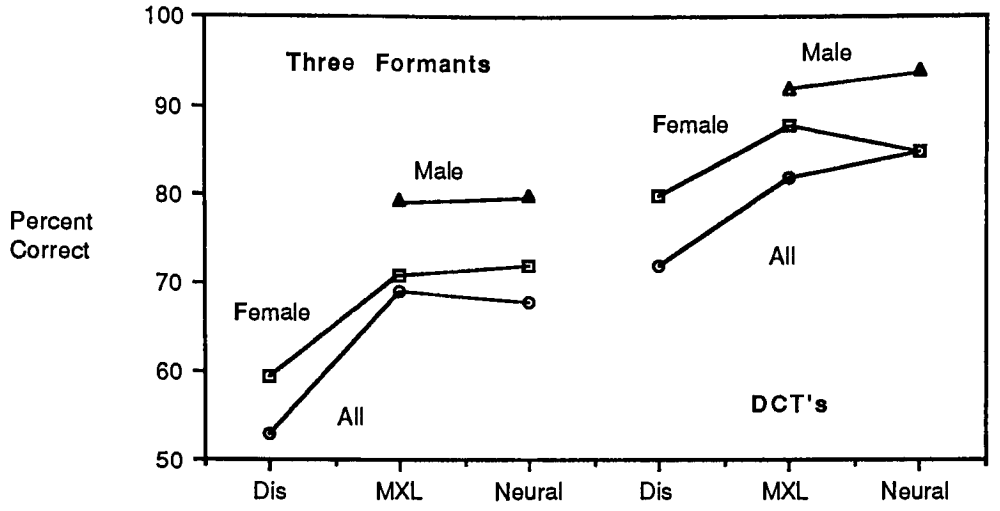


Figure 3-12 Automatic Recognition Accuracy for Eleven Vowels as a Function of Classification Algorithm *(Adapted from [ZAHO87]).*

The overall recognition rate for the data based upon the spectral shape is significantly higher than for the corresponding formant frequency values. Their results also show that the neural network yields a slightly higher accuracy rate than for the maximum likelihood measure as depicted in Figure 3-13.

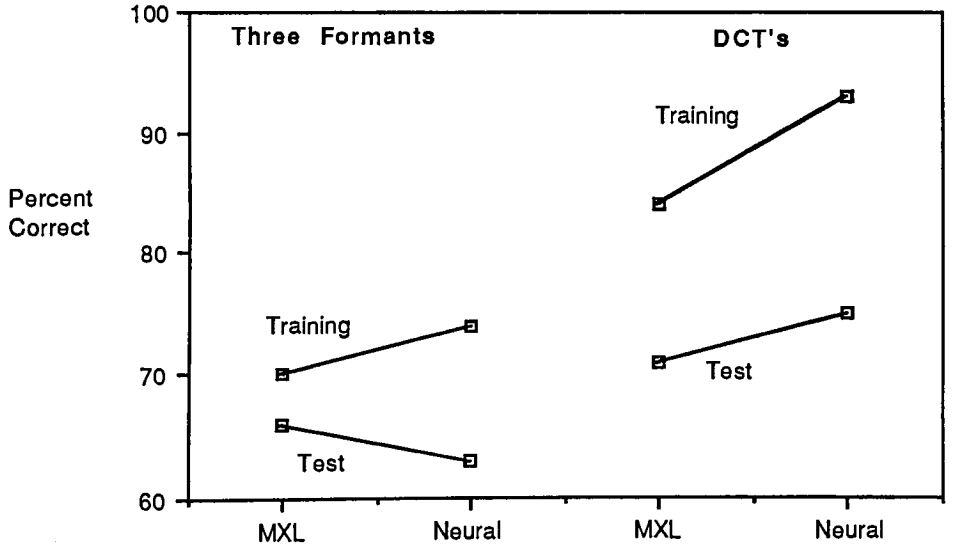


Figure 3-13 Automatic recognition accuracy for eleven vowels as a function of classification algorithm for test data *(Adapted from [ZAHO87]).*

Summary

The source-filter theory proclaims that a vowel sound is produced by both the amount of tension applied to the vocal folds and the shape of the vocal tract filter. Different sounds are then formed by the modification of either the source, vibrating body, or resonator, or any such combination. Classification of these diverse sounds requires the ability to find the appropriate set of features that adequately characterize the vowel region. On the basis of the previously cited studies it has been demonstrated that using only the steady-state regions within a vowel is not sufficient in classifying a particular vowel. Some of the important vowel characteristics to study are the formant transitions from phoneme to phoneme, the location of the first three formant and fundamental frequency values as well as the overall spectral shape.

Chapter 4

System Implementation

General Description

The present system represents an attempt to classify vowels based on the dynamic pattern of acoustic information throughout the course of the vowel. Training of the vowel recognizer involves examining many samples of speech data in order to allow the system to learn the salient characteristics of each vowel category. This vowel recognition system is separated into four different stages (Figure 4-1). The first stage locates the speech signals and performs all the signal processing routines that best characterize the vowel regions. The second stage extracts all the vowel regions from the various derived signals in order to form a database of vowels along with their acoustic characteristics. These regions are used to train either a maximum likelihood or neural network preclassifier. The third stage involves a preclassification of the vowels based upon the static spectrum. This stage produces a sequence of decisions that can be thought of as a pathway through phonetic space. The final stage produces a mapping from the sequence of discrete decisions to a single vowel state with the aid of a hidden Markov model.

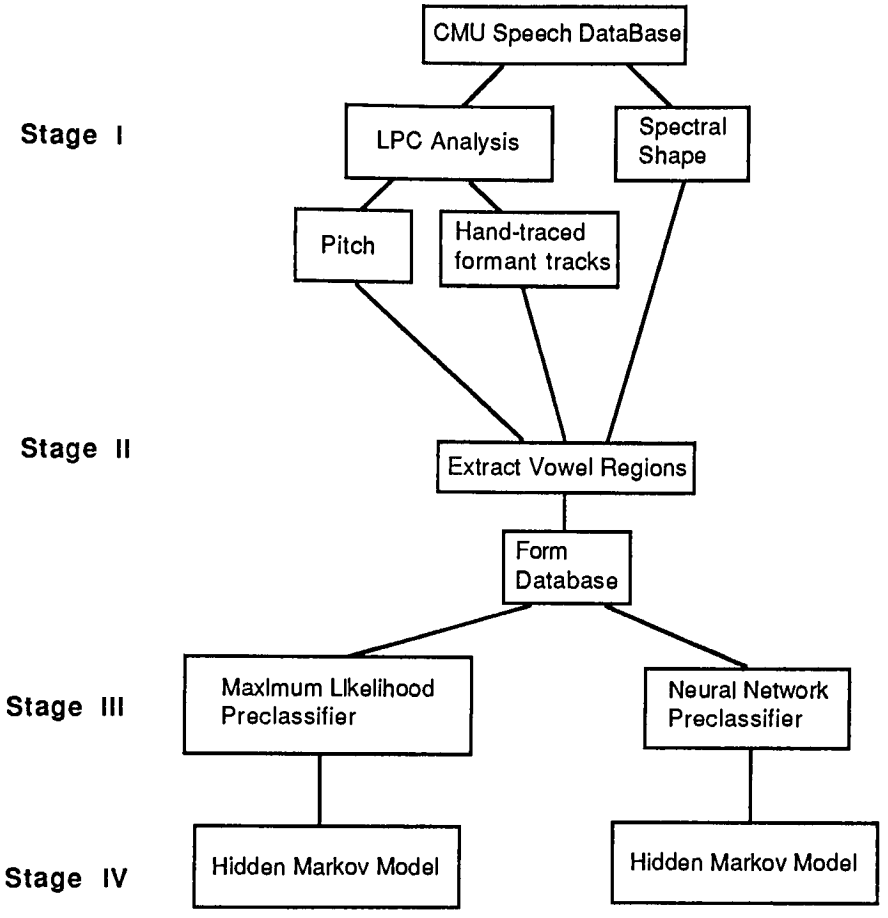


Figure 4-1 Vowel Recognition Architecture

Figure 4-1 Vowel Recognition Architecture

Database

The speech database to be used in this thesis is a subset of a database supplied by Carnegie-Mellon University. It includes 79 utterances spoken by 39 men and 40 women. The speech files consist of the phonetic transcription information and a binary data file of 12 bit PCM sampled at 16 kHz. The signals were subsequently low-pass filtered at 6 kHz and down sampled to 12.8 kHz. The feature set produced by this database was used for both training and testing phases of this work. Since several tests were performed, sometimes all of these data were used at one time for training and testing. Finally the data was split in half with an equal distribution of male and female speakers; half was used for training and the other half was used for testing.

Feature Sets

Each of the speech utterances in the database was used as input to the various feature-extraction algorithms to produce a sequence of feature vectors. These algorithms are described in detail below.

Linear Predictive Coding

One of the characteristics of an acoustic signal is that over time the frequency spectrum does not remain constant, even though nearby sample values may be correlated. By subdividing the signal into a series of smaller time segments, the redundancy implied by the correlation can be exploited. This in turn reduces the amount of information needed to represent the signal. That is, a speech sample x(n) can be predicted with a considerable amount of accuracy by the previous sample x(n-1). Linear prediction is considered to be one of the better coding techniques since it is able to derive a model of the vocal tract by separating much of the prosodic from the segmental information. The signal then can be represented by a set of LPC coefficients of order N. The poles of the model correspond to the resonances or formant frequencies within the signal. A fourteenth order model was used in this study.

Spectral Moments

When a set of values have a tendency to cluster around some particular value, it then may be useful to characterize this collection by a set of a few numbers. These numbers, the moments of the spectrum, are related to the sums of integer powers of the values. The four moments, mean, variance, skewness, and kurtosis, were calculated in hopes of capturing the shape of the vowel spectrum.

The mean value of a spectrum quantifies the values around which central clustering occurs. Mean is calculated by the following equation, given the values $x_1,\ldots,x_n$,

$$\bar{x} = \frac{1}{N}\sum_{j=1}^{N} x_j$$

The variability around the mean value is the next moment that characterizes the distribution. This variance is calculated by:

$$\mathrm{Var}(x_1,\ldots,x_n) = \frac{1}{N-1}\sum_{j=1}^{N}\left[x_j - \bar{x}\right]^2$$

Skewness characterizes the shape of a distribution around its mean. The definition is:

$$\mathrm{Skew}(x_1,\ldots,x_n) = \frac{1}{N}\sum_{j=1}^{N}\left[\frac{x_j - \bar{x}}{\sigma}\right]^3$$

where $\sigma = \sigma(x_1,\ldots,x_n) = \sqrt{\mathrm{Var}(x_1,\ldots,x_n)}$ = standard deviation

Kurtosis measures the relative peakedness or flatness of a distribution to that of a normal distribution [PRES88].

$$\mathrm{Kurt}(x_1,\ldots,x_n) = \left(\frac{1}{N}\sum_{j=1}^{N}\left[\frac{x_j - \bar{x}}{\sigma}\right]^4\right)$$

Median Value

A more robust estimator than the mean is called the median. If a distribution of data has a tendency to concentrate in a focal area or under a single peak, then the median can estimate this central value. This value is such that smaller and larger values occur with equal probability [PRES88]. The median is calculated by taking a list of numbers of length N and sorting them in either ascending or descending order. The formula for the median is:

$$\text{med} = X_{(N+1)/2} \qquad \text{for N odd}$$

$$= 1/2(X_{N/2} + X_{(N/2)+1}) \quad \text{for N even.}$$

Formants and Fundamental Frequency

Fundamental frequency values were calculated using the Simple Inverse Filter Tracking (SIFT) algorithm [MARK72] with the values smoothed by a median smoother. The formant frequency values* were extracted from hand traced signals. The hand-edited formant trajectories were produced from a fourteenth order LPC spectrogram with the phonetic transcription and waterfall display aligned for a reference. A mouse-oriented tool was used to trace the plot peaks and to fill in missing values. A detailed description of the hand-tracing technique can be found in Gayvert [GAYV89 p8]. Occasionally there were missing formant frequency values in the data, which are represented by a zero value (Figure 4-2). When this occurred, the entire frame was extracted from the vowel token so as to minimize the chances of having the zero value being interpreted as a characteristic feature. Thus, a frame was only used when all three formant frequency values were present.

* A Markel formant tracker was investigated but it was later dropped from this study. It was decided that this thesis should deal with vowel classification rather than signal processing. This thesis assumes that the formant frequencies are accurate.

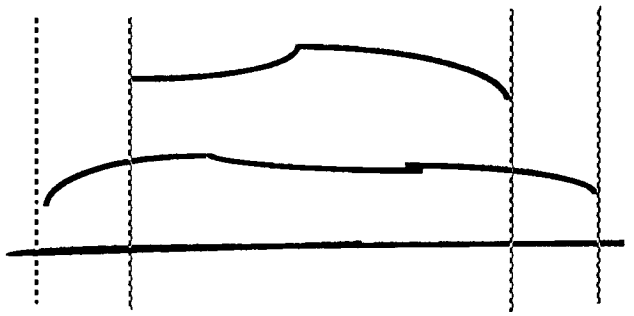


Figure 4-2 Example of missing formant frequency values.

Vowel Extraction

In stage two, vowel regions were extracted from the derived signals based upon the hand labeled segments. This produced slightly less than a minute of vowel-rich speech. The twelve vowels /iy, ih, eh, ae, ix, ao, ah, uw, uh, aa, retroflex/ were studied with the retroflex vowel class representing the two similar vowels /axr/ and /er/. These two vowels were clustered together since they are similar in nature. These extracted segments are used to create the feature vectors for the database in this thesis.

Preclassification

At the third stage, the vowel phonemes were classified based upon the characteristics found within single frames taken at 5 msec intervals. Note that by examining each vowel region every 5 msec, information such as duration and coarticulation effects are temporarily lost. The goal here was to produce a sequence of discrete points through a normalized phonetic space that allows for speaker variations. The hidden Markov-based dynamic classifier is capable of using this pathway along with duration information in mapping the region onto a particular vowel phoneme. Classification based on two different methods were examined at this stage. A comparison was made between a maximum likelihood classifier and a back-propagation neural network.

Maximum Likelihood

Given that speech features often result in overlapping clusters, an elaborate method is engaged to measure the distance between a specific value and a vowel cluster. Provided with a set of vowel phoneme observations containing several features, the maximum likelihood distance measure develops a discriminant model to classify each observation into one of the eleven possible groups. A model is determined by finding the values that maximize the probability of the feature vector that best fits the normal (Gaussian) distribution of the "true" class. "Each observation is placed in the class from which it has the smallest generalized squared distance" [SAS82 p381] based upon the individual within group covariance matrices. The following describes the generalized squared distance formula:

t	=	a subscript to distinguish the groups
C_t	=	the covariance matrix within group t
$\|C_t\|$	=	the determinant of C_t
x	=	a vector containing the features of an observation
$\bar{x}_t$	=	the vector containing the means of the variables in the group t.

The generalized squared distance of x to group t is then:

$$D^2_t(x) = (x - \bar{x}_t)^T C^{-1}_t (x - \bar{x}_t) + \log |C_t|$$

Neural Network

"The back-propagation training algorithm is an iterative gradient [descent] algorithm designed to minimize the mean square error between the actual output of a multi-layer feed-forward perceptron and the desired output" [LIPP87 p17]. In this study the back-propagation model used a single hidden layer with the hyperbolic tangent as its sigmoid nonlinearity function. Since several combinations of the input feature set were used to determine the optimum result, the number of nodes used in the hidden layer is dependent upon the number of input values.

Input is presented continuously in a vector format (x_0, ..., x_{n-1}) with the desired outputs (d_0, ...,d_{m-1}) being a vector of all zeros except for a positive value at the location of the particular class to be trained upon (Figure 4-3). Output values

$(y_0, \ldots, y_{m-1})$ are calculated based upon the sigmoid nonlinear function mentioned above. By working backward from the output layer to the first hidden layer, the weights are adjusted by the convergence algorithm with a momentum term added:

$$W_{ij}(t+1) = W_{ij}(t) + \varepsilon \delta_j x'_i + \alpha \left(W_{ij}(t) - W_{ij}(t-1) \right)$$

"In this equation $W_{ij}(t)$ is the weight from the hidden node i or from input node j at time t; x'_i is either an input value or the output from node i; ε is a gain term and δ_j is an error term for node j. If node j is an output node, then :

$$\delta_j = (1 - y^2_j)(d_j - y_j)$$

where δ_j is the desired output of node j and y_j is the actual output. If node j is an internal hidden node, then:

$$\delta_j = (1 - x_j^2) \sum_k \delta_k W_{jk}$$

where k is over all nodes in the layers above node j" [LIPP87 p17]. Note that if a different nonlinearity function is used, the δ function will also have to be modified since the first term in the δ equation is based upon the derivative of the specific nonlinearity function being used.

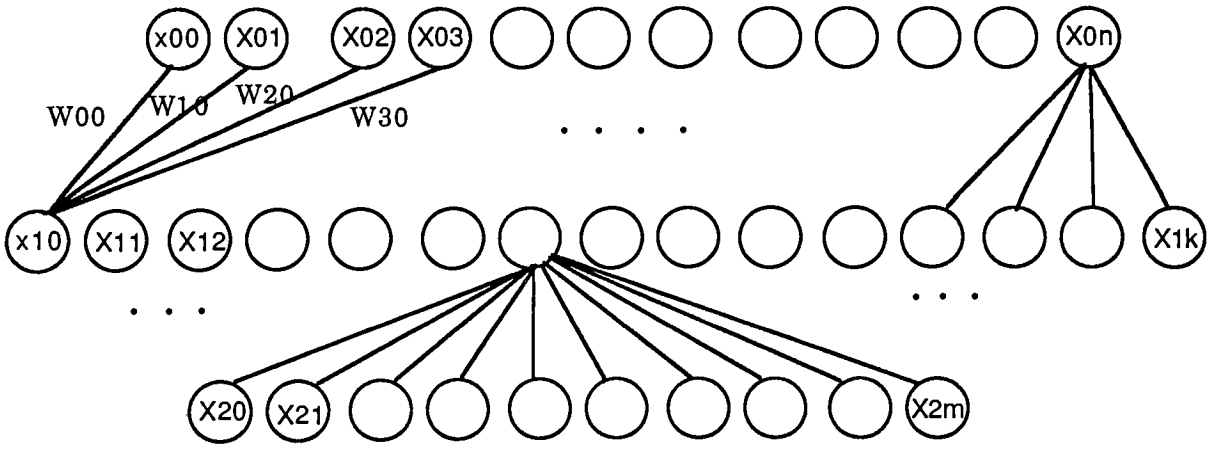


Figure 4-3 A sample Neural network with N input nodes, K nodes in one hidden layer and M nodes in the output layer.

Dynamic Classification Using Hidden Markov Models

Modeling a time-varying process by a direct concatenation of short time segments assumes that every such segment is a unit with a prechosen duration. In this study, each time segment represents 5 msec of time. This assumption rarely holds true for acoustic signals since the variations in the signal do not occur in a predetermined time frame. A signal can be thought of randomly varying time frames that transpire in a non predictable manner. Thus if the assumption is made that the signal remains relatively steady for a period of time with minor fluctuations, then a model can be built to capture this behavior. An efficient model can be designed if these steady-state periods are first identified. The temporal variations within these steady periods can then be assumed to be statistical in nature. "A more efficient representation may then be obtained by using a common short time model for each of the steady, or well-behaved parts of the signal, along with some characterization of how one such period evolves to the next. This is how hidden Markov models (HMM) came about" [RABI86 p5]. Since these transition are based upon probabilities and are not readily observable except by "another set of stochastic process that produce the sequence of observations" [RABI89 p259] the Markov model obtains the "hidden" name. In order to be useful in real world applications there are three key problems for a HMM to solve given the following elements:

T: length of an observation sequence
N: number of states in the model denoted by $S = \{s_1, ..., s_N\}$,
and the state at time t as q_t
M: number of discrete observation symbols per-state $V = \{v_1, ..., v_M\}$

A: $\{a_{ij}\}$ a state transition probability distribution where

$$a_{ij} = \Pr(q_j \text{ at time } t+1 \mid q_i \text{ at time } t),\ 1 \le i, j \le N$$

B: $\{b_j(k)\}$, observation symbol probability distribution in state j, where

$$b_j(k) = \Pr(v_k \text{ at } t \mid q_t = s_j),\ 1 \le j \le N,\ 1 \le k \le M$$

π: $\{\pi_i\}$ initial state distribution, where

$$\pi_i = \Pr(q_1 = s_i),\ 1 \le i \le N$$

The first problem is an evaluation problem. Given a sequence of observations, $O = O_1, ...,O_T$ and a model $\lambda = (A,B,\pi)$, (Figure 4-4), this problem solves the question: 'how can one compute the probability of the observation actually occurring?' The second problem tries to determine a state sequence $I = i_1 ,...,i_t$ which is optimal. The third problem attempts to optimize (adjust) the model parameters so as to maximize the output probabilities of the observation occurring.

The idea of using a Markov model in this study is to model the vowel over time. Given a vowel token that is classified by the preclassifiers as a sequence of steady states such as /ih ih ih iy iy iy eh eh/, the model should be able to distinguish between the initial onglide transition of the formant frequency, the central region and then the offglide. If a model is designed and trained well enough, it should be able to figure out the transitions from state to state to capture the vowel transitions. Figure 4-4 shows an example of a three state left to right model. One would hope that each of the states would model the initial, central, and final formant frequency transitions that occur naturally within a vowel.

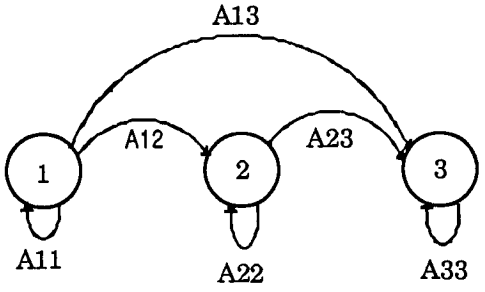


Figure 4-4 A simple three state left to right hidden Markov model

Chapter 5

Results and Conclusions

Database Size

At the onset of this project it was intended that the classifiers would be able to distinguish among the twelve vowel phonemes: /ah, aa, ao, ae, ih, iy, ix, eh, er, axr, uw, uh/. Preliminary tests indicated that identification of these classes could not be determined with a strong amount of confidence. The difficulty does not come from any assumption that the vowel phonemes can not be separated, but rather from the amount of data available to train and then test the classifiers. From the entire CMU speech database, only 472 vowel tokens had the hand-traced formant tracks that were used to classify the vowels. Given the amount of useful vowels that could be extracted, it was thus decided to reduce the scope of the classification to the ten original vowels that were used in the Peterson and Barney study [PETE52] (hereafter P&B). While this decision lessened the total number of tokens to 313, it allowed comparisons to previously cited studies.

Significant differences between the current study and aforementioned ones exist in the manner in which the vowel tokens were acquired and classified. In this study the entire vowel region was extracted from continuous speech based upon hand labeled phonetic transcriptions as found in the sentences in Appendix B. Other studies, such as the one performed by P&B, selected vowel tokens from a controled consonant-vowel-consonant (CVC) context (/h/-v-/d/) without the influence of any precursor words. The vowel tokens for many previous studies were typically extracted from the central part of the vowel, which is not influenced by the surrounding consonants. This would give the researcher a single frame or a small number of frames to represent the vowel, whereas the present study used all the frames within the vowel token.

Vowel Separability

Peterson and Barney showed that vowels can be separated with some degree of accuracy based solely upon the first two formant frequency values (Figure 5-1). It is quite obvious from visual inspection that the vowel phonemes used in this study are not as easily separable (Figure 5-2).

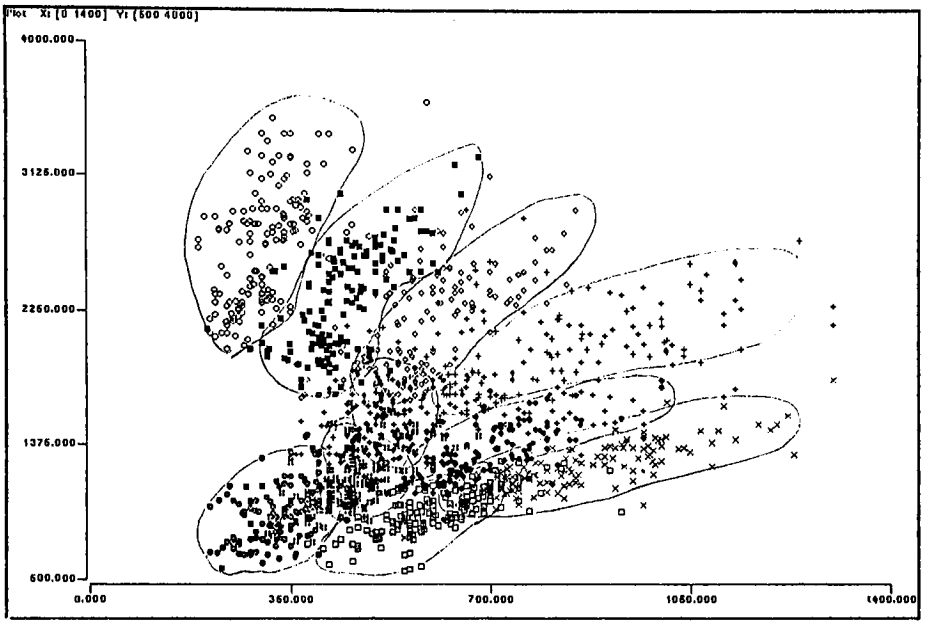


Figure 5-1 Classifying vowel sounds in terms of their first two formant frequency values (Peterson and Barney database)

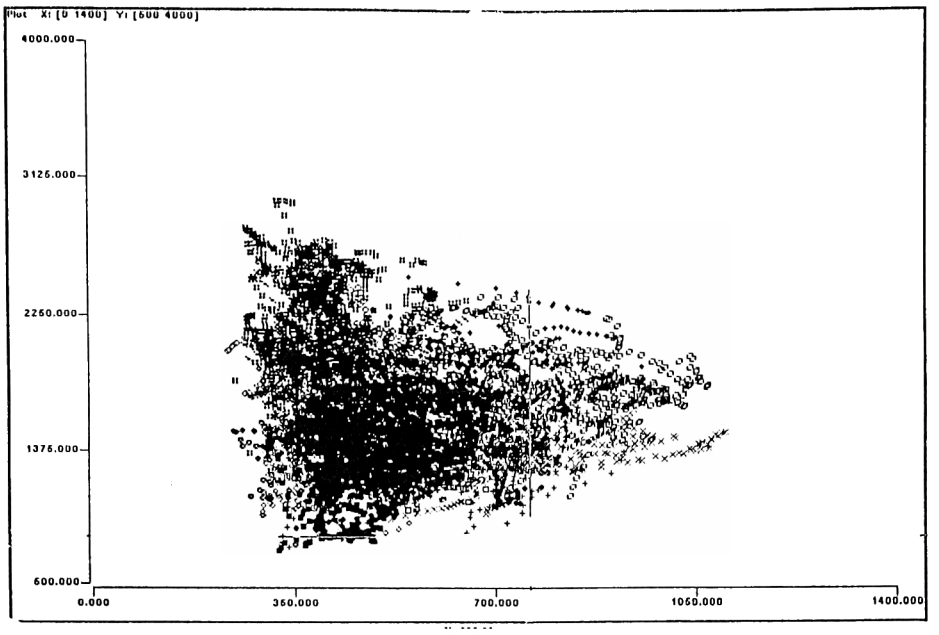


Figure 5-2 Classifying vowel sounds in terms of their first two formant frequency values. The graph displays the database used in this study.

Two reasons why it is difficult to visually distinguish the vowels in this study stem from the amount of data used to represent a single vowel token, and the manner in which the vowels were articulated. As previously mentioned, the entire vowel token was used in this study versus the single frames as found in P&B's study. The manner in which vowels are spoken determines how successfully they can be identified from their formant frequency values. In other words, it is more likely that a vowel in a controled CVC environment will reach a reasonably consistent target configuration. Primarily this is a coarticulation problem, where similar vowels look different due to adjacent phoneme sounds. Figures 5-3 through 5-12 show the effects of coarticulation and context upon various vowels by displaying lots of F_1-F_2 trajectories for vowels in different contexts. Notice that the vowel /iy/ looks markedly different in the different examples. These figures show some of the possible transitions that the vowels go through while being articulated. As can be seen, it is understandable why some of the vowels were not successfully identified.

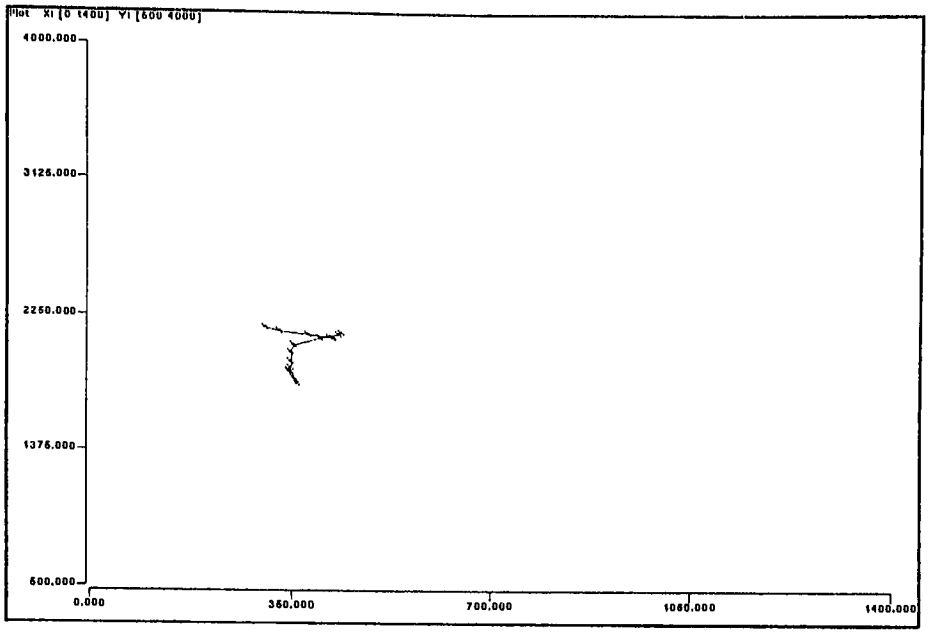


Figure 5-3 /iy/ in /p-h/ /n/context, classified as /iy/

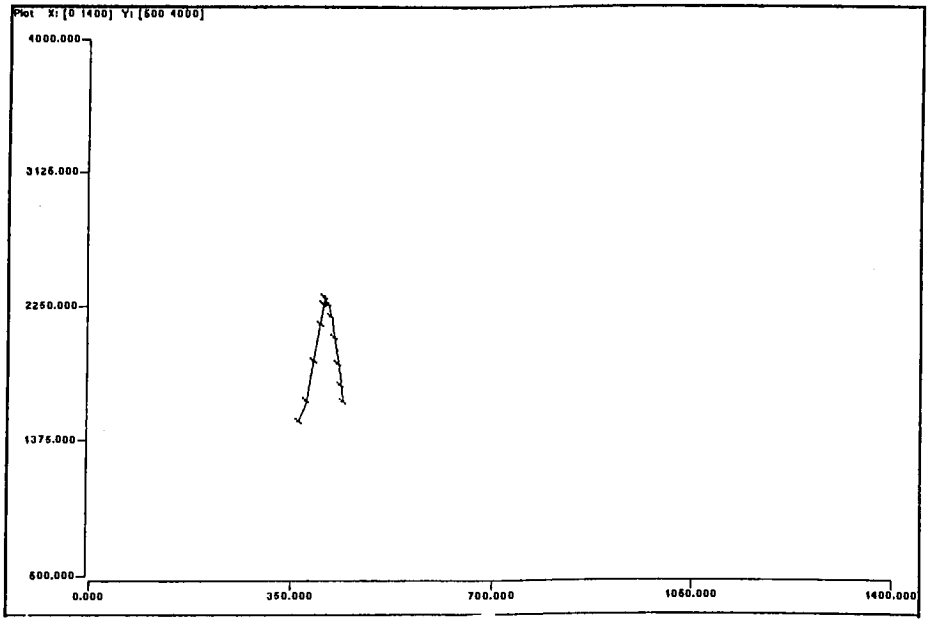


Figure 5-4 /iy/ in /r/ /l/ context, classified as /iy/

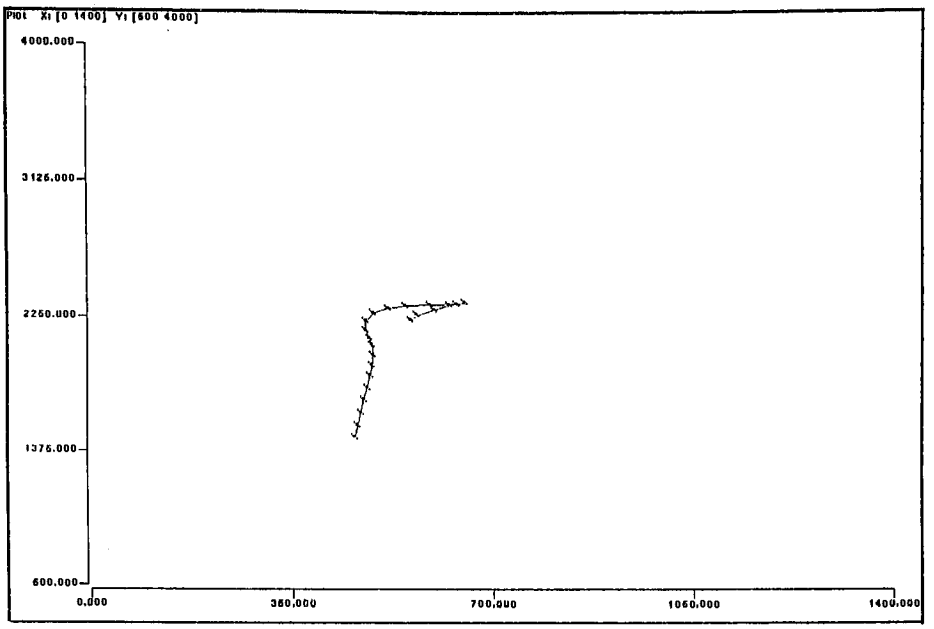


Figure 5-5 /iy/ in /dh/ /axr/ context, classified as /iy/

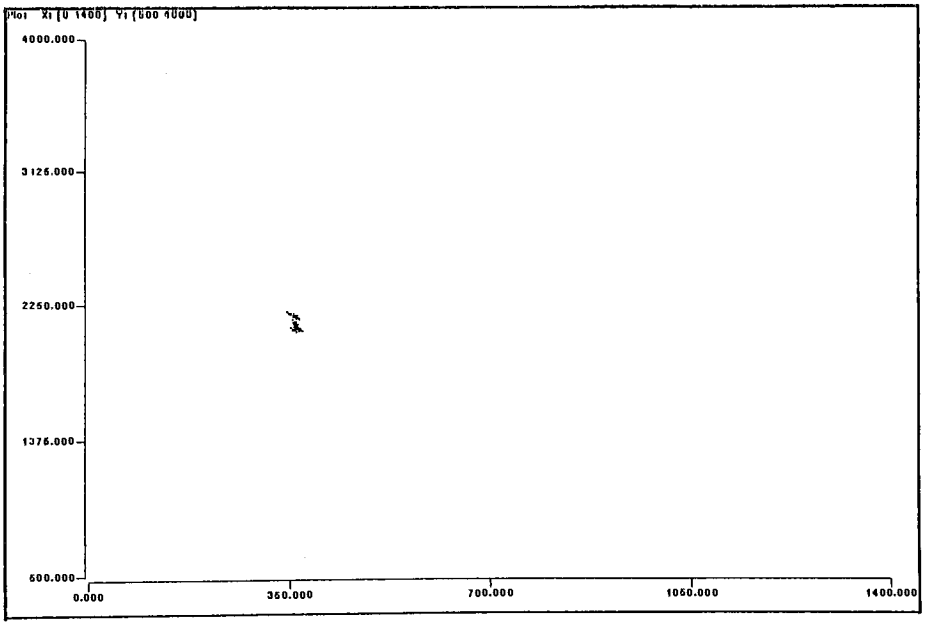


Figure 5-6 /iy/ in /th/ /s/ context, classified as /iy/

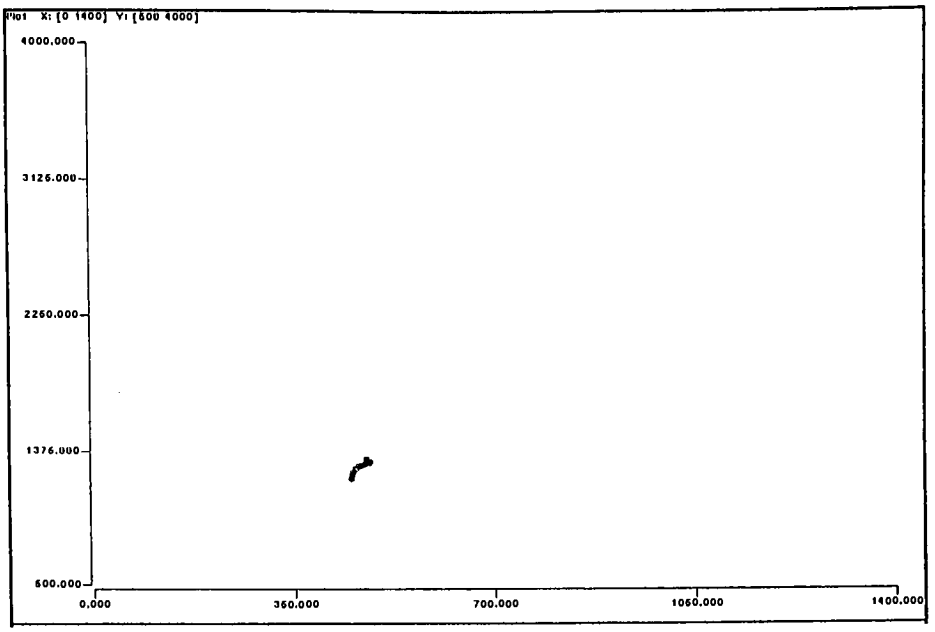


Figure 5-7 /ah/ in /l/ /dx/ context, classified as /uh/

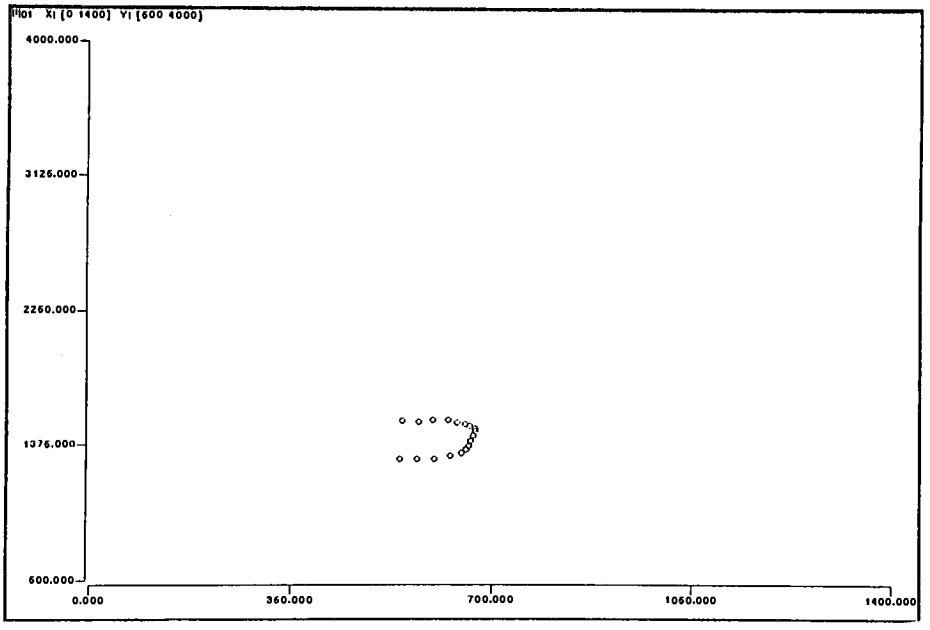


Figure 5-8 /ae/ in /r/ /m/ context, classified as /ah/

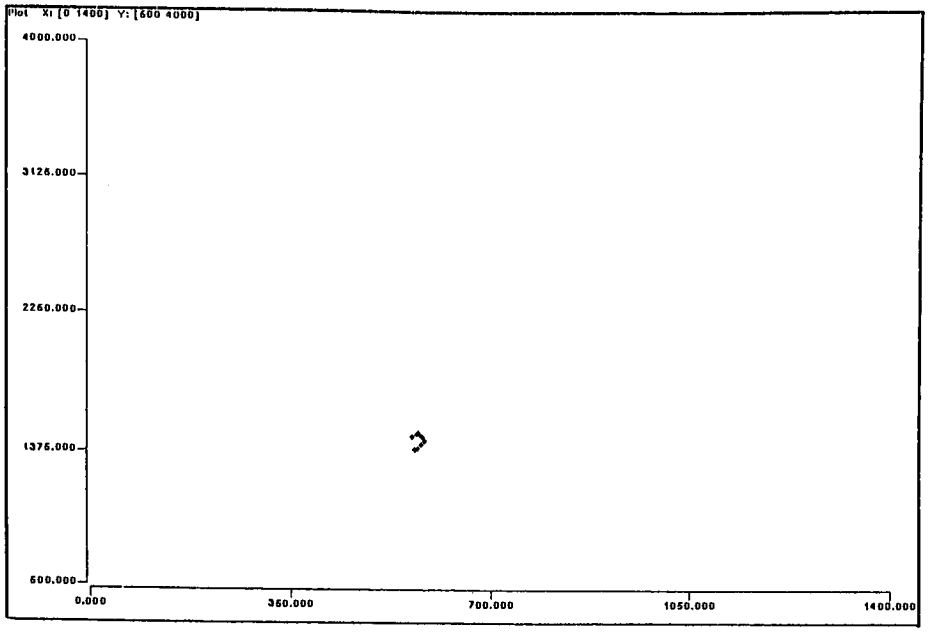


Figure 5-9 /eh/ in /r/ /kcl/ context, classified as /ae/

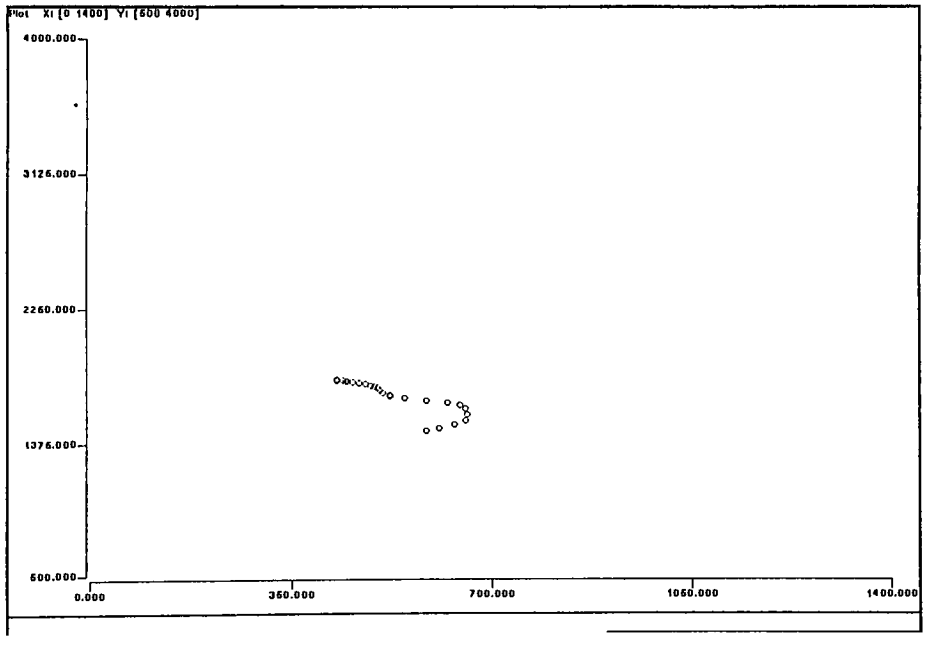


Figure 5-10 /ae/ in /jh/ /m/ context, classified as /ae/

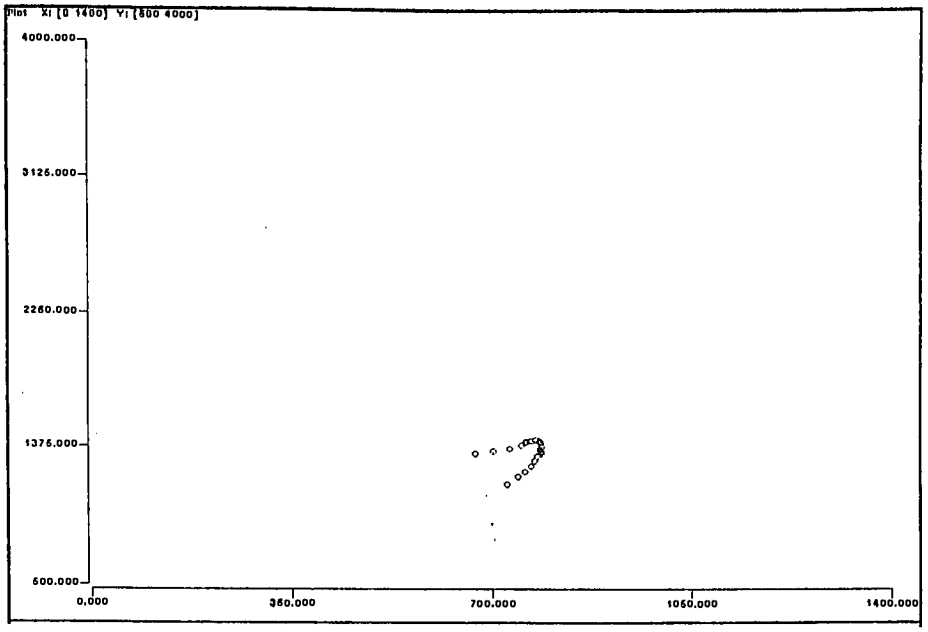


Figure 5-11 /ae/ in /l/ /v/ context, classified as /ae/

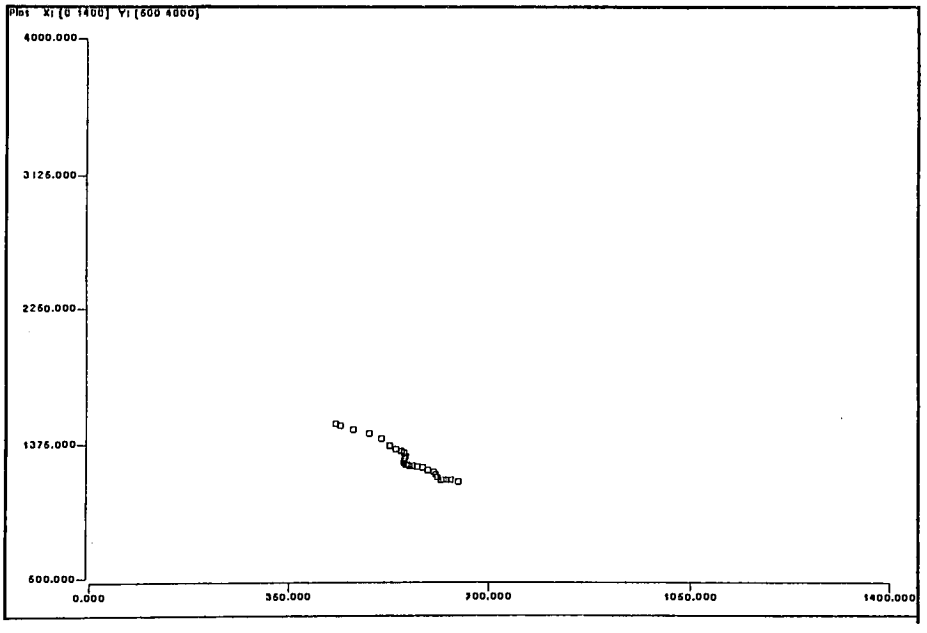


Figure 5-12 /er/ in /b/ /axr-q/ context, classified as /er/

Feature Set

As previously mentioned, the features that were used to classify the vowels consisted of the first three formant frequency values, (F_1,F_2,F_3), pitch (F0), mean (M1), variance (M2), skewness (M3), kurtosis (M4), median (M5), and the mean formant frequency values for each individual speaker (mF_1, mF_2, mF_3). Median was not derived from the spectral moments as the numbering scheme would imply, but it is easier to follow this progression when labeling, since these features deal with the distribution of the vowel spectrum. It is not surprising that the more successful classification involved feature sets that included F_1 and F_2. These formant frequencies best characterize tongue height and advancement. This is important since it enables a distinction to be made between front, central and back vowels.

Preclassification Results

Classification by the neural network and Gaussian preclassifiers was done on a frame-by-frame basis at 5 millisecond intervals. Training a back-propagation neural network involved finding an optimal configuration of hidden layer nodes that corresponded to the number of input features presented to the system. Once a favorable model was found, the focus shifted to finding ways to further improve the classification accuracy by adjusting both the momentum and gain terms used in adjusting the weights in the model. In order to test the model, the database was split in half with each half containing an equal distribution of vowel tokens. Each separate half was used for training and testing. The results achieved for a model given all twelve features with 18 nodes in one hidden layer and momentum and gain terms of 0.5 and 0.3 produced training accuracy at 64.46 % and testing accuracy at 47.38% (Tables 5-1a, 5-1b, Figure 4-3).

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy (509)	**85.66**	7.47	0.98	1.77	0.20	1.96	0.79	0.39	–	0.79
ih (363)	9.37	**66.94**	5.51	4.41	3.03	1.65	6.34	0.55	1.10	1.10
eh (328)	3.35	6.71	**47.87**	23.48	5.49	3.05	4.88	3.35	0.91	0.91
ae (498)	2.21	5.22	13.25	**67.47**	2.41	0.80	1.41	5.62	1.00	0.60
ah (239)	2.93	4.60	7.53	5.86	**61.09**	2.09	6.28	5.44	2.51	1.67
uw (207)	8.21	8.70	1.45	2.90	1.45	**66.18**	8.70	0.48	0.48	0.97
uh (196)	5.10	13.27	14.29	6.12	9.69	12.76	**32.14**	2.55	3.57	0.51
aa (184)	2.17	3.26	5.98	21.74	8.15	0.54	4.89	**47.83**	3.26	2.17
ao (111)	0.90	2.70	13.51	14.41	6.31	–	9.91	9.91	**40.54**	1.80
er (215)	2.33	2.33	2.33	0.93	1.86	0.47	1.86	0.47	0.93	**86.51**
Total	19.0	14.0	12.0	19.0	8.0	7.0	6.0	6.0	3.0	7.0
Average rate of correct decisions						64.46				

Table 5-1a Training Results. The tables/confusion matrices have the tokens presented to the model running across each row, and the recognized vowel phoneme runs down each column. A '-' value represents a zero value. All values reported here are in percentages. The total value represents the percent of time that is spent in each class.

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy (519)	**75.92**	13.29	0.77	–	0.39	6.94	1.93	0.77	–	–
ih (262)	14.50	**62.21**	3.44	1.53	2.29	12.60	3.44	–	–	–
eh (323)	2.17	11.15	**26.32**	26.01	12.38	1.86	17.96	–	–	2.17
ae (477)	–	8.60	13.63	**63.73**	6.50	–	2.94	3.98	–	0.63
ah (213)	–	14.80	5.16	15.49	**14.08**	–	46.48	3.76	–	0.94
uw(249)	7.63	3.21	–	–	0.40	**48.59**	38.96	–	–	1.20
uh (208)	–	5.77	–	–	12.98	29.33	**47.60**	3.37	–	0.96
aa (173)	–	4.62	4.62	21.97	42.77	0.58	18.50	**4.62**	–	2.31
ao (119)	–	–	–	21.85	31.93	–	32.77	–	–	13.45
er (144)	–	9.72	–	11.11	12.50	1.39	14.58	2.78	–	**47.92**
Total	15.0	14.0	7.0	19.0	10.0	10.0	18.0	2.0	0.0	4.0
Average rate of correct decisions						47.38				

Table 5-1b Results after (a) training and (b) testing a back-propagation neural network with 18 nodes in one hidden layer and $\varepsilon = 0.3$ and $\alpha = 0.5$

Training the Gaussian preclassifier was managed in a slightly different fashion. An exhaustive search of which combination of N features best characterized the vowel tokens was performed, where N ranged between 3 and 7. It was initially determined that the more features that were used to train the Gaussian model, the better the model would perform. Preclassification results of the optimal feature set given N features are shown in Table 5-2. Notice that the percent of correct preclassifications progressed logarithmically as the number of features was increased. Once testing was performed it was obvious by the drop of approximately twenty percent in classification how feature sets greater than five tended to model noise or the specific training vowel. Note that the low performance values are not reported here in this document, but somewhat can be seen in the differences between resubsition and jack-knifing in Figure 5-4.

Features Used	% Correct
F_1,F_2,F_3	55.68
F_1,F_2,F_3,F0	61.82
F_1,F_2,F_3,F0,mF_2	64.89
F_1,F_2,F_3,F0,mF_2,M5	66.99
F_1,F_2,F_3,F0,mF_2,M5,mF_3	68.54

Table 5-2 Gaussian preclassification results based upon feature set

Upon reflection, this modeling of noise is probably what also happened with the neural net. There are two ways to correct this problem; either the size of the training data base should be increased, or a different feature set to train the model should be selected. As already stated, the data set is fixed, so the only chance of improving the results is to select a different feature set. This is unfortunate for the neural net model since it already uses all of the features in the data set. Every feature was used since a neural net should be capable of determining which features are less important. Theoretically this should be true for the Gaussian model also, but due to this observation it was decided to re-evaluate the features that are presented the Gaussian preclassifier. The aim was to reduce the disparity between its training and testing results.

Several feature sets for the Gaussian preclassifier that were successful in nearly matching the accuracy of the optimal feature set of size five (F_1, F_2, F_3, F0,

mF_2) are in Table 5-3. The success of these new feature sets is not unexpected, since Hillenbrand and Gayvert [HILL87] showed classification accuracy in the low 90% range using similar features.

Features Used	% Correct
F_1,F_2,F_3,mF_1,mF_2	64.57
F_1,F_2,F_3,mF_1,mF_3	64.39
F_1,F_2,F_3,mF_2,mF_3	63.37
F_1,F_2,F_3,F0,M5	64.28

Table 5-3 Near optimal feature sets of size 5 for training a Gaussian preclassifier.

The feature set finally used for the Gaussian preclassifier was F_1, F_2, F_3, mF_1, mF_3 due to the minimal difference between the training and preliminary testing results (Table 5-4). mF_3 is used rather than mF_2 due to its slightly better testing results. The Gaussian preclassifier has two possible testing options resubsitution and jack-knife. The former involves simply training and testing the model against the same database. The later option is more accurate but computationally more intensive. The reason is that for every token presented to the system, its influence in the design of the distribution is extracted from the system. In other words, for the specific token used to train a Gaussian-distribution, its influence is removed from the distribution entirely before any testing is performed. After the token is tested against all possible distributions it is then reentered into its original distribution and another token is then extracted and the whole process is performed all over again. We can see from Table 5-4 that the resubsitution method produces a more favorable outcome over the jack-knife method by a factor of approximately 10%. Unfortunately, the resubsitution testing method allows the Gaussian model to "recognize" a former training token, and thus favorably influence the testing results. In either case, the preclassification results are still better than that of the neural net as compared to the results in Table 5-1b.

The difference in classification accuracy between this study and Hillenbrand and Gayvert's arises from the vowel tokens used to train and test the model. Hillenbrand and Gayvert utilized the static formant frequency measurements from P&B's study. Hillenbrand and Gayvert's results are more

desirable, based upon their high rate of success, but they used only a single frame for each vowel in a similar fashion as P&B. Given vowel dynamics, the results achieved in the current study are reasonable since this is how a true speech recognition system would most likely be presented a vowel token.

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy (1028)	**82.88**	8.37	0.88	0.29	0.78	4.77	0.19	–	–	1.85
ih (625)	11.52	**60.80**	7.04	–	4.96	10.24	4.0	0.16	–	1.28
eh (651)	1.38	9.22	**51.0**	17.36	8.29	0.92	2.76	3.53	2.92	2.61
ae (975)	1.33	3.69	16.82	**60.21**	3.59	0.10	0.41	12.21	0.41	1.23
ah (452)	0.44	4.87	5.31	3.10	**29.65**	3.32	16.81	23.45	11.95	1.11
uw (456)	7.02	1.54	–	0.22	1.32	**71.93**	15.13	–	–	2.85
uh (404)	–	4.21	5.45	–	10.15	16.58	**55.45**	1.73	4.70	1.73
aa (357)	0.84	2.52	2.52	9.52	15.13	–	1.12	**56.02**	10.64	1.68
ao(230)	–	–	0.87	2.17	0.43	2.61	5.65	–	**88.26**	–
er(359)	0.56	0.84	1.39	–	–	3.06	1.11	0.84	1.67	**90.53**
Total	.18	.11	.11	.14	.07	.10	.08	.08	.06	.07
Resubsitution						64.39				
Jack-knife[†]						54.92				

Table 5-4 Testing results for a Gaussian preclassifier with a feature set built from F_1, F_2, F_3, mF_1, and mF_3

Understanding Classification Errors

In analyzing the classification errors of both the neural net and Gaussian models, it is encouraging to note that most of the errors are reasonable when compared to the mistakes performed by human listeners. P&B's listeners achieved an error rate of approximately 5 percent (Table 5-5), with most of the confusions occurring among adjacent vowel categories. By inspecting the errors that occured in the preclassifiers, it is encouraging to note that the confusion lies with an entire vowel token rather than within a portion of a vowel. An example where one vowel was classified as another is shown in Figure 5-7. The entire

[†] See Syrdal and Gopal [SYRD86] for a discussion on jack-knifing. Basically train on N-1 tokens, and test with the remaining token. Then select a different set of N-1 tokens to train with and test with the remaining token. Do this for all N tokens.

vowel /ah/ was labeled as /uh/ which was reasonable, given its position in F_1-F_2 space. Basically, as Stevens and House [STEV63] would say, these vowels are undershooting their target position due to the coarticulatory effects of the context in which the vowel was articulated. This type of misidentification is the predominant error that occurred in the preclassification of the vowels. If there is a way to capture this undershooting of the vowel, the results achieved here suggest that vowel identification might be closer to that reported by Hillenbrand and Gayvert [HILL87].

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy	**99.9**	-	-	-	-	-	-	-	-	-
ih	-	**92.9**	6.8	-	-	-	-	-	-	0.3
eh	-	2.5	**87.7**	9.2	-	-	-	-	-	0.5
ae	-	-	2.9	**96.5**	0.1	-	-	-	-	0.4
ah	-	-	-	-	**92.2**	-	1.0	5.3	1.2	0.2
uw	-	-	-	-	-	**99.2**	0.8	-	-	-
uh	-	-	-	-	1.7	0.9	**96.5**	0.2	0.5	0.2
aa	-	-	-	0.2	2.2	-	0.7	**87.0**	9.9	-
ao	-	-	-	-	0.6	-	0.7	5.7	**92.8**	0.1
er	-	-	0.2	-	-	-	-	-	-	**99.7**
Total	10.0	9.5	9.8	10.6	9.8	10.4	10.0	10.4	9.7	10.1

Table 5-5 Confusion Matrix from the original Peterson and Barney study [PETE52]

Dynamic Classification

The last part of this study involved an attempt to model the dynamic nature of vowel trajectories through a feature space by taking the results of the preclassifiers and classifying the vowels over time with the aid of several hidden Markov models. A three-state left-to-right model using the Baum-Welsh algorithm [RABI89] was constructed for every vowel class that was investigated (Figure 4-4). Three states were chosen for each model in an attempt to capture the formant transitions of the onglides, central part, and offglides of the particular vowel. Tables C2-C21 in Appendix C list all the state values for each model derived from both the neural network and Gaussian preclassifiers. The results of taking

the phonetic pathway trained from the back-propagation preclassifier are shown in Table 5-6.

By comparing the results of Table 5-1b to Table 5-6, we can see how the Markov model is capable of taking a coarse pathway through phonetic space and mapping the vowel token to its correct class with a twenty percent increase in accuracy over that of the neural net alone. The word 'coarse' refers to the less than successful testing results produced by the back-propagation model. A logical explanation of why the jack-knife method of testing performed much better than the resubsitution method lies in the number of tokens used to train the individual models. Take, for example, the vowel class /ao/ which has only ten vowel tokens. The table suggests that each individual vowel token has a strong influence upon the testing results. Lee [LEE 89] suggests that a hidden Markov model would perform much better by over training the system. The size of the database used here suggests that 313 distinct vowel tokens is not enough to tap into the robustness of a Markov model.

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy (64)	**89.06**	4.68	–	–	–	6.25	–	–	–	–
ih (43)	4.65	**76.74**	4.65	–	2.33	4.65	4.65	2.33	–	–
eh (40)	–	5.0	**65.00**	12.50	5.0	2.50	2.50	5.0	–	2.50
ae (40)	–	2.50	7.50	**80.00**	2.50	–	–	7.50	–	–
ah (31)	–	9.68	9.68	3.23	**48.39**	–	16.13	12.90	–	–
uw(25)	4.0	8.0	4.0	–	–	**56.00**	24.00	–	–	4.0
uh (28)	–	12.0	12.0	–	7.14	17.86	**50.00**	3.57	–	–
aa (17)	–	5.88	5.88	5.88	29.41	–	5.88	**52.94**	–	–
ao (10)	–	–	–	–	30.00	–	–	40.00	**20.00**	10.00
er (15)	–	6.66	–	6.66	–	–	–	13.33	–	**73.33**
Total	19.0	16.0	12.0	12.0	9.0	8.0	9.0	8.0	0.6	4.0
Resubsitution						56.17				
Jack-knife						68.05				

Table 5-6 Confusion matrix results for several hidden Markov models after being trained from the neural net preclassification results. One model for each vowel class

The results of dynamic smoothing by taking the phonetic pathway produced by the Gaussian preclassifier are shown in Tables 5-7a and 5-7b. Two different tables are presented, which correspond to the two different results achieved by the Gaussian preclassifier. The first table represents the results of the hidden Markov model when its training pathways were produced by the resubsitution method of the Gaussian preclassifier. The second table is derived from taking the jack-knifed results produced by the Gaussian preclassifier and using it as training input for the HMM. The values in table 5-7b more accurately reflect the classification ability of the Gaussian preclassifier and HMM combination since it is not based upon the resubsitution method.

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy(64)	**90.62**	4.69	–	1.56	–	1.56	–	–	–	1.56
ih(43)	6.98	**72.09**	4.65	–	6.98	6.98	2.33	–	–	–
eh(40)	–	5.00	**62.50**	12.50	5.00	–	7.50	5.00	–	2.50
ae(40)	–	2.50	15.00	**70.00**	2.50	–	–	10.00	–	–
ah(31)	–	3.23	6.45	–	**29.03**	3.23	16.13	32.26	6.45	3.23
uw(25)	4.00	–	–	–	–	**80.00**	12.00	4.00	–	–
uh(28)	–	–	7.14	–	14.29	17.86	**53.57**	3.57	3.57	–
aa(17)	–	–	5.88	5.88	11.76	–	–	**58.82**	11.76	5.88
ao(10)	–	–	–	–	–	10.00	–	–	**90.00**	–
er(15)	–	–	–	–	–	–	–	–	–	**100.0**
Total	20.0	12.0	12.0	11.0	7.0	13.0	8.0	9.0	4.0	6.0
Resubsitution						70.29				
Jack-knife						66.77				

Table 5-7a Confusion Matrix for Gaussian classifier after dynamic smoothing by several hidden Markov models. Results shown here are achieved by using the results of the Gaussian preclassifier that was tested using the resubsitution method.

	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
iy(64)	**89.06**	4.69	–	3.12	–	1.56	–	–	–	1.56
ih(43)	9.30	**60.47**	9.30	–	2.33	11.63	4.65	2.33	–	–
eh(40)	–	7.50	**50.00**	20.00	5.00	2.50	5.00	7.50	–	2.50
ae(40)	–	2.50	17.50	**65.00**	2.50	–	–	12.50	–	–
ah(31)	–	6.45	3.23	–	**25.81**	3.23	19.35	32.26	6.45	3.23
uw(25)	4.00	4.00	4.00	–	–	**64.00**	16.00	4.00	–	4.00
uh(28)	–	–	7.14	–	10.71	21.43	**50.00**	–	7.14	3.57
aa(17)	–	–	5.88	5.88	11.76	–	–	**58.82**	11.76	5.88
ao(10)	–	–	–	–	10.00	10.00	–	10.00	**70.00**	–
er(15)	–	–	–	–	–	–	–	–	–	**100.0**
Total	20.0	12.0	12.0	12.0	6.0	10.0	9.0	10.0	4.0	7.0
Resubsitution						63.58				
Jack-knife						58.15				

Table 5-7b Confusion Matrix for Gaussian classifier after dynamic smoothing by several hidden Markov models. Results show here is achieved by using the results of the Gaussian preclassifier that was tested using the jack-knife method.

Average Center Values

Given that the classification of vowels in continuous speech is roughly 70 percent accurate with these models, work was conducted to find another way in which a vowel can be characterized in order to improve the classification accuracy. Given the results of previous studies that used a small portion of the vowel extracted from the steady state region or central part, it seemed reasonable to examine the effects of using the center value(s) in the Gaussian classifier that was developed for this study. Using a central value when measuring the formant frequencies, Hillenbrand and Gayvert [HILL87] achieved accuracy in the high 80% range (Table 3-2). Training the Gaussian classifier with formant frequency values extracted from the central part of the vowel yielded results that did not prove to be as accurate (Table 5-8). Averaging two, three, four or even five central values produced comparable results. These results do not suggest this is a bad approach, but it implies that initial and final vowel transitions provide important information about the identity of the vowel.

Features	2 Values	3 Values	4 Values	5 Values
F1,F2	43.70	45.22	45.00	44.78
F1,F2,mF1	50.22	50.65	51.30	50.65
F1,F2,mF1,mF2	–	51.30	50.43	50.87

Table 5-8 Averaging N center values to train a Gaussian classifier

Three-Frame Sampling

Realizing that this study is trying to classify vowels extracted from continuous speech, three frames were extracted from the vowel token in hopes that they would characterize the entire vowel more accurately. These three values were selected in a manner that would characterize the vowel formant transitions. Three values were selected for the same reason three states were selected for the Markov model. Various methods were used to select the times at which the three samples were taken. The first method that was tested involved sampling the formant frequency pattern at the first frame, center frame and last frame. Using only the first two formant frequency values, this idea achieved a success rate near 62 percent (Table 5-9). Moving both the front and back sampling location in a few frames did not seem to adversely affect the classification results. These results are practically the same as previous values found in this study even though the amount of data is significantly less. Adding the normalized formant frequency values into this three-frame sampling scheme significantly improved classification results.

	F_1,F_2	F_1,F_2 mF_1,mF_2
In 0 frames	61.90	75.24
In 1 frame	61.59	74.92
In 2 frames	60.63	74.60
In 3 frames	58.41	74.60

Table 5-9 Three frame sampling based upon the first two formant frequencies. This test was performed by a Gaussian model using the resubsitution method.

For this classification method, training and testing were performed with the same data due to the size of the database. Even though the results here match and

surpass those achieved by the Markov model, remember that the resubsitution method was used instead of jack-knifing to perform this experiment. These preliminary results suggest that the three-frame sampling method is a robust method with which to classify vowels. It is encouraging to note that there is no significant performance degradation due to the shifting of the frames at which the vowel token is sampled. This implies that a system using this three-frame concept might handle errors made in broad phonetic segmentation.

Projecting Results

One of the problems encountered in this thesis was the lack of data available to adequately train and then test this system. Early training results of the Gaussian preclassifier were roughly 75% accurate when using all of the 313 vowel tokens (Figure 5-13). By splitting the database in half, it is possible to truly test the capability of the Gaussian preclassifier used in this thesis. The training curve represents the results of the Gaussian preclassifier when training and testing is performed on the same database. The testing curve represents the results of the preclassifier when trained on half of the data and tested on the other half. The preclassifier is roughly 54% accurate with 156 tokens for training and testing. The effect of further splitting the database into quarters with equal distribution of vowel tokens enables us to see the disparity between training and testing formed by presenting the model with relatively few tokens. By connecting the dots, one can envision a logarithmetic progression if more data were available to train and test this system. Thus at 313 tokens the Gaussian preclassifier should be capable of accurately identifying vowel tokens at the rate of approximately 58%. Following the progression a bit further, it seems that the Gaussian preclassifier would be able to accurately classify the vowel tokens at a rate of 65%. Given the success rate of the hidden Markov model, one would then presumably be able to classify the vowel tokens at an even higher rate of accuracy.

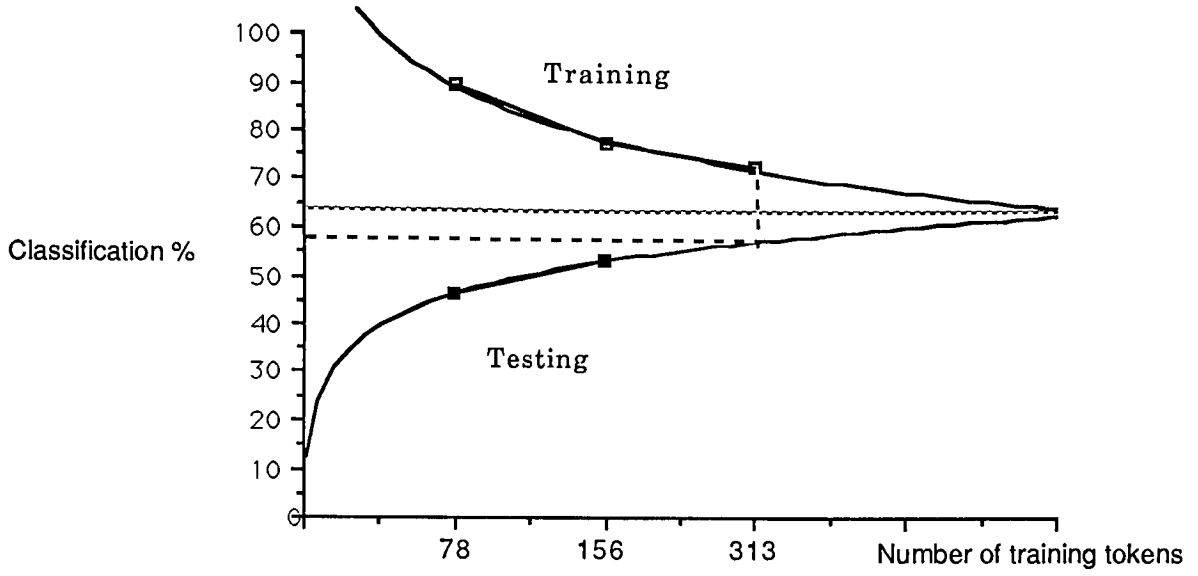


Figure 5-13 Projecting training results

Conclusions

As shown in previously sited studies [PETE52, STEV55, STEV61, STEV63, COLE80, HILL89], formant frequency values tend to be the best features that characterize a vowel. Unfortunately, a significant drawback in using these values reside in the difficulty of accurately tracking these frequency values. The spectral moment in the database were used to investigate the effects of vowel recognition based solely upon the spectral shape. As the tables in Appendix D show, the spectral shape is capable of classifying vowels, but with less accuracy than that found with the formant frequencies. Just because the spectral shape values did not perform as well as one would like, however they should not be completely ruled out. By combining the spectral shape information with formant frequency values, it can be noticed that the classification accuracy is modestly improved. The result of using the formant frequency values with the spectral shape information is quite similar to the results achieved if the spectral shape information were to be replaced by the normalized formant frequency values of each speaker. This is fortunate, since it is easier to calculate values from a distribution than to accurately track the formant frequency values and then normalize them in a timely fashion.

In comparing the overall classification accuracy of the Gaussian classifier to that of a neural net, one would like to say that it is still too early to make a fair

determination on which model is superior based upon the findings of this study. It seems however, that the neural net has more weaknesses than does the Gaussian classifier. This is unusual since they should be similar mathematically. A neural net should be more forgiving when presented irrelevant features since it can determine which feature is more useful in characterizing the desired classes. The results presented here contradict to this. The results achieved are significantly worse to that of the Gaussian model. An explanation for this is that is seems that the net tended to "memorize" the insignificant features and thus lose its ability to generalize about the specific class the training token came from. One way to determine if a neural net is capable of ignoring irrelevant features is to present the model with a significant amount of extra training data and compare the testing results of the model to earlier results. The neural net was presented with roughly 150 training tokens compared to the 313 tokens presented to the Gaussian classifier. This small data size does not allow one to make any strong correlations between the vowel classes or to justly compare the two models. At the moment, the Gaussian classifier is definitely the model of choice, especially since the amount of time required to train and test the model is significantly less by a total of 5 or more hours than that of the neural net.

When comparing the results achieved in this study to those sited earlier, one is initially disturbed by the poor outcome of the approach taken here to classify vowels in continuous speech. However, by further examining the classification ability of this model, one can see the potential success of dynamically identifying the vowels. This can be seen in the various tables in Appendix D. Classification accuracy always increased every time a new feature was added to the database, but we can see that the jump in accuracy decreased once the models were inundated with features. Presently it is unclear whether the model reached a point of saturation or a local maximum in feature set size. Unfortunately the only way to determine this would be to use different vowel tokens and rerun this study. Early static models that were sited [ZAHO88, STRA89, MILL89] roughly achieved classification accuracy rates 10 to 20% greater than this model. Seeing that dynamic classification of an entire vowel token from continuous speech is far more difficult than static classification of a single frame from a CVC environment, one can be pleased with the results achieved here. By increasing the database size and/or finding more characteristic features, the model presented here has the potential to match the results achieved at a static level.

The approach taken by this study seems to imply that dynamic classification of vowels is possible when preclassification is performed in a static fashion. In every case the HMM increased the classification accuracy of the static models. If the static models were capable of nearly 75 to 80 percent accuracy, it is conceivable that the dynamic model might be able to increase the performance up to the mid 90 percent range. Remember that the preclassifiers also were used to prevent the computationally expensive method of training a full Markov model. By presenting a phonetic pathway produced by the preclassifiers, one side steps the time intensive training method of the HMM. If one is looking for speed and a high rate of accuracy, then, the results here suggest using a Gaussian model as a front end to a hidden Markov model to recognize vowels in continuous speech.

Further Studies

One of the problems faced in this thesis was the lack of adequate amounts of hand-traced formant frequency data. The inadequacy does not reside with the amount of speech files available, but rather from the lack of a method to produce reliable formant frequency values. Once a procedure has been formulated, either with human interaction or automation, then further testing with the current model or the use of different models can be investigated. Some models include the Time Delay Neural Net (TDNN), Simulated Annealing, a Kohonen net, and Pineda's generalization of back-propagation [PINE87].

Even though the projected results suggest that using more data to train and test the Gaussian model would not improve the classification accuracy, it would be a good idea to verify or disprove this claim. There is a strong possibility that the projected results achieved were based upon context sensitive subsets of each vowel class. In other words, the model could be influenced by an /iy/ followed by a strong-fricative and then would fail when presented an /iy/ that is preceded by a strong-fricative. This could even be the case for the larger model. Thus, further work can be done in determining where the vowel identification error occurs. There is a strong possibility that ten classes are not enough to identify the ten different vowels. One should investigate to see if there is some sort of reoccurring pattern where a vowel in a certain context, say P_1 V P_2, is properly identified and a

vowel in P_3 V P_4 context is not; (where P_1, P_2 and P_3, P_4 are four phonemes that are not necessarily similar but are from similar coarse phonetic classes). This can be performed only with an extensive database full of multiple occurrences of vowels in a common context. An easy way to do this is by using some sort of cluster analysis scheme such as the K-Means work performed by Delmege [DELM89].

The results achieved by the three-frame sampling method are quite encouraging. Once a larger database is formed, more work can be done here. In fact, the first thing that could be performed is to rerun this simple study through the Gaussian classifier using the jack-knife method even before more data are made available. If the results only drop by a few percentage points, then this method of gathering sample points has the potential to produce better results in a timely manner than those achieved by using a dynamic model such as the hidden Markov model. Presently, only the first two formant frequency values, along with the normalizing values mF_1 and mF_2, have been used as a feature set to train a Gaussian model. Work can be applied in an exhaustive method to find an optimum feature set for this dynamic approach. The logical place to start is to add the third formant frequency value and then add its corresponding normalizing value. Then one can investigate what would happen if the spectral shape values were used to replace the normalizing values. The preliminary ability of the three-frame sampling method to classify the vowel better than the static method used in this study stems from the number of data values used to represent the vowel. That is, if only two features are used, then the static model only has two values to represent the vowel, whereas the three-frame method has six points.

If and when the three-frame method is thought to be the logical next step in classifying vowels, one should try to determine the proper location from which to select the center value. At the moment, the center value is taken at the midpoint of the vowel region. Figures 5-3 through 5-12 show us that the center value is not necessarily the most interesting value to examine. One should attempt to find the value that provides more meaning about the center region. One place to start looking for this value would be at the frame that has the highest energy, or the lowest rate of formant frequency change, or the greatest velocity change. For example, this can be performed by using some sort of windowing technique (such

as used by the autocorrelation routine) to compare groups of successive frames in order to find the desired meaningful frame.

Another modification that can be performed easily in the present architecture would be to investigate the use of other methods of spectral analysis. Currently, only linear predictive analysis has been used. It would be simple to change the LPC analysis box in stage I of Figure 4-1 to one that included the use of fast Fourier or discrete cosine transforms. One should be careful not to intermix the different transformed values during the training and testing stages. That is, it would be quite confusing to have pitch values extracted from a LPC spectra and formant frequency values taken from an FFT spectra.

Additional work should be done to make these models more flexible to speakers. To achieve this, a speaker independent feature set such as F_1, F_2, F_3, F0, M1 should be sought. That is, the models should not have to rely on the normalizing values, such as mF_1, mF_2, and mF_3, since these values are obtained by averaging the formant frequency values of an individual speaker over time. It may be the case that the features used in this study are not adequate to achieve this, and different features need to be found.

Finally, once a reliable method has been found to classify vowels, then more vowel classes should be added to the list of ten vowels. At the moment, only one phoneme for the central and retroflexed vowel classes exist in this model. Different vowel phonemes, such as /ix, ax, axr/, should be included as well as the addition of diphthongs and semi-vowels such as /ey, oy, l, r/. This addition is easily achieved by modifying the class-extractor or the phoneme-extractor routines to include the extra desired phoneme classes (e.g. diphthong) or individual phoneme (e.g. /oy/).

Chapter 6

User Documentation

All of the routines for this thesis were written in Common Lisp on a TI Explorer II Workstation. Over 100 routines have been written for this thesis, but only the important ones will be mentioned. These are the ones that require interaction with the user!

Building the database

1. PERFORM-SIGNAL-FUNCTIONS: (signal-list &optional (spectral-analysis 'lpc-spectra) (show-menu 't) (init-specifier nil))

This function takes a list of signals and performs signal processing routines upon them. The optional routines that can be performed are listed in both the signal-functions and signal-function-names constants. Some routines are: zero crossing rate, first moment, second moment, smoothed sift, relative energy, etc. A menu will "pop" up and allow the user to select which routines they wish to perform upon the signals. Each of the routines will perform the spectral analysis based upon that optional input feature. There are three possible routines that can be selected, either lpc-spectra, or fft-spectra or dct-spectra. Once the routines have been selected, then another menu will "pop" up and allow the user to modify the input parameters for the various selected routines. The show-menu option just allows the user to perform this routine in some sort of batch routine. If show-menu is nil, then all of the signal processing routines are not selected, and nothing will run, unless the init-specifier is set to T. The init-specifier sets all of the routines to be either true or false. This is useful if only one routine is desired, then set the specifier to nil, and the desired routine can be selected without having to say "no" to the others. Each signal processing routine performed will place itself onto the derived signal list of the original signal. That way it will not get misplaced. Thus this routine returns the original signal list.

2. CLASS-EXTRACTOR: (signal-list phoneme-class-list)

This function takes a list of signals and extracts all of the phonemes in the phoneme-class-list from each signal. Each signal has a label that is a list of starting, stopping times as well as a phoneme label for that particular region. Therefore, this routine just looks for a phoneme match between the phonemes in the phoneme-class-list and the phonemes in the signals' label. This returns a signal for each phoneme match found. The size of the signal depends upon the starting and stopping time of the label. This is similar to the phoneme-extractor routine which only extracts specific phonemes. (e.g. a phoneme-class-list would be '(front-vowel, central-vowel, ...) whereas a phoneme-list would be '(iy, eh, ae, oy, ...)).

3. BUILD-PS-FROM-DERIVED-SIGNALS: (signal-list &optional
(name nil) (static-feature-list nil))

This function takes a list of signals and builds a pattern-set from the derived signals found within each signal. This routine "pops" up a menu to ask the user to select which particular feature routines they wish to include in the pattern-set. Only the routines that are common to all of the signals will be allowed to be selected. If the user wishes to add an additional feature to the list of features in the signal-list, they can do so by using the static-feature-list option. For example, the average formant frequencies for each speaker was added at this point. This option requires a list of feature-name feature value lists; i.e. ((mf1 500) (mf2 1500) (mf3 2400)). Name is optional, and it just makes it easier to identify the pattern-set. The output of this function is an instance of a pattern-set flavor. A pattern-set is the data structure that interacts with the Gaussian classifier and Neural net.

4. SELECT-FEATURES-AND-CLASSES: (&optional (class-structure nil)
(init-state 't))

This method of a pattern-set does not require any input from the user. This routine will pop up a menu to allow the user to select which features in the pattern-set that the wish to use in the particular study. This sets the features-to-use variable within the pattern-set. The init-state just makes it easier to answer the questions in the menu. This method also sets the class-breakdown variable in this flavor. The optional argument class-structure allows the user to enter in a

list of symbols that they wish to group the individual classes in. An example of a class-structure that was used in this thesis is: '(ah aa er uw eh ae uh ih iy ao). This variable performs two tasks. First it allows one to select which classes in the pattern-set that are to be used verses those not to be used, and it also allows clustering of several classes. This method will pop up a menu and ask the user to select the particular clustering. Originally the vowel classes /er/ and /axr/ were clustered together. Later on, when it was decided to drop the /axr/ class, then all that was one needed to do was not select the class when clustering the vowels. Thus the cluster for the symbol 'er became one vowel class - specifically /er/. The routine will return a new pattern-set to the user.

Designing the Neural Network

1. INITIALIZE-NEURAL-NET: (input-info &optional (load-nn-info nil)
(layers nil) (number-of-passes 25) (eta .75) (alpha .25)
(results-to-use nil) (limit 'tanh) (confusion-file nil))

This function will create an instance of a neural net. Input-info is an instance of a pattern-set flavor with all the classes and features properly selected by the routine mentioned above. The load-nn-info is a boolean that asks the user if he wishes to train an already existing network. If so, then this routine will load an instance of the back-propagation neural network and continue to train the model. Layers is a list specifying the number of nodes in each of the hidden layers. If there is one value, (e.g. (18)) then a neural net is designed with one hidden layer with 18 nodes in the middle layer. The number of hidden layers depends upon the length of the list. Number-of-passes specifies how many times to train the model. Eta and alpha are momentum and gain terms used to adjust the training weights of the network. These values should between 0 and 1. When results-to-use is nil the entire pattern-set is used, otherwise only the classes that match the specific result are used. This is only used when designing a tree structure for the neural net. An example would be if a neural network clustered vowels into three classes: front, central, and back. Then results-to-use can be set to either one of these three values and then training will only occur on the values in the pattern-set that has a result label that matches. (This should be left alone unless working in a tree structured network). The optional limit variable specifies which type of limiting function to used when computing the output values of the

nodes. At the moment only the hyperbolic tangent and the inverse of the exponential are implemented in this routine. They perform approximately the same, but the hyperbolic tangent seems to train a bit faster, so that is why it is selected. The last optional parameter is the confusion file. If this value is not nil and is a valid pathname, then after every five training passes the neural network will print out a confusion matrix of the results. This is useful in determining if the network is performing as expected. This routine returns an instance of a back-propagation model.

2. NEURAL-NET-TRAIN (input-pattern-info)

This method of a back-propagation model only requires one parameter - a pattern-set. Presumably the pattern-set will have its features and classes selected by the above routine. This will call the routines that train the model, such as adjusting the weights, and computing the output. This returns both the pattern-set and the backprop instance. This way they can be saved for further training or testing.

3. NEURAL-NET-TEST (input-pattern-info)

This method of back-propagation tests the model based upon the inputed pattern-set. This routine will only work if the selected features and classes both match. This will return both the pattern-set and backprop instance.

Designing the Gaussian classifier

1. TRAIN-GAUSSIAN-CLASSIFIER (pattern-set &optional (name nil))

This function creates an instance of a gaussian-classifier. Like the neural network, this requires that the pattern-set has both the features and classes properly selected. Once an instance is created, this routine computes all the possible distributions based upon the class structure. A distribution is determined by the particular label given to the feature vector. This returns an instance of a gaussian-classifier with all the distributions collected in a list within the classifiers' flavor.

2. TEST-GAUSSIAN-CLASSIFIER (pattern-set &optional (jack-knife nil))

This method of a Gaussian-classifier requires a pattern-set as input. The features and classes do not have to be specified at this stage since the training routine handles that part. The jack-knife option allows the user to test his system when there is not enough training data. This was used often in this study. The idea behind jack-knife is that for every vowel token presented to the system, its influence in the design of the distribution is extracted from the system. In other words, if the vowel class /ao/ had 10 tokens that were used train the gaussian-distribution, then when presented one /ao/ token, its influence will be removed from the distribution all together. A new /ao/ distribution will be created from the remaining 9 tokens, and the one token can be tested. After testing, the one token will be placed back into the distribution and another token can be tested in the same fashion. This option is very time consuming, but is useful when not enough training data is available.

Designing the Hidden Markov Model

1. INIT-HIDDEN-MARKOV-MODEL (number-of-states symbol-list &optional (name nil) (even-distribution 't) (left-to-right 't) (epsilon 0.1))

This function will create a simple Markov model with number-of-states states. Symbol list is a list of symbols that could possibly be outputed at a particular state. That is, if the symbol list has three values, then each state will have three possible output values. Name is used to have label the model. It is also important to name the model when testing is later performed. During testing this name is used to assign a value to the result class. If the even-distribution boolean value is true, then each model will be initialized to have the same probability values. Otherwise, random values will be in each state. The boolean left-to-right is used to assign transition probabilities to each state in the model. If the model is not left to right, then each model will have equal transition probabilities. The epsilon value is used to avoid having a zero value in any of the probability values. If a zero value were to occur, then the multiplication of further values will also be zero and make training of the model more difficult. This is used during the Baum-Welsh training algorithm.

2. BUILD-SEQUENCE-SET (&optional (names 'result) (class nil))

This method of a pattern-set will create a new flavor called a sequence-set. The Markov model requires this type of input data to be presented to it. We no longer need the extensive data that is found in the feature array or in the context list, so this new flavor is a subset of a pattern-set. It will convert similar consecutive labels in the pattern-set into a sequence of concatenated symbols. The sequences are derived from the optional names parameter. Thus they can be derived from either the result-array or the original label-array of the pattern-set. The class parameter allows a smaller sequence-set to be built based upon a matching class. Thus one could say: (send pattern-set :build-sequence-set 'labels "iy") and a sequence-set with only sequences that are labeled /iy/ will be created from the label-array . This is useful when trying to train a Markov model for a particular class. (There is another routine that performs this option also. Given a sequence set, we can reduce the set to a specific class. i.e.: (send sequence-set :reduce-set "iy")).

3.TRAINING (sequence-set &optional (scalep nil) (epsilon-value 0.1))

This method of a Markov-model will train the individual model based upon the Baum-Welsh training algorithm. By presenting a Markov-model with a bunch of sequences in the sequence-set, this routine will adjust the models parameters to maximize the probability of the observances occurring more often. The optional scale parameter is used to prevent data underflow that sometimes occurs when multiplying small probability values. If a value gets to be too small it will be set to the epsilon-value to prevent unexpected errors from occurring.

4.EVALUATE-MULTIPLE-MODELS (model-list &optional (jack-knife nil))

This method of a sequence-set will present an individual sequence to every Markov model in the model-list and then the model with the greatest probability value of producing the sequence will be the resultant answer. The answer will be taken from the NAME field of the Markov model, so be sure that each model has an unique name! This routine calls the forward-backward algorithm. The optional jack-knife option is the same as the one in the test-gaussian-classifier routine. If it is set to nil, then every sequence in the sequence-set will be tested against every model in the model-list. Remember that one of the sequences was used to train the model, so this might influence a specific model.

For a more detailed discussion about Baum-Welsh, Forward-backward, Viterbi or Markov models, refer to Rabiners tutorial paper on Markov models that appeared in IEEE,**77** 2, Feb 1989.

Extra Useful Routines

PRINT-CONFUSION-MATRIX: method of both a sequence-set and a pattern-set. This routine enables one to view the results by the various models.

EXPLAIN-CONTEXT: method of a pattern-set. This routine will find all contexts for a specific vowel. This way, some of the errors that might have occured in classifying the vowel can be interpreted and hopefully understood.

SHOW-RESULTS: method of a markov-model. This routine will show the state transitions and output probabilities of the model. This is useful in building a markov-model, and also understanding why a model would produce a certain output.

SHOW-WEIGHTS, SHOW-STATES, SHOW-DELTA: methods of a backprop neural-net. These routines allows the user to gain insight into the workings of a neural net. If anyone can interpret these values - good luck!

PATTERN-PLOT-SPECS: method of a pattern-set this is able to graphically display features on a X-Y axis. This was used for some of the diagrams in chapter 5.

References

[ASSM87] Assmann, P., and Nearey, T. "Perception of Front Vowels: The role of Harmonics in the first Formant Region", *J. Acoust. Soc. Am.* **81**(2), 520-534, Feb 1987.

[BEAU87] Beauchamp, K.G. *Transforms for Engineers, A Guide to Signal Processing*. Oxford Science Publications, 1987.

[BORD80] Borden, G., and Harris, K. *Speech Science Primer, Physiology, Acoustics and Perception of Speech.* Baltimore, Williams and Wilkins 1980.

[COLE80] Cole, R.A., Rudnicky, A.I., Zue, V.W., and Readdy, D.R. "Speech as Patterns on Paper," in *Perception and Production of Fluent Speech* R.A. Cole, Ed. Hillsdale, New Jersey: Lawrence Erlbaum Assoc, 1980.

[DELM89] Delmege, James. "Class: A Coarse Phonetic Classifier", unpublished Masters Thesis, Rochester Institute of Technology, 1989.

[GAYV89] Gayvert, R. "A Statistical Approach To Formant Tracking", unpublished Masters Thesis, Rochester Institute of Technology, 1989.

[GERS68] Gerstman, L. "Classification of Self-Normalized Vowels", *IEEE Transactions on Audio and Electroacoustics.* **1**, March 1968.

[GOTT80] Gottfried, T., and Strange, W. "Identification of Coarticulated Vowels", *J. Acous. Soc. Am.* **68** (6), 1626-1635, 1980.

[HILL87] Hillenbrand, J., and Gayvert, R. "Speaker-Independent Vowel Classification Based On Fundamental and Formant Frequencies". NAIC Technical Series Report, 1987.

[HILL88] Hillenbrand, J., and Gayvert, R. "Effects of fundamental frequency contour on the identification of resynthesized vowels with static formant frequency patterns". NAIC Technical Series Report, 1988.

[HILL89] Personal conversation with Dr. Hillenbrand, 8 April 1989.

[HOUS53] House and Fairbanks, "The influence of Consonant Environment upon the Secondary Acoustical Characteristics of Vowels", *J. Acous. Soc. Am.* **25** (1), 105-113, 1953.

[LEE 89] Lee, Kai-Fu. *Automatic Speech Recognition, The development of the SPHINX System.* Kluwer Academic Publishers, 1989.

[LEHI61] Lehiste, I., and Peterson, G., "Transitions, Glides and Diphthongs", *J. Acous. Soc. Am.* **33**, 268-277, 1961.

[LIEB77] Lieberman, P. *Speech Physiology and Acoustic Phonetics.* New York, MacMillan Publishing Co., 1977.

[LIPP87] Lippmann, R. "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, 4-22, April 1987.

[MARK72] Markel, J. "The SIFT Algorithm for Fundamental Frequency Estimation". *IEEE Trans. Audio Electroacoustics, AU-20* **5**, 367-372, 1972.

[MILL87,89] Miller, J.D. "Auditory Perceptual Correlates of the Vowel". *J. Acous. Soc. Am. Suppl* 1, S79 (A) 1987, and 85 (5), 2114-2134, May 1989.

[MINI73] Minifie, F. *Normal Aspects of Speech, Hearing and Language.* New Jersey ,Prentice-Hall, 1973.

[ÖHMA66] Öhmann, S.E.G.,"Coarticulation in VCV Utterances: Spectrographic Measurements". *J. Acous. Soc. Am.* **39**. 151 - 168, Jan 1966.

[PARS86] Parsons, T. *Voice and Speech Processing.* New York, McGraw-Hill, 1986.

[PETE52] Peterson, G., and Barney, H. "Control Methods used in a Study of the Vowels", *J. Acous. Soc. Am.* **24**, 175-184, 1952.

[PICK80] Pickett, J. M. *The Sounds of Speech Communication.* Baltimore, University Park Press, 1980.

[PINE87] Pineda, Fernando. "Generalization of Back-Propagation to Recurrent Neural Networks". *Physical Review Letters*, **59** 19 2229-2232, 9 Nov. 1987.

[PRES88] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W., *Numerical Recipes.* New York, Cambridge University Press, 1988.

[RABI86] Rabiner, L.R., and Juang, B.H. "An Introduction to Hidden Markov Models". *IEEE ASSP Magazine*, 4-16, January 1986.

[RABI89] Rabiner, L.R. "Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proc. IEEE* **77**, 257-285, Feb 1989.

[RAKE84] Rakerd, Verbrugge, R., and Shankweiler, D. "Monitoring for Vowels in Isolation and in a Consonantal Context". *J. Acous. Soc. Am.* **76** (1), July 1984.

[SAS 82] SAS, Statistical Analysis System, *User's Guide: Statics,* 381-414, 1982 edition.

[STEV55] Stevens, K.A., and House, A.S. "Development of Quantative Description of Vowel Articulation", *J. Acous. Soc. Am.* **27**(3), 484-493, 1955.

[STEV61] Stevens, K.A., and House, A.S. "An Acoustical Theory of Vowel Production and Some of Its Implications". *J. Speech Hear.Res.* 4, 303-320, Dec 1961.

[STEV63] Stevens, K.A., and House, A.S. "Perturbation of Vowel Articulations by Consonantal Context: An Acoustical Study", *J.Speech Hear. Res.* 6(2), 111-128, June 1963.

[SHAN75] Shankweiler, D., Strange, W., and Verbrugge, R. "Speech and the Problem of Perceptual Constancy", in *Perceving, Acting and Comprehending: Toward an Ecological Psychology* ed. by R. Shaw and J. Bransford. Hillsdale, New Jersey, 1975.

[STRA87,89] Strange, W., "Evolving Theories of Vowel Perception", *J. Acous. Soc. Am.* Sup. 1, s16, 1987, and 85(5), 2081-2087, 1989.

[SYRD86] Syrdal, A.K., and Gopal, H.S. "A perceptual Model of Vowel Recognition based on the Auditory Representation of American English Vowels". *J. Acous. Soc. Am.* **79** (4), 1086-1100, Apr 1986.

[WILD75] Wilder, L., "Articulatory and Acoustic Characteristics of Speech Sounds", in *Understanding Language,* Massaro, D (ed) New York, Academic Press, 1975.

[ZAHO88] Zahorian, S., and Jagharghi, A. "Speaker Independent Automatic Vowel Recognition based on Overall Spectral Shape vs Formants". *J. Acous. Soc. Am. Suppl* 1988.

[ZUE 85] Zue, V., "The Use of Speech Knowledge in Automatic Speech Recognition". *Proc. IEEE*, 73(11), 1602-1615, Nov 1985.

Appendix A

IPA Symbol	Arpabet		Examples	IPA Symbol	Arpabet		Examples
i	i	IY	heed	v	v	V	verve
I	I	IH	hid	θ	T	TH	thick
e	e	EY	hayed	δ	D	DH	those
E	E	EH	head	s	s	S	cease
æ	@	AE	had	z	z	Z	pizzaz
a	a	AA	hod	ʃ	S	SH	mesh
ɔ	c	AO	hawed	ʒ	Z	ZH	measure
o	o	OW	hoed	h	h	HH	heat
U	U	UH	hood	m	m	M	mom
u	u	UW	who'd	n	n	N	noon
	R	ER	heard		G	NX	ringing
	x	AX	*a*head	l	l	L	lulu
^	A	AH	bud	l	L	EL	batt*le*
ai	Y	AY	hide	m	M	EM	bott*om*
au	W	AW	how'd	n	N	EN	but*ton*
ɔI	O	OY	boy		F	DX	ba*tt*er
	X	IX	ros*es*	?	Q	Q	§
p	p	P	pop	w	w	W	wow
b	b	B	bob	j	y	Y	yoyo
t	t	T	tug	r	r	R	roar
d	d	D	dug	tʃ	C	CH	church
k	k	K	kick	dʒ	J	JH	judge
g	g	G	gig		H	WH	where
f	f	F	fife				

Table 1 Phonetic alphabets *(Adapted from [PARS86])*

§ Glottal Stop

Phonetic classes used in the analysis of the preclassifier are as follows:

Front Vowel:	iy, ih, eh, ae
Central Vowel	ah, ix, ax
Back Vowel	aa, ao, uh, uw
Retroflexed Vowel	er, axr

Appendix B

The database used in this thesis consisted of 79 utterances from a large continuous speech database obtained from Carnegie-Mellon University. The following is a list of the specific utterances in which the vowel tokens were extracted. The database consists of both male and female speakers.

v1-1 They toiled in the fields all day long.
v1-2 The angry crowd pushed open the door.
v1-3 He left them with a reason to believe in themselves.
v1-4 The auctioneer accepted the bid.
v1-5 The yellow rose is the most beautiful of all flowers.
v1-6 The child lured the rabbit into the cage.
v1-7 Put the damp towel over your head for protection.
v1-8 Always look before you leap.
v1-9 While you were away, we opened the package.
v1-10 The acrobat walked the tightrope.

v2-1 He bought a new clock at the Tick-Tock Shop.
v2-2 She has a twenty percent hearing loss.
v2-3 She allowed the boy to eat the cookie.
v2-4 The old hound was unethused at the sight of the cat.
v2-5 The owl swooped down upon the mouse.
v2-6 The photograph proved he was guilty.
v2-7 A youth has many lessons to learn.
v2-8 If it never rained, we'd never grow.
v2-9 Where in the world is the Fountain of Youth?
v2-10 The handsome wool jacket was an oxford gray.

v3-1 A cooked yam is a tasty sweet potato.
v3-2 He recorded a new album with his younger partner.
v3-3 Take Cloey to the show.
v3-4 We fell for it hook, line, and sinker.
v3-5 She won a blue ribbon at the county fair.
v3-6 He was covered with soot from head to foot.
v3-7 Are you aware of the good things in life?
v3-8 Outside, the nights are only colder.
v3-9 The old woman rocked away the hours.
v3-10 He had a deep gouge over his left eye.

v4-1 No one aroused his curiosity like Eunice.
v4-2 Annoying a wild boar is insane.
v4-3 Get out before it's too late.
v4-4 The girl had a collection of wooden dolls.
v4-5 One is the loneliest number.
v4-6 The bull chased the clown from the arena.
v4-7 The thirsty girl rehearsed her lines.
v4-8 The little pooch wagged his tail.
v4-9 The hoodlum was full of malice.
v4-10 Why note make a white oak chair?

v5-1 You rang?
v5-2 A loud alarm can be an eye-opener.
v5-3 We arranged to look at the young animal.
v5-4 They stashed the loot in the pumpkin patch.
v5-5 Toast and jam tastes good for breakfast.
v5-6 The robot was programmed to clean house.
v5-7 The sauerkraut boiled till it burned.
v5-8 I am amused at the cowboy's style.
v5-9 Awhile ago, we knew very little.
v5-10 Try to remember the joyous occasions.

Appendix C

A hidden Markov model was used in this thesis in attempts to capture the dynamic nature of vowels in continuous speech. The following models (one for each vowel class) were trained by using the Baum-Welsh training algorithm [RABI89]. Note that the architecture for this study used neural net and maximum likelihood (Gaussian) preclassifiers as input to the hidden Markov stage. Each model was first initialized so that the output probabilities for each state in each model would be 0.01%. Each model was designed in a left to right fashion. Thus for a three state model, the state transitions for each node are as found in table C-1. A '–' symbol in these tables represents a zero value.

	Probability of being next state		
	State 1	State 2	State 3
State 1	.33	.33	.33
State 2	–	.50	.50
State 3	–	–	1.0

Table C-1 Probability of being in a state and moving to another state given a left to right model.

Probability of outputing a particular vowel label for class IY

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.56	.21	.03	–	.01	.02	.04	.02	–	.02
2	.61	.20	.03	–	.01	.02	.03	.02	–	.02
3	.82	.10	.01	.01	–	–	.01	.01	–	–

Table C-2 A 3 state Markov model trained from the phonetic pathway for the vowel class /iy/ that was produced by the Neural Network preclassifier.

Probability of outputing a particular vowel label for class **IH**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.15	.57	.03	.03	.03	.07	.08	.01	.02	.02
2	.13	.59	.04	.03	.03	.06	.08	.01	.01	.02
3	.11	.66	.05	.03	.03	.06	.05	–	.01	–

Table C-3 A 3 state Markov model trained from the phonetic pathway for the vowel class /ih/ that was produced by the Neural Network preclassifier.

Probability of outputing a particular vowel label for class **EH**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.05	.24	.23	.17	.11	.09	.09	.01	.01	–
2	.05	.20	.27	.18	.12	.07	.10	.01	.01	–
3	.03	.08	.38	.25	.09	.02	.12	.02	–	.02

Table C-4 A 3 state Markov model trained from the phonetic pathway for the vowel class /eh/ that was produced by the Neural Network preclassifier.

Probability of outputing a particular vowel label for class **AE**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	–	.15	.20	.41	.05	.01	.05	.05	.02	.06
2	.01	.15	.19	.44	.06	.01	.05	.06	.01	.04
3	.01	.07	.13	.67	.04	–	.02	.05	.01	–

Table C-5 A 3 state Markov model trained from the phonetic pathway for the vowel class /ae/ that was produced by the Neural Network preclassifier.

Probability of outputing a particular vowel label for class **AH**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.04	.13	.06	.05	.47	.03	.18	.01	–	.03
2	.03	.12	.06	.07	.45	.02	.20	.02	.01	.02
3	.02	.10	.05	.08	.42	.01	.24	.05	.02	.01

Table C-6 A 3 state Markov model trained from the phonetic pathway for the vowel class /ah/ that was produced by the Neural Network preclassifier.

Probability of outputing a particular vowel label for class **UW**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.25	.14	–	.01	–	.41	.19	–	–	–
2	.23	.11	–	.02	–	.43	.21	–	–	–
3	.07	.06	.01	.01	.01	.53	.29	–	–	.01

Table C-7 A 3 state Markov model trained from the phonetic pathway for the vowel class /uw/ that was produced by the Neural Network preclassifier.

Probability of outputing a particular vowel label for class **UH**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.02	.06	.08	.05	.06	.27	.43	.01	–	.02
2	.02	.09	.07	.04	.10	.25	.40	.02	.01	.01
3	.03	.10	.07	.03	.12	.21	.40	.03	.02	.01

Table C-8 A 3 state Markov model trained from the phonetic pathway for the vowel class /uh/ that was produced by the Neural Network preclassifier.

Probability of outputing a particular vowel label for class **AA**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.04	.06	.08	.09	.22	.04	.31	.15	–	–
2	.02	.07	.09	.12	.27	.02	.22	.18	–	–
3	.01	.04	.06	.23	.28	–	.12	.22	.02	.03

Table C-9 A 3 state Markov model trained from the phonetic pathway for the vowel class /aa/ that was produced by the Neural Network preclassifier.

Probability of outputing a particular vowel label for class **AO**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	–	–	.23	.01	.14	–	.25	.03	.18	.15
2	–	–	.18	.04	.18	–	.26	.04	.14	.15
3	–	.01	.03	.09	.28	–	.19	.03	.26	.11

Table C-11 A 3 state Markov model trained from the phonetic pathway for the vowel class /ao/ that was produced by the Neural Network preclassifier.

Probability of outputing a particular vowel label for class **ER**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	–	.13	–	–	.15	.05	.15	.02	–	.50
2	–	.10	.01	–	.16	.02	.15	.04	–	.52
3	–	.06	.01	.07	.06	.01	.08	.01	.01	.69

Table C-11 A 3 state Markov model trained from the phonetic pathway for the vowel class /er/ that was produced by the Neural Network preclassifier.

The following tables are the output probabilities for each model after being trained from the results of a Gaussian preclassifier.

Probability of outputing a particular vowel label for class **IY**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.57	.16	.03	–	.04	.12	–	–	–	.07
2	.62	.16	.02	–	.03	.11	.01	–	–	.06
3	.84	.08	.01	–	.01	.04	–	–	–	.01

Table C-12 A 3 state Markov model trained from the phonetic pathway for the vowel class /iy/ that was produced by the Gaussian preclassifier.

Probability of outputing a particular vowel label for class **IH**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.18	.47	.05	–	.07	.09	.11	–	–	.02
2	.17	.50	.06	–	.07	.09	.11	–	–	.02
3	.11	.62	.07	–	.05	.10	.03	–	–	.01

Table C-13 A 3 state Markov model trained from the phonetic pathway for the vowel class /ih/ that was produced by the Gaussian preclassifier.

Probability of outputing a particular vowel label for class **EH**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.03	.30	.27	.07	.07	.05	.12	.03	.01	.05
2	.04	.25	.36	.06	.10	.03	.08	.03	.01	.04
3	.01	.07	.53	.18	.08	.01	.02	.04	.03	.02

Table C-14 A 3 state Markov model trained from the phonetic pathway for the vowel class /eh/ that was produced by the Gaussian preclassifier.

Probability of outputing a particular vowel label for class **AE**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.05	.07	.21	.41	.06	.02	.03	.05	–	.10
2	.04	.08	.21	.44	.07	.01	.02	.05	–	.07
3	.01	.04	.17	.61	.03	–	–	.12	–	.01

Table C-15 A 3 state Markov model trained from the phonetic pathway for the vowel class /ae/ that was produced by the Gaussian preclassifier.

Probability of outputing a particular vowel label for class **AH**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	–	.07	.07	–	.34	.03	.19	.20	.07	.03
2	–	.06	.06	–	.35	.04	.19	.19	.08	.03
3	–	.05	.04	.03	.32	.04	.18	.23	.10	.01

Table C-16 A 3 state Markov model trained from the phonetic pathway for the vowel class /ah/ that was produced by the Gaussian preclassifier.

Probability of outputing a particular vowel label for class **UW**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.17	.07	–	–	–	.61	.11	–	–	.04
2	.16	.05	–	–	–	.64	.10	–	–	.04
3	.07	.01	–	–	.02	.69	.17	–	–	.03

Table C-17 A 3 state Markov model trained from the phonetic pathway for the vowel class /uw/ that was produced by the Gaussian preclassifier.

Probability of outputing a particular vowel label for class **UH**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	–	–	.04	–	.11	.28	.53	.04	–	–
2	–	–	.05	–	.12	.27	.51	.04	.02	–
3	–	.05	.06	–	.10	.15	.56	.02	.05	.02

Table C-18 A 3 state Markov model trained from the phonetic pathway for the vowel class /uh/ that was produced by the Gaussian preclassifier.

Probability of outputing a particular vowel label for class **AA**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	.06	.06	.06	.04	.21	–	.10	.26	.15	.06
2	.05	.07	.04	.02	.22	–	.07	.27	.19	.06
3	.01	.02	.03	.11	.08	–	.01	.61	.11	.02

Table C-19 A 3 state Markov model trained from the phonetic pathway for the vowel class /aa/ that was produced by the Gaussian preclassifier.

Probability of outputing a particular vowel label for class **AO**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	–	–	.11	.04	.01	.12	.08	–	.63	–
2	–	–	.08	.08	.02	.11	.06	–	.65	–
3	–	–	.01	.03	.01	.03	.05	–	.88	–

Table C-20 A 3 state Markov model trained from the phonetic pathway for the vowel class /ao/ that was produced by the Gaussian preclassifier.

Probability of outputing a particular vowel label for class **ER**

State	iy	ih	eh	ae	ah	uw	uh	aa	ao	er
1	–	–	–	–	–	.20	.10	–	.04	.67
2	–	–	–	–	–	.16	.05	–	.07	.73
3	.01	.01	.01	–	–	.03	.01	–	.02	.91

Table C-21 A 3 state Markov model trained from the phonetic pathway for the vowel class /er/ that was produced by the Gaussian preclassifier.

Appendix D

The following tables represent the best results achieved by the maximum likelihood distance measure based upon a cut-off range, which was determined based upon the general testing results. For example, out of a feature set of size three, the more interesting results appeared when the classification accuracy was greater than 50 %. The last set of features in each table represents the classification results if the model was presented a feature set that consisted of only spectral moments. Since multiple combinations of features are possible, only the best few are presented with the top performer being highlighted in bold face.

Features	Classification %
F_1, F_2, F_3	53.40
F_1, F_2, M1	50.17
F_1, F_2, M2	50.17
F_1, F_2, M3	51.62
F_1, F_2, F0	**55.68**
F_1, F_2, mF_1	54.52
F_1, F_2, mF_2	52.99
F_1, F_2, mF_3	53.93
M1, M2, M4	35.43
M1, M2, M5	34.96
M2, M3, M4	38.23

Table D-1 Feature set of size three with classification accuracy greater than 50%

Top 4 training features for a Gaussian classifier

Features	Classification %	Features	Classification %
$F_1, F_2, F_3, M1$	55.34	$F_1, F_2, F_3, M2$	55.54
$F_1, F_2, F_3, M3$	56.56	$F_1, F_2, F_3, M4$	55.14
$F_1, F_2, F_3, M5$	56.01	**$F_1, F_2, F_3, F0$**	**61.82**
F_1, F_2, F_3, mF_1	61.59	F_1, F_2, F_3, mF_2	61.39
F_1, F_2, F_3, mF_3	60.10	$F_1, F_2, M1, F0$	57.70
$F_1, F_2, M1, mF_1$	56.55	$F_1, F_2, M1, mF_2$	57.65
$F_1, F_2, M1, mF_3$	56.20	$F_1, F_2, M2, F0$	57.47
$F_1, F_2, M2, mF_1$	55.95	$F_1, F_2, M2, mF_2$	56.53
$F_1, F_2, M2, mF_3$	55.70	$F_1, F_2, M3, F0$	58.37
$F_1, F_2, M3, mF_1$	57.09	$F_1, F_2, M3, mF_2$	56.91
$F_1, F_2, M3, mF_3$	56.37	$F_1, F_2, M4, F0$	57.40
$F_1, F_2, M4, mF_1$	55.75	$F_1, F_2, M4, mF_2$	56.29
$F_1, F_2, M5, F0$	57.54	$F_1, F_2, M5, mF_1$	56.31
$F_1, F_2, M5, mF_2$	56.58	$F_1, F_2, M5, mF_3$	55.84
$F_1, F_2, F0, mF_1$	57.50	$F_1, F_2, F0, mF_2$	58.14
$F_1, F_2, F0, mF_3$	59.09	F_1, F_2, mF_1, mF_2	57.07
F_1, F_2, mF_1, mF_3	58.88	F_1, F_2, mF_2, mF_3	57.87
M1, M2, M3, M4	42.78	M1, M2, M4, M5	40.69
M1, M3, M4, M5	37.60	M2, M3, M4, M5	41.97

Table D-2 Feature set of size four with classification accuracy greater than 55 %

Top 5 training features for a Gaussian classifier

Features	Classification %	Features	Classification %
F_1, F_2, F_3, M1, F0	63.19	F_1, F_2, F_3, M1, mF_1	62.72
F_1, F_2, F_3, M1, mF_3	63.21	F_1, F_2, F_3, M1, mF_2	61.64
F_1, F_2, F_3, M2, F0	62.60	F_1, F_2, F_3, M2, mF_1	62.90
F_1, F_2, F_3, M2, mF_2	62.65	F_1, F_2, F_3, M2, mF_3	61.78
F_1, F_2, F_3, M3, F0	63.19	F_1, F_2, F_3, M3, mF_1	63.16
F_1, F_2, F_3, M3, mF_2	63.18	F_1, F_2, F_3, M3, mF_3	62.40
F_1, F_2, F_3, M4, F0	62.87	F_1, F_2, F_3, M4, mF_1	61.95
F_1, F_2, F_3,M4,mF_2	62.51	F_1, F_2, F_3, M4, mF_3	61.12
F_1, F_2, F_3, M5, F0	64.28	F_1, F_2, F_3, M5, mF_1	63.08
F_1, F_2, F_3, M5, mF_2	63.79	F_1, F_2, F_3, M5, mF_3	62.20
F_1, F_2, F_3, F0, mF_1	63.54	**F_1, F_2, F_3, F0, mF_2**	**64.89**
F_1, F_2, F_3, F0, mF_3	63.73	F_1, F_2,F_3,mF_1, mF_2	64.57
$F_1, F_2, F_3, mF_1, mF_3$	64.39	F_1, F_2,F_3, mF_2,mF_3	63.37
F_1, F_2, M1, M5, F0	60.61	F_1, F_2, M1, F0,mF_1	60.38
F_1, F_2, M1, F0, mF_2	61.24	F_1, F_2, M1, F0, mF_3	61.89
F_1, F_2, M1, mF_1, **mF_2**	60.16	F_1,F_2, M1,mF_1,mF_3	60.34
F_1, F_2, M1, mF_2, mF_3	60.23	F_1,F_2,M2, mF_1,mF_3	60.48
F_1, F_2, M3, M5, F0	60.65	F_1, F_2, M3, F0, mF_1	60.83
F_1, F_2, M3, F0, mF_2	60.97	F_1, F_2, M3, F0, mF_3	61.37
F_1, F_2, M4, F0, mF_3	60.01	F_1, F_2, M5, F0, mF_2	60.85
F_1, F_2, M5, F0, mF_3	60.81	F_1,F_2,M5,mF_1,mF_3	60.01
F_1, F_2, M5, mF_2, mF_3	60.00	F_1, F_2,F0,mF_1 ,mF_2	60.01
F_1, F_2, F0, mF_1, mF_3	61.66	F_1, F_2, F0,mF_2,mF_3	60.30
M1, M2, M3, M4,M5	44.50	M1, M2, M3, M4, F0	45.57

Table D-3 Feature set of size five with classification accuracy greater than 60 %

Top 6 training features for a Gaussian classifier

Features	Classification %	Features	Classification %
F_1, F_2, F_3, M1, M5, F0	65.43	F_1, F_2, F_3, M1, F0, mF_1	65.78
F_1, F_2, F_3, M1, F0, mF_2	66.05	F_1, F_2, F_3, M1, F0, mF_3	65.25
F_1, F_2,F_3,M1,mF_1, mF_2	66.26	F_1,F_2,F_3,M1,mF_1,mF_3	65.70
F_1,F_2,F_3, M1, mF_2,mF_3	65.13	F_1,F_2,F_3,M2,F0,mF_2	65.02
F_1, F_2,F_3,M2,mF_1, mF_2	65.90	F_1, F_2,F_3,M2,mF_1, mF_3	65.41
F_1, F_2, F_3, M3, M5, F0	65.70	F_1, F_2, F_3, M3, M5, mF_2	65.20
F_1, F_2, F_3, M3, F0, mF_1	65.41	F_1, F_2, F_3, M3, F0, mF_2	66.25
F_1, F_2, F_3,M3, F0, mF_3	65.50	F_1, F_2,F_3,M3,mF_1, mF_2	66.17
F_1, F_2,F_3,M3,mF_1, mF_3	65.87	F_1, F_2, F_3, M4, F0, mF_2	65.96
F_1, F_2,F_3,M4, mF_1,mF_2	65.67	F_1, F_2,F_3,M4,mF_1, mF_3	65.32
F_1, F_2, F_3, M5, F0, mF_1	65.72	F_1, F_2,F_3,M5,F0, mF_2	**66.99**
F_1, F_2, F_3, M5, F0, mF_3	65.99	F_1, F_2,F_3,M5,mF_1, mF_2	66.15
F_1, F_2,F_3,M5,mF_1, mF_3	65.70	F_1, F_2,F_3,M5, mF_2,mF_3	65.74
F_1, F_2,F_3, F0, mF_1, mF_2	65.99	F_1, F_2,F_3, F0, mF_1, mF_3	65.36
F_1, F_2, F_3, F0, mF_2, mF_3	66.21		

Table D-4 Feature set of size six with classification accuracy greater than 65 %

Top 7 features for training a Gaussian classifier

Features	Classification %
F_1, F_2, F_3, M4, F0, mF_1, mF_2	68.00
F_1, F_2, F_3, M5, F0, mF_2, mF_3	68.54

Table D-5 Feature set of size seven with classification accuracy greater than 68%

Appendix E - Glossary

Coarticulation: The natural phenomena when two phoneme sounds occur at the same time. That is, with vowels, they are often influenced by other phonemes.

Filter: The part of the head that filters and modifies the sounds produced by the source. The filter contains the tongue, mouth, nasal cavity, etc.

Formant Frequency: The dark bands of energy found in the spectrogram. Vowels can be distinguished from each other based upon the positions of the first three frequency values F1 and F2 correspond to the tongue height and advancement, F3 corresponds to whether the tongue points to the front or the back of the mouth.

Phoneme: The linguistic term for a sound produced by the filter. A phoneme can be transcribed by one or more alphabetic letters. See appendix A for a listing of various phonemes.

Source: The source of a phoneme sound begins at the lungs as an outward airflow and causes the vibration of the larynx.

Spectrogram: A three dimensional graph projected in two dimensions. This displays, time, amplitude, and frequency of a signal.

Vowel:

Back: When produced in isolation, the position of the vocal tract is consistent with the highest point of the tongue in the back of the mouth. The lips are quite rounded when the tongue is high and progressively less rounded as the tongue height decreases.

Central: The sound produced by the filter when the tongue is in a rest position. A schwa sound is normally produced.

Front: The position of the tongue, which is close to the lips, when a vowel is articulated.