Sādhanā Vol. 36, Part 5, October 2011, pp. 745–782. © Indian Academy of Sciences

Group delay functions and its applications in speech technology

HEMA A MURTHY[1,*] and B YEGNANARAYANA[2]

[1]Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai, India
e-mail: hema@cse.iitm.ac.in
[2]International Institute of Information Technology, Gachibowli, Hyderabad, India
e-mail: yegna@iiit.ac.in

Abstract. Traditionally, the information in speech signals is represented in terms of features derived from short-time Fourier analysis. In this analysis the features extracted from the magnitude of the Fourier transform (FT) are considered, ignoring the phase component. Although the significance of the FT phase was highlighted in several studies over the recent three decades, the features of the FT phase were not exploited fully due to difficulty in computing the phase and also in processing the phase function. The information in the short-time FT phase function can be extracted by processing the derivative of the FT phase, i.e., the group delay function. In this paper, the properties of the group delay functions are reviewed, highlighting the importance of the FT phase for representing information in the speech signal. Methods to process the group delay function are discussed to capture the characteristics of the vocal-tract system in the form of formants or through a modified group delay function. Applications of group delay functions for speech processing are discussed in some detail. They include segmentation of speech into syllable boundaries, exploiting the additive and high resolution properties of the group delay functions. The effectiveness of segmentation of speech, and the features derived from the modified group delay are demonstrated in applications such as language identification, speech recognition and speaker recognition. The paper thus demonstrates the need to exploit the potential of the group delay functions for development of speech systems.

Keywords. Fourier transform phase; group delay functions; feature extraction from phase; feature switching; mutual information; K-L divergence.

1. Introduction

Speech is the output of a quasistationary process, since the characteristics of speech change continuously with time. As the ear perceives frequencies to understand sound, speech is analysed

*For correspondence

and processed using the short-time Fourier transform (STFT). Representation of speech signals depends on the task for which the speech system is built. The acoustics of the speech waveform contains information about the speaker, language and sound unit. For a speech recognition system, the speaker information must be suppressed, while in a speaker recognition system, the information about the sound unit must be suppressed. For a language identification system, the phonotactics is important. A speech recognition system consists of two phases, namely, training (model building) and testing (model testing). In either phase, for any of the aforementioned tasks, the first task is feature extraction.

Information lost during the feature extraction process cannot be recovered. Feature extraction can be thought of as *lossy* encoding of the speech signal. It is thus important to make a judicious choice of the feature that is required for the task at hand. In any speech recognition system the features extracted are mel frequency cepstral coefficients (MFCC). MFCC are derived from the warped magnitude spectrum of the STFT of a frame of speech. Although both magnitude and phase of the STFT provide a complete description of the speech signal, it is generally believed that the ear is phase deaf. Experiments performed by Aarabi *et al* (2006) show that human recognition of speech in the presence of significant phase distortions is quite poor. A systematic approach to phase-based processing of speech signals can be found in Shi *et al* (2006). Paliwal *et al* (Paliwal & Alsteris 2005; Alsteris & Paliwal 2006) have performed a number of perceptual experiments to show the importance of phase in speech signals. Although the importance of phase in speech processing is established, there is very little effort on processing the phase spectrum to extract features for applications such as speech synthesis or recognition. Paliwal & Alsteris (2005) have shown that deviations in phase that are not linear are important for perception. Deviations in phase can be represented using the group delay function.

In this paper we discuss the significance of phase spectrum in speech processing. Phase by itself is difficult to process, we therefore propose processing the group delay function (negative derivative of phase).

The paper is divided into eight parts. In section 2, we briefly review the work on signal representation using phase spectrum. We discuss the properties of magnitude and phase spectra. The group delay function is defined, and its properties are studied. The issues in processing the group delay function to extract features for recognition are then discussed.

In section 3, algorithms for formant extraction from the group delay function of the speech signal are developed. In particular, three different ways of estimating group delay functions for formant extraction are studied. Although the extracted formants are quite robust, the representation of the spectrum in terms of formants frequencies alone is insufficient for building statistical speech recognition systems.

In section 4 methods for identifying syllable boundaries using group delay functions are developed. These boundaries can be considered as acoustic events, which can be exploited for various tasks as shown in the following sections.

In section 5, using the segmented models of syllables are built for every language. A new paradigm called *implicit language identification* is developed, where an unsupervised Hidden Markov model (HMM) clustering approach is used. This is based on the conjecture that humans use only a few key sounds to identify a language. Event-based processing of speech is used to improve the quality of synthesis, as shown in section 6, and to reduce the word error rate (WER) in speech recognition, as shown in section 7.

In section 7 we also develop a new paradigm for speech recognition, namely, feature switching. We show that use of information theoretic measures for choosing appropriate feature actually improves recognition performance. In section 8, feature switching as a paradigm for speaker verification is explored.

2. Representations of signals

In this section, we study different representations of the filter in the digital model for speech production given in figure 1.

In this model, it is clear that the parameters required to be extracted from the speech are *source* and *system* parameters. The changes in the vocal tract system, are characterized by the parameters of the digital filter in figure 1. The closing and opening of the glottis and vibrations of the vocal folds are characterized by the parameters of the source. The digital filter represents the formants (resonances) and antiformants (antiresonances). The speech signal $s[n]$ is modelled as the convolution of the source $e[n]$ and the impulse response of the digital filter $h[n]$:

$$s[n] = e[n] * h[n]$$

$$S(e^{j\omega}) = E(e^{j\omega})H(e^{j\omega}), \tag{1}$$

where $S(e^{j\omega})$, $E(e^{j\omega})$ and $H(e^{j\omega})$ are discrete-time Fourier transforms of $s(n)$, $e(n)$ and $h(n)$, respectively. The features that characterize the digital filter can be obtained directly from the spectrum or can be derived from the model. The source information may be derived by passing the speech signal through the inverse of the model system.

The source and system become additive as in homomorphic processing, i.e.,

$$\log S(e^{j\omega}) = \log E(e^{j\omega}) + \log H(e^{j\omega}).$$

Taking the inverse Fourier transform (IFT) on both sides, we get

$$\hat{s}[n] = \hat{e}[n] + \hat{h}[n], \tag{2}$$

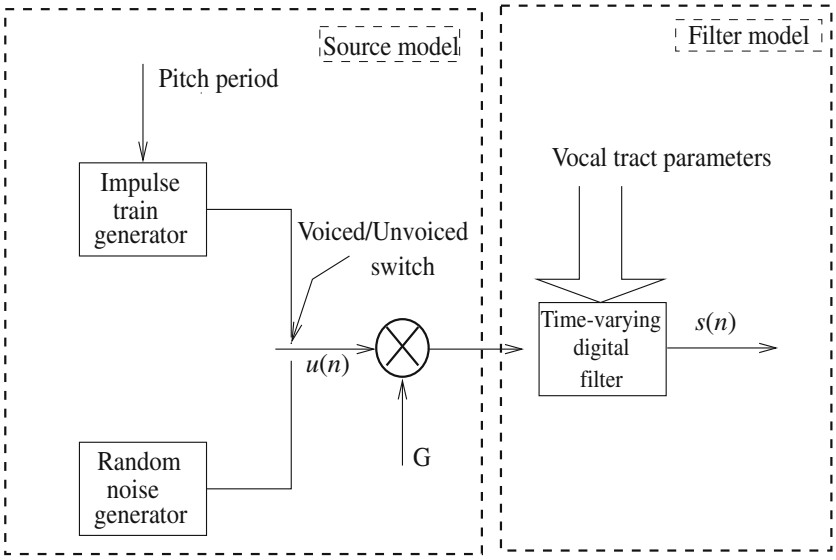

Figure 1. A source system model for speech production.

where $\hat{s}[n]$, $\hat{e}[n]$ and $\hat{h}[n]$ correspond to the cepstra of $s[n]$, $h[n]$ and $e[n]$, respectively. Since $\hat{e}[n]$ corresponds to the rapidly varying component of the spectrum, and $\hat{h}[n]$ corresponds to the slowly varying component in the spectrum, they can be separated by a lifter in the cepstral domain.

In the model-based approach to speech processing, the system $H(z)$ is represented by a digital filter. For the all-pole model,

$$s[n] = -\sum_{k=1}^{p} a_k s[n-k] + e[n].$$

Taking the z-transform on both sides, we get

$$S(z) = E(z)H(z). \tag{3}$$

$$H(z) = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \tag{4}$$

The a_ks are coefficients of a polynomial that approximate the envelope of the power spectrum of a frame of speech. These coefficients are estimated by minimising the squared error between the signal sample and its predicted value.

2.1 *Fourier transform representation of speech signal*

We list here some of the properties of the Fourier transform magnitude and phase spectra.

(a) Properties of the Fourier transform magnitude spectrum (FTMS)

- (i) For any real $x[n]$, the FTMS is an even function ω.
- (ii) The IFT of the FTMS is a noncausal even function of time. This function can be expressed as the autocorrelation function of some sequence $y[n]$ (Papoulis 1977). This signal is also called a zero phase signal.
- (iii) If a signal $x[n]$ is the impulse response of a cascade of resonators and antiresonators, the overall FTMS of $x[n]$ is a product of the magnitude spectra of the individual resonators. The resonators are characterized by peaks in the magnitude spectrum, while the antiresonators are characterized by the valleys in the magnitude spectrum.

(b) Properties of the Fourier transform phase spectrum (FTPS)

- (i) For any real $x[n]$, the FTPS is an odd function of ω.
- (ii) For any $x[n]$, the computed values of the FTPS are restricted to $\pm\pi$ (wrapped).
- (iii) If the signal $x[n]$ is shifted in the time domain by n_0 samples, a linear phase component of $e^{-j\omega n_0}$ is introduced in the FTPS.
- (iv) The IFT of $e^{j\theta(e^{j\omega})}$ is an all pass signal.
- (v) If a signal $x[n]$ is the impulse response of a cascade of resonators and antiresonators, the overall FTPS of $x[n]$ is a sum of the unwrapped phase spectra of individual resonators. The resonators and antiresonators are characterized by rapid variations in the phase function. These variations are better observed in the group delay function, which is the focus of the next section. Owing to the additive

property of the phase function, even low amplitude resonances are emphasized in the phase function.

(vi) Computation of the unwrapped phase function is nontrivial (Tribolet 1979).

2.2 *Group delay functions*

Definition:. Let $x[n]$ be a signal, whose continuous phase spectrum is given by $\theta(e^{j\omega})$. The group delay function is defined as

$$\tau(e^{j\omega}) = -\frac{d(\theta(e^{j\omega}))}{d\omega}. \tag{5}$$

If the Fourier transform of the sequence $v[n]$ is represented by

$$V(e^{j\omega}) = |V(e^{j\omega})|e^{j\theta(e^{j\omega})}, \tag{6}$$

then it can be shown that (Oppenheim & Schafer 1990)

$$\ln|V(e^{j\omega})| = c[0]/2 + \sum_{n=1}^{\infty} c[n]\cos n\omega \tag{7}$$

and the unwrapped phase function is given by:

$$\theta(e^{j\omega}) = -\sum_{n=1}^{\infty} c[n]\sin n\omega, \tag{8}$$

where $c[n]$ are the cepstral coefficients. A detailed description of the cepstrum and its properties can be found in Oppenheim & Schafer (1990). Taking the negative derivative of Eq. 8 with respect to ω, we get

$$\tau(e^{j\omega}) = -\sum_{n=1}^{\infty} nc[n]\cos n\omega. \tag{9}$$

From Eqs (7) and (8), we note that for a minimum phase signal, the spectral phase and magnitude are related through the cepstral coefficients. Further, the group delay function $\tau(e^{j\omega})$ can be obtained as the Fourier transform of the weighted cepstrum (Eq. 9).

For minimum phase signals, using the Eqs (7), (8) and (9), the signal can be directly obtained from its group delay function.

For mixed phase signals, two sets of cepstral coefficients are defined, namely, $c_1[n]$ and $c_2[n]$ for magnitude and phase functions separately (Yegnanarayana *et al* 1984):

$$\ln|V(e^{j\omega})| = c_1[0]/2 + \sum_{n=1}^{\infty} c_1[n]\cos n\omega \tag{10}$$

$$\theta(e^{j\omega}) = -\sum_{n=1}^{\infty} c_2[n]\sin n\omega, \tag{11}$$

where $c_1[n]$ and $c_2[n]$ are the cepstral coefficients of the unique minimum phase signals derived from the spectral magnitude and phase, respectively (Yegnanarayana *et al* 1984).

Using Eqs (9) and (11), two different group delay functions are defined:

$$\tau_m(e^{j\omega}) = -\sum_{n=1}^{\infty} nc_1[n]\cos n\omega$$

$$\tau_p(e^{j\omega}) = -\sum_{n=1}^{\infty} nc_2[n]\sin n\omega$$

as the group delay function derived from the magnitude and phase for a mixed phase signal, respectively.

2.2a *Properties of group delay functions*:

(i) Poles (zeros) of the transfer function show as peaks (valleys) in the group delay function.
(ii) Additive property: Convolution of signals in the time domain is reflected as a summation in the group delay domain.
(iii) High resolution property: The (anti) resonance peaks (due to complex conjugate (zero) pole pairs) of a signal are better resolved in the group delay domain than in the spectral domain. Furthermore, the resonance information is confined to the narrow region around the zero or pole location as shown in figure 2.
(iv) For minimum phase signals

$$\tau_p(e^{j\omega}) = \tau_m(e^{j\omega}).$$

(v) For maximum phase signals

$$\tau_p(e^{j\omega}) = -\tau_m(e^{j\omega}).$$

(vi) For mixed phase signals

$$\tau_p(e^{j\omega}) \neq \tau_m(e^{j\omega}).$$

(vii) If a root (pole or zero) is on the unit circle, at the location of the roots

$$\tau_p(e^{j\omega}) = \tau_m(e^{j\omega}) = \infty.$$

In figure 2, the additive property of the group delay spectrum is illustrated using three different systems, (i) a complex conjugate pole pair at an angular frequency ω_1, (ii) a complex conjugate pole pair at an angular frequency ω_2, and (iii) two complex conjugate pole pairs one at ω_1, and the other at ω_2. From the magnitude spectra of these three systems (figure 2b, e and h), it is observed that for a system consisting of two poles, the peaks are not resolved well (see figure 2h). This is due to the multiplicative property of the magnitude spectra. From figure 2i, it is evident that in the group delay spectrum obtained by combining the poles together, the peaks are resolved well. This is owing to the additive property of the group delay functions:

Let $\mathcal{H}_1(e^{j\omega})$, $\mathcal{H}_2(e^{j\omega})$ and $\mathcal{H}(e^{j\omega})$, be the frequency responses of the systems given in figures 2(I), (II) and (III), respectively. Then we have

$$\mathcal{H}(e^{j\omega}) = \mathcal{H}_1(e^{j\omega}).\mathcal{H}_2(e^{j\omega}) \tag{12}$$

$$|\mathcal{H}(e^{j\omega})| = |\mathcal{H}_1(e^{j\omega})|.|\mathcal{H}_2(e^{j\omega})|, \tag{13}$$

$$\arg(\mathcal{H}(e^{j\omega})) = \arg(\mathcal{H}_1(e^{j\omega})) + \arg(\mathcal{H}_2(e^{j\omega})). \tag{14}$$

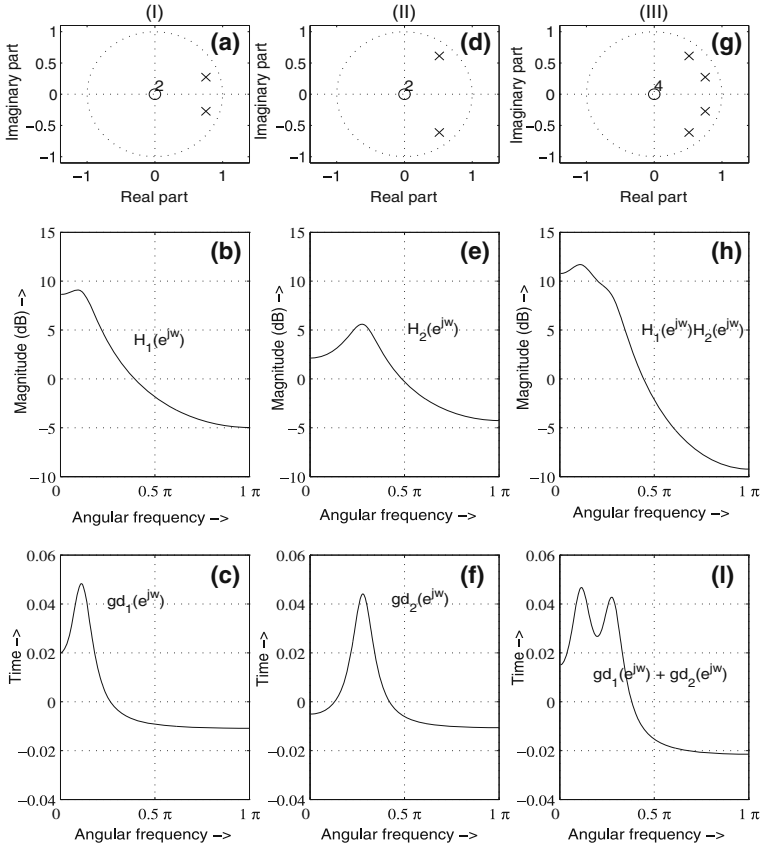

Figure 2. Resolving power of the group delay spectrum: z-plane (**a**, **d**, **g**), magnitude spectrum (**b**, **e**, **h**) and group delay spectrum (**c**, **f**, **i**). (**I**) A pole inside the unit circle at $(0.8, \pi/8)$. (**II**) A pole inside the unit circle at $(0.8, \pi/4)$. (**III**) A pole at $(0.8, \pi/8)$ and another pole at $(0.8, \pi/4)$ inside the unit circle.

The group delay function for the cascaded system is given by:

$$\begin{aligned}\tau_h(e^{j\omega}) &= -\partial(\arg(\mathcal{H}(e^{j\omega})))/\partial\omega \\ &= -\partial(\arg(\mathcal{H}_1(e^{j\omega})))/\partial\omega - \partial(\arg(\mathcal{H}_2(e^{j\omega})))/\partial\omega \\ \tau_h(e^{j\omega}) &= \tau_{h1}(e^{j\omega}) + \tau_{h2}(e^{j\omega}),\end{aligned} \tag{15}$$

where $\tau_{h1}(e^{j\omega})$ and $\tau_{h2}(e^{j\omega})$ correspond to the group delay function of $\mathcal{H}_1(e^{j\omega})$ and $\mathcal{H}_2(e^{j\omega})$, respectively.

From Eqs (12) and (15), we see that multiplication in the spectral domain becomes addition in the group delay domain.

2.2b *Minimum phase group delay function*: In the minimum phase group delay function, poles and zeros can be distinguished easily as peaks correspond to poles while valleys correspond to zeros. Non-minimum phase signals do not possess this property.

For analysis purposes, minimum phase and non-minimum phase signals in figure 3 are chosen such that the signals are causal outputs of stable systems. Further, the magnitude spectrum of all the three different signals are identical. The corresponding system function $\mathcal{H}(z)$ is given by

$$\mathcal{H}(z) = \frac{(z - b_1)(z - b_1^*)(z + b_2)(z + b_2^*)}{(z - a_1)(z - a_1^*)(z + a_2)(z + a_2^*)}, \tag{16}$$

where $|a_i| < 1$ for $i = 1, 2$ for all types of signals (stable systems).

Consider the following systems:

(i) minimum phase system: $|b_i| < 1$ for $i = 1, 2$
(ii) Type 1: $|b_1| < 1$ and $|b_2| > 1$
(iii) Type 2: $|b_i| > 1$ for $i = 1, 2$

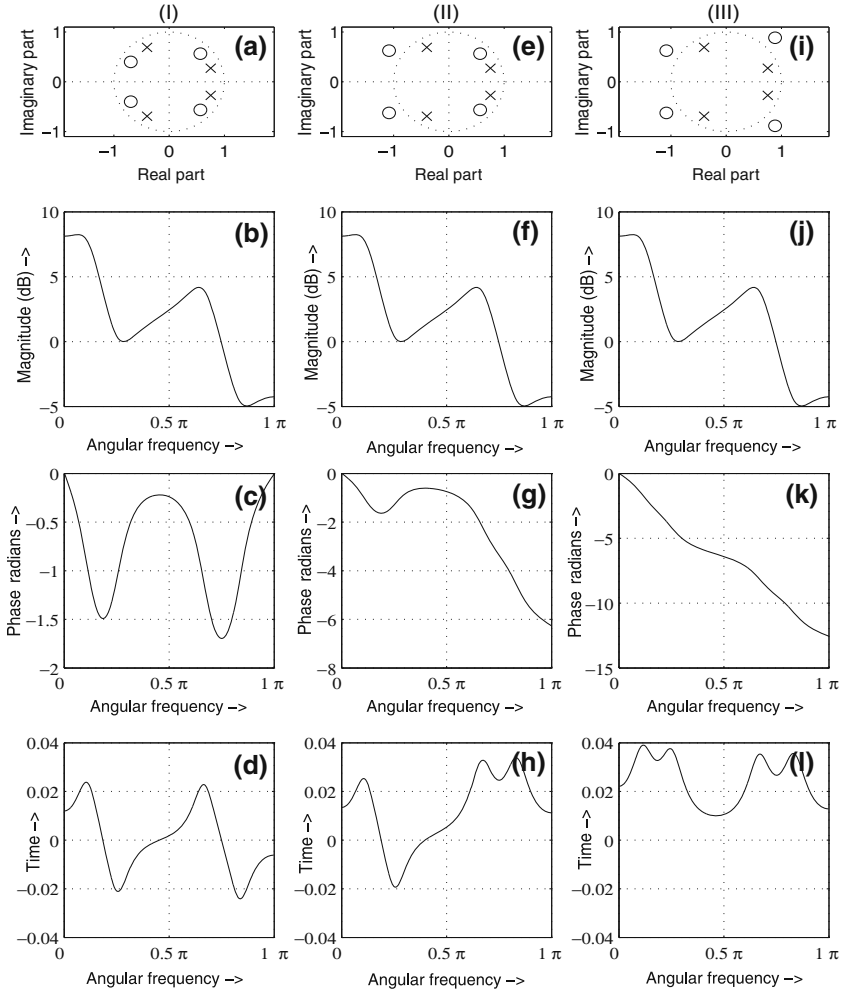

Figure 3. Group delay property of different types of signals: the z-plane (**a**, **e**, **i**), the magnitude spectrum (**b**, **f**, **j**), the phase spectrum (**c**, **g**, **k**), and the group delay spectrum (**d**, **h**, **l**) for (**I**) minimum phase (**II**) non-minimum phase – Type 1 and (**III**) non-minimum phase – Type 2 system.

The magnitude, phase, and, group delay spectra are computed from the system function given in Eq. 16 (see figure 3). From figure 3, we observe that:

i) For all the three types of systems, the magnitude spectra are identical in shape (figure 3b, f and j).
ii) For the minimum phase system (figure 3a, the net phase changes from 0 *to* π radians, $(\arg(\mathcal{H}(\pi)) - \arg(\mathcal{H}(0)))$ is negligible (figure 3c). For non-minimum phase systems (figure 3e and i), the net phase change is dependent on the number of zeros outside the unit circle (figure 3g and k).
iii) In the group delay spectrum, for the minimum phase system, both the peaks and valleys are resolved correctly (figure 3d), where peaks correspond to poles and valleys correspond to zeros. In the case of non-minimum phase systems, the zeros which are outside the unit circle are not resolved properly as shown in figure 3h and 3l. The zeros outside the unit circle appear as peaks at the corresponding angular frequencies. It is therefore, difficult to distinguish between poles and zeros (when the zeros are outside the unit circle) in the group delay spectrum.

A non-model, root cepstrum based approach is proposed, to derive a minimum phase signal $x_{mp}[n]$ from any signal $x[n]$ under the constraint that it is derived from the magnitude spectrum of $x[n]$, i.e. $|X(e^{j\omega})|$. The reason for this constraint is that the magnitude spectrum of a given root inside the unit circle (at a radial distance $'\alpha'$ from the origin of the unit circle) is the same as that of a root outside the unit circle (at a distance $'1/\alpha'$ at the same angular frequency). In general, if a system function has $\mathcal{N}$ roots, then there are $2^{\mathcal{N}}$ possible pole/zero configurations that will yield the same magnitude spectrum. Therefore, it is not possible to determine whether a given signal is minimum phase or non-minimum phase from the magnitude spectrum alone. However clearly, there is only one minimum phase system corresponding to that of the magnitude spectrum.

Instead of taking the squared magnitude spectrum, $|X(e^{j\omega})|^{\gamma}$ can be taken, where γ can be any value[1]. In that case, if the signal $x[n]$ is an energy bounded signal, from the Akhiezer–Krein and Fejer–Reisz theorems (Papoulis 1977), it can be shown that,

$$\begin{aligned}\mathcal{F}^{-1}(|X(e^{j\omega})|^{\gamma}) &= \mathcal{F}^{-1}(|X(e^{j\omega})|^{0.5\gamma}|X(e^{j\omega})|^{0.5\gamma}) \\ &= \mathcal{F}^{-1}\mathcal{Y}(e^{j\omega})\mathcal{Y}^{c}(e^{j\omega}) \\ &= y[n] * y[-n],\end{aligned}$$

where c and $*$ denote complex conjugation and convolution operations, respectively. Thus, $|X(e^{j\omega})|$ can be expressed as the Fourier transform of the autocorrelation of some sequence $y[n]$. Basically, the root cepstrum of any signal $x[n]$ can be thought of as the autocorrelation of some other sequence $y[n]$. The inverse Fourier transform of $|X(e^{j\omega}|^{\gamma}$ is referred to as the root cepstrum.

In Yegnanarayana (1979), the squared magnitude behaviour of the group delay function is explained. We review this property in the following.

---

[1] Other values of γ say, $\gamma < 1$, are especially useful in formant and antiformant extraction from the speech signal when the dynamic range is very high.

Consider the spectrum of any signal obtained as an impulse response of a cascade of M resonators, the frequency response of the overall filter is given by Yegnanarayana (1979).

$$X(e^{j\omega}) = \prod_{i=1}^{\mathcal{M}} \frac{1}{\alpha_i^2 + \beta_i^2 - \omega^2 - 2j\omega\alpha_i}, \tag{17}$$

where $\alpha_i \pm j\beta_i$ is the complex pair of poles of the i^{th} resonator. The squared magnitude spectrum is given by

$$|X(e^{j\omega})|^2 = \prod_{i=1}^{\mathcal{M}} \frac{1}{[(\alpha_i^2 + \beta_i^2 - \omega^2)^2 + 4\omega^2\alpha_i^2]} \tag{18}$$

and the phase spectrum is given by

$$\theta(e^{j\omega}) = \angle X(e^{j\omega}) = \sum_{i=1}^{\mathcal{M}} \tan^{-1} \frac{2\omega\alpha_i}{\alpha_i^2 + \beta_i^2 - \omega^2}. \tag{19}$$

It is well known that the magnitude spectrum of an individual resonator has a peak at $\omega^2 = \beta_i^2 - \alpha_i^2$ and a half-power bandwidth of α_i. The group delay function can be derived using Eq. 19 and is given by:

$$\tau(e^{j\omega}) = -\theta'(e^{j\omega}) = -\frac{d\theta(e^{j\omega})}{d\omega} = \sum_{i=1}^{\mathcal{M}} \frac{2\alpha_i(\alpha_i^2 + \beta_i^2 + \omega^2)}{(\alpha_i^2 + \beta_i^2 - \omega^2)^2 + 4\omega^2\alpha_i^2}. \tag{20}$$

For $\beta_i^2 >> \alpha_i^2$, the i^{th} term $\theta_i'(e^{j\omega})$ in Eq. 20 can be approximated around the resonance frequency $\omega_i^2 = \beta_i^2 - \alpha_i^2$, as in Yegnanarayana (1979) (here ω_i is the resonance frequency),

$$\tau(e^{j\omega}) = \theta_i'(e^{j\omega}) = \left[\frac{K_i}{(\alpha_i^2 + \beta_i^2 - \omega^2)^2 + 4\omega^2\alpha_i^2}\right] = K_i|H_i(e^{j\omega})|^2, \tag{21}$$

where K_i is a constant. Hence, the group delay function behaves like a squared magnitude response (Yegnanarayana 1979).

2.3 *Feature extraction from phase*

The group delay functions can be used to represent signal information as long as the roots of the z-transform of the signal are not too close to the unit circle in the z-plane (Murthy & Yegnanarayana 1989). The zeros that are close to the unit circle manifest as spikes in the group delay function, and the strength of these spikes is proportional to the proximity of these zeros to the unit circle. The group delay function becomes spiky in nature also owing to pitch periodicity effects. These spikes form a significant part of the fine structure of the spectrum, and cannot be eliminated by normal smoothing techniques. Hence, the group delay function has to be modified to eliminate the effects of these spikes (Murthy & Yegnanarayana 1991; Murthy 1997). Figure 4 shows the effects of zeros on the group delay function.

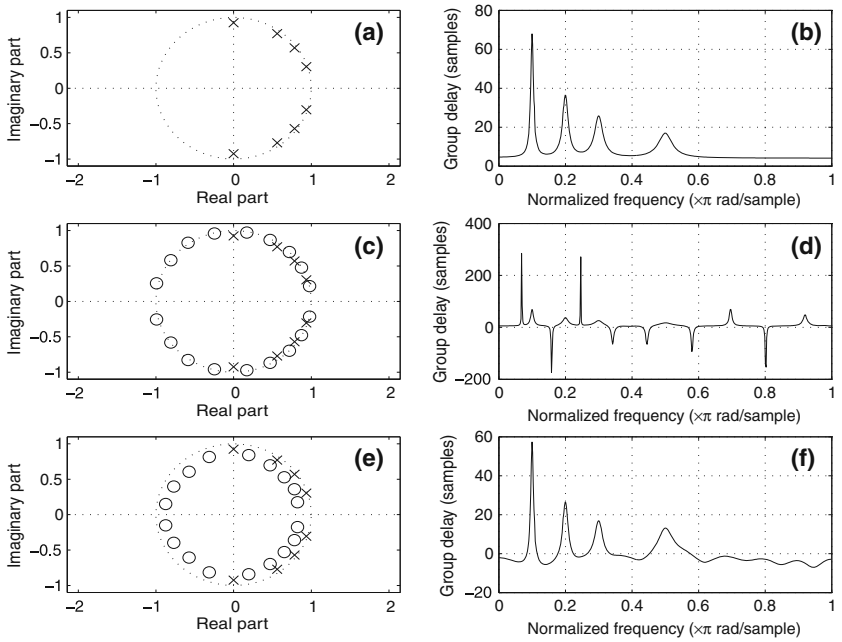

Figure 4. Significance of proximity of zeros to the unit circle. (**a**) The z-plane with four poles inside the unit circle. (**b**) The group delay spectrum of the system shown in (**a**). (**c**) The z-plane with four poles inside the unit circle and zeros added uniformly on the unit circle. (**d**) the group delay spectrum of the system shown in (**c**). (**e**) The z-plane with zeros pushed radially inwards into the unit circle. (**f**) the group delay spectrum of the system shown in (**e**).

3. Formant extraction using group delay functions

In this section, we develop a number of different approaches to reduce the effects of the roots that are close to the unit circle. The first is based on the linear prediction model, the second is a nonmodel-based technique based on spectral root homomorphic processing and the third approach modifies the equation corresponding to that of the non-minimum phase group delay function to estimate the vocal tract parameters accurately. Work by other researchers (Bozkurt *et al* 2007) include the computation of the chirp group delay function. Finally, the robustness of group delay functions in different noise scenarios is theoretically established.

3.1 *Formant extraction from linear prediction phase spectra*

In this method, the group delay function is computed from linear prediction (LP) phase spectra. Successive differences of the phase of the Fourier transform are computed to give the group delay function. In some cases, the group delay has sharp jumps owing to the wrapping of the phase. In such situations, the group delay is replaced by the average of the adjacent samples. The peaks of the group delay function correspond to formants. Figure 5a corresponds to synthetic vowel data. Figure 5b and f correspond to the LP group delay and LP magnitude spectrum, respectively, of the segment of speech given in figure 5a.

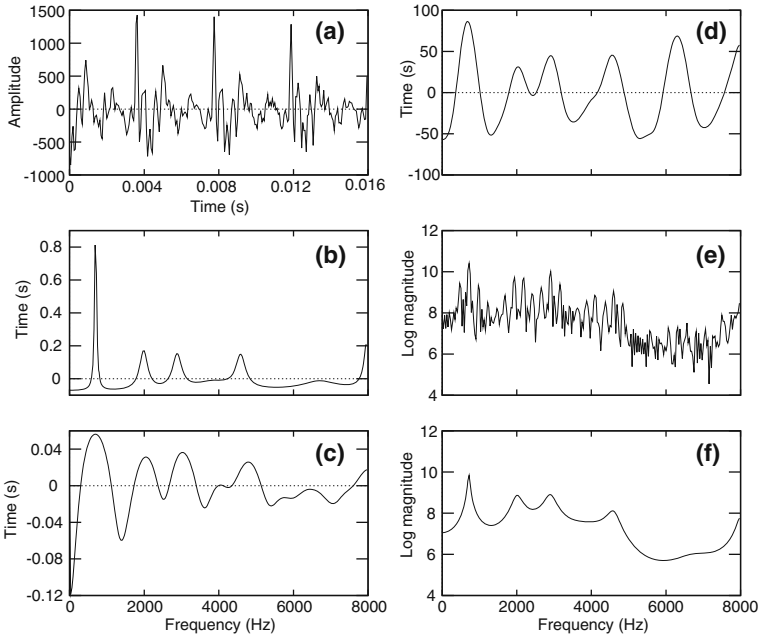

Figure 5. A segment of voiced speech and its corresponding spectra. (**a**) A segment of voiced speech. (**b**) Linear prediction phase group delay for the segment of (**a**). (**c**) Minimum phase group delay for the segment of (**a**). (**d**) Modified group delay for the segment of (**a**). (**e**) Magnitude spectrum of the segment of (**a**). (**f**) LP magnitude spectrum.

3.2 *Minimum phase group delay functions for formant estimation*

In this approach, a method is developed to extract the group delay function using spectral root homomorphic deconvolution (Lim 1979). The proposed method derives a signal with the characteristics of a minimum phase signal from the magnitude spectrum of the given signal. Peaks of the group delay function derived from this phase function correspond to formants. This technique is similar to cepstral based smoothing, except that (i) the γ^{th} power operation is performed in place of the log operation, and (ii) the phase group delay is computed instead of the smoothed magnitude spectrum. The exponent γ flattens the spectrum, and the vocal tract information is concentrated around the origin. The use of γ generates multiple peaks corresponding to multiples of the pitch period. Figure 5c corresponds to the minimum phase group delay function of figure 5a. Observe that the higher formants are emphasized well in the group delay spectrum compared to that figure 5f. As mentioned earlier, group delay processing requires that the zeros are not present on the unit circle. In the next section, a new algorithm is proposed to obtain an equivalent minimum phase group delay function from the standard phase.

3.3 *Modified group delay functions for formant estimation*

As mentioned in section 2.2, for the group delay function to be a meaningful representation, it is necessary that the roots of the transfer function are not too close to the unit circle in the z-plane. In this section, the computation of the group delay function is modified to reduce the effects of

the roots close to unit circle. The group delay function obtained directly from the speech signal as

$$\tau_x(e^{j\omega}) = -\{\mathrm{Im}\}[\frac{d(\log(X(e^{j\omega})))}{d\omega}] \tag{22}$$

$$= \frac{X_R(e^{j\omega})Y_R(e^{j\omega}) + Y_I(e^{j\omega})X_I(e^{j\omega})}{|X(e^{j\omega})|^2}, \tag{23}$$

where Im refers to the imaginary part, the subscripts R and I denote the real and imaginary parts of the Fourier transform. $X(e^{j\omega})$ and $Y(e^{j\omega})$ are the Fourier transforms of $x[n]$ and $nx[n]$, respectively. It should be noted that the denominator term $|X(e^{j\omega})|^2$ in Eq. 23 becomes zero or very small at zeros that are located close to the unit circle in the z-plane. The group delay function sampled on the unit circle in the z-domain becomes very spiky. Since, we are not interested in the source information, $|X(e^{j\omega})|$ in the denominator of the group delay function can be replaced by its envelope which corresponds to that of the system. In Eq. 23, $|X(e^{j\omega})|$, can be replaced with its cepstrally smoothed version $S_c(e^{j\omega})$.

The modified group delay function can be defined as:

$$\tau_c(e^{j\omega}) = \frac{X_R(e^{j\omega})Y_R(e^{j\omega}) + Y_I(e^{j\omega})X_I(e^{j\omega})}{S_c^2(e^{j\omega})}, \tag{24}$$

$\tau_c(e^{j\omega}))$ is referred as the modified group delay function. Figure 5d corresponds to the cepstrally smoothed modified group delay function of the segment of speech given in figure 5a.

Since the peaks at the formant locations are very peaky[2] in nature, two new parameters γ and α are introduced. This reduces the dynamic range of the modified group delay spectrum. The new modified group delay function is defined as

$$\tau_c(e^{j\omega}) = (\frac{\tau(e^{j\omega})}{|\tau(e^{j\omega})|})\ (|\tau(e^{j\omega})|)^{\alpha}, \tag{25}$$

where

$$\tau(e^{j\omega}) = (\frac{X_R(e^{j\omega})Y_R(e^{j\omega}) + Y_I(e^{j\omega})X_I(e^{j\omega})}{S_c(e^{j\omega})^{2\gamma}}). \tag{26}$$

The two parameters α and γ can vary such that ($0< \alpha \leq 1.0$) and ($0< \gamma \leq 1.0$).

Figure 6 shows the spectrogram of the synthesized speech signal. Figure 7a shows the formant data for the synthetic data in figure 6. Formants are extracted using different techniques and the results are illustrated in figures 7b to 7f. Observe that even closely spaced formants (around 0.6sec) are resolved well in figures 7b, 7c, and 7d, compared to those in figures 7e and 7f. This clearly illustrates that phase spectrum based techniques are able to resolve even closely spaced formants well, compared to that of magnitude spectrum-based techniques.

3.4 *Chirp group delay processing of speech signals*

Bozkurt *et al* (2007) developed a group delay function called the chirp group delay function. This is defined as the negative derivative of the phase spectrum computed from chirp z-transform

[2]This is due to the approximate compensation of zeros in Eq. 24.

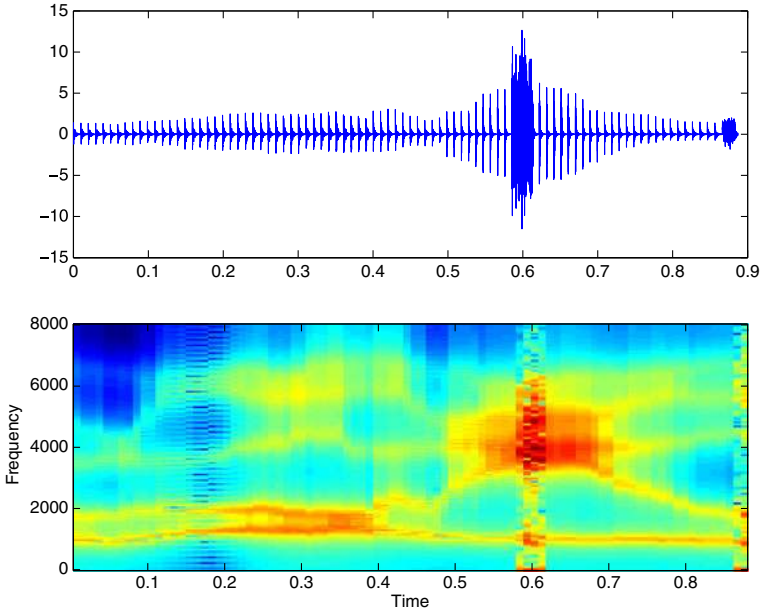

Figure 6. Synthesized speech waveform and its corresponding spectrogram.

(Rabiner & Schafer 1969). The chirp z-transform is the z-transform computed on a circle/spiral other than that on the unit circle. Given a signal $x[n]$, the chirp Fourier transform $\tilde{X}(e^{j\omega})$ is defined as:

$$\begin{aligned}\tilde{X}(e^{j\omega}) &= X(z)|_{z=\rho e^{j\omega}} \\ &= \sum_{n=0}^{N-1} x[n](\rho e^{j\omega})^{-n} \\ &= |\tilde{X}(e^{j\omega}|e^{j\tilde{\theta}(e^{j\omega})}\end{aligned} \tag{27}$$

and the chirp group delay function is given by

$$\tau_g(e^{j\omega}) = -\frac{d\tilde{\theta}(e^{j\omega})}{d\omega} \tag{28}$$

and

$$\begin{aligned}\tilde{X}(e^{j\omega}) &= \sum_{n=0}^{N-1} x[n](\rho e^{j\omega})^{-n} \\ &= \sum_{n=0}^{N-1} x[n]\rho^{-n}(e^{j\omega})^{-n} \\ &= \sum_{n=0}^{N-1} \tilde{x}[n](e^{j\omega})^{-n}.\end{aligned} \tag{29}$$

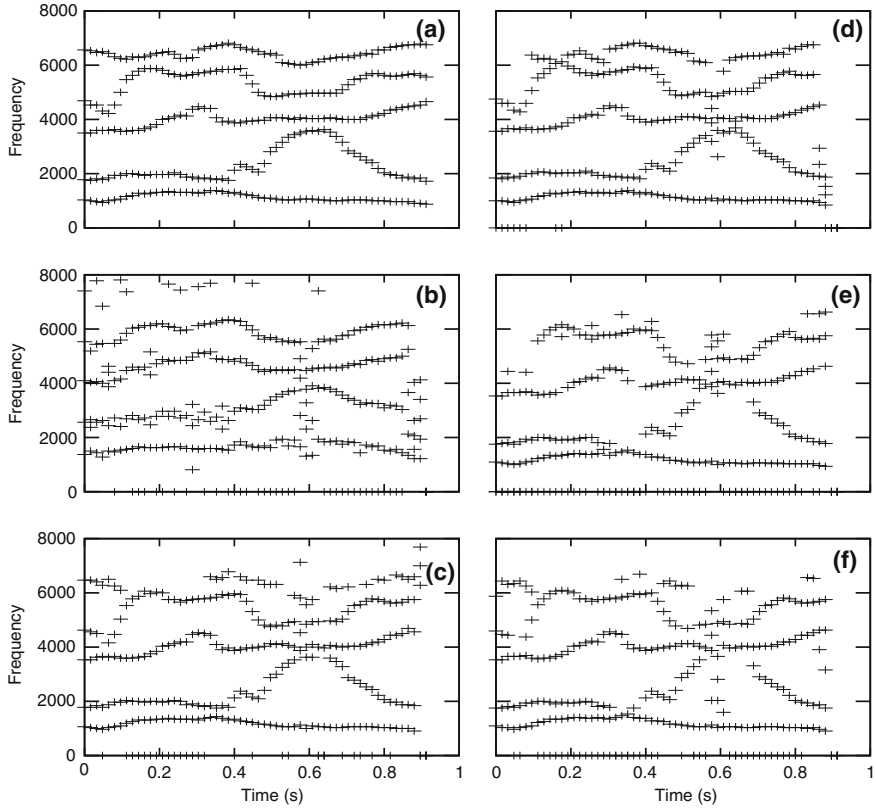

Figure 7. Formant extraction using various techniques. (**a**) Original data. (**b**) Formant extraction using linear prediction phase spectra. (**c**) Formant extraction using modified group delay. (**d**) Formant extraction using minimum phase group delay. (**e**) Formant extraction using linear prediction magnitude. (**f**) Formant extraction using cepstral smoothing.

The chirp z-group delay is computed with the window centre synchronized with the glottal closure instant. The window size length is chosen to be between T_0 and $2T_0$, where T_0 is the pitch period. These are then converted to cepstral coefficients similar to that in MFCC. In Bozkurt *et al* (2007), it is shown that extracted features help in significant reduction in the word-error-rates on the AURORA-2 (Hirsch & Pearce 2000) database at different noise levels.

3.5 *Robustness of group delay functions to additive noise*

In this section, the robustness of group delay functions to noise is established (Padmanabhan *et al* 2009; Yegnanarayana & Murthy 1992).

Let $x[n]$ denote a clean speech signal degraded by uncorrelated, zero-mean, additive noise $r[n]$ with variance $\sigma^2(e^{j\omega})$. Then, the noisy speech $y[n]$ can be expressed as

$$y[n] = x[n] + r[n]. \tag{30}$$

Taking the Fourier transform, we have

$$Y(e^{j\omega}) = X(e^{j\omega}) + R(e^{j\omega}). \tag{31}$$

Multiplying by the corresponding complex conjugates and taking the expectation, we have the power spectrum

$$P_Y(e^{j\omega}) = P_X(e^{j\omega}) + \sigma^2(e^{j\omega}), \tag{32}$$

where $P_Y(e^{j\omega}) = |Y(e^{j\omega})|^2$, $P_X(e^{j\omega}) = |X(e^{j\omega})|^2$ [3]. The power spectra of the resulting noisy speech signal can be related to noise power and (clean) speech power in one of three mutually exclusive frequency regions: (i) the high noise power region where $P_X(e^{j\omega}) \ll \sigma^2(e^{j\omega})$, (ii) the high signal power region where $P_X(e^{j\omega}) \gg \sigma^2(e^{j\omega})$, and (iii) the equal power region where $P_X(e^{j\omega}) \approx \sigma^2(e^{j\omega})$. The power spectrum of the noisy speech signal in each case is represented by $P_Y^n(e^{j\omega})$, $P_Y^s(e^{j\omega})$ and $P_Y^e(e^{j\omega})$, respectively. The group delay representation of noisy speech in the three cases mentioned is given by:

– *High noise power spectral regions ($P_Y^n(e^{j\omega})$)*: Consider frequencies ω such that $P_X(e^{j\omega}) \ll \sigma^2(e^{j\omega})$, i.e., regions where the noise power is higher than signal power. From Eq. 32, we have

$$\begin{aligned} P_Y^n(e^{j\omega}) &= P_Y(e^{j\omega}) \quad \forall\, \omega \quad \text{s.t.} \quad P_X(e^{j\omega}) \ll \sigma^2(e^{j\omega}) \\ &= P_X(e^{j\omega}) + \sigma^2(e^{j\omega}) \\ &= \sigma^2(e^{j\omega}) \left(1 + \frac{P_X(e^{j\omega})}{\sigma^2(e^{j\omega})}\right). \end{aligned}$$

Taking logarithms on both sides, using the Taylor series expansion of $\ln(1 + \frac{P_X(e^{j\omega})}{\sigma^2(e^{j\omega})})$, and ignoring the higher-order terms,

$$\begin{aligned} \ln\left(P_Y^n(e^{j\omega})\right) &= \ln\left[\sigma^2(e^{j\omega})\left(1 + \frac{P_X(e^{j\omega})}{\sigma^2(e^{j\omega})}\right)\right] \\ &\approx \ln\left(\sigma^2(e^{j\omega})\right) + \frac{P_X(e^{j\omega})}{\sigma^2(e^{j\omega})}. \end{aligned} \tag{33}$$

Expanding $P_X(e^{j\omega})$ as a Fourier series, ($P_X(e^{j\omega})$ is a periodic continuous function of ω with period $\omega_0 = 2\pi$),

$$\ln\left(P_Y^n(e^{j\omega})\right) \approx \ln\left(\sigma^2(e^{j\omega})\right) + \frac{1}{\sigma^2(e^{j\omega})}\left[\frac{d_0}{2} + \sum_{k=1}^{\infty} d_k \cos\left(\frac{2\pi}{\omega_0}\,\omega\, k\right)\right], \tag{34}$$

where d_ks are the Fourier series coefficients in the expansion of $P_X(e^{j\omega})$. Since $P_X(e^{j\omega})$ is an even function, the coefficients of the sine terms are zero.

For a minimum-phase signal, the group delay function can be computed in terms of the cepstral coefficients of the log-magnitude spectrum, as given in Yegnanarayana *et al* (1984) (see Eq. 9). From Eq. 9, it can be observed that the group delay function can be obtained from the log-magnitude response by ignoring the dc term, and by multiplying

[3] Assume expectation of noise is zero

each coefficient with n. Using this observation in Eq. 34, we get the group delay function as

$$\tau_{Y^n}(e^{j\omega}) \approx \frac{1}{\sigma^2(e^{j\omega})} \sum_{k=1}^{\infty} k\, d_k \cos(\omega k). \tag{35}$$

This expression shows that the group delay function is inversely proportional to the noise power ($\sigma^2(e^{j\omega})$) in regions where the noise power is greater than the signal power.

– *High signal power spectral regions ($P_Y^s(e^{j\omega})$)*: Now consider frequencies ω such that $P_X(e^{j\omega}) \gg \sigma^2(e^{j\omega})$. Starting with Eq. 32, and following the steps similar to the earlier analysis, we get

$$\ln\left(P_Y^s(e^{j\omega})\right) \approx \ln\left(P_X(e^{j\omega})\right) + \frac{\sigma^2(e^{j\omega})}{P_X(e^{j\omega})}. \tag{36}$$

Since $P_X(e^{j\omega})$ is non-zero, continuous and periodic in ω, $\frac{1}{P_X(e^{j\omega})}$ is also periodic and continuous. Consequently, $\ln(P_X(e^{j\omega}))$ and $\frac{1}{P_X(e^{j\omega})}$ can be expanded using Fourier series, giving

$$\ln\left(P_Y^s(e^{j\omega})\right) \approx \frac{d_0 + \sigma^2(e^{j\omega})\, e_0}{2} + \sum_{k=1}^{\infty} \left(d_k + \sigma^2(e^{j\omega})\, e_k\right) \cos(\omega k).$$

Using Eq. 9, and following the steps as in the previous case, we obtain the expression for the group delay function as,

$$\tau_{Y^s}(e^{j\omega}) \approx \sum_{k=1}^{\infty} k\, (d_k + \sigma^2(e^{j\omega})\, e_k) \cos(\omega k), \tag{37}$$

where d_ks and e_ks are the Fourier series coefficients of $\ln(P_X(e^{j\omega}))$ and $\frac{1}{P_X(e^{j\omega})}$, respectively. It is satisfying to observe that if $\sigma^2(e^{j\omega})$ is negligible, the group delay function can be expressed solely in terms of the log-magnitude spectrum.

– *Signal power ≈ noise power regions ($P_Y^e(e^{j\omega})$)*: For frequencies ω such that $P_X(e^{j\omega}) \approx \sigma^2(e^{j\omega})$, we again start with Eq. 32, and follow steps similar to those in the previous case, except, that in this case we do not need the Taylor series expansion:

$$\begin{aligned} P_Y^e(e^{j\omega}) &\approx 2P_X(e^{j\omega}) \\ \ln\left(P_Y^e(e^{j\omega})\right) &\approx \ln 2 + \ln\left(P_X(e^{j\omega})\right). \end{aligned} \tag{38}$$

Expanding $\ln\left(P_X(e^{j\omega})\right)$ as a Fourier series, since it is a periodic and continuous function of ω with a period 2π, the group delay function can be computed as,

$$\tau_{Y^e}(e^{j\omega}) \approx \sum_{k=1}^{\infty} k\, d_k \cos(\omega k), \tag{39}$$

where d_ks are the Fourier series coefficients of $\ln(P_X(e^{j\omega}))$.

Summarising the behaviour of minimum phase group delay functions in noise, we have:

From Eqs. 35, 37 and 39, the estimated group delay functions are summarized respectively for the three cases:

$$\tau(e^{j\omega}) \approx \begin{cases} \frac{1}{\sigma^2(e^{j\omega})} \sum_{k=1}^{\infty} k\, d_k \cos(\omega k) & \text{for } \omega \ : \ P_X(e^{j\omega}) \ll \sigma^2(e^{j\omega}), \\ \sum_{k=1}^{\infty} k\, (d_k + \sigma^2(e^{j\omega})\, e_k) \cos(\omega k) & \text{for } \omega \ : \ P_X(e^{j\omega}) \gg \sigma^2(e^{j\omega}), \\ \sum_{k=1}^{\infty} k\, d_k \cos(\omega k) & \text{for } \omega \ : \ P_X(e^{j\omega}) \approx \sigma^2(e^{j\omega}). \end{cases} \quad (40)$$

From Eq. 40, we note that the group delay function of a minimum-phase signal is inversely proportional to the noise power for frequencies corresponding to high noise regions in the power spectrum. Similarly, for low noise regions, the group delay function becomes *directly* proportional to the signal power. In other words, its behaviour is similar to that of the magnitude spectrum. This shows that the group delay function of a minimum-phase signal preserves the peaks and valleys in the magnitude spectrum well even in the presence of additive noise.

In this section, we looked at properties of the group delay function. In the next few sections, we show how some of these properties can be exploited in building practical systems. In the process of building systems, a number of new paradigms for specific application are also developed.

4. Speech segmentation using group delay functions

Conventional speech recognition systems transcribe a given speech utterance based on statistical models that are iteratively trained. The accuracy of the transcription depends critically on the amount of training data. Not much effort has been made to exploit the detection of acoustic cues in the speech signal in building recognition systems.

Many languages of the world possess a relatively simple syllable structure consisting of several canonical forms (Greenberg 1999). The syllables contain just two phonetic segments, typically of CV type (for example, Japanese language)[4]. The remaining syllabic forms are generally of V or VC variety. In contrast, English and German possess a highly heterogeneous syllable structure, in that the onset and (or) coda often contain two or more consonants. Further, in both stress and syllable-timed languages there is a preference for CV syllabic forms in spontaneous speech. Nearly half of the forms in English and over 70% of the syllables in Japanese are of this variety. There is also a substantial proportion of CVC syllables in spontaneous speech in both the languages (Greenberg 1999). An analysis of the switchboard corpus shows that nearly 88% of the syllables are of simple structure, and only 12% of the syllables belong to more complex structure with consonant clusters (Greenberg 1999).

The syllable consists of three parts, the onset, rime and coda. The onset and coda can consist of consonants, while the rime consists of vowel. The definition of a syllable in terms of short-term energy function is suitable for almost all the languages, in the case of spontaneous speech. The vowel region corresponds to much higher energy region compared to that of a consonant region. A time-domain acoustic segmentation approach is now proposed. This approach segments the speech signal into syllable-like units, without the knowledge of phonetic transcription.

---

[4]C corresponds to the consonant and V corresponds to the vowel

The similarity between the Fourier transform magnitude spectrum and the short-term energy signal, is exploited. The property that any arbitrary positive function can be thought as a magnitude spectrum of a minimum phase signal is exploited for segmenting the speech signal into syllables. A variation of the algorithm proposed for formant extraction in section 2.2 is used for determining the syllable boundaries. Applications of segmentation information for speech synthesis and recognition can be found in sections 6 and 7.

4.1 *Short-term energy based segmentation*

The high energy regions in the short-term energy (STE) function correspond to syllable nuclei, and the valleys at both ends approximately correspond to syllable boundaries. The raw STE function cannot be directly used to perform segmentation due to significant local energy fluctuations, owing to the presence of transient consonants and f_0 (see figure 8b). Techniques such as fixed or even adaptive thresholding will not work when the energy variation across the signal is quite high.

To overcome the problems of local energy fluctuations, the STE function should be smoothed. Smoothing the STE function can be performed in several ways. The STE function can be computed with a large window size. This will lead to a shift in the boundary locations. The STE function is normally smoothed with a moving average filter (see figure 8d). The order of the filter will affect the boundaries (i) too large an order can result in shift in boundaries or missed boundaries and (ii) too short an order can result in false alarms. In Greenberg *et al* (1996), it is mentioned that the syllable duration can be conceptualized in terms of *modulation frequency*. For example, a syllable duration of 200 ms is equivalent to a modulation frequency of 5Hz. Further,

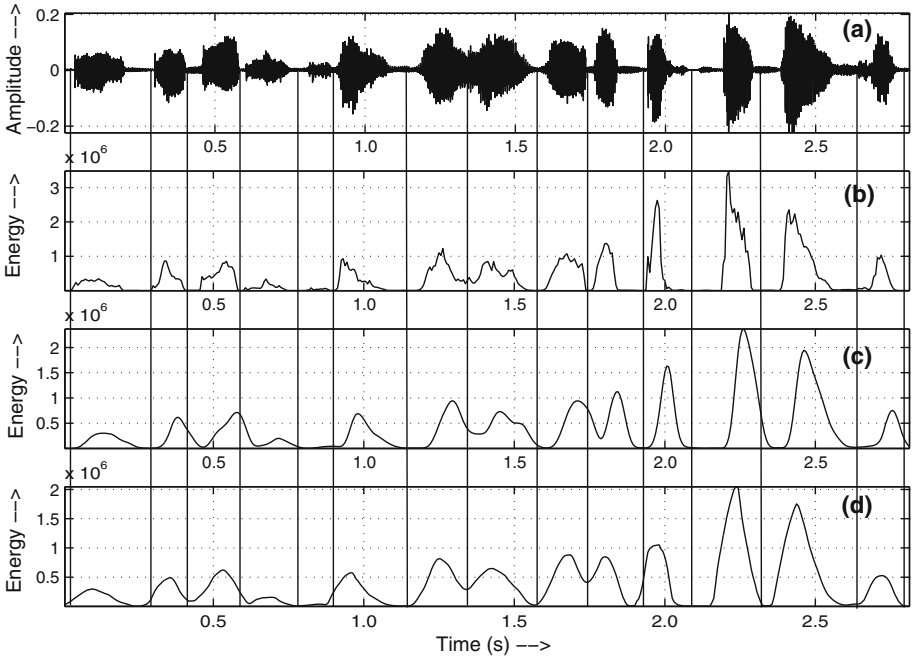

Figure 8. Short-term energy-based segmentation. (**a**) Speech signal. (**b**) Corresponding STE function. (**c**) Low-pass filtered STE function. (**d**) Mean-smoothed STE function.

the syllable duration analysis (Greenberg 1999) performed on the *switchboard corpus* (Godfrey *et al* 1992), shows that the duration of syllables mostly varies from 100 ms to 300 ms with a mean of 200 ms. In terms of modulation frequency, it varies from 3 Hz to 10 Hz, with a peak around of 5 Hz. Using this rationale, in Pfitzinger *et al* (1996), a low-pass filter with cut-off frequency of 10 Hz is applied on the logarithm of the STE amplitude to suppress the ripples caused by f_0 or transient consonants (figure 8c). This forces the system to fluctuate at syllable frequencies. The selection of cut-off frequency is crucial; it should be different for different speech rates. In this work, an attempt is made to overcome the above-mentioned issues. The STE function is a non-zero, positive function. The magnitude spectrum of any real signal satisfies the symmetry property, i.e.,

$$|X(e^{j\omega})| = |X(e^{-j\omega}|. \tag{41}$$

Therefore, techniques applied for processing the magnitude spectrum can be applied to the energy function. The IDFT of this function will be a two-sided signal (the real cepstrum). If the causal portion of this signal alone is considered, it is a perfect minimum phase signal, since it is derived from the magnitude spectrum alone (Nagarajan *et al* 2001). The smoothing of this assumed magnitude spectrum can be performed in one of several ways:

- *Cepstrum-based smoothing:* High frequency ripples can be removed by applying a lifter in the cepstral domain, thereby, retaining the low-frequency ripples alone (Noll 1967).
- *Cepstrum-LP-based smoothing:* By choosing a proper order, which is based on the number of syllables present in the speech signal, the cepstrum can be modelled.
- *Root cepstrum-based smoothing (Lim 1979):* Spectral root homomorphic deconvolution performance is similar to, or even better than the log homomorphic deconvolution. The root cepstrum corresponds to the IDFT of $|X(e^{j\omega})|^{\gamma}$. The factor γ in $|X(e^{j\omega})|^{\gamma}$ is chosen such that $0 < \gamma << 1$.

Earlier in section 3.2, we saw that minimum phase group delay functions are very useful in formant/anti-formant extraction (Murthy & Yegnanarayana 1991), and spectrum estimation (Yegnanarayana & Murthy 1992). Here, we extend the same idea to extract boundary information from the STE function.

4.2 *Algorithm for segmentation*

- Let $x[n]$ be the given continuous speech utterance (figure 9a).
- Compute the STE function $E[m]$, where m = 1,2, ..., M (figure 9b), using overlapped windows. Let the minimum value of the STE function be $E_{\min}$.
- Compute the order (N) of FFT, where

$$N = 2^{\lceil(\log(2M)/\log(2)\rceil}. \tag{42}$$

- Invert the function $E[m]^{\gamma}$ (say $E^i[m]$), (where $\gamma = 0.001$)[5] after appending $(\frac{N}{2} - M)$ number of $E_{\min}$ to the sequence $E[m]$ (figure 9c).
- Construct the symmetric part of the sequence by producing a lateral inversion of this sequence about the Y-axis. Let this sequence be $E^i[k]$ (figure 9d). Here, the sequence

[5]A small value of γ is chosen to reduce the dynamic range of the STE.

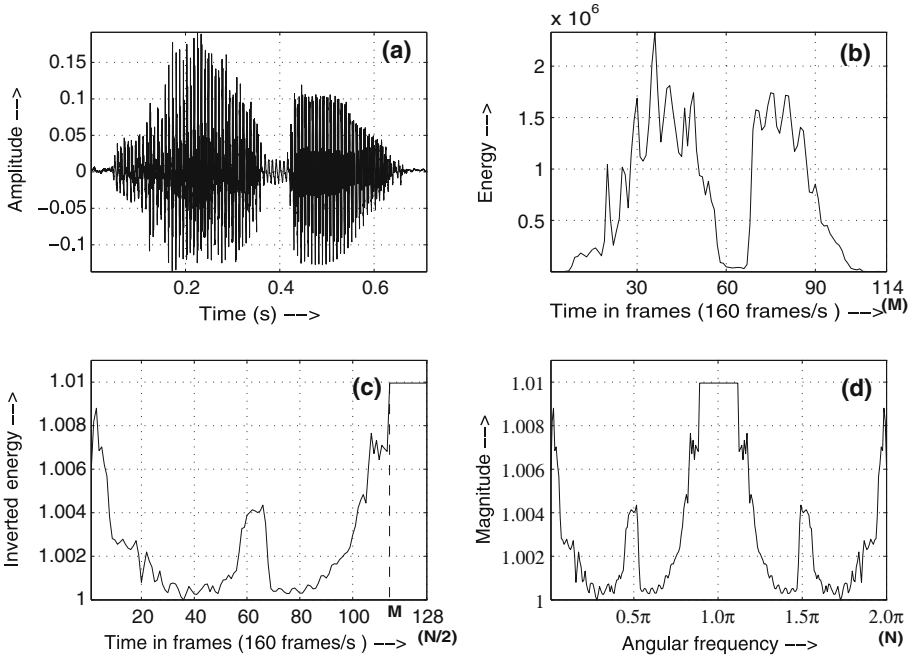

Figure 9. Segmentation using the STE (**a**) Speech signal, (**b**) corresponding STE function, (**c**) inverted STE function, (**d**) inverted and symmeterized STE function.

$E^i[k]$ is treated as the magnitude spectrum of some arbitrary signal. In this time-frequency substitution, N is replaced by 2π irrespective of the value of N.

– Compute the inverse DFT of the sequence $E^i[k]$. This resultant sequence $e[n']$ is the root cepstrum. The causal portion of $e[n']$ has the properties of a minimum phase signal. Let this signal be of length M.
– Compute the minimum phase group delay function of the windowed causal sequence of $e[n']$ (Murthy 1997; Murthy & Yegnanarayana 1991). Let this sequence be $E_{gd}(k)$. Let the size of the window applied on this causal sequence, i.e., the size of the cepstral lifter, be $N_c = \frac{M}{W}$. Where W is called window scale factor. This ensures that N_c is proportional to the length of the utterance.
– Detect the positive peaks in the minimum phase group delay function ($E^i_{gd}[k]$) as given below. If $E^i_{gd}[k]$ is positive, and if

$$E^i_{gd}[k-1] < E^i_{gd}[k] < E^i_{gd}[k+1], \tag{43}$$

then $E^i_{gd}[k]$ is considered as a peak. These peaks approximately correspond to the syllable boundaries.
– If the duration of a segment is much longer than the average duration of a syllable (then it is possibly a polysyllable) in the given database, resegment the given segment of speech using group delay again.

Figure 10a shows a Tamil (an Indian language) speech utterance. Figure 10b shows the group delay function obtained using the proposed method. The group delay functions derived using

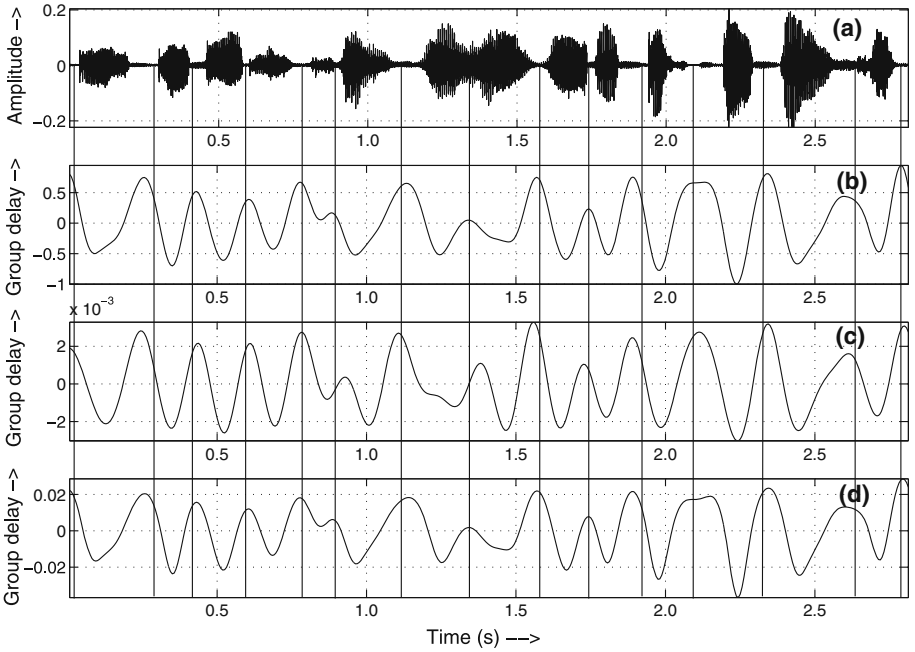

Figure 10. Group delay-based segmentation – an example. (**a**) Speech signal for the utterance '*dha indian airlinesai valiyaha bamba bambai sen*' (Tamil). (**b**) Group delay function derived from root-cepstrum. (**c**) Group delay function derived from cepstrum-LP. (**d**) Group delay function derived from conventional cepstrum.

the other two methods, i.e., cepstrum and cepstrum-LP based smoothing methods, are also given in figure 10c and d, along with the group delay function derived using root-cepstrum-based smoothing, for comparison. Interestingly, all the three group delay functions are almost similar, except for a slight shift in the boundary locations in the case of cepstrum-LP-based smoothing. However, each method has got its own advantages as well as disadvantages. If the cepstrum and root-cepstrum-based smoothing are considered for comparison, the group delay functions are exactly similar in shape. However, the computation of conventional cepstrum requires the log operation. The common problem with these two methods, is the choice of the cepstral lifter size (N_c). The choice of this parameter is crucial for the segmentation algorithm. If cepstrum-LP-based method is used, the cepstral lifter size is not crucial, and in fact the whole causal portion of the cepstrum can be considered for prediction. Even though, this seems to be very attractive, this methods suffers from the fact that the choice of the predictor order is related to the number of boundaries. In the next section, the segmentation algorithm is used to implicitly segment the speech signal into syllable-like units. The segmented units are then used to build unique syllable models for different languages. The property of these syllables is then used to perform language identification.

5. Group delay segmentation and language identification

Successful approaches to language identification (LID) use phone recognizers of several languages in parallel. The basic requirement to build parallel phone recognition (PPR) system is

a segmented and labeled speech corpus. Building segmented and labeled speech corpora for all the languages to be recognized, is both time consuming and expensive, requiring trained human annotators and substantial amount of supervision (Greenberg 1999). Even though the performance of the implicit LID systems[6] is slightly inferior to that of explicit LID systems[7], unavailability of segmented and labeled speech corpora makes implicit LID systems attractive.

One successful approach uses phone recognizers of several languages in parallel (Zissman 1996). This approach requires segmented and labeled speech corpora of more than one language, although it need not be available for all the languages to be identified. In Ramasubramanian *et al* (2003), a parallel sub-word recognition system for the LID task is proposed, in a framework similar to the PPR approach in the literature (Zissman 1996). The difference is that this approach does not require segmented and labelled speech corpora.

Using phonemes as the basic sound unit for LID task may not be optimal in the sense that most of the phonemes are common between languages. Only very few phonemes are unique for a particular language. If a longer sound unit, say syllable is used, then the number of unique syllables in any language can be very high, which may have potential information for discriminating languages. Li (1994) has shown that spectral features derived from syllabic units are reliable for distinguishing languages.

In the proposed work, a novel approach is proposed for the LID task which uses parallel syllable-like unit recognizers, in a framework similar to PPR approach in the literature with one significant difference. The difference is that unsupervised syllable models are built from the training data.

5.1 *Parallel syllable-like unit recognition*

The basic requirement for building syllable-like unit recognizers for all the languages to be identified, is an efficient segmentation algorithm. Earlier, an algorithm (Kamakshi Prasad *et al* 2004) was proposed, which segmented the speech signal into syllable-like units. Several refinements (Nagarajan *et al* 2003) are made to improve the segmentation performance of the baseline algorithm (Kamakshi Prasad *et al* 2004). Using this algorithm (Kamakshi Prasad *et al* 2004; Nagarajan *et al* 2003) each of the language training utterances is first segmented into syllable-like units[8]. Similar syllable segments are then grouped together and syllable models are trained incrementally. These language-dependent syllable models are then used for identifying the language of the unknown test utterances.

Using the modified segmentation algorithm, each 45s of the OGI_MLTS corpus (OGI 1992) is segmented into syllable-like units. The segmentation performance is quite satisfactory, except that occasionally the syllables are merged. A simple duration model is used to resegment a given syllable. Unlike speech synthesis or recognition, the syllable segmentation is not crucial, in that one may have bisyllables or trisyllables or demisyllables in a segment. The only requirement is that these segments are representative of the language. Once the segmentation is performed, an unsupervised HMM clustering of these segments is performed. Upon convergence, the clustering process results in a set of syllables for every language. The clusters are used in several ways for

[6] A system where no information about the language is used.
[7] A system where the phonotactics of the language is explicitly used.
[8] In the context of LID, accurate segmentation is not crucial.

performing language identification: (i) the accumulated acoustic log likelihood, (ii) a voting rule and (iii) unique syllables of a language. The methods (ii) and (iii) can be very useful clues. They are somewhat akin to what humans do when performing language identification, even if they do not know the language. We now briefly discuss the clustering process (figure 11).

5.1a *Initial cluster selection*: For any iterative training process, the assumed initial condition is crucial for the speed of convergence. After all the syllable segments have been obtained, the first task is to select some unique syllable segments or groups of unique syllable segments for training. The initial groups of syllable segments should be carefully chosen to ensure fast convergence. At the initial stage itself, if the selected group of syllable segments are unique, the convergence may be accelerated during iterative training. For selecting such initial clusters, the following procedure is adopted.

(i) From the $\mathcal{M}_l$ syllable segments of language $\mathcal{L}_l$, a subset ($\mathcal{N}1$) syllable segments, $\mathcal{S}_1, \mathcal{S}_2, ...\mathcal{S}_{\mathcal{N}1}$, where $\mathcal{N}1 < \mathcal{M}_l$, are taken for initialization.
(ii) Features (13-dimensional MFCC + 13 delta + 13 acceleration coefficients, after cepstral mean subtraction) are extracted from these $\mathcal{N}1$ syllable segments with multiple resolutions (i.e., with different window sizes and frame shifts). Multi-resolution feature extraction (MRFE) ensures a reasonable variance for each Gaussian mixture in the models. Details of this approach can be found in Lakshmi Sarada *et al* (2004).
(iii) $\mathcal{N}1$ HMMs ($\lambda_1, \lambda_2, ..., \lambda_{\mathcal{N}1}$) are initialized. To initialize model parameters, the Viterbi algorithm is used to find the most likely state sequence corresponding to each of the training examples, and then the HMM parameters are estimated. Here each of the feature vectors are derived from the same syllable segment but with different resolutions.

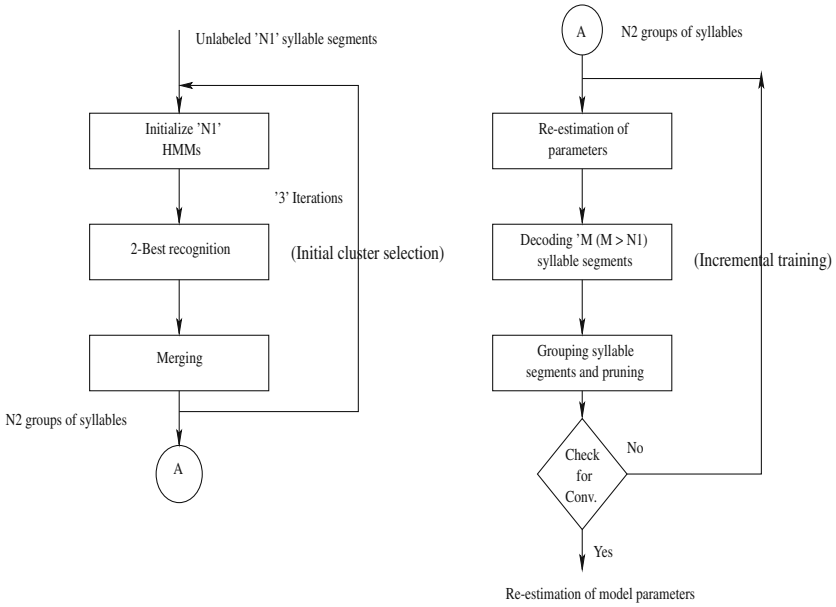

Figure 11. Flow chart: Unsupervised clustering and incremental training of HMMs.

(iv) The Viterbi decoding process is used to decode the $\mathcal{N}1$ syllable segments using two-best criteria, resulting in $\mathcal{N}1$ pairs of syllable segments (p_1, p_2, ..., $p_{\mathcal{N}1}$).

$$p_i = [\arg \max^{1}_{1 \leq i \leq \mathcal{N}1} \mathcal{P}(\mathcal{O}|\lambda_i), \quad \arg \max^{2}_{1 \leq i \leq \mathcal{N}1} \mathcal{P}(\mathcal{O}|\lambda_i)], \tag{44}$$

where

- p_i is the i^{th} pair of syllable segments (where $1 \leq i \leq \mathcal{N}_1$)
- $\mathcal{P}(\mathcal{O}|\lambda_i)$ is the probability of the observation sequence $\mathcal{O}$ ($o_1 o_2 ... o_n$) for the given model λ_i
- $\max^1$ and $\max^2$ denotes the 1-best and 2-best results, respectively.

This step gives $\mathcal{N}1$ pairs of syllable segments.

(v) Among $\mathcal{N}1$ pairs (p_1, p_2, ..., $p_{\mathcal{N}1}$), if a syllable segment is found to be repeated in more than one pair, the other pairs are removed, and the number of models is thus pruned.

(vi) New models are created with these reduced number of pairs.

(vii) Steps iv–vi are repeated m times (here, $m = 3$). After m iterations, each cluster will have 2^m syllable segments grouped together.

This initial cluster selection procedure will lead to $\mathcal{N}2$ clusters ($\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{\mathcal{N}2}$). In the next step, the model parameters are re-estimated incrementally.

5.1b *Incremental training*: After selecting the initial clusters ($\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{\mathcal{N}2}$), where the models are only initialized, the parameters of the models of each of the clusters is re-estimated using Baum–Welch re-estimation procedure. This training procedure is referred to as incremental training. This training strategy must be contrasted to conventional batch training, where the models are updated only after all the data in the training set are processed. The steps followed for this incremental training are given below.

(i) The model parameters of the initial clusters ($\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{\mathcal{N}2}$) derived from the previous step are re-estimated using Baum–Welch re-estimation. Each model is a 5 state 3 Gaussian mixtures/state HMM.

(ii) The new models are used to decode all the syllable segments ($\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{\mathcal{M}_l}$) using Viterbi decoding.

(iii) Clustering is done based on the decoded sequence.

(iv) If a particular cluster is found to have less than 3 syllable segments, that cluster is removed and number of models is reduced by one.

(v) Steps i–iii are repeated until convergence criterion is met.

The convergence criteria followed for the incremental training is based on 'number of migrations of syllables between clusters'. The convergence is said to be met if the number of migrations between clusters reaches zero. When this condition is met, the incremental training procedure terminates. This incremental training process produces $\mathcal{N}3$ syllable clusters ($\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{\mathcal{N}3}$), and in turn $\mathcal{N}3$ syllable models ($\lambda_1, \lambda_2, ..., \lambda_{\mathcal{N}3}$).

A set of labels are finally assigned for each of the clusters. During testing, the speech signal is segmented and tested against the syllable models for each language. We now use the three

Table 1. Language-wise performance of the LID systems using AAL, the system using voting, the system using US alone, and the system using AAL and US segments.

Method	1-best performance in %											
	En	Fa	Fr	Ge	Hi	Ja	Ko	Ma	Sp	Ta	Vi	Average
AAL	95	90	95	80	70	65	45	35	80	65	75	**72.27**
Voting	80	80	90	60	55	65	40	25	65	70	60	**62.7**
US	80	65	85	65	80	55	55	40	70	55	60	**64.5**
AAL + US	80	80	90	85	85	90	60	40	85	65	80	**75.9**

different ways of determining the identity of a language as mentioned earlier, namely, (i) acoustic likelihood for language (AAL), (ii) voting, and (iii) by cross-recognition[9] of syllable segments across languages to determine unique syllables (US) for every language.

The 1-best performance of the LID using these methods is given in table 1. From the table, based on the performance, it can be conjectured that the clustering process does yield the syllables that make up a language. The voting rule primarily helps in removing the noise in the AAL approach. The unique syllables method is perhaps a very important clue. For example, when identifying an Indian language, the occurence of the sound /*zha*/ reduces the search to two languages, namely, Tamil and Malayalam.

6. Application of group delay functions in concatenative speech synthesis

Concatenative speech synthesis using unit selection relies on a large database of basic units. The quality of a speech synthesis system is dependent upon the quality of the basic units. For naturalness, basic units are collected from continuous speech of a single speaker. The basic units are collected from different contexts. The idea behind unit selection synthesis is to select the best sequence of speech units from all possible contexts from a database of speech units. Accurate labeling of units in continuous speech is thus a requirement. Most systems today use ergodic HMMs (e-HMMs) (Black *et al* 1998) to label the units automatically. e-HMMs label data accurately, provided they are trained with large amounts of data. The algorithm developed in section 4 for segmenting the speech signal at syllable boundaries is used to accurately segment the speech at syllable boundaries. This is particularly well-suited for Indian languages, which are syllable-centred[10].

The algorithm for segmentation developed in section 4 is modified using vowel onset point (Prasanna *et al* 2009) detection to reduce insertion and deletion errors. This is then used effectively for generating labels semiautomatically.

A labelling tool is developed around this idea. This tool has been tested for six different Indian languages, namely, Tamil, Hindi, Bengali, Malayalam, Telugu and Marathi. The labelling tool generates labels that are consistent with that of the Festival speech synthesis system (Black *et al* 1998). It was observed that the labels generated by this system were more systematic compared to the labels generated manually or using e-HMMs.

---

[9]Here by cross-recognition, we mean the following. Consider two languages *A* and *B*. During training, the syllable segments of *A* are tested against the syllable models of *B* and vice-versa.

[10]It has been well established in the literature (Kishore & Black 2003; Rao *et al* 2005) that syllables as the basic units are appropriate for Indian languages.

In the next section, we discuss a syllable-based recognizer that utilizes the above algorithm to segment speech into syllable-like units. In contrast to the conventional speech recognizer, boundary information is incorporated into the recognition phase as well to reduce the search space.

7. Application of group delay functions in speech recognition

In this section, the group delay function is used in speech recognition in two different ways. First, the segmentation algorithm developed earlier is used to segment the speech signal at syllable boundaries. These boundaries are then incorporated in the linguistic framework to reduce the WER. Next, the modified group delay function developed in section 3.3, is converted to a feature (similar to MFCC) that can be used in speech recognition.

7.1 *Speech recognition using segmentation information*

In this section, we exploit the segmentation information in two different ways. A conventional automatic speech recognition (ASR) does not use the boundary information available in the data. In this section, we propose a new paradigm in speech recognition, where we incorporate the segmentation information into the training and testing phases of the recognizer. This recognizer is referred to as the automatically annotated recognition (AAR).

– Training the AAR system: It is assumed that both text and waveforms are available during training. The text is segmented using a rule-based segmentation algorithm into syllables (Lakshmi & Murthy 2008). The waveform is segmented using the segmentation process described in section 4 with parameters tuned appropriately to generate the same number of syllables segments as in the text. The time-aligned waveforms and text are used to build an isolated style recognizer. Using the text segments as labels for the corresponding waveform segments, all the syllable models are trained. If a certain syllable did not occur often enough, resulting in a poorly trained model, the multiple frame size (MFS) and multiple frame rate (MFR) method (Lakshmi Sarada *et al* 2004) was used to improve its reliability. Figure 12 shows the AAR system (Lakshmi & Murthy 2008). The syllable models were built using embedded training mechanism followed by token passing model for state alignment. The Hidden Markov Model (HTK) toolkit was used for this purpose (CUED 2002). 39-dimensional MFCC features were used for training the system.

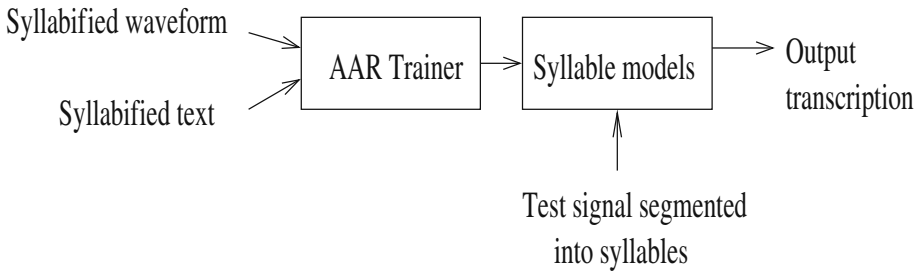

Figure 12. AAR system.

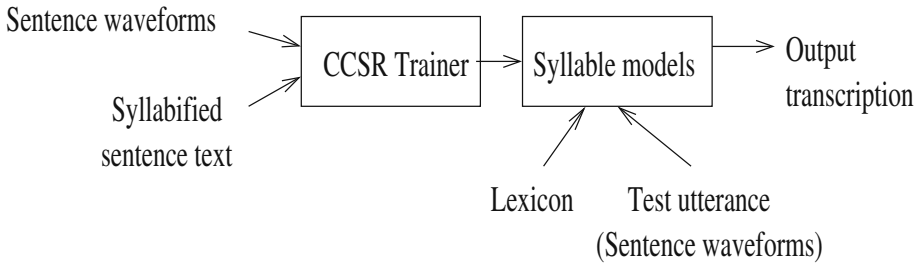

Figure 13. CCSR-based recognition system.

– Testing the AAR system: During testing the segmentation, information is used to segment the speech into syllable-like units. The syllables are then recognized using isolated style syllable recognition.

This is in sharp contrast to that of conventional speech recognition system (CCSR). A brief description of a CCSR system follows.

– *Training the CCSR system:* A conventional recognizer, which uses flat-start. In a flat-start recognition system, the waveform is divided uniformly into a number of segments as dictated by the text. The system is iteratively trained using embedded training with state alignment to correct the labels. The sentence waveforms and syllabified sentence level transcriptions are used to train the system.
– *Testing the CCSR system:* During testing, the CCSR system uses a lexicon and language models (figure 13) for generating the recognition output. To make this system comparable to the AAR engine, a dummy lexicon was used and language models were dispensed with. The system outputs a syllable sequence similar to that of the AAR system.

Table 2 shows the results obtained on Tamil language (from the DBIL (DDNews 2001) database) and TIMIT (TIMIT 1990). Since TIMIT is labeled at the phoneme level, the manually marked boundaries were used for both training and testing[11]. From the table, it is clear that the AAR system outperforms the CCSR system. This is because the CCSR system does not use the syllable boundary information utilized by the AAR system during training. Further, the amount of data available for training is rather small for the HMMs to learn the acoustic boundaries accurately.

The advantage of this approach is that we only require the vocabulary of syllables that make up the language. A language model is not required. The drawback of this approach is that the number of syllables in a language can be large. This slows down the testing process. In the next section, we incorporate the segmentation information into the linguistic search space in a conventional speech recognizer that uses language models.

7.2 *Incorporating segmentation information in the linguistic search space*

Conventional recognizers use the language information to derive the word output from the recognizer. The language models are specified as a grammar or $\mathcal{N}$-gram language models.

Language models can be generated from: (i) training data with smoothing incorporated to account for words that do not occur in the training data but only occur in the testing data or (ii)

[11] This is referred to as TIMIT(M) in the table.

Table 2. Syllable recognition accuracy using AAR and CCSR.

System	Tamil	TIMIT (M)
AAR	30%	78.04%
CCSR	20.06%	68.5%

generated from huge text corpora (a few hundred million words) obtained from various domains. During testing, the language model generates N-best hypotheses for the utterance spoken using unigram, bigram and possibly trigram probabilities[12]. The acoustic model is used in tandem with the language model to determine the probability that the spoken utterance corresponds to a specific word sequence.

The maximum *a posteriori* probability approach to speech recognition uses the Bayes' rule (Jelinek 1999). A word sequence W produces an acoustic observation sequence Y, with joint probability $p(W, Y)$. During recognition, the estimated word sequence $\hat{W}$ is given by

$$\hat{W} = \arg\max_{W} p(W|Y) = \arg\max_{W} p(Y|W)P(W) \tag{45}$$

$p(Y|W)$ is generally called the acoustic model. The second term $P(W)$ is called the language model, as it describes the probability associated with a postulated sequence of words. The statistical language model estimates the probability of occurrence of some word w_k given its history $W_1^{k-1} = w_1 \ldots w_{k-1}$ from text corpora.

In this work, the above mentioned traditional language modelling framework (TLM) is incorporated into the syllable-recognition framework, but with a difference. The acoustic segmentation information obtained from the group delay-based segmentation algorithm is incorporated into the linguistic search space. In the conventional system, the language model is accessed at every frame. In the proposed system, the language model is only accessed at syllable boundaries. In the proposed approach, the search space can therefore be significantly less. To see this, consider the example sentence from the TIMIT test corpus: 'Salvation reconsidered[13].' Figure 14 shows the frame number along the X-axis and the number of active states at each frame along the Y-axis. The graph shows two curves, one corresponding to that of a TLM-based system and another corresponding to that of the proposed system. In this example, the average number of active states for the TLM system/frame is $\approx$ 15000, while that for the proposed system is about $\approx$ 6000. Observe that there is significant difference in the number of active states. This is primarily because the language model is accessed only at segment boundaries in the proposed system. This can be observed in figure 14. Observe that in regions corresponding to the duration of a syllable, the number of active states is constant (as indicated by dotted vertical lines). This experiment was repeated on the entire TIMIT test corpus consisting of 198 sentences. The average number of active states for the TLM was found to be 15875.0, and 5683.6 for the proposed system, respectively.

Table 3 compares the results of the modified language model-based recognizer and conventional recognizer. Clearly, the use of segmentation does significantly improve the performance. The word recognition performance for TIMIT and NTIMIT using AAR and CCSR are also presented in table 3. (N)TIMIT(M) corresponds to the results obtained using the manual segmentation information, while (N)TIMIT(A) corresponds to that of the automatically segmented

[12]Use of trigram requires huge amounts of text corpora.

[13]We have used the manually marked boundaries from the database for this.

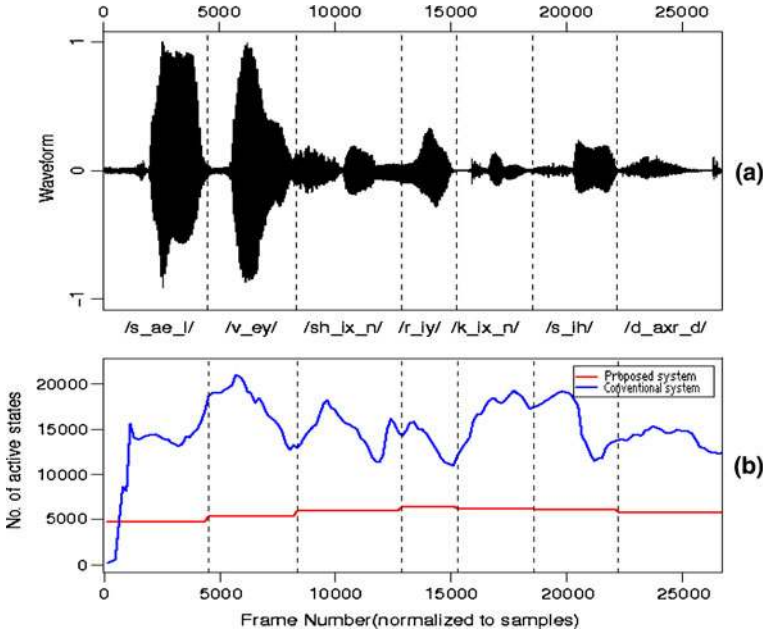

Figure 14. Number of active states as a function of frame number.

data[14]. It can be seen from the tables that for both syllable and word recognition, the AAR outperforms CCSR. The table also mentions the WERs corresponding to a CCSR system, in which the syllable models were initialized from the corresponding triphone models (Sethi & Narayanan 2003). It is interesting to note that the automatic segmentation actually improves performance on TIMIT and NTIMIT databases. We conjecture that this must be due to errors in manual segmentation. The importance of automatic segmentation is thus vindicated by this example. Although, HMMs and language models do postpone the decision to the very end, clearly acoustic cues can be used in tandem with language models to improve performance.

7.3 *New features for speech recognition*

In this section, we derive new features similar to MFCC, from the Fourier transform phase function. These features are then used in the linguistic search space. We first convert the 'modified group delay function' into a feature. This feature in then used as a feature for recognition. We conjecture that different features recognize different sounds better. We show how this idea can be incorporated in the linguistic framework, to perform what we term as 'feature switching'.

7.3a *Parameterising the modified group delay function*: Since the modified group delay function exhibits a squared magnitude behaviour at the location of the roots, the modified group delay function is referred to as the modified group delay spectrum. Homomorphic processing is the most commonly used approach to convert spectra derived from the speech signal to meaningful

[14]For the TIMIT and NTIMIT databases, various modifications were made to the segmentation algorithm to compensate for variations in the syllable-rate. For details, see Janakiram *et al* (2010).

Table 3. WER for AAR and CCSR systems.

System	Tamil	TIMIT (M)	TIMIT (A)	NTIMIT (M)	NTIMIT (A)
AAR	39.2%	6.5%	4.4%	29.4%	21.2%
CCSR (triphone)	42.17%	-	13%	-	36%

features. This is primarily because this approach yields features that are linearly decorrelated, which allows the use of diagonal covariances in modelling the speech vector distribution. In this context, the discrete cosine transform (DCT Forms II and III) (Yip & Rao 1997), are the most commonly used transformations that can be used to convert the modified group delay spectra to cepstral features. Hence, the group delay function is converted to cepstra using the DCT II as

$$c[n] = \sum_{k=0}^{N_f} \tau_c(k) \cos(n(2k+1)\pi/N_f), \tag{46}$$

where N_f is the DFT order, and $\tau_c(k)$ is the modified group delay spectrum. The DCT can also be used in the reconstruction of the modified group delay spectra from the modified group delay cepstra. Velocity and acceleration parameters for the new group delay function are defined in the cepstral domain, in a manner similar to the velocity and acceleration parameters for MFCC.

7.3b *Feature switching: Kullback–Liebler divergence*: A wide variety of acoustic features have been used in speech recognition such as MFCC (Davis & Mermelstein 1980), Linear prediction cepstral coefficients, perceptual linear prediction coefficients (Hermansky 1990) and modified group delay feature (Murthy & Rao 2003). It has been observed that different feature streams capture different characteristics of a sound. To capitalize on the diverse characteristics captured by different features, there are three different ways in which feature streams are combined.

- *Feature combination:* This is also referred to as early integration (fusion), where different feature streams are concatenated into a new single feature stream. This is sometimes followed by a dimensionality reduction technique (Halberstadt & Glass 1998; Neti *et al* 2001).
- *Likelihood score combination:* This is also referred to as middle integration, where decoding is performed using individual feature streams for each sub-word model. This generates multiple likelihood score for each sub-word model. The likelihood scores of a given sub-word are combined across all the feature streams by weighting each feature stream based on the reliability of the feature stream (Kumar & Murthy 2009; Rasipuram *et al* 2008; Halberstadt 1998; Dupont & Luettin 2000). Measure of reliability is estimated directly from training data.
- *Hypotheses fusion:* This is also referred to as late integration. Here, complete recognition hypotheses are first generated in parallel from each individual recognition system, which are then combined together to generate the final combined hypothesis (Li & Stern 2003).

Of the three types of feature stream combinations, middle integration is the most successful approach in the literature. There exist several algorithms to calculate the reliability of a feature from training data (Kumar & Murthy 2009; Rasipuram *et al* 2008; Gurban & Thiran 2008). According to Kumar & Murthy (2009), Rasipuram *et al* (2008) and Gurban & Thiran (2008) more reliable the feature stream, greater is the weight associated it. However, although middle

Table 4. WERs on the TIMIT, NTIMIT and DBIL (Tamil) for each type of system (D – delta, A – Acceleration, E – Energy).

Features	TIMIT %WER	NTIMIT %WER	DBIL %WER
MFCC+D+A+E	6.5	29.4	35.7
MODGD+D+A+E	10.1	31.6	30.4
MFCC+MODGD+D+A+E	4.3	26.8	23.5
Feature switching between MFCC and MODGD using Bhattacharyya divergence	2.86	22.1	18.4
Feature switching between MFCC and MODGD using KL divergence	2.95	23.4	19.7

integration gives better recognition accuracy, it becomes computationally inefficient as the number of feature streams increases. The use of m different feature streams increases computations roughly by a factor of m in comparison with that of a single feature stream.

Current speech recognizers depend heavily on language models. The language model hypotheses are evaluated using the acoustic models. In this scenario, it is quite possible that the language model generates a set of confusable hypotheses. In such a situation, a feature that discriminates between a set of confusable hypotheses is all that may be required to arrive at the correct hypothesis.

To address this issue, Kullback–Leibler divergence or Bhattacharya distance between every pair of sub-word HMMs for each feature stream is computed during the training process. This measure is utilized during the recognition phase to prune the search space and to switch features. Details of the approach can be found in Kumar *et al* (2010). We now perform a controlled experiment to test feature switching as a paradigm. In this experiment, manually marked segment boundaries are used in all the three databases. In table 4, we present the results of using multiple features in different ways. From table 4, it is clear that feature switching indeed helps improve performance. Clearly, when a feature is irrelevant for recognising a sound, use of joint features is equivalent to adding noise.

8. Application of group delay functions for speaker recognition

In an earlier section, we saw the use of feature switching for recognising different sounds. In this section, we show how feature switching can be used in a speaker verification framework.

For classification tasks, one must consider two aspects of feature representation: (i) the ability to capture maximum information from the acoustic space into the feature space (representative property), and (ii) the ability to discriminate between different classes (discriminative property) (Padmanabhan & Murthy 2010).

8.1 *Using mutual information to measure representative property*

Mutual information (MI) is used to quantify the amount of information captured from the acoustic space to the feature space. Two features are considered, namely, MFCC and MODGDF. The MI between speech signal and each of the features is a measure of the information captured by the feature. We thus measure the MI between the complex short-time Fourier spectrum (which represents the signal in the acoustic space) and the individual feature streams (representing the signal in the respective feature space.)

8.2 *Using KL-divergence to measure discriminative property*

A feature that efficiently captures information need not be efficient in discriminating between them. The discriminative property of a feature representation is a measure of the inter-class separability. The Kulback–Leibler divergence or relative entropy is a measure of the distance between two probability distributions. For two Gaussians $\hat{f}$ and $\hat{g}$, the KL-divergence has the closed form expression

$$\mathrm{kld}(\hat{f}, \hat{g}) = \frac{1}{2}\left[\log\frac{|\Sigma_{\mathbf{g}}|}{|\Sigma_{\mathbf{f}}|} + \mathrm{Tr}|\Sigma_g^{-1}\Sigma_{\mathbf{f}}| - d + (\mu_{\mathbf{f}} - \mu_{\mathbf{g}})^T \Sigma_{\mathbf{g}}{}^{-1}(\mu_{\mathbf{f}} - \mu_{\mathbf{g}})\right] \tag{47}$$

with $\hat{f} = \mathcal{N}(\mu_{\mathbf{f}}, \Sigma_{\mathbf{f}})$ and $\hat{g} = \mathcal{N}(\mu_{\mathbf{g}}, \Sigma_{\mathbf{g}})$. Here $\mu_{\mathbf{f}}$, and $\Sigma_{\mathbf{f}}$ correspond to the mean and variance of $\hat{f}$ and $\mu_{\mathbf{f}}$ and $\Sigma_{\mathbf{f}}$ correspond to the mean and variance of $\hat{g}$.

For Gaussian mixture models (GMMs), the KL-divergence has no closed form expression. Many contemporary speaker verification systems build Gaussian mixture speaker models by adaptation of a universal background model (UBM). For adapted speaker models, there is a one-to-one correspondence between the component mixtures of the speaker model and the UBM. The KL-divergence between the speaker model and the background model is a measure of the discriminability between the target speaker and imposter. The feature representation that gives higher KL-divergence better separates the speaker model from imposters.

8.3 *Application of feature diversity to speaker verification*

Preliminary experiments reveal that different features are effective in discriminating different speakers. We now apply the above conjecture to speaker verification.

A feature representation that efficiently represents a given speaker, as well as discriminates against other speakers, is termed an optimal feature for that speaker. The optimal feature for a speaker can be determined (from a list of candidate features) at enrolment time or by using development data.

In a speaker verification system, a claimed identity is given along with the test utterance and the score computed, only for the claimed model. The optimal feature of the claimed speaker can be used, as this will result in better modelling and discrimination using the speaker GMM. This results in using different features for different claims or feature-switching.

Assume that there are N features indexed by $i = 1, .., N$. The optimal feature of a given speaker can be determined from the training/development data of the speaker. For a given speaker, we define the representative function θ_i and the discriminative function γ_i for feature representation i:

$$\theta_i = \mathrm{MI}(\mathcal{X}, \mathcal{Y}_i) \tag{48}$$

$$\gamma_i = \mathrm{kld}(\lambda_{\mathrm{spk},i}, \lambda_{\mathrm{UBM},i}), \tag{49}$$

where $i \in \{\mathrm{MFCC}, \mathrm{MODGDF}\}$, $\mathcal{X}$ represents the complex Fourier spectrum, $\mathcal{Y}_i$ represents the i^{th} feature representation, $\lambda_{\mathrm{spk},i}$ is the speaker model and $\lambda_{\mathrm{UBM},i}$ is the background model, for the i^{th} feature representation. The optimal feature function ϕ_i is defined as a linear combination of θ_i and γ_i (a line search on α is performed):

$$\phi_i = \alpha\theta_i + (1 - \alpha)\gamma_i, \tag{50}$$

where the weighting factor α is used to emphasize the representative or discriminative measure.

The optimum feature stream $\hat{i}$ is selected as

$$\hat{i} = \arg\max_{i}\{\phi_i\}. \tag{51}$$

8.4 *Speaker verification framework*

In the training phase, the optimal feature is determined for each speaker using the optimal feature function (Eq. 51). The ⟨speaker, optimal feature⟩ pair is stored in a lookup table (LUT), which is indexed by speaker identity. The LUT contains an entry for each of the registered speakers in the system. Different parameters of (Eq. 51) result in different LUTs for the same set of registered speakers.

In the evaluation phase, the optimal feature of the claimed speaker is determined from the lookup table. The optimal features are extracted from the input speech waveform. The TNorm score (Auckentaler *et al* 2000) is computed against the corresponding models and the verification decision is made. This results in the verification system performing feature switching, by extracting different features for different claims.

We evaluate the proposed speaker verification system using feature switching and compare the performance to conventional systems that use only a single feature representation, as well as joint feature representation (early fusion).

The database used in this study is the one-speaker detection task of the NIST 2003 speaker recognition evaluation (NIST 2003). There are 149 male speakers and 207 female speakers, with about 2 minutes of training data for each speaker. Each test utterance is about 30 s long. More details about the database can be found in NIST (2003).

The proposed speaker verification system incorporating feature switching is developed as follows. The optimal feature for each speaker is determined by using Eq. 51. The parameters of feature extraction for the optimal feature is the same as that of the respective baseline system. The weighting factor α represents a trade-off between representative features and discriminative features. Speaker-dependent values of α were determined empirically and were used to determine the optimal feature.

From table 5, we observe that the baseline MODGDF system shows better verification performance (equal error rate (EER) of 11.92%) than that of the baseline MFCC system (EER of 13.58%). This indicates the usefulness of phase-based features, as described in Padmanabhan *et al* (2009). Early fusion of feature representations generally improves the verification performance. From table 5, although the performance improves over that of the baseline MFCC, the performance of joint features is worse than that of MODGDF. Clearly, in this case, the use of

Table 5. Equal error rates of various speaker verification systems.

System	EER (%)
Baseline	
Baseline MFCC	13.58
Baseline MODGDF	11.92
Joint (early fusion)	
MFCC-MODGDF	13.21
Feature-switching	
MFCC/MODGDF	11.63

joint features is actually hurting performance. Various feature-switched systems in general, give performance better than the joint systems (Padmanabhan & Murthy 2010). Feature switching between MFCC, and MODGDF gives an EER of 11.63%. These experiments once again reinforce the importance of feature switching in speech recognition systems. For more details on feature switching across a suite of features, see Padmanabhan & Murthy (2010).

9. Conclusions

In this paper, we detailed the evolution of phase-based processing of speech. Initially, we showed how group delay spectrum can be usefully processed to extract formants. Next, we show that phase information can be used to identify events in a speech signal.

Finally, we extract features from phase, similar to MFCC. Although, phase-based processing is nascent, nevertheless, we have been able to show that the phase-based features perform almost as well as the current state-of-the-art MFCC-based features. We also show that by judicious choice of feature appropriate for the sound unit/speaker, performance of the system can be improved quite significantly. In conclusion, phase-based features do a lot of promise, and need to be explored in greater detail. Other applications of segmentation include the development of segment vocoders (Chevireddy *et al* 2008a,b; Pradhan *et al* 2010).

One of the authors, Hema A Murthy would like to thank all her students, V Kamakshi Prasad, T Nagarajan, Rajesh M Hegde, A Lakshmi, Samuel Thomas, R Ramya, P Sreehari Krishnan, Sadhana Chevireddy, Rajesh Janakiraman, J Chaitanya Kumar, R Padmanabhan and Abhijit Pradhan for not only collaborating with her, but also, for permitting her to reproduce figures and proofs from their theses in this paper. Further, she would like to thank Veena T, Dileep A D and R Padmanabhan for reading through drafts of this paper and giving critical comments. Hema A Murthy would also like to thank her PhD guide Prof Yegnanarayana (co-author) for introducing her to the world of group delay functions. Finally, we thank the anonymous reviewers for their useful comments.

References

Aarabi P, Shi G, Shanechi M M, Rabi S A 2006 *Phase based processing speech* (Singapore: World Scientific Publishing Co. Pte. Ltd.)

Alsteris L D, Paliwal K K 2006 Further intelligibility results from human listening tests using the short-time phase spectrum. *Speech Commun.* 48: 727–736

Auckentaler R, Carey M, Lloyd-Thomas H 2000 Score normalisation for text-independent speaker verification systems. *Digital Signal Process.* 10: 42–54

Black A, Taylor P, Caley R 1998 The festival speech synthesis system. http://festvox.org/festival/

Bozkurt B, Couvreur L, Dutoit T 2007 Chirp group delay analysis of speech signals. *Speech Commun.* 49(3): 159–176

Chevireddy S, Murthy H A, Chandrasekhar C 2008a A syllable-based segment vocoder. *Proc. National Conference on Communications*, Mumbai, India, 442–445

Chevireddy S, Murthy H A, Chandrasekhar C 2008b Signal processing based segmentation and hmm based automatic clustering for a syllable based segment vocoder at 1.4kbps. *Proc. EUSIPCO*, Lausanne, Switzerland. www.eurasip.org/Proceedings/Eusipco2008/papers/1569104947.pdf

Childers D G 1977 The cepstrum: A guide to processing. *Proc. IEEE* 68: 1428–1443

CUED 2002 HTK Speech Recognition Toolkit. http://htk.eng.cam.ac.uk

Davis S, Mermelstein 1980 Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust. Speech, Signal Process* 28: 357–366

DDNews 2001 *Database for Indian languages*. India, Speech and Vision Lab, IIT Madras, Chennai

Dupont S, Luettin J 2000 Audio-visual speech modeling for continuous speech recognition. *IEEE Trans. Multimedia* 2(3): 141–151

Godfrey J J, Holliman E C, McDaniel J 1992 SWITCHBOARD: Telephone speech corpus for research and development. *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, San Francisco, California, USA, 1. 517–520

Greenberg S 1999 Speaking in short hand - A syllable centric perspective for understanding pronounciation variation. *Speech Commun*. 29: 159–176

Greenberg S, Hollenback J, Ellis D 1996 Insights into spoken language gleaned from phonetic transcription of the switchboard corpus. *Proc. Int. Conf. Spoken Language Process*, Philadelphia, USA, 24–27

Gurban M, Thiran J-P 2008 Using entropy as a stream reliability estimate for audio-visual speech recognition. *Proc. EUSIPCO,* Lausanne, Switzerland. http://www.eurasip.org/Proceedings/Eusipco/Eusipco2008/papers/1569104998.pdf

Halberstadt A K, Glass J R 1998 Heterogeneous acoustic measurements and multiple classifiers for speech recognition. *Proc. Int. Conf. Spoken Language Process*. Sydney, Australia, paper 0396

Halberstadt A K 1998 Heterogeneous acoustic measurements and multiple classifiers for speech recognition. Ph.D. thesis, Massachussets Institute of Technology

Hermansky H 1990 Perceptually linear predictive (plp) analysis of speech. *J. of the Acoust. Soc. of Am.* 87: 1738–1752

Hirsch H, Pearce D 2000 The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. *Proc. ISCA Tutorial and Research Workshop on Automatic Speech Recognition*, Paris, France, 181–188

Janakiram R, Kumar C J, Murthy H A 2010 Robust syllable segmentation its application to syllable-centric continuous speech recognition. *Proc. National Conference on Communications*, Chennai, India, 276–280

Jelinek F 1999 *Statistical methods for speech recognition* (Cambridge, Massachusetts: The MIT Press)

Kamakshi Prasad V, Nagarajan T, Murthy H A 2004 Automatic segmentation of continuous speech using minimum phase group delay functions. *Speech Commun.* 42: 429–446

Kishore S P, Black A W 2003 Unit size in unit selection speech synthesis. *Proc. EUROSPEECH*, Geneva, Switzerland, 1317–1320

Kumar C J, Murthy H A 2009 Entropy based measures for incorporating feature stream diversity in the linguistic search space for syllable based automatic annotated recognizer. *Proc. National Conference on Communication*, Guwahati, India, 286–289

Kumar J C, Janakiraman R, Murthy H A 2010 Kl divergence based feature switching in the linguistic search space for automatic speech recognition. *Proc. National Conference on Communication*, Chennai, India, 281–285

Lakshmi Sarada G, Nagarajan T, Murthy H A 2004 Multiple frame size and multiple frame rate feature extraction for speech recognition. *Proc. SPCOM*, Bangalore, India, 592–595

Lakshmi A, Murthy H A 2008 A new approach to continuous speech recognition in indian languages. *Proc. National Conference on Communication*, Mumbai, India, 277–281

Li K 1994 Automatic language identification using syllabic spectral features. *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, Adelaide, South Australia, 1. 297–300

Li X, Stern R 2003 Training of stream weights for the decoding of speech using parallel feature streams. *Proc. IEEE Int. Conf. Acoust. Speech Signal Process,* 1: 832–835

Lim J 1979 Spectral root homomorphic deconvolution system. *IEEE Trans. Acoust. Speech Signal Process* 27: 223–233

Murthy H A 1997 The real root cepstrum and its applications to speech processing. *Proc. National Conference on Communication*, Chennai, India, 180–183

Murthy H A, Rao G V R 2003 The modified group delay function and its application to phoneme recognition. *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, Hongkong, 1.68–71

Murthy H A, Yegnanarayana B 1991 Formant extraction from minimum phase group delay function. *Speech Commun.* 10: 209–221

Murthy K V M, Yegnanarayana B 1989 Effectiveness of representation of signals through group delay functions. *Elsevier Signal Process.* 17: 141–150

Nagarajan T, Murthy H A, Hegde R M 2003 Segmentation of speech into syllable-like units. *Proc. EUROSPEECH*, Geneva, Switzerland, 2893–2896

Nagarajan T, Prasad V K, Murthy H A 2001 The minimum phase signal derived from the magnitude spectrum and its applications to speech segmentation. *Proc. SPCOM*, Bangalore, India, 95–101

Neti C P, Luettin G, Matthews J, Vergyri J H G 2001 Large-vocabulary audio-visual speech recognition: A summary of the johns hopkins summer 2000 workshop. *Proc. IEEE Fourth Workshop on Multimedia Signal Processing*, Cannes, France, 619–624

NIST 2003 The NIST year 2003 speaker recognition evaluation plan. http://www.itl.nist.gov/iad/mig/tests/sre/2003/index.html

Noll A M 1967 Cepstrum pitch determination. *J. Acoust. Soc. Am.* 41(2): 179–195

OGI 1992 The OGI multi-language telephone speech corpus. *Proc. Int. Conf. Spoken Lang.*, Banff, Alberta

Oppenheim A V, Schafer R W 1990 *Discrete time signal processing* (New Jersey: Prentice Hall, Inc.)

Padmanabhan R, Murthy H A 2010 Acoustic feature diversity and speaker verification. *Proc. INTERSPEECH*, Makuhari, Japan, 2110–2113

Padmanabhan R, Parthasarthi S H K, Murthy H A 2009 Robustness of phase based features for speaker recognition. *Proc. INTERSPEECH*, Brighton, U.K., 2355–2358

Paliwal K K, Alsteris L D 2005 On the usefulness of stft phase spectrum in human listening tests. *Speech Commun.* 45 153–170

Papoulis A 1977 *Signal analysis* (New York: McGraw Hill)

Pfitzinger H R, Burger S, Heid S 1996 Syllable detection in read and spontaneous speech. *Proc. Int. Conf. Spoken Language Process.*, Philadelphia, USA, 1261–1264

Pradhan A, Chevireddy S, Veezhinathan K, Murthy H A 2010 A low-bit rate segment vocoder using minimum residual energy criteria. *Proc. National Conference on Communication*, Chennai, India, 246–250

Prasanna S, Reddy S B, Krishnamoorthy P 2009 Vowel onset point detection using source, spectral peaks and modulation spectrum energies. *IEEE Trans. Audio Speech Language Process.* 17(4): 556–565

Rabiner L R, Schafer R W 1969 The chirp z-transform algorithm and its application. *Bell Syst. Tech. J.* 48(5): 1249–1292

Ramasubramanian V, Jayaram A K V S, Sreenivas T V 2003 Language identification using parallel sub-word recognition — an ergodic hmm equivalence. *Proc. EUROSPEECH*, Geneva, Switzerland, 1357–1360

Rao M N, Thomas S, Nagarajan T, Murthy H A 2005 Text-to-speech synthesis using syllable-like units. *Proc. National Conference on Communications*, Kharagpur, India, 227–280

Rasipuram R, Hegde R M, Murthy H A 2008 Incorporating acoustic diversity into the linguistic feature space for syllable recognition. *Proc. EUSIPCO 2008*, Lausanne, Switzerland, www.eurasip.org/Proceedings/Eusipco/papers/1569104561.pdf

Sethi A, Narayanan S 2003 Split-lexicon based hierarchial recognition of speech using syllable and word level acoustic units. *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, Hong Kong, 185–187

Shi G, Shanechi M, Aarabi P 2006 On the importance of phase in human speech recognition. *IEEE Trans. on Audio Speech Language Processing* 14(5): 1867–1874

TIMIT 1990 Acoustic-phonetic continuous speech corpus. National Institute of Standards and Technology Speech Disc 1-1.1. Fisher W, Doddington G, Goudie Marshall K M 1986 The DARPA speech recognition research database: Specifications and status. *Proc. DARPA Workshop on Speech Recognition*, California, 93–99

Tribolet J 1979 A new phase unwrapping algorithm. *IEEE Trans. Acoust. Speech Signal Process* 2: 170–179

Yegnanarayana B 1979 Formant extraction from linear-prediction phase spectra. *J. Acoust. Soc. Am.* 63: 1638–1640

Yegnanarayana B, Murthy H A 1992 Significance of group delay functions in spectrum estimation. *IEEE Trans. Signal Process.* 40(9): 2281–2289

Yegnanarayana B, Saikia D K, Krishan T R 1984 Significance of group delay functions in signal reconstruction from spectral magnitude or phase. *IEEE Trans. Acoust. Speech Signal Process* 3: 610–623

Yip P, Rao K R 1997 *Discrete cosine transform: Algorithms, advantages and applicatons* (Boston, USA: Academic Press)

Zissman M A 1996 Comparison of four approaches to automatic language identification of telephone speech. *IEEE Trans. Speech Audio Process* 4(1): 31–44