IMPLEMENTATION OF VOICE ACTIVITY ALGORITHM USING GROUP DELAY FUNCTIONS

A Project Report

submitted by

DILEEP G

EE12M004

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2014

THESIS CERTIFICATE

This is to certify that the thesis titled **IMPLEMENTATION OF VOICE ACTIVITY ALGORITHM USING GROUP DELAY FUNCTIONS**, submitted by **DILEEP G, EE12M004**, to the Indian Institute of Technology, Madras, for the award of the degree of **MASTER OF TECHNOLOGY**, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. C. S. Ramalingam
Research Guide
Associate Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 14th May, 2014

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my research advisors Prof. C. S Ramalingam, Prof. Umesh S for their great concern, continual encouragement and invaluable advice. Their enthusiasm, knowledge and loyalty towards students have been truly inspirational for me.

I would like to thank all the professors who taught me during my days at IITM. I would like to thank all my friends specially Akhil, Sriram and Vaisak for their support and making my stay more memorable. I would especially thank my friends in speech processing lab for their continuous help and support. And I would like to extend my sincere gratitude to Mr. Sree Hari Krishnan (earlier MS scholar from IIT Madras) for his work and the thesis done by him, which I have used as a reference.

I am very grateful to my parents for their encouragement and the sacrifices they have been doing throughout their lives. I also thank my sister for her constant love and support. I am forever indebted to my family and I dedicate this work to them.

ABSTRACT

The primary objective of the work is to implement the VAD algorithm using group delay function which is a phase based feature. Here I've put my effort to reproduce a portion of results reported in the MS Thesis titled 'Voice Activity Detection using Group Delay Functions' of Mr. Sree Hari Krishnan (earlier MS scholar from IIT Madras). Throughout my work, I have used the thesis prepared by him as a reference. Earlier works have used time domain parameters, or spectral shaped parameters, or a combination of both for addressing VAD problem. Using group delay function, a much more robust representation of speech signal for VAD can be achieved.

Short term energy (STE) is a quantity which can be used for VAD. Our algorithm, GD-VAD, based on minimum phase group delay processing of STE yields a good performance in terms of speech and non-speech detection. Although STE is not robust to noise, group delay processing makes it more robust to noise. The group delay function of a minimum phase signal retains its structure even in the presence of noise.

The group delay based VAD is implemented and performance of the algorithm in the presence of noise is investigated. The voice samples for the project are taken from the AURORA database.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Voice activity detection is a technique in which presence or absence of human speech is detected. By exploiting the information about speech and non-speech in a speech signal, these algorithms had its main use in speech coding and in speech recognition. It helps to avoid unnecessary coding and transmission of non-speech packets in VoIP applications. VAD algorithm can be viewed as a binary classifier that labels each frame of speech as speech or non-speech respectively.

The result of the VAD algorithm should remain identical for clean and noisy speech signal. A typical speech conversation speech is characterized by a speech to non-speech ratio of forty to sixty. The primary motivation for VAD is rooted on this. VAD's benefits in applications such as speech coding and voice over IP (VoIP), include decreasing the average bit rate, increasing the number of users and lowering the power consumption. VAD is also used as a front-end to automatic speech recognition (ASR) systems to improve recognition accuracy and resource utilization. Its other uses include noise estimation for speech enhancement.

Since Mr. Sree Hari Krishnan had carried out this study earlier itself, what I needed was to obtain a result which matches with what he got in his work. Also the algorithm is exactly same as that he discussed in his thesis. I've to use a platform for the implementation purpose and I preferred Matlab.

1.1 Issues in VAD

The variability and time varying nature of background noise and the speech itself is one of the issues in VAD. The background speakers and vehicles results in varying background noise, while the difference among speakers and variations among speech units results in variability in speech. As a result, simple methods like energy or zero crossing level based VAD will fail in lower SNR environments.

VAD methods have to deal with the problem of misclassification between speech and non-speech frames. The intelligibility is seriously compromised when speech frames are classified as non-speech. VAD algorithm should be constrained to identify speech regions without compromising the detection of non-speech frames.

1.2 Structure of a Typical VAD Algorithm

Basically, in every VAD algorithms, features or parameters extracted from the speech signal are compared against an adaptive threshold or against explicit speech/non-speech models. Every VAD algorithm involves the following steps:

- Extraction of parameters: The first step in most of the VAD algorithms involves the extraction of time-domain or frequency domain parameters such as ,STE, zero crossing rate, LPCs etc.
- Initialization of counters: Normally VAD algorithms use some speech or non-speech threshold levels, here these counters are initialized.
- Making the VAD decision: These decisions are made based on models or adaptive thresholds. These thresholds are estimated using the counters from last step.
- Smoothing the VAD decision: The output of the VAD algorithm might fluctuates due to errors in decisions. These have to be smoothed. Such schemes are implemented a state machines that prevent rapid transitions from speech to non-speech, while allowing faster transitions from non-speech to speech.
- Updating the counters: Finally after the VAD decisions are made, the background counters updated.

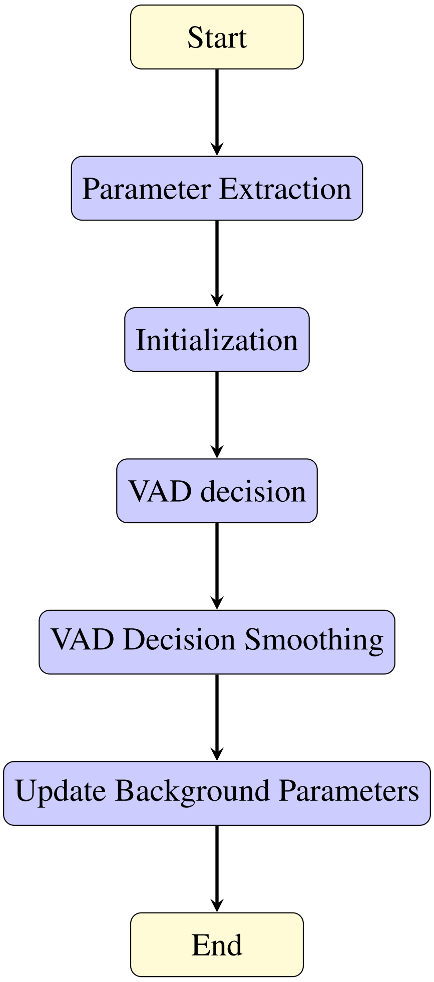

Figure 1.1: Flowchart of a typical VAD algorithm

1.3 Organization of the thesis

- The properties of group delay functions need to be studied. A theory of group delay relevant for the study is given in chapter 2
- The development of the GD-VAD algorithm is briefed in chapter 3. The theory of Buf-GD VAD is also described here.
- The major results of the study is discussed in chapter 4.
- Conclusions from the study is discussed in chapter 5.

CHAPTER 2

Theory of Group Delay Functions

2.1 Introduction

Group delay function is defined as the negative derivative of the Fourier transform phase function. The advantage of group delay is that it can be computed directly from the speech signal without need for phase unwrapping. but when we process just the short term phase, the formants which appear as transitions are masked by the wrapping of phase at multiples of 2π. Group delay functions have been applied to many speech processing tasks and its properties are used extensively in this thesis as well.

In this chapter, a brief overview of the theory of group delay functions is given. Also a summary of minimum phase signals and the significance of their properties are discussed. Procedures to convert from non-minimum phase signals to minimum phase signals in order to facilitate group delay processing is described.

2.2 Overview of Group Delay Functions

Let $X(\omega)$ denote the discrete time Fourier transform (DTFT) of a signal $x[n]$. Then,

$$X(\omega) = |X(\omega)| \exp^{j\theta(\omega)} \tag{2.1}$$

The group delay function is defined as the negative derivative of the unwrapped Fourier transform phase $(\theta(\omega))$. Mathematically, it is defined as follows:

$$\tau(\omega) = \frac{d(\theta(\omega))}{d(\omega)} \tag{2.2}$$

It is interpreted as the time delay that a frequency ω undergoes as it passes from the input to the output of the system. The group delay function is expressed in seconds.

When $\theta(\omega)$ is linear, the group delay is a constant and all frequencies undergo the same amount of delay. The group delay function has the property that the poles and zeros of the transfer function show up as peaks and valleys respectively.

2.3 Minimum Phase Group Delay Function

A signal $x[n]$ is minimum phase if the poles and zeros of its transfer function, lie within the unit circle. This imposes the constraint that not only is a signal causal and stable, its inverse is also causal and stable. Mathematically, this can be expressed as:

$$X(z) = \frac{b0 \prod_{i=1}^{m}(1 - b_i z^{-1})}{a0 \prod_{j=1}^{n}(1 - a_j z^{-1})} \tag{2.3}$$

where $\forall i, j \; |b_i| < 1$ and $|a_j| < 1$. The group delay function is well behaved only when the signal is minimum phase. The group delay function derived from the minimum phase signal is called the minimum phase group delay function.

2.4 Properties of Minimum Phase Group Delay Functions

The two properties of group delay function which are used extensively in this study are:

1. Additive property
2. High resolution property

2.4.1 Additive Property

Let $x_i[n]$ and $X_i(\omega)$ be the Fourier transform pairs, and let

$$X_3(\omega) = X_1(\omega).X_2(\omega) \tag{2.4}$$

Then,

$$|X_3(\omega)| = |X_1(\omega)|.|X_2(\omega)| \tag{2.5}$$

$$arg(X_3(\omega)) = arg(X_1(\omega)) + arg(X_2(\omega)) \tag{2.6}$$

$$\tau_{x_3}(\omega) = \tau_{x_1}(\omega) + \tau_{x_2}(\omega) \tag{2.7}$$

where $\tau_{x_3}(\omega)$, $\tau_{x_1}(\omega)$ and $\tau_{x_2}(\omega)$ corresponds to the group delay function of $X_3(\omega)$, $X_1(\omega)$ and $X_2(\omega)$ respectively. From Equations (2.4 - 2.6), we see that the multiplication in the spectral domain corresponds to addition in the group delay domain.

2.4.2 High Resolution Property

It has been established that minimum phase group delay functions exhibit squared magnitude behaviour in the neighbourhood of a pole. Consequently, the poles of the z-transform are resolved well in the group delay domain. Figures 3.4(a), (b) and(c) show a linear time-invariant (LTI) minimum phase system with a complex conjugate pole pair, its magnitude spectrum, and the corresponding group delay function. It can be seen that the pole is much better resolved in the group delay domain than it is in the magnitude spectral domain. Figures 3.4(d), (e) and (f) demonstrate this property with another complex conjugate pole pair. Lastly, Figure 3.4(g) shows an LTI system with the poles from Figures 3.4(a) and (d). In Figure 3.4(i), not only are poles sharper in comparison to Figure 3.4(h), but also do not exhibit a negative interference as evidenced in the magnitude spectrum. This is due to the combined effect of the additive and the high resolution properties of the group delay function.

2.5 Conversion of a Non-Minimum Phase Signal to a Minimum Phase Signal

Here, techniques that can be used to convert a non-minimum phase signal to a minimum phase signal or its estimate, is outlined. And these techniques are used in the development of VAD algorithm discussed in this thesis.

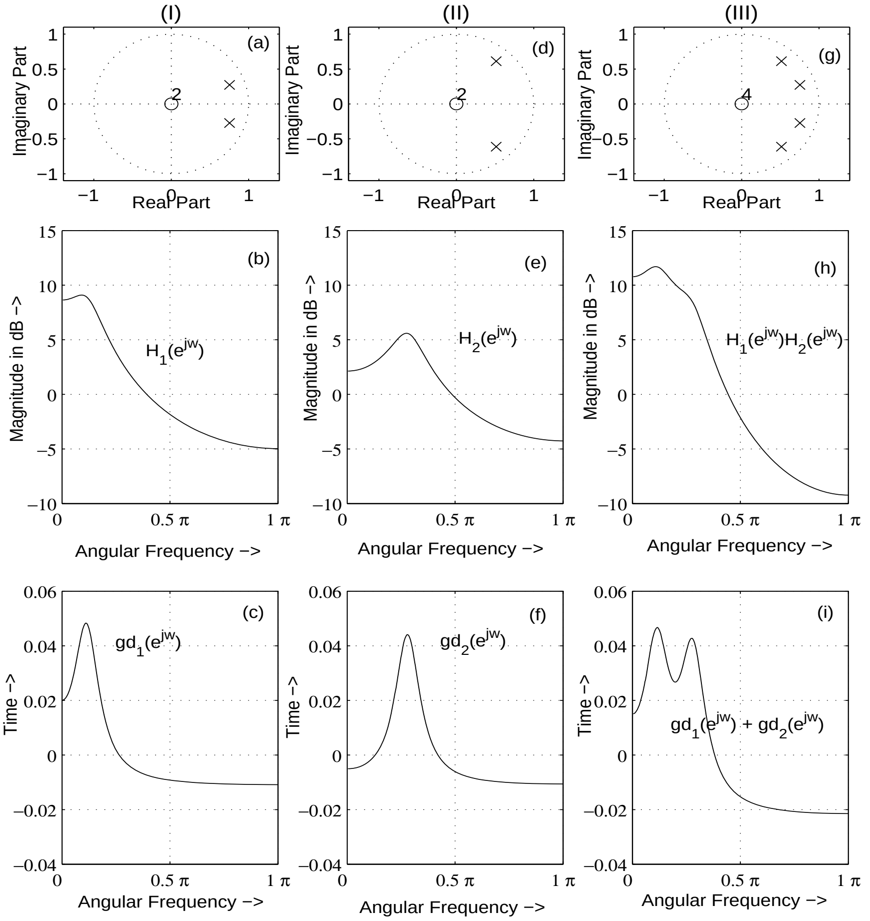

Figure 2.1: Illustration of high resolution property of group delay functions

2.5.1 Root Cepstrum based Minimum Phase Group Delay

A root cepstrum based method was proposed to compute a minimum phase equivalent signal of a non-minimum phase signal. It was shown that, for any non-minimum phase signal, the causal portion of the inverse Fourier transform of the squared magnitude spectrum is a minimum phase signal. The steps in converting a non-minimum phase signal $x_1[n]$ to its minimum phase equivalent $x_c[n]$, are given below:

- Let $X_1[k]$ be the DFT of $x_1[n]$
- Raise magnitude spectrum to power γ: $|X_1[k]|^\gamma$
- Compute the IDFT of $|X_1[k]|^\gamma$: $x_2[n]$
- Extract the causal portion of $x_2[n]$: $x_c[n]$
- Windowing $x_c[n]$ using a half hann window gives $\tilde{x}_c[n]$

In this transformation $x_c[n]$ retains the exact location of poles and zeros as that of $x_1[n]$.

CHAPTER 3

Group Delay based VAD Scheme

3.1 GD-VAD Algorithm

3.1.1 Short Term Energy

The short term energy of each frame of a speech signal ($x[n]$) is defined mathematically as:

$$E[m] = \sum_{n=0}^{N-1} (w[n]x[m-n])^2 \tag{3.1}$$

where $w[n]$ is the analysis window, N is the frame size in samples, and m is a multiple of the frame shift.

3.1.2 The Algorithm

In the GD-VAD the STE of the speech signal is viewed as the positive frequency part of the magnitude spectrum of an arbitrary minimum phase signal. The group delay of this signal is then computed. The speech regions of the signal are characterized by well-defined positive regions in the group delay spectrum, while the non-speech regions are identified by well-defined negative regions. A block diagram illustrating the proposed GD-VAD algorithm is shown in figure and a formal description of the same is listed below.

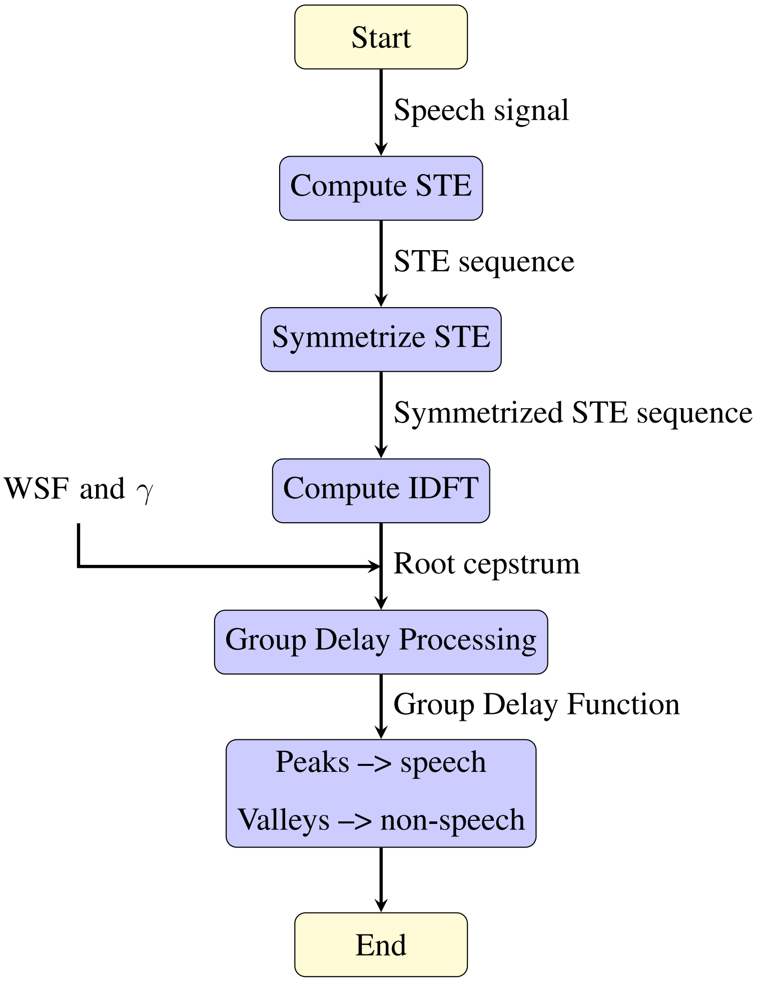

Figure 3.1: Flowchart of gd-VAD algorithm

1. Let $x[n]$ be the speech signal.

2. Compute its STE. Let us denote it by $E[m]$ with $0 \leq m \leq K - 1$ and K is the length of the STE defined as $K = \frac{\text{length of speech signal in samples}}{\text{frame-shift in samples}}$

3. Obtain $\tilde{E}[m]$ from E[m] by zero padding, making $\tilde{E}[m]$ an exact power of two.

$$\tilde{E}[m] = E[m] \qquad 0 \leq m \leq K-1 \tag{3.2}$$

$$\tilde{E}[m] = 0 \qquad K \leq m \leq M-1 \tag{3.3}$$

where $M = 2^{\lceil \log_2 K \rceil}$

4. Form the symmetric sequence $E_s[m]$

$$E_s[m] = \tilde{E}[m] \qquad 0 \leq m \leq M-1 \tag{3.4}$$

$$E_s[m] = \tilde{E}[2M-m-1] \qquad M \leq m \leq 2M-1 \tag{3.5}$$

   where $2M$ is the DFT order. This new sequence is considered as a magnitude spectrum of an arbitrary signal of $2M$ points between $-\pi$ and π and is denoted as $E_s[k]$. Let $2M-1 = N$.

5. To reduce the dynamic range, perform the following:

$$\acute{E}_s[k] = [E_s[k]]^\gamma \qquad 0 \leq k \leq N-1 \tag{3.6}$$

6. Compute the IDFT of the function $\acute{E}_s[k]$. The causal portion of the resulting signal denoted by $e[n]$ is minimum phase signal.

7. Compute the group delay function of $e[n]w[n]$, where w[n] is a low pass filter of length N_l. Then the group delay is computed as follows.

   - Compute the phase spectrum $\phi[k]$ of $e[n]w[n]$.
   - Compute the forward difference

$$\tau[k] = \phi[k] - \phi[k-1] \qquad 0 \leq n \leq N-1 \tag{3.7}$$

   where $\tau[k]$ is the group delay function.

8. For every peak in the group delay function ($\tau[k]$), compute the following:

   - Identify the valley before the peak as $f_b[i]$.
   - Identify the valley after the peak as $f_e[i]$.
   - The frames between $f_b[i]$ and $f_e[i]$ denote speech regions.

9. Output the VAD sequence $V_{GD}[n]$ consisting of 0 and 1, where 0 denotes a non-speech frame and 1 denotes a speech frame.

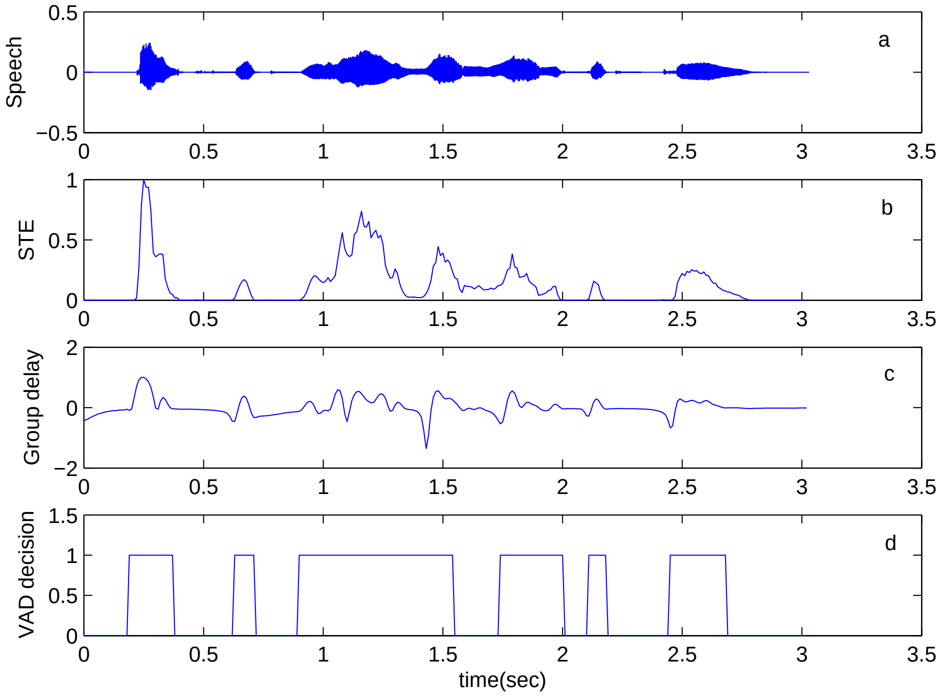

Figure 3.2: Steps involved in performing GD-VAD

Window Scale Factor ($WSF = \frac{N}{N_l}$) and γ are used to control the resolution of the group delay. In this algorithm those are empirical constants, those values are determined after large no of iterations. The steps involved in GD-VAD is described in the figure 3.2(a) to (d).

- Figure 3.2(a) refers to a clean speech signal.
- Figure 3.2(b) refers to the STE of the clean speech signal.
- Figure 3.2(c) refers to the group delay of the STE function.
- Figure 3.2(d) refers to the VAD decison.

3.1.3 Robustness of Group Delay Functions to Noise

The performance of the GD-VAD algorithm can be attributed to the robustness of the group delay representation.

- Figure 3.3(a) to 3.3(e) show the short term energy of speech signal recorded in the absence of noise, snr = 15dB, 10dB, 5dB and 0dB respectively.
- Figure 3.4(a) to 3.4(e) show the group delay function of the STE of speech signal recorded in the absence of noise, snr = 15dB, 10dB, 5dB and 0dB respectively.

The figures show an increase in the distortion of STE with an increase in noise level. This distortion can be characterized in terms of a change in the contour and a shift in the dc.

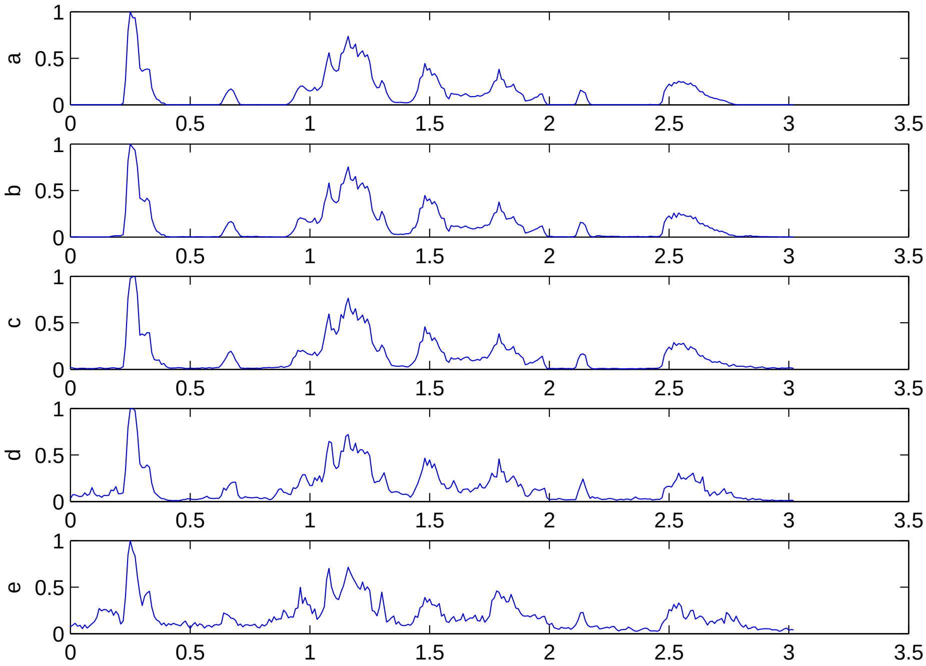

Figure 3.3: Short Term Energy

Figures 3.4(a) to (e) show the group delay processed STE of the speech signal in exactly the same environments. It can be observed from the figures that the group delay processed STE representation is much less distorted in the two aspects:

- The location of the peaks and valleys of the group delay processed STE function in clean environment is close to that of the group delay processed STE functions in noisy environments.

- There is also no dc shift in the group delay representation.

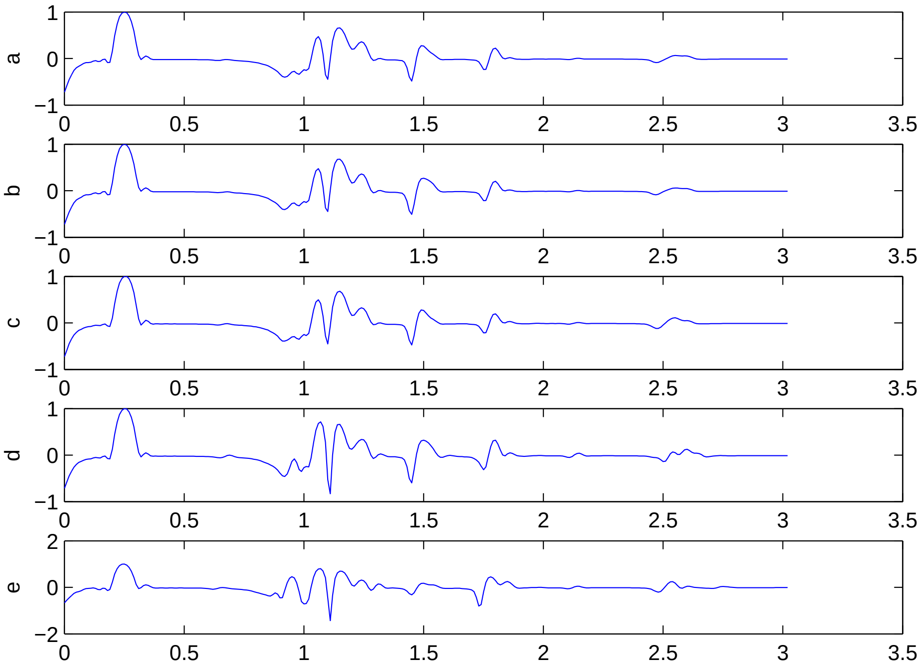

Figure 3.4: Group Delay Function

Since the GD-VAD is having a latency equal to that of the signal strength, a new algorithm with less latency is explored theoretically.

3.2 The BufGD-VAD Algorithm

- We have to apply GD-VAD on shorter segments of speech, which introduces the dual problems of a segment being entirely speech or being entirely noise. When either case occurs, the group delay function being an excellent resolver of peaks and valleys, enhances the variations in that segment. This leads to misclassification of frames.

- So we have to reduce the high resolution property, which can be done by appending a short segment of speech under consideration, with a known surrogate signal, so that length of the signal under consideration increases. This step, while solving the problem indicated in the previous step, presents another issue: the buffer size of the segment under consideration being too small to hold an entire speech or non-speech region, so that no peaks and valleys are entirely contained in the segment, to make a VAD decision.

- To solve the issue,we make an estimate of the maximum value attained by the group delay function in the noise regions. This estimated value is then subtracted from the group delay function. To make this estimate, the first few frames of the speech signal are assumed to be non-speech and the maximum value of the group delay function is estimated using this. The VAD decision is made on this noise compensated group delay function as follows: positive values indicate speech and the negative values indicate non-speech.
- A 5 point is performed to remove the rapid fluctuations finally

3.2.1 The Algorithm

The algorithm processes a buffer of the given speech signal as follows: For the bth buffer, the STE for each frame in that buffer is denoted as $\acute{e}_b[m]$. This is appended with the surrogate signal, $\beta s[n]$ to form the sequence $\tilde{e}_b[m]$, where $s[n]$ is a signal normalized to 1, and β is a scale factor, determined empirically. This sequence is viewed as the positive part of the magnitude spectrum of an arbitrary signal and is converted into its minimum phase equivalent. The group delay of this signal is obtained. A noise-compensated group delay, $\tau_b^n[k]$,is then obtained by subtracting the maximum value of the group delay of the first few frames (assumed to be noise). Next, a median filtering on $\tau_b^n[k]$ using the current and past elements is performed to yield $\bar{\tau}_b^n[k]$. VAD decisions are made on $\bar{\tau}_b^n[k]$ as follows: positive values of $\bar{\tau}_b^n[k]$ are classified as speech regions and negative values are classified as non-speech. This algorithm is formally listed below.

1. Given a speech signal $x[n]$, let us consider a buffer of contiguous frames. If the length of the buffer is B, then the number of buffers $P = \frac{\text{length of x[n]}}{B}$.
2. For each buffer $b (0 \leq b \leq P - 1)$, repeat steps 3 - 12:
3. Compute the STE, $\acute{e}_b[m]$, where $0 \leq m \leq B - 1$.

4. Append STE with surrogate signal. Form the sequence, $\tilde{e}_b[m]$

$$\begin{aligned} \tilde{e}_b[m] &= \acute{e}_b[m] \qquad 0 \leq m \leq B-1 && (3.8) \\ \tilde{e}_b[m] &= \beta rect[m-B] \qquad B \leq m \leq B+L-1 && (3.9) \\ \tilde{e}_b[m] &= 0 \qquad B+L \leq m \leq M-1 && (3.10) \end{aligned}$$

where $M = 2^{\lceil log_2(B+L) \rceil}$

5. Form the symmetric sequence, $\tilde{e}_{sb}[m]$

$$\tilde{e}_{sb}[m] = \tilde{e}_b[m] \qquad 0 \leq m \leq M-1 \tag{3.11}$$

$$\tilde{e}_{sb}[m] = \tilde{e}_b[2M-m-1] \qquad M \leq m \leq 2M-1 \tag{3.12}$$

6. Improving resolution using γ. To improve the resolution, perform the following:

$$\check{e}_{sb}[m] = [\tilde{e}_{sb}[m]]^{\gamma} \qquad 0 \leq m \leq 2M-1 \tag{3.13}$$

7. $\check{e}_{sb}[m]$ is considered as a magnitude spectrum of an arbitrary signal of $2M-1$ points in $(-\pi, \pi]$ and is denoted by $E_b[k]$.

8. Minimum phase equivalent. Compute the IDFT of $E_b[k]$. The causal portion of the resulting sequence denoted by $e_b[l]$ is a minimum phase signal.

9. Group delay computation. Compute the group delay function of $e_b[l]w[l]$, where $w[l]$ is a cepstral lifter of length $W1$ as follows.

   - Compute the phase spectrum $\phi_b[k]$ of $e_b[l]w[l]$.
   - Compute the forward difference

$$\tau_b[k] = \phi_b[k] - \phi_b[k-1] \qquad 1 \leq k \leq 2M-1 \tag{3.14}$$

   where $\tau_b[k]$ is the group delay function.

10. Compute the noise compensated group delay $\tau_b^n[k]$ as:

$$\tau_b^n[k] = \tau_b[k] - \tau_{max} \qquad 0 \leq k \leq 2M-1 \tag{3.15}$$

where $\tau_{max} = \max \tau_b[n]$ for $0 \leq n \leq T$ and T is an empirically determined threshold index.

11. Perform median filtering on the noise-compensated group delay $\tau_b^n[k]$ as:

$$\bar{\tau}_b^n[k] = median(\tau_b^n[k]) \qquad 0 \leq k \leq 2M - 1 \tag{3.16}$$

    where $median(.)$ computes a 5-point median.

12. VAD decision. If $\bar{\tau}_b^n[k] \geq 0$, classify frame as speech else if $\bar{\tau}_b^n[k] \leq 0$ classify frame as non-speech.

CHAPTER 4

Results

By executing the program on 500 test samples from AURORA database, we obtained a relation between WSF and γ with the SNR of the speech signal. We obtained an inverse relationship between WSF and SNR. This can be attributed to the fact that, at lower SNR (more noisy conditions), the STE function fluctuates more rapidly, and consequently, a low pass filter with a lower cutoff is mandated. Therefore a higher WSF is needed at a lower SNR. Also we got a similar inverse relationship between γ and SNR. As SNR noise level decreases, noise level increases. Consequently, there are more ripples in the STE contour. Thus to increase the dynamic range between the speech (characterized by high-energy regions) and non-speech regions (characterized by low-energy regions), the power ($\gamma \geq 1$) to which STE function needs to be raised has to be increased as well.

- Figure 4.1 is the result of GD-VAD on clean speech.
- Figure 4.2 is the result of GD-VAD on speech signal with snr 15dB.
- Figure 4.3 is the result of GD-VAD on speech signal with snr 10dB.
- Figure 4.4 is the result of GD-VAD on speech signal with snr 5dB.

In figures 4.1 to 4.4 the signals from top to bottom are in the order, speech signal, group delay of the STE function, VAD classifier for γ = 2 and VAD classifier for γ = 3 respectively.

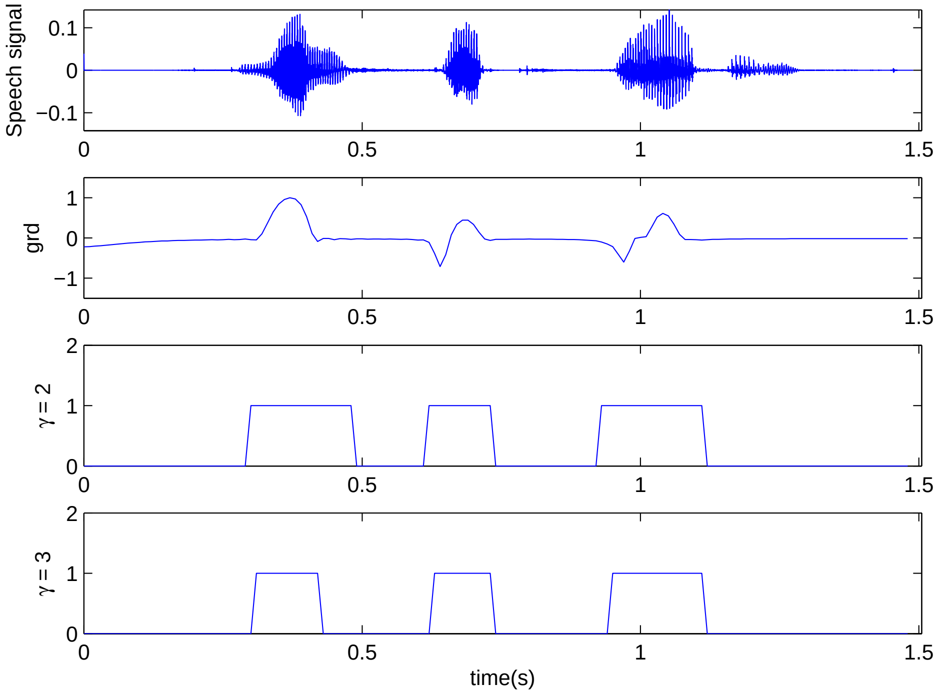

Figure 4.1: GD-VAD on clean speech signal

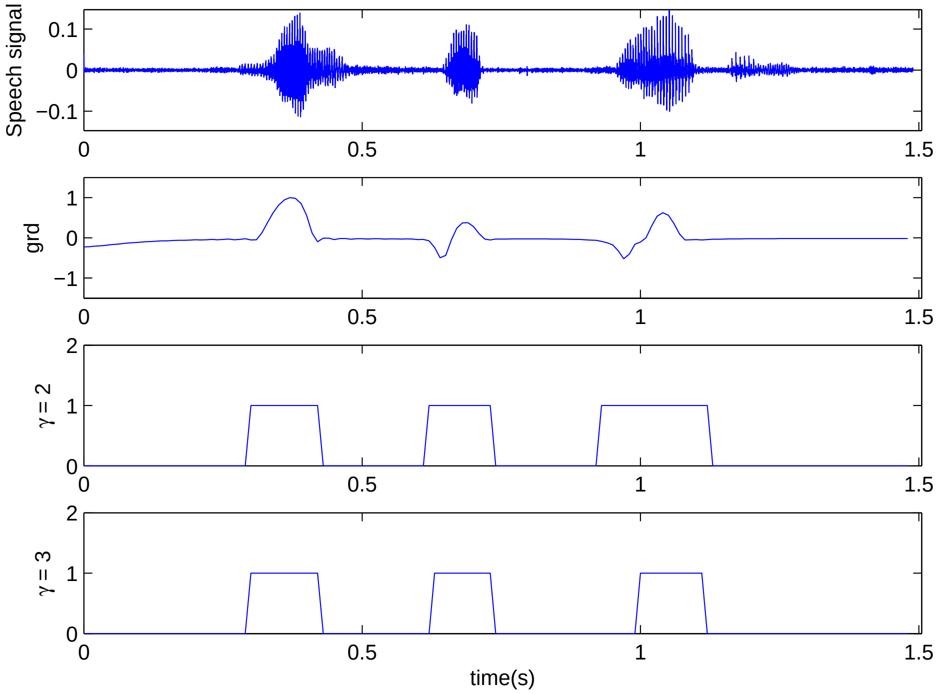

Figure 4.2: GD-VAD on speech signal with SNR 15dB

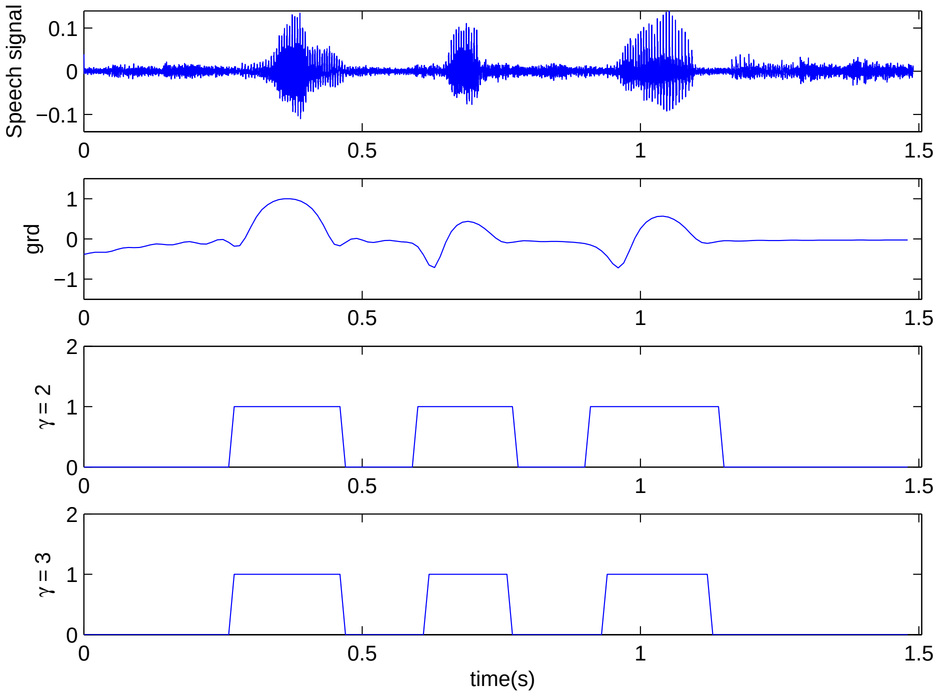

Figure 4.3: GD-VAD on speech signal with SNR 10dB

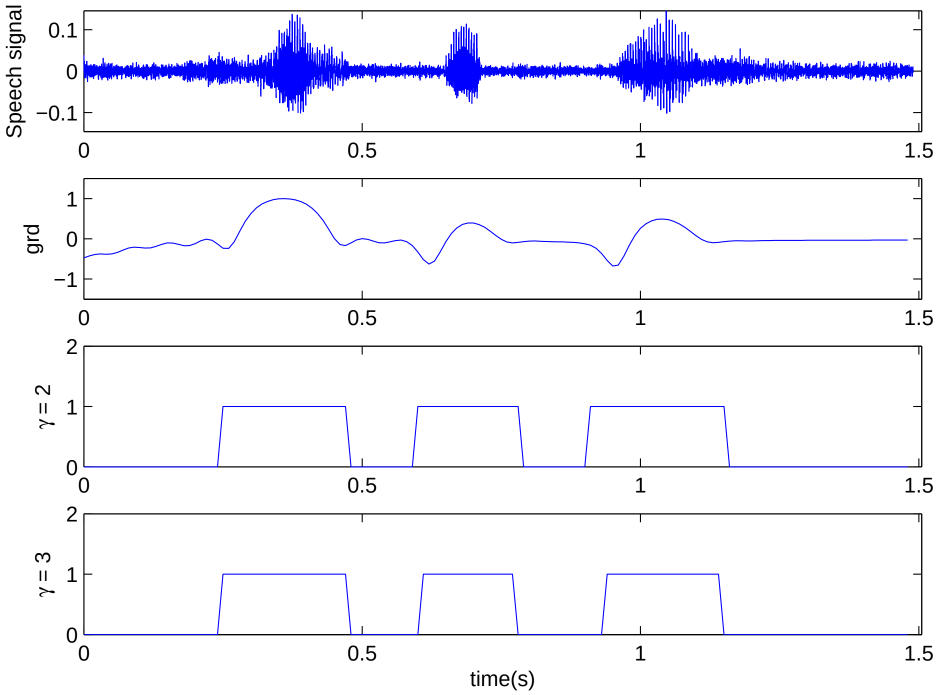

Figure 4.4: GD-VAD on speech signal with SNR 5dB

- For clean speech signal WSF is set at 2.

- For speech signal with SNR 15 dB WSF is set at 3.

- For speech signal with SNR 10 dB WSF is set at 6.
- For speech signal with SNR 5 dB WSF is set at 7.

CHAPTER 5

Summary and Conclusions

5.1 Summary

The results I've got were matching with that of what Mr. Sree Hari got in his work. Traditionally VAD algorithms have used time-domain characteristics or features derived from the spectral shape. Time-domain features show dependencies on estimates of absolute levels, such as dc values. Spectral shape based features have a tendency to become flatter as the noise level increases. As the spectrum becomes flatter, the information about the formants in the signal, starts to decrease. For a VAD problem, this could lead to poorer identification of speech regions.

So, one of the primary objectives of this thesis was to study alternate representations of the speech signal for VAD. In particular, the use of phase based methods,specifically group delay functions, for solving the voice activity detection was explored. The GD-VAD algorithm yielded good performance, but suffers from a latency equal to the signal length.Framing needs to be done before starting the algorithm to calculate the STE. Since the signal on which group delay processing is performed is a STE function, which only gives one value per frame shift. So to to have a reasonable number of samples for group delay processing, an equal number of frame-shifts are needed which results in increased latency in processing Also this method needed two quantities(WSF andγ) to be updated manually depending on the SNR of the speech signal. An improvement can be added to the GD-VAD algorithm for processing shorter segments of speech.

5.2 Conclusions

Since I've tried to implement the VAD algorithm discussed by earlier MS scholar Mr. Sree Hari Krishnan, I would like to share his conclusions also along with mine.

- The group delay function of a minimum phase signal preserve its structure even in the presence of noise which makes group delay a useful quantity for the purpose.

- The implemented algorithm works well for clean speech and a degradation in reliability of the classifier is observed as the signal to ratio approaches zero dB.

- As the proposed algorithm is having a latency equal to the signal length, algorithm can be modified to process on shorter segments of speech.

- Since the WSF and gamma depends on the SNR, an SNR estimation scheme can be employed which can be used to estimate WSF automatically

CHAPTER 6

Bibliography

- V. K. Prasad, T. Nagarajan, and H. A. Murthy, “Automatic segmentation of continuous speech using minimum phase group delay functions,” Speech Comm., vol. 42, pp. 429– 446, 2004.

- T. Nagarajan, V. K. Prasad, and H. A. Murthy, “Minimum phase signal derived from root cepstrum,” IEE Electronics Lett., vol. 39, 2003.

- Sree Hari Krishnan.P, Padmanabhan.R and Hema A Murthy, “Robust Voice Activity Detection using Group Delay Functions”, Proceedings of IEEE ICIT 2006, Dec 2006, pp 2603-2607.

- Sree Hari Krishnan.P, Padmanabhan.R and Hema A Murthy, “Voice Activity Detection using Group Delay Processing on Buffered Short-term Energy”, Proceedings of NCC2007, Jan 2007, pp 169-172.

APPENDIX A

Matlab Code

```
clc;

% clear ;
% close all;
delete('VAD*.fig');

%%% Frame Details  %%%
%%% Copy wav file to current folder %%%
frame_interval=25e-3;
frame_overlap=15e-3;
title1=4;
WSF=[2 3 4 5 7];
gamma1=[2 3 4 ];
% global  j;



nameList = ls('Aurora\clean1\*.wav');
% folderList={'Aurora\clean1\';'Aurora\N1_SNR-5\';...
%'Aurora\N1_SNR0\';'Aurora\N1_SNR5\';'Aurora\N1_SNR10\';...
%'Aurora\N1_SNR15\';'Aurora\N1_SNR20\'};
folderList={'Aurora\clean1\';'Aurora\N1_SNR20\';...
    'Aurora\N1_SNR15\';'Aurora\N1_SNR10\';'Aurora\N1_SNR5\'};
var_fol=1;


    accuracyInDiffSpeech = [];
```

```
for varInDiffSpeech = 535 %length(nameList)
fprintf(1,'%%%%%%%%%%%%%%%%%%%%%%%%%%%\n');
fprintf(1,'         %dth speech file\n',varInDiffSpeech);
 for WSFtemp=5
VAD = {};
grdcell={};



for varInSameSpeech = var_fol %1:length(folderList)
for gammatemp=1:3
    dat=[];M=[];N=[];VAD2=[];c1=[];c2=[];e=[];e_tem=[];...
        grd=[];grd1=[];half_hann_win=[];newdat=[];...
        ph_ind=[];phase1=[];refVAD=[];ste=[];ste1=[];
    ste2=[];tem_win=[];time_index=[];valley=[];win=[];win1=[];


    eval(['[dat_' num2str(varInDiffSpeech) '_' ...
        num2str(varInSameSpeech) ...
        ' fs] = wavread([folderList{varInSameSpeech}'...
        ' nameList(varInDiffSpeech,:)]);']);
    dat=eval(['dat_' num2str(varInDiffSpeech) '_' ...
        num2str(varInSameSpeech)]);

    [dat fs] = wavread('FBA_367A_4');
    dat(1)=dat(2);
    %%% Calculating no of samples %%%
    fi=frame_interval*fs;
    fo=frame_overlap*fs;
    fa=fi-fo;
    %%% No of frames %%%
    nofr=1+floor((length(dat)-fi)/fa);
    newdat=zeros(fi,nofr);
```

```
%%% Framing and making the matrix with columns as frames %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tem_win=hann(floor(2*fi));
half_hann_win=tem_win(floor(length(tem_win)/2)+1:end);
%         half_hann_win=rectwin(fi);

j=1;
k=fi;
for i=1:nofr
    temp=dat(j:k);
    newdat(:,i)=temp;
    j=j+fa;
    k=k+fa;
end;
clear temp;


%%%%  Frame Energy %%%%
ste=zeros(1,nofr);
for j=1:nofr
    for i=1:fi
        ste(j)=ste(j)+(newdat(i,j)*half_hann_win(i)).^2;
    end
end
ste=ste./max(ste);
l=length(ste);
M=2^(ceil(log2(l)));
N=2*M;
ste1=zeros(1,N);
for i=1:l
```

```
            ste1(i)=ste(i);
            ste1(2*M-i+1)=ste(i);
        end
        ste2=(abs(ste1)).^(gamma1(gammatemp));
        e_tem=ifft(ste2,N);

        %%% causal conversion %%%
        e=zeros(1,N);
        e1=e;
        e(1:N/2)=e_tem(1:N/2);
%         e(1)=e_tem(1)/2;
        %%%%%%%%%%%%%%%%%%%%%%%%%
        %%% windowing %%%%
        win1=hann(floor(2*N/WSF(WSFtemp)));
        win=win1(floor(length(win1)/2)+1:end);
        for i=1:length(win)
            e1(i)=e(i)*win(i);
        end
        E1=fft(e1,N);



        phase1=angle(E1);
        % phase2=unwrap(phase1);
        %%% forward diff %%%
        grd=[];
        for i=2:N
            grd=[grd    phase1(i)-phase1(i-1)];
        end;
        grd=-grd;
%         grd=grpdelay(e1,N);
%         grd=grd/abs(max(grd));
VAD2=zeros(1,nofr);
grd1=grd(1:nofr);
```

```
grd1=grd1/max(grd1);
thr=max(grd1)/600;
grdcell=[grdcell; grd1];
thr=0;
[c1 c2]=get_peaks(grd1,thr);
for i=1:length(c2)
    [valley(i,1) valley(i,2)]=get_valleys(grd1,c2(i));
end

for i=1:size(valley,1)
    VAD2(valley(i,1):valley(i,2))=ones(1,valley(i,2)-...
        valley(i,1)+1);
end
skip1=8;
for i=1:length(VAD2)-skip1
    if ((VAD2(i) == 1) && (VAD2(i+skip1)==1))
        VAD2(i+1:i+skip1)=ones(1,skip1);
    end
end

datcpy=dat(1:nofr*fa);
%         end
          frame_index=0:nofr-1;
%         time_index=0:1/fs:(fi-1+(nofr-2)*fa)/fs;
        time_index=0:1/fs:(nofr*fa-1)/fs;
        ph_ind=0:fa/fs:(nofr*fa-1)/fs;
          datcpy=dat(1:nofr*fa);
%         end
          frame_index=0:nofr-1;
%         time_index=0:1/fs:(fi-1+(nofr-2)*fa)/fs;
        time_index=0:1/fs:(nofr*fa-1)/fs;
        ph_ind=0:fa/fs:(nofr*fa-1)/fs;
        figure;
        subplot(4,1,1)
%         plot(time_index,dat(1:fi+(nofr-2)*fa));
        plot(time_index,datcpy);

%         xlim([0 (nofr-1)*fa./fs]);
        % title(title1);
        title(WSF(WSFtemp))
        % plot(dat(1:nofr*fa));



        subplot(4,1,2)
        plot(frame_index*fa./fs,ste)
        % plot(ste);

        ph_ind=0:fa/fs:(nofr*fa-1)/fs;
        subplot(4,1,3)
        plot(ph_ind,grd(1:nofr));
        % plot(grd1);

        subplot(4,1,4)
        plot(ph_ind,VAD2(1:nofr))
        ylim([-.5 1.5]);
        close



%         saveas(gcf,['VAD_' num2str(varInDiffSpeech)...
%'_' num2str(gamma1(gammatemp)) '_' num2str...
%(WSF(WSFtemp)) '.fig']);
        % plot(VAD2(1:nofr));
        VAD = [VAD ; VAD2];
    end
```

```
%       subplot(6,1,1);
%       plot(time_index,datcpy);
%       subplot(6,1,2);
%       plot(ph_ind,VAD{1});
%       subplot(6,1,3);
%       plot(ph_ind,VAD{2});
%       subplot(6,1,4);
%       plot(ph_ind,VAD{3});
%       subplot(6,1,5);
%       plot(ph_ind,VAD{4});
%       subplot(6,1,6);
%       plot(ph_ind,VAD{5});
%       saveas(gcf,['VAD_' num2str(varInDiffSpeech) ...
%           '_' num2str(WSF(WSFtemp)) '_' num2str...
%           (gamma1(gammatemp)) '.fig']);
%

    nofr_time=(nofr*(fi-fo)+fo)/fs;
    figure;
subplot(4,1,1);
     plot(time_index,datcpy);
     ylabel('Speech signal')
%       title(['VAD ', num2str(varInDiffSpeech), '  ',...
%           num2str(gamma1(gammatemp)),  '.fig']);
     xlim([0 nofr_time]);
     maxi=max(abs(datcpy));
     ylim([-maxi maxi]);
     subplot(4,1,2);
     plot(ph_ind,grd1);
     ylabel('grd')
     xlim([0 nofr_time]);
     ylim([-1.5 1.5]);
```

```
 for plotvar=1:2
     subplot(4,1,plotvar+2);
     plot(ph_ind,VAD{plotvar});
     ylabel(['\gamma = ', num2str(gamma1(plotvar))])
     xlim([0 nofr_time]);
     ylim([0 2])
 end

 saveas(gcf,['VAD_' num2str(varInDiffSpeech) ...
     '_' num2str(gamma1(gammatemp)) '(WSF 2-10).fig']);
end
end
end
```