A UNIFIED APPROACH TO SPEECH ENHANCEMENT AND VOICE ACTIVITY DETECTION

by

Ceyhan Kasap

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2009

ACKNOWLEDGEMENTS

With my deepest gratitude; I would like to thank my thesis supervisor Prof. Levent Arslan for his polite guidance, great patience and endless support all through this thesis. It was a great pleasure to work with him. I am also grateful to Assist. Prof. Murat Saraçlar and Assoc. Prof. Engin Erzin for participating in my thesis jury. Their criticism and helpful comments greatly improved the thesis.

I also would like to thank my mother İsmet, my father Fazlı, my sister Neslihan and my love Ayşe for their endless love, encouragement, and support.

Lastly, I would like to express my gratitude to TÜBİTAK for supporting me with National Scholarship Programme for M.Sc. Students - 2228.

ABSTRACT

A UNIFIED APPROACH TO SPEECH ENHANCEMENT AND VOICE ACTIVITY DETECTION

In this work, a unified system for voice activity detection (VAD) and speech enhancement is proposed. In the proposed system, there is mutual exchange of information between VAD and speech enhancement blocks. A new, robust and low complexity VAD algorithm is implemented for the VAD block of the unified system. The newly proposed VAD algorithm uses a periodicity measure and an energy measure obtained from spectral energy distribution and spectral energy difference of the input speech data. For the speech enhancement block, the Modified Wiener Filtering (MWF) algorithm is utilized. It has been shown that the utilization of information exchange between the VAD and MWF algorithms in the unified system increases the performance of both algorithms and the proposed unified system improves the robustness of a speech recognition system significantly. Both of the enhanced algorithms are non-iterative. Therefore, the proposed unified system is computationally attractive for real-time applications.

ÖZET

KONUŞMA İŞARETİNİN İYİLEŞTİRİLMESİ VE SES AKTİVİTESİ ALGILAMA İÇİN BÜTÜNCÜ BİR YAKLAŞIM

Bu çalışmada, konuşma işaretinin iyileştirilmesi ve ses aktivitesi algılama (SAA) için bütüncü bir sistem önerilmiştir. Önerilen sistemde SAA ve konuşma işareti iyileştirme blokları arasında karşılıklı bilgi alışverişi bulunmaktadır. Bütüncü sistemin SAA bloğunda kullanılmak üzere; yeni, gürbüz ve düşük karmaşıklıklı bir SAA algoritması önerilmiştir. Önerilen yeni SAA algoritması periyodiklik ve spektral enerji dağılımı ve spektral enerji farklılığından elde edilen enerji ölçütlerini kullanmaktadır. Konuşma işareti iyileştirme bloğu için Modifiye Wiener Süzgeci (MWS) algoritmasından yaralanılmıştır. SAA ve MWS algoritmaları arasındaki karşılıklı bilgi alışverişinin her iki algoritmanın performansını arttırması ve bütüncü sistemin ses tanıma sistemlerine gürbüzlük açısından sağladığı katkılar gösterilmiştir. Geliştirilmiş algoritmaların her ikisi de döngüsüzdür. Bu nedenle, önerilen bütüncü sistem gerçek zamanlı uygulamalar için caziptir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

$\boldsymbol{A^H}$	Hermitian of matrix A
A^*	Complex conjugate of A
c_i	i^{th} element of cepstral vector $\bar{c}$
c'_i	i^{th} element of cepstral vector $\bar{c'}$
d	Weighted Euclidean distance
$e^{j\phi_n(\omega)}$	Phase spectrum of $N(\omega)$
$e^{j\phi_s(\omega)}$	Phase spectrum of $S(\omega)$
$e^{j\phi_y(\omega)}$	Phase spectrum of $Y(\omega)$
$e(n)$	Prediction error
E	Mean squared error
$E[.]$	Expectation operator
E_n	Noise energy
$\hat{E_n}$	Average noise energy estimate
$(E_n)_k$	Noise energy for the k^{th} frame
E_s	Short time energy of $s(n)$
E_y	Noisy speech energy
$E(\omega)$	Estimation error at frequency domain
$F^{-1}\{.\}$	Inverse Fourier transform operation
F_s	Sampling frequency
$\hat{g}_s$	DC gain of the noise-free speech signal
g_y	DC gain of the noisy speech signal
$h(k)$	Filter coefficient
$H(\omega)$	DFT of h(n)
I_0	Modified Bessel functions of order zero
I_1	Modified Bessel functions of order one
$\boldsymbol{I}$	Identity matrix
L	Number of frames used in SD computation
m_energy	Energy measure for the current frame
$m\_energyDifference$	Spectrally weighted energy difference parameter

$m\_energyWeighted$	Spectrally weighted energy parameter
$m\_maxEnergy$	Maximum energy threshold for a speech frame
$m\_minEnergy$	Minimum energy threshold for a speech frame
$m\_subBandEnergy[i]$	Energy of the frame in the i^{th} subband
$m\_subBandAvg[i]$	Average energy for the i^{th} subband
m_Vad	Sum of the soft decision values for the last 20 frames
$n(t)$	Noise signal
N	Length of $s(n)$
$N(\omega)$	DFT of noise signal
$\|N(\omega)\|$	Magnitude spectrum of $N(\omega)$
$\|\hat{N}(\omega)\|$	Estimate of the magnitude of noise spectrum
p	Period of $s(n)$
$prob\_voice$	Probability of voicing parameter
P	Order of computation
$\boldsymbol{P}$	Orthogonal projection matrix
$P_n(\omega)$	Noise power spectrum
$\hat{P}_n(\omega)$	Estimated noise power spectrum
$P_n(\omega)_k$	Noise power spectrum estimate for the k^{th} noisy input frame
$Pn(w)_{\text{EMWF}}$	Noise power spectrum estimate of EMWF algorithm
$Pn(w)_{\text{MWF}}$	Noise power spectrum estimate of MWF algorithm
$\hat{P}_s(\omega)$	Estimated clean speech power spectrum
$P_{sy}(\omega)$	Cross power spectrum of s(n) and y(n)
$P_y(\omega)$	Power spectrum of y(n)
$P_{ys}(\omega)$	Cross power spectrum of y(n) and s(n)
$P_y(\omega)_k$	Power spectrum of the k^{th} noisy input frame
R^n	n dimensional vector space
$R_{ss}(k)$	k^{th} autocorrelation lag for $s(n)$
$R_{nn}(\tau)$	Autocorrelation function of $n(t)$ for a time difference of τ
$R_{sy}(k)$	k^{th} cross correlation lag between s(n) and y(n)
$R_{yy}(k)$	k^{th} autocorrelation lag for $y(n)$
$sgn[.]$	Signum function

$silence\_counter$	Variable to keep the number of successive silence frames
$soft\_decision$	Soft decision value assigned to the current frame
$speech\_threshold$	Speech threshold value for the current frame
$s(n)$	Speech frame
$\hat{s}(n)$	Wiener filter output
$\tilde{s}(n)$	Linear predictor for $s(n)$
$s(t)$	Clean speech signal
$\hat{s}(t)$	Clean speech signal estimate
$S(\omega)$	DFT of clean speech signal
$\hat{S}(\omega)$	Clean speech spectrum estimate
$S(\omega_k)$	k^{th} frequency component of $S(\omega)$
$\hat{S}(\omega_k)$	k^{th} frequency component of $\hat{S}(\omega)$
$\|S(\omega)\|$	Magnitude spectrum of $S(\omega)$
$\boldsymbol{U}$	m by m unitary matrix
$\bar{v}$	A vector in subspace V
V	Vector subspace V
$\boldsymbol{V}$	n by n unitary matrix
$\bar{w}$	A vector in subspace W
W	Vector subspace W
$\bar{x_i}$	i^{th} basis vector
$y(n)$	Wiener filter input
$y(t)$	Noisy speech signal
$\bar{y}$	Noisy speech vector
$\bar{y_1}$	Component of $\bar{y}$ that lies on the signal subspace
$\bar{y_2}$	Component of $\bar{y}$ that lies on the noise subspace
$Y(\omega)$	DFT of noisy speech signal
$Y(\omega_k)$	k^{th} frequency component of $Y(\omega)$
$\|Y(\omega)\|$	Magnitude spectrum of $Y(\omega)$
Z	Zero-crossing rate of $s(n)$
α	Noise suppression factor

α_k	Prediction coefficients
α_t	Time dependent noise suppression factor
α'	Constant that scales E_n/E_y term
β	Power of the Wiener filter
γ	Interpolation factor
γ_k	A posteriori SNR
δ	Variable for upconstant value computation
$\lambda_n(k)$	Variance of the k^{th} spectral component of the noise
$\lambda_s(k)$	Variance of the k^{th} spectral component of the speech
λ_t	Time dependent multiplication factor that scales the noise spectrum
μ	Heuristically determined constant for computation of α_t
μ_n	Mean of $n(t)$
ν	Heuristically determined constant for computation of α_t
ξ_k	A priori SNR
σ_i	Singular value of A
$\mathbf{\Sigma}$	Diagonal matrix of σ_i
ω_k	k^{th} frequency index

DFT	Discrete-time Fourier Transform
EMWF	Enhanced Modified Wiener Filtering
FIR	Finite Impulse Response
HMM	Hidden Markov Model
IIR	Infinite Impulse Response
LPC	Linear Prediction Coding
LSPE	Least Squares Periodicity Estimator
MMSE	Minimum Mean Square Estimate
MOS	Mean Opinion Score
MWF	Modified Wiener Filtering
PESQ	Perceptual Evaluation of Speech Quality
RF	Radio Frequency
SAF	Sum of Activation Function

SD	Spectral Distortion
SNR	Signal to Noise Ratio
SVD	Singular Value Decomposition
TIMIT	Texas Instruments and Massachusetts Institute of Technology
VAD	Voice Activity Detection
VoIP	Voice Over IP
ZCR	Zero-Crossing Rate

1. INTRODUCTION

1.1. Motivation

Voice activity detection (VAD) and speech enhancement systems have been extensively studied by the speech processing community since 1970s because of their importance in many different applications like wireless communications, speech coding, speech recognition, hands-free conference and so on.

During a typical conversation between two people, the average time in which speech exists is less than 40 per cent of the total conversation time [1]. A system that tries to separate the speech and non-speech portions of a conversation is known as a voice activity detector. The ability to separate the speech and non-speech portions of a conversation is of interest in many speech applications. For example, the use of VAD in cellular telephone systems facilitates an efficient consumption of the available radio frequency (RF) bandwidth [2]. VAD systems also reduce the power consumption in portable devices [3]. Clipping of the non-speech frames enables reduction of bandwidth consumption in voice over IP (VoIP) systems, too [4].

The binary detection problem of the speech signal in the audio input becomes difficult in the presence of rapidly changing background noise and low signal to noise ratio (SNR) conditions. VAD problem is also complicated by the fact that many kinds of acoustic noise share similar properties with unvoiced speech. The major focus of research in voice activity detection has been the development of low complexity, efficient and robust VAD algorithms for a variety of acoustic noises and SNRs.

Speech enhancement is concerned with improving the perceptual aspects of speech that has been degraded by background noise. In fact, speech enhancement has three major goals [5] : a) to improve perceptual aspects (e.g., quality, intelligibility) of a given sample function of degraded speech signal; b) to increase robustness of speech coders to input noise; and, c) to increase robustness of speech recognition systems in

the presence of background noise. Since the goal of speech enhancement algorithms is reducing or suppressing the background noise to some degree, they are referred to as noise suppression algorithms [6].

The need to eliminate the effects of noise in speech signals arises in many different situations and applications [6]. Voice communication over cellular network is one example. Due to the mobility of users, cellular communication systems are likely to suffer from the background noise present in the environment. Utilization of a speech enhancement algorithm at the transmitting end enhances the quality of speech at the receiving end. Speech enhancement is also crucial for military communications. In an air-ground communication scenario, speech enhancement techniques are needed to improve quality and preferably intelligibility of the pilot's speech that has been corrupted by the high levels of cockpit noise. In a teleconferencing system, noise sources present for one participant will inevitably be broadcast to all participants. Therefore, suppression of background noise is required to improve the overall quality of the teleconferencing system. Finally, speech enhancement has an important role in the design of hearing aids. Hearing aids tend to increase the level of all kinds of input signal whether it is noise or actual speech. As a result, hearing-impaired people experience difficulty when using hearing aids under noisy conditions. Speech enhancement algorithms can be used to clean the noisy signal before amplification.

1.1.1. Unified System for VAD and Speech Enhancement

This work is concentrated on implementing a unified system for VAD and speech enhancement. Most of the speech applications that incorporate VAD and speech enhancement blocks typically utilize these blocks separately and independently. One example might be the speech recognition systems. Before processing the audio input for recognition, speech recognition systems typically require a VAD module in order to determine the utterances in the input. After the VAD module, a noise removal system is generally applied to reduce the background noise for efficient recognition. Another example might be the mobile phones for cellular communication systems. Mobile phones typically eliminate the silence regions of the input using a VAD block and then sup-

press the noise in the remaining speech portions using a speech enhancement block. Our proposed unified system aims to provide a single and efficient framework for these seemingly separated functionalities.

Our basic motivation for the proposed unified system is primarily rooted on the assumption that VAD and speech enhancement problems are closely related. We expect that although VAD and speech enhancement systems operate for distinct purposes, the two systems might operate better if they mutually exchange information in a unified system. Suppression of acoustic noise in the audio input signal would improve VAD performance. Similarly, the discrimination of speech/non-speech character of the input frames, which would be obtained from a VAD algorithm, would enable better noise attenuation.

We propose a new VAD algorithm to be used for the VAD block of the unified system. Since the newly proposed VAD algorithm is particularly designed for the unified system, computational speed and robustness to noise are the most important criteria for its implementation. The newly proposed VAD algorithm uses a periodicity measure and an energy measure obtained from spectral energy distribution and spectral energy difference of the input speech data. Utilization of speech enhancement enables us to implement the so called *Hybrid VAD* algorithm to be used in the proposed unified system. Availability of noised suppressed frames in the Hybrid VAD algorithm allows the extraction of more reliable VAD features and therefore improves the speech/non-speech detection performance.

The speech enhancement block of the unified system relies on the Modified Wiener Filtering (MWF) approach proposed in [7]. By making the VAD decisions explicitly available for MWF algorithm, the algorithm is modified to implement the so called *Enhanced Modified Wiener Filtering (EMWF)* algorithm with increased noise suppression performance. EMWF algorithm in the unified system utilizes speech/non-speech information to better characterize and attenuate the background noise.

Figure 1.1 depicts a schematic representation of our proposed unified system for

VAD and speech enhancement.

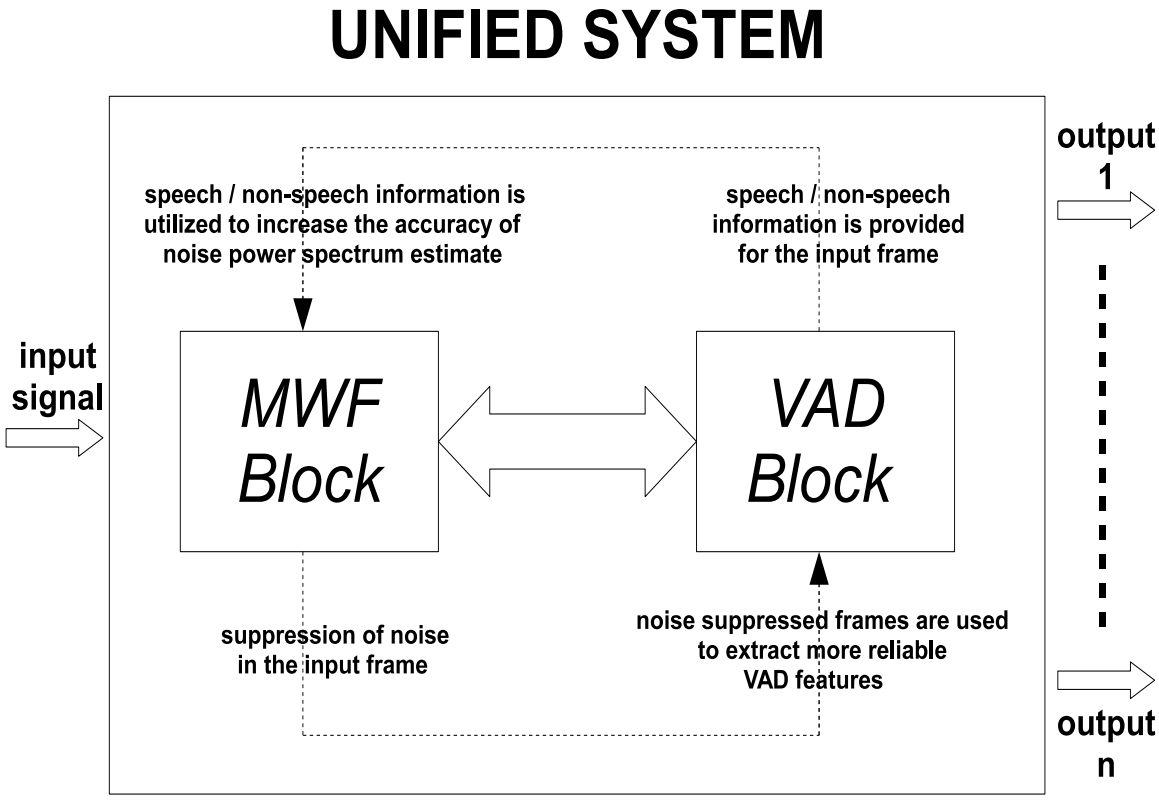

Figure 1.1. Unified system for VAD and speech enhancement

As can be seen in Figure 1.1, the proposed unified system for VAD and speech enhancement incorporates the capabilities of speech/non-speech detection and noise suppression in a single system. The input speech signal to the system is partitioned into multiple outputs where each output contains a distinct speech portion of the input signal (VAD functionality). The outputs of the system are comprised of noise suppressed speech (speech enhancement functionality).

1.2. Goals, Approach and Contributions

There are two major goals in this thesis. First, we would like to prove that VAD and speech enhancement problems are closely related. Our aim is to demonstrate that the utilization of information exchange between VAD and speech enhancement algorithms in a unified system improves the performance of both algorithms.

Second, we would like to formulate a new, robust and low complexity VAD algorithm to be used in our proposed system. Suitability for real-time operation is the most important requirement that is desired to be met in the newly proposed VAD algorithm. At this point, it should be stressed that we do not concentrate on implementing a VAD algorithm with superior speech/non-speech detection performance. For our proposed unified system, the error of treating non-speech as speech at the beginning or end is considered as less harmful than classifying speech frames as non-speech due to the information loss in the latter case. For this reason, our proposed VAD algorithm has a relaxed condition to find the exact talkspurt boundaries and small silence margins at speech boundaries are tolerated. In that respect, the newly proposed VAD algorithm is mostly suited for speech recognition applications.

Throughout the thesis, we describe the VAD and speech enhancement blocks of our proposed unified system separately. We demonstrate the performance improvements, which are obtained from the utilization of information exchange between the VAD and MWF algorithms. Performance improvement of VAD is verified on a huge amount of speech data recorded by various users in a car under different conditions. Objective and subjective quality measures of MWF and EMWF outputs are compared in order to demonstrate the increased performance of EMWF algorithm. The performance improvement of both algorithms are also verified on an in car speech recognition task.

The most significant contribution of this thesis is the demonstration of the increased performance obtained from the utilization of information exchange between VAD and speech enhancement algorithms in a unified system. This work demonstrates the performance improvements only for the new VAD and MWF algorithms. However, it is expected that such performance improvements can be demonstrated for any VAD and speech enhancement algorithms. Another contribution is the implementation of the new VAD algorithm. Although there exists a huge number of proposed VAD algorithms in literature, robustness to noise and low complexity properties of the new VAD algorithm represent an important contribution to the field of speech processing.

1.3. Thesis Outline

The outline of this thesis is as follows. In Chapter 2, the background material related with VAD and speech enhancement is presented. Fundamentals of human speech, human speech production system and noise characteristics are elaborated.

Chapter 3 describes VAD systems in detail. First, the fundamentals of VAD systems are presented. Second, a literature review of the various features used in different VAD algorithms is provided. Finally, the newly proposed VAD algorithm, which we use for the VAD block of the unified system, is explained in detail.

Chapter 4 is devoted to speech enhancement. First, the mathematical model of the speech enhancement problem is explained. Second, a literature review of single channel speech enhancement algorithms is provided. Finally, the EMWF algorithm, which we use for the speech enhancement block of the unified system, is described in detail.

In Chapter 5, the evaluations are presented. The benefits of utilizing mutual information exchange between the VAD and speech enhancement blocks of the proposed unified system are demonstrated.

Finally, a summary of the results obtained is given with related discussions and future work in Chapter 6.

2. BACKGROUND

In this chapter, we review general concepts that are related to VAD and speech enhancement and discuss their functionality in motivating the research in the subsequent chapters of this thesis. Discrimination between speech and noise is crucial for both VAD and speech enhancement algorithms. In order to provide a comparative analysis, this chapter presents the characteristics of speech and noise. Due to its role in determining speech characteristics, human speech production system is also explained.

2.1. Fundamentals of Human Speech

Speech signals are time varying pressure waves that are transmitted by a speaker in order to communicate information [8]. Speech signals are composed of a sequence of sounds. These sounds and the transitions between them serve as a symbolic representation of information [9]. In order to apply signal processing techniques for VAD and speech enhancement, it is essential to understand the speech production process and the structure of human speech.

2.1.1. Speech Production

The source of the human speech is the airstream produced by the lungs. The air flow produced by the lungs is perturbed by a constriction somewhere in the vocal tract. The air flow in the vocal tract changes the air pressure at the lip end. This, in turn, results in the radiation of acoustic waves. These radiated waves are perceived as speech by listeners. The organs in the vocal tract such as the teeth, tongue and etc. are referred as the articulators of vocal tract and they determine the type and place of constriction during speech production process. In the average male, the total length of the vocal tract is about 17 cm. The cross-sectional area of the vocal tract, determined by the positions of the articulators varies from zero (complete closure) to about 20 cm^2 [9]. Figure 2.1 shows a schematic diagram of the vocal tract.

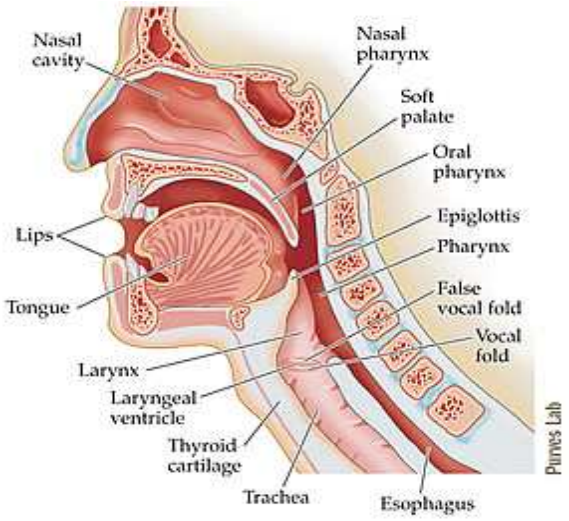

Figure 2.1. Vocal tract for human speech production [10]

Speech sounds can be classified into 3 distinct classes, namely voiced sounds, unvoiced sounds and plosive sounds, according to their mode of excitation [9]. *Voiced sounds* are produced by forcing air through the glottis with the tension of the vocal cords adjusted so that they vibrate in a relaxation oscillation. Vibrations of the vocal cords produce quasi-periodic pulses of air which excite the vocal tract. *Unvoiced sounds* are generated by forming a constriction at some point in the vocal tract (usually toward the mouth end), and forcing air through the constriction at a high enough velocity to produce turbulence. This creates a broad-spectrum noise source to excite the vocal tract. *Plosive sounds* result from making a complete closure (again, usually toward the mouth end), building up pressure behind the closure and abruptly releasing it.

Human speech production system can be modeled by the well known source/filter production model. Figure 2.2 depicts the source/filter production model. The model contains a time varying digital filter which is driven by an excitation function. The excitation function is a periodic impulse train with a period of the pitch period for voiced sounds. For unvoiced sounds, the excitation function is a random noise generator. The changes in the shape of vocal tract caused by articulators to produce different sounds are modeled by representing the vocal with a time varying digital filter. A variable gain factor determines the intensity of the produced speech.

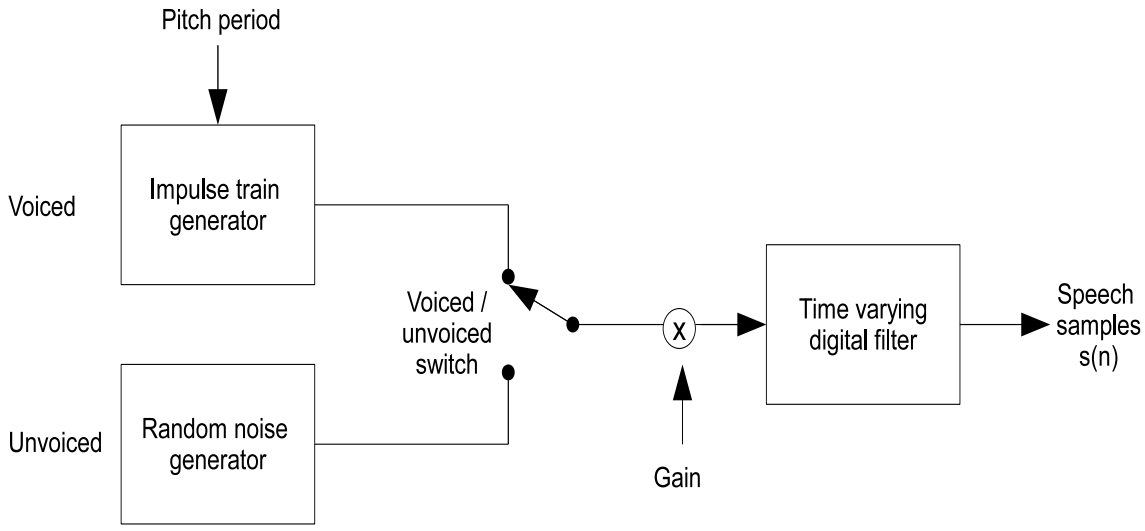

Figure 2.2. Source/filter model for speech production (adapted from [9])

2.1.2. Characteristics of Speech

The frequency spectrum of human speech is shaped by the frequency selectivity of the vocal tract. The periodic air flow associated with voiced sounds results in the formation of the resonant frequencies (formant frequencies in speech processing terminology) in human speech spectrum [9]. Peterson *et al.* [11] demonstrated that the first three formants of vowels in English are located at frequencies lower than 3 kHz. A similar study was done in Türk *et al.* [12] for Turkish vowels and this study indicated that the first three formants of Turkish vowels are also located at frequencies lower than 3 kHz. The first three formants contain much of the energy of voiced sounds and they are generally enough to characterize voiced speech.

Spectrogram plots are frequently used in the analysis of the time-varying characteristics of human speech. In a spectrogram plot, the horizontal dimension corresponds to time and the vertical dimension demonstrates the energy distribution over frequency. The darkness of the spectrogram plot is proportional to the signal energy [9]. In a typical spectrogram, voiced sounds are characterized by a striated appearance due to the periodicity of the time waveform. On the other hand, unvoiced sounds are more solidly filled in due to their broader spectrum [9]. These characteristics of human speech

sounds can be observed in the spectrogram analysis for the Turkish word “eczane” which is shown in Figure 2.3.

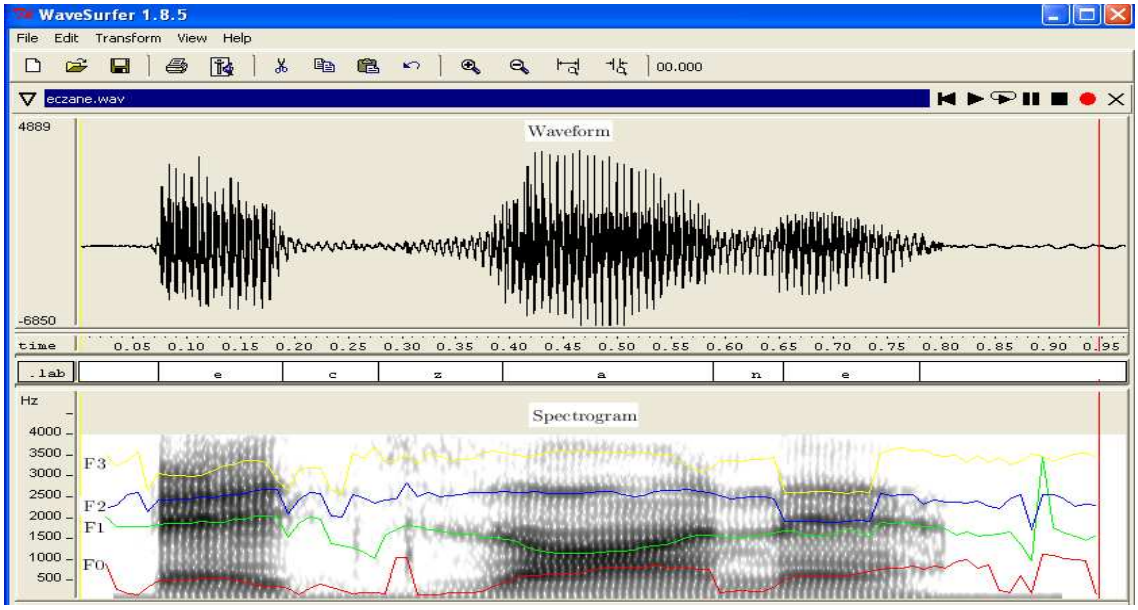

Figure 2.3. The spectrogram analysis of the Turkish word “eczane” spoken by a male speaker using the WaveSurfer tool [13]

A distinct property of average human vocal spectrum is the nonuniform distribution over the frequency range. Average human vocal spectrum contains more energy in low frequency bands. Schwartz *et al.* [14] demonstrated this property by investigating the statistical characteristics of spoken American English spectrum. They also showed that the statistical structure of vocal spectrum in Farsi, Mandarin Chinese, and Tamil are remarkably similar to American English.

A spectrum range up to 4 kHz contains much of the frequency content of average human speech. This range also contains the intelligible part. For these reasons, general purpose speech communication applications typically use a bandwidth of 4 kHz for the voice channel.

2.2. Characteristics of Noise

In general terms, noise refers to the unwanted random signal added to some desired signal. Noise added to human speech degrades the perceptual aspects of speech

such as quality or intelligibility [6]. Noise may be either correlated or uncorrelated with the actual speech signal. In this thesis, we will only be dealing with the additive noise problem, where the speech signal and noise are uncorrelated.

2.2.1. Statistical Properties of Noise

Since noise is a random phenomenon, noise suppression algorithms use various statistical properties of noise for analysis. The most important characteristics of noise for analysis are stationarity and spectral energy distribution over frequency.

In broadest terms, a random process is called *stationary* if its distribution functions or certain expected values are invariant with respect to a translation of the time axis. The degree of stationarity for a random process ranges from stationarity in strict sense to a less restrictive form of stationarity called wide-sense stationarity. Wide-sense stationarity for a real noise process $n(t)$, implies a constant mean ($E[n(t)] = \mu_n$) and an autocorrelation function that depends only on the time difference ($E[n(t)n(t+\tau)] = R_{nn}(\tau)$) [15]. Because of the variations in the statistics, suppression of nonstationary noise is often more difficult than that of suppressing stationary noise [6].

The average shape of the spectrum for noise, particularly the average distribution of the noise energy in the frequency domain, is another important statistics for various types of noise [6]. Figure 2.4 and Figure 2.5 show example time waveforms and corresponding long-term average spectra of car noise and train noise from the NOIZEUS corpus [16]. The differences between the two noise types are better displayed in the frequency domain rather than the time domain. As can be seen in the figures, car noise is relatively stationary compared to the train noise. Most of the energy of car noise is concentrated in the low frequencies whereas the train noise is more broadband as it occupies a wider frequency range [6].

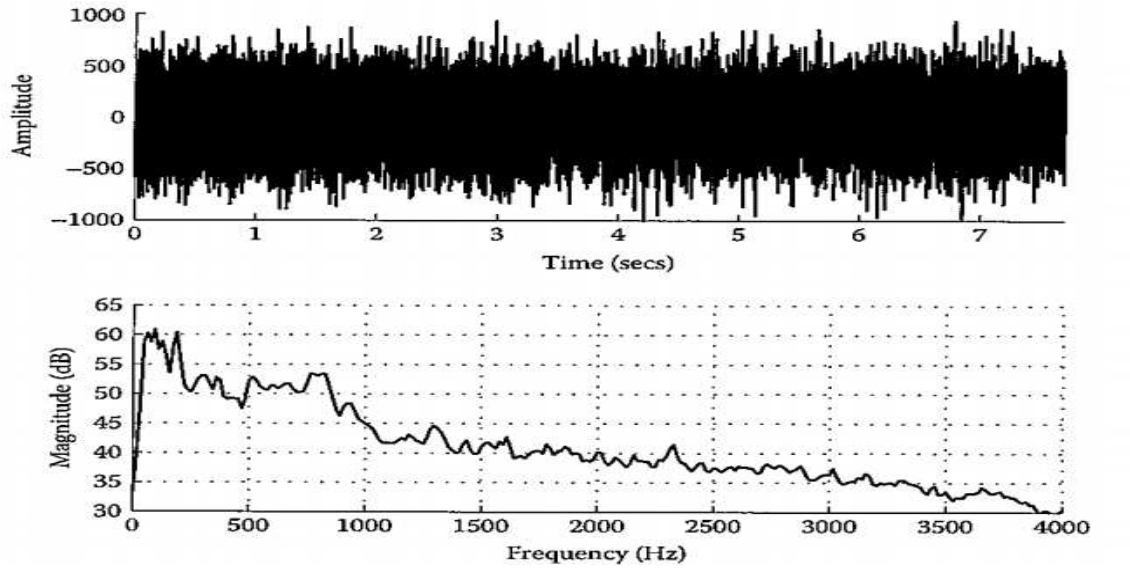

Figure 2.4. Example waveform of car noise and its long-term average spectrum [6]

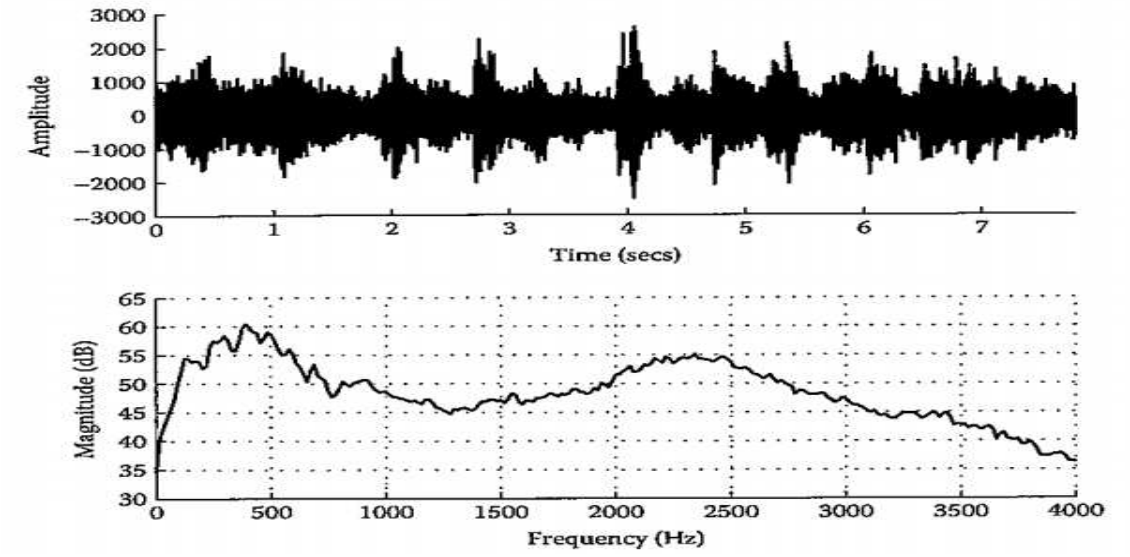

Figure 2.5. Example waveform of train noise and its long-term average spectrum [6]

3. VOICE ACTIVITY DETECTION

In this chapter we describe VAD systems in detail. First, the operating principles of VAD systems and the challenges with speech detection are investigated. Second, a literature review of the various features used in different VAD algorithms is provided. In order to demonstrate the usage of the features in algorithm implementations, some of the important VAD algorithms are also presented. Third, the newly proposed VAD algorithm is elaborated. The features and operating principles of the proposed VAD algorithm are described in detail. Finally, the Hybrid VAD algorithm that we use as the VAD block of the proposed unified system is explained.

3.1. Fundamentals of VAD Systems

The main objective of a VAD system is to correctly decide if a given audio signal portion is speech or non-speech. The task of the determination of presence of speech segments in a given signal can be considered as a statistical hypothesis problem for VAD systems where the challenge is the determination to which category (either speech or non-speech) the given signal belongs. A typical VAD system segments the input audio signal into frames of smaller length. The frame length is generally chosen according to the real-time constraints of the system and it may be different for each application. Several computations are performed in order to extract various features from the input frame. These features form an observation vector for the current frame. This observation vector serves as the input to a decision rule that makes the final speech or non-speech decision. Operation of a typical VAD system is schematically shown in Figure 3.1.

As indicated in Chang *et al.* [17], earlier algorithms for VAD were mostly based on the features like linear prediction coding (LPC) parameters [18, 19], energy levels, formant shape [20], zero-crossing rate (ZCR) [21], cepstral coefficients [22], and the periodicity measure [23]. More recently, VAD approaches based on a pattern recognition [24] and higher order cumulants of the LPC residual have been presented as new

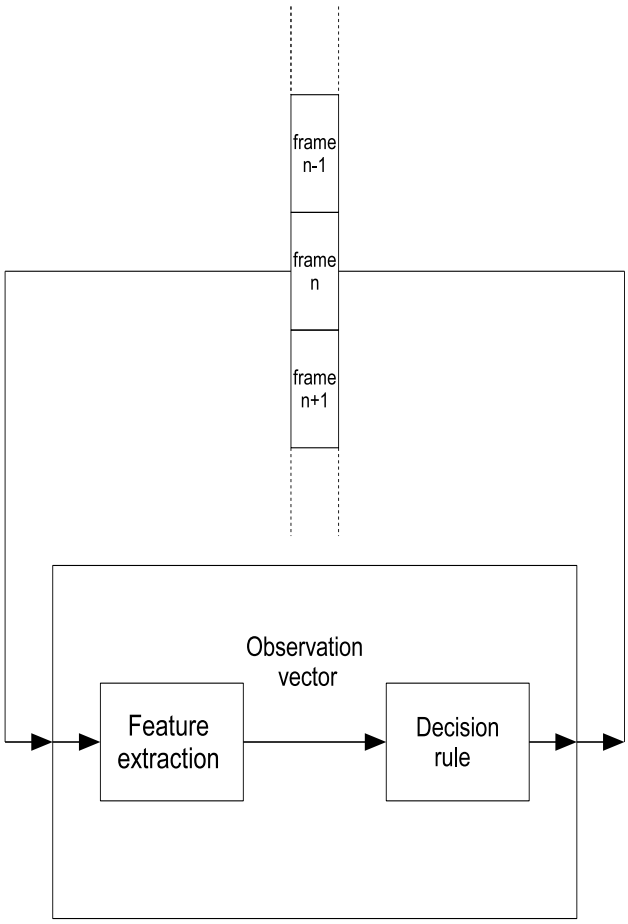

Figure 3.1. Operation of a typical VAD system

strategies [25].

3.1.1. Problems with VAD

In order to develop a robust VAD algorithm, it is important to point out the challenges with speech detection. Rapidly changing noise environments and low SNR conditions degrade the performance of VAD systems severely. Under low SNR conditions, VAD systems suffer from false triggering. High level noise may cause the VAD algorithm to misclassify noise frames as speech frames. High level noise also avoids the detection of speech frames that contain unvoiced sounds. Since unvoiced sounds have broad spectra, their energy and spectral characteristics are similar to that of noise. Because of these reasons, correct differentiation between unvoiced sounds and noise may not always be possible.

Another problem for VAD systems originates from the nature of sounds. It is generally difficult to locate the beginning and end of an utterance if there are : a) weak fricatives at the beginning or end, b) weak plosive bursts at the beginning or end, c) nasals at the end, d) voiced fricatives that become devoiced at the end of words, and e) trailing off of vowel sounds at the end of an utterance [9]. Reduced SNR conditions increase these difficulties.

3.2. Literature Review of the Features Used for VAD Systems

Performance of a VAD algorithm relies on the discriminative quality of the extracted features. VAD algorithms generally use a hybrid combination of more than one feature to make the final speech/non-speech decision. This section is devoted to the literature review of the features proposed for VAD systems.

3.2.1. Linear Predictive Coding

Linear Predictive Coding (LPC) is a commonly used speech analysis technique for representing speech signals by a finite number of LPC coefficients. A P^{th} order linear predictor $\tilde{s}(n)$, for a speech frame $s(n)$ of length N, is defined by the difference equation of

$$\tilde{s}(n) = \sum_{k=1}^{P} \alpha_k s(n-k) \tag{3.1}$$

where α_k are the prediction coefficients [9]. The prediction error, $e(n)$, is defined as

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{P} \alpha_k s(n-k) \tag{3.2}$$

The prediction coefficients are chosen to minimize the mean squared error, E, defined by

$$E = \sum_{n=1}^{N} \big(s(n) - \tilde{s}(n)\big)^2 \tag{3.3}$$

and are determined by the following set of equations

$$\begin{bmatrix} R_{ss}(0) & R_{ss}(1) & \dots & R_{ss}(P-1) \\ R_{ss}(1) & R_{ss}(0) & \dots & R_{ss}(P-2) \\ \vdots & \vdots & \dots & \vdots \\ R_{ss}(P-1) & R_{ss}(P-2) & \dots & R_{ss}(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_P \end{bmatrix} = \begin{bmatrix} R_{ss}(1) \\ R_{ss}(2) \\ \vdots \\ R_{ss}(P) \end{bmatrix} \tag{3.4}$$

where $R_{ss}(k)$ is the k^{th} autocorrelation lag for $s(n)$.

By training the VAD algorithm, Rabiner *et al.* [18, 19] obtained a spectral characterization for voiced, unvoiced and silence classes using the LPC coefficients. Average energy measures for each of the 3 classes are also calculated during the training period. During its operation, algorithm computes the LPC distance metric proposed by Itakura *et al.* [26] and energy distance metric for the processed frames. The calculated metrics are nonlinearly combined to make final speech/non-speech discrimination.

3.2.2. Energy Feature

Energy feature is probably the simplest feature for voice activity detection. It is used as a supplementary feature in most of the VAD systems. For a speech frame $s(n)$ of length N, the short time energy of the signal E_s is defined as

$$E_s = \sum_{n=1}^{N} |s(n)|^2 \tag{3.5}$$

VAD algorithms that utilize the energy feature generally determine an energy threshold for speech character. If the energy of a frame is greater than that threshold, the

frame is considered as speech, otherwise it is considered as non-speech. The update of the threshold value during non-speech frames generally increases the algorithm performance.

Energy feature can be used on its own to distinguish speech from silence for very high quality audio signals (high SNR conditions). Typically, energy values for unvoiced segments are smaller than for voiced segments. Because of this, energy feature can also be used to locate approximately the time instant at which voiced speech becomes unvoiced and vice versa [9].

In order to improve the reliability of the energy feature, several modifications may be done on its computation. One such modification is performed by Ye *et al.* [27]. Their proposed VAD algorithm relies on the fusion of the energy values for selected sub-bands of the input speech. This fusion method is implemented through a specific function called SAF (sum of activation function). Their results indicate that the subband approach may give reliable voice activity detection results in low SNR conditions and even in the presence of non-stationary noise.

3.2.3. Zero-Crossing Rate

For discrete time signals, a zero-crossing occurs when two successive samples of the signal have different algebraic signs. Zero-crossing rate is a measure of the frequency content, particularly for narrowband signals [9].

The zero-crossing rate, Z, for a speech frame $s(n)$ of length N, is computed as

$$Z = \sum_{n=2}^{N} \mid sgn[s(n)] - sgn[s(n-1)] \mid \tag{3.6}$$

where

$$\begin{aligned} sgn[s(n)] &= 1 \quad \text{if } s(n) \geq 0 \\ &= -1 \quad \text{if } s(n) < 0 \end{aligned} \tag{3.7}$$

As mentioned previously, the energy content of voiced speech is concentrated below about 3 kHz, whereas for unvoiced speech most of the energy is found at higher frequencies. Since high frequencies imply higher zero-crossing rates, there is a strong correlation between zero-crossing rate and energy distribution with frequency. It is reasonable to assume that if the zero-crossing rate is high, the speech signal is unvoiced, and if the zero-crossing rate is low, the speech signal is voiced [9].

Zero-crossing rate representations can be combined with other features to implement a robust VAD algorithm. Rabiner *et al.* [28] combined the energy and zero-crossing rate features for locating the beginning and end of a speech signal in the context of an isolated-word speech recognition system.

3.2.4. Formant Shape

The patterns associated with resonant frequencies (formant frequencies) in human speech spectrum are another feature for VAD algorithms. Hoyt *et al.* [20] examined a speech corpus that consists of 630 different speakers which are divided approximately evenly between male and female, and between the eight major dialects of American English. Their noise corpus includes data from compact disks and recordings from the reception of commercial FM radio stations. Upon examination of the two signal classes, they observed that all speech signals provided convex and concave formant shapes in frequency bands between 400 Hz and 4 kHz. These observations led to the development of the formant tracking algorithm for voice activity detection. By using a type of chain code algorithm [29], their proposed VAD algorithm searches the peak data from the smoothed spectrograms (obtained from the DFT of the 20 pole linear predictive coding) for convex, concave and straight sections. These shapes must meet

a minimum length (duration) criteria, or they are discarded as noise. The minimum length of each shape section (concave, convex, or straight) was experimentally varied to maximize the detection of speech. Other parameters that needed adjustment include; a maximum length for each shape section in a formant (to prevent false positives for such sounds as a police siren); a maximum difference between peaks in adjacent frames that will be included in the same formant; and a minimum length (time duration) for a formant.

3.2.5. Cepstral Coefficients

Cepstral analysis can be considered as an attempt to de-convolve the speech signal into its excitation component and a "vocal apparatus" component, via homomorphic filtering [22]. In the late 1970s, coefficients derived from cepstrum began replacing the LPC coefficients as the basic parameter set for speech recognition applications due to the superior performance [30].

Haigh *et al.* [22] investigated the accurate modeling of the slowly varying parts of speech, as provided by the cepstra, for the discrimination of speech and non-speech. Their algorithm relies on the differences in the cepstra computed from noise and speech. The discrimination measure that they utilize is the weighted Euclidean distance, d, between a frame and a non-speech frame

$$d = \frac{1}{P}\sum_{i}^{P}(c_i - c'_i)^2 \tag{3.8}$$

where P is the order of the cepstral analysis (10 in their computations), and c_i and c'_i are the i^{th} elements of two cepstral vectors. The initial frames of the recordings are used to compute the cepstral coefficients for the non-speech signal class. The increase of the distance above a threshold indicates the presence of speech in the current frame. Figure 3.2 depicts the increased cepstral distance for the speech segments of an audio signal.

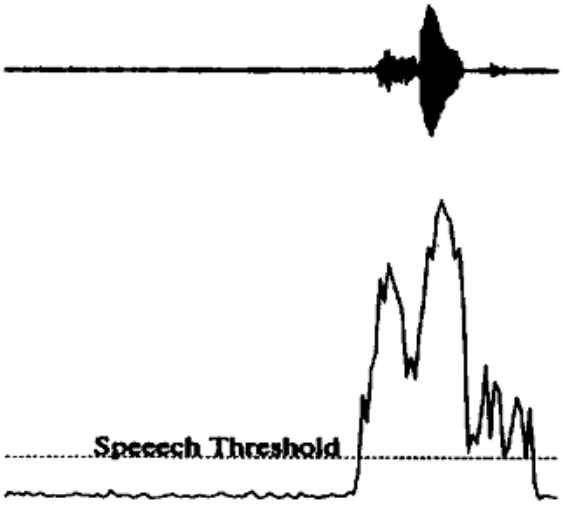

Figure 3.2. Time waveform and corresponding cepstral distance for an utterance of "sheep" [22]

3.2.6. Periodicity Measure

Voiced sounds in human speech are generated by the periodic vibration of vocal cords. Moreover, most of the noise types, either stationary or nonstationary, are not periodic. For these reasons, periodicity in the input audio frame is an indication of speech character rather than the non-speech character.

Among several methods for measuring the periodicity of speech corrupted by noise, the autocorrelation function is the simplest. Tucker *et al.* [23] proposed the application of a least-squares periodicity estimator (LSPE) [31] directly to the signal for the speech/non-speech discrimination. The LSPE-based VAD also includes an energy detector in order to prevent the detection of very low-level signals in the presence of larger signals. The performance of LSPE-based VAD is shown to be about 5 dB better than that of the autocorrelation function. However, LSPE approach has the disadvantage of increased computational complexity compared to the autocorrelation function.

3.3. The Newly Proposed VAD Algorithm

Computational speed and robustness to noise are the most important criteria for the newly proposed VAD algorithm. In order to be attractive for real-time operation, processing of the audio input signal should not take too long. The newly proposed VAD algorithm is also desired to be robust against a wide range of SNRs and various types of noise.

As discussed in Section 3.2, there are many features that a VAD algorithm can utilize to make the final speech/non-speech decision. Increasing the number of extracted features provides the algorithm with more information. This, in turn, enables better speech/non-speech characterization. However, the inclusion of every separate feature brings about the additional cost of computation. In order to enable real-time operation, the total number of extracted features in the proposed VAD algorithm is limited.

Most of the VAD algorithms in the literature are concerned with the speech/non-speech decision on a frame by frame basis. This approach may be beneficial for performance evaluation of the implemented algorithm. However, frame-only decisions generally have no use for physical systems. Generally, physical systems that require VAD need the identification of the beginning and end of the utterances with continuity. For example, automatic speech recognition systems often make use of VAD as a preprocessor. They need the actual utterances from beginning to end. For proper operation, the small silence regions between the phonemes must also be interpreted as voice activity. This is generally not the case for the VAD approaches that make the final decision just on a frame by frame basis. The proposed VAD algorithm emphasizes continuity in voice activity. Small silence regions between the phonemes are interpreted as voice activity and actual utterances of the audio input are determined from beginning to end.

Another distinctive property of the proposed VAD is the relaxation on the condition to find the exact talkspurt boundaries. For physical systems that require VAD,

the error of treating non-speech as speech at the beginning or end is generally less harmful than classifying speech frames as non-speech. Although the error of including silence at the boundaries of the utterances can generally be tolerated to some extent in practical systems that use VAD, such a tolerance does not exist for the absence of the actual speech frames. This situation can be exemplified by a communication device that relies on VAD decisions to send signals to the network for voice communication. On the receiving end, the listener can understand the utterance even if there is some silence included with the utterance at the boundaries. However, the context may completely be lost if some part of the utterance is lost at the beginning or end. The situation is the same for the automatic speech recognition systems that use VAD as a preprocessor. For example, if the /s/ sound at the beginning of the word "smile" is not detected by the preprocessing VAD block, it is nearly impossible for the automatic speech recognition system to distinguish the actual word "smile" from the word "mile".

The features used for the proposed VAD algorithm are carefully chosen according to their computational complexity and their discriminative contribution to the ultimate speech/non-speech decision in order to satisfy the above-mentioned requirements. The proposed VAD implements frame by frame processing and features are extracted per each frame. Periodicity measure and an energy measure obtained from spectral energy distribution and spectral energy difference are used for feature extraction. The next section describes the features utilized in the proposed VAD algorithm.

3.3.1. Periodicity Measure

Because of its discriminative property and ease of computation, autocorrelation method is used for measuring the periodicity of input frames in the proposed VAD algorithm. The k^{th} autocorrelation lag of a deterministic speech signal $s(n)$ of length N is defined by

$$R_{ss}(k) = \sum_{n=1}^{N} s(n)s(n+k) \tag{3.9}$$

The autocorrelation of a periodic function is also periodic with the same period. I.e., if the speech signal $s(n)$ is periodic with a period of p samples, it is easy to show that

$$R_{ss}(p) = R_{ss}(k+p) \tag{3.10}$$

The periodicity measure used for the proposed VAD algorithm is the probability of voicing parameter that is defined by the ratio of the peak autocorrelation lag to the signal energy (zeroth autocorrelation lag) [30]. The value of this parameter ranges from 0 to 1 and it increases as the periodicity increases. Typical fundamental frequency range of human speech tend to range from 200 Hz to 125 Hz, so the peak autocorrelation lag is searched in a range between 3 ms (333 Hz) and 18 ms (55.5 Hz).

One problem associated with this measure could be the assignment of high probability of voicing values for frames that contain periodic noise. However, periods of most periodic noise types are different than fundamental period of human speech. The constriction of searching the peak autocorrelation lag in 3 ms to 18 ms range avoids high probability of voicing values for frames that contain periodic noise. Moreover, expected value of the probability of voicing parameter for speech frames typically ranges from 0.3 to 0.7. For that reason, frames that have a periodicity greater than 0.8 are considered as noise.

3.3.2. Energy Measure

Energy measure is the second feature utilized by the proposed VAD algorithm. In order to increase the reliability of the feature, statistical properties of the average human vocal spectrum are elaborated. Energy parameter computed for a frame is the weighted sum of two measures, namely the spectrally weighted energy measure and the spectrally weighted energy difference measure. Adaptive threshold values are used during the computation of the energy parameter. This enables more robust operation under varying SNR conditions and different noise types. The next section describes the measures that are used in the computation of the energy parameter for a frame.

3.3.2.1. Spectrally Weighted Energy Measure. The average spectral distribution of human speech over frequency is nonuniform. For speech frames, lower frequency bands statistically contain more energy. The proposed VAD algorithm utilizes this fact by extracting a spectrally weighted energy measure of the input frame. In order to compute the spectrally weighted energy measure of the input frame, energy of the frame is computed in 5 equal length frequency subbands. The energy content of the frame is considered to be located in 300 Hz to 3400 Hz range (I.e. each subband has a length of 620 Hz). The motivation for 300 Hz to 3400 Hz range is the filtering approach used in analog speech communications. In analog telephone communications, the speech signal is sent through a band-pass filter with cut off frequencies of 300 Hz and 3400 Hz in order to eliminate noise from the voice signal [32].

The weighting coefficients are chosen to maximize the calculated energy parameter with the constraint of having a total sum of *1* for the coefficients. After simulations, it is concluded that the application of the coefficients of *0.3, 0.35, 0.2, 0.1, 0.05* to successive subbands maximizes the calculated energy for speech frames. Weighting coefficients of subbands for spectrally weighted energy parameter, *m_energyWeighted* in our computations, is shown in Figure 3.3. The algorithm used to compute *m_energyWeighted* is shown in Figure 3.4.

3.3.2.2. Spectrally Weighted Energy Difference Measure. Since the average energy distribution over the frequency is not uniform for human speech and typically three to four formants are present in the speech signal for a frequency range of 0-4 kHz, large energy differences over 0-4 kHz frequency range are supposed to be an indication of the presence of actual speech in the input signal.

The proposed VAD algorithm calculates a spectrally weighted energy difference parameter to evaluate the speech likeliness of the input frame. The same subbands that are used in the computation of the spectrally weighted energy measure, are also utilized for the calculation of the spectrally weighted energy difference parameter. First, an average energy value for each of the 5 subbands is calculated for the current frame.

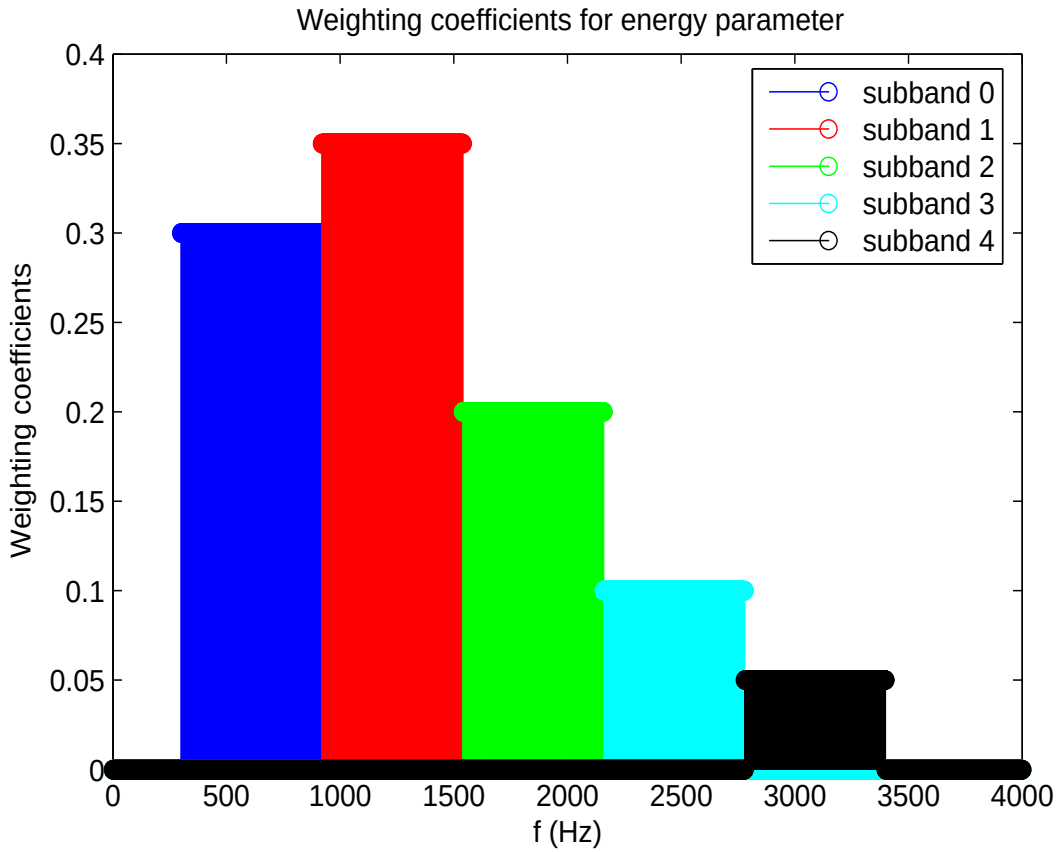

Figure 3.3. Weighting coefficients of subbands for spectrally weighted energy parameter computation

If it is the first frame, the average energy is simply set to the total energy of that subband. If it is not the first frame, 90 per cent of the previous average energy is summed with 10 per cent of the current total subband energy to calculate the average energy. The difference between the average subband energy and the total subband energy gives the subband energy difference value for that subband. Lastly, heuristically determined weighting coefficients are applied to the energy difference values calculated for each subband. The sum of these weighted energy difference values gives the overall spectrally weighted energy difference parameter.

The weighting coefficients are chosen to maximize the calculated energy difference parameter with the constraint of having a total sum of *5* for the coefficients. After simulations, it is concluded that the application of the coefficients of *1.0, 1.5, 1.0, 0.75, 0.75* to successive subbands maximizes the calculated energy difference for

Algorithm to compute spectrally weighted energy parameter, *m_energyWeighted*

***i*)** Compute subband energies in five bands

$$m\_subBandEnergy[i] \text{ for } i = 0 \ldots 4$$

***ii*)** Compute weighted sum of subband energies

$$\begin{aligned} m\_energyWeighted &= 0.30 * m\_subBandEnergy[0] + 0.35 * m\_subBandEnergy[1] \\ &\quad +0.20 * m\_subBandEnergy[2] + 0.10 * m\_subBandEnergy[3] \\ &\quad +0.05 * m\_subBandEnergy[4] \end{aligned}$$

Figure 3.4. Algorithm to compute the spectrally weighted energy parameter, *m_energyWeighted*

speech frames. Weighting coefficients of subbands for spectrally weighted energy difference parameter, *m_energyDifference* in our computations, is shown in Figure 3.5. The algorithm used to compute *m_energyDifference* is shown in Figure 3.6.

3.3.3. Soft Decision Assignment and Decision Smoothing

After the features are extracted from the input frame, the proposed VAD associates a soft decision value, rather than a strict speech/non-speech decision, with the frame. Soft decision value assigned to a frame is nonnegative and increases as the speech likeliness of the frame increases.

Final speech/non-speech decision for the processed frame is based on the history of soft decisions for the last 20 frames. The reason for this approach is the fact that human speech may show very large energy level variations even in very short time intervals. There may be small silence regions between the phonemes. Frames corresponding to these regions must be evaluated as speech. However, a speech/non-speech decision based only on the current frame can not evaluate the frame as speech in that situation. The history approach also avoids the determination of clicks as actual speech. For these reasons, the history of soft decisions is used to make the final speech/non-speech

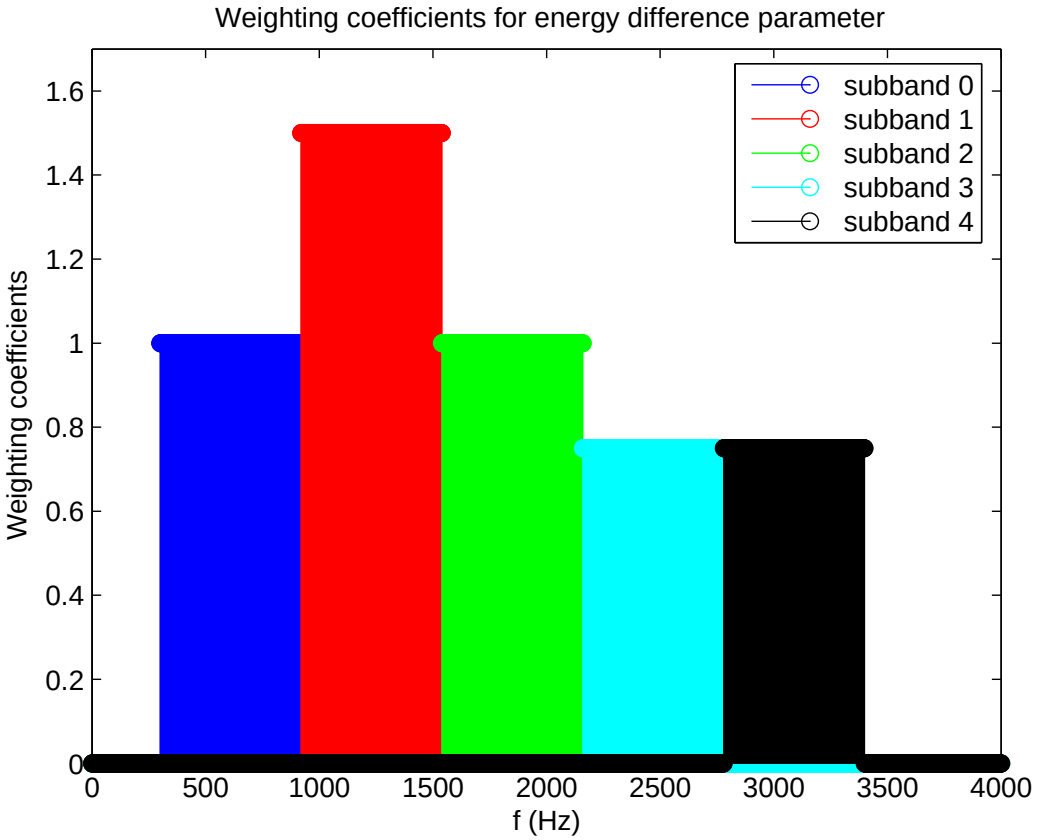

Figure 3.5. Weighting coefficients of subbands for spectrally weighted energy difference parameter computation

decision.

A problem with the 20 frame history approach would be the delay in detecting the start of the speech. We use a skip length of 10 ms for the proposed VAD algorithm, so delays around 200 ms are possible. Smaller values for the window length of the soft decision history are likely to decrease the delay of the algorithm. After running test simulations, smaller values for the window length of the soft decision history were indeed found to decrease the delay in detecting the start of the speech. But in that case, we had the problem of losing the continuity in speech. Frames corresponding to the small silence regions between the phonemes or words were evaluated as non-speech, a property that we wanted to avoid. We also had the problem of misclassifying the high energy noise frames as speech when the window length of the soft decision history was shortened. For these reasons, the optimal value of the window length of the soft

Algorithm to compute spectrally weighted energy difference parameter, *m_energyDifference*

i) Compute subband average energy for each band, *m_subBandAvg[i]* for $i = 0 \ldots 4$

if input frame is the first frame **then**

$$m\_subBandAvg[i] = m\_subBandEnergy[i]$$

end if

else

$$m\_subBandAvg[i] = m\_subBandAvg[i] * 0.9 + m\_subbandEnergy[i] * 0.1$$

end else

ii) Compute subband energy difference for each band, *m_subBandEnergyDifference* for $i = 0 \ldots 4$

$$m\_subBandEnergyDifference[i] = m\_subBandEnergy[i] - m\_subBandAvg[i]$$

iii) Compute weighted sum of subband energy differences

$$\begin{aligned} m\_energyDifference \;=\; & 1.00 * m\_subBandEnergyDifference[0] \\ & +1.50 * m\_subBandEnergyDifference[1] \\ & +1.00 * m\_subBandEnergyDifference[2] \\ & +0.75 * m\_subBandEnergyDifference[3] \\ & +0.75 * m\_subBandEnergyDifference[4] \end{aligned}$$

Figure 3.6. Algorithm to compute the spectrally weighted energy difference parameter, *m_energyDifference*

decision history was found to be 20. As we will explain in the next section, the use of the configurable *prespeech buffer* has the effect of decreasing the delay in detecting the start of the speech.

The proposed VAD implements decision smoothing for better speech/non-speech characterization. The implemented decision smoothing approach relies on the heuristic rule of increasing the soft decision value for the current frame if the previous frame is

tagged as a speech frame.

3.3.4. Operation of the Proposed VAD

The proposed VAD system is implemented using C++ programming language. The system operates on the audio input signal and separates the input into numerous portions, where each portion contains actual speech in a distinct output (I.e. the non-speech portions between the utterances are removed). A circular speech buffer continuously reads the input speech signal frame by frame. A frame length of 20 ms and a skip length of 10 ms is used for the system.

Based on the speech/non-speech decisions made for the frames, the system is either in a *speech started* or *speech ended* state. The system is initially at *speech ended* state. Detection of speech in the current frame makes the VAD system to switch its state from *speech ended* to *speech started*. This in turn triggers the system to start writing the input signal into a separate output stream. After a while, when the utterance ends, detection of non-speech in the incoming frames triggers the VAD system to switch to the *speech ended* state and this stops the writing operation. Transitions between the two states enable the partitioning of the input signal into separate outputs and these transitions continue until the whole audio input stream is processed by the system.

As previously mentioned, the proposed VAD has a relaxed condition to find the exact talkspurt boundaries. In order to prevent the loss of sounds at the beginning or end, the error of treating non-speech as speech at the boundaries is justified to some extent. For this reason, the proposed VAD algorithm uses configurable *prespeech buffer* and *postspeech buffer* values. *Prespeech buffer* and *postspeech buffer* have default values of 200 ms and 250 ms, respectively. They extend the duration of speech decisions and enable the detection of actual speech instances that are very much likely to be missed by the VAD algorithm.

Other than *prespeech buffer* and *postspeech buffer*, there are 3 additional config-

urable parameters, namely *speech trigger*, *silence trigger* and *sensitivity*, that are used to adjust the sensitivity of the proposed VAD. *Speech trigger* has a default value of 8 and it determines the threshold for speech started decision. When the sum of the objective soft decision values of the last 20 frames exceeds this value, the current frame is considered as actual speech instance and the system transitions to *speech started* state (If the current state is already *speech started*, the current state is preserved). *Silence trigger* has a default value of 700 ms and it determines the total required signal length to trigger speech ended decision. The system keeps track of the total length of successive frames, where the sum of the objective soft decision values in the previous 20 frames is smaller than the value of *speech trigger* for each frame. When this length exceeds the *silence trigger*, current state of the system is set to *speech ended*. *Sensitivity* has a default value of 3 and it is restricted to be between 0 and 12. A decreased *sensitivity* value increases the value of soft decisions associated to the frame and this increases the number of speech started decisions. Under low SNR conditions, lower *sensitivity* values must be used to avoid false triggering.

The default values for the configurable parameters used for the proposed VAD algorithm are aimed to be optimized for a wide range of audio signals. They are finalized after extensive number of observations. If the user has apriori knowledge about the characteristics of the audio signal to be processed, these parameters may be altered for better performance. Figure 3.7 shows the screenshots of the graphical user interface of the proposed VAD system and the output file structure after a sample run.

3.3.5. Step-by-step Algorithm Description of the Proposed VAD

The flowchart of the proposed VAD algorithm is shown in Figure 3.8. By making use of the final speech/non-speech decisions, the proposed VAD system partitions the audio input into several outputs, where each output contains a continuous utterance section of the audio input.

Step-by-step algorithm description and the operational details of the proposed VAD system are as follows:

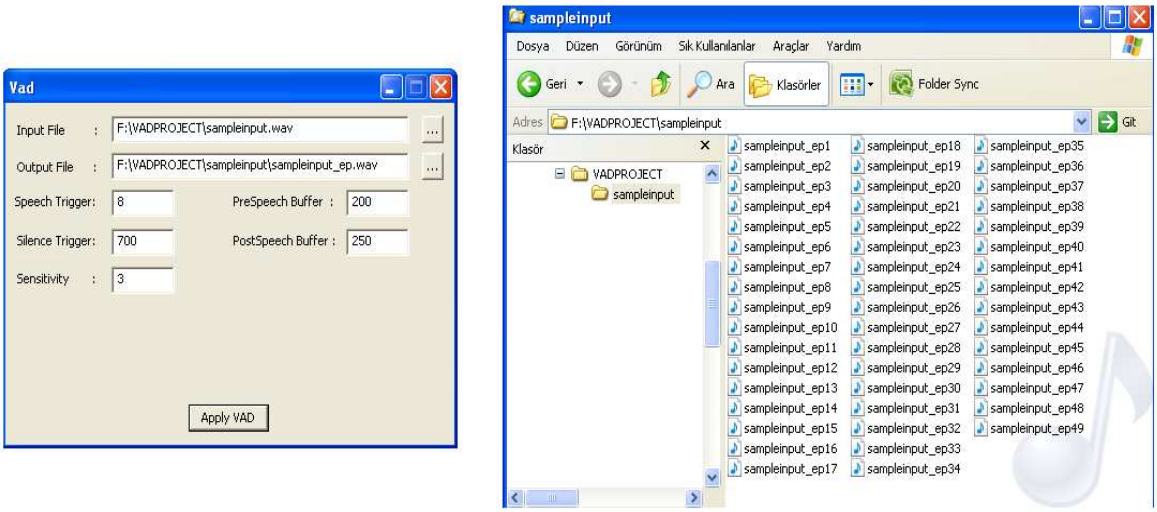

Figure 3.7. Screenshots of the graphical user interface of the proposed VAD system and the output file structure after a sample run

• *STEP 1)* A circular speech buffer of size of 5 seconds starts to read the audio input. The circular speech buffer normalizes the audio input for processing. It determines the sample rate, number of bits per sample and encoding type of the audio input.

• *STEP 2)* A frame of speech data with a frame length of 20 ms and a skip length of 10 ms is taken from the speech buffer. This data corresponds to the current frame that will be processed by the system.

• *STEP 3)* The largest autocorrelation peak of the input frame in 3-18 ms range is determined. The probability of voicing parameter, *prob_voice*, is calculated by dividing the autocorrelation peak by the frame energy (zeroth autocorrelation lag).

• *STEP 4)* Power spectrum of the current frame is computed by taking the DFT of the frame. Frequency range from 300Hz to 3400 Hz is divided into 5 equal length frequency subbands. The logarithm of the total power in each frequency subband, *m_subBandEnergy[i]* for $i = 0 \ldots 4$, is calculated. The calculated log-power values are interpreted as the energy measure of that subband.

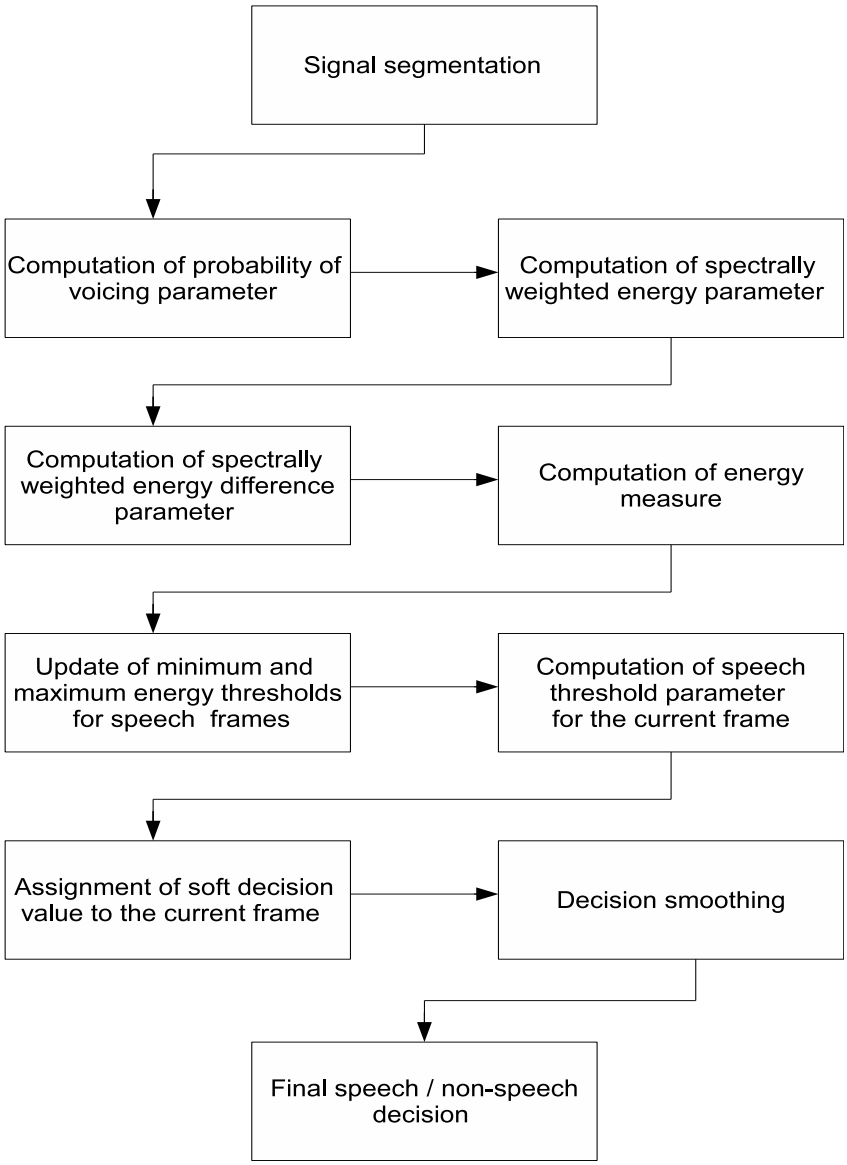

Figure 3.8. Flowchart for the proposed VAD algorithm

• *STEP 5)* An average energy value for each subband, *m_subBandAvg[i]*, is computed. If it is the first frame, *m_subBandAvg[i]* is calculated as

$$m\_subBandAvg[i] = m\_subBandEnergy[i] \tag{3.11}$$

for $i = 0 \ldots 4$. If it is not the first frame, *m_subBandAvg[i]* is determined as

$$m\_subBandAvg[i] = 0.9 * m\_subBandAvg[i] + 0.1 * m\_subBandEnergy[i] \tag{3.12}$$

for $i = 0 \ldots 4$.

• *STEP 6)* Energy difference value for each subband, *m_subBandEnergyDifference[i]* for $i = 0 \ldots 4$, is calculated as

$$m\_subBandEnergyDifference[i] = m\_subBandEnergy[i] - m\_subBandAvg[i] \tag{3.13}$$

• *STEP 7)* A spectrally weighted energy parameter, *m_energyWeighted*, for the current frame is calculated as

$$\begin{aligned} m\_energyWeighted &= 0.30 * m\_subBandEnergy[0] + 0.35 * m\_subBandEnergy[1] \\ &\quad +0.20 * m\_subBandEnergy[2] + 0.10 * m\_subBandEnergy[3] \\ &\quad +0.05 * m\_subBandEnergy[4] \end{aligned} \tag{3.14}$$

• *STEP 8)* A spectrally weighted energy difference parameter, *m_energyDifference*, for the current frame is calculated as

$$\begin{aligned} m\_energyDifference &= 1.00 * m\_subBandEnergyDifference[0] \\ &\quad +1.50 * m\_subBandEnergyDifference[1] \\ &\quad +1.00 * m\_subBandEnergyDifference[2] \\ &\quad +0.75 * m\_subBandEnergyDifference[3] \\ &\quad +0.75 * m\_subBandEnergyDifference[4] \end{aligned} \tag{3.15}$$

• *STEP 9)* The energy measure for the current frame, *m_energy*, is computed. It is the sum of 1.10 of the spectrally weighted energy parameter, 0.25 of the spectrally weighted energy difference parameter and the probability of voicing parameter. During summation, upper bounds are set in order to guarantee that the contribution of spectrally weighted energy difference and probability of voicing parameters are below

0.5 and 1.0, respectively. The energy measure is computed as

$$\begin{aligned} m\_energy &= \Big(1.10 * m\_energyWeighted\Big) \\ &+\Big(0.25 * min(m\_energyDifference, 2)\Big) \\ &+\Big(min(1.0, 0.5 * prob\_voice)\Big) \end{aligned} \tag{3.16}$$

where $min(x, y)$ returns the minimum of x and y.

• *STEP 10)* A speech threshold value for the current frame, *speech_threshold*, is calculated. Computation of the speech threshold relies on expected thresholds of maximum and minimum energy levels for a speech frame.

The proposed VAD algorithm keeps track of the expected maximum and minimum energy level thresholds, *m_maxEnergy* and *m_minEnergy*, for a speech frame. Initial values for *m_maxEnergy* and *m_minEnergy* are 2.5 and 5.8, respectively. In order to provide robustness against varying SNR conditions, the thresholds are adaptively updated.

Computation of *speech_threshold* depends on the current state of the VAD system. If the current state is speech ended (meaning that we are operating on non-speech region) the value of *speech_threshold* is calculated as

$$\begin{aligned} speech\_threshold &= \Big[0.01 * \big(m\_maxEnergy - m\_minEnergy\big) * \\ &\big(40 + 5 * (10 - sensitivity)\big)\Big] + m\_minEnergy \end{aligned} \tag{3.17}$$

Conversely, if the current state is speech started (meaning that we are operating on speech region), the value of *speech_threshold* is set to the 0.4 less of the value computed by Eq. 3.17. This is done in order to detect speech end more robustly.

The flowchart of the algorithm to compute *speech_threshold* is shown in Figure

3.9.

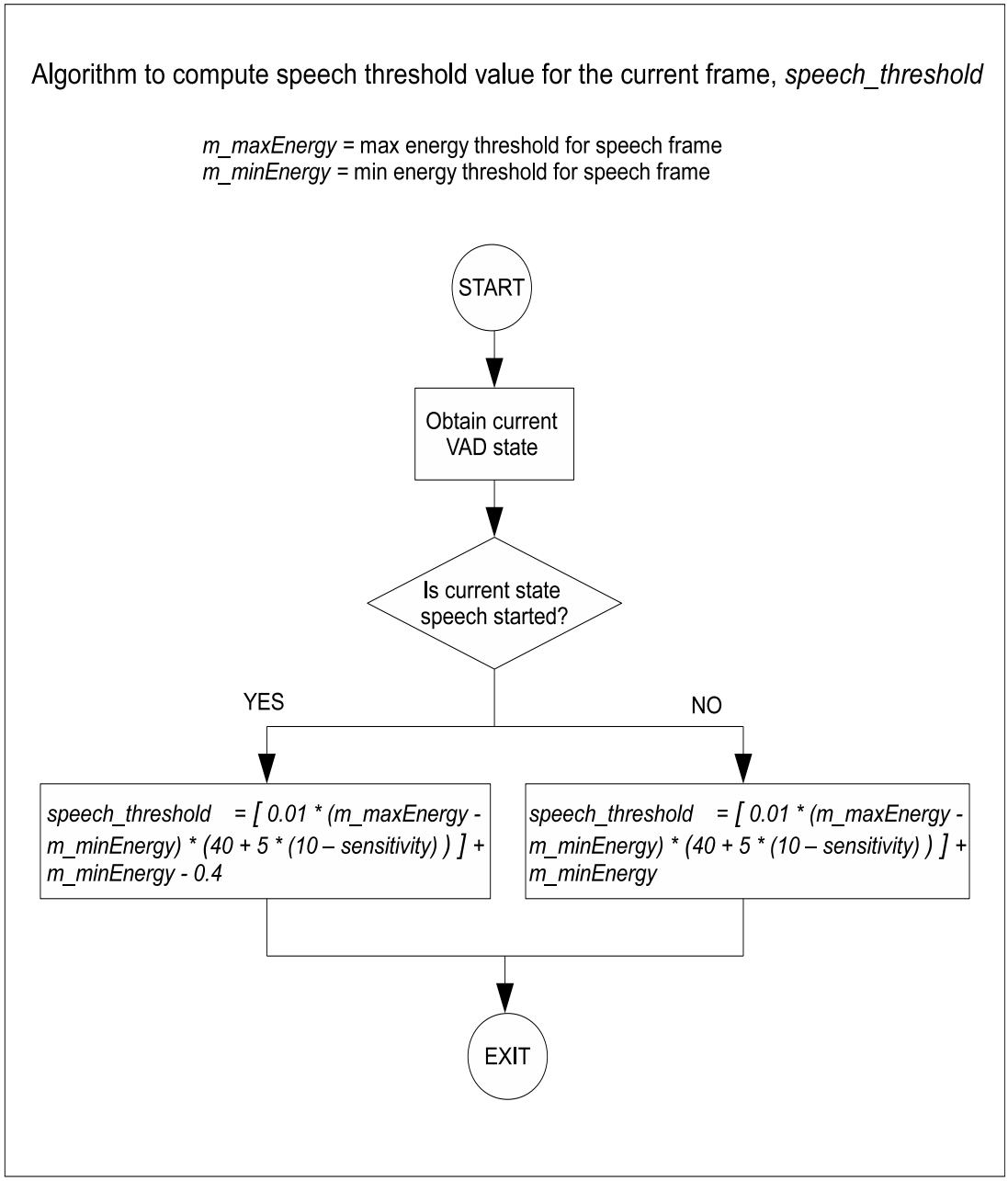

Figure 3.9. Flowchart of the algorithm to compute *speech_threshold*

• *STEP 11)* The thresholds for minimum/maximum energy levels, *m_minEnergy* and *m_maxEnergy*, are updated. The update of the *m_minEnergy* is done as follows: If *m_energy* is lower than *m_minEnergy*, *m_minEnergy* is decreased. *m_minEnergy* is

decreased as

$$m\_minEnergy = (m\_minEnergy * 0.99) + (m\_energy * 0.01) \tag{3.18}$$

If the *m_energy* is higher or equal to *m_minEnergy* and the value of (*m_maxEnergy*-*m_energy*) is bigger than 1.5, *m_minEnergy* is increased. If the present state of the system is speech started, the increased *m_minEnergy* is calculated as

$$m\_minEnergy = (m\_minEnergy * 0.998) + (m\_energy * 0.002) \tag{3.19}$$

Conversely, if no actual speech has been determined previously, *m_minEnergy* is calculated as in Eq. 3.18. The reason for using different coefficients for updating *m_minEnergy* in speech and non-speech regions is the need of smoother updates after the start of speech. It is ensured that the value of *m_minEnergy* is always greater than 2.0.

The flowchart of the algorithm to update *m_minEnergy* is shown in Figure 3.10.

The update of the *m_maxEnergy* is done as follows:

$$m\_maxEnergy = (m\_maxEnergy * 0.99) + (m\_energy * 0.01) \tag{3.20}$$

After this update, no other update is applied if *m_energy* is smaller than *m_maxEnergy*. However, if *m_energy* is bigger than *m_maxEnergy*, *m_maxEnergy* is increased. If the present state of the system is speech started, the increased *m_maxEnergy* is calculated as

$$m\_maxEnergy = (m\_maxEnergy * 0.998) + (m\_energy * 0.002) \tag{3.21}$$

If no actual speech has been determined previously, a faster update is required and the new *m_maxEnergy* is calculated as

$$m\_maxEnergy = (m\_maxEnergy * 0.9) + (m\_energy * 0.1) \tag{3.22}$$

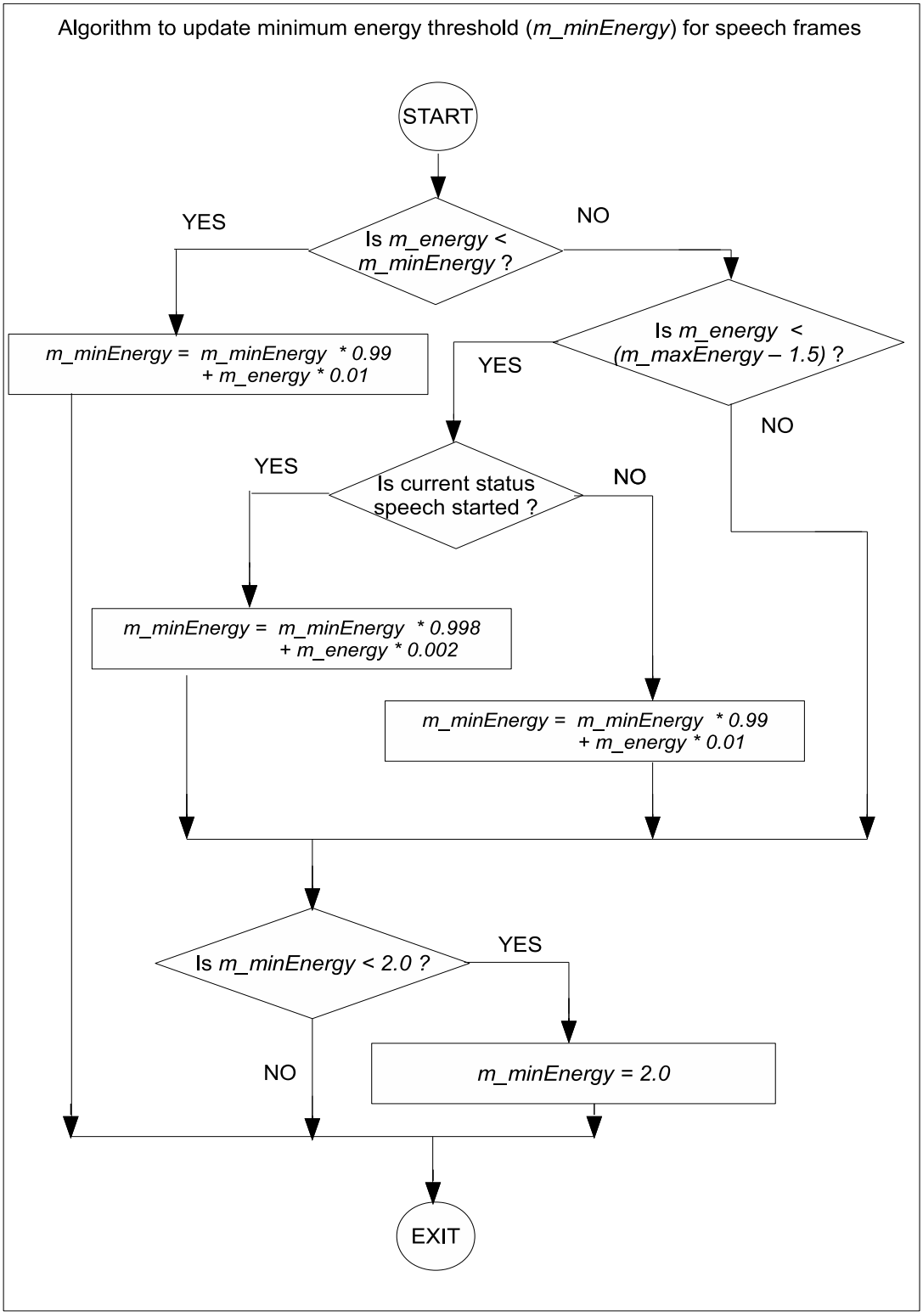

Figure 3.10. Flowchart of the algorithm to update *m_minEnergy*

Similar to the case for *m_minEnergy*, the reason for using different coefficients for *m_maxEnergy* update is that a smoother update is required after the start of speech. It is ensured that the value of *m_maxEnergy* is always above 4.5 .

The flowchart of the algorithm to update *m_maxEnergy* is shown in Figure 3.11.

• *STEP 12)* A soft decision value, *soft_decision*, is associated with the current frame. The *soft_decision* value assigned to the frame is a measure of the speech likeliness of that frame (I.e. higher *soft_decision* values are assigned to speech frames compared to non-speech frames). Since *m_energy* parameter contains the combined effects of the energy measure and the periodicity measure due to the summation operation in Eq. 3.16, the assignment of *soft_decision* is based on the values of *m_energy* and *speech_threshold*.

If *m_energy* $\geq$ (*speech_threshold* - 0.5) and *prob_voice* is greater than 0.4, the soft decision value is calculated as

$$soft\_decision = 0.75 + m\_energy - speech\_threshold \tag{3.23}$$

Else if *m_energy* $\geq$ (*speech_threshold* - 0.5) but *prob_voice* is not greater than 0.4, the soft decision value is calculated as

$$soft\_decision = 0.5 + m\_energy - speech\_threshold \tag{3.24}$$

If none of the above conditions are met, the assigned soft decision value for the current frame is 0.

The flowchart of the algorithm to compute *soft_decision* is shown in Figure 3.12.

• *STEP 13)* Decision smoothing is applied on the calculated soft decision value, *soft_decision*. The smoothing operation is done by rewarding the *soft_decision* value

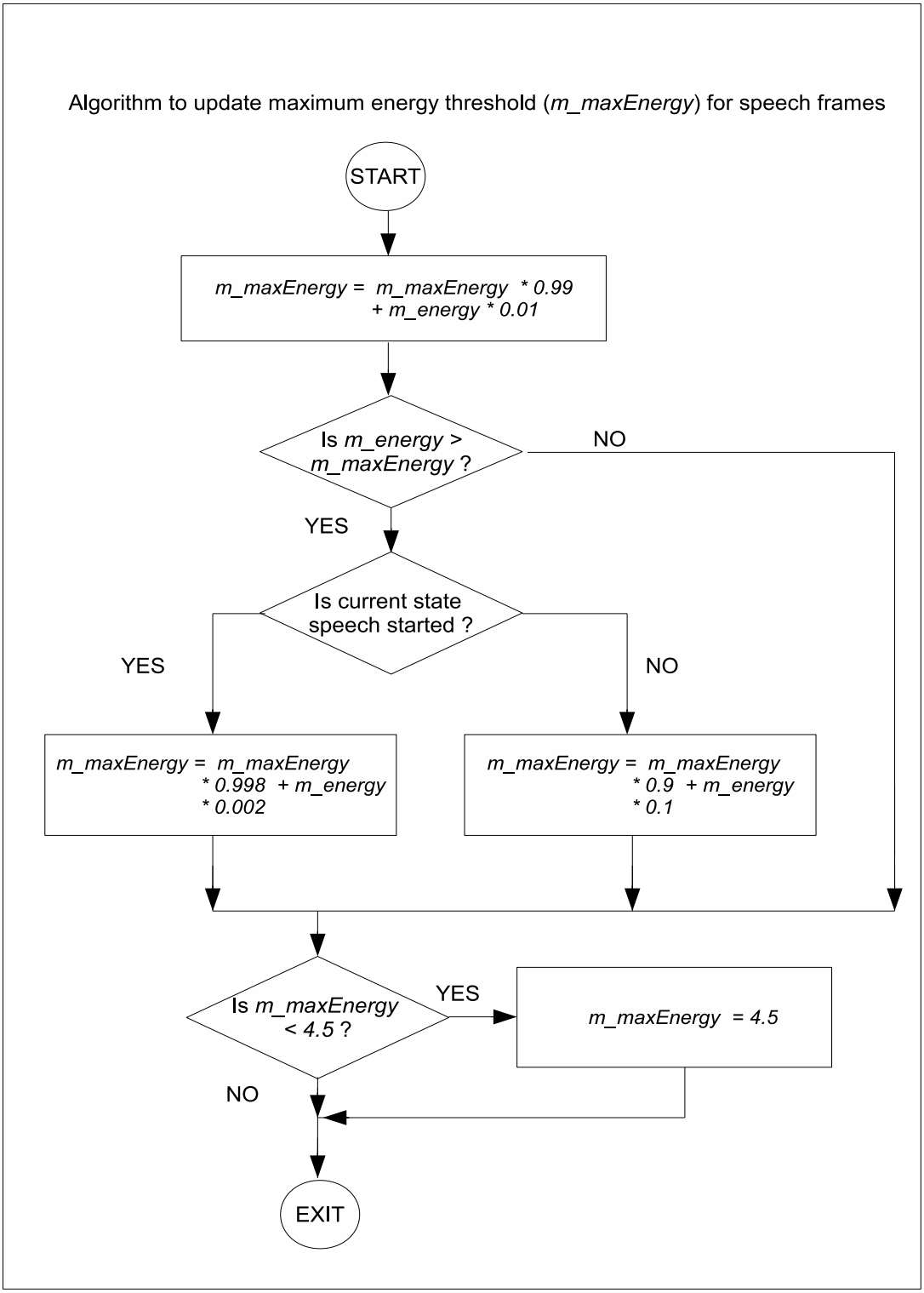

Figure 3.11. Flowchart of the algorithm to update *m_maxEnergy*

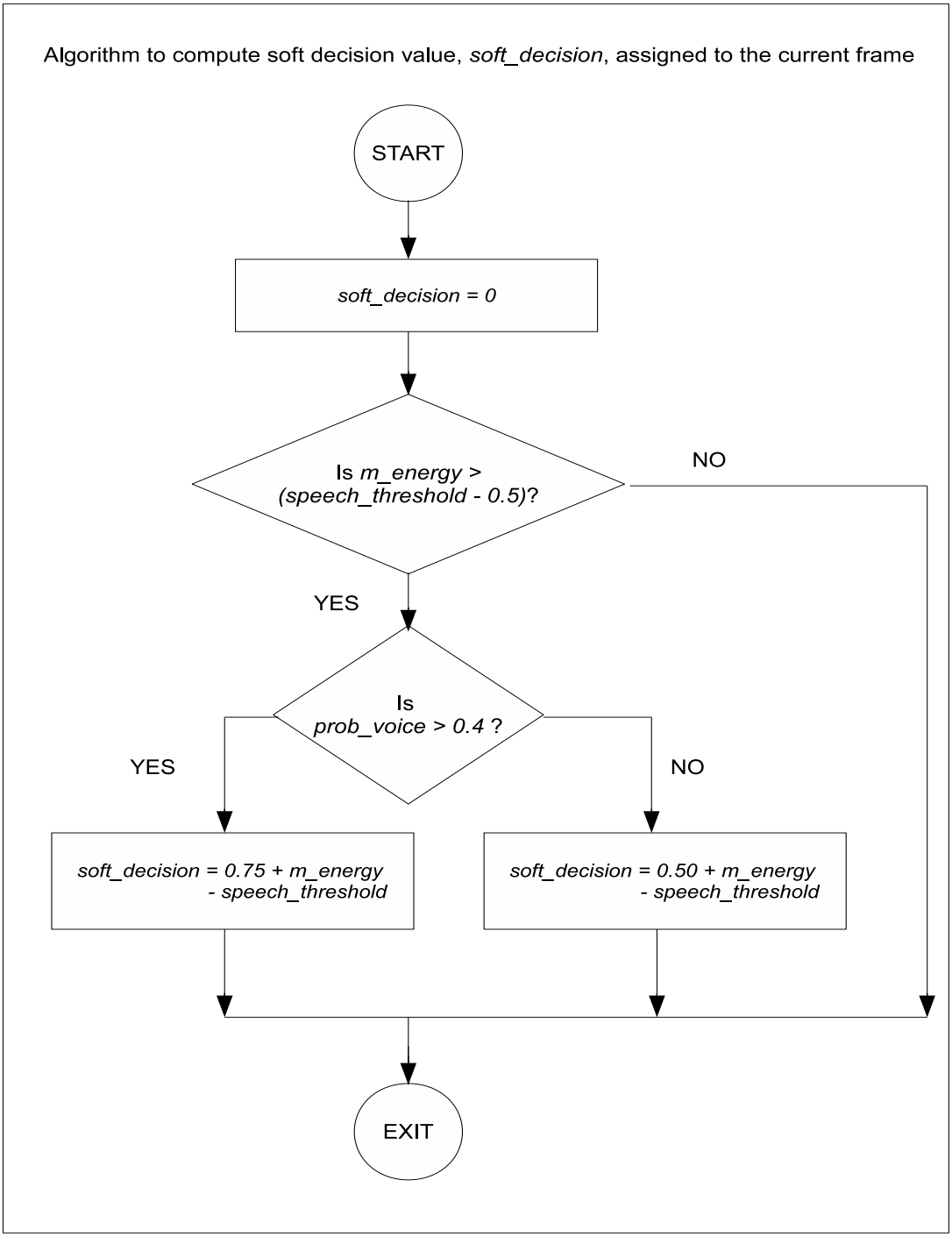

Figure 3.12. Flowchart of the algorithm to compute *soft_decision*

of the current frame if the *soft_decision* value for the previous frame is high. If both the current *soft_decision* and the previous *soft_decision* values exceed 0.5, the current *soft_decision* value is increased by 0.3 .

The flowchart of the algorithm for decision smoothing is shown in Figure 3.13.

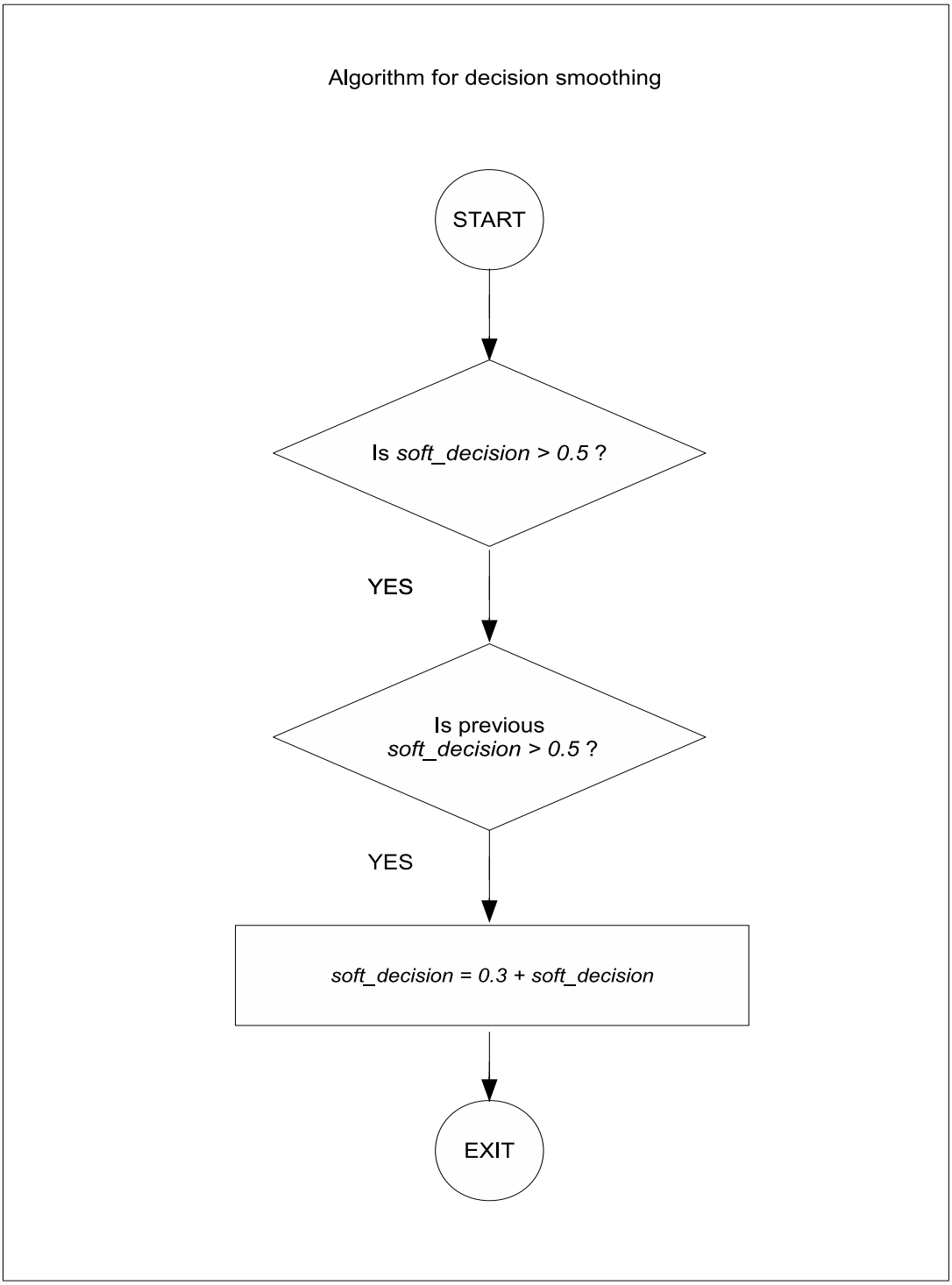

Figure 3.13. Flowchart of the algorithm for decision smoothing

• *STEP 14)* Lastly, the final speech/non speech decision is made. The decision

is based on the history of the last 20 subjective *soft_decision* values. The sum of the *soft_decision* values for the last 20 frames is calculated as the variable *m_Vad*. Then, *m_Vad* is compared with *speech trigger*.

If *m_Vad* > *speech trigger*, the processed frame is considered as a speech frame and the current state is set to speech started. If a state change from speech ended to speech started is also accompanied with this change, this means that the start of a new utterance is detected. The system starts to write the current frame into a new, separate output stream (Current frame is the first frame of the speech). If there is no state change (I.e., previous state was also speech started), this means that the write operation to the output stream has already started, so no change is done.

If *m_Vad* $\leq$ *speech trigger*, and the algorithm has previously detected the presence of speech frames, the variable *silence_counter*, that keeps track of the number of successive silence frames, is incremented. At that point, the soft decision value assigned to the current frame is elaborated. If *soft_decision* > 0.5 and *m_Vad* $\geq$ (0.5 * *speech trigger*), the current frame is considered as a continuation of speech and *silence_counter* is set to 0. Then, the total length of the successive silence frames is found by multiplying *silence_counter* with the frame length. If this value exceeds the *silence trigger* the current state is set to speech ended. If a state change from speech started to speech ended is also accompanied with this change, this means that the end of a new utterance is detected. The system stops the write operation. If there is no state change (I.e., previous state was also speech ended), this means that the write operation to the output stream has not started yet, so no change is done.

The flowchart of the algorithm for final speech/non-speech decision is shown in Figure 3.14.

Finally, *prespeech buffer* and *postspeech buffer* values are used to extend the limits of the speech boundaries for final VAD output. As previously mentioned, these extensions are required in order to enable the detection of actual speech instances at the speech/non-speech boundaries that might be missed by standard VAD decisions.

Figure 3.15 demonstrates the use of *prespeech buffer* and *postspeech buffer* for final VAD output.

3.3.6. Hybrid VAD

As expected, the performance of the proposed VAD is degraded under low SNR conditions. After experiments, two main problems were observed under low SNR conditions. Firstly, noise frames with high energy could easily be detected as actual speech. The adaptively changing thresholds, *m_maxEnergy* and *m_minEnergy*, avoided the errors to some extent for stationary noise types. However, the problem persisted especially for non-stationary or impulsive noise types. Secondly, detection of actual speech frames that contain low energy was problematic when the SNR was low. This was mainly because of the increased thresholds due to the high background noise.

The *Hybrid VAD* algorithm, that we implement as the VAD block of our proposed unified system for VAD and speech enhancement, tries to eliminate the above problems by utilizing speech enhancement. By making use of speech enhancement in the form of MWF, the performance of the proposed VAD is improved. Two main modifications are done in the proposed VAD algorithm. Firstly, noised suppressed frames are used to extract features for Hybrid VAD algorithm. This results in better periodicity characterization for speech frames and avoids the artificial increase of energy threshold levels under low SNR conditions. Secondly, since the spectral energy variation is more distinct for noise suppressed frames, the contribution of the spectral energy difference measure in the final speech/non-speech decision is emphasized by modifying Eq. 3.16 to

$$\begin{aligned} m\_energy &= \Big(1.10 * m\_energyWeighted\Big) \\ &\quad + \Big(0.375 * min(m\_energyDifference, 2)\Big) \\ &\quad + \Big(min(1.0, 0.5 * prob\_voice)\Big) \end{aligned} \tag{3.25}$$

for Hybrid VAD. This modification compensates for the reduced contribution of the

spectrally weighted energy parameter, *m_energyWeighted*, due to the removal of the noise from input signal. The net effect of these modifications is the increased accuracy of soft decision values assigned to frames.

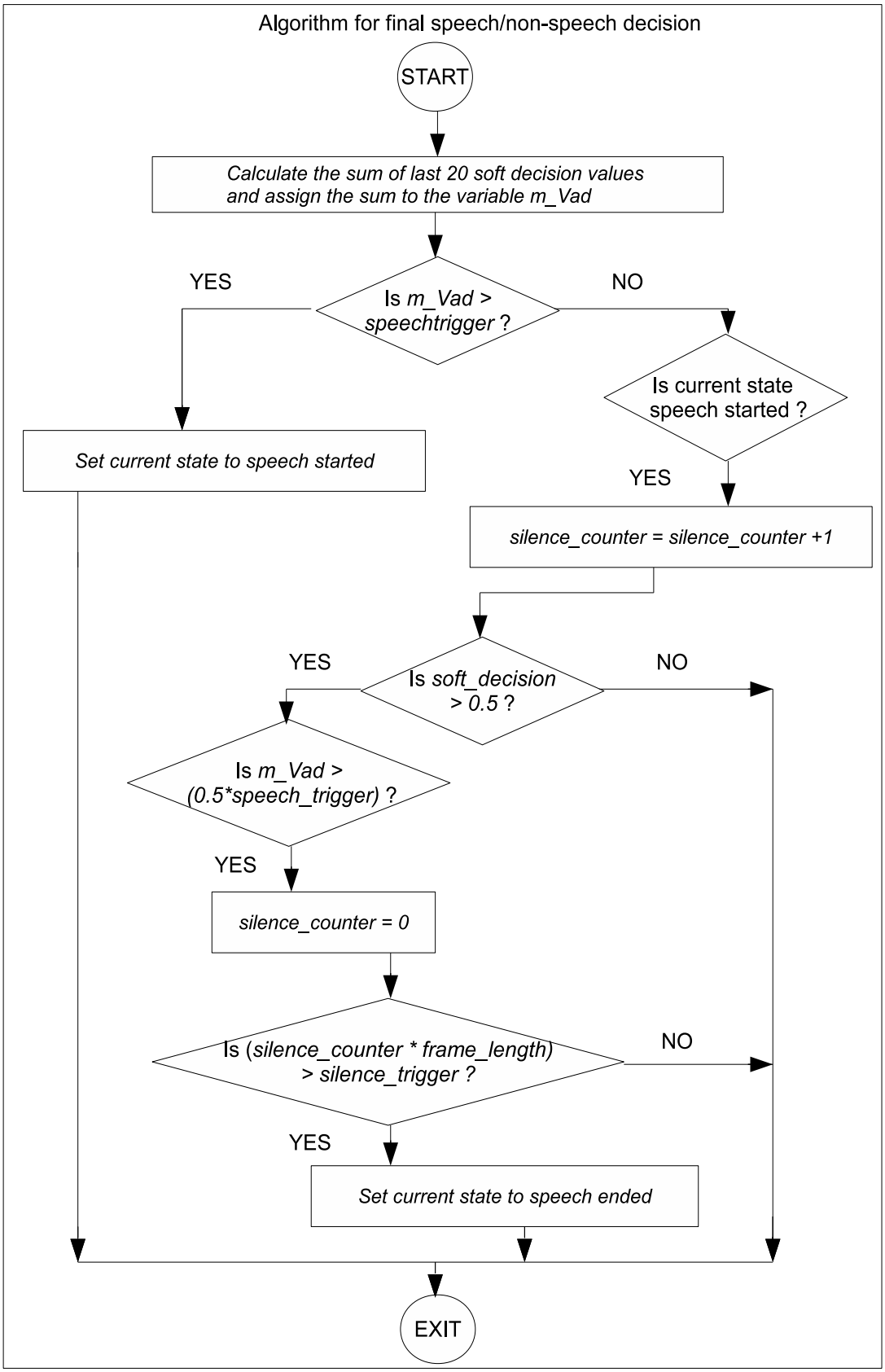

Figure 3.14. Flowchart of the algorithm for final speech/non-speech decision

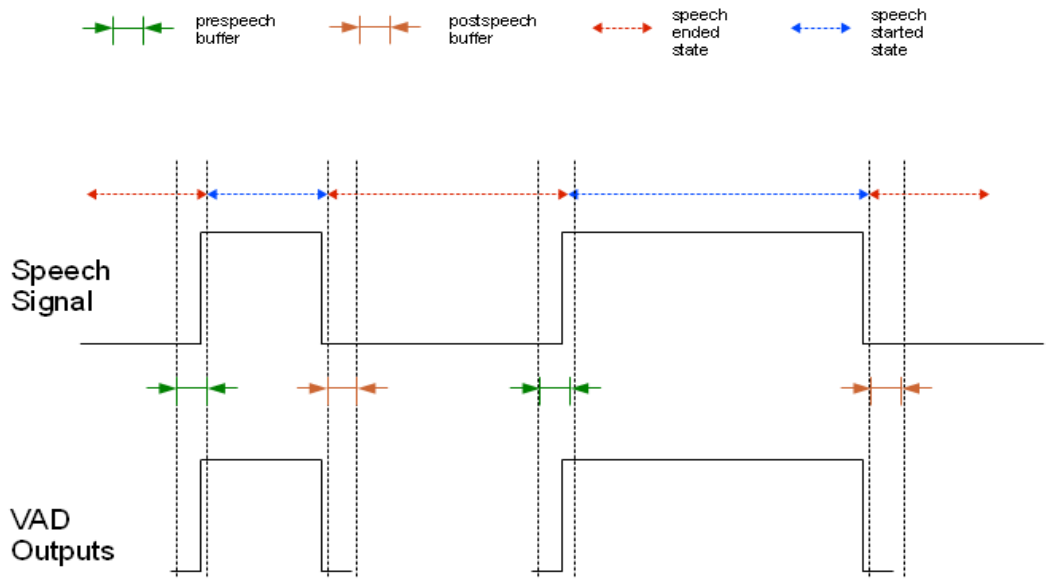

Figure 3.15. Use of *prespeech buffer* and *postspeech buffer* for final VAD output

4. SPEECH ENHANCEMENT

In this chapter, we first explain the mathematical model of the speech enhancement problem. Second, a literature review of single channel speech enhancement algorithms is provided. Finally, the EMWF algorithm that we use as the speech enhancement block of our proposed unified system is described in detail.

4.1. Speech Enhancement Problem

Let $\{s(t)\},\ s(t)\ \epsilon\ \Re$ denote a random process that models the clean speech. Let $\{n(t)\},\ n(t)\ \epsilon\ \Re$ denote a second random process for representing background noise. The additive noise model assumes that the noisy speech is modeled by another random process $\{y(t)\},\ y(t)\ \epsilon\ \Re$ where

$$y(t) = s(t) + n(t) \tag{4.1}$$

In this formulation $s(t)$ and $n(t)$ are statistically independent. Eq. 4.1 represents the additive noise model that we will use throughout the thesis.

Speech enhancement is the estimation of clean speech $s(t)$, from the noisy speech $y(t)$. Speech enhancement can be either single-channel or multi-channel. In single-channel enhancement, speech is available from only a single microphone, whereas multi-channel systems make use of more than one microphone [7]. Multi-channel speech enhancement techniques have the advantage of multiple signal inputs to the system and this enables better noise characterization and therefore better noise suppression. However, these systems are inherently more complex in nature and this imposes constraints in terms of the algorithmic complexity and cost. In addition, there may be applications where multiple microphone input is not possible due to hardware constraints. For these reasons, single-channel speech enhancement techniques became more popular. In this thesis, we will only be dealing with single channel speech enhancement algorithms.

The schematic representation of single channel speech enhancement problem is shown in Figure 4.1.

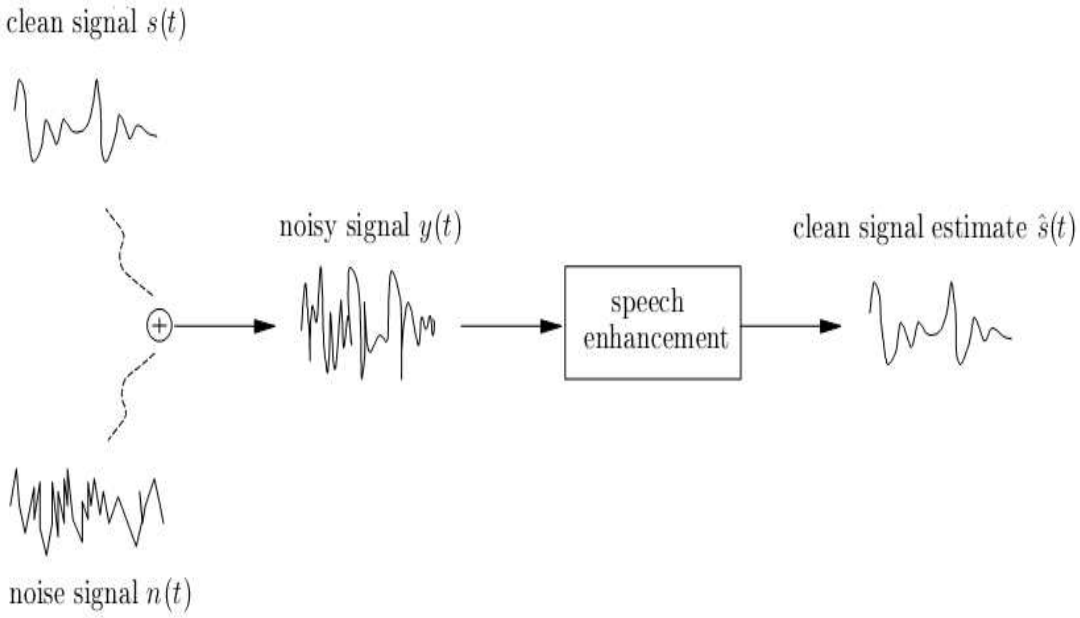

Figure 4.1. Schematic representation of single channel speech enhancement problem

As indicated in Ephraim *et. al.* [33], there are two principal perceptual criteria for measuring the performance of a speech enhancement system. First criterion is the quality of the enhanced signal. Quality of the enhanced signal measures its clarity, distorted nature, and the level of residual noise in that signal. Quality is a subjective measure that is indicative of the extent to which the listener is comfortable with the enhanced signal. Intelligibility of the enhanced signal is the second perceptual criterion. This is an objective measure which provides the percentage of words that could be correctly identified by listeners. The words in this test do not need to be meaningful. The two performance measures are not correlated. A signal may be of good quality and poor intelligibility and vice versa. So far, no enhancement system was capable of improving both the quality and intelligibility of the noisy signal [34]. Most speech enhancement systems improve the quality of the signal at the expense of reducing its intelligibility. Listeners can usually extract more information from the noisy signal than from the enhanced signal by carefully listening to that signal [33].

Next section presents a literature review of single channel speech enhancement algorithms.

4.2. Literature Review of Single Channel Speech Enhancement Algorithms

Single-channel enhancement methods can be divided into mainly four groups: (i) spectral subtraction-based methods [35, 36, 37], (ii) Wiener filtering-based methods [38, 39, 40, 41, 42], (iii) statistical model-based methods [43, 44, 45, 46] and (iv) subspace methods [34, 47].

4.2.1. Spectral Subtraction Algorithms

Spectral subtraction methods are historically the first algorithms proposed for noise reduction [6]. These algorithms rely on the assumption that as long as we have an estimate for the noise spectrum, it is possible to obtain an estimate of the clean signal spectrum by subtracting the noise spectrum estimate from the noisy speech spectrum. Clean signal is assumed to have the same phase with noisy speech. Since these algorithms require only a single forward and an inverse Fourier transform, they are computationally attractive.

4.2.1.1. Basic Principles of Spectral Subtraction. DFT (Discrete-time Fourier Transform) of both sides of the additive noise equation, Eq. 4.1, gives

$$Y(\omega) = S(\omega) + N(\omega) \tag{4.2}$$

In polar form, it is possible to express $Y(\omega)$ as

$$Y(\omega) = |Y(\omega)|e^{j\phi_y(\omega)} \tag{4.3}$$

where $|Y(\omega)|$ is the magnitude spectrum and $e^{j\phi_y(\omega)}$ is the phase spectrum of the corrupted noisy signal. The noise spectrum, $N(\omega)$, can also be expressed in terms of

its magnitude and phase spectra as

$$N(\omega) = |N(\omega)|e^{j\phi_n(\omega)} \tag{4.4}$$

The magnitude of the noise spectrum, $|N(\omega)|$ is unknown. Spectral subtraction algorithms form an initial estimate of $|N(\omega)|$ and update this estimate during non-speech activity. The assumption made is that noise is stationary or a slowly varying process, and that the noise spectrum does not change significantly during non-speech periods. Phase component of noise, $\phi_n(\omega)$, is directly replaced by the noisy speech phase $\phi_y(\omega)$. This approach is motivated by the fact that phase information does not affect the speech intelligibility and may affect speech quality only up to some degree [48]. After making these substitutions in Eq. 4.2, we can express the clean speech spectrum estimate $\hat{S}(\omega)$ as

$$\hat{S}(\omega) = \Big[|Y(\omega)| - |\hat{N}(\omega)|\Big]e^{j\phi_y(\omega)} \tag{4.5}$$

where $|\hat{N}(\omega)|$ denotes the estimate of the magnitude of noise spectrum. Clean speech signal estimate, $\hat{s}(t)$, is found by taking the inverse Fourier transform of $\hat{S}(\omega)$.

Block diagram of a typical spectral subtraction algorithm is shown in Figure 4.2.

Performance of spectral subtraction algorithms depend on the accuracy of the noise spectrum estimation. Overestimation of noise spectrum causes speech information loss, whereas underestimation of it results in the residual of the interfering noise. Another problem associated with the inaccuracies in noise spectrum estimation is the negative values in magnitude spectrum of the enhanced signal. Caution must be exercised during the subtraction operation in order to ensure that the magnitude spectrum of the estimated signal, $|\hat{S}(\omega)|$, is never negative. One solution offered to this problem is to half-wave-rectify [36] the difference spectra ($|Y(\omega)| - |\hat{N}(\omega)|$) and set the negative

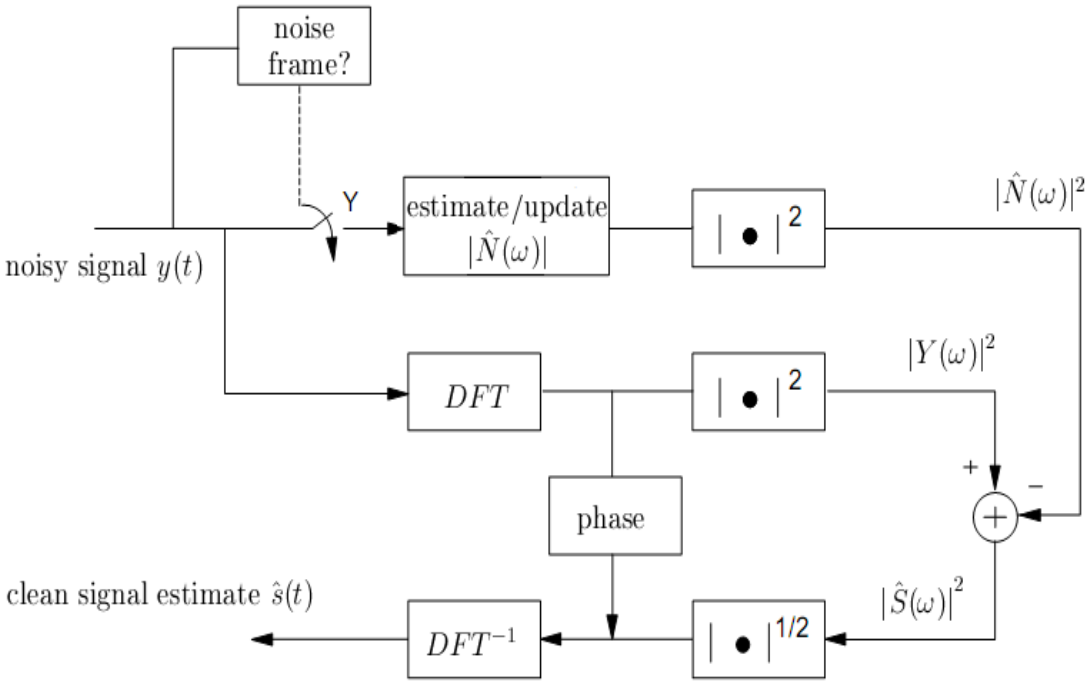

Figure 4.2. Block diagram of a typical spectral subtraction algorithm

spectral components to zero as follows:

$$\begin{aligned} |\hat{S}(\omega)| &= |Y(\omega)| - |\hat{N}(\omega)| \quad \text{if } |Y(\omega)| > |\hat{N}(\omega)| \\ &= 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{aligned} \tag{4.6}$$

4.2.2. Wiener Filtering Algorithms

Wiener filtering was first proposed by Wiener et. al [49] as the optimized filter for eliminating the effects of noise from a signal. The motivation behind Wiener filtering can be best understood by the statistical filtering problem that is depicted in Figure 4.3.

As can be seen in Figure 4.3, the input signal $y(n)$ goes through a linear time invariant system to produce an output signal $\hat{s}(n)$. The problem is to design the system in such a way that the output signal $\hat{s}(n)$ is as close as possible to the desired signal

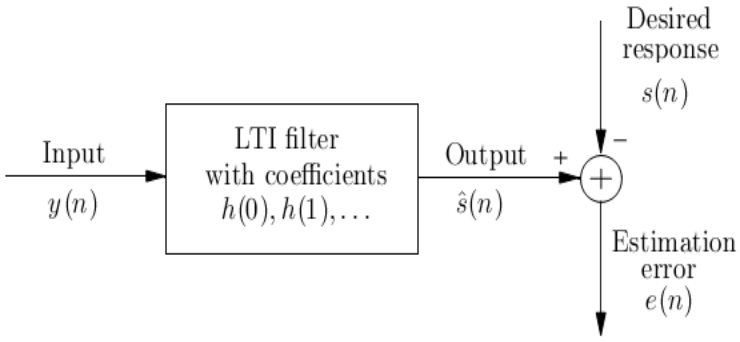

Figure 4.3. Block diagram of the Wiener filtering problem

$s(n)$. Theoretically, the filter representing the system could be either infinite impulse response (IIR) or finite impulse response (FIR), but often a FIR system is desired because of stability. Assuming a FIR solution of order P, the output of the filter, $\hat{s}(n)$, is given as

$$\hat{s}(n) = \sum_{k=1}^{P} h(k)y(n-k) \tag{4.7}$$

where $h(k)$ for $k = 1 \ldots P$ are the filter coefficients.

The residual error of this approximation denoted by $e(n)$ is

$$e(n) = s(n) - \hat{s}(n) \tag{4.8}$$

The Wiener filter coefficients $h(k)$ must be chosen to minimize the mean square error $E[e(n)^2]$, where $E[.]$ denotes the expectation operator. The mean square error is

$$\begin{aligned} E[e(n)^2] &= E[\Big(s(n) - \hat{s}(n)\Big)^2] \\ &= E[s^2(n)] + E[\hat{s}^2(n)] - 2E[s(n)\hat{s}(n)] \\ &= E[s^2(n)] + E[\Big(\sum_{k=1}^{P} h(k)y(n-k)\Big)^2] - 2E[s(n)\sum_{k=1}^{P} h(k)y(n-k)] \end{aligned} \tag{4.9}$$

In order to determine the Wiener filter coefficients, $h(k)$ for $k = 1 \ldots P$, that minimize the above expression, we calculate the derivative of the above equation with respect to each $h(k)$ as

$$\begin{aligned} \frac{\partial E[e(n)^2]}{\partial h(k)} &= 2E[\Big(\sum_{l=1}^{P} h(l)y(n-l)\Big)y(n-k)] - 2E[s(n)y(n-k)] \\ &= 2\sum_{l=1}^{P} E[y(n-l)y(n-k)]h(l) - 2E[s(n)y(n-k)] \end{aligned} \tag{4.10}$$

for $k = 1 \ldots P$. The autocorrelation term, $R_{yy}(l-k) = E[y(n-l)y(n-k)]$ and the cross-correlation term, between $R_{sy}(k) = E[s(n)y(n-k)]$ enables us to rewrite Eq. 4.10 as

$$\frac{\partial E[e(n)^2]}{\partial h(k)} = 2\sum_{l=1}^{P} R_{yy}(l-k)h(l) - 2R_{sy}(k) \qquad \text{for } k = 1 \ldots P \tag{4.11}$$

Letting the derivative be equal to zero, we obtain

$$\sum_{l=1}^{P} R_{yy}(l-k)h(l) = R_{sy}(k) \qquad \text{for } k = 1 \ldots P \tag{4.12}$$

We can represent Eq. 4.12 in matrix form as

$$\begin{bmatrix} R_{yy}(0) & R_{yy}(1) & \ldots & R_{yy}(P-1) \\ R_{yy}(1) & R_{yy}(0) & \ldots & R_{yy}(P-2) \\ \vdots & \vdots & \ldots & \vdots \\ R_{yy}(P-1) & R_{yy}(P-2) & \ldots & R_{yy}(0) \end{bmatrix} \begin{bmatrix} h(1) \\ h(2) \\ \vdots \\ h(P) \end{bmatrix} = \begin{bmatrix} R_{sy}(1) \\ R_{sy}(2) \\ \vdots \\ R_{sy}(P) \end{bmatrix} \tag{4.13}$$

Eq. 4.12 is time domain interpretation of Wiener filtering.

Most of the Wiener filtering-based speech enhancement algorithms operate on frequency domain, so it may be useful to elaborate the frequency domain interpretation

of Wiener filtering. In order to obtain Wiener filter equation in frequency domain, we first note that the input of the Wiener filter, $y(n)$, is related to the output $\hat{s}(n)$ as

$$\hat{s}(n) = h(n) * y(n) \tag{4.14}$$

where $*$ denotes the convolution operator. Therefore, in frequency domain we have

$$\hat{S}(\omega) = H(\omega)Y(\omega) \tag{4.15}$$

where $H(\omega)$ and $Y(\omega)$ are the discrete-time Fourier transforms of $h(n)$ and $y(n)$. We can define the estimation error at frequency domain as [6]

$$\begin{aligned} E(\omega) &= S(\omega) - \hat{S}(\omega) \\ &= S(\omega) - H(\omega)Y(\omega) \end{aligned} \tag{4.16}$$

We need to compute $H(\omega)$ that minimizes the mean square error. The mean square error in frequency domain is given by

$$\begin{aligned} E[|E(\omega)|^2] &= E\Big[[S(\omega) - H(\omega)Y(\omega)][S(\omega) - H(\omega)Y(\omega)]^*\Big] \\ &= E[|S(\omega)|^2] - H(\omega)E[S^*(\omega)Y(\omega)] - H^*(\omega)E[Y^*(\omega)S(\omega)] \\ &\quad +|H(\omega)|^2 E[|Y(\omega)|^2] \end{aligned} \tag{4.17}$$

In the above equation if we note that $P_y(\omega) = E[|Y(\omega)|^2]$ is the power spectrum of $y(n)$, and $P_{ys}(\omega) = E[Y(\omega)S^*(\omega)]$ is the cross power spectrum of $y(n)$ and $s(n)$, we can express the mean square error as

$$\begin{aligned} E[|E(\omega)|^2] &= E[|S(\omega)|^2] - H(\omega)P_{ys}(\omega) - H^*(\omega)P_{sy}(\omega) \\ &\quad +|H(\omega)|^2 P_y(\omega) \end{aligned} \tag{4.18}$$

In order to find the optimal filter $H(\omega)$, we take the derivative of the mean square error in Eq. 4.18 as

$$\begin{aligned}\frac{\partial E[|E(\omega)|^2]}{\partial H(\omega)} &= 2\Big[H^*(\omega)P_y(\omega) - P_{ys}(\omega)\Big] \\ &= 2\Big[H(\omega)P_y(\omega) - P_{sy}(\omega)\Big]^*\end{aligned} \tag{4.19}$$

and set it equal to 0. Solving for $H(\omega)$ gives the general form of Wiener filter in frequency domain as

$$H(\omega) = \frac{P_{sy}(\omega)}{P_y(\omega)} \tag{4.20}$$

In the context of speech enhancement, Wiener filtering is used to produce an estimate of the clean signal $s(t)$ for the additive noise model of $y(t) = s(t) + n(t)$. According to Eq. 4.20 we need to compute $P_{sy}(\omega)$ and $P_y(\omega)$. We have

$$\begin{aligned}P_{sy}(\omega) &= E[S(\omega)Y^*(\omega)] \\ &= E\Big[S(\omega)[S(\omega) + N(\omega)]^*\Big] \\ &= E[S(\omega)S^*(\omega)] + E[S(\omega)N^*(\omega)] \\ &= P_s(\omega)\end{aligned} \tag{4.21}$$

where we used the fact that since $s(t)$ and $n(t)$ are uncorrelated, $E[S(\omega)N^*(\omega)] = 0$. Next we find $P_y(\omega)$ as

$$\begin{aligned}P_y(\omega) &= E\Big[[S(\omega) + N(\omega)][S(\omega) + N(\omega)]^*\Big] \\ &= E[S(\omega)S^*(\omega)] + E[N(\omega)N^*(\omega)] + E[S(\omega)N^*(\omega)] + E[N(\omega)S^*(\omega)] \\ &= P_s(\omega) + P_n(\omega)\end{aligned} \tag{4.22}$$

Substituting Eq. 4.21 and Eq. 4.22 into Eq. 4.20, we have the noise suppressing Wiener filter as

$$H(\omega) = \frac{P_s(\omega)}{P_s(\omega) + P_n(\omega)} \tag{4.23}$$

More generally, parametric Wiener filters of the form [41]

$$H(\omega) = \left(\frac{P_s(\omega)}{P_s(\omega) + \alpha P_n(\omega)} \right)^{\beta} \tag{4.24}$$

are more commonly used in speech enhancement algorithms.

Eq. 4.23 and 4.24 are intuitively appealing in the sense that when $P_s(\omega) >> P_n(\omega)$, the magnitude response of the filter becomes unity, meaning that all the input is preserved at the output without modification. Conversely when $P_s(\omega) << P_n(\omega)$, the magnitude response of the filter becomes zero, meaning that all the input is blocked at the output.

The main problem of Wiener filtering-based speech enhancement algorithms is the requirement to know the power spectrum of the clean signal. One approach to deal with this problem is to compute the speech power spectrum estimate from the Wiener filter output in an iterative fashion [6]. The iterative approach is implemented as follows:

• Step 1) Obtain an estimate of the Wiener filter $H_i(\omega)$ based on the enhanced signal at the i^{th} iteration, $\hat{s}_i(n)$. In the first iteration, initialize $\hat{s}_i(n)$ with the noisy speech signal $y(n)$.

• Step 2) In order to get the new enhanced signal $\hat{s}_{i+1}(n)$ for the next iteration, filter the noisy speech signal $y(n)$ through the newly obtained Wiener filter $H_i(\omega)$ according to equation

$$\hat{S}_{i+1}(\omega) = H_i(\omega) Y(\omega) \tag{4.25}$$

Take inverse discrete-time Fourier transform (DFT^{-1}) of $\hat{S}_{i+1}(\omega)$ to obtain $\hat{s}_{i+1}(n)$.

- Step 3) Repeat Step 2 by using $\hat{s}_{i+1}(n)$ in place of $\hat{s}_i(n)$.

As with all iterative algorithms, there is the issue of convergence for Wiener filtering-based speech enhancement algorithms [6]. Moreover, iterative approach may not be suitable for real-time applications due to the increased number of computations.

4.2.3. Statistical Model-Based Algorithms

Statistical model-based algorithms try to obtain a good estimator of some parameter for the clean signal by using a set of measurements for the corresponding noisy speech [6]. The most popular approach of the statistical model-based algorithms is the Minimum Mean Square Estimate (MMSE) introduced by Ephraim *et. al.* [43]. In their work, they use MMSE to determine the optimal filter amplitude based on the assumption that the Fourier transform coefficients of speech and noise conform to a Gaussian distribution. They exploit MMSE to solve

$$\hat{S}(\omega_k) = E\Big[S(\omega_k)|Y(\omega_k)\Big] \tag{4.26}$$

where $\hat{S}(\omega_k)$ is the k^{th} frequency component of the estimated clean speech signal. The optimal gain function is given in terms of the a priori and the a posteriori SNRs, ξ_k and γ_k, respectively. The a priori and the a posteriori SNRs are defined by

$$\xi_k = \frac{\lambda_s(k)}{\lambda_n(k)} \tag{4.27}$$

and

$$\gamma_k = \frac{Y(\omega_k)^2}{\lambda_n(k)} \tag{4.28}$$

where $\lambda_s(k) = E\Big[|S(\omega_k)|^2\Big]$ is the variance of the k^{th} spectral component of the speech, $\lambda_n(k) = E\Big[|N(\omega_k)|^2\Big]$ is the variance of the k^{th} spectral component of the noise and $Y(\omega_k)$ is the k^{th} spectral component of the noisy speech $y(t)$. The estimator is given as

$$\hat{S}(\omega_k) = \frac{\sqrt{\pi}}{2} J(v_k) \frac{\sqrt{v_k}}{\gamma_k} e^{-\frac{v_k}{2}} Y(\omega_k) \tag{4.29}$$

where

$$J(v_k) = \Big[(1 + v_k) I_0(\frac{v_k}{2}) + v_k I_1(\frac{v_k}{2})\Big] \tag{4.30}$$

and

$$v_k = \frac{\xi_k}{1 + \xi_k} \gamma_k \tag{4.31}$$

and I_0 and I_1 are the modified Bessel functions of order zero and one, respectively.

Different statistical model-based approaches use Laplacian pdf or Gamma pdf rather than Gaussian pdf in order to model the probability density of the distribution function of the real and imaginary parts of the Fourier transform coefficients of speech and noise of speech samples. It has been shown that when the signal frame is smaller than 100 ms, the Laplacian and Gamma densities give better performance [45, 46].

4.2.4. Subspace Algorithms

Subspace algorithms rely on the assumption that given a method of decomposing the vector space of the noisy signal into a subspace that is occupied primarily by the clean signal and another subspace occupied primarily by the noise signal, one can estimate the clean signal simply by removing the noise component, which lies in the subspace occupied by the noise, from the noisy speech [6]. These algorithms are primarily rooted on linear algebra theory so in order to understand the basics of subspace algorithms, a treatment of the related linear algebra concepts is required.

Following is a small treatment of the basic principles of subspace algorithms that is largely adapted from [50].

4.2.4.1. Basic Principles of Subspace Algorithms. The vector space R^n contains the space of all vectors with n components. Two subspaces V and W are orthogonal if every vector $\bar{v}$ in space V is orthogonal to every vector $\bar{w}$ in W, i.e., $\bar{v}^T\bar{w} = 0$ for all $\bar{v} \in V$ and $\bar{w} \in W$. A subspace is spanned by a set of linearly independent vectors called the *basis vectors*. The linear combination of the basis vectors $\bar{x_1}, \bar{x_2}, \ldots, \bar{x_N}$ defines the subspace in matrix form that is shown by the matrix $\boldsymbol{A} = [\bar{x_1}, \bar{x_2}, \ldots, \bar{x_N}]$.

An important concept in linear algebra is the projection of a vector onto a space. Projection of a vector onto a space denotes a vector that is constrained to be in the space and resembles the vector that is projected. In signal processing applications, orthogonal projections are very important because they provide an excellent tool for decoupling a vector into two components that occupy different subspaces. It can be shown that [50] the projection matrix $\boldsymbol{P}$, that orthogonally projects a vector onto space A, can be given by $\boldsymbol{P} = \boldsymbol{A}(\boldsymbol{A^H A})^{-1}\boldsymbol{A^H}$, where $\boldsymbol{A^H}$ denotes the Hermitian of matrix A. The primary aim of the subspace algorithms is to define the two orthogonal subspaces that are primarily occupied by noise and clean speech. The input to the algorithm is the noisy speech that contains the components from both spaces. The enhanced speech signal is determined by the projection of the noisy input signal onto the subspace of clean speech.

An important theorem of linear algebra is the singular value decomposition (SVD) theorem [50]. It states that every m by n matrix $\boldsymbol{A}$ can be factored as $\boldsymbol{A} = \boldsymbol{U\Sigma V^H}$ where $\boldsymbol{U}$ is an m by m unitary matrix, $\boldsymbol{V}$ is an n by n unitary matrix and $\boldsymbol{\Sigma}$ is a diagonal matrix of $diag(\sigma_1, \sigma_2, \ldots, \sigma_p)$ where $p = min(m, n)$. The diagonal elements of $\boldsymbol{\Sigma}$ are called the singular values of $\boldsymbol{A}$ and usually ordered so that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$.

Substituting the SVD of $\boldsymbol{A}$ into the orthogonal projection matrix equation $\boldsymbol{P}$, we get $\boldsymbol{P} = \boldsymbol{UU^H}$. This last equation demonstrates the two successive operations

for orthogonal projection, namely analysis and synthesis operations. Firstly, the inner product of the vector with all the basis vectors is calculated (analysis). Then the basis vectors are linearly combined to construct the representation of the vector in the subspace of A (synthesis).

When the matrix A is not full rank, i.e. it has linearly dependent columns, the SVD of $\boldsymbol{A}$ will have the form

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{H}} = [\boldsymbol{U_1}\boldsymbol{U_2}] = \begin{bmatrix} \boldsymbol{\Sigma_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V_1}^{\boldsymbol{H}} \\ \boldsymbol{V_2}^{\boldsymbol{H}} \end{bmatrix} = \boldsymbol{U_1}\boldsymbol{\Sigma_1}\boldsymbol{V_1}^{\boldsymbol{H}} \tag{4.32}$$

and the projection matrix becomes $\boldsymbol{P} = \boldsymbol{U_1}\boldsymbol{U_1}^{\boldsymbol{H}}$. The matrix that projects to the complement subspace of $\boldsymbol{P}$ is given by $\boldsymbol{I} - \boldsymbol{P}$ where $\boldsymbol{I}$ is the identity matrix. Since the matrix $\boldsymbol{U}$ is constrained to be unitary we have

$$\boldsymbol{U}\boldsymbol{U}^{\boldsymbol{H}} = \boldsymbol{I} = [\boldsymbol{U_1}\boldsymbol{U_2}] \begin{bmatrix} \boldsymbol{U_1}^{\boldsymbol{H}} \\ \boldsymbol{U_2}^{\boldsymbol{H}} \end{bmatrix} = \boldsymbol{U_1}\boldsymbol{U_1}^{\boldsymbol{H}} + \boldsymbol{U_2}\boldsymbol{U_2}^{\boldsymbol{H}} \tag{4.33}$$

So $\boldsymbol{I} - \boldsymbol{P}$ is simply $\boldsymbol{U_1}\boldsymbol{U_1}^{\boldsymbol{H}} + \boldsymbol{U_2}\boldsymbol{U_2}^{\boldsymbol{H}} - \boldsymbol{U_2}\boldsymbol{U_2}^{\boldsymbol{H}} = \boldsymbol{U_1}\boldsymbol{U_1}^{\boldsymbol{H}}$. To summarize, we can use the SVD to compose a vector $\bar{y}$ as

$$\begin{aligned} \bar{y} &= \boldsymbol{U_1}\boldsymbol{U_1}^{\boldsymbol{H}}\bar{y} + \boldsymbol{U_2}\boldsymbol{U_2}^{\boldsymbol{H}}\bar{y} \\ &= \bar{y_1} + \bar{y_2} \end{aligned} \tag{4.34}$$

In the above decomposition, $\bar{y_1}$ is the component of $\bar{y}$ that lies in the space spanned by $\boldsymbol{U_1}$ and $\bar{y_2}$ is the component of $\bar{y}$ that lies in the space spanned by $\boldsymbol{U_2}$. $\bar{y_1}$ and $\bar{y_2}$ are orthogonal to each other since the spaces spanned by $\boldsymbol{U_1}$ and $\boldsymbol{U_2}$ are orthogonal to each other.

In the context of speech enhancement, if vector $\bar{y}$ represents the noisy speech, we can use the projection matrices of $\boldsymbol{U_1}\boldsymbol{U_1}^{\boldsymbol{H}}$ and $\boldsymbol{U_2}\boldsymbol{U_2}^{\boldsymbol{H}}$ to obtain the speech and noise vectors [6].

Subspace algorithms project the noisy speech vector onto "signal subspace" using $\boldsymbol{U_1 U_1}^{\boldsymbol{H}}$ and retain only the projected vector, $\bar{y_1}$, which lies on the signal subspace. $\bar{y_2}$ is discarded as it represents the noise.

4.3. Enhanced Modified Wiener Filtering

Enhanced Modified Wiener Filtering (EMWF) algorithm proposed by this work is a modification of the MWF algorithm proposed in [7] and it is used as the speech enhancement block of our proposed unified system. The modification is based on the utilization of VAD decisions for frames. The newly proposed VAD algorithm is used in the EMWF algorithm. Compared to MWF, speech/non-speech information, which is provided by the VAD algorithm, enables the EMWF algorithm to better characterize the background noise. The EMWF algorithm employs more aggressive updates for the estimated noise spectrum during non-speech intervals. This results in more rapid convergence of the actual and estimated noise spectra. During speech intervals, the noise power spectrum estimate update is restricted within smaller ranges so that erroneous updates are minimized. The net result of these enhancements is the increased accuracy of noise spectrum estimate. The EMWF algorithm is non-iterative like the original MWF and hence it is also attractive for real-time implementation. The following section explains the basics of the EMWF algorithm.

For an additive noise signal model of $y(t) = s(t) + n(t)$, where $y(t)$ is noisy speech, $s(t)$ is noise-free speech, and $n(t)$ is noise signal, a generalized Wiener filter can be formulated as

$$H(\omega) = \left(\frac{\hat{P}_s(\omega)}{\hat{P}_s(\omega) + \alpha P_n(\omega)} \right)^{\beta} \tag{4.35}$$

where $\hat{P}_s(\omega)$ is the clean speech power spectrum estimate, $P_n(\omega)$ is the noise power spectrum, α is the noise suppression factor, and β is the power of the filter. Application of this filter to the noisy input speech signal produces an estimate for noise-free speech signal. It is assumed that the noisy speech and noise-free speech have the same phase,

so the filter just alters the amplitude at each frequency. Thus, we have

$$\hat{S}(\omega) = H(\omega)Y(\omega) \tag{4.36}$$

$$\hat{s}(t) = F^{-1}\{\hat{S}(\omega)\} \tag{4.37}$$

where $Y(\omega)$ is the Fourier transform of the noisy speech, $F^{-1}\{.\}$ is the inverse Fourier transform operation and $\hat{S}(\omega)$ is the estimate of the Fourier transform of the clean speech signal. In this formulation, it is assumed that we have an estimate of the clean speech power spectrum, $\hat{P}_s(\omega)$. This estimate is calculated from the Fourier transform of the LPC coefficients of the noisy speech, $P_y(\omega)$, by only a DC gain modification of $P_y(\omega)$ as

$$\hat{P}_s(\omega) = \frac{\hat{g_s}^2}{{g_y}^2} P_y(\omega) \tag{4.38}$$

where $\hat{g_s}$ and g_y are the DC gains of the noise-free speech signal and the noisy speech signal, respectively. EMWF algorithm assumes that noise and speech are uncorrelated and power spectra of the noisy speech signal, noise-free speech signal and noise signal are related as

$$P_y(\omega) = \hat{P}_s(\omega) + P_n(\omega) \tag{4.39}$$

If we integrate both sides of the equation over ω and use the expression for $\hat{P}_s(\omega)$ stated in Eq. (4.38) we have

$$\int_{-\pi}^{\pi} P_y(\omega)d\omega = \int_{-\pi}^{\pi} \frac{\hat{g_s}^2}{{g_y}^2} P_y(\omega)d\omega + \int_{-\pi}^{\pi} P_n(\omega)d\omega \tag{4.40}$$

Using Parseval's relation, the above equation can be simplified to

$$\frac{\hat{g_s}^2}{g_y^2} = \begin{cases} \frac{E_y - E_n}{E_y} & \text{if } E_y > E_n, \\ 0 & \text{otherwise,} \end{cases} \tag{4.41}$$

where E_n is the noise energy and E_y is the noisy speech energy. If we substitute the expression for $\hat{g_s}^2/g_y^2$ in the dc gain modification equation, the clean speech spectrum estimate becomes

$$\hat{P}_s(\omega) = \frac{E_y - E_n}{E_y} P_y(\omega) \tag{4.42}$$

Using the above expression in Eq. (4.35) and introducing a time-dependent noise suppression factor α_t we obtain

$$H(\omega) = \left(\frac{[(E_y - E_n)/E_y]P_y(\omega)}{[(E_y - E_n)/E_y]P_y(\omega) + \alpha_t P_n(\omega)} \right)^{\beta} \tag{4.43}$$

The above equation can be simplified to

$$H(\omega) = \left(\frac{P_y(\omega)}{P_y(\omega) + [E_y/(E_y - E_n)]\alpha_t P_n(\omega)} \right)^{\beta} \tag{4.44}$$

Eq. (4.44) indicates that more aggressive filtering is applied for increasing values of α_t. For proper speech enhancement, the value of α_t must be high for noise only frames and low for speech only frames. I.e. an inverse relation between the SNR value of the frame (E_s/E_n) and α_t must be introduced. This inverse relation is simply obtained by replacing α_t with E_n/E_y α' where α' is a constant. With this modification Eq. (4.44) becomes

$$H(\omega) = \left(\frac{P_y(\omega)}{P_y(\omega) + [E_n/(E_y - E_n)]\alpha' P_n(\omega)} \right)^{\beta} \tag{4.45}$$

Let us denote the time dependent multiplication factor that scales the noise spectrum,

the $[E_n/(E_y - E_n)]\alpha'$ term, by λ_t. Then the above equation is equivalent to

$$H(\omega) = \left(\frac{P_y(\omega)}{P_y(\omega) + \lambda_t P_n(\omega)} \right)^{\beta} \tag{4.46}$$

Like the original MWF method proposed in [7], EMWF assumes that the first frame is noise and the algorithm forms an initial noise spectrum estimate based on the first frame. Starting from the second frame, noise spectrum estimate is updated according to the speech/non-speech information provided by the VAD algorithm. More aggressive updates are done on the noise spectrum estimate if the current frame is decided to represent background noise. Conversely, less modification is applied on the noise spectrum estimate if the current frame is decided to represent actual speech. Utilization of speech/non-speech information to update the noise spectrum estimate enables better noise characterization compared to the "hard" noise spectrum update decisions employed in the MWF algorithm proposed in [7].

The flowchart of the proposed EMWF algorithm is shown in Figure 4.4.

Step-by-step algorithm description of the new EMWF algorithm is as follows:

• *STEP 1)* A frame length of 20 ms with a skip length of 10 ms is provided as the input to the algorithm.

• *STEP 2)* Hanning window is applied on the frame.

• *STEP 3)* Autocorrelation lags of order 18 are calculated for the input frame. Let us denote the index of the successive input frames by k. If this is not the first frame ($k \neq 0$), an interpolation factor of $\gamma = 0.7$ is applied on the autocorrelation lags of the k^{th} frame. The i^{th} autocorrelation lag for the k^{th} frame, $R[i]_k$, is set to $R[i]_k = \gamma R[i]_k + (1-\gamma) R[i]_{k-1}$. If this is the first frame ($k = 0$), autocorrelation lags are unchanged. Then, 18^{th} order LPC coefficients are calculated from the autocorrelation lags using Durbin's recursive procedure [51]. Finally, DFT of the LPC coefficients are

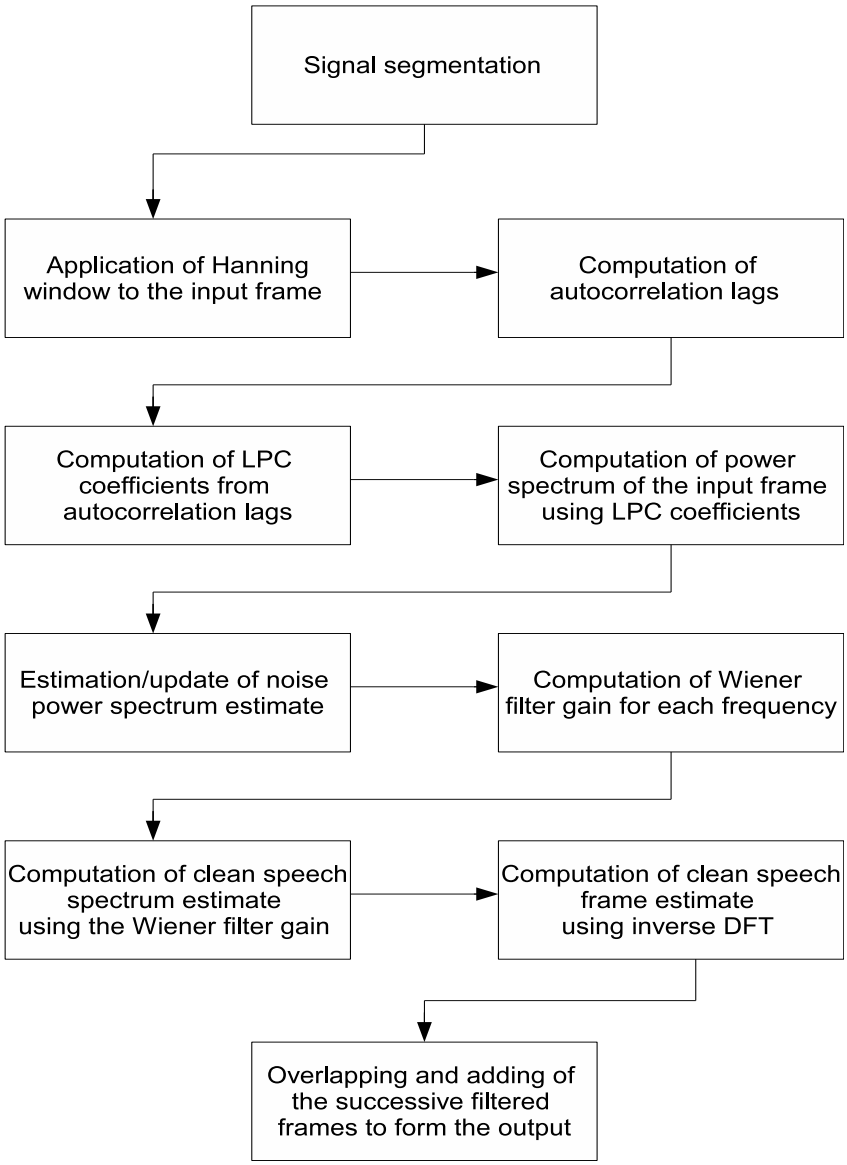

Figure 4.4. Flowchart for the proposed EMWF algorithm

calculated in order to find $P_y(\omega)_k$, power spectrum of the k^{th} noisy input frame.

• *STEP 4)* The noise power spectrum estimate for the k^{th} frame, $P_n(\omega)_k$, is calculated for each frequency. If it is the first frame ($k = 0$), it is assumed that the first frame purely denotes noise and the initial noise power spectrum estimate, $P_n(\omega)_0$, is calculated by taking DFT of 8th order LPC coefficients of the input frame. The reason for not directly setting $P_n(\omega)_0 = P_y(\omega)_0$ and using an order of 8 for LPC coefficients instead, is that LPC order of 8 results in a smoother spectrum compared to 18. Test

simulations verified that this initial smoothed spectrum results in less distortion.

If it is not the first frame ($k \neq 0$), the noise power spectrum estimate for the k^{th} frame, $P_n(\omega)_k$, is found by an update of the previous value $P_n(\omega)_{k-1}$. VAD decisions are used to convey the information of whether EMWF algorithm is operating in speech or non-speech region. Soft decision values for the last 20 frames, which are provided by the VAD algorithm, are summed. If the sum is found to be smaller than 10, then the current frame is tagged as non-speech, so aggressive updates for the noise spectrum estimate are performed. Conversely, if the sum is higher than 10, this is interpreted as we are currently in speech region, so only mild updates must be applied to minimize estimation errors. In order to adjust the level of milder updates in speech region, the algorithm keeps track of the average noise energy estimate, $\hat{E_n}$. The initial value of $\hat{E_n}$ is set to the energy of the first input frame, $(E_y)_0$. A counter j keeps track of the number of times the algorithm operates in non-speech region. If the algorithm is operating in non-speech region for the k^{th} frame, the value of $\hat{E_n}$ is updated after $P_n(\omega)_k$ is found. Update of $\hat{E_n}$ is done by setting $\hat{E_n} = [\hat{E_n} + (E_n)_k(j-1)]/\ j$, where $(E_n)_k$ denotes the noise energy for the k^{th} frame and is calculated from $P_n(\omega)_k$. The calculation of $P_n(\omega)_k$ when $k \neq 0$ is as follows:

For each frequency, the previous noise power spectrum estimate $P_n(\omega)_{k-1}$ is compared with the noisy speech power spectrum $P_y(\omega)_k$ calculated in Step 3. If $P_y(\omega)_k > P_n(\omega)_{k-1}$ and we are in a non-speech region, $P_n(\omega)_{k-1}$ is updated by an upconstant of 1.023 (10 dB/s for 10 ms skip length).

If $P_y(\omega)_k > P_n(\omega)_{k-1}$ and we are in actual speech region, a different upconstant value of $(1+\delta)$ is calculated. This upconstant is inversely related to the log energy difference between the current frame energy and average noise energy estimate, $\hat{E_n}$. The inverse relation is achieved by setting δ to $\delta = 1\ /\ 1000(\log_{10}(E_y)_k - \log_{10}\hat{E_n})$. We prefer to restrict the upwards noise update within certain limits when we are in actual speech region in order to minimize errors. Therefore, an upper limit of 1.016 (7 dB/s for 10 ms skip length) is set for $(1+\delta)$.

If $P_y(\omega)_k < P_n(\omega)_{k-1}$, $P_n(\omega)_{k-1}$ is updated by a downconstant of 0.933 (-30 dB/s for 10 ms skip length) regardless of the operating region. If $P_y(\omega)_k = P_n(\omega)_{k-1}$, no update is applied on the previous estimates and $P_n(\omega)_k$ is made equal to $P_n(\omega)_{k-1}$, again regardless of the operating region.

The update logic on the noise spectrum can be summarized as

$$P_n(\omega)_k = \begin{cases} 1.023P_n(\omega)_{k-1} & \text{if } P_y(\omega)_k > P_n(\omega)_{k-1} \text{ and we are in non-speech region} \\ (1+\delta)P_n(\omega)_{k-1} & \text{if } P_y(\omega)_k > P_n(\omega)_{k-1} \text{ and we are in speech region} \\ P_n(\omega)_{k-1} & \text{if } P_y(\omega)_k = P_n(\omega)_{k-1} \\ 0.933P_n(\omega)_{k-1} & \text{if } P_y(\omega)_k < P_n(\omega)_{k-1} \end{cases} \tag{4.47}$$

After updates, for each frequency value, $P_y(\omega)_k$ is compared with $P_n(\omega)_k$ for a sanity check. If $P_y(\omega)_k > P_n(\omega)_k$, the assignment of $P_n(\omega)_t = P_y(\omega)_t$ is made since the noise spectrum amplitude value can by no means be greater than the noisy speech spectrum amplitude.

• *STEP 5)* The time dependent λ_t factor which scales the noise spectrum in Eq. 4.46 is calculated. After test simulations, it has been determined that an exponential relation, rather than a linear relation, between the SNR and the scaling factor λ_t results in less distorted speech and the time dependent scaling factor is found as

$$\lambda_t = \left(\frac{(E_n)_k}{max\left[((E_y)_k - (E_n)_k), ((E_n)_k/50)\right]} \right)^{\mu} \nu \tag{4.48}$$

where $\mu = 0.4$ and $\nu = 63.01$ (18 dB) are heuristically determined constants.

• *STEP 6)* The Wiener filter gain for each frequency is calculated from Eq. 4.46, where $\beta = 0.5$ and the value of λ_t is determined from Step 5.

• *STEP 7)* DFT of the k^{th} noisy-speech frame, $Y(\omega)_k$, is calculated.

• *STEP 8)* The spectrum of the k^{th} noise-free speech frame, $\hat{S}(\omega)_k$, is found by multiplying $Y(\omega)_k$ with $H(\omega)_k$ at each frequency.

• *STEP 9)* The real part of the inverse DFT of $\hat{S}(\omega)_k$ is calculated to obtain the k^{th} noise-free speech frame $\hat{s}_k$.

• *STEP 10)* Overlap-add method is used for combining the filtered frames to form the overall enhanced signal output.

5. EVALUATIONS AND RESULTS

5.1. Evaluations for the Proposed VAD

Although implementing a VAD algorithm with superior speech/non-speech detection performance was not the main focus of this study, we still wanted to have a quantitative measure for the performance of the newly proposed VAD algorithm. To evaluate the performance of the newly proposed VAD algorithm, we compared the proposed algorithm against a standard VAD algorithm. ITU-T G.729 Annex B VAD [52] was chosen for comparison.

The database for the VAD performance comparison experiment was prepared by adding white Gaussian noise to clean speech signals (5 TIMIT database sentences) at SNR levels of 5, 10, 15, 20, 25, 30 dB. We used a configuration of *prespeech buffer* = 50 ms, *postspeech buffer* = 50 ms, *speech trigger* = 8, *silence trigger* = 700 ms and *sensitivity* = 3, for the proposed VAD algorithm. The samples were processed by both the proposed VAD algorithm and ITU-T G.729 Annex B VAD. In order to perform a comparative analysis, all of the input utterances were carefully endpointed by labeling the start and stop time of the individual utterances. Error analysis was performed in four different categories as

a) *ES*: extension at the start of speech (Detection of non-speech as speech at the start of the utterance),

b) *CS*: clipping at the start of speech (Detection of speech as non-speech at the start of the utterance),

c) *EE*: extension at the end of speech (Detection of non-speech as speech at the end of the utterance),

c) *CE*: clipping at the end of speech (Detection of speech as non-speech at the

end of the utterance),

Figure 5.1 shows a schematic representation of possible VAD errors that we used in measuring the performances of the two VAD algorithms.

ITU-T G.729 Annex B VAD makes frame by frame VAD decisions and small silence regions between the words or phonemes may be interpreted as non-speech. In order to make a fair error comparison with the proposed method, we emphasized the continuity in speech for ITU-T G.729 Annex B VAD and considered only the errors at the beginning or end of the sentences.

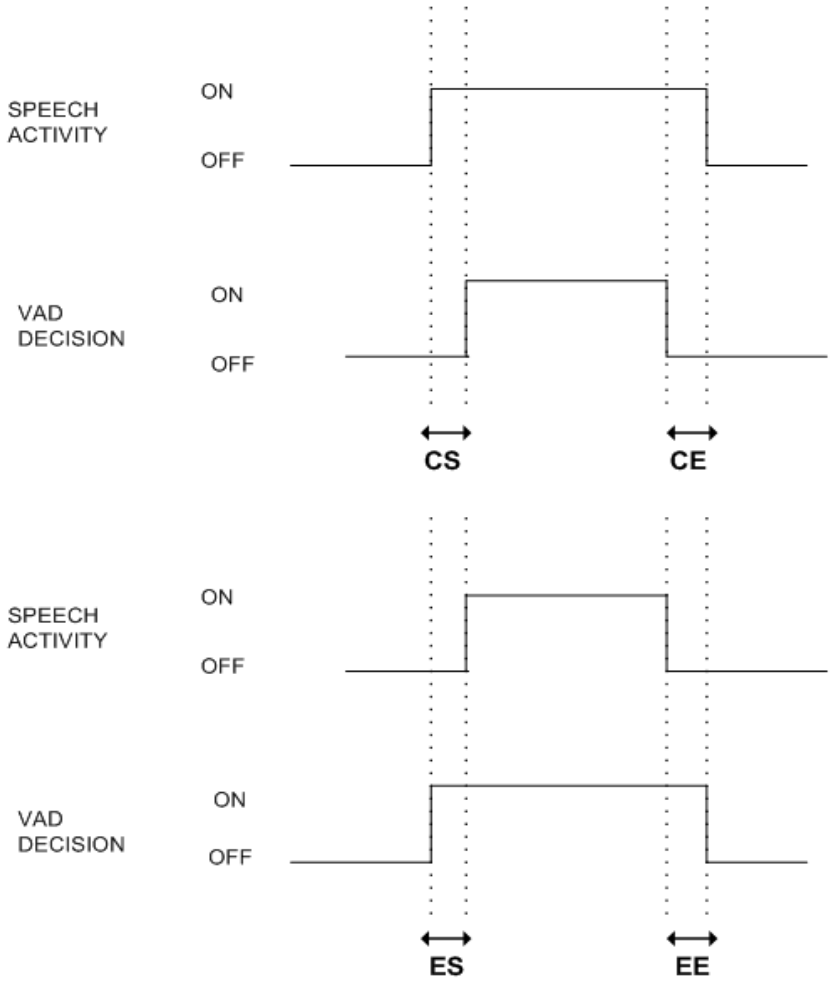

Figure 5.1. Possible VAD errors

A noteworthy result of the experiment was the observation of only extension type of noise. This was most probably due to the noise frames being detected as speech at

low SNR conditions. Table 5.1 demonstrates the average errors at each SNR level for the two VAD algorithms.

Table 5.1. Average error comparison of the proposed VAD and G.729 Annex B VAD under additive white Gaussian noise

SNR level (dB)	ES in proposed VAD (ms)	ES in G.729 Annex B VAD (ms)	EE in proposed VAD (ms)	EE in G.729 Annex B VAD (ms)
5	154.6	146.2	150.4	182.8
10	144.2	142.2	142.4	155.4
15	138.2	125.2	140.4	139.0
20	134.8	118.4	139.0	130.4
25	132.2	116.4	137.0	127.0
30	132.2	116.4	137.0	125.6

Figure 5.2 graphically shows the total error in both algorithms for varying SNR levels. As can be seen in the figure, although G.729 Annex B VAD has better performance at high SNR levels, the proposed VAD algorithm enables robust voice activity detection at low SNR conditions. Moreover, our proposed VAD algorithm makes use of only two features, so it has lower computational complexity than G.729 Annex B VAD.

5.2. Hybrid VAD Improvements

The hybrid version of the proposed VAD algorithm utilizes speech enhancement and this increases the overall VAD performance. In order to demonstrate the increased performance of the Hybrid VAD, we used 10 recordings of actual noisy speech data in Turkish that are collected in a car under different conditions. These recordings contain a total of 644 words, phrases or sentences spoken by 4 male speakers. The properties of the recordings are provided in Table 5.2.

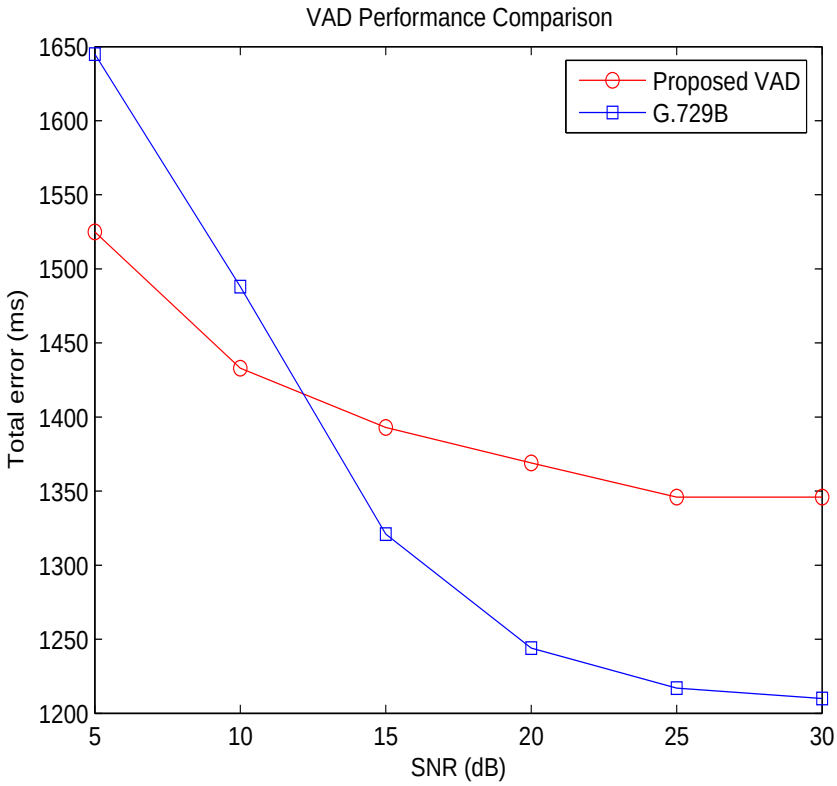

Figure 5.2. Performance comparison of the proposed VAD and G.729 Annex B VAD under additive white Gaussian noise

Using a configuration of *prespeech buffer* = 50 ms, *postspeech buffer* = 50 ms, *speech trigger* = 8, *silence trigger* = 700 ms and *sensitivity* = 3, the recordings were presented as inputs both to the proposed VAD algorithm and to its hybrid version. Among 644 words, phrases or sentences, 484 of them were successfully detected by the proposed VAD, whereas detected instance number increased to 584 for Hybrid VAD. For the common detected words, phrases or sentences, a comparative error analysis was performed.

Average errors for commonly detected utterances and total number of detected utterances per each recording for the proposed VAD are shown in Table 5.3.

Average errors for commonly detected utterances and total number of detected utterances per each recording for the Hybrid VAD are shown in Table 5.4.

Table 5.2. Properties of the recordings used for VAD performance comparison

Sample	Sample properties	Sample length (min:s)	Number of utterances
Sample 1	air conditioner on at level 2, windows open, noisy traffic	6:02	182
Sample 2	air conditioner off, windows open, traffic	2:30	46
Sample 3	air conditioner on at level 1, windows closed, traffic	2:43	47
Sample 4	air conditioner on at level 2, windows closed, car at idle	1:53	46
Sample 5	air conditioner on at level 2, windows closed, traffic	2:58	51
Sample 6	air conditioner on at level 3, windows closed, car at idle	1:51	46
Sample 7	air conditioner off, windows open, noisy traffic	3:47	90
Sample 8	air conditioner off, windows open, noisy traffic	1:26	45
Sample 9	air conditioner off, windows open, traffic	1:35	44
Sample 10	air conditioner off, windows open, traffic	1:49	47

Table 5.5 demonstrates the increased performance of the Hybrid VAD per each recording. As can be seen in Table 5.5, the total error in speech/non-speech boundary decisions for the common detected utterances are smaller in the Hybrid VAD relative to the standard VAD. Detection rate, which is computed as the ratio of the number of successfully detected utterances to the number of actual utterances is also higher for the Hybrid VAD compared to the proposed VAD. False alarm rate, which is computed as the ratio of the number of faulty detected noise only utterances to the number of actual utterances is smaller for the Hybrid VAD compared to the proposed VAD, too.

Table 5.3. Average errors for commonly detected utterances and total number of detected utterances for the proposed VAD

Sample	ES (ms)	CS (ms)	EE (ms)	CE (ms)	Number of detected utterances
Sample 1	18.25	53.96	54.82	50.46	114
Sample 2	16.55	61.52	64.43	66.45	44
Sample 3	11.04	50.44	33.69	78.78	45
Sample 4	10.05	184.28	6.31	108.18	39
Sample 5	14.60	61.06	21.72	44.96	50
Sample 6	17.20	84.09	18.48	83.95	44
Sample 7	14.00	131.26	171.52	49.78	23
Sample 8	11.64	158.92	5.83	96.69	36
Sample 9	12.33	70.42	21.09	45.81	43
Sample 10	14.46	50.59	13.17	57.70	46

Table 5.4. Average errors for commonly detected utterances and total number of detected utterances for the Hybrid VAD

Sample	ES (ms)	CS (ms)	EE (ms)	CE (ms)	Number of detected utterances
Sample 1	16.99	56.44	23.07	71.61	164
Sample 2	16.16	55.27	57.20	60.36	46
Sample 3	15.13	41.18	34.13	80.20	47
Sample 4	14.69	56.13	8.21	103.49	46
Sample 5	17.54	32.90	33.46	43.16	51
Sample 6	19.39	35.95	15.43	72.59	46
Sample 7	33.83	114.65	196.30	46.48	54
Sample 8	19.33	57.92	12.83	94.83	40
Sample 9	19.79	64.14	36.09	41.00	44
Sample 10	14.50	39.37	23.39	34.28	46

Table 5.5. Performance comparison of the proposed VAD and Hybrid VAD

Sample	Total error in proposed VAD (ms)	Total error in Hybrid VAD (ms)	Detection rate in proposed VAD (%)	Detection rate in Hybrid VAD (%)	False alarm rate in proposed VAD (%)	False alarm rate in Hybrid VAD (%)
Sample 1	20235	19165	62.6%	90.1%	2.7%	1.6%
Sample 2	9194	8316	95.7%	100.0%	0	0
Sample 3	7828	7679	95.7%	100.0%	0	0
Sample 4	12044	7118	84.8%	100.0%	0	0
Sample 5	7117	6353	98.0%	100.0%	0	0
Sample 6	8964	6308	95.7%	100.0%	0	0
Sample 7	8431	8999	25.6%	60.0%	4.4%	2.2%
Sample 8	9831	6657	80.0%	88.9%	2.2%	2.2%
Sample 9	6435	6924	97.7%	100.0%	0	0
Sample 10	6252	5131	97.9%	97.9%	0	0
Total	96331	82650	75.2%	90.7%	0.015%	0.009%

5.3. Enhanced Modified Wiener Filtering Improvements

The accuracy of the noise power spectrum estimate is the key factor for increased performance of the Wiener filtering-based speech enhancement algorithms. In order to compare the accuracy of the noise power spectrum estimates of EMWF and MWF, we first added 0.5 seconds of silence to 5 different TIMIT database utterances. Then car noise and pink noise, which were obtained from the NOISEX database, were added to the clean speech signals at two different SNR levels, 5 dB and 20 dB. Noisy signals were filtered using both of the two methods separately. At every 10th frame, starting from the first frame, estimated noise power spectra of the algorithms were compared with real noise power spectra over 512 frequency bins.

Fig. 5.3 to Fig. 5.6 demonstrate comparisons at several frame indices. As can be seen in these figures, during non-speech intervals, EMWF algorithm employs more aggressive updates for the estimated noise spectrum. This enables faster convergence of estimated noise power spectrum to real noise power spectrum. During speech intervals, noise power spectrum estimate update is restricted within smaller ranges in order to minimize errors. The net effect is the increased accuracy of noise power spectrum estimation for EMWF compared to MWF.

In order to obtain a quantitative measure for increased noise power spectrum estimation accuracy of EMWF algorithm, spectral distortion between estimated and real noise power spectra were computed at every 10th frame starting from the first frame for both of the algorithms. The spectral distortion measure, SD, is defined as

$$SD = \frac{10}{L} \sum_{i=0}^{L-1} \int_{0}^{Fs} \left[ln|P_n(\omega)| - ln|\hat{P}_n(\omega)| \right]^2 d\omega \tag{5.1}$$

where $P_n(\omega)$ is the real noise spectrum, $\hat{P}_n(\omega)$ is the estimated noise spectrum and L is the number of frames used in computation. Spectral distortion measures are tabulated in Table 5.6. As can be seen in Table 5.6, average spectral distortion measures are smaller for EMWF compared to MWF. This implies increased accuracy of noise spectrum estimation for EMWF algorithm compared to MWF algorithm.

Another experiment was performed in order to compare the objective quality measures of MWF and EMWF outputs. The database for this experiment was prepared by adding white Gaussian noise to clean speech signals (20 TIMIT database utterances) at 10 dB SNR. This database was then used to compare the PESQ (Perceptual Evaluation of Speech Quality) scores of the noisy and enhanced speech signals, where the enhancement was implemented by applying MWF and EMWF algorithms to the noisy signals. PESQ scores are shown in Table 5.7. As can be seen in the results, both algorithms increase the speech quality and EMWF achieves better PESQ scores compared to original MWF.

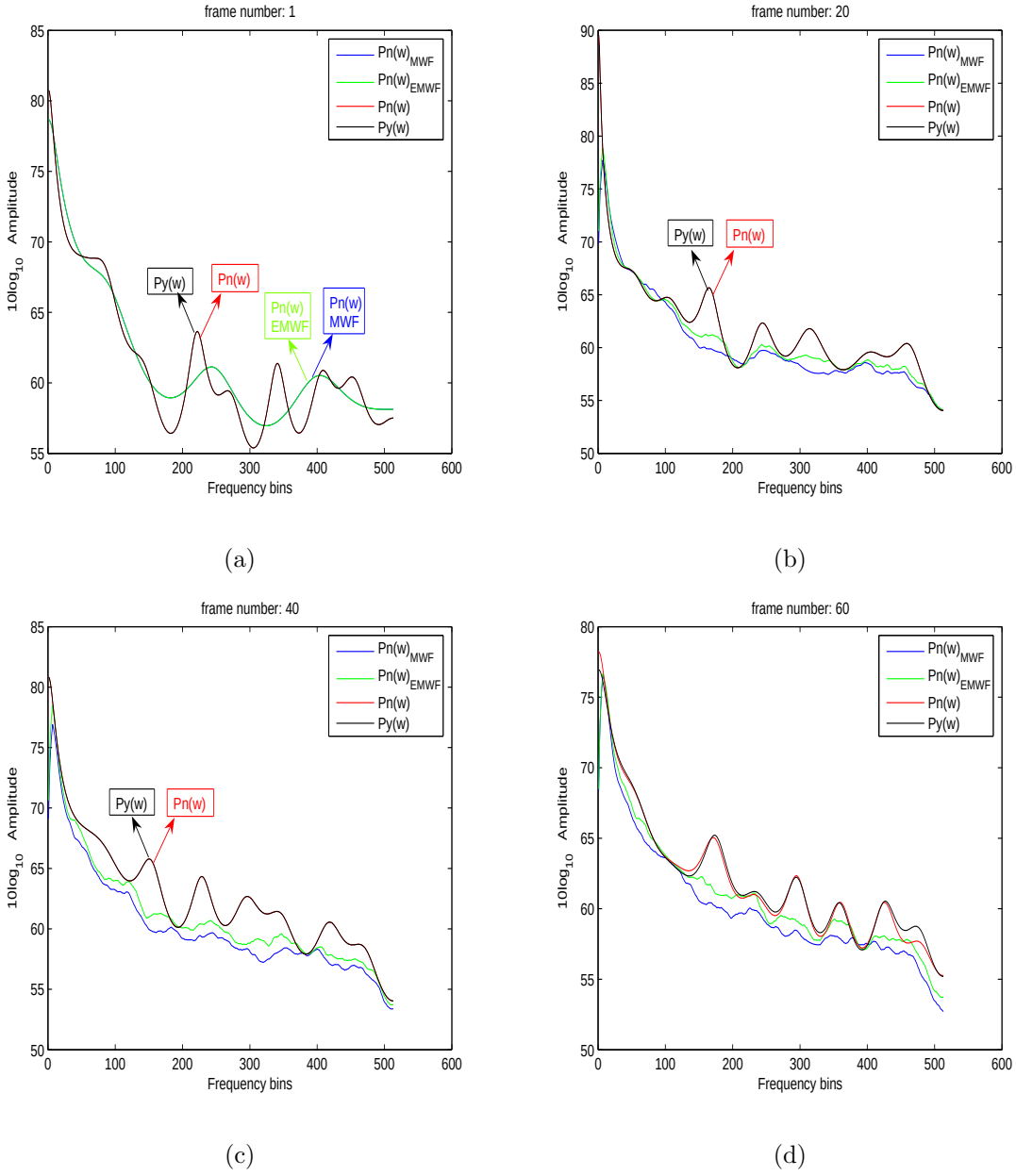

Figure 5.3. Comparison of noise power spectrum estimate of MWF ($Pn(w)_{\mathrm{MWF}}$), noise power spectrum estimate of EMWF ($Pn(w)_{\mathrm{EMWF}}$), real noise power spectrum ($Pn(w)$) and noisy input speech power spectrum ($Py(w)$) for Sample 1 with 20 dB pink noise at frames (a) 1, (b) 20, (c) 40 and (d) 60

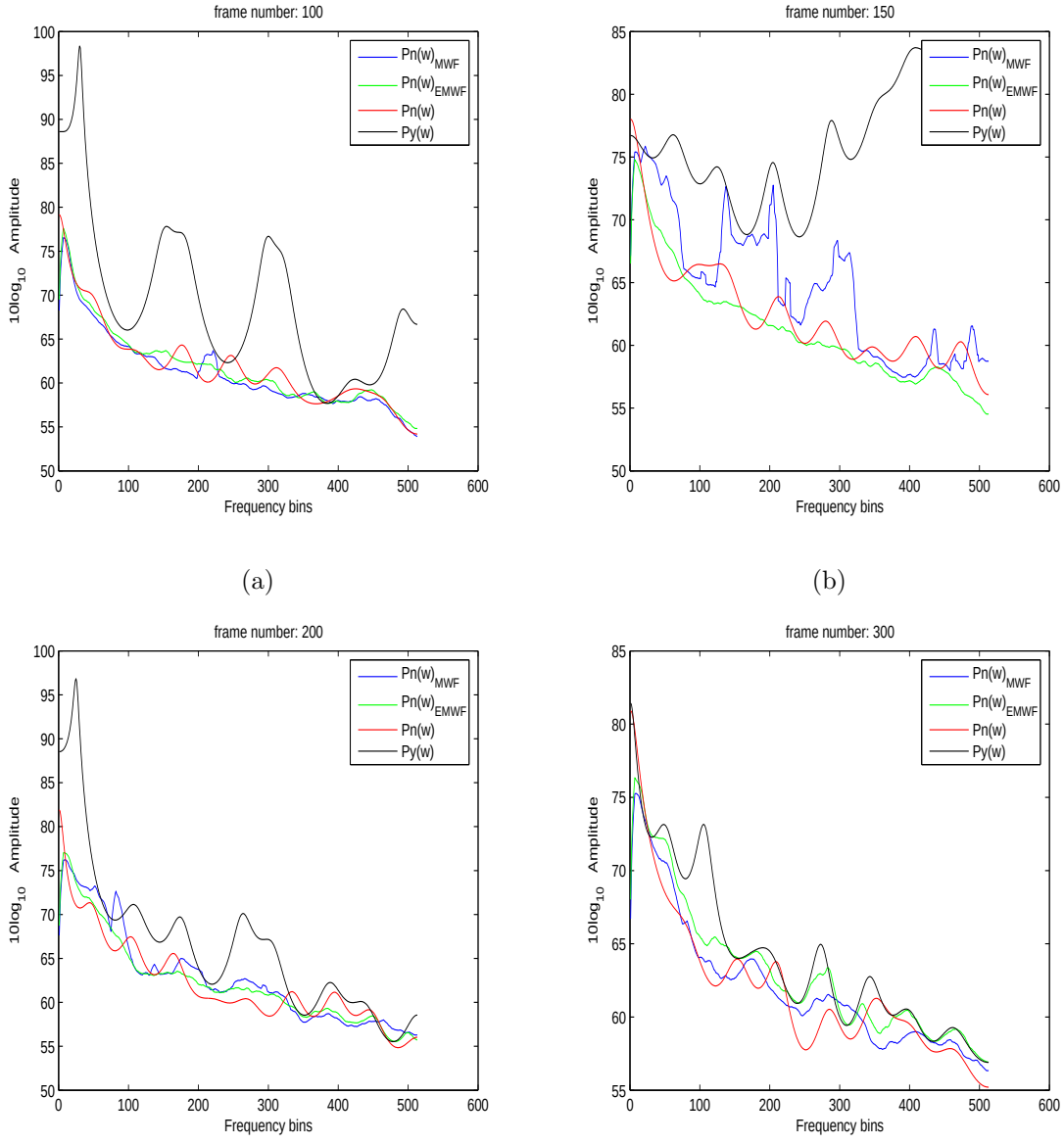

Figure 5.4. Comparison of noise power spectrum estimate of MWF ($Pn(w)_{\mathrm{MWF}}$), noise power spectrum estimate of EMWF ($Pn(w)_{\mathrm{EMWF}}$), real noise power spectrum ($Pn(w)$) and noisy input speech power spectrum ($Py(w)$) for Sample 1 with 20 dB pink noise at frames (a) 100, (b) 150, (c) 200 and (d) 300

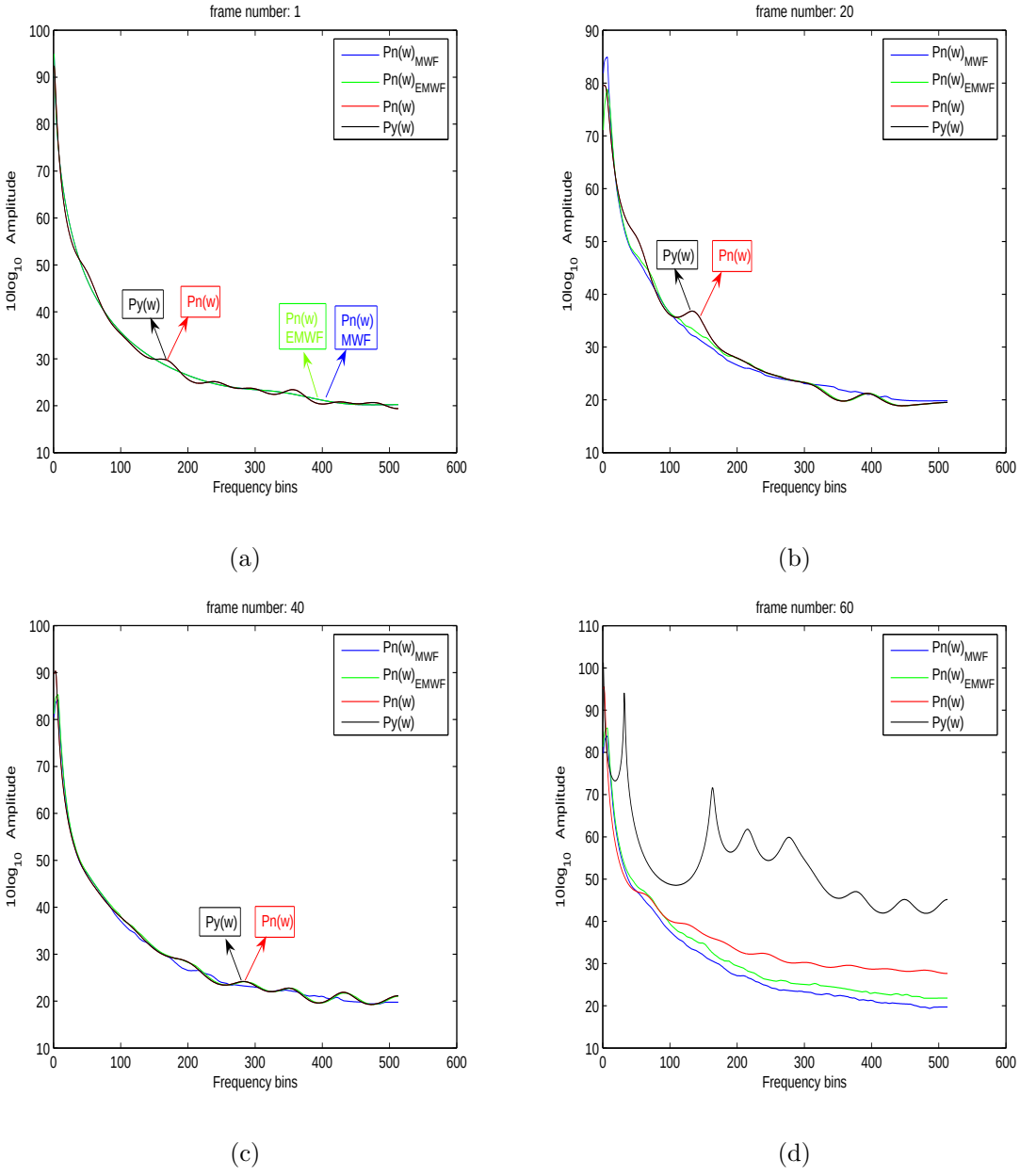

Figure 5.5. Comparison of noise power spectrum estimate of MWF ($Pn(w)_{\mathrm{MWF}}$), noise power spectrum estimate of EMWF ($Pn(w)_{\mathrm{EMWF}}$), real noise power spectrum ($Pn(w)$) and noisy input speech power spectrum ($Py(w)$) for Sample 2 with 20 dB car noise at frames (a) 1, (b) 20, (c) 40 and (d) 60

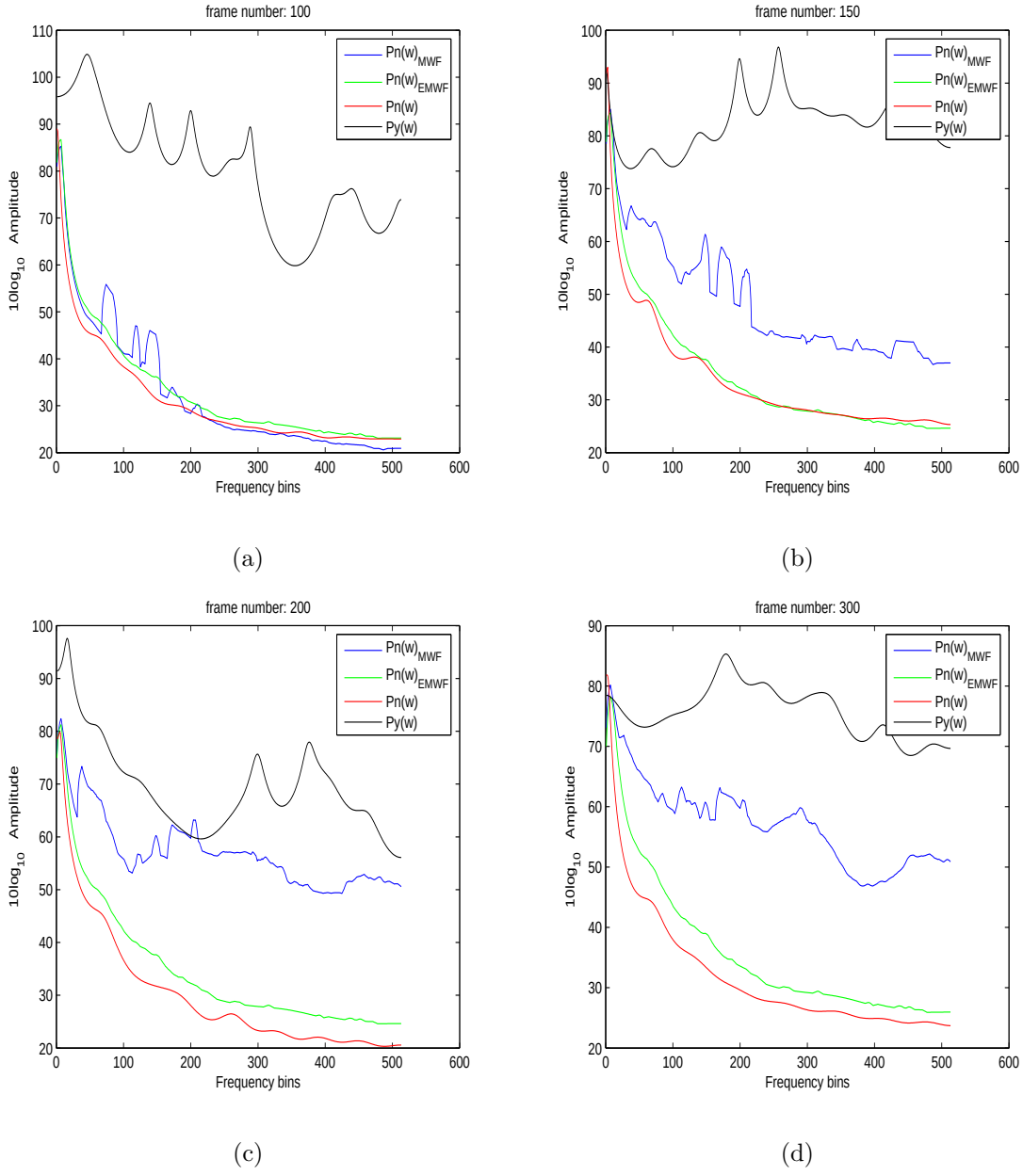

Figure 5.6. Comparison of noise power spectrum estimate of MWF ($Pn(w)_{\mathrm{MWF}}$), noise power spectrum estimate of EMWF ($Pn(w)_{\mathrm{EMWF}}$), real noise power spectrum ($Pn(w)$) and noisy input speech power spectrum ($Py(w)$) for Sample 2 with 20 dB car noise at frames (a) 100, (b) 150, (c) 200 and (d) 300

Table 5.6. Comparison of spectral distortion measures for noise power spectrum estimation

Sample used	SD for MWF	SD for EMWF
Sample 1 at 5dB car noise	$13.843 * 10^4$	$9.208 * 10^4$
Sample 1 at 5dB pink noise	$25.375 * 10^4$	$24.748 * 10^4$
Sample 1 at 20dB car noise	$10.603 * 10^4$	$4.545 * 10^4$
Sample 1 at 20dB pink noise	$16.772 * 10^4$	$15.921 * 10^4$
Sample 2 at 5dB car noise	$15.540 * 10^4$	$10.721 * 10^4$
Sample 2 at 5dB pink noise	$27.294 * 10^4$	$26.663 * 10^4$
Sample 2 at 20dB car noise	$12.190 * 10^4$	$5.579 * 10^4$
Sample 2 at 20dB pink noise	$17.667 * 10^4$	$17.183 * 10^4$
Sample 3 at 5dB car noise	$17.771 * 10^4$	$9.420 * 10^4$
Sample 3 at 5dB pink noise	$23.898 * 10^4$	$22.824 * 10^4$
Sample 3 at 20dB car noise	$9.478 * 10^4$	$4.952 * 10^4$
Sample 3 at 20dB pink noise	$15.091 * 10^4$	$14.685 * 10^4$
Sample 4 at 5dB car noise	$13.064 * 10^4$	$11.597 * 10^4$
Sample 4 at 5dB pink noise	$29.204 * 10^4$	$28.420 * 10^4$
Sample 4 at 20dB car noise	$10.769 * 10^4$	$6.441 * 10^4$
Sample 4 at 20dB pink noise	$18.472 * 10^4$	$17.856 * 10^4$
Sample 5 at 5dB car noise	$13.782 * 10^4$	$9.251 * 10^4$
Sample 5 at 5dB pink noise	$26.030 * 10^4$	$25.112 * 10^4$
Sample 5 at 20dB car noise	$10.292 * 10^4$	$4.777 * 10^4$
Sample 5 at 20dB pink noise	$16.996 * 10^4$	$15.685 * 10^4$

Table 5.7. PESQ scores of noisy and enhanced speech signals for white Gaussian noise with 10 dB SNR

Sample	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Sample 1	2.224	2.271	2.285
Sample 2	2.294	2.463	2.589
Sample 3	2.297	2.364	2.391
Sample 4	1.730	2.241	2.254
Sample 5	2.568	2.530	2.862
Sample 6	1.897	2.015	2.033
Sample 7	2.275	2.342	2.440
Sample 8	2.015	2.371	2.435
Sample 9	2.145	2.412	2.498
Sample 10	2.346	2.246	2.630
Sample 11	2.033	2.121	2.221
Sample 12	2.105	2.310	2.359
Sample 13	1.975	1.948	2.373
Sample 14	2.408	2.303	2.349
Sample 15	2.357	2.376	2.504
Sample 16	2.096	2.207	2.202
Sample 17	2.182	2.154	2.203
Sample 18	2.294	2.489	2.584
Sample 19	1.928	2.299	2.389
Sample 20	2.083	2.067	2.393
Mean	2.163	2.276	2.400

We also wanted to evaluate the performance increase in the objective quality for different SNR levels. For that purpose, we added speech babble, car, pink and white noise to 5 TIMIT database sentences at SNR levels of 5, 10, 15, 20, 25 and 30 dB using the NOISEX database. Mean PESQ scores of noisy speech and MWF and EMWF outputs were computed for each noise type. Fig. 5.7 demonstrates the results. As can be seen in Fig. 5.7, the performance improvement of EMWF is more explicit at low SNR levels.

Finally, we performed another experiment to compare the subjective quality measures of MWF and EMWF outputs. 10 utterances collected in a car driven in traffic are used as the test database. MOS tests were performed on this database to evaluate the subjective speech quality measures of MWF and EMWF outputs. In the MOS test, 15 subjects (3 females) were used. The subjects listened the original noisy samples and filtered samples (MWF and EMWF outputs) using headphones. They were instructed to rate the sentences on a scale of 1-5 where 1 is very poor and 5 is excellent. Some speech samples of speech coders having different MOS scores were presented to the subjects to ensure consistency in evaluating the speech quality. Average MOS scores per each subject are shown in Table 5.8. As can be seen in the results, both algorithms increase the subjective speech quality and EMWF achieves a slightly better performance compared to original MWF. Utterances used for the MOS tests and detailed scores for each subject is given in Appendix A.

5.4. Unified System Evaluations

For unified system performance evaluation, we investigated the performance of the system as a preprocessor to a speech recognition engine.

The speech recognizer that we used for unified system performance evaluation was the hidden Markov model-based (HMM-based), SPHINX system [53]. SPHINX uses LPC-derived parameters and triphones to model the sound units. In the triphone model, the left and the right neighboring phones are considered [54]. The HMM in SPHINX models the triphones as having a sequence of states, where each state has its

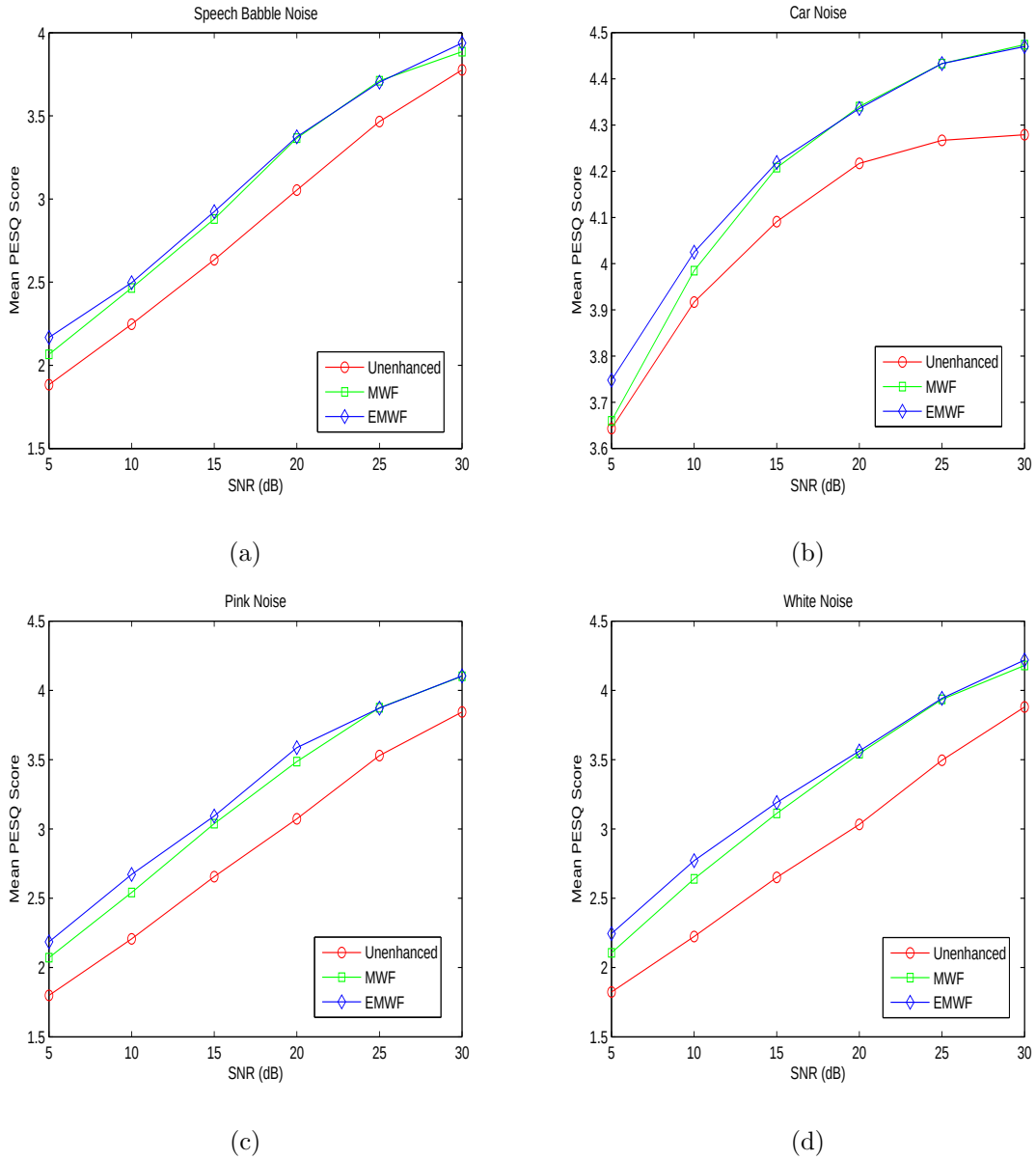

Figure 5.7. PESQ performance of MWF and EMWF outputs compared to noisy speech for (a) speech babble noise, (b) car noise, (c) pink noise, (d) white noise

Table 5.8. Average MOS Scores on a scale from 1 to 5 over 10 utterances recorded in a car driven in traffic for 3 different conditions: (i) Original noisy speech, (ii) enhanced speech using MWF and (iii) enhanced speech using EMWF.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	2.56	2.97	3.05
Subject 2	2.00	2.34	2.39
Subject 3	2.91	3.39	3.43
Subject 4	2.63	3.07	3.09
Subject 5	2.46	2.91	2.91
Subject 6	2.91	3.36	3.43
Subject 7	2.74	3.17	3.22
Subject 8	2.43	2.87	2.87
Subject 9	2.17	2.59	2.61
Subject 10	3.07	3.54	3.55
Subject 11	2.51	2.93	2.97
Subject 12	2.13	2.46	2.49
Subject 13	2.65	3.08	3.11
Subject 14	2.85	3.30	3.35
Subject 15	1.87	2.35	2.40
Mean	2.53	2.96	2.99
St. Dev.	0.36	0.38	0.38

own unique distribution. The parameters of the state distributions must be learned from training data, and recognition using these models involves a fairly computationally intensive decoding algorithm, where the computer evaluates HMMs for several hypotheses in order to determine the most probable one [55].

Speech recognition performances were measured for both actual and artificially noise added noisy speech samples. For actual noise performance comparison, the same 10 recordings of actual noisy speech data in Turkish, that we used in Section 5.2, were utilized. For artificially added noise performance comparison, 30 different TIMIT database sentences (Appendix B provides used samples) were grouped into 24 different sets, where each set was degraded with a certain type of noise at a certain SNR level. The NOISEX database was used to add speech babble, car, pink and white noise types to the samples at SNR levels of 5, 10, 15, 20, 25 and 30 dB. Noisy signals were first processed separately by the proposed VAD algorithm and the unified system in order to eliminate the silence regions. The processed signals were then presented as inputs to the speech recognizer.

Table 5.9 and Table 5.10 demonstrate the increased recognition performance of the unified system compared to the proposed VAD system. Recognition performance increased from 65.2 per cent to 83.2 per cent for actual noisy samples and from 87.5 per cent to 91.3 per cent for artificially noise added samples. As can be seen from the results, unified system offers significant performance increase in speech recognition at low SNR levels due to the speech enhancement capability embedded in the system.

Table 5.9. Unified System evaluations for samples in Turkish (actual noise)

Sample	Recognition rate in standard VAD (%)	Recognition rate in Unified System (%)
Sample 1	40.1%	71.4%
Sample 2	91.3%	100.0%
Sample 3	93.6%	95.7%
Sample 4	76.1%	100.0%
Sample 5	94.1%	98.0%
Sample 6	91.3%	100.0%
Sample 7	23.9%	54.5%
Sample 8	65.9%	79.5%
Sample 9	90.9%	95.5%
Sample 10	93.6%	95.7%
Total	65.2%	83.2%

Finally, we wanted to evaluate the relative contributions of the speech enhancement and VAD blocks for increasing the speech recognition performance of the unified system. We also wanted to compare MWF and EMWF algorithms in terms of speech recognition performance. For this purpose, we added white noise to 30 different TIMIT database sentences at SNR levels of 5, 10, 15, 20, 25 and 30 dB. The proposed VAD algorithm, Hybrid VAD algorithm (VAD block of the unified system), MWF algorithm, EMWF algorithm (speech enhancement block of the unified system) and the unified system were separately used as preprocessors to a SPHINX speech recognition engine. The overall recognition rates were 77.2 percent for the proposed VAD algorithm, 74.3 percent for Hybrid VAD algorithm, 86.1 percent for MWF algorithm, 87.2 percent for EMWF algorithm and 88.3 percent for the unified system. As can be inferred from these results, unified system achieves the best recognition performance. It is mainly the speech enhancement block of the unified system that enables increased speech recognition performance for the unified system. Compared to MWF, EMWF algorithm slightly improves speech recognition performance and the performance improvement

of EMWF is more explicit at low SNR conditions. Fig. 5.8 demonstrates the detailed recognition results of the algorithms for varying SNR levels.

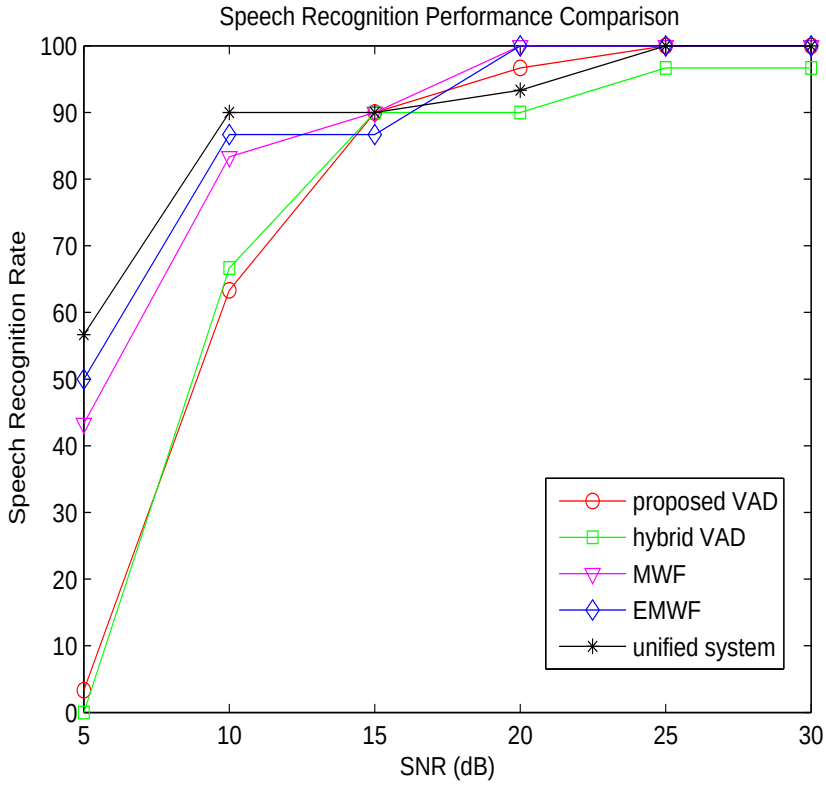

Figure 5.8. Comparison of recognition results at varying SNR levels

Table 5.10. Unified System evaluations for samples in English (artificially added noise)

Sample set	Noise characteristics	Recognition rate in standard VAD (%)	Recognition rate in Unified System (%)
Set1	speech babble, SNR 5dB	73.3%	90.0%
Set2	speech babble, SNR 10dB	100.0%	93.3%
Set3	speech babble, SNR 15dB	100.0%	90.0%
Set4	speech babble, SNR 20dB	100.0%	86.7%
Set5	speech babble, SNR 25dB	100.0%	90.0%
Set6	speech babble, SNR 30dB	96.7%	90.0%
Set7	car noise, SNR 5dB	100.0%	100.0%
Set8	car noise, SNR 10dB	100.0%	96.7%
Set9	car noise, SNR 15dB	100.0%	96.7%
Set10	car noise, SNR 20dB	96.7%	96.7%
Set11	car noise, SNR 25dB	96.7%	96.7%
Set12	car noise, SNR 30dB	96.7%	96.7%
Set13	pink noise, SNR 5dB	6.7%	70.0%
Set14	pink noise, SNR 10dB	73.3%	83.3%
Set15	pink noise, SNR 15dB	96.7%	90.0%
Set16	pink noise, SNR 20dB	100.0%	96.7%
Set17	pink noise, SNR 25dB	100.0%	96.7%
Set18	pink noise, SNR 30dB	100.0%	100.0%
Set19	white noise, SNR 5dB	3.3%	56.7%
Set20	white noise, SNR 10dB	63.3%	90.0%
Set21	white noise, SNR 15dB	96.7%	90.0%
Set22	white noise, SNR 20dB	100.0%	93.3%
Set23	white noise, SNR 25dB	100.0%	100.0%
Set24	white noise, SNR 30dB	100.0%	100.0%
Total		87.5%	91.3%

6. CONCLUSIONS

In this thesis, we follow a unified approach for VAD and speech enhancement problems. We demonstrate that the two problems are interrelated and implement a unified system for VAD and speech enhancement. The Hybrid VAD and EMWF algorithms constitute the VAD and speech enhancement block of the proposed unified system, respectively.

A new, robust and low complexity VAD algorithm, which has reliable performance at low SNR levels, is proposed. The proposed VAD algorithm uses a periodicity measure and an energy measure which is computed according to the spectral properties of human speech. The new algorithm associates a soft decision value, rather than a strict speech/non-speech decision, with each frame to indicate the speech likeliness of that frame. Final speech/non-speech decision is based on a history of soft decision values. Employing speech enhancement in the form of MWF is shown to improve the performance of the proposed VAD algorithm. The Hybrid VAD, that utilizes speech enhancement, not only decreased the average error but also increased the utterance detection rate from 75.2 per cent to 90.7 per cent.

Benefits of utilizing VAD for speech enhancement are demonstrated by implementing the EMWF algorithm. EMWF algorithm results in less spectral distortion of noise power spectrum estimate compared to the standard MWF algorithm. Increased noise power spectrum estimation accuracy of EMWF relative to MWF enables EMWF algorithm to employ a more aggressive enhancement at non-speech intervals, and a milder filtering at the speech segments compared to MWF. This provides better speech enhancement. For comparison, EMWF algorithm is tested under simulated and actual car noise conditions and is shown to outperform the standard MWF in both subjective and objective speech quality evaluations.

Finally, the unified system is evaluated as a preprocessor to a speech recognition engine where actual noisy and artificially noise added signals are used in the experi-

ment. Compared to the single VAD system, the usage of the unified system enabled performance increase in speech recognition rates, especially at low SNR levels. Usage of the unified system instead of the single VAD system increased the recognition performance from 65.2 per cent to 83.2 per cent for actual noisy samples and from 87.5 per cent to 91.3 per cent for artificially noise added samples. It is also demonstrated that it is the speech enhancement block of the unified system that enables increased speech recognition performance for the unified system.

6.1. Remarks and Future Directions

The main contributions from this thesis fall into two areas which include (i) the demonstration of the high correlation between VAD and speech enhancement problems, and (ii) implementation of a new, robust and low complexity VAD algorithm.

Features used in the proposed VAD algorithm are carefully chosen according to their computational complexity and their discriminative contribution to the ultimate speech/non-speech decision. More features may be included in the algorithm to improve performance as long as real-time operation capability is not impaired. As explained in the detailed algorithm description, determination of the values of some of the constants used in the proposed VAD algorithm and the configuration parameters rely on heuristic observations. By performing more observations, these values may be better tuned. Yet another desirable property may be the dynamic changing of the configuration parameters internally according to the operating conditions.

In this thesis, we elaborated the joint usage of the new VAD and MWF algorithms for the proposed unified system. However, it is expected that as long as the mutual information exchange is successfully established, the joint usage of any kind of VAD and speech enhancement algorithms enables performance improvements. Integration of different VAD and speech enhancement algorithms may be a future area of study.

APPENDIX A: Utterances Used for MOS Tests and Detailed Scores

Test Samples

Sample 1: yüzbeş nokta sekiz

Sample 2: en yakın sinema nerede

Sample 3: amaçlarının ne olduğunu

Sample 4: yörelerinde bulunmaktadır

Sample 5: insanların arasındayken

Sample 6: sekiz olarak bildirilmektedir

Sample 7: bundan sonra da kendisine

Sample 8: seksen sekiz nokta bir

Sample 9: seksen dokuz nokta iki

Sample 10: doksan nokta üç

Detailed MOS Scores for Each Sample

Table A.1. MOS Scores for Sample 1.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	2.3	3.1	3.1
Subject 2	1.8	2.2	2.1
Subject 3	2.6	3.3	3.5
Subject 4	2.4	3.0	3.1
Subject 5	2.3	2.8	2.9
Subject 6	2.7	3.3	3.5
Subject 7	2.5	3.2	3.3
Subject 8	2.2	2.8	2.8
Subject 9	2.0	2.4	2.5
Subject 10	2.8	3.5	3.5
Subject 11	2.2	3.0	3.0
Subject 12	1.9	2.1	2.2
Subject 13	2.4	3.1	3.1
Subject 14	2.6	3.2	3.4
Subject 15	1.7	2.0	2.0
Mean	2.29	2.87	2.93
St. Dev.	0.33	0.47	0.52

Table A.2. MOS Scores for Sample 2.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	2.1	2.8	2.9
Subject 2	1.5	1.9	2.0
Subject 3	2.5	3.0	2.9
Subject 4	2.2	2.8	2.8
Subject 5	2.0	2.7	2.7
Subject 6	2.5	3.1	3.1
Subject 7	2.3	2.9	2.9
Subject 8	2.1	2.7	2.7
Subject 9	2.0	2.5	2.5
Subject 10	2.7	3.4	3.6
Subject 11	2.0	2.7	2.7
Subject 12	1.7	2.2	2.3
Subject 13	2.2	2.8	2.8
Subject 14	2.4	3.0	3.0
Subject 15	1.5	2.0	2.0
Mean	2.11	2.70	2.73
St. Dev.	0.35	0.41	0.41

Table A.3. MOS Scores for Sample 3.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	2.0	2.5	2.7
Subject 2	1.5	1.8	1.9
Subject 3	2.5	3.0	3.3
Subject 4	2.1	2.7	2.8
Subject 5	2.0	2.6	2.6
Subject 6	2.5	3.1	3.3
Subject 7	2.2	2.8	3.0
Subject 8	2.0	2.5	2.6
Subject 9	1.8	2.3	2.4
Subject 10	2.7	3.4	3.4
Subject 11	2.0	2.4	2.5
Subject 12	1.6	2.0	2.2
Subject 13	2.1	2.7	2.8
Subject 14	2.4	3.0	3.2
Subject 15	1.5	2.0	2.0
Mean	2.06	2.59	2.71
St. Dev.	0.36	0.45	0.47

Table A.4. MOS Scores for Sample 4.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	2.2	2.6	2.7
Subject 2	1.7	2.1	2.0
Subject 3	2.4	3.0	3.0
Subject 4	2.2	2.7	2.6
Subject 5	2.1	2.6	2.7
Subject 6	2.4	3.0	3.1
Subject 7	2.3	2.8	2.8
Subject 8	2.1	2.6	2.6
Subject 9	1.9	2.4	2.4
Subject 10	2.7	3.2	3.3
Subject 11	2.1	2.5	2.5
Subject 12	2.0	2.3	2.2
Subject 13	2.2	2.7	2.7
Subject 14	2.3	2.9	3.0
Subject 15	1.5	2.0	2.5
Mean	2.14	2.63	2.67
St. Dev.	0.29	0.33	0.34

Table A.5. MOS Scores for Sample 5.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	3.2	3.5	3.4
Subject 2	2.3	2.8	3.0
Subject 3	3.4	3.7	3.7
Subject 4	3.2	3.4	3.4
Subject 5	2.9	3.2	3.2
Subject 6	3.4	3.6	3.6
Subject 7	3.3	3.6	3.5
Subject 8	2.8	3.2	3.2
Subject 9	2.5	2.9	3.0
Subject 10	3.5	3.7	3.7
Subject 11	3.0	3.3	3.4
Subject 12	2.5	2.9	2.8
Subject 13	3.1	3.5	3.5
Subject 14	3.3	3.6	3.5
Subject 15	2.0	2.5	2.5
Mean	2.96	3.29	3.29
St. Dev.	0.45	0.37	0.34

Table A.6. MOS Scores for Sample 6.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	3.2	3.5	3.7
Subject 2	2.3	2.7	2.7
Subject 3	3.5	4.0	4.0
Subject 4	3.2	3.5	3.6
Subject 5	2.9	3.3	3.3
Subject 6	3.5	3.8	3.7
Subject 7	3.4	3.7	3.8
Subject 8	2.8	3.2	3.2
Subject 9	2.4	2.9	2.8
Subject 10	3.5	3.9	3.8
Subject 11	3.1	3.4	3.5
Subject 12	2.5	2.9	2.9
Subject 13	3.3	3.6	3.7
Subject 14	3.4	3.8	3.8
Subject 15	2.0	2.5	2.5
Mean	3.00	3.38	3.40
St. Dev.	0.49	0.46	0.47

Table A.7. MOS Scores for Sample 7.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	3.0	3.5	3.5
Subject 2	3.0	3.4	3.3
Subject 3	3.2	3.6	3.6
Subject 4	3.0	3.5	3.4
Subject 5	2.9	3.4	3.3
Subject 6	3.0	3.5	3.6
Subject 7	3.1	3.5	3.5
Subject 8	2.9	3.4	3.3
Subject 9	2.6	3.1	3.0
Subject 10	3.3	3.7	3.7
Subject 11	3.0	3.6	3.6
Subject 12	2.8	3.2	3.2
Subject 13	3.1	3.6	3.6
Subject 14	3.1	3.5	3.5
Subject 15	2.5	3.0	3.0
Mean	2.97	3.43	3.41
St. Dev.	0.21	0.20	0.22

Table A.8. MOS Scores for Sample 8.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	2.5	2.7	2.7
Subject 2	1.8	2.1	2.3
Subject 3	3.0	3.3	3.5
Subject 4	2.7	2.9	3.0
Subject 5	2.5	2.8	2.8
Subject 6	2.9	3.3	3.3
Subject 7	2.8	3.0	3.1
Subject 8	2.4	2.7	2.7
Subject 9	2.1	2.4	2.5
Subject 10	3.1	3.5	3.5
Subject 11	2.5	2.8	2.8
Subject 12	2.0	2.2	2.3
Subject 13	2.7	2.9	3.0
Subject 14	3.0	3.2	3.4
Subject 15	2.0	2.5	2.5
Mean	2.53	2.82	2.89
St. Dev.	0.41	0.41	0.41

Table A.9. MOS Scores for Sample 9.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	2.4	2.6	2.7
Subject 2	1.9	2.0	2.0
Subject 3	3.0	3.5	3.3
Subject 4	2.6	2.9	2.8
Subject 5	2.4	2.8	2.7
Subject 6	3.1	3.5	3.7
Subject 7	2.7	3.0	3.0
Subject 8	2.4	2.7	2.7
Subject 9	2.1	2.5	2.4
Subject 10	3.1	3.5	3.5
Subject 11	2.6	2.8	2.9
Subject 12	1.8	2.0	2.0
Subject 13	2.7	2.9	2.8
Subject 14	3.0	3.4	3.3
Subject 15	2.0	2.5	2.5
Mean	2.52	2.84	2.82
St. Dev.	0.43	0.49	0.49

Table A.10. MOS Scores for Sample 10.

Listener	Noisy speech	Enhanced speech with MWF	Enhanced speech with EMWF
Subject 1	2.7	2.9	3.1
Subject 2	2.2	2.4	2.6
Subject 3	3.0	3.5	3.5
Subject 4	2.7	3.3	3.4
Subject 5	2.6	2.9	2.9
Subject 6	3.1	3.4	3.4
Subject 7	2.8	3.2	3.3
Subject 8	2.6	2.9	2.9
Subject 9	2.3	2.5	2.6
Subject 10	3.3	3.6	3.5
Subject 11	2.6	2.8	2.8
Subject 12	2.5	2.8	2.8
Subject 13	2.7	3.0	3.1
Subject 14	3.0	3.4	3.4
Subject 15	2.0	2.5	2.5
Mean	2.67	3.01	3.05
St. Dev.	0.35	0.38	0.35

APPENDIX B: TIMIT Sentences Used for Speech Recognition

Sample 1: Tim takes Sheila to see movies twice a week.

Sample 2: Even then, if she took one step forward he could catch her.

Sample 3: The kid has no manners, boys.

Sample 4: Soon the office work claimed all her time.

Sample 5: Don't ask me to carry an oily rag like that.

Sample 6: A lawyer was appointed to execute her will.

Sample 7: Change involves the displacement of form.

Sample 8: Each stag surely finds a big fawn.

Sample 9: A note of awe came into his voice.

Sample 10: Add remaining ingredients and bring to a boil.

Sample 11: As these maladies overlap, so must the cure.

Sample 12: Those musicians harmonize marvelously.

Sample 13: Religion thus becomes integrated with life.

Sample 14: Now he'll choke for sure.

Sample 15: The misprint provoked an immediate disclaimer.

Sample 16: Eating spinach nightly increases strength miraculously.

Sample 17: Military personnel are expected to obey government orders.

Sample 18: A chosen few will become Generals.

Sample 19: Combine all the ingredients in a large bowl.

Sample 20: She is thinner than I am.

Sample 21: There are also honest seekers after truth.

Sample 22: Receiving no answer they set the fire.

Sample 23: She spouted a mouthful of water into the air.

Sample 24: What shall these effects be?

Sample 25: Shaving cream is a popular item on Halloween.

Sample 26: How good is your endurance?

Sample 27: Not surprisingly, this approach did not work.

Sample 28: Highway and freeway mean the same thing.

Sample 29: Even a simple vocabulary contains symbols.

Sample 30: Pam gives driving lessons on Thursdays.

REFERENCES

1. Lee, H. H. and C. K. Un, "A Study of On-Off Characteristics of Conversational Speech", *IEEE Transactions on Communications*, Vol. 54, No. 6, pp. 630-637, May 1987.

2. Lee, I. D., H. P. Stern and S.A. Mahmoud, "A voice activity detection algorithm for communication systems with dynamically varying background acoustic noise", *Proceedings of the IEEE International Conference on Vehicular Technology*, Vol. 2, No. 1, pp. 1214 - 1218, May 1998.

3. Beritelli, F., S. Casale and S. Serrano, "A low-complexity speech-pause detection algorithm for communication in noisy environments", *European Transactions on Telecommunications*, Vol. 15, No. 1, pp. 33-38, January 2004.

4. James, J., B. Chen and L. Garrison, "Implementing VoIP: A Voice Transmission Performance Progress Report", *IEEE Communications Magazine*, Vol. 42, No. 7, pp. 36-41, July 2004.

5. Ephraim, Y., "Statistical-model-based speech enhancement systems", *Proceedings of IEEE*, Vol. 80, No. 10, pp. 1526-1555, October 1992.

6. Loizou, P. C., *Speech Enhancement: Theory and Practice*, CRC Press Inc., Boca Raton, FL, 2007.

7. Arslan, L. M., "Modified Wiener Filtering", *Signal Processing*, Vol. 86, No. 2, pp. 267-272, 2006.

8. O'Shaughnessy, D., *Speech Communications: Human and Machine*, Wiley-IEEE Press, 2nd edition, New York, 1999.

9. Rabiner, L. and R. Schafer , *Digital Processing of Speech Signals*, Prentice-Hall Inc.,

New Jersey, 1978.

10. Than, K., *In Search of Music's Biological Roots*, http:// www.dukemagazine.duke.edu/issues/050608/music2.html, 2008.

11. Peterson, G. E. and H. L. Barney, "Control Methods Used in a Study of the Vowels", *The Journal of the Acoustical Society of America*, Vol. 24, No. 2, pp. 175-184, March 1952.

12. Türk, O., Ö. Şayli, A. S. Özsoy and L. M. Arslan, "Türkçede Ünlülerin Formant Frekans İncelemesi", *18. Ulusal Dilbilim Kurultayı, Ankara, Turkey*, 2004.

13. *Wavesurfer Speech Analysis tool*, http:// www.speech.kth.se/wavesurfer.

14. Schwartz, D. A., C. Q. Howe and D. Purves, "The Statistical Structure of Human Speech Sounds Predicts Musical Universals", *The Journal of Neuroscience*, Vol. 23, No. 18, pp. 7160-7168, August 2003.

15. Shanmugan, K. S. and A. M. Breipohl, *Random Signals: Detection Estimation and Data Analysis*, John Wiley & Sons, New York, 1988.

16. Hu, Y. and P. C. Loizou, "Subjective Comparison of Speech Enhancement Algorithms", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 153-156, May 2006.

17. Chang, J. H., N. S. Kim and S. K. Mitra, "Voice Activity Detection Based on Multiple Statistical Models", *IEEE Transactions on Signal Processing*, Vol. 54, No. 6, pp. 1965-1976, June 2006.

18. Rabiner, L. and M. Sambur, "Voiced-Unvoiced-Silence Detection Using the Itakura LPC Distance Measure", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 323-326, May 1977.

19. Rabiner, L. and M. Sambur, "Application of an LPC Distance Measure to the

Voiced-Unvoiced-Silence Detection Problem", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 25, No. 4, pp. 338-343, August 1977.

20. Hoyt, J.D. and H. Wechsler, "Detection of Human Speech in Structured Noise", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 237-240, April 1994.

21. Junqua, J. C., B. Reaves and B. Mark, "A Study of Endpoint Detection Algorithms in Adverse Conditions: Incidence on a DTW and HMM Recognizer", *Proceedings of the Second European Conference on Speech Communication and Technology*, pp. 1371-1374, September 1991.

22. Haigh, J. A. and J. S. Mason, "Robust Voice Activity Detection using Cepstral Features", *Proceedings of the IEEE Conference on Computer, Communication, Control and Power Engineering*, Vol. 3, pp. 321-324, October 1993.

23. Tucker, R., "Voice activity detection using a periodicity measure", *Proceedings of the IEE Conference on Communications, Speech and Vision*, Vol. 139, No. 4, pp. 377-380, August 1992.

24. Beritelli, F., S. Casale and A. Cavallaro, "A Robust Voice Activity Detector for Wireless Communications Using Soft Computing", *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 9, pp. 1818-1829, December 1998.

25. Nemer, E., R. Goubran and S. Mahmoud, "Robust Voice Activity Detection Using Higher-Order Statistics in the LPC Residual Domain", *IEEE Transactions on Speech and Audio Processing*, Vol. 9, No. 3, pp. 217231, March 2001.

26. Itakura, F., "Minimum Prediction Residual Principal Applied to Speech Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 23, No. 1, pp. 67-72, February 1975.

27. Ye, L., W. Tong, C. Huijuan and T. Kun, "Voice Activity Detection in Non-stationary Noise", *IMACS Multiconference on Computational Engineering in Systems Applications*, Vol. 2, pp. 1573-1575, October 2006.

28. Rabiner, L. R. and M. R. Sambur, "An Algorithm for Determining the Endpoints for Isolated Utterances", *The Bell System Technical Journal*, Vol. 54, No. 2, pp. 297-315, February 1975.

29. Gonzalez, R. and P. Wintz, *Digital Image Processing*, Addison-Wesley, 1987.

30. Arslan, L. M., *Foreign Accent Classification in American English*, Ph.D. Thesis, Duke University, 1996.

31. Friedman, D. H., "Pseudo-Maximum-Likelihood Speech Pitch Extraction", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 25, No. 3, pp. 213-221, June 1977.

32. *Wideband Audio and IP Telephony*, http://
www.cisco.com/en/US/prod/collateral/voicesw/ps6788/phones/ps379/
ps8537/prod_white_paper0900aecd806fa57a.html#wp9000001

33. Ephraim, Y., H. Lev-Ari and W. J. J. Roberts, "A Brief Survey of Speech Enhancement", *The Electronic Handbook, CRC Press*, April 2005.

34. Ephraim, Y. and H. L . V. Trees, "A signal subspace approach for speech enhancement", *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 4, pp. 251-266, July 1995.

35. Berouti, M., R. Schwartz and J. Makhoul, "Enhancement of Speech Corrupted by Acoustic Noise", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, pp. 208-211, April 1979.

36. Boll, S. F., "Suppression of Acoustic Noise in Speech Using Spectral Subtraction", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 27, No. 2, pp.

113-120, April 1979.

37. McAulay, R. and M. Malpass, "Speech Enhancement Using a Soft-Decision Noise Suppression Filter", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 28, No. 2, pp. 137-145, April 1980.

38. Arslan, L. M. and J. N. L. Hansen, "Minimum Cost Based Phoneme Class Detection for Improved Iterative Speech Enhancement", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 45-48, April 1994.

39. Hansen, J. H. L. and L. M. Arslan, "Markov Model Based Phoneme Class Partitioning for Improved Constrained Iterative Speech Enhancement", *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 1, pp. 98-104, January 1995.

40. Hansen, J. H. L. and M. A. Clements, "Constrained Iterative Speech Enhancement with Application to Speech Recognition", *IEEE Transactions on Signal Processing*, Vol. 39, No. 4, pp. 795-805, April 1991.

41. Lim, J. and A. V. Oppenheim, "Enhancement and Bandwidth Compression of Noisy Speech", *Proceedings of IEEE*, Vol. 67, No. 12, pp. 1586-1604, December 1979.

42. Scalart, P. and J. V. Filho, "Speech Enhancement Based on Apriori Signal to Noise Estimation", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, No. 2, pp. 629-632, May 1996.

43. Ephraim, Y. and D. Malah, "Speech Enhancement Using a Minimum-Mean Square Error Short-Time Spectral Amplitude Estimator", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 32, No. 6, pp. 1109-1121, December 1984.

44. Ephraim, Y. and D. Malah, "Speech Enhancement Using a Minimum-Mean Square Error Short-Time Log-Spectral Amplitude Estimator", *IEEE Transactions*

on Acoustics, Speech and Signal Processing, Vol. 33, No. 2, pp. 443-445, April 1985.

45. Martin, R., "Speech Enhancement Using MMSE Short Time Spectral Estimation with Gamma Distributed Speech Priors", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, No. 1, pp. 253-256, May 2002.

46. Lotter, T. and P. Vary, "Noise Reduction by Maximum A Posteriori Spectral Amplitude Estimation With Supergaussian Speech Modeling", *Proceedings of the International Workshop on Acoustic Echo and Noise Control*, pp. 8386, September 2003.

47. Dendrinos, M., S. Bakamidis and G. Carayannis, "Speech Enhancement from Noise: A Regenerative Approach", *Speech Communication*, Vol. 10, No. 1, pp. 45-57, February 1991.

48. Paliwal, K. K. and L. D. Alsteris, "On the usefulness of STFT phase spectrum in human listening tests", *Speech Communication*, Vol. 45, No. 2, pp. 153-170, February 2005.

49. Wiener, N., *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*, MIT Press, Cambridge, MA, 1949.

50. Moon, T. K. and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice-Hall, Upper Saddle River, NJ, 2000.

51. Makhoul, J., "Linear Prediction: A Tutorial Review", *Proceedings of the IEEE*, Vol. 63, No. 4, pp. 561-580, April 1975.

52. ITU, "A silence compression scheme for G.729 optimized for terminals conforming to ITU-T V.70", *ITU-T Rec. G. 729, Annex B*, 1996.

53. *Sphinx Speech Recognition Software*, http://cmusphinx.sourceforge.net /html/cmusphinx.php.

54. Lee, K. F., H. W. Hon and R. Reddy, "An overview of the SPHINX speech recognition system", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 38, No. 1, pp. 3545, January 1990.

55. Singh, R., *The Sphinx Speech Recognition Systems*, http://www.cs.cmu.edu/~rsingh/homepage/sphinx_history.html, 2003.