Auckland University of Technology

School of Engineering, Computer and Mathematical Sciences

Single-Channel Speech enhancement using statistical modelling

Sarang Chehrehsa

2016

A thesis submitted to Auckland University of Technology in fulfilment of the requirements for the degree of Doctor of Philosophy (PhD)

To my lovely wife, Nina

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Sarang Chehrehsa

Abstract

A new speech enhancement method based on Maximum A-Posteriori (MAP) estimation on Gaussian Mixture Models (GMMs) of speech and different noise types is introduced. The GMMs model the distribution of speech and noise periodograms in a high dimensional space and hence decrease the complexity of estimation procedure. Using the GMMs the Probability Density Functions (PDFs) of clean speech and noise can be calculated and by applying MAP on these PDFs, the estimates of speech and noise periodograms that form the noisy speech periodogram of the observed noisy speech frame can be estimated. These estimates are then used in a Wiener filter to enhance the noisy speech and recover the speech signal as close as possible to the original one. Since the PDFs are complicated and hence the realization of a MAP criterion can become even more complicated, some approximations are used to find the MAP criterion. Some improvements on this MAP estimation based on the characteristics of periodograms are also introduced in which the approximations are improved in a way which leads to more accurate estimates of speech and noise periodograms. Since the accuracy of the introduced MAP estimate is highly dependent on the accuracy of speech and noise power estimation in the noisy frame, a new power estimation method using Gamma modelling is introduced to replace the older methods like Minimum Statistics. The results of all the estimation methods are used in a classic Wiener filter to be applied on the noisy frame to enhance it. Since all the estimation algorithms can have some errors, we introduce an improvement of Wiener filter in which we can attenuate the effect of these errors on the enhanced speech signal. The performance of all the introduced methods are analyzed in terms of quality and intelligibility and reported thus.

Contribution to knowledge

- A new Gaussian Mixture Model is introduced to model the distribution of periodograms in vector space. To do so, the normalized periodograms are considered as vectors in a multi-dimensional space that can form colonies of periodograms with similar shapes. The Gaussian Mixture Model can model the number of periodograms in each colony based on the mean vector and the covariance matrix of that colony. Also a method introduced for finding a proper number of Gaussians when dealing with high dimensional spaces.
- Using these models, a Maximum A-Posteriori criterion is calculated on the Probability Density Functions of clean speech and noise periodograms to estimate the speech and noise periodograms from the observed noisy speech periodogram.
- A new Gamma model is introduced for modelling the distribution of speech and noise periodogram powers. Using this model, a Maximum A-Posteriori criterion is calculated on the Probability Density Functions of clean speech and noise powers to estimate the speech and noise periodogram powers from the observed noisy speech periodogram power.
- A parameter is added to the classic Wiener filter formula with which a higher level of noise reduction is attained. This parameter will help to decrease the effect of the resulted error from the periodogram estimation algorithms.

Acknowledgements

I would like to thank the school of engineering for awarding me the PhD stipend to help me financially during my study.

Big thanks to my supervisors and also my dear parents for their help and support through this achievement.

List of publications

Journal Papers

1. S. Chehrehsa, T. J. Moir, “Speech Enhancement Using Maximum A-Posteriori and Gaussian Mixture Models for Speech and Noise Periodogram Estimation”, *Computer Speech & Language*, Vol. 36, pp. 58-71, March 2016.

Conference paper

1. S. Chehrehsa, T. J. Moir, “Speech enhancement using improved MAP estimation and Wiener filter”, International Conference on Audio, Language and Image Processing (ICALIP), Shanghai, July 2016.

Table of Contents

List of figures

List of abbreviations

Abbreviation	Description
AR	Auto Regressive
BSS	Blind Source Separation
DFT	Discrete Fourier Transform
EM	Estimate Maximization
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Modelling
HMM	Hidden Markov Model
HNM	Harmonic Noise pulse Model
MAP	Maximum A Posteriori
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MS	Minimum Statistics
MSE	Mean Square Error
NMF	Nonnegative Matrix Factorization
PDF	Probability Density Function
PESQ	Perceptual Evaluation of Speech Quality
SAR	Source to Artefact Ratio
SDR	Source to Distortion Ratio
SIR	Source to Interference Ratio
SNR	Signal to Noise Ratio
STSA	Short Time Spectral Amplitude
VAD	Voice Activity Detector

Nomenclature

Symbol	Description
$x(m)$	Noisy speech signal in discrete time domain
$s(m)$	Clean speech signal in discrete time domain
$n(m)$	Noise in discrete time domain
$w(m)$	Window in discrete time domain
m	Discrete time index
ω	Discrete frequency index
Ω	Total number of frequency bins
$\boldsymbol{X}$ or $X(\omega)$	Noisy speech spectrum in discrete frequency domain
$\boldsymbol{S}$ or $S(\omega)$	Clean speech spectrum in discrete frequency domain
$\boldsymbol{N}$ or $N(\omega)$	Noise spectrum in discrete frequency domain
$\hat{S}(\omega)$	Estimate of clean speech spectrum in discrete frequency domain
$\widehat{N}(\omega)$	Estimate of noise spectrum in discrete frequency domain
$\hat{s}(t)$	Estimate of clean speech in time domain in discrete frequency domain
$H(\omega)$	Gain function in discrete frequency domain
ξ	*a-priori* SNR
f	PDF
R	Bayesian risk function
C	Cost-of-error function
γ	*a-posteriori* SNR
$\boldsymbol{\Psi}$	Rank deficient matrix
$\boldsymbol{\Sigma}_s$	Covariance matrix of clean speech
$\boldsymbol{\Sigma}_n$	Covariance matrix of noise
$\boldsymbol{V}$	Eigenvector
$\boldsymbol{\Lambda}$	Eigenvalue
$\boldsymbol{P}_x$ or $P_x(\omega)$	Noisy speech periodogram in discrete frequency domain
$\boldsymbol{P}_s$ or $P_s(\omega)$	Clean speech periodogram in discrete frequency domain

$\boldsymbol{P}_n$ or $P_n(\omega)$	Noise periodogram in discrete frequency domain
$\boldsymbol{W}$ or $W(\omega)$	Wiener filter in discrete frequency domain
$P_{SM}(\omega)$	Smoothed periodogram in discrete frequency domain
$P_{s,i}^{cb}(\omega)$	i-th centroid from speech periodogram codebook in discrete frequency domain
$P_{n,j}^{cb}(\omega)$	j-th centroid from noise periodogram codebook in discrete frequency domain
G	Gaussian function
$\boldsymbol{\mu}$ or $\mu(\omega)$	Gaussian mean vector in discrete frequency domain
Σ	Gaussian covariance matrix
π	Gaussian probability
L	Likelihood function
L_{log}	Log-likelihood function
$\boldsymbol{\sigma}$ or $\sigma(\omega)$	Gaussian variance vector in discrete frequency domain
$\bar{\bar{\boldsymbol{S}}}$	Space of normalized speech
$\bar{\bar{\boldsymbol{N}}}$	Space of normalized noise
$\bar{P}_s$	Power of speech periodogram
$\bar{P}_n$	Power of noise periodogram
$\bar{P}_x$	Power of noisy speech periodogram
$\boldsymbol{Q}_s$ or $Q_s(\omega)$	Normalized speech periodogram in discrete frequency domain
$\boldsymbol{Q}_n$ or $Q_n(\omega)$	Normalized noise periodogram in discrete frequency domain
$\boldsymbol{A}_s$ or $A_s(\omega)$	Speech spectral amplitude in discrete frequency domain
$\boldsymbol{A}_n$ or $A_n(\omega)$	Noise spectral amplitude in discrete frequency domain
Γ	Gamma function
λ	Skewness
a_s	Shape parameter of speech power Gamma model
b_s	rate parameter of speech power Gamma model
a_n	Shape parameter of noise power Gamma model
b_n	rate parameter of noise power Gamma model
a_s	Shape parameter of noisy speech power Gamma model
b_x	rate parameter of noisy speech power Gamma model
$u(\omega)$	Unity vector in discrete frequency domain

Notice on vector notation

All the vectors in discrete frequency domain are shown either in bold italic font, or normal italic font with (ω) next to it. For example, $\boldsymbol{P}_x$ or $P_x(\omega)$.

The multiplication and division of two vectors are element-wise. For example, for $W(\omega)$ and $X(\omega)$ defined as below:

$$\boldsymbol{X} = X(\omega) = \{X_0, X_1, \dots, X_\Omega\} \qquad 0 < \omega < \Omega$$

$$\boldsymbol{W} = W(\omega) = \{W_0, W_1, \dots, W_\Omega\} \qquad 0 < \omega < \Omega$$

Their element-wise multiplication is shown as

$$W(\omega)X(\omega) = [W_0X_0, W_1X_1, \dots, W_\Omega X_\Omega]$$

The element-wise division is shown as

$$\frac{W(\omega)}{X(\omega)} = \left[\frac{W_0}{X_0}, \frac{W_1}{X_1}, \dots, \frac{W_\Omega}{X_\Omega}\right]$$

More explanation is give in Appendix A.

1 Introduction

Speech is the simplest communication method among human beings. Starting with simple fixed-line telephone networks for limited distances to the use of improved mobile communication networks, there have been numerous improvements to improve communication among people. Using mobile communications, it has become possible to communicate everywhere and in different environments. Such flexibilities in mobile communication have some challenges. The subscribers of mobile communication communicate in different environments with different background noises and different levels such as traffic noise, car engine noise, interference of multiple speakers in a café and etc. Another application of the speech signal is in hearing aids which can be carried with the user in different environments of different noises and levels and can lead to the same challenges of mobile communication. In other applications there might be a recorded version of a speech signal which is corrupted by some noise which might not even be known to the listener. The reduction of all these mentioned noise is always of great importance and sometimes complicated. Noise reduction will reduce hearing difficulties and will increase the quality and intelligibility of the enhanced signals for the listener. The enhancement process will improve the performance of coding algorithms which are the basis of mobile communication. The resistance to environmental noise is a limiting factor for the comprehensive use of communication systems. Although the mentioned algorithms have a good performance in the noiseless and controlled environments, their performance will highly degrade in noisy conditions.

Most of speech enhancement algorithms are classified into two main groups, single-channel and multi-channel. The single-channel algorithms are applied on the resulting

input signal of one microphone and since they are less complicated, are of more interest in mobile communications. In return, the multi-channel algorithms use an array of two or more microphones to input the noisy signal to capture the spatial variety of the signal. These two methods are not necessarily separated and to have better performance they might be combined [1, 2].

In speech enhancement it is considered that the noisy speech is the summation of the clean or pure speech and the pure noise and hence the noise is additive. In speech enhancement algorithms, either single-channel or multi-channel, only the noisy signal is available as an observations and the aim is to extract the forming clean speech signal as close as possible to the original one that was combined with the noise.

In multi-channel methods, there are multiple microphones in different locations with respect to the speech source and hence they all receive almost the same clean speech signal but with different noises or different levels of the same noise based on their location. In this way, always multiple versions of noisy speech with different noises and/or different levels of Signal to Noise Ratio (SNR) are available and hence the clean speech can be considered as the common signal between these noisy observations. In single-channel methods, only one noisy observation is available and there is no other separate information source about the interfering noise or noises and their levels and hence all the information about the clean speech and noise should be extracted from that single noisy speech observation. In this way the single-channel algorithms are more challenging and can become mathematically more complicated. Moreover the single-channel methods can be expanded to be used in multi-channel applications and hence in this research we are going to concentrate more on single-channel methods.

As discussed in [3], speech enhancement algorithms can be classified into two main categories: unsupervised and supervised algorithms. The simplest and most famous speech enhancement method is spectral subtraction [4] in which the spectral amplitude

of noise is estimated from the spectral amplitude of noisy speech and subtracted from it to get to the spectral amplitude of clean speech. Since there are no considerations of the speech spectrum in spectral subtraction, it results in an artificial noise called musical noise. To reduce this musical noise some methods discussed in [5] can be used which are simple but require an efficient Voice Activity Detector (VAD) to estimate the noise spectrum. In these methods the amplitude of the spectrum is used, but in [6] the complex spectrum is considered. In Wiener filtering [7], speech and noise probabilistic properties are used and hence the enhanced speech suffers from less musical noise with respect to spectral subtraction methods. A method called Short Time Spectral Amplitude (STSA) is discussed in [8], which is based on Minimum Mean Square Error (MMSE) estimation with the assumption that the speech and noise spectral components are statistically independent and Gaussian random variables. The MMSE-STSA method is derived by minimizing a conditional mean square value of the short time spectral amplitude. As discussed in [9], when Probability Density Functions (PDF) of speech and noise spectrums are assumed to be Gaussian, the spectral gain will become the Wiener filter gain. This method is based on the *a priori* SNR estimation on a frame-by-frame basis by a decision directed approach and Maximum Likelihood (ML) estimator with the assumption that the noise variance is known or can be estimated during the silence intervals. There are different methods to calculate the *a priori* SNR. A decision directed method is discussed in [10]. A data driven approach to calculating *a priori* SNR is discussed in [11] in which two trained artificial neural networks, one for speech and one for noise, is used. As confirmed in the literature, the MMSE spectral gain is superior to the spectral subtraction method but is computationally complicated to implement. To overcome this issue, a method discussed in [9] called the Maximum a posteriori (MAP) method, which can result in relatively good enhancement results if used.

In supervised speech enhancement algorithms we use some additional information about noise and speech such as noise type, speaker identity etc. to improve the enhancement. In supervised methods we create some offline models for speech and noise which are trained using large observed samples of each signal. Some examples of this class of algorithms include the codebook based approaches as discussed in [12], where LPC codebooks of speech are used and in [13], AR coefficient codebooks of speech and noise are used which leads to ML estimates of clean speech. Another well-known and high performance supervised speech enhancement methods are Hidden Markov Model (HMM) based systems and the state-of-the-art approaches are discussed in references [14-16]. In these methods, the waveform signal is modeled as an autoregressive (AR) process, and hence the waveforms of speech and noise signals are modeled by HMMs. In recent HMM based methods as discussed in [17-19], distribution of the power spectral coefficients of speech and noise are modelled using HMMs using Gamma distribution. There are also other methods that are based on modelling the spectral amplitude of speech using Gaussian Mixture Modelling (GMM) in which estimates of speech are attained using a MAP criterion as discussed in [20-22] and a minimum mean-square error (MMSE) criterion as discussed in [23]. The advantage of the supervised approaches such as the HMM based denoising algorithm is that it produces high quality enhanced speech signals. The reason for this is that for each noise type, a system is trained *a priori*. This is a tedious task in practice and is addressed in [3].

In most supervised model based algorithms, GMM is used for the modelling of spectral amplitudes, log spectral amplitudes or periodograms with their true power and some of these methods give excellent enhancement results. In this research we are going to use normalized periodograms (with power equal to one) as the vectors to be modelled. In most Bayesian estimation criterions especially MAP estimation, due to the complexity of the estimation criterion formula, some mathematical distributions like Laplacian or

Gamma are used to simplify the formula. Here we aim to use GMM to generate a model for the distribution of speech and noise periodograms in a multi-dimensional space and then using MAP on these GMMs to estimate the speech and noise periodograms that form the noisy speech periodograms [9, 24]. Using these estimated periodograms we will construct Wiener filters to enhance the noisy speech and recover the clean speech. In the next chapter we are going to discuss different speech enhancement methods and their pros and cons.

2 Speech enhancement methods review

We are going to introduce some well-known speech enhancement methods that have been introduced in the past decade. There are different methods to remove or suppress the additive noise for the observations. These methods are called speech enhancement techniques. From the additive assumption for the noise in the time domain we have

$$x(m) = s(m) + n(m) \tag{2.1}$$

where x, s and n are the noisy speech, clean speech and noise respectively and the index m represents the discrete time index. In single channel speech enhancement methods only the noisy speech observation $x(m)$ are available and through this observation an estimate of $s(m)$ should be extracted. This procedure is called speech enhancement and is classified into some main groups. These main groups are conventional methods, binary time-frequency masks and subspace methods.

2.1 Conventional speech enhancement methods

A large group of speech enhancement methods is called conventional methods in which a filter is used to remove an estimate of noise from the noisy speech to generate an estimate of the clean speech. These methods focus on the enhancement of the spectral amplitude of the speech signal and are also known as Short-Time-Spectral-Amplitude (STSA) methods.

2.1.1 Basics of conventional speech enhancement methods

These methods contain three steps which are analysis, enhancement and synthesis.

Analysis

The noisy speech signal is processed on a frame-by-frame basis. To preserve the stationarity assumption for the analyzed signal, it is divided into equal frames of length 10-30ms [25]. Applying a square window that sharply extracts the frame can cause frequency overshoots after the Fourier transform of the frame. Since the analysis is performed in the frequency domain and there are limitations in the use of Discrete Fourier Transform (DFT), the frames are windowed using Hanning or Hamming windows. To avoid data loss and aliasing in the modulation domain and to compensate for the window effect which is like weighting in the time domain, these frames are overlapped. An overlap of 75% can completely compensate the time domain weighting of the signal components.

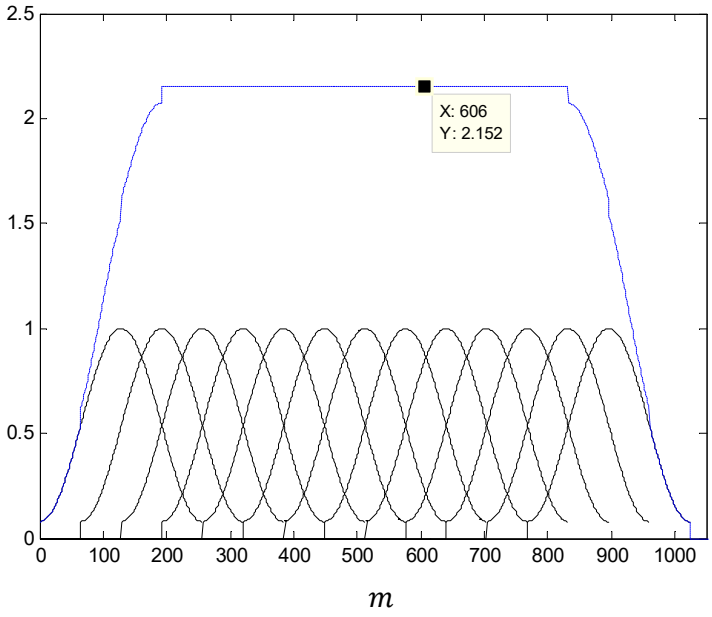

Figure 2.1: Hamming windows with 75% overlap (solid line) and their sum (dashed line)

Applying the Hamming window on the signals denoted in (2.1) will lead to

$$x_k(m) = s_k(m) + n_k(m) \tag{2.2}$$

where the k index shows the k-th frame of each signal. For simplicity and to avoid the k index in the equations, in this document (2.2) will be replaced by (2.1) but we know that it is actually written on the frames of these signals and not the whole signal. To find the spectrum of one frame of a windowed noisy speech, the DFT will be applied accordingly

$$X(\omega) = \sum_{m=0}^{\Omega-1} w(m)x(m)e^{-j\frac{2\pi\omega m}{\Omega}} \quad for \quad 0 \leq \omega \leq \Omega - 1 \tag{2.3}$$

where $x(m)$ and $w(m)$ are the m-th samples of the noisy speech and window (for example Hamming) respectively with total Ω time samples in the frame and $X(\omega)$ is the ω-th frequency bin of the complex spectrum on Ω bins. In the same way discussed in (2.3) the (2.1) can be rewritten in frequency domain as:

$$X(\omega) = S(\omega) + N(\omega) \tag{2.4}$$

where S and N are the spectrums of clean speech and noise respectively. The spectrum of noisy speech $X(\omega)$ is a vector of complex values and hence the spectral amplitude of noisy speech is defined as $|X(\omega)|$ which represents the amplitude of each frequency component. In the same way we can define $|S(\omega)|$ and $|N(\omega)|$ as the spectral amplitude of clean speech and noise, respectively.

Enhancement

For the enhancement methods which are based on STSA, the focus is on removing the noise frequency components from the noisy speech spectral amplitude. The effect of noise on phase is assumed to be inaudible [26] and in this case the enhancement is only applied on the spectral amplitude and the phase of clean speech spectrum is taken the same as noisy speech phase [27]. The noisy speech spectral amplitude is considered to be a function of the clean speech and noise spectral amplitudes as in $|X(\omega)| =$

$f(|S(\omega)|, |N(\omega)|)$ and hence the enhancement step is to find a function that by applying it to the noisy speech spectral amplitude, an estimate of clean speech spectral amplitude will be attained as $\left|\hat{S}(\omega)\right| = f^{-1}\left(|X(\omega)|, \left|\widehat{N}(\omega)\right|\right)$ in which $\left|\hat{S}(\omega)\right|$ and $\left|\widehat{N}(\omega)\right|$ represent an estimate of clean speech and noise spectral amplitude. The implementation of this step is highly dependent to these approaches:

1) An accurate estimate of noise spectral amplitude $\left|\widehat{N}(\omega)\right|$
2) An appropriate and accurate noise removal function f^{-1}

The noise removal function $f^{-1}\left(|X(\omega)|, \left|\widehat{N}(\omega)\right|\right)$ is mostly considered as a gain function as $H(\omega)$ in which

$$\left|\hat{S}(\omega)\right| = H(\omega)|X(\omega)| \tag{2.5}$$

The gain function or the filter is computed using *a-priori* and *a-posteriori* SNRs.

Synthesis

After the noisy speech frames were divided into some overlapping frames and transferred to frequency domain using DFT as discussed in "Analysis" section and went through the enhancement and filtering process as discussed in "Enhancement" section, now it is the time that all the enhanced frames being transferred back to time domain and being re-arranged as the whole signal and this procedure is called synthesis. In this way, the spectrum of clean speech could be formed using the estimated spectral amplitude of clean speech $\left|\hat{S}(\omega)\right|$ and the phase of noisy speech as

$$\hat{S}(\omega) = \left|\hat{S}(\omega)\right| e^{j\angle X(\omega)} \tag{2.6}$$

where $\angle X(\omega)$ represents the phase of noisy speech. By applying inverse DFT on the estimated clean speech frame in time domain is calculated as:

$$\hat{s}(m) = \frac{1}{\Omega}\sum_{\omega=0}^{\Omega-1} \hat{S}(\omega) e^{j\frac{2\pi\omega m}{\Omega}} \quad for \quad 0 \le m \le M-1 \tag{2.7}$$

Since the resulted $\hat{s}(t)$ is just one frame of the enhanced speech, using Overlap-Add method, all the enhanced frames could be put together to form the whole enhanced speech signal. In this way if the noisy speech was divided to frames with 75% overlap then the recovered $\hat{s}(m)$ frames should be added with their neighboring frames with 75% overlap.

2.1.2 **Spectral subtraction**

This method is the most basic and the simplest speech enhancement method. Since we assumed that the noise is additive, by subtracting the noise estimate from the noisy speech, the clean speech estimate can be calculated. As discussed in section 2.1.1 since most of the conventional methods are dealing with the spectral amplitude and not caring about the spectral phase, the spectral subtraction method could be implemented through applying a gain factor to the spectral amplitude of noisy speech as in (2.5) in which the gain function $H(\omega)$ could be calculated as:

$$H(\omega) = \frac{\left|\hat{S}(\omega)\right|}{\left|\hat{S}(\omega)\right| + \left|\hat{N}(\omega)\right|} = \frac{\left|\hat{S}(\omega)\right|}{|X(\omega)|} = 1 - \frac{\left|\hat{N}(\omega)\right|}{|X(\omega)|} \tag{2.8}$$

Applying the resulting $H(\omega)$ in (2.5) will give an estimate of the clean speech spectral amplitude as:

$$\left|\hat{S}(\omega)\right| = f^{-1}\left(|X(\omega)|, \left|\hat{N}(\omega)\right|\right) = |X(\omega)| - \left|\hat{N}(\omega)\right| \tag{2.9}$$

The resulting subtraction can be implemented in different domains with the index p as:

$$\left|\hat{S}(\omega)\right| = \sqrt[p]{f^{-1}\left(|X(\omega)|^p, \left|\hat{N}(\omega)\right|^p\right)} \tag{2.10}$$

where $p = 1$ represents the magnitude spectrum (spectral amplitude) and $p = 2$ represents power spectrum (periodogram). Since the subtraction of noise spectral amplitude from noisy speech spectral amplitude is done individually for each frequency bin, by having larger noise spectral amplitude values, it can result in negative values for speech spectral amplitude which is not valid (since we cannot have negative values for spectral amplitude). In this way (2.9) and (2.10) for $p = 2$ can be rewritten as:

$$\left|\hat{S}(\omega)\right|^2 = \begin{cases} |X(\omega)|^2 - \left|\widehat{N}(\omega)\right|^2 & \text{if } |X(\omega)|^2 > \left|\widehat{N}(\omega)\right|^2 \\ 0 & \text{else} \end{cases} \tag{2.11}$$

Using this method we can find valid clean speech spectral amplitude for half-wave while the zeroing of resulted negative values will expose some random peaks to the spectrum which can cause artifacts in the reconstructed speech. Since the locations of these peaks are random among the frames, a random structured tone will be added to the enhanced speech which is referred to as musical noise. There are some improvements on this zeroing process in which any negative spectral bin will be floored spectrally to a proportion of noise power spectrum estimate [28]. The noise power spectrum estimate is multiplied by an over-subtraction factor α and then subtracted from the noisy speech power spectrum. All the negative bins are then replaced by noise power spectrum estimate scaled by spectral floor parameter β as:

$$\left|\hat{S}(\omega)\right|^2 = \begin{cases} |X(\omega)|^2 - \alpha\left|\widehat{N}(\omega)\right|^2 & \text{if } |X(\omega)|^2 > (\alpha + \beta)\left|\widehat{N}(\omega)\right|^2 \\ \beta\left|\widehat{N}(\omega)\right|^2 & \text{else} \end{cases} \tag{2.12}$$

In this way some high amplitude peaks associated with the enhanced speech will be terminated. The amplitude of the broadband peaks will be reduced by the use of the over-subtraction of the noise and in this way only some low amplitude narrowband peaks will be left. These narrowband peaks are then masked through the use of a fraction of the noise back on to the spectrum. The parameter β controls the amount of residual noise and hence the level of musical noise and the parameter α controls the

level of speech distortion. These two parameters can be determined by experiment or by a Minimum Mean Square Error (MMSE) estimate of the optimal parameters [29]. These parameters can also be calculated in spectral band or spectral bin level as $\alpha(\omega)$ and $\beta(\omega)$. The various configurations of spectral subtraction based methods are being tested in terms of intelligibility and quality of the enhanced speech [30, 31]. This method exhibits relatively good quality and intelligibility for the enhanced speech and in terms of high suppression of the background noise the intelligibility of the speech might be slightly reduced.

2.1.3 **The Wiener filter**

From (2.5) we know that an estimate of clean speech spectral amplitude can be calculated as $|\hat{S}(\omega)| = H(\omega)|X(\omega)|$ and since we can use the noisy speech phase for the enhanced speech hence for the enhanced speech spectrum we can write:

$$\hat{S}(\omega) = H(\omega)X(\omega) \tag{2.13}$$

The Wiener filter is based on minimization of the Mean Square Error (MSE) between the estimate of the clean speech spectrum $\hat{S}(\omega)$ and the real clean speech spectrum $S(\omega)$. To find the MSE between $\hat{S}(\omega)$ and $S(\omega)$ we can write:

$$MSE = E\left[\left(S(\omega) - \hat{S}(\omega)\right)\left(S(\omega) - \hat{S}(\omega)\right)^*\right] \tag{2.14}$$

where the superscript $*$ represents the complex conjugate and $E[.]$ represents the statistical expectation value which mathematically is the mean of the corresponding function. By replacing $\hat{S}(\omega)$ in (2.14) from (2.13), the MSE can be rewritten as:

$$MSE = E\left[\left(S(\omega) - H(\omega)X(\omega)\right)\left(S(\omega) - H(\omega)X(\omega)\right)^*\right] \tag{2.15}$$

We can also replace $X(\omega)$ with $S(\omega) + N(\omega)$ using (2.4) and rewrite (2.15) as:

$$MSE = E\left[\Big(S(\omega) - H(\omega)\big(S(\omega) + N(\omega)\big)\Big)\Big(S(\omega) - H(\omega)\big(S(\omega) + N(\omega)\big)\Big)^*\right] \quad (2.16)$$

For simplicity we ignore the index ω and show the spectrums just by their names in bold representing their vectors as $\boldsymbol{S} = S(\omega)$, $\boldsymbol{N} = N(\omega)$ and $\boldsymbol{H} = H(\omega)$ and expand (2.16) as:

$$\begin{aligned} MSE = E[&\boldsymbol{S}\boldsymbol{S}^* - \boldsymbol{H}^*(\boldsymbol{S}\boldsymbol{S}^* + \boldsymbol{S}\boldsymbol{N}^*) - \boldsymbol{H}(\boldsymbol{S}\boldsymbol{S}^* + \boldsymbol{S}^*N) \\ &+ \boldsymbol{H}\boldsymbol{H}^*(\boldsymbol{S}\boldsymbol{S}^* + \boldsymbol{S}\boldsymbol{N}^* + \boldsymbol{S}^*N + \boldsymbol{N}\boldsymbol{N}^*)] \end{aligned} \quad (2.17)$$

In the gain function $\boldsymbol{H} = H(\omega)$ in terms of Wiener filter the frequency components are all real values since they are supposed to either amplify or suppress the noisy speech spectral amplitude and hence $\boldsymbol{H} = \boldsymbol{H}^*$. Multiplying a spectrum to its complex conjugate is the power spectrum of that signal as $\boldsymbol{S}\boldsymbol{S}^* = |\boldsymbol{S}|^2 = |S(\omega)|^2$ and $\boldsymbol{N}\boldsymbol{N}^* = |\boldsymbol{N}|^2 = |N(\omega)|^2$. Using these definitions we can rewrite (2.17) as:

$$\begin{aligned} MSE = E[&|\boldsymbol{S}|^2 - \boldsymbol{H}(|\boldsymbol{S}|^2 + \boldsymbol{S}\boldsymbol{N}^*) - \boldsymbol{H}(|\boldsymbol{S}|^2 + \boldsymbol{S}^*N) \\ &+ \boldsymbol{H}^2(|\boldsymbol{S}|^2 + \boldsymbol{S}\boldsymbol{N}^* + \boldsymbol{S}^*N + |\boldsymbol{N}|^2)] \end{aligned} \quad (2.18)$$

Now to find the minimum of the MSE which is called Minimum Mean Square Error (MMSE), we need to find the first derivative of MSE with respect to $\boldsymbol{H}$ and take it equal to zero.

$$\begin{aligned} &\frac{\partial}{\partial \boldsymbol{H}} MSE = 0 \rightarrow \\ &E[-2|\boldsymbol{S}|^2 - \boldsymbol{S}\boldsymbol{N}^* - \boldsymbol{S}^*N + 2\boldsymbol{H}(|\boldsymbol{S}|^2 + \boldsymbol{S}\boldsymbol{N}^* + \boldsymbol{S}^*N + |\boldsymbol{N}|^2)] = 0 \end{aligned} \quad (2.19)$$

By solving (2.19) for the resulting two-sided Wiener filter, we have:

$$H(\omega) = \frac{2E[|S(\omega)|^2] + E[S(\omega)N^*(\omega) + S^*(\omega)N(\omega)]}{2E[|S(\omega)|^2 + |N(\omega)|^2] + 2E[S(\omega)N^*(\omega) + S^*(\omega)N(\omega)]} \quad (2.20)$$

The $|S(\omega)|^2$ and $|N(\omega)|^2$ values are all real and positive and hence their expectation value is positive. Since $\boldsymbol{S}\boldsymbol{N}^* = (\boldsymbol{S}^*\boldsymbol{N})^*$, hence $\boldsymbol{S}\boldsymbol{N}^* + \boldsymbol{S}^*\boldsymbol{N}$ will just contain real values

which could be either positive and negative. The expectation value of all these positive and negative values could be very small with respect to the expectation value of $|\boldsymbol{S}|^2$ and $|\boldsymbol{N}|^2$ and hence we can neglect the terms $E[\boldsymbol{SN}^* + \boldsymbol{S}^*\boldsymbol{N}]$ in (2.20). In this way, we can rewrite (2.20) and by neglecting expectation value function to have separate frequency components, and define Wiener filter as:

$$W(\omega) = H(\omega) = \frac{|S(\omega)|^2}{|S(\omega)|^2 + |N(\omega)|^2} \tag{2.21}$$

where $W(\omega)$ is used everywhere in this thesis as the Wiener filter to not to be confused with other filters. We can calculate the power spectrum of noisy speech through multiplying the two sides of (2.4) to their complex conjugate as:

$$|\boldsymbol{X}|^2 = \boldsymbol{X}\boldsymbol{X}^* = (\boldsymbol{S} + \boldsymbol{N})(\boldsymbol{S} + \boldsymbol{N})^* = |\boldsymbol{S}|^2 + |\boldsymbol{N}|^2 + \boldsymbol{S}\boldsymbol{N}^* + \boldsymbol{S}^*\boldsymbol{N} \tag{2.22}$$

Based on the discussion that we had before about neglecting the effect of $\boldsymbol{SN}^* + \boldsymbol{S}^*\boldsymbol{N}$ term, we can say that the noisy speech spectrum is as:

$$|X(\omega)|^2 = |S(\omega)|^2 + |N(\omega)|^2 \tag{2.23}$$

Using (2.23) and (2.21) we can show Wiener filter as:

$$W(\omega) = \frac{|S(\omega)|^2}{|S(\omega)|^2 + |N(\omega)|^2} = \frac{|S(\omega)|^2}{|X(\omega)|^2} = 1 - \frac{|N(\omega)|^2}{|X(\omega)|^2} \tag{2.24}$$

An alternative way of constructing Wiener filter is to use an *a-priori* SNR as:

$$W(\omega) = \frac{\xi(\omega)}{1 + \xi(\omega)} \quad , \quad \xi(\omega) = \frac{|S(\omega)|^2}{|N(\omega)|^2} \tag{2.25}$$

where $\xi(\omega)$ is the *a-priori* SNR. We can see that for the frequency bins with high SNRs, the corresponding Wiener component will be close to one and for small SNRs, the Wiener component will be close to zero. In [32] a method for finding this *a-priori* SNR by tracking the noise is introduced. Prior to this, there were an iterative method for

finding such an *a-priori* SNR as in [33] and a method for tracking noise using HMMs as in [34]. A more recent method is introduced in [35] where a model-based Wiener filter is derived from log-Mel feature vectors. By using MMSE estimation, the feature vectors were enhanced and inverted to compute the filter response. Using Mel filter bank during feature extraction process resulted in smooth Wiener filter over frequency and hence resulted in better spectral details in the noise cancellation process.

2.1.4 **Statistical model-based enhancement methods**

These methods use the statistical methods of estimation to extract the proper $H(\omega)$ filter. The main methods in this category which are Maximum Likelihood (ML), Minimum Mean Square Error (MMSE) and Maximum A-Posteriori (MAP) estimation which are the Bayesian estimation criteria. From (2.4) we want to recover the speech spectrum $S(\omega)$ form the observed noisy speech spectrum $X(\omega)$. Since the Bayesian estimation is based on statistics of the signal its focus is on the Probability Density Functions (PDF) of the mentioned signals. By applying Bayes rule we have:

$$f_{\boldsymbol{S}|\boldsymbol{X}}(\boldsymbol{S}|\boldsymbol{X}) = \frac{1}{f_{\boldsymbol{X}}(\boldsymbol{X})} f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{X}|\boldsymbol{S}) f_{\boldsymbol{S}}(\boldsymbol{S}) \tag{2.26}$$

where $f_{\boldsymbol{S}|\boldsymbol{X}}(\boldsymbol{S}|\boldsymbol{X})$ is the posterior PDF, $f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{X}|\boldsymbol{S})$ is the likelihood, $f_{\boldsymbol{S}}(\boldsymbol{S})$ is the prior PDF and $f_{\boldsymbol{X}}(\boldsymbol{X})$ is a constant which has only a normalizing effect. The Bayesian estimation of a speech spectrum $S(\omega)$ from the observed noisy speech spectrum $X(\omega)$ is based on the minimization of a Bayesian risk function which is defined as an average cost-of-error function as:

$$R(\widehat{\boldsymbol{S}}) = E\left[C(\widehat{\boldsymbol{S}}, \boldsymbol{S})\right] = \int_{\boldsymbol{S}} \int_{\boldsymbol{X}} C(\widehat{\boldsymbol{S}}, \boldsymbol{S}) f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{X}|\boldsymbol{S}) f_{\boldsymbol{S}}(\boldsymbol{S}) d\boldsymbol{X} d\boldsymbol{S} \tag{2.27}$$

where the cost-of-error function $C(\widehat{\boldsymbol{S}}, \boldsymbol{S})$ generates the proper weighting for the desired outcomes of the estimator and $E[\,]$ is the expectation value. For a given $\boldsymbol{X}$, $f_{\boldsymbol{S}|\boldsymbol{X}}(\boldsymbol{S}|\boldsymbol{X})$ is

constant and has no effect on the minimization process. Knowing that $f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{X}|\boldsymbol{S})f_{\boldsymbol{S}}(\boldsymbol{S}) = f_{\boldsymbol{S}|\boldsymbol{X}}(\boldsymbol{S}|\boldsymbol{X})f_{\boldsymbol{X}}(\boldsymbol{X})$ and $f_{\boldsymbol{X}}(\boldsymbol{X})$ being a constant, we can show the resulted Bayesian risk function from (2.27) as:

$$R(\widehat{\boldsymbol{S}}|\boldsymbol{X}) = \int_{\boldsymbol{S}} C(\widehat{\boldsymbol{S}}, \boldsymbol{S}) f_{\boldsymbol{S}|\boldsymbol{X}}(\boldsymbol{S}|\boldsymbol{X}) d\boldsymbol{S} \tag{2.28}$$

The Bayesian estimate $\widehat{\boldsymbol{S}}$ is obtained as the minimum-risk parameter vector as:

$$\begin{aligned} \widehat{\boldsymbol{S}}_{\text{Bayesian}} &= \arg\min_{\widehat{\boldsymbol{S}}} R(\widehat{\boldsymbol{S}}|\boldsymbol{X}) = \arg\min_{\widehat{\boldsymbol{S}}} \left[\int_{\boldsymbol{S}} C(\widehat{\boldsymbol{S}}, \boldsymbol{S}) f_{\boldsymbol{S}|\boldsymbol{X}}(\boldsymbol{S}|\boldsymbol{X}) d\boldsymbol{S} \right] \\ &= \arg\min_{\widehat{\boldsymbol{S}}} \left[\int_{\boldsymbol{S}} C(\widehat{\boldsymbol{S}}, \boldsymbol{S}) f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{X}|\boldsymbol{S}) f_{\boldsymbol{S}}(\boldsymbol{S}) d\boldsymbol{S} \right] \end{aligned} \tag{2.29}$$

In this way by calculating the first derivative of (2.29) with respect to $\widehat{\boldsymbol{S}}$ and equating it to zero the Bayesian estimate of the clean speech spectrum will be attained. Here we are going to describe each of these methods.

2.1.4.1 Maximum Likelihood estimation

It is widely used for parameter estimation and first used in speech enhancement in [36]. In this method we try to find the most likely clean speech spectral amplitude $|S(\omega)|$ that builds up the noisy speech spectral amplitude $|X(\omega)|$. It is assumed that the relationship between $|S(\omega)|$ and $|X(\omega)|$ is deterministic and not random. In this way the estimation of $|S(\omega)|$ could be done through maximization of likelihood function f as:

$$\left|\hat{S}(\omega)\right| = \arg\max_{|\hat{S}(\omega)|} f_{\boldsymbol{X}|\boldsymbol{S}}(|X(\omega)| \mid |S(\omega)|) \tag{2.30}$$

To solve this equation, we assume that the likelihood function has Gaussian distribution and then we should find the first derivative of the likelihood function f with respect to $\left|\hat{S}(\omega)\right|$ and by taking it equal to zero we will have [27]:

$$\left|\hat{S}(\omega)\right| = \frac{1}{2}\left[|X(\omega)| + \sqrt{|X(\omega)|^2 - \left|\widehat{N}(\omega)\right|^2}\right] \tag{2.31}$$

where $\left|\widehat{N}(\omega)\right|^2$ is the estimate noise power spectral amplitude. Such an estimator can be shown as a filter in terms of the *a-posteriori* SNR as:

$$H_{ML}(\omega) = \frac{1}{2} + \frac{1}{2}\sqrt{\frac{\gamma(\omega) - 1}{\gamma(\omega)}} \tag{2.32}$$

where $\gamma(\omega)$ is the *a-posteriori* SNR and can be calculated as:

$$\gamma(\omega) = \frac{|X(\omega)|^2}{\left|\widehat{N}(\omega)\right|^2} \tag{2.33}$$

By applying the resulted $H_{ML}(\omega)$ from (2.32) in (2.5) to find a ML estimate of the clean speech spectral amplitude.

2.1.4.2 Minimum Mean Square Error estimation

In this method a MSE criterion is minimized between $\left|\hat{S}(\omega)\right|$ and $|S(\omega)|$. This is a statistical estimation method in which the proper gain function $H(\omega)$ through Bayesian estimation. In this method we should have some prior knowledge of the Probability Density Function (PDF) of the speech and noise and hence this method is more accurate than the ML method. The cost-of-error function for MSE criterion will be as $E\left[\left(\left|\widehat{\boldsymbol{S}}\right| - |\boldsymbol{S}|\right)^2\right]$ and hence the Bayesian risk function based on (2.28) using MSE criterion will be:

$$R_{MSE}\left(\left|\widehat{\boldsymbol{S}}\right||\boldsymbol{X}\right) = \int_{|\boldsymbol{S}|}\left(\left|\widehat{\boldsymbol{S}}\right| - |\boldsymbol{S}|\right)^2 f_{\boldsymbol{S}|\boldsymbol{X}}(|\boldsymbol{S}||\boldsymbol{X})d|\boldsymbol{S}| \tag{2.34}$$

To minimize R_{MSE}, we find its first derivative with respect to $\left|\widehat{\boldsymbol{S}}\right|$ and equate it to zero. In this way the MMSE estimate of the clean speech spectral amplitude will be as:

$$\left|\hat{S}(\omega)\right| = \int_{|S(\omega)|} |S(\omega)| f_{\boldsymbol{S}|\boldsymbol{X}}\left(|S(\omega)||X(\omega)\right) d|S(\omega)| \tag{2.35}$$

It could be seen that $\left|\hat{S}(\omega)\right|$ is dependent on $X(\omega)$ and the posterior PDF which was mentioned in (2.26). By assuming the statistical independence between the coefficients and from the definition in (2.7) and by taking s_t as the t-th component of $s(t)$, the Bayesian MSE estimator can be simplified to:

$$\left|\hat{S}(\omega)\right| = \int_{s_t} s_t f_{\boldsymbol{S}|\boldsymbol{X}}\left(s_t|X(\omega)\right) ds_t = \frac{\int_0^\infty s_t f_{\boldsymbol{X}|\boldsymbol{S}}(X(\omega)|s_t) f_{\boldsymbol{S}}(s_t) ds_t}{\int_0^\infty f_{\boldsymbol{X}|\boldsymbol{S}}(X(\omega)|s_t) f_{\boldsymbol{S}}(s_t) ds_t} \tag{2.36}$$

In this way the spectral amplitude of clean speech can be calculated by MMSE estimator. Since human ears are sensitive to the logarithmic levels of sound intensity, the MMSE method for the calculation of log-magnitude spectrum was introduced in [37]. In this method the cost-of-error function for MSE criterion will be as $E\left[\left(\log\left(\left|\hat{S}(\omega)\right|\right) - \log(|S(\omega)|)\right)^2\right]$ and hence the MMSE estimator becomes like:

$$\log\left(\left|\hat{S}(\omega)\right|\right) = E[\log(|S(\omega)|)\,|X(\omega)] \tag{2.37}$$

In this way the estimate of clean speech spectral amplitude becomes:

$$\left|\hat{S}(\omega)\right| = \exp(E[\log(|S(\omega)|)\,|X(\omega)]) \tag{2.38}$$

Using (2.38), the corresponding gain function can be calculated as:

$$H(\omega) = \frac{\xi(\omega)}{1+\xi(\omega)} \exp\left(\frac{1}{2}\int_{v(\omega)}^{\infty} \frac{e^{-t}}{t} dt\right) \tag{2.39}$$

where $\xi(\omega)$ is the *a-priori* SNR as $\frac{|S(\omega)|^2}{|\bar{N}(\omega)|^2}$ from (2.25) and $v(\omega)$ is defined as:

$$v(\omega) = \frac{\xi(\omega)}{1+\xi(\omega)} \gamma(\omega) \tag{2.40}$$

where $\gamma(\omega)$ is the *a-posteriori* SNR as discussed in (2.33). In this way the gain function or filter can be shown as a function of *a-priori* and *a-posteriori* SNRs applied to the noisy speech spectral amplitude as:

$$\left|\hat{S}(\omega)\right| = f^{-1}\left(\left|\hat{S}(\omega)\right|, \xi(\omega), \gamma(\omega)\right) = H(\omega)|X(\omega)| \tag{2.41}$$

It is reported that such a log MMSE estimator has fewer musical noise (artifacts) compared to ML estimator [37]. It is mentioned in [38] that such a better performance of log MMSE method is a result of having a filter as a function of *a-priori* SNR. The *a-posteriori* SNR has less influence on noise suppression and since the ML method is more related to the *a-posteriori* SNR it has lower performance compared to log MMSE method.

2.1.4.3 Maximum A-Posteriori

As discussed in 2.1.4.2, the MMSE estimator is actually the mean of the *a-posteriori* SNR. For MAP estimator, the cost-of-error function will be as $C\left(\left|\widehat{\boldsymbol{S}}\right|, |\boldsymbol{S}|\right) = 1 - \delta\left(\left|\widehat{\boldsymbol{S}}\right| - |\boldsymbol{S}|\right)$ and by replacing it in (2.27) the Bayesian risk function based on MAP criterion will be as:

$$R_{MAP}\left(\left|\widehat{\boldsymbol{S}}\right||\boldsymbol{X}\right) = \int_{|\boldsymbol{S}|}\left(\left|\widehat{\boldsymbol{S}}\right| - |\boldsymbol{S}|\right)^2 f_{\boldsymbol{S}|\boldsymbol{X}}(|\boldsymbol{S}||\boldsymbol{X})d|\boldsymbol{S}| = 1 - f_{\boldsymbol{S}|\boldsymbol{X}}\left(\left|\widehat{\boldsymbol{S}}\right||\boldsymbol{X}\right) \tag{2.42}$$

In (2.42), the minimum value for the risk function corresponds to the $|\boldsymbol{S}|$ value for which the posterior function attains a maximum. Since the MAP estimate of $|\boldsymbol{S}|$ is equivalent to the minimization of the risk function or the maximization of the posterior function as:

$$\begin{aligned}\left|\hat{S}(\omega)\right| &= \arg\max_{|S(\omega)|} f_{\boldsymbol{S}|\boldsymbol{X}}\left(|S(\omega)||X(\omega)\right) \\ &= \arg\max_{|S(\omega)|} f_{\boldsymbol{X}|\boldsymbol{S}}(X(\omega)||S(\omega)|) f_{\boldsymbol{S}}(|S(\omega)|)\end{aligned} \tag{2.43}$$

If the a-posteriori distribution is Gaussian then the MAP and MMSE estimators will become equal. There are some discussion on the MAP estimator on non-Gaussian PDFs in [27].

2.2 Binary Time-Frequency masking

In these speech enhancement algorithms, a time-frequency mask is used to remove the noise from the noisy speech. These masks are matrices of time-frequency scaling factors which are applied to the spectrogram of the noisy speech as:

$$\left|\hat{S}(j,\omega)\right| = M(j,\omega)|X(j,\omega)| \tag{2.44}$$

where $M(j,\omega)$ is a vector of mask values for the frequency bins ω applied to the j-th frame of the noisy speech and we have $0 \leq M(j,\omega) \leq 1$. If we use any value in between 0 and 1 as the mask, it is called soft-decision mask and it is like conventional filtering method which discussed in section 2.1. The values for $M(j,\omega)$ can also be binary (0 or 1) and it can be shown as:

$$M(j,\omega) = \begin{cases} 1 & \text{if speech} \\ 0 & \text{otherwise} \end{cases} \tag{2.45}$$

In this way the non-speech regions will be completely removed. Application of such a binary mask will still result in some noise in the enhanced speech but it also exhibits higher intelligibility for the enhanced signal [39]. To attain better binary mask, we can measure the *a-priori* SNR at each time-frequency component and then set a cut off level where the noise is more powerful than the speech as:

$$M(j,\omega) = \begin{cases} 1 & \text{if } 10\log_{10}\left(\dfrac{|S(j,\omega)|^2}{|N(j,\omega)|^2}\right) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.46}$$

where $10\log_{10}\left(\frac{|S(j,\omega)|^2}{|N(j,\omega)|^2}\right)$ is the instantaneous *a-priori* SNR in decibel. The *a-priori* SNR should be estimated since it is often unknown. In [8] am method introduced to estimate *a-priori* SNR using a gain function and *a-posteriori* SNR as:

$$\xi(j,\omega) = \alpha \frac{(H(j-1,\omega)|X(j-1,\omega)|)^2}{\left|\widehat{N}(j-1,\omega)\right|^2} + (1-\alpha)\max(\gamma(j,\omega)-1,0) \qquad (2.47)$$

Where empirically for good tracking $0 < \alpha < 1$ and (we choose $\alpha = 0.98$) and $H(j-1,\omega)$ is a gain function as discussed before. $\gamma(j,\omega)$ is the *a-posteriori* SNR and $|X(j-1,\omega)|$ and $|N(j-1,\omega)|$ are the noisy speech and noise spectral magnitudes of the previous frame respectively. The gain function (filter) can be calculated using all the previously discussed estimators. In addition to the type of estimator, the performance of this method is very dependent to the accuracy of noise estimation. In [40] a range of gain functions and noise estimators have been tested to find the best combination. It has been reported in [27] that the MMSE-based methods using VAD-based or MCRA2 noise estimators are of the best performance. Some more experiments on the variation of binary mask using different estimators have been mentioned in [41, 42].

2.3 Subspace enhancement

In the previous speech enhancement methods it is assumed that the effect of noise can be removed from the noisy speech by means of filtering. In subspace methods it is assumed that the clean speech occupies a small subspace of the noisy speech space and hence some other subspaces or the whole noisy speech space is occupied by noise. In these methods, the aim is to detect the subspaces which are exclusively occupied by noise and remove them and then through resynthesizing the modified frames recover the estimate of the clean speech. Practically the subspaces of speech and noise are not separate and can be highly overlapped and hence some other considerations should be

taken to effectively remove the noise. These methods are generally implemented in 3 steps as:

1) Separating the subspaces of clean speech plus noise and pure noise
2) Removing the noise subspace
3) Pot-processing the subspace of clean speech plus noise to remove the effect of noise from the clean speech

In the second step, the noise subspace will be removed without doing any modifications on the speech signal. The third step is of great importance since it will modify the resulted enhanced speech but on the other hand it can introduce some distortion to the enhanced speech due to the existing overlap between the noise and speech subspaces [43]. In [44] a subspace method discussed with the assumption of additive noise in the noisy speech as in (2.1) which is $x(t) = s(t) + n(t)$. In this method some short time frames are applied to the long time domain signals to preserve the assumption of stationarity for them.

$$\boldsymbol{s} = \boldsymbol{\Psi y} \tag{2.48}$$

where $\boldsymbol{s} = s(t)$, $\boldsymbol{\Psi}$ is a rank deficient $K \times M$ matrix with rank M where $M < K$ and $\boldsymbol{y}$ is a weighting vector of size $M \times 1$. To be able to separate the subspaces occupied by speech and noise, the $\boldsymbol{\Psi}$ should be rank deficient [43]. As discussed in [27], from such a linear model, a linear estimator can be computed as:

$$\hat{\boldsymbol{s}} = \boldsymbol{H}.\boldsymbol{x} \tag{2.49}$$

in which the optimal estimator $\boldsymbol{H}$ is defined as:

$$\boldsymbol{H} = \boldsymbol{\Sigma}_s(\boldsymbol{\Sigma}_s + \mu\boldsymbol{\Sigma}_n)^{-1} \tag{2.50}$$

where $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_n$ represent the covariance matrix of the clean speech and noise respectively. During the non-speech portion of the noisy speech signal, the value of $\boldsymbol{\Sigma}_n$

can be calculated but since $\boldsymbol{\Sigma}_s$ is not available, another approach for the calculation of $\mathbf{H}$ should be used. We define a matrix $\boldsymbol{J}$ as:

$$\boldsymbol{J} = \boldsymbol{\Sigma}_n^{-1}\boldsymbol{\Sigma}_x - \boldsymbol{I} \tag{2.51}$$

where $\boldsymbol{I}$ is the identity matrix. Using eigenvector decomposition, the eigenvectors and eigenvalues of $\boldsymbol{J}$ can be calculated as:

$$\boldsymbol{JV} = \boldsymbol{V\Lambda}_s \tag{2.52}$$

where $\boldsymbol{V}$ and $\boldsymbol{\Lambda}_s$ represent the eigenvectors and eigenvalues of $\mathbf{J}$ respectively. Assuming that the largest eigenvalues represent the signal, through setting the non-positive eigenvalues we can remove the noise subspace. Based on (2.49) and (2.50), the proper estimator $\mathbf{H}$ can be calculated as

$$\boldsymbol{H} = \boldsymbol{VGV}^T \tag{2.53}$$

where $\boldsymbol{G}$ is a $K\times K$ matrix with the diagonal elements as:

$$\boldsymbol{G} = \begin{cases} 1 & \text{for } \Lambda(k,k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } k = 1, \dots, K \tag{2.54}$$

Using the resulted estimator in (2.49) we can resynthesize the noisy speech signal with the noise subspace removed. The resulting resynthesized subspace is still affected by noise and to retrieve a good quality estimate of clean speech some more processing should be performed. In this phase one of the speech enhancement methods discussed in previous sections can be used, which Wiener filter can be a good candidate. Such a Wiener filter can be constructed as:

$$G(k,k) = \begin{cases} \dfrac{\Lambda(k,k)}{\Lambda(k,k)+\mu} & \text{for } \Lambda(k,k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } k = 1, \dots, K \tag{2.55}$$

in which μ is the Lagrange multiplier as:

$$\mu = \begin{cases} \mu_0 - \dfrac{SNR_{dB}}{d} & -5 < SNR_{dB} < 20 \\ 1 & SNR_{dB} \geq 20 \\ 5 & SNR_{dB} \leq -5 \end{cases} \tag{2.56}$$

which as mentioned in [44] has $\mu_0 = 4.2$ and $d = 6.25$. In this way using μ from (2.56) in (2.55) and then the resulting $\boldsymbol{G}$ from (2.55) in (2.53) the optimal estimator of $\boldsymbol{H}$ can be calculated. Such an estimator will null the noise subspace and also suppress the noise in the noisy speech, simultaneously.

2.4 Speech enhancement by reconstruction

These methods are somehow similar to the conventional speech enhancement methods. The difference is that instead of using inverse Fourier transform to resynthesize (recover) the whole enhanced speech after the filtering process, the speech is reconstructed using a construction model that is driven by a set of acoustic features. These methods could be divided in the following steps:

1) Acoustic feature extraction (Analysis)
2) Acoustic feature enhancement
3) Speech reconstruction using the enhanced acoustic features (Synthesis)

The speech reconstruction models were developed to be used in channel coding but they have some properties that make them such a good tool for speech enhancement. One of the benefits of the speech reconstruction model is the constraints that it applies on the reconstructed signal which makes is more accurate than the normal re-synthesis process. The reconstruction models are designed in a way to only reconstruct the speech related components and those ones which are related to noise will not be reconstructed. Using these models with some previously discussed filtering methods can result in more noise reduction as mentioned in [45, 46]. The use of these reconstruction models as a post-

filtering process in conventional methods, is mentioned in [47] where before the speech reconstruction using the LPC vocoders, a spectral subtraction is used for the noise reduction. Such a method but using Wiener filtering instead of spectral subtraction is discussed in [48]. These methods resulted in less artifacts in enhanced speech while it was almost inevitable in conventional methods. In [49] a conventional post-processing method is introduced in which the regions of speech spectra which are distorted within noise reduction process will be reconstructed. To do so, a reconstruction model called Harmonic Noise pulse Model (HNM) is used to reconstruct the damaged harmonics [50]. In the HNM method, the speech is reconstructed as a sum of harmonic sinusoids which are modulated by amplitude and frequency and offset for relative phase:

$$s_r(t) = \sum_{l=1}^{L} A(lf_0)\cos\big(2\pi l f_0 t + \theta(lf_0)\big) + n_r(t) \tag{2.57}$$

where $s_r(t)$ is the t-th sample of the reconstructed speech signal, L is the total number of harmonics, $A(lf_0)$ is the value of spectral envelope sampled at the l-th harmonic in which f_0 is the fundamental frequency, θ is the phase spectrum and $n_r(t)$ is the filtered noise. Using this method we can make sure that the speech energy is reconstructed in voiced frames. In [49], the harmonic amplitude $A(lf_0)$ and frequencies lf_0 are tracked and the damaged or missing harmonic components are recovered using codebooks. Such codebooks are trained on the clean speech signals. Another method discussed in [51] for suppression of the noise of a recorded speech in the car. In such environments, high frequency components are of higher SNR and vice versa because of engine and wind noise. At first a conventional speech enhancement method will be used and then using an IIR filter the spectral envelope is extracted. For low SNR regions of speech, a codebook is used and for voiced speech, by using inverse Fourier transform, the signal is reconstructed at harmonic frequencies to generate a reconstruction model like those ones of the HNM method.

The HNM method can also be used as an enhancement method rather than a post-filtering method. Such a method is discussed in [52] where an iterative process of Wiener filtering was used to estimate the required acoustic features and these features could be used for reconstruction and noise reduction stages. Another method in [53] uses spectral subtraction in pre-processing phase to analyze the spectral envelope. A very advanced feature extraction is discussed in [54] where the HNM is used for reconstruction. In this method through analysis of a pre-cleaned speech the voice activity and fundamental frequencies are estimated through the use of time-frequency tracking. All these methods revealed good enhancement results with no musical noise but some distortion in the signal.

2.5 Summary

A variety of speech enhancement methods have been introduced in this section. Some methods are based on filtering and are so simple to implement but suffer from some lack of accuracy. Some other methods which are more complicated but more accurate are also introduced which make use of some models for speech and noise distributions. In the next chapter we are going to introduce methods in which the model based and filter based methods are mixed to get to more accurate speech enhancement methods.

3 Speech enhancement using the combination of conventional and reconstruction methods

As discussed in chapter 2 the conventional methods are very simple to implement and they can easily be used in different applications, but they suffer from a residual musical noise. It is also reported that the reconstruction methods are of high accuracy but they are more complicated and need some models trained on clean speech or even different noises. Here we are going to discuss two algorithms in which a combination of conventional and reconstruction methods can be used and such methods are the concentration of this thesis.

Using the assumptions made in section 2.1.3 as shown in (2.23) we can say that the sum of the power spectrum of clean speech and noise is equal to the power spectrum of the noisy speech and by defining the power spectrum as the periodogram, we can rewrite (2.23) as:

$$P_x(\omega) = P_s(\omega) + P_n(\omega) \tag{3.1}$$

where P_x, P_s and P_n are the periodograms of noisy speech, clean speech and noise respectively. Using this definition, the two-sided Wiener filter transfer function in (2.4) can be rewritten as:

$$W(\omega) = \frac{P_s(\omega)}{P_s(\omega) + P_n(\omega)} \tag{3.2}$$

By using (3.1) to replace the denominator of Wiener filter we can re-write (3.2) as:

$$W(\omega) = \frac{P_s(\omega)}{P_x(\omega)} \tag{3.3}$$

Such a Wiener filter can be effectively used for the enhancement of any noisy frame if we can accurately estimate the periodograms of clean speech and noise. In this chapter we are going to concentrate on the methods which can be used to estimate these periodograms.

3.1 Voice Activity Detection (VAD)

Using a VAD, we can show that a frame of a signal is voiced, unvoiced or silent. Voiced frames are of higher energy than unvoiced or silent frames. Unvoiced frames are more like noise and also of higher energy than the silent frames. The silent frames have the least energy and what exists within them can be considered as the environment noise [55]. The VAD is of great use for noise estimation in the methods based on spectral subtraction since it will increase the accuracy of noise estimation and hence decrease the resulting distortion and musical noise in the enhanced speech. A VAD is used to make the decision when to update noise information as discussed in section 2.1.2. If in a VAD it is considered that the first frames are silent and just contain noise and hence we can have an initial estimate of the statistics of noise in terms of mean and standard deviation, then we can proceed. A threshold can then be calculated, which can be used for making the decision about a frame being speech or noise (silent) and in this way in case of silent frame detection all its statistical components will be updated. If we take $X_k(\omega)$ as the spectrum (Fourier transform) of the k-th frame of the noisy speech then we can find the initial estimate of the noise spectral amplitude as:

$$|N_1(\omega)| = |X_1(\omega)| \tag{3.4}$$

And hence the initial estimate of noise mean can be calculated as:

$$\mu_{N_1} = \frac{1}{\Omega}\sum_{\omega=0}^{\Omega}|N_1(\omega)| \tag{3.5}$$

When the mean of the observed signal is close to the noise mean, it represents that there is no signal. If the algorithm detects that there is no speech signal, then the noise spectral amplitude, mean and the standard deviation of current frame will be updated. The first couple of frames that are taken as noise will result in more accurate initial estimates of the noise properties. The noise spectral amplitude update process for the next frames ($k > 1$) will be as:

$$|N_k(\omega)| = \alpha|N_{k-1}(\omega)| + (1-\alpha)|X_k(\omega)| \tag{3.6}$$

The update process for the mean value will be as:

$$\mu_{N_k} = \beta\mu_{N_{k-1}} + \frac{(1-\beta)}{\Omega}\sum_{\omega=0}^{\Omega}|N_k(\omega)| \tag{3.7}$$

where μ_{N_k} is the mean of noise spectral amplitude in the k-th frame. The update process for the standard deviation will be as:

$$\sigma_{N_k} = \sqrt{\beta\sigma_{N_{k-1}}^2 + (1-\beta)\mu_{N_k}^2} \tag{3.8}$$

where σ_{N_k} and $\sigma_{N_k}^2$ are the standard deviation and variance of noise spectral amplitude in the k-th frame, respectively. The update parameters of α and β are both taken 0.95 and relate the estimation of noise properties of each frame to the previous frames. In this way each estimate is smoothed with the previous ones [56]. The thresholds used to detect whether a frame is noise or speech can be determined as follows. If the current frame is noise, then it will be updated using the standard deviation and mean of noise. These thresholds can be defined as:

$$T_S = \mu_N + \alpha_S\sigma_N \tag{3.9}$$

$$T_N = \mu_N + \alpha_N \sigma_N$$

where T_S and T_N are the thresholds of speech and noise and α_S and α_N are the coefficients of speech and noise that can be calculated experimentally. The VAD decision-making can be implemented using T_S and T_N in a way that when the energy of the current frame is two times the noise standard deviation more than noise mean, then the frame is taken as speech ($\alpha_S = 2$). If the energy of the frame is a fraction of noise standard deviation, the frame will be taken as noise ($\alpha_N < 1$). This decision making can be shown as:

$$\begin{gathered} \text{if Energy}(k) > T_S \text{ then } VAD(k) = 1 \\ \text{if Energy}(k) < T_N \text{ then } VAD(k) = 0 \\ \text{else } VAD(k) = VAD(k-1) \end{gathered} \tag{3.10}$$

As can be seen, making a wrong decision about a frame being noise or signal is probable and if the properties of noise change from one frame to another or the noise shows nonstationary behavior then the probability of wrong decisions will be even higher. Using the resulting noise spectral amplitude estimate from (3.6) we can find the noise periodogram as $P_n(\omega) = |N(\omega)|^2$. Since we can find the noisy speech periodogram from the observed noisy speech as $P_x(\omega)$, by substituting $P_n(\omega)$ and $P_x(\omega)$ in (3.1) we can calculate an estimate of clean speech periodogram as $P_s(\omega)$. By substituting the resulting $P_s(\omega)$ and $P_n(\omega)$ in (3.2), the suitable Wiener filter for the enhancement of the current noisy frame can be constructed.

3.2 Minimum statistics

Despite the VAD method in which an algorithm is used to detect the speech or noise frames and the noise estimation was only updated in the noise frames, in Minimum

Statistics the minimums of the sub bands of the smoothed power spectral amplitude of noisy speech are used to detect noise boundaries. This algorithm is dependent on the maxima and minima of the power spectral amplitude of short time frames of the noisy speech. The maxima represent the speech activity and the minima of the smoothed power spectral amplitude can be used for the estimation of sub band noise [25, 57]. In this method it is considered that the relationship in (3.1) exist between the periodograms of clean speech, noisy speech and noise. To find the noise estimate we define a smoothed power spectral amplitude using exponential smoothing accordingly:

$$P_{SM_k}(\omega) = \alpha P_{SM_{k-1}}(\omega) + (1-\alpha)|X_k(\omega)|^2 \tag{3.11}$$

where $P_{SM_k}(\omega)$ is the smoothed power spectral amplitude of the k-th frame, $|X_k(\omega)|^2$ is actually $P_{x_k}(\omega)$ which is the power spectral amplitude of the noisy speech of the k-th frame and α is the smoothing coefficient which ranges as $0.9 \leq \alpha \leq 0.95$. The estimate of noise power spectral amplitude can be attained as a weighted minimum of the smoothed power spectral amplitude as:

$$P_{n_k}(\omega) = O_{min} \min\left(P_{SM_k}(\omega)\right) \tag{3.12}$$

where $\min\left(P_{SM_k}(\omega)\right)$ is the estimated minimum power and O_{min} is a coefficient for the compensation of bias of the estimated minimum power. Knowing that the power spectrum has Ω frequency components, a buffer with b vectors of size Ω are used to keep the smoothed power spectrum $P_{SM_k}(\omega)$. This buffer will be initialized with values larger than the power spectral amplitude (periodogram) of the first frame of the noisy speech. After entering the noisy speech periodogram to smoothing equation of (3.11), the minimum of the resulting smoothed periodogram and the first vector of the buffer will replace the value of the first vector of the buffer. After reading a pre-defined number of noisy speech frames say f, all the vectors of the buffer will be shifted to the

right (the vector 1 will replace vector 2, vector 2 will replace vector 3 and so on) so vector $b-1$ will replace vector b. The estimate of noise periodogram $P_{n_k}(\omega)$ from (3.12) will be attained by finding the minimum of each frequency component over all b vectors of the buffer. The value of f which shows the number of frames that are averaged, and by reading this number of frames all the values of the buffer will be shifted, should be large enough to cover all the maximums of speech activity and small enough to track a variety of nonstationary noises. In the practical implementation of this algorithm $b=4$ and $f=46$ are considered. The smoothed periodogram and the estimated noise periodogram using it are shown in Figure 3.1.

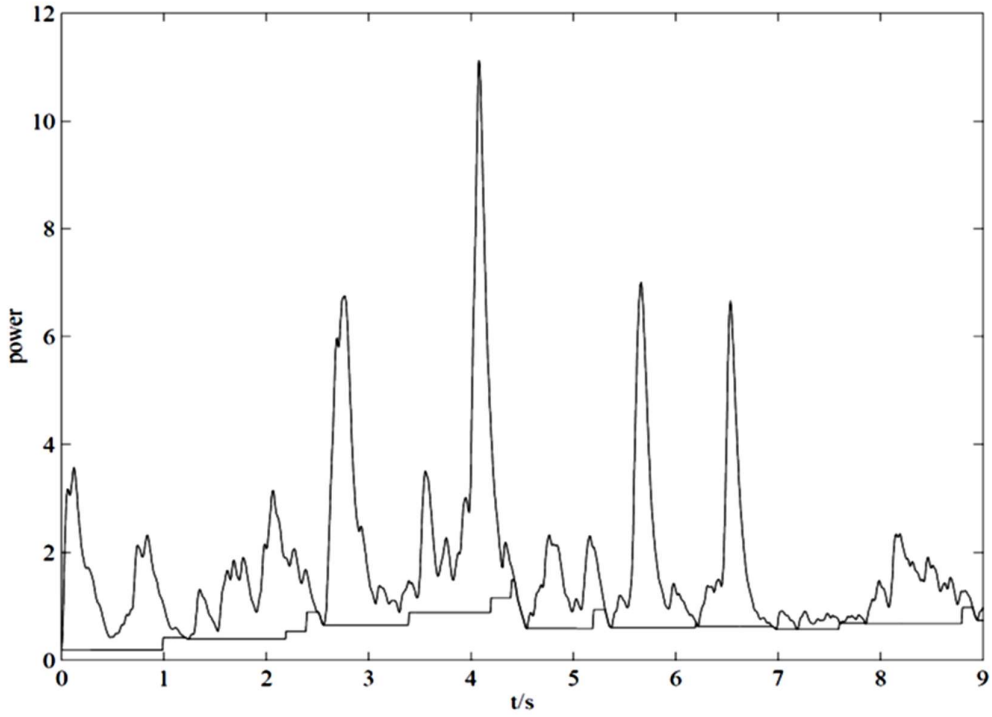

Figure 3.1:The smoothed noisy speech periodogram and the estimated noise periodogram as the minimums [25]

In the practical implementation of the Minimum Statistics algorithm, the bias of the first frame is used for the calculation of O_{min}. Since it is considered that the first couple of frames contain no speech signal, the periodogram of these frames is considered as the

periodogram of noise and by dividing the energy (sum of periodogram components) of these first frames to the energy of the resulted periodogram from the noise estimation algorithm, the suitable bias can be calculated. This value can be used in the next frames as the coefficient of the noise estimation algorithm.

Using the resulting noise periodogram from (3.12) and replacing it in (3.1) we can calculate an estimate of clean speech periodogram as $P_s(\omega)$. By substituting $P_s(\omega)$ and $P_n(\omega)$ in (3.2), the suitable Wiener filter for the enhancement of the current noisy frame can be constructed.

3.3 Codebook constrained speech and noise estimation

As discussed in section 2.4, the reconstruction methods use a model of speech or noise as a reference for better enhancement of noisy speech. Here we are going to concentrate on the codebook constrained Wiener filter. Using VAD and Minimum Statistics methods, we can estimate noise and speech periodograms but there is no guarantee that all the components in the estimated speech periodogram belong to the speech. Using a reference model we can improve the enhanced speech periodogram and maintain as many number of speech components as possible.

3.3.1 Full search codebook

In this method some codebooks for the speech and noise periodograms are created. To create such codebooks, a large database of different speech files and also different noise types should be collected. These codebooks can preserve a relationship between the frequency components of the speech and noise periodograms which is mostly in terms of the shape of the periodograms. The energy of a speech signal is mostly concentrated in lower frequencies and hence speech periodograms have large peaks at low

frequencies and in high frequencies they almost have no energy. Noises such as White noise have energy in almost all frequencies and hence their periodograms have a flat shape having components in all frequencies. Other types of noises have their own shape of periodograms which are notably different from speech periodogram shapes due to the physics of generation which is different from speech. Speech is generated through the vibrations of the vocal chords and this will end up in some specific fundamental frequencies and their harmonics while the other noises are generated from some devices such as cars, destroyer engines, trains and etc. whose fundamental frequencies and frequency ranges are totally different from speech.

After collecting a reasonable amount of observation files of different signal types, speech and different noises, we should construct the codebooks on their periodograms. The larger the datasets the more accurate the resulting codebooks will be. In the same way that the noisy speech is divided into some overlapping frames to preserve the stationarity of the signals, the files that are going to be used in codebook construction process will be divided into overlapping windowed frames (Hamming window in this work). Using the DFT (practically an FFT) all these frames will be transformed to the frequency domain and then by finding the squared value of their spectral amplitude the periodograms of all these frames will be calculated. The numerous periodograms of each signal type (either speech or noise) will be entered into their corresponding dataset of periodograms. Since we want to classify these periodograms based on their shapes to have each cluster in charge of one specific shape of periodogram, all the periodograms in the dataset will be normalized which means making their energy equal to one and this can be achieved through dividing each periodogram by the sum of its components. Now these datasets should be classified into some clusters using clustering algorithms which in this case is called K-means. K-means algorithm will look for the members that are closer to each other in an iterative manner. The criterion for the distance is the Euclidian

distance. The number of clusters in such an algorithm should be set manually and after summing we will end up with some clusters each containing periodograms of similar shapes. By finding the mean periodogram (average of all periodograms) of the cluster and finding the periodogram in that cluster which is the closest to this mean vector, we can find the centroid of that cluster. In this way numerous periodograms of the dataset (nearly one million periodograms in our experimental datasets) are classified into some clusters (say 100) and for each cluster the centroid is calculated and from now on these centroids will represent the whole large dataset.

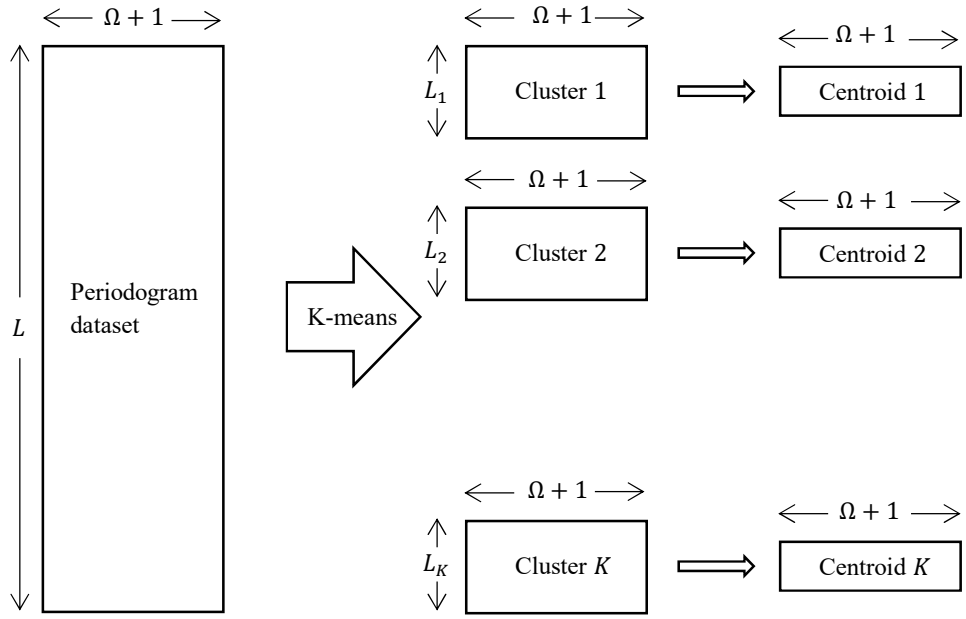

Figure 3.2: Full-search codebook construction procedure

In Figure 3.2, a large periodogram dataset is shown with as a collection of L vecrtors of length Ω (number of frequency components). Using K-means this dataset is divided to K cluster each containing some periodograms which are closer to each other and we have $L = L_1 + L_1 + \cdots + L_K$. Finding the centroids of each cluster gives the codebook centroids as K vectors of length $\Omega + 1$.

3.3.2 **Tree-structured codebooks**

Another method that can be used for creating such clusters is called tree-structured codebooks. In these codebooks the large periodogram datasets of each signal type will be classified into two clusters using the K-means algorithm. On the next step, each of these two clusters will be classified into two more clusters using the K-means algorithm and hence we will have 4 clusters. This procedure is shown in Figure 3.3.

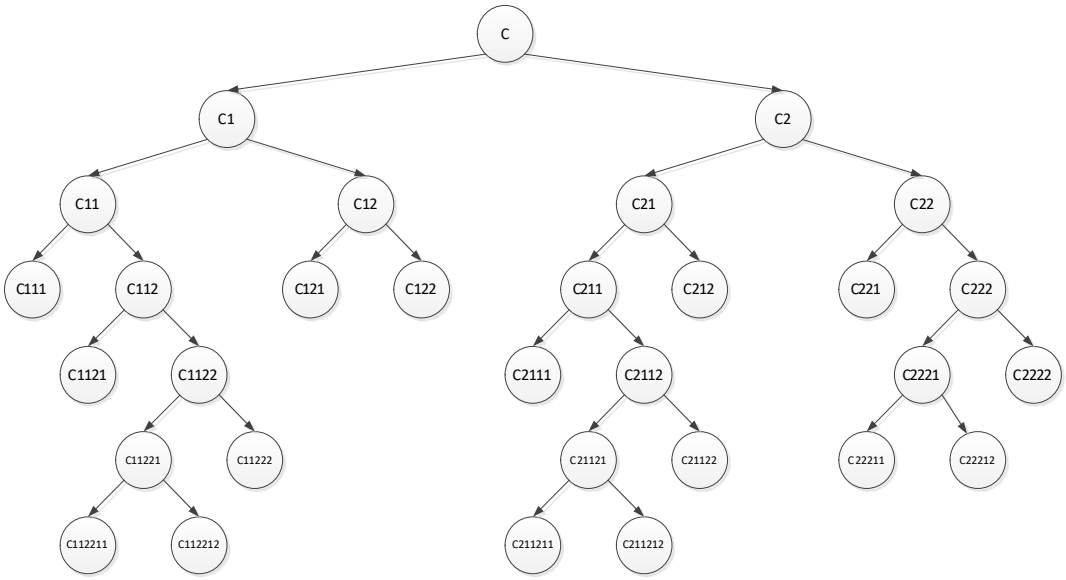

Figure 3.3: Tree-structured codebook clustering procedure

As can be seen from Figure 3.3, in each level of this procedure the clusters from the previous level will be divided into two new clusters using the K-means algorithm [58]. The number of levels for this procedure can be found by experiment. Since this procedure can continue till there is just one member in the very lower level clusters and this is against the idea of clustering, we need to set some restrictions on the clustering process to stop it. These restrictions can be as the minimum number of members in each cluster or the average distance of the members from the centroid. We can stop the division of clusters to the two new clusters when these restrictions are violated. In this

way we can make sure that we have reliable centroids to represent the whole dataset. Such procedure is discussed in [59, 60].

3.3.3 Periodogram estimation by solving a set of over-determined equations

Now that we were able to decrease the variety of periodograms of the observed dataset by clustering it into some clusters and representing those clusters with their centroids, we need to find a way of applying these periodogram models in speech and noise periodogram estimation. We can assume that the periodogram of the noisy speech in the current frame is $P_x(\omega) = \left[P_{x_1}, P_{x_2}, \dots, P_{x_\Omega}\right]$ which shows the total Ω frequency components of the periodogram. We assume that we have a speech codebook of size I which means that it has I centroid periodograms and each periodogram can be shown as $P_{s,i}^{cb}(\omega) = \left[P_{s,i}^{cb}{}_1, P_{s,i}^{cb}{}_2, \dots, P_{s,i}^{cb}{}_\Omega\right]$ where $i = 1, \dots, I$. In the same manner we can introduce a noise codebook of size J as $P_{n,j}^{cb}(\omega) = \left[P_{n,j}^{cb}{}_1, P_{n,j}^{cb}{}_2, \dots, P_{n,j}^{cb}{}_\Omega\right]$ where $j = 1, \dots, J$. Since we assumed that the periodogram of noisy speech is equal to the sum of the periodograms of speech and noise and we want to use the created codebooks of speech and noise as the reference of the periodogram, we will consider all the speech and noise codebook centroids as the building block of the noisy speech periodogram. In this way we are assuming that each noisy speech periodogram can be formed as weighted sum of one speech periodogram centroid and one noise periodogram centroid. The weightings are actually a compensation of the bias since the codebooks are created on normalized periodograms (power equal to 1) while in practice the periodograms can be of any power. To do so, we need to find the two centroids (one from speech codebook and one from noise codebook) that their weighted sum is the closest to the noisy speech periodogram. In this way we will have:

$$P_x(\omega) = a_{i,j} P_{s,i}^{cb}(\omega) + b_{i,j} P_{n,j}^{cb}(\omega) + e_{i,j}(\omega) \tag{3.13}$$

where $e_{i,j}(\omega)$ represents the error between the real periodogram of the noisy speech and the one estimated using the codebook centroid periodograms. Since $a_{i,j}$ and $b_{i,j}$ are constants and are multiplied by all the Ω components of $P_{s,i}^{cb}(\omega)$ and $P_{n,j}^{cb}(\omega)$ we can expand (3.13) as:

$$\begin{cases} P_{x_1} = a_{i,j}\, P_{s,i}^{cb}{}_1 + b_{i,j}\, P_{n,j}^{cb}{}_1 + e_{i,j}{}_1 \\ P_{x_2} = a_{i,j}\, P_{s,i}^{cb}{}_2 + b_{i,j}\, P_{n,j}^{cb}{}_2 + e_{i,j}{}_2 \\ \quad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ P_{x_\Omega} = a_{i,j}\, P_{s,i}^{cb}{}_\Omega + b_{i,j}\, P_{n,j}^{cb}{}_\Omega + e_{i,j}{}_\Omega \end{cases} \tag{3.14}$$

In (3.14), $a_{i,j}$ and $b_{i,j}$ are unknown coefficients and since they are going to bias the centroid periodograms of the codebooks we need to find them in this equation. To find these coefficients we want to minimize the MSE of the $e_{i,j}(\omega)$ vector. This value can be calculated as:

$$E_{i,j} = \frac{1}{\Omega}\sum_{\omega=1}^{\Omega} e_{i,j}{}_\omega^2 = \frac{1}{\Omega}\sum_{\omega=1}^{\Omega} \left(a_{i,j}\, P_{s,i}^{cb}{}_\omega + b_{i,j}\, P_{n,j}^{cb}{}_\omega - P_{x}{}_\omega\right)^2 \tag{3.15}$$

where $E_{i,j}$ represents the MSE corresponding to the use of i-th centroid of the speech codebook and j-th centroid of the noise codebook. To minimize this MSE we need to find its partial fractions with respect to $a_{i,j}$ and $b_{i,j}$ and equate them to zero [60].

$$\begin{cases} \dfrac{\partial E_{i,j}}{\partial a_{i,j}} = \dfrac{2}{\Omega}\displaystyle\sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_\omega \left(a_{i,j} P_{s,i}^{cb}{}_\omega + b_{i,j} P_{n,j}^{cb}{}_\omega - P_{x}{}_\omega\right) = 0 \\ \dfrac{\partial E_{i,j}}{\partial b_{i,j}} = \dfrac{2}{\Omega}\displaystyle\sum_{\omega=1}^{\Omega} P_{n,j}^{cb}{}_\omega \left(a_{i,j} P_{s,i}^{cb}{}_\omega + b_{i,j} P_{n,j}^{cb}{}_\omega - P_{x}{}_\omega\right) = 0 \end{cases} \tag{3.16}$$

By simplifying (3.16) we will have:

$$\begin{cases} a_{i,j} \sum_{\omega=1}^{\Omega} \left(P_{s,i}^{cb}{}_{\omega}\right)^2 + b_{i,j} \sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_{\omega} P_{n,j}^{cb}{}_{\omega} = \sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_{\omega} P_{x}{}_{\omega} \\ a_{i,j} \sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_{\omega} P_{n,j}^{cb}{}_{\omega} + b_{i,j} \sum_{\omega=1}^{\Omega} \left(P_{n,j}^{cb}{}_{\omega}\right)^2 = \sum_{\omega=1}^{\Omega} P_{nj}^{cb}{}_{\omega} P_{x}{}_{\omega} \end{cases} \quad (3.17)$$

Using (3.17) we can find the values of $a_{i,j}$ and $b_{i,j}$ as:

$$a_{i,j} = \frac{\left(\sum_{\omega=1}^{\Omega}\left(P_{n,j}^{cb}{}_{\omega}\right)^2\right)\left(\sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_{\omega} P_{x}{}_{\omega}\right) - \left(\sum_{\omega=1}^{\Omega} P_{n,j}^{cb}{}_{\omega} P_{x}{}_{\omega}\right)\left(\sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_{\omega} P_{n,j}^{cb}{}_{\omega}\right)}{\left(\sum_{\omega=1}^{\Omega}\left(P_{s,i}^{cb}{}_{\omega}\right)^2\right)\left(\sum_{\omega=1}^{\Omega}\left(P_{n,j}^{cb}{}_{\omega}\right)^2\right) - \left(\sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_{\omega} P_{n,j}^{cb}{}_{\omega}\right)^2}$$
$$b_{i,j} = \frac{\left(\sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_{\omega} P_{x}{}_{\omega}\right)\left(\sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_{\omega} P_{n,j}^{cb}{}_{\omega}\right) - \left(\sum_{\omega=1}^{\Omega}\left(P_{s,i}^{cb}{}_{\omega}\right)^2\right)\left(\sum_{\omega=1}^{\Omega} P_{n,j}^{cb}{}_{\omega} P_{x}{}_{\omega}\right)}{\left(\sum_{\omega=1}^{\Omega} P_{s,i}^{cb}{}_{\omega} P_{n,j}^{cb}{}_{\omega}\right)^2 - \left(\sum_{\omega=1}^{\Omega}\left(P_{s,i}^{cb}{}_{\omega}\right)^2\right)\left(\sum_{\omega=1}^{\Omega}\left(P_{n,j}^{cb}{}_{\omega}\right)^2\right)} \quad (3.18)$$

For the implementation of such algorithm in MATLAB we can use *pinv* function. Solving these overdetermined equations might lead to negative values for $a_{i,j}$ and $b_{i,j}$ and since multiplying these values in the codebook centroids will lead to negative periodograms which is impossible, we should use the positive values of these coefficients. In this way the MSE from the calculated $a_{i,j}$ and $b_{i,j}$ for a frame will be sorted in ascending order and the first pair of $a_{i,j}$ and $b_{i,j}$ which are both positive will be considered as the biasing coefficients of the current frame. In this way the proper indexes of i and j can be found accordingly as:

$$(i,j) = \arg\min_{i,j} E_{i,j} \qquad \text{where} \qquad a_{i,j}\,, b_{i,j} > 0 \quad (3.19)$$

As discussed in [60], since we have different codebooks for different noise types such over-determined equations will be solved between the speech codebook centroids and each noise codebook centroids and their corresponding $E_{i,j}$ and also resulted $a_{i,j}$ and $b_{i,j}$ will be recorded and at the end the $a_{i,j}$ and $b_{i,j}$ coefficients that exhibited the minimum value of $E_{i,j}$ will be considered as the biasing coefficients of codebook

centroids. Using this method we will be able to detect the noise type based on detecting the codebook that exhibited the minimum $E_{i,j}$.

If we use full search codebooks with speech codebook of size I and each noise codebook of size J and totally we have N noise codebooks, then we need to solve $N \times I \times J$ sets of over-determined equation with each set containing Ω equations and hence this noise estimation method using full search codebooks can be too time consuming. Using tree-structured codebooks at each step we can solve the over-determined equations in the same levels from speech and noise codebooks and then by finding the minimum MSE, just go through the branch that it's up-level centroid represented this value. In this way we can exclude some centroids from the over-determined equation- solving process while going deeper in the tree-structured codebook and hence the use of these codebooks is notably faster than the full search codebooks.

After finding the estimate of speech and noise periodograms using such a codebook constrained method we need to construct the proper Wiener filter for the enhancement of the current noisy frame. Such a Wiener filter can be constructed as:

$$W(\omega) = \frac{a_{i,j} P_{s,i}^{cb}(\omega)}{a_{i,j} P_{s,i}^{cb}(\omega) + b_{i,j} P_{n,j}^{cb}(\omega)} \tag{3.20}$$

where i, j, $a_{i,j}$ and $b_{i,j}$ are calculated by solving the over-determined equations on the clean speech and noise periodogram codebooks.

3.4 Gaussian Mixture Modelling of speech and noise periodograms

In codebook constrained periodogram estimation methods, we discussed different methods of constructing codebooks for speech and noise. These codebooks are a reference for the estimation process and actually these methods are case-based in which a collection of observed vectors (periodograms in here) are classified based on their Euclidian distance (using the K-means algorithm) and each class will then be represented with the centroid of that class. The problem in this method is that all the various members of a class that can contain hundreds or thousands of the observed vector, are represented with only one vector which is the centroid of that class. In this way the codebook constrained methods cannot track the fast changing behavior of the speech periodogram and the corresponding periodograms from this method will be smoothed.

Speech signals are the collections of sentences; sentences are the collections of words; words are the collections of vowels and consonants are generated by passing the voice generated by the vibrations of human vocal chords through mouth and nasal paths. Since we have a finite number of vowels and consonants to make our human speech, we can classify them into a finite number of classes. As mentioned before, we divide the noisy speech observation into some overlapping short frames (of length of 20-30 ms) and calculate the periodogram of each frame. The periodogram shows the power of signal in a specific frequency and in this way we have mapped our sharply varying time domain signal space to the smoother frequency domain periodogram space. Since the vowels and consonants that form each short time frame are a power shared between some of these frequency components, for different speakers saying different words, we have slight changes in the periodogram (the frequency powers) representing a specific

vowel or consonant or a combination of them for that frame. In this way, we may have finite number of clusters representing different vowels or consonants or a combination of them in which we can classify a large training periodogram dataset. These clusters have some statistic characteristics such as probability, mean and covariance. The probability characteristic means the portion of periodogram vectors in that cluster with respect to all periodogram vectors. The mean characteristic means the average of periodogram vectors in that cluster. The covariance characteristic means the average distant of all periodogram vectors in that cluster from the mean vector of that cluster. As illustrated in Figure 3.4, each circle can represent a cluster and each dot can represent a periodogram vector.

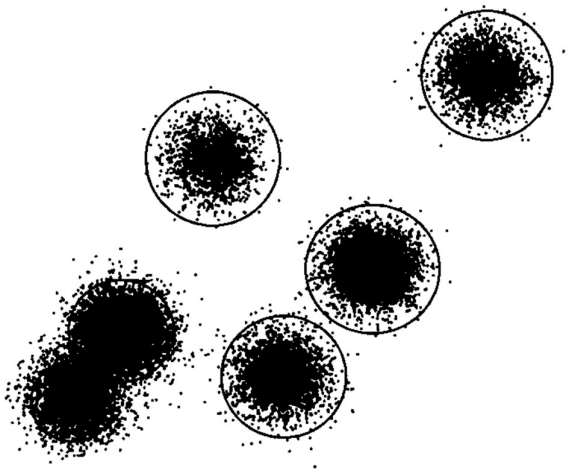

Figure 3.4: GMM classification

The centers of these circles (that represent the mean of that cluster) are quite dense and when we get to the boundaries the density decreases. The population of one cluster could be shown in Figure 3.5.

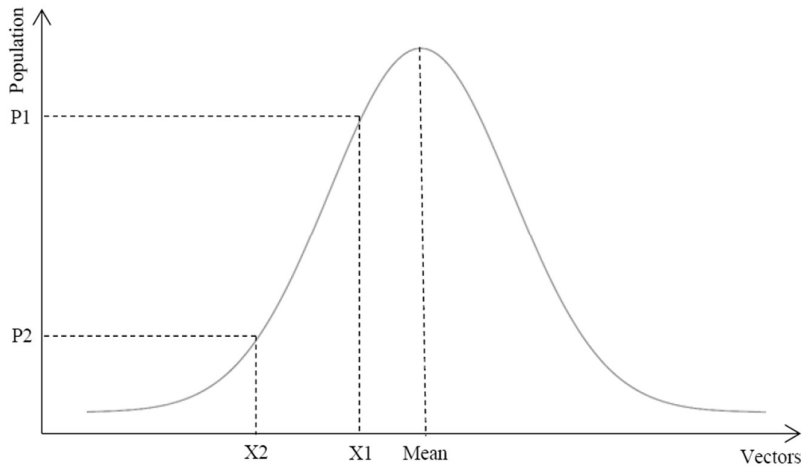

Figure 3.5: Gaussian distribution

The vector X1 is closer to the mean than vector X2 and hence has larger population (P1 > P2). This distribution of vectors is similar to a Gaussian distribution. We can use some Gaussians to model the space of periodogram vectors into some finite clusters and this method is called Gaussian Mixture Modeling [61]. A Gaussian distribution for a sample periodogram $P(\omega)$ can be shown in (3.21).

$$G_k\big(P(\omega);\mu_k(\omega),\Sigma_k(\omega)\big) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_k(\omega)|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\big(P(\omega)-\mu_k(\omega)\big)^T\Sigma_k^{-1}(\omega)\big(P(\omega)-\mu_k(\omega)\big)\right\} \tag{3.21}$$

where index k shows the characteristics of k-th Gaussian, μ is the mean vector, Σ is the covariance matrix and d is the space dimension. Also $|\Sigma_k(\omega)|$ represents the determinant of the covariance matrix $\Sigma_k(\omega)$. The Probability Density Function (PDF) of this sample periodogram can be calculated as:

$$f\big(P(\omega)\big) = \sum_{k=1}^{K}\pi_k G_k\big(P(\omega);\mu_k(\omega),\Sigma_k(\omega)\big) \quad , \quad \sum_{k=1}^{K}\pi_k = 1 \tag{3.22}$$

where π_k is the probability of k-th Gaussian that can be calculated as the portion of total periodogram vectors in the k-th Gaussian. If we assume that $P(\omega)$ is a periodogram

selected among a periodogram dataset with M periodograms, each periodogram of this dataset is as $P_m(\omega)\,, m = 1, \dots, M$ must have:

$$L(P_m(\omega)\,, m = 1, \dots, M) = \prod_{m=1}^{M} f\big(P_m(\omega)\big) \tag{3.23}$$

where L is the Likelihood of the GMM with K components and can be assumed as a criterion of how good the GMM models of the periodogram dataset are. We also can calculate another criterion like L which is called Log-Likelihood as in (3.24).

$$L_{log}(P_m(\omega)\,, m = 1, \dots, M) = \sum_{m=1}^{M} \log\Big(f\big(P_m(\omega)\big)\Big) \tag{3.24}$$

Since we are dealing with large periodogram datasets, L may result in large values which most of the time are assumed as infinity in computer programming languages, hence in our calculations we use L_{log}. The larger L_{log} represents the better GMM for modelling the database which means the larger K value (number of GMM components).

To create such a model for speech signals and different noise types, we divide each signal into some overlapping short time frames, calculate their periodograms, normalize their power to 1 and in this way we make large periodogram datasets for speech signal and different noise types. To check the possibility of classifying speech and different noise periodograms, we calculated the Histogram of these periodogram datasets. Since the periodogram vectors that we are dealing with are of size of Ω elements (in our experiments $\Omega = 257$), we calculated these Histograms based on the distances of these periodograms from the origin and are shown in Figure 3.6.

It can be seen from Figure 3.6 that the Histograms of the periodogram distances can be shown as the sum of some Gaussians and hence using GMM in this case seems logical. To create a proper GMM, we apply the Estimate Maximization (EM) algorithm on the

periodogram datasets. EM is an iterative algorithm in which after setting the desired number of Gaussians, the probability π_k, mean vector μ_k and covariance matrix Σ_k for all K Gaussians would be calculated.

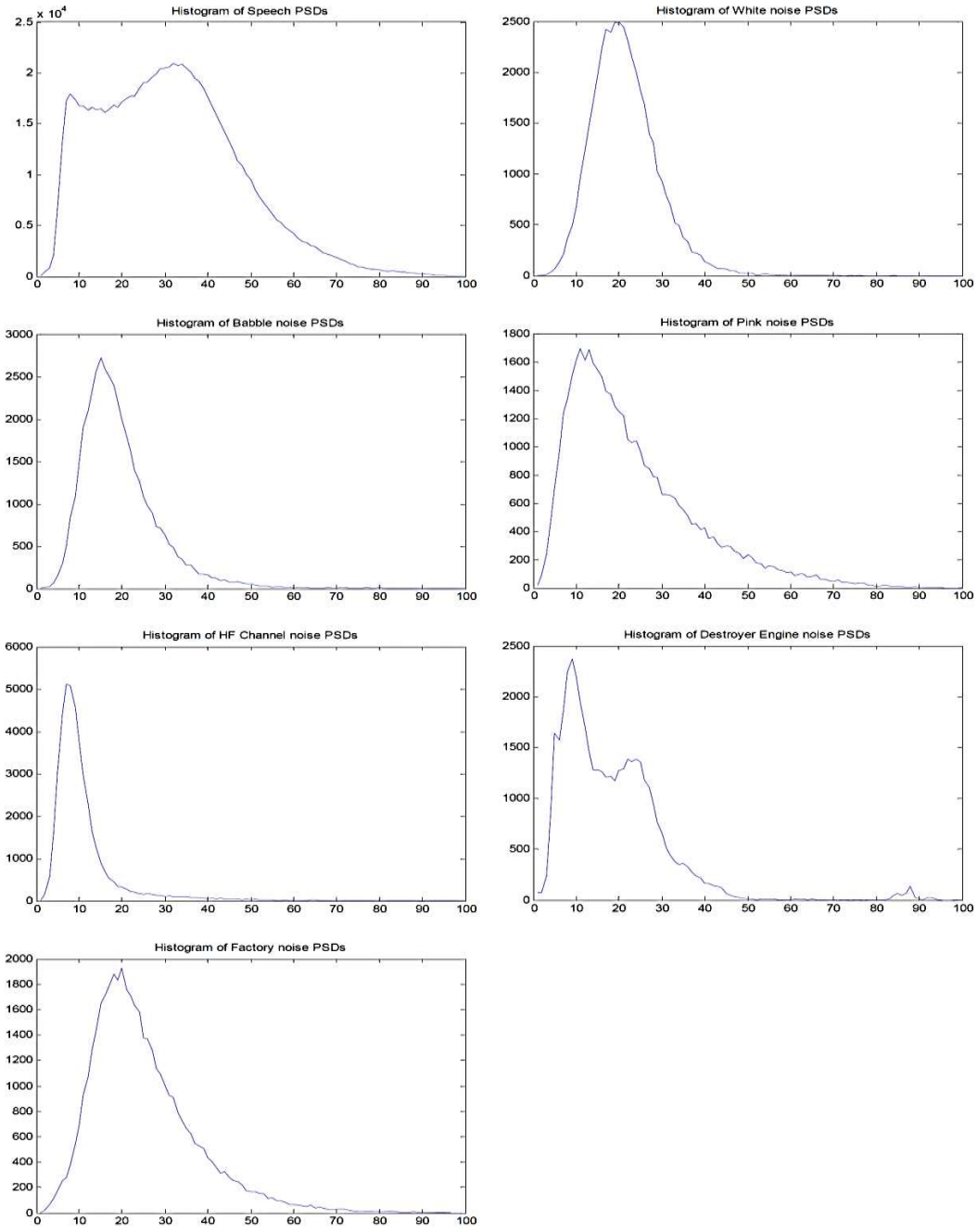

Figure 3.6: Histogram of the distance of normalized periodograms of speech and different noises from the origin

As mentioned previously, the larger K (number of GMM components) will result in larger Log-Likelihood which means more accurate GMM but as the number of GMM components increases we will deal with more complex calculations which will increase the speech enhancement processing time. As can be seen in Figure 3.7, the circles represent the GMM clusters and could be increased till each member of this space represents one GMM component which will not be logical since we are using the GMM to decrease the number of components we are dealing with. In this figure GMMs with 5 and 6 components seem reasonable for modeling of this sample space but we need to find a theoretical solution to be implemented in different spaces with different dimensions and much more number of elements.

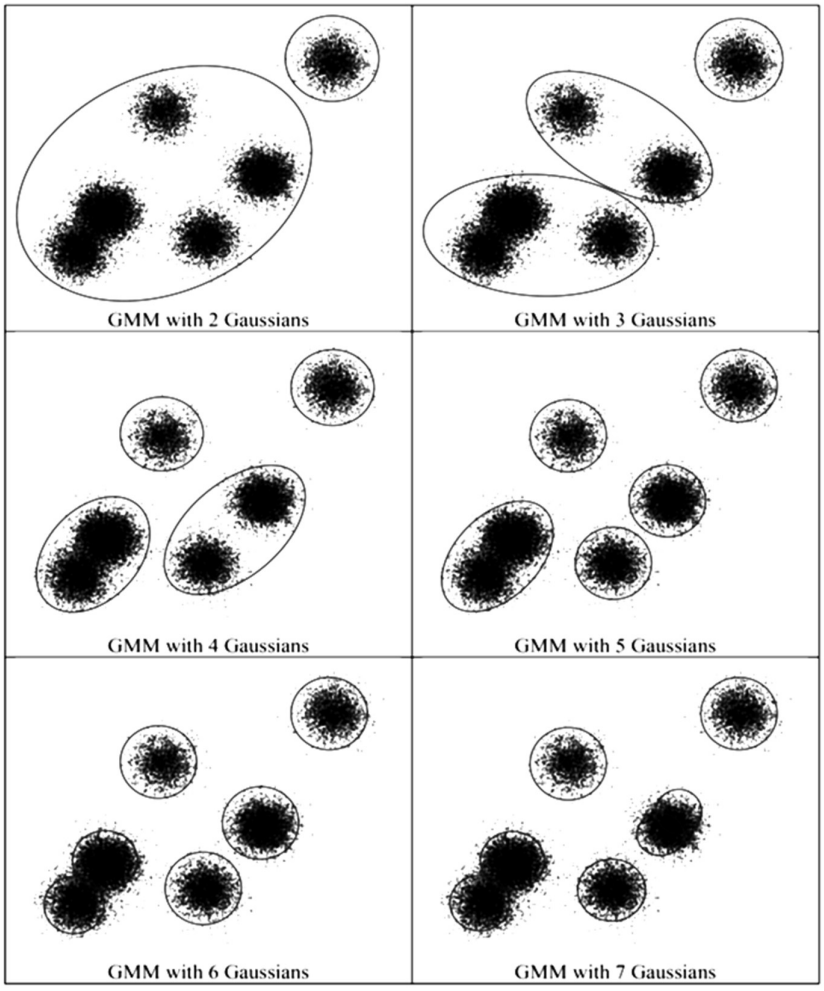

Figure 3.7: Different number of GMMs to model a space

As discussed in (3.21) and (3.22) we are dealing with a covariance matrix Σ_k to calculate the PDF. The covariance matrix contains variances as its diagonal and covariances of different frequencies as its off-diagonal elements. Working with covariance matrixes will increase the complexity of PDF calculations while we are already dealing with complicated equations. Since off-diagonal elements are quite small compared to diagonal elements, we can neglect them (assume them equal to zero) and in this way change the covariance matrixes to variance vectors which are the diagonals of covariance matrixes so the notation Σ_k for covariance matrix is replaced with σ_k for variance vector (orthogonal assumption for periodograms). In this way (3.21) and (3.22) are rewritten for speech periodograms as in (3.25).

$$f_{\boldsymbol{S}}(\boldsymbol{P}_s) = \sum_{i=1}^{K_s} \pi_{s_i} G_{s_i}(\boldsymbol{P}_s; \boldsymbol{\mu}_{s_i}, \boldsymbol{\sigma}_{s_i}) \quad , \quad \sum_{i=1}^{K_s} \pi_{s_i} = 1$$

$$G_{s_i}(\boldsymbol{P}_s; \boldsymbol{\mu}_{s_i}, \boldsymbol{\sigma}_{s_i}) = \frac{1}{(2\pi)^{\frac{d}{2}} \left|\boldsymbol{\sigma}_{s_i}\right|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\sum_{\omega=1}^{\Omega} \frac{\left(P_{s_\omega} - \mu_{s_{i,\omega}}\right)^2}{\sigma_{s_{i,\omega}}}\right\} \tag{3.25}$$

where $\boldsymbol{P}_s = P_s(\omega)$, $\boldsymbol{\mu}_{s_i} = \mu_{s_i}(\omega)$, $\boldsymbol{\sigma}_{s_i} = \sigma_{s_i}(\omega)$. Also P_{s_ω}, $\mu_{s_{i,\omega}}$ and $\sigma_{s_{i,\omega}}$ represent the ω-th component of $\boldsymbol{P}_s$, $\boldsymbol{\mu}_{s_i}$ and $\boldsymbol{\sigma}_{s_i}$ respectively. $\left|\boldsymbol{\sigma}_{s_i}\right|$ is the multiplication of all elements of $\boldsymbol{\sigma}_{s_i}$. In the same way for the noise periodogram we will have:

$$f_{\boldsymbol{N}}(\boldsymbol{P}_n) = \sum_{j=1}^{K_n} \pi_{n_j} G_{n_j}\left(\boldsymbol{P}_n; \boldsymbol{\mu}_{n_j}, \boldsymbol{\sigma}_{n_j}\right) \quad , \quad \sum_{j=1}^{K_n} \pi_{n_j} = 1$$

$$G_{n_k}\left(\boldsymbol{P}_n; \boldsymbol{\mu}_{n_j}, \boldsymbol{\sigma}_{n_j}\right) = \frac{1}{(2\pi)^{\frac{d}{2}} \left|\boldsymbol{\sigma}_{n_j}\right|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\sum_{\omega=1}^{\Omega} \frac{\left(P_{n_\omega} - \mu_{n_{j,\omega}}\right)^2}{\sigma_{n_{j,\omega}}}\right\} \tag{3.26}$$

where $\boldsymbol{P}_n = P_n(\omega)$, $\boldsymbol{\mu}_{n_j} = \mu_{n_j}(\omega)$, $\boldsymbol{\sigma}_{n_j} = \sigma_{n_j}(\omega)$. Also P_{n_ω}, $\mu_{n_{j,\omega}}$ and $\sigma_{n_{j,\omega}}$ represent the ω-th component of $\boldsymbol{P}_n$, $\boldsymbol{\mu}_{n_j}$ and $\boldsymbol{\sigma}_{n_j}$ respectively. $\left|\boldsymbol{\sigma}_{n_j}\right|$ is the multiplication of all elements of $\boldsymbol{\sigma}_{n_j}$.

In [62, 63] a number of 6 Gaussians for the speech GMM ($K_s = 6$) and 9 Gaussians for each noise GMMs ($K_n = 9$) are used and these numbers are selected with trial and error. These models can be used with Bayesian speech and noise estimation methods (MMSE and MAP) that will be discussed later.

3.4.1 Periodogram estimation by solving over-determined equations on the GMMs

In [63, 64] these GMMs of speech and different noise types were treated like full search codebooks that discussed in section 3.3.1. A set of over-determined equations were solved on the mean vectors of these GMMs to find the proper centroids and their power coefficients that closely build the noisy speech periodogram.

In [63] the GMMs are treated exactly like full search method in which at each time a set of overdetermined equations will be solved between one centroid of speech and one centroid of one noise type and at the end the power coefficients and centroids corresponding to the minimum MSE will be selected for periodogram estimation.

In [64], all the centroids of speech GMM and all centroids of one noise type GMM will be put together to solve the overdetermined equations. In this case we will have coefficients for each centroid of the GMMs. The estimated periodograms of speech and noise will be as:

$$P_s(\omega) = \sum_{i=1}^{K_s} a_{s_i}\mu_{s_i}(\omega) \qquad , \qquad P_n(\omega) = \sum_{j=1}^{K_n} a_{n_j}\mu_{n_j}(\omega) \tag{3.27}$$

where a_{s_i} and a_{n_j} are the power coefficients of the i-th and j-th centroids of speech and noise GMMs, respectively and are the result of solving over-determined equations on all the GMM centroids of speech and one noise type.

Using these centroids and their power coefficients a Wiener filter like (3.20) will be constructed for the enhancement of the noisy frame. The difference in this method versus full search method is that here we had 6 centroids for speech and 9 centroids for each noise type while in full search codebook we had like a hundred centroid for each codebook. In this case the search process using GMMs will be considerably faster than full search codebooks and the experiments exhibited better enhancement results [63, 64].

3.4.2 MMSE periodogram estimation using GMM

By using the MMSE Bayesian estimation discussed in section 2.1.4.2 and by rewriting (2.35) for speech and noisy speech periodograms we will have:

$$\boldsymbol{P}_s^{MMSE} = \int_{\boldsymbol{P}_s} \boldsymbol{P}_s f_{\boldsymbol{S}|\boldsymbol{X}}(\boldsymbol{P}_s|\boldsymbol{P}_x) d\boldsymbol{P}_s \tag{3.28}$$

Using (2.36) we can rewrite (3.28) as:

$$\boldsymbol{P}_s^{MMSE} = \frac{\int_0^\infty \boldsymbol{P}_s f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{P}_x|\boldsymbol{P}_s) f_{\boldsymbol{S}}(\boldsymbol{P}_s) d\boldsymbol{P}_s}{\int_0^\infty f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{P}_x|\boldsymbol{P}_s) f_{\boldsymbol{S}}(\boldsymbol{P}_s) d\boldsymbol{P}_s} \tag{3.29}$$

As mentioned in [65] by substituting (3.25) and (3.26) in (3.29) and using the Gamma distribution, a MMSE estimate of speech periodogram is calculated as in (3.30).

$$\boldsymbol{P}_s^{MMSE} = \frac{\sum_{i=1}^{K_s} \sum_{j=1}^{K_n} \frac{\pi_{s_i}}{\sqrt{\boldsymbol{\sigma}_{s_i}}} \frac{\pi_{n_j}}{\sqrt{\boldsymbol{\sigma}_{n_j}}} I_2(b_{i,j}, c_{i,j}, d_{i,j})}{\sum_{i=1}^{K_s} \sum_{j=1}^{K_n} \frac{\pi_{s_i}}{\sqrt{\boldsymbol{\sigma}_{s_i}}} \frac{\pi_{n_j}}{\sqrt{\boldsymbol{\sigma}_{n_j}}} I_1(b_{i,j}, c_{i,j}, d_{i,j})} \tag{3.30}$$

The elements of this equation are expressed as follows:

$$b_{i,j} = \frac{1}{2}\left(\frac{1}{\boldsymbol{\sigma}_{s_i}} + \frac{1}{\boldsymbol{\sigma}_{n_j}}\right), c_{i,j} = -\left(\frac{\boldsymbol{\mu}_{s_i}}{\boldsymbol{\sigma}_{s_i}} + \frac{\boldsymbol{P}_x - \boldsymbol{\mu}_{n_j}}{\boldsymbol{\sigma}_{n_j}}\right),$$

$$d_{i,j} = \frac{1}{2}\left(\frac{\boldsymbol{\mu}_{s_i}^2}{\boldsymbol{\sigma}_{s_i}} + \frac{\left(\boldsymbol{P}_x - \boldsymbol{\mu}_{n_j}\right)^2}{\boldsymbol{\sigma}_{n_j}}\right)$$

$$z = \frac{c_{i,j}}{\sqrt{2b_{i,j}}}, D_{-1} = \mathrm{e}^{\frac{z^2}{4}}\sqrt{\frac{\pi}{2}}\left\{1 - \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right)\right\}, D_{-2} \tag{3.31}$$

$$= \mathrm{e}^{\frac{z^2}{4}}\sqrt{\frac{\pi}{2}}\left\{\sqrt{\frac{\pi}{2}}\mathrm{e}^{\frac{z^2}{4}} - z\left[1 - \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right)\right]\right\}$$

$$I_v = \int_0^{+\infty} P_s^{v-1}\mathrm{e}^{\left(-b_{i,j}P_s^2 - c_{i,j}P_s - d_{i,j}\right)} d\,P_s = \mathrm{e}^{-d_{i,j}}\left(2b_{i,j}\right)^{-\frac{v}{2}}\Gamma(v)\mathrm{e}^{\frac{z^2}{4}}D_{-v}(z)$$

Since the models are all normalized (the power of used periodograms is normalized to one), they must be scaled according to the input SNR before being used. The power biasing is discussed in [66].

There are 6 different noise source candidates whose models must be used for the estimation of $\boldsymbol{P}_s^{MMSE}$ and $\boldsymbol{P}_n^{MMSE}$ for each noisy speech frame. To alleviate the computation we use the method described in [63], using speech and noise mean vectors to model the space of the noisy speech periodogram, as a pre-processing step to find the suitable noise model i.e. the best noise candidate. Therefore, the following processing is limited to a single noise source. To calculate $\boldsymbol{P}_n^{MMSE}$ in (3.31) we can replace $\boldsymbol{\sigma}_{s_i}$ with $\boldsymbol{\sigma}_{n_j}$ and $\boldsymbol{\mu}_{s_i}$ with $\boldsymbol{\mu}_{n_j}$ and vice versa. By the use of estimated $\boldsymbol{P}_s^{MMSE}$ and $\boldsymbol{P}_n^{MMSE}$ and (3.2) we can construct a Wiener filter to enhance the noisy frame. Some other methods for realization of MMSE estimation of speech and noise periodograms are also discussed in [64] and [62]. Such MMSE method exhibited better enhancement results with respect the full search codebooks.

3.4.3 MAP periodogram estimation using GMM

By rewriting (2.43) for a speech periodogram we will have:

$$\begin{aligned} \boldsymbol{P}_s^{MAP} &= \arg\max_{\boldsymbol{P}_s} f_{\boldsymbol{S}|\boldsymbol{X}}(\boldsymbol{P}_s|\boldsymbol{P}_x) \\ &= \arg\max_{\boldsymbol{P}_s} \left(f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{P}_x|\boldsymbol{P}_s) f_{\boldsymbol{S}}(\boldsymbol{P}_s) \right) \end{aligned} \tag{3.32}$$

The last part of (3.32) is like the maximization of the denominator of (3.29). Since (3.30) is the representation of (3.29) using GMMs, the maximization of the denominator of (3.29) is like the maximization of the denominator of (3.30). As mentioned in [66], such a maximization process can be done with some assumptions and hence the MAP estimate of speech periodogram can be written as:

$$\boldsymbol{P}_s^{MAP} = \frac{-\sum_{i=1}^{K_s} \sum_{j=1}^{K_n} c_{i,j}}{2 \sum_{i=1}^{K_s} \sum_{j=1}^{K_n} b_{i,j}} \tag{3.33}$$

where the values for $b_{i,j}$ and $c_{i,j}$ are the ones mentioned in (3.31). To calculate $\boldsymbol{P}_n^{MAP}$ in (3.33) we can replace $\boldsymbol{\sigma}_{s_i}$ with $\boldsymbol{\sigma}_{n_j}$ and $\boldsymbol{\mu}_{s_i}$ with $\boldsymbol{\mu}_{n_j}$ and vice versa. By the use of estimated $\boldsymbol{P}_s^{MAP}$ and $\boldsymbol{P}_n^{MAP}$ and (3.2) we can construct a Wiener filter to enhance the noisy frame. Such a MAP estimation method, exhibited very good enhancement performance with respect to the MMSE method [67].

3.5 Summary

Some model-based speech enhancement methods were introduced in this chapter. Two different models as full search codebooks and also GMMs and also different estimation methods such as MMSE, MAP and over-determined equation solving were discussed. Some improvements on the modelling, estimation and filtering procedure is going to be introduced in the next chapter.

4 Proposed algorithms for improvement of speech enhancement

Some well-known speech enhancement algorithms were discussed in chapter 3. These algorithms were based on Wiener filtering due to its high performance and low residual noise. To make it even more accurate, the algorithms use some models of speech and noise features. These features were the shape and the distribution of speech and noise periodograms. All these methods were compared based on their performance and here we are going to propose methods to overcome their shortcomings to have the highest possible improvement to corrupted noisy speech.

4.1 Finding the reasonable size of the GMMs

One of the important issues that we must deal with is the best size of GMMs (the number of Gaussians in each GMM) since in previous methods which discussed in section 3.4 the size of GMMs were found by trial and error. In this way, we should find the best number of GMM components that is not too small to decrease the accuracy of modeling and is not too large to increase the processing time. In our experiments, we are dealing with clean speech files among the TIMIT database and different noise types such as Babble, Babble, Pink, HF Channel, destroyer Engine and Factory noises. The TIMIT database is made up of 4620 speech files spoken by different speakers and we divided them to some overlapping frames and for each frame calculated the periodogram and put them all together as a periodogram dataset. Also for different noise types we have one large file for each recorded noise and we make the periodogram

dataset by the same procedure as for the clean speech dataset. To find the reasonable number of Gaussians in the GMMs we use the Bayesian Information Criterion (BIC) as discussed in [68, 69]. We used the *gmdistribution.fit* command in MATLAB to apply the EM algorithm on the periodogram datasets. The outputs of this command are the probability, mean vectors and covariance matrices of the GMM centroids and also it calculates the BIC. Each periodogram dataset is fed to the EM algorithm to make the GMM from $K = 2$ to $K = 26$. As discussed in [69], the number of mixtures that represent the maximum BIC, is the best number of mixtures. Since the BIC values for speech and different noise types have different ranges and we just want to find their maximums, to be able to show them on the same figure, we mapped all these different values to the range of 0 to 1. The actual BIC values are negative and hence we take the minimum value as 0 and the maximum value as 1 and every other value in between will be mapped with the same ratio. The mapped values of BIC for speech and different noise types with respect to the number of mixtures are shown in Figure 4.1.

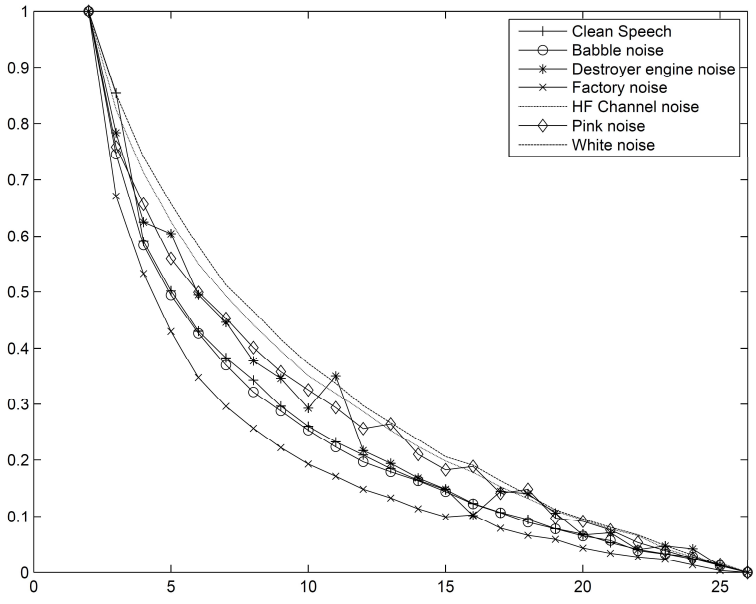

Figure 4.1: BIC (vertical axis) versus number of GMM mixtures (horizontal axis) for clean speech and different noise types

In a similar manner as like the BIC criterion, we calculated Log-Likelihood as L_{log} from (3.24) for each GMM with the number of mixtures changing from 2 to 26. Again

to have all the Log-Likelihoods together despite their different ranges for speech and different noise types, we mapped their range of variation between 0 to 1. These Log-Likelihoods with respect to the number of mixtures for speech and different noise types are shown in Figure 4.2.

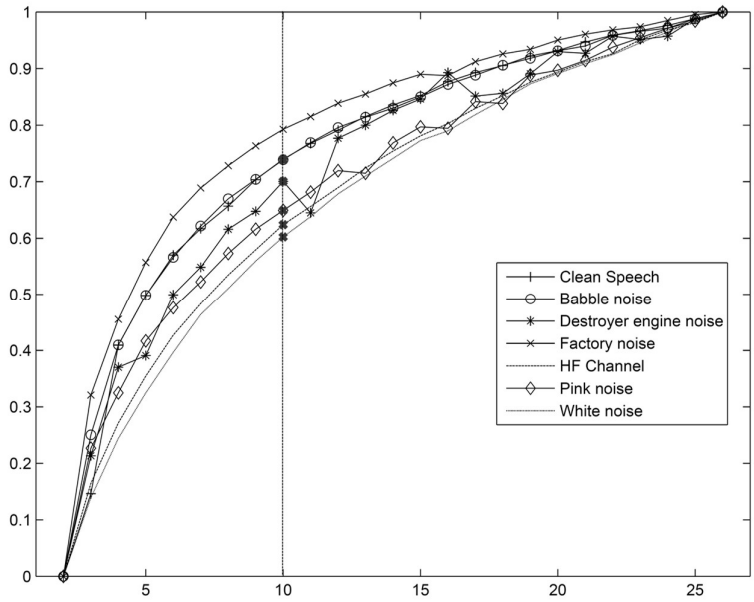

Figure 4.2: The Log-Likelihood of different noise types and clean speech (vertical axis) with respect to the number of GMM mixtures (horizontal axis)

As discussed in [70] and as can be seen in Figure 4.1 and Figure 4.2, for small numbers of mixtures, the curves are quite steep and with increasing number of mixtures the slopes are reduced. In all the plots at around 10 mixtures the slopes decrease and hence we take 10 as a reasonable number of mixtures for our experiments. This is later verified by simulation experiments. This number does not look too small to decrease the accuracy and does not look too big to make calculations complicated. In this way we created GMMs with 10 mixtures for speech and different noise types and used it for the enhancement procedure. As discussed in [70], this is not a deterministic method for finding the number of mixtures in GMMs but a way to select the number of mixtures with a degree of rigor. In this way for all the introduced speech enhancement methods

that are based on GMM, we use 10 Gaussians in them and hence 10 mean vectors in each GMM.

4.2 Explicit MAP using Optimization algorithms

To create the GMMs of speech and noise periodograms, we are dealing with large datasets of periodograms which could be assumed as I vectors of $\Omega + 1$ components for speech and J vectors of $\Omega + 1$ components for noise which are large numbers (I and J can be as large as a couple of hundred thousand to a few million). We are dealing with a space of normalized speech and noise periodograms according to:

$$\begin{aligned} \bar{\bar{\boldsymbol{S}}} &= \left\{ Q_s^i(\omega) = \frac{P_s^i(\omega)}{\bar{P}_s^i}, i = 1, \dots, I \right\} \\ \bar{\bar{\boldsymbol{N}}} &= \left\{ Q_n^j(\omega) = \frac{P_n^j(\omega)}{\bar{P}_n^j}, j = 1, \dots, J \right\} \end{aligned} \tag{4.1}$$

where $\bar{\bar{\boldsymbol{S}}}$ and $\bar{\bar{\boldsymbol{N}}}$ are the space of normalized speech and normalized noise, respectively. $\bar{P}_s^i$ and $\bar{P}_n^j$ are the power of i-th speech periodogram and j-th noise periodogram found as:

$$\begin{aligned} \bar{P}_s^i &= \sum_{\omega=0}^{\Omega} P_s^i(\omega) \\ \bar{P}_n^j &= \sum_{\omega=0}^{\Omega} P_n^j(\omega) \end{aligned} \tag{4.2}$$

We assume that all the noisy speech periodogram observations which can be considered as the space of noisy speech periodograms, can be created as the sum of the periodograms in $\bar{\bar{\boldsymbol{S}}}$ and $\bar{\bar{\boldsymbol{N}}}$ spaces with proper biasing as:

$$\bar{\bar{\boldsymbol{X}}} = \{P_x^h(\omega) = cQ_s^i(\omega) + dQ_n^j(\omega)\ , h = 1, \dots, I{\times}J\} \tag{4.3}$$

where c and d are the biasing that compensate the power since $Q_s^i(\omega)$ and $Q_n^j(\omega)$ have normalized powers. We are going to find a MAP estimate of speech and noise periodograms through the maximization of the posterior PDF of the speech periodogram. Applying the Bayes rule from (2.26) we have:

$$\underbrace{\mathrm{f}_{\bar{\bar{\mathbf{S}}}|\bar{\bar{\mathbf{X}}}}(\boldsymbol{P}_s|\boldsymbol{P}_x)}_{\text{Posterior}} = \frac{1}{\mathrm{f}_{\bar{\bar{\mathbf{X}}}}(\boldsymbol{P}_x)} \underbrace{\mathrm{f}_{\bar{\bar{\mathbf{X}}}|\bar{\bar{\mathbf{S}}}}(\boldsymbol{P}_x|\boldsymbol{P}_s)}_{\text{Likelihood}} \underbrace{\mathrm{f}_{\bar{\bar{\mathbf{S}}}}(\boldsymbol{P}_s)}_{\text{Prior}} \tag{4.4}$$

where f is the PDF as discussed in (3.22) and for the bold vectors we have $\boldsymbol{P}_s = P_s(\omega)$ and $\boldsymbol{P}_x = P_x(\omega)$ which are written in this form for the sake of simplicity of the equation. Assuming that the space of speech periodogram $\bar{\bar{\boldsymbol{S}}}$ and noise periodogram $\bar{\bar{\boldsymbol{N}}}$ are statistically independent, we can replace the likelihood term of $f_{\bar{\bar{\boldsymbol{X}}}|\bar{\bar{\boldsymbol{S}}}}(\boldsymbol{P}_x|\boldsymbol{P}_s)$ with $f_{\bar{\bar{\boldsymbol{N}}}}(\boldsymbol{P}_x - \boldsymbol{P}_s)$ as discussed in section 5.6.2 of [61]. We also know that $\frac{1}{f_{\bar{\bar{\boldsymbol{X}}}}(\boldsymbol{P}_x)}$ is a constant and in this way the MAP estimates of speech and noise periodograms are as:

$$\begin{aligned} \boldsymbol{P}_s^{MAP} &= \arg\max_{\boldsymbol{P}_s} \left[f_{\bar{\bar{\mathbf{N}}}}(\boldsymbol{P}_x - \boldsymbol{P}_s) f_{\bar{\bar{\mathbf{S}}}}(\boldsymbol{P}_s) \right] \\ \boldsymbol{P}_n^{MAP} &= \arg\max_{\boldsymbol{P}_n} \left[f_{\bar{\bar{\mathbf{S}}}}(\boldsymbol{P}_x - \boldsymbol{P}_n) f_{\bar{\bar{\mathbf{N}}}}(\boldsymbol{P}_n) \right] \end{aligned} \tag{4.5}$$

where $f_{\bar{\bar{\mathbf{S}}}}(\boldsymbol{P}_s)$ and $f_{\bar{\bar{\mathbf{N}}}}(\boldsymbol{P}_n)$ are calculated by substituting $\frac{\boldsymbol{P}_s}{\bar{P}_s}$ and $\frac{\boldsymbol{P}_n}{\bar{P}_n}$ in (3.25) and (3.26), respectively. In (4.5) as discussed in [61], $f_s(\boldsymbol{P}_s)$ and $f_n(\boldsymbol{P}_n)$ are assumed as the prior probabilities of $\boldsymbol{P}_s$ and $\boldsymbol{P}_n$ respectively and $f_{\bar{\bar{\mathbf{N}}}}(\boldsymbol{P}_x - \boldsymbol{P}_s)$ and $f_{\bar{\bar{\mathbf{S}}}}(\boldsymbol{P}_x - \boldsymbol{P}_n)$ as the corresponding likelihoods. Using (3.1), we can rewrite these likelihoods as $f_{\bar{\bar{N}}}(\boldsymbol{P}_n)$ and $f_{\bar{\bar{S}}}(\boldsymbol{P}_s)$ respectively. By merging the two equations in (4.5) we can write:

$$\boldsymbol{P}_s^{MAP}, \boldsymbol{P}_n^{MAP} = \arg\max_{\boldsymbol{P}_s, \boldsymbol{P}_n} [f_s(\boldsymbol{P}_s) f_n(\boldsymbol{P}_n)] \tag{4.6}$$

This equation is so complicated since the sum of $K = 10$ exponentials (due to the use of GMMs with 10 mean vectors) for the speech PDF multiplied by the sum of $K = 10$ exponentials for the noise PDF which are totally 100 exponential terms and since there is no explicit solution for it, we use optimization algorithm to solve it and find the value of $\boldsymbol{P}_s^{MAP}$ and $\boldsymbol{P}_n^{MAP}$ in a numerical basis. We have different classes of the optimization algorithms which are derivative based algorithms and genetic based algorithms. In genetic based algorithms there is no consideration on the optimum path to get to the final answer. In the derivative based algorithms, the best path could be found by finding the derivative of the cost function or by trial and error. In MATLAB there are a variety of optimization algorithms and one of them is *fmincon* command which is a derivative based algorithm to minimize the cost function. In this command we can enter the derivative equation of the cost function to increase the accuracy of the algorithm. This algorithm can be set to perform within some constraints and these constraints can be fed to it with some simple equations. Using this algorithm for MAP estimation can become too time-consuming due to dealing with relatively large periodogram vectors. This algorithm will input an initial estimate of the speech and noise periodograms and then change them within the constraints in an iterative manner to decrease the cost function from one iteration to the other. These iterations will continue while these changes that are made to the periodograms, decrease the value of the cost function and do not violate the predefined constraints. In here, the decrement of the cost function from one iteration to another is actually the increment of the PDF multiplication in (4.6). The required initial estimates of the speech and noise periodograms for this algorithm can be found using Minimum Statistic method discussed in 3.2. In the implementation of this algorithm the Minimum Statistics method of [71] was used in which an estimate of the noise periodogram will be resulted and also by decreasing it from the noisy speech periodogram and estimate of speech periodogram will be attained. The power of these

initial estimates will be then normalized by dividing them by the sum of all their components. The distance between the normalized noise periodogram and the mean vectors of different noise type GMMs will be calculated and the GMM and the noise GMM that has the closest mean vector to the normalized noise periodogram will be taken as the noise GMM of the whole noisy file. Such an optimization algorithm is illustrated in Figure 4.3.

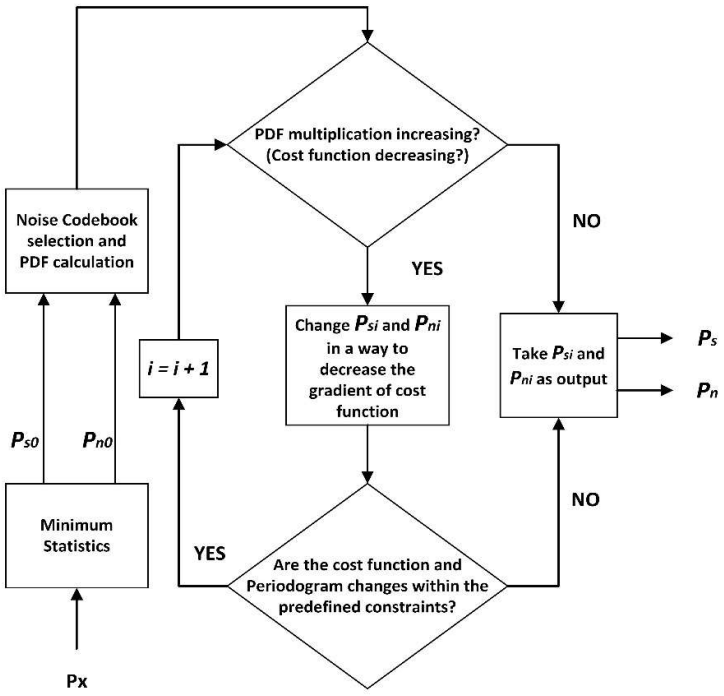

Figure 4.3: Optimization algorithm procedure

In this figure the i index represents the i-th iteration and hence $\boldsymbol{P}_{s\,i}$ and $\boldsymbol{P}_{n\,i}$ are the periodograms of speech and noise estimated in the i-th iteration. Using the final output periodograms of $\boldsymbol{P}_s$ and $\boldsymbol{P}_n$ which are the MAP estimates of speech and noise periodograms from the noisy periodogram, we can construct a Wiener filter to enhance the analyzing noisy frame. Different parts of this optimization procedure to attain the MAP estimations of speech and noise periodograms are explained in details as follows.

4.2.1 **Cost function**

As discussed earlier, in the cost function the PDFs of the speech and noise periodograms are calculated. In the cost function, since we want to maximize $f_{\boldsymbol{S}}(\boldsymbol{P}_s)f_{\boldsymbol{N}}(\boldsymbol{P}_n)$ and the *fmincon* command in MATLAB tries to find the minimum of the cost function when we put the value of $\frac{1}{f_{\boldsymbol{S}}(\boldsymbol{P}_s)f_{\boldsymbol{N}}(\boldsymbol{P}_n)}$ in the cost function because its minimization is equivalent to the maximization of the $f_{\boldsymbol{S}}(\boldsymbol{P}_s)f_{\boldsymbol{N}}(\boldsymbol{P}_n)$ term. The *fmincon* optimization command can just accept one vector as the input and since we want to estimate both $\boldsymbol{P}_s^{MAP}$ and $\boldsymbol{P}_n^{MAP}$, we can concatenate them as one single vector. By assuming that $\boldsymbol{P}_s = P_s(\omega) = \left[P_{s_0}, \dots, P_{s_\Omega}\right]$ and $\boldsymbol{P}_n = P_n(\omega) = \left[P_{n_0}, \dots, P_{n_\Omega}\right]$ where the number of frequency bins is $\Omega + 1$, we can define $P_{sn}(\omega) = [P_s(\omega), P_n(\omega)] = \left[P_{s_0}, \dots, P_{s_\Omega}, P_{n_0}, \dots, P_{n_\Omega}\right]$ and rewrite (4.4) as:

$$\boldsymbol{P}_{sn}^{MAP} = [\boldsymbol{P}_s^{MAP}, \boldsymbol{P}_n^{MAP}] = \arg\max_{\boldsymbol{P}_{sn}} [f_s(\boldsymbol{P}_s)f_n(\boldsymbol{P}_n)] \tag{4.7}$$

where $\boldsymbol{P}_{sn}^{MAP}$ is a vector made up of the two vectors $\boldsymbol{P}_s^{MAP}$ and $\boldsymbol{P}_n^{MAP}$ of length $2(\Omega + 1)$.

4.2.2 **Constraints**

We need to apply some constraints to the *fmincon* algorithm to make sure it does not violate some considerations about the properties of periodograms or their relationships. These constraints can be shown as in (4.8).

$$\arg\min_{\boldsymbol{P}_{sn}} \left[\frac{1}{f_{\boldsymbol{S}}(\boldsymbol{P}_s)f_{\boldsymbol{N}}(\boldsymbol{P}_n)}\right] \quad \text{such that} \quad \begin{cases} \boldsymbol{A}\boldsymbol{P}_{sn} \leq \boldsymbol{b} \\ \boldsymbol{A}_{eq}\boldsymbol{P}_{sn} = \boldsymbol{b}_{eq} \\ \boldsymbol{lb} \leq \boldsymbol{P}_{sn} \leq \boldsymbol{ub} \end{cases} \tag{4.8}$$

The definition of matrix $\boldsymbol{A}$ is in a way such that its multiplication by $\boldsymbol{P}_{sn}$ results in a vector of smaller than the vector $\boldsymbol{b}$. The matrix $\boldsymbol{A}_{eq}$ definition is in a way that its multiplication by $\boldsymbol{P}_{sn}$ results in a vector equal to the vector $\boldsymbol{b}_{eq}$. The vectors $\boldsymbol{lb}$ and $\boldsymbol{ub}$

are of the same length of $\boldsymbol{P}_{sn}$ and define the lower bound and upper bound of $\boldsymbol{P}_{sn}$. All these matrices and vectors are defined in next paragraphs.

To have good estimates of speech and noise periodograms, we should be able to estimate the power of the speech and noise periodograms in an accurate way. Experiments showed that if we do not apply any limitation on the power of the speech and noise periodograms, the resulted periodograms will be very different from the original ones. To solve this problem we need to set the limitations to force the optimization algorithm to work within tolerable ranges. Since we used Minimum Statistics to find the initial estimates of the periodograms, we can use the power of these initial periodograms as the constraints. If we assume that $P_s^{MS}(\omega)$ and $P_n^{MS}(\omega)$ are the periodograms of speech and noise from the Minimum Statistics algorithm, we can define the power of speech and noise in each frame as (4.9).

$$\begin{aligned} P_s^{MS}(\omega) = \left[P_{s_1}^{MS}, \dots, P_{s_\Omega}^{MS}\right] \rightarrow \bar{P}_s = \sum_{\omega=0}^{\Omega} P_{s_\omega}^{MS} \\ P_n^{MS}(\omega) = \left[P_{n_1}^{MS}, \dots, P_{n_\Omega}^{MS}\right] \rightarrow \bar{P}_n = \sum_{\omega=0}^{\Omega} P_{n_\omega}^{MS} \end{aligned} \tag{4.9}$$

We define the $\boldsymbol{A}$ matrix in a way that forces the power of estimated speech and noise periodograms to be within a little range of tolerance around these estimated powers. If we assume the tolerance as $\pm\alpha$ and the calculated powers of speech and noise periodograms from Minimum statistics as $\bar{P}_s$ and $\bar{P}_n$ we can write the resulted power equations for the estimated speech and noise periodograms as (4.10).

$$\begin{aligned} (1-\alpha)\bar{P}_s \leq \sum_{\omega=0}^{\Omega} P_{s_\omega} \leq (1+\alpha)\bar{P}_s \\ (1-\alpha)\bar{P}_n \leq \sum_{\omega=0}^{\Omega} P_{n_\omega} \leq (1+\alpha)\bar{P}_n \end{aligned} \tag{4.10}$$

In MATLAB we can define the $\boldsymbol{A}$ matrix and $\boldsymbol{b}$ vector as (4.11).

$$\underbrace{\begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 \\ -1 & \dots & -1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 \\ 0 & \dots & 0 & -1 & \dots & -1 \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} P_{s_0} \\ \vdots \\ P_{s_\Omega} \\ P_{n_0} \\ \vdots \\ P_{n_\Omega} \end{bmatrix}}_{\boldsymbol{P}_{sn}} \leq \underbrace{\begin{bmatrix} (1+\alpha)\bar{P}_s \\ -(1-\alpha)\bar{P}_s \\ (1+\alpha)\bar{P}_n \\ -(1-\alpha)\bar{P}_n \end{bmatrix}}_{\boldsymbol{b}} \tag{4.11}$$

where $\boldsymbol{A}$ is a $4\times(\Omega+1)$ matrix, $\boldsymbol{P}_{sn}$ is a $(2\Omega+2)\times1$ vector and $\boldsymbol{b}$ is a 4×1 vector.

There is another important consideration that should be taken account of and it is the sum of estimated speech and noise periodograms which should be always equal to the noisy speech periodogram as $P_x(\omega) = P_s^{MAP}(\omega) + P_n^{MAP}(\omega)$. To force this condition to the algorithm, we define $\boldsymbol{A}_{eq}$ matrix and $\boldsymbol{b}_{eq}$ vector as (4.12).

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & & 0 & 0 & & 1 & & \vdots \\ \vdots & \vdots & & 1 & \vdots & \vdots & & & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}_{eq}} \underbrace{\begin{bmatrix} P_{s_0} \\ \vdots \\ P_{s_\Omega} \\ P_{n_0} \\ \vdots \\ P_{n_\Omega} \end{bmatrix}}_{\boldsymbol{P}_{sn}} = \underbrace{\begin{bmatrix} P_{x_0} \\ \vdots \\ P_{x_\Omega} \end{bmatrix}}_{\boldsymbol{b}_{eq}} \tag{4.12}$$

where $\boldsymbol{A}_{eq}$ is a $(\Omega+1)\times(2\Omega+2)$ matrix and $\boldsymbol{b}_{eq}$ is a $(\Omega+1)\times1$ vector. Another important consideration is that the components of the resulted periodogram estimates cannot become negative and hence we can set $\boldsymbol{lb}$ as a zero vector with the length of $2\Omega+2$ (since the input is $\boldsymbol{P}_{sn}$).

There are some other conditions that can be set for the *fmincon* optimization algorithm which are input tolerance or TolX, output tolerance or TolFun and constraint tolerance or TolCon. TolFun is the minimum accepted difference of the cost function output between two successive iterations and if during the iterations of the *fmincon* the difference of the cost function output value becomes less than TolFun the iterations will stop. By setting TolCon we can determine the least possible change in limitations. By

setting TolX we can determine the least possible change in the input vector. In our experiments the suitable values for these parameters are calculated through trial and error.

In this way, using the *fmincon* algorithm we were able to implement explicit MAP with no approximation. The enhancement results using this algorithm are discussed in section 5. We tried to simplify the cost function in a way to extract its derivative directly and by equating it to zero find the explicit MAP estimate of the periodograms without using an optimization algorithm and this method is the focus of the next section.

4.3 A Simple explicit MAP estimation

As discussed earlier, the multiplication of the two PDFs of speech and noise periodograms is so complicated and hence using optimization algorithm and applying all the constraints can become drastically time consuming. Here we are going to introduce an explicit solution to maximize the multiplication of the PDFs of speech and noise periodograms. To do so, we need to simplify the multiplication and since the PDFs contain positive values, the maximization of the multiplication of these positive PDFs is equivalent to the maximization of the logarithm of their product as in (4.13).

$$\begin{aligned} \boldsymbol{P}_s^{MAP}, \boldsymbol{P}_n^{MAP} &= \arg\max_{\boldsymbol{P}_s, \boldsymbol{P}_n} [f_{\boldsymbol{S}}(\boldsymbol{P}_s) f_{\boldsymbol{N}}(\boldsymbol{P}_n)] \\ &\equiv \arg\max_{\boldsymbol{P}_s, \boldsymbol{P}_n} \left[\ln\left(f_{\boldsymbol{S}}(\boldsymbol{P}_s) f_{\boldsymbol{N}}(\boldsymbol{P}_n)\right)\right] \\ &= \arg\max_{\boldsymbol{P}_s, \boldsymbol{P}_n} \underbrace{\left[\ln\left(f_{\boldsymbol{S}}(\boldsymbol{P}_s)\right) + \ln\left(f_{\boldsymbol{N}}(\boldsymbol{P}_n)\right)\right]}_{T} \end{aligned} \tag{4.13}$$

The maximization of the multiplication $f_{\boldsymbol{S}}(\boldsymbol{P}_s) f_{\boldsymbol{N}}(\boldsymbol{P}_n)$ can be replaced with the maximization of the summation $T = \ln\left(f_{\boldsymbol{S}}(\boldsymbol{P}_s)\right) + \ln\left(f_{\boldsymbol{N}}(\boldsymbol{P}_n)\right)$. In (4.13) each PDF is the sum of 10 exponentials as in (3.25) and (3.26) and hence we can consider $f_{\boldsymbol{S}}(\boldsymbol{P}_s) =$

$f_{\boldsymbol{S}_1}(\boldsymbol{P}_s) + \cdots + f_{\boldsymbol{S}_{10}}(\boldsymbol{P}_s)$ and $f_{\boldsymbol{N}}(\boldsymbol{P}_n) = f_{\boldsymbol{N}_1}(\boldsymbol{P}_n) + \cdots + f_{\boldsymbol{N}_{10}}(\boldsymbol{P}_n)$. From Jensen's inequality we know that:

$$\ln\left(f_{\boldsymbol{S}}(\boldsymbol{P}_s)\right) \geq \ln\left(f_{\boldsymbol{S}_1}(\boldsymbol{P}_s)\right) + \cdots + \ln\left(f_{\boldsymbol{S}_{10}}(\boldsymbol{P}_s)\right)$$

$$\ln\left(f_{\boldsymbol{N}}(\boldsymbol{P}_n)\right) \geq \ln\left(f_{\boldsymbol{N}_1}(\boldsymbol{P}_n)\right) + \cdots + \ln\left(f_{\boldsymbol{N}_{10}}(\boldsymbol{P}_n)\right) \tag{4.14}$$

$$T \geq \ln\left(f_{\boldsymbol{S}_1}(\boldsymbol{P}_s)\right) + \cdots + \ln\left(f_{\boldsymbol{S}_{10}}(\boldsymbol{P}_s)\right) + \ln\left(f_{\boldsymbol{N}_1}(\boldsymbol{P}_n)\right) + \cdots + \ln\left(f_{\boldsymbol{N}_{10}}(\boldsymbol{P}_n)\right)$$

As discussed in [72], to maximize the T term in (4.13) we can maximize its lower bound as shown in the last equation of (4.14). In the experiments we found out that the maximization of the lower bound of T almost always results in the maximum of the T term and the exceptional cases can be considered as the estimation error. By such an assumption we can say that the maximization of the logarithm of sum of all positive exponentials in the PDF formula of speech or noise can be taken as the maximization of sum of the logarithm of all those values. In this way we can replace T in (4.13) with the right hand side of the last equation in (4.14) and replace all the PDF values from (3.25) and (3.26) and rewrite (4.13) as:

$$\begin{aligned}\boldsymbol{P}_s^{MAP}, \boldsymbol{P}_n^{MAP} &= \arg\max_{\boldsymbol{P}_s, \boldsymbol{P}_n} \left[\sum_{k=1}^{K} \left[\ln\left(\pi_{s_k} G_{s_k}\right) + \ln\left(\pi_{n_k} G_{n_k}\right)\right]\right] \\ &= \arg\max_{\boldsymbol{Q}_s, \boldsymbol{Q}_n} \left[\sum_{k=1}^{K} \left\{ C_k - \frac{1}{2} \sum_{\omega=0}^{\Omega} \left[\frac{\left(\boldsymbol{Q}_s - \boldsymbol{\mu}_{s_k}\right)^2}{\boldsymbol{\sigma}_{s_k}} + \frac{\left(\boldsymbol{Q}_n - \boldsymbol{\mu}_{n_k}\right)^2}{\boldsymbol{\sigma}_{n_k}}\right]\right\}\right]\end{aligned} \tag{4.15}$$

where $\boldsymbol{Q}_s = Q_s(\omega) = P_s(\omega)/\sum_{\omega=0}^{\Omega} P_s(\omega)$ and $\boldsymbol{Q}_n = Q_n(\omega) = P_n(\omega)/\sum_{\omega=0}^{\Omega} P_n(\omega)$ are the normalized speech and noise periodograms, respectively. Also G_{s_k} and G_{n_k} are actually $G_{s_k}\left(\boldsymbol{P}_s; \boldsymbol{\mu}_{s_k}, \boldsymbol{\sigma}_{s_k}\right)$ and $G_{n_k}\left(\boldsymbol{P}_n; \boldsymbol{\mu}_{n_k}, \boldsymbol{\sigma}_{n_k}\right)$. The parameter C_k is a constant as $C_k = \ln\left(\pi_{s_k}\right) - \frac{\Omega+1}{2}\ln(2\pi) - \frac{1}{2}\ln\left(\left|\boldsymbol{\sigma}_{s_k}\right|\right) + \ln\left(\pi_{n_k}\right) - \frac{\Omega+1}{2}\ln(2\pi) - \frac{1}{2}\ln\left(\left|\boldsymbol{\sigma}_{n_k}\right|\right)$. We also know that since we have 10 mean vectors in each GMM, we have $K = 10$. At this stage

to find the values of $\boldsymbol{P_s, P_n}$ that maximize the final extracted function in (4.15), we calculate its first derivatives with respect to $\boldsymbol{Q}_s$ and $\boldsymbol{Q}_n$ and we take them equal to zero and hence equal to each other as:

$$\sum_{k=1}^{K}\sum_{\omega=0}^{\Omega}\frac{\boldsymbol{\mu}_{s_k}-\boldsymbol{Q}_s}{\boldsymbol{\sigma}_{s_k}}=\sum_{k=1}^{K}\sum_{\omega=0}^{\Omega}\frac{\boldsymbol{\mu}_{n_k}-\boldsymbol{Q}_n}{\boldsymbol{\sigma}_{n_k}}=0 \tag{4.16}$$

In our very first consideration about the speech and noise periodograms we assume that $P_s(\omega)+P_n(\omega)=P_x(\omega)$ and since all the vectors are periodograms, their components cannot get any negative values. Therefor, when we want to find $P_n(\omega)=P_x(\omega)-P_s(\omega)$ or $P_s(\omega)=P_x(\omega)-P_n(\omega)$ as used to get from (4.5) to (4.6), we assume that there is no negative components in the resulted periodograms from the subtraction process and in this way each frequency component is treated independently. In this way we can remove the summation over ω from the two sides of (4.16) and assume that the sum of K different values of each frequency component are equal to zeros and hence rewrite (4.16) as:

$$\sum_{k=1}^{K}\frac{\boldsymbol{\mu}_{s_k}-\boldsymbol{Q}_s}{\boldsymbol{\sigma}_{s_k}}=\sum_{k=1}^{K}\frac{\boldsymbol{\mu}_{n_k}-\boldsymbol{Q}_n}{\boldsymbol{\sigma}_{n_k}} \tag{4.17}$$

In the same way as discussed in section 4.2, we use Minimum Statistics method to find the best noise GMM for the noisy file. Also the result of Minimum Statistics can be used as the power estimate for MAP process. To solve (4.17), $\boldsymbol{Q}_s$ and $\boldsymbol{Q}_n$ are replaced with $\boldsymbol{P}_s/\bar{P}_s^{MS}$ and $\boldsymbol{P}_n/\bar{P}_n^{MS}$ where $\bar{P}_s^{MS}$ and $\bar{P}_n^{MS}$ are resulted from Minimum Statistics, and then $\boldsymbol{P}_n$ is replaced with $\boldsymbol{P}_x-\boldsymbol{P}_s$ as:

$$\sum_{k=1}^{K}\frac{\boldsymbol{\mu}_{s_k}-\frac{\boldsymbol{P}_s}{\bar{P}_s^{MS}}}{\boldsymbol{\sigma}_{s_k}}=\sum_{k=1}^{K}\frac{\boldsymbol{\mu}_{n_k}-\frac{\boldsymbol{P}_x-\boldsymbol{P}_s}{\bar{P}_n^{MS}}}{\boldsymbol{\sigma}_{n_k}} \tag{4.18}$$

The vectors $\boldsymbol{P}_s$ and $\boldsymbol{P}_x$ can be brought out of the summations as:

$$\sum_{k=1}^{K}\frac{\boldsymbol{\mu}_{s_k}}{\boldsymbol{\sigma}_{s_k}}-\frac{\boldsymbol{P}_s}{\bar{P}_s^{MS}}\sum_{k=1}^{K}\frac{1}{\boldsymbol{\sigma}_{s_k}}=\sum_{k=1}^{K}\frac{\boldsymbol{\mu}_{n_k}}{\boldsymbol{\sigma}_{n_k}}-\frac{\boldsymbol{P}_x-\boldsymbol{P}_s}{\bar{P}_n^{MS}}\sum_{k=1}^{K}\frac{\boldsymbol{\mu}_{n_k}}{\boldsymbol{\sigma}_{n_k}} \tag{4.19}$$

By simplifying (4.19) we can calculate the MAP estimate of the speech periodogram as:

$$P_s^{MAP}(\omega)=\frac{\sum_{k=1}^{K}\left(\frac{\mu_{s_k}(\omega)}{\sigma_{s_k}(\omega)}-\frac{\mu_{n_k}(\omega)}{\sigma_{n_k}(\omega)}\right)+\frac{P_x(\omega)}{\bar{P}_n^{MS}}\sum_{k=1}^{K}\frac{1}{\sigma_{n_k}(\omega)}}{\frac{1}{\bar{P}_s^{MS}}\sum_{k=1}^{K}\frac{1}{\sigma_{s_k}(\omega)}+\frac{1}{\bar{P}_n^{MS}}\sum_{k=1}^{K}\frac{1}{\sigma_{n_k}(\omega)}} \tag{4.20}$$

We can estimate the MAP estimate if the noise periodogram $\boldsymbol{P}_n^{MAP}$ through $\boldsymbol{P}_n^{MAP}=\boldsymbol{P}_x-\boldsymbol{P}_s^{MAP}$ and zero any resulting negative elements. Using these periodograms in (3.2) we can enhance the current noisy speech frame. The enhancement results of this method is discussed in section 5 in Figure 5.1.

4.4 Improved explicit MAP estimation

In section 4.3 a method for the MAP estimation of speech and noise periodograms from a noisy observation using speech and noise GMMs was introduced which is discussed in [73]. This MAP estimation is simplified to the maximization of the multiplication of the PDFs of the speech and noise periodograms. This PDF multiplication can be really complicated to maximize and hence we introduced some new variables as $\boldsymbol{Q}_s=Q_s(\omega)$ and $\boldsymbol{Q}_n=Q_n(\omega)$ to replace the $P_s(\omega)/\sum_{\omega=0}^{\Omega}P_s(\omega)$ and $P_n(\omega)/\sum_{\omega=0}^{\Omega}P_n(\omega)$ in (4.15). In this way, the derivative of the multiplication of the two PDFs once with respect to $\boldsymbol{Q}_s$ and then with respect to $\boldsymbol{Q}_n$ is calculated and both taken equal to zero. After this the values of $\boldsymbol{Q}_s$ and $\boldsymbol{Q}_n$ were again replaced with $P_s(\omega)/\sum_{\omega=0}^{\Omega}P_s(\omega)$ and $P_n(\omega)/\sum_{\omega=0}^{\Omega}P_n(\omega)$ this time the noise periodogram $P_n(\omega)$ is replaced with $P_x(\omega)-P_s(\omega)$ and from these two equations the value of $P_s(\omega)$ could be calculated as (4.20). Mathematically, the use of $\boldsymbol{Q}_s$ and $\boldsymbol{Q}_n$ and using them as independent variables, can

affect the performance of the MAP estimation algorithm. To prevent this effect, the maximization of the multiplication of the two PDFs as in (4.15) can be written as:

$$\boldsymbol{P}_s^{MAP}, \boldsymbol{P}_n^{MAP} = \arg\max_{\boldsymbol{P}_s, \boldsymbol{P}_n}[f_s(\boldsymbol{P}_s) f_n(\boldsymbol{P}_n)] = \arg\max_{\boldsymbol{P}_s, \boldsymbol{P}_n}\left[\ln\left(f_s(\boldsymbol{P}_s) f_n(\boldsymbol{P}_n)\right)\right]$$

$$= \arg\max_{\boldsymbol{P}_s}\left[\sum_{k=1}^{K}\left\{C_k - \frac{1}{2}\sum_{\omega=0}^{\Omega}\left[\frac{\left(\frac{\boldsymbol{P}_s}{\bar{P}_s} - \boldsymbol{\mu}_{s_k}\right)^2}{\boldsymbol{\sigma}_{s_k}} + \frac{\left(\frac{\boldsymbol{P}_x - \boldsymbol{P}_s}{\bar{P}_n} - \boldsymbol{\mu}_{n_k}\right)^2}{\boldsymbol{\sigma}_{n_k}}\right]\right\}\right] \quad (4.21)$$

Now in this equation we find the first derivative with respect to $P_s(\omega)$ and by equating it to zero we can come up with a more accurate MAP estimate of clean speech periodogram. The only variable in this equation is $P_s(\omega)$ and all the other parameters are constants and we assume that all the frequency bins are independent [73], hence we have:

$$P_s^{MAP}(\omega) = \frac{\frac{P_x(\omega)}{\bar{P}_n^2}\sum_{k=1}^{K}\left(\frac{1}{\sigma_{n_k}(\omega)}\right) + \frac{1}{\bar{P}_s}\sum_{k=1}^{K}\left(\frac{\mu_{s_k}(\omega)}{\sigma_{s_k}(\omega)}\right) - \frac{1}{\bar{P}_n}\sum_{k=1}^{K}\left(\frac{\mu_{n_k}(\omega)}{\sigma_{n_k}(\omega)}\right)}{\frac{1}{\bar{P}_s^2}\sum_{k=1}^{K}\left(\frac{1}{\sigma_{s_k}(\omega)}\right) + \frac{1}{\bar{P}_n^2}\sum_{k=1}^{K}\left(\frac{1}{\sigma_{n_k}(\omega)}\right)} \quad (4.22)$$

As mentioned in [73], the values $\bar{P}_s$ and $\bar{P}_n$ can be calculated using the Minimum Statistics method as discussed in [71]. Using this estimated $P_s^{MAP}(\omega)$ we can calculate $P_n^{MAP}(\omega)$ as $P_x(\omega) - P_s^{MAP}(\omega)$ and then construct the proper Wiener filter as mentioned in (3.2) to enhance the noisy file. The enhancement results of this method are discussed in section 5 and illustrated in Figure 5.7 “MAP periodogram”.

The enhanced speech using this improvement in the MAP formula exhibits some residual noise in each frame which could be the result of inaccurate power estimation using the Minimum Statistics method. Rather than periodogram estimation, this time we estimate the amplitude of the speech spectrum knowing that practically the periodogram is the squared value of the amplitude. Since the residual noise which can occur in

different frequency bins in the speech amplitude are normally smaller than 1, squaring the amplitude value to calculate their corresponding periodogram value will decrease the total residual noise. In the frequency domain we have $X(\omega) = S(\omega) + N(\omega)$ where X, S ans N are the spectrum of noisy speech, speech and noise respectively and hence the noise periodogram becomes $N(\omega) = X(\omega) - S(\omega)$. Using this equation, the noise periodogram can be written as:

$$\begin{aligned} P_n(\omega) = N(\omega)N^*(\omega) &= \big(X(\omega) - S(\omega)\big)\big(X(\omega) - S(\omega)\big)^* \\ &= S(\omega)S^*(\omega) + X(\omega)X^*(\omega) - S(\omega)X^*(\omega) - S^*(\omega)X(\omega) \end{aligned} \tag{4.23}$$

Knowing that $P_s(\omega) = S(\omega)S^*(\omega)$ and $P_x(\omega) = X(\omega)X^*(\omega)$, we can rewrite (4.23) as:

$$P_n(\omega) = P_s(\omega) + P_x(\omega) - S(\omega)X^*(\omega) - S^*(\omega)X(\omega) \tag{4.24}$$

Based on (4.24), the minimum value for $P_n(\omega)$ will become:

$$\begin{aligned} P_n^{min}(\omega) &= P_s(\omega) + P_x(\omega) - |S(\omega)X^*(\omega)| - |S^*(\omega)X(\omega)| \\ &= P_s(\omega) + P_x(\omega) - 2|S(\omega)||X(\omega)| \end{aligned} \tag{4.25}$$

This minimum value happens when the phase difference between the spectrums of speech and noisy speech is zero. By defining $A_s(\omega) = \sqrt{P_s(\omega)} = |S(\omega)|$ and $A_x(\omega) = \sqrt{P_x(\omega)} = |X(\omega)|$ in which $A_s(\omega) = \boldsymbol{A}_s$ and $A_x(\omega) = \boldsymbol{A}_x$ represent the spectrum amplitudes of clean speech and noisy speech respectively, the minimum noise periodogram of (4.25) can be re-written as:

$$P_n^{min}(\omega) = A_s^2(\omega) + A_x^2(\omega) - 2A_s(\omega)A_x(\omega) = \big(A_x(\omega) - A_s(\omega)\big)^2 \tag{4.26}$$

This is then replaced with the $P_n(\omega)$ value (or actually $\boldsymbol{P}_x - \boldsymbol{P}_s$ value) in equation (4.21) and the resulting MAP estimation of speech spectrum amplitude will become:

$$\boldsymbol{A}_s^{MAP} = \arg\max_{\boldsymbol{A_s}} \left[\sum_{k=1}^{K} \left\{ C_k - \frac{1}{2} \sum_{\omega=0}^{\Omega} \left[\frac{\left(\frac{\boldsymbol{A}_s^2}{\bar{P}_s} - \boldsymbol{\mu}_{s_k} \right)^2}{\boldsymbol{\sigma}_{s_k}} + \frac{\left(\frac{(\boldsymbol{A}_x - \boldsymbol{A}_s)^2}{\bar{P}_n} - \boldsymbol{\mu}_{n_k} \right)^2}{\boldsymbol{\sigma}_{n_k}} \right] \right\} \right] \quad (4.27)$$

The idea of replacing $P_n(\omega)$ with its minimum is that to make sure that it will at least meet its minimum expected value. Since the noisy speech periodogram is considered as the sum of clean speech and noise periodograms, and if the minimum expected value for the noise periodogram is not met, the remainder would be considered as the speech periodogram which can cause residual noise in the enhanced speech. The next step is to calculate the first derivative of the resulted equation with respect to $A_s(\omega)$ and then making it equal to zero and solving it. Since (4.27) is of order 4 for $A_s(\omega)$ variable, its first derivative with respect to $A_s(\omega)$ becomes of order 3 and in this way equating it to zero will lead to 3 roots. Again here we are going to use the same assumption as [73] where we treat each frequency bin independently and hence ignoring the summation over ω in (4.27). The resulted equation for $A_s(\omega)$ that needs to be solved becomes:

$$a_3 A_s^3(\omega) + a_2 A_s^2(\omega) + a_1 A_s(\omega) + a_0 = 0$$

$$a_3 = \frac{1}{\bar{P}_s^2} \sum_{k=1}^{K} \frac{1}{\sigma_{s_k}(\omega)} + \frac{1}{\bar{P}_n^2} \sum_{k=1}^{K} \frac{1}{\sigma_{n_k}(\omega)}$$

$$a_2 = \frac{-3A_x(\omega)}{\bar{P}_n^2} \sum_{k=1}^{K} \frac{1}{\sigma_{n_k}(\omega)} \quad (4.28)$$

$$a_1 = \frac{3A_x^2(\omega)}{\bar{P}_n^2} \sum_{k=1}^{K} \frac{1}{\sigma_{n_k}(\omega)} - \frac{1}{\bar{P}_s} \sum_{k=1}^{K} \frac{\mu_{s_k}(\omega)}{\sigma_{s_k}(\omega)} - \frac{1}{\bar{P}_n} \sum_{k=1}^{K} \frac{\mu_{n_k}(\omega)}{\sigma_{n_k}(\omega)}$$

$$a_0 = \frac{A_x(\omega)}{\bar{P}_n} \sum_{k=1}^{K} \frac{\mu_{n_k}(\omega)}{\sigma_{n_k}(\omega)} - \frac{A_x^3(\omega)}{\bar{P}_n} \sum_{k=1}^{K} \frac{1}{\sigma_{n_k}(\omega)}$$

Since each amplitude vector contains $\Omega + 1$ frequency bins, we need to deal with $\Omega + 1$ sets of 3 roots and among these sets we used the minimum of the real value of the 3 roots. In this way we can make sure that the least possible residual noise will exist in the enhanced speech. The enhancement results of this method are discussed in section 5 and illustrated in Figure 5.7 as "MAP amplitude".

4.5 Power estimation using Gamma modelling

Minimum Statistics (MS) is a well-known power estimation method in which the minima of smoothed noisy speech periodogram in some successive frames is taken as the periodogram of noise [25, 71, 74]. The method discussed in [71] is used for speech and noise power estimation in the MAP estimation method discussed in 4.3. There is also another power estimation method called Unbiased MMSE discussed in [75] in which a MMSE criterion is used on a GMM model of speech and noise periodograms to estimate noise power. In this section we are going to improve power estimation for the sake of MAP estimation algorithm improvement.

4.5.1 Gamma model of power distributions

The MS power estimation method is online and this means that the power at each frame can be estimated based on the current frame and some information collected from previous frames. Sometimes there is no need for online enhancement like a recorded noisy speech and etc. and hence the whole noisy speech signal or at least a large portion of it is available and in this way the power estimation can become offline. The power of periodograms can be calculated as

$$\bar{P}_x = \sum_{\omega} P_x(\omega) \ , \bar{P}_s = \sum_{\omega} P_s(\omega) \ , \bar{P}_n = \sum_{\omega} P_n(\omega) \tag{4.29}$$

in which $\bar{P}_x$, $\bar{P}_s$ and $\bar{P}_n$ are the power of noisy speech, speech and noise respectively. If we do the summation over ω on the two sides of (3.1) and from the definition given in (4.29) we can write the relationship between these powers as:

$$\bar{P}_x = \bar{P}_s + \bar{P}_n \tag{4.30}$$

In the same way discussed in section 3.4, some large files from speech and different noise types divided into some overlapping frames and for all these frames the periodogram calculated. For those periodogram datasets, the power (sum of all components) of each periodogram is calculated and hence we will have large datasets of powers for speech and different noise types. Our experiments were done on Clean Speech and White, Babble, Pink, Destroyer Engine, HF channel and Factory noises and hence for each signal type the power dataset is created. For each signal type (speech and different noise types), the range of the power dataset (the difference between maximum and minimum values) is divided into some classes. For each class, the number of elements that fall into that class is counted and this number can represent the histogram. To be able to compare all these histograms, each one is divided to the sum of all its values and in this way the histogram will be changed to the probability distribution (in which sum of all its values will be equal to 1) which is the Probability Density Function (PDF) in practice. These PDFs are shown in Figure 4.4. In This figure the number of power classes is taken as 100.

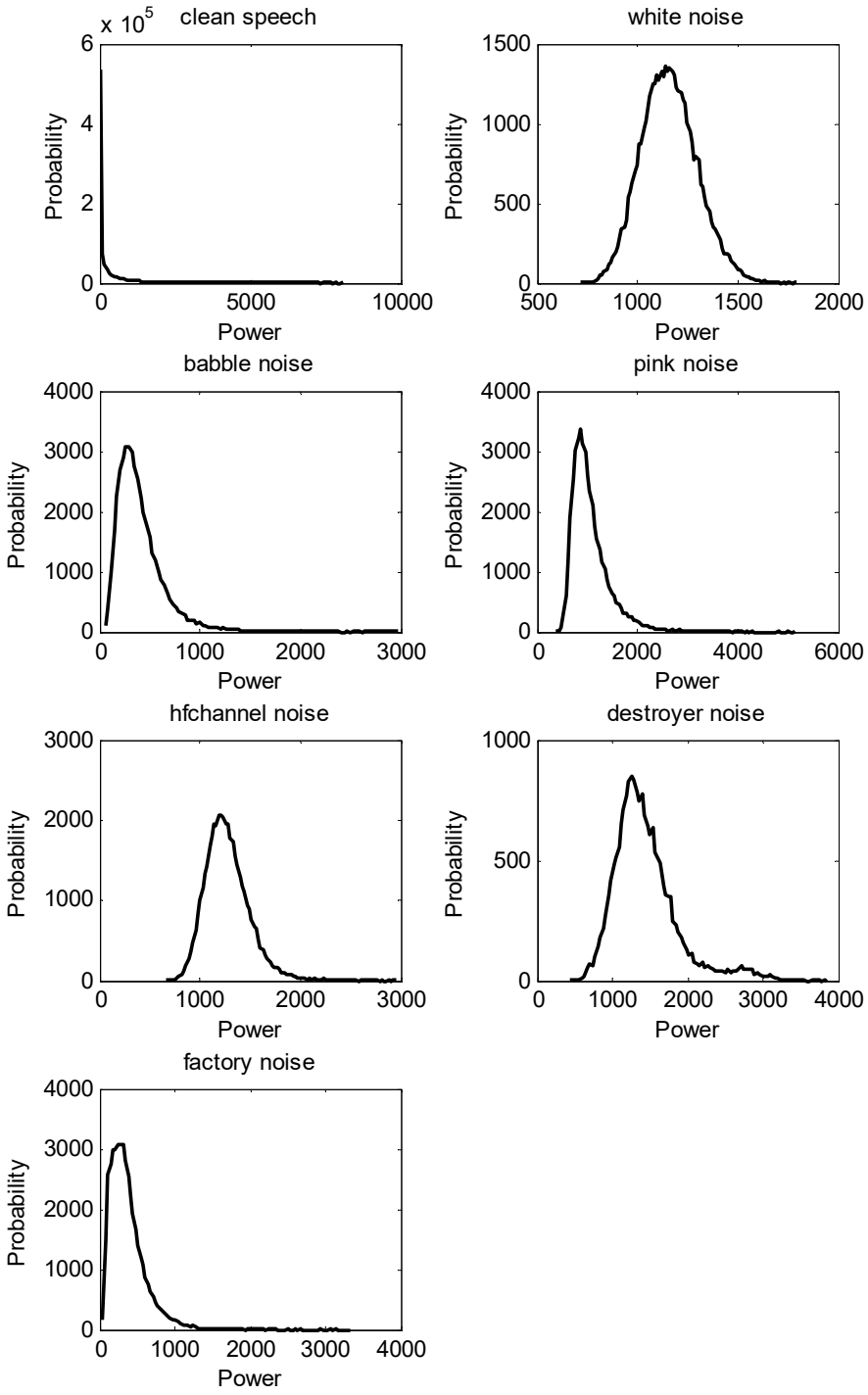

Figure 4.4: PDF of clean speech and different noise types periodogram power (power is considered as the sum of all frequency components of the periodograms)

The shapes of PDFs shown in Figure 4.4 are quite similar to the shapes created using variations of Gamma distribution as in Figure 4.5.

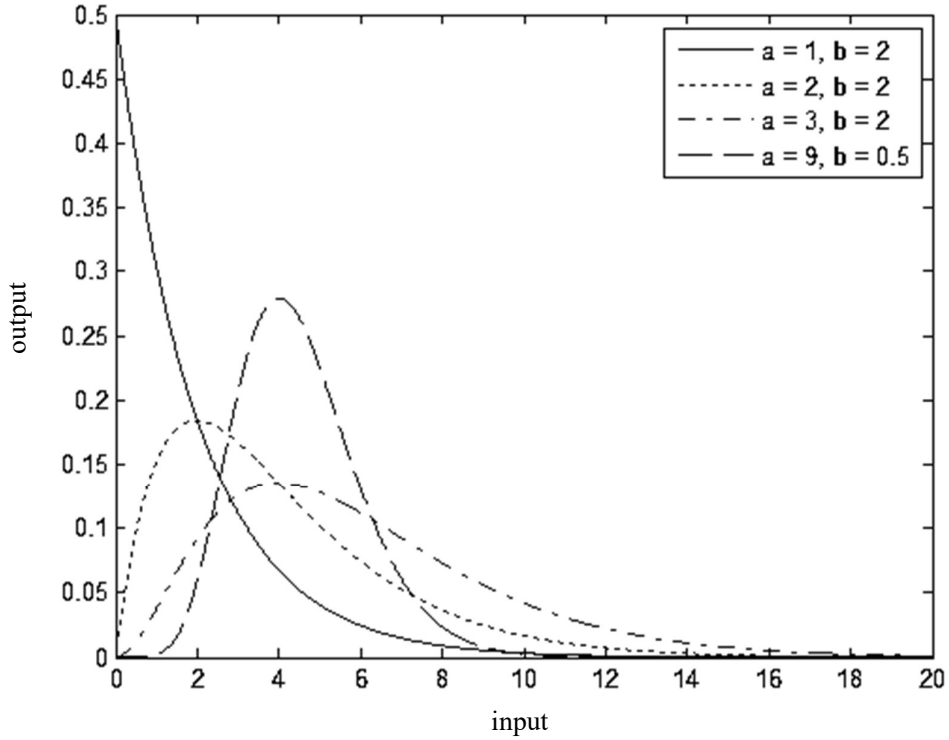

Figure 4.5: Variations of Gamma distribution using shape and rate parameters

The variations of Gamma distribution which are shown in Figure 4.5 are based on the two parameters of a and b which are called shape and rate parameters respectively. If $\bar{P}$ is taken as the variable of the horizontal axis as power, the values on the vertical axis as PDF can be calculated using the parameters of the Gamma distribution as (4.31).

$$f(\bar{P}) = \frac{\bar{P}^{a-1} e^{-\frac{\bar{P}}{b}}}{b^a \Gamma(a)} \tag{4.31}$$

where f represents the PDF of the corresponding power of $\bar{P}$ with the shape and rate parameters of a and b respectively. To fit a distribution on these PDFs which is equivalent to finding the proper parameters of the Gamma distribution, the Maximum Likelihood Estimator (MLE) is used [76]. In this way, variations of Gamma

distributions, different a and b parameters, are found and then the closest one to the original PDF is considered as the desired distribution. Using the *fitdist* command of MATLAB we can fit a Gamma distribution on these periodogram power PDFs. This command will give the two parameters of a and b that can generate the proper Gamma distribution that can fit the PDF. All the speech and different noise type power PDFs and their fitted Gamma distribution are shown in Figure 4.6. In This figure the number of power classes is taken as 100.

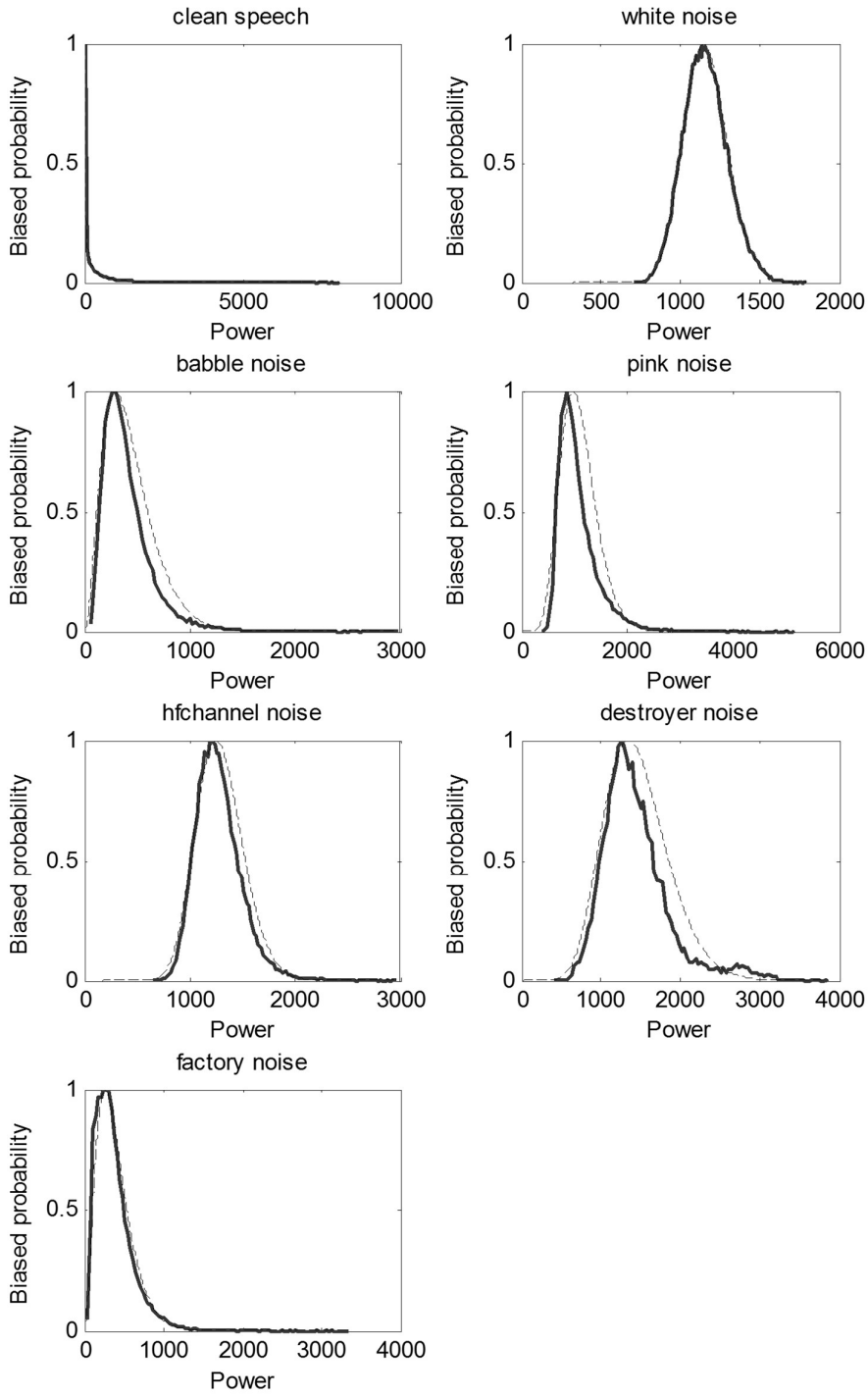

Figure 4.6: The power PDFs of speech and different noise types and the fitted Gamma distribution on them

The resulted a and b parameters for speech and different noises after applying *fitdist* on the power PDFs of their training files, for the distributions shown in Figure 4.6 are exhibited in Table 4.1.

Table 4.1: Shape and Rate parameters of the power distributions of large speech and noise PSD datasets

Signal	Shape parameter a	Rate parameter b
Speech	0.232	1201.285
White noise	69.516	16.735
Babble noise	3.164	136.370
Pink noise	10.055	108.118
HF Channel noise	32.699	39.357
Destroyer Engine noise	12.948	112.743
Factory noise	3.093	125.519

The shape and rate parameters reported in Table 4.1 are calculated through the training set of data for speech and different noise types. To test the accuracy of these shape parameters, 50 test speech files were taken and mixed with all 6 noise types with -5, 0, 5 and 10 dB input SNR and for each noisy speech file, the PDF of speech and noise periodogram power were calculated. For each PDF the corresponding shape parameters were calculated using MATLAB *fitdist*. All these shape parameters then averaged on each signal type, either speech or different noises. These averaged shape parameters on the test data and their corresponding value from the training data are shown in the following table:

Table 4.2: Averaged shape parameters of speech and noise from 50 noisy speech files for each noise

Signal	Shape parameter a from training data	Averaged shape parameter a on test data
Speech	0.232	0.229
White noise	69.516	74.005
Babble noise	3.164	5.072
Pink noise	10.055	11.020
HF Channel noise	32.699	48.457
Destroyer Engine noise	12.948	23.731
Factory noise	3.093	6.195

Viewed from Table 4.2, the averaged values are quite close to the real values. In this way the shape parameter values which are calculated from the large periodogram power datasets of the train files are be used as the shape parameters of speech and noise in the noisy speech with for input SNR. The modelling procedure is illustrated in Figure 4.7.

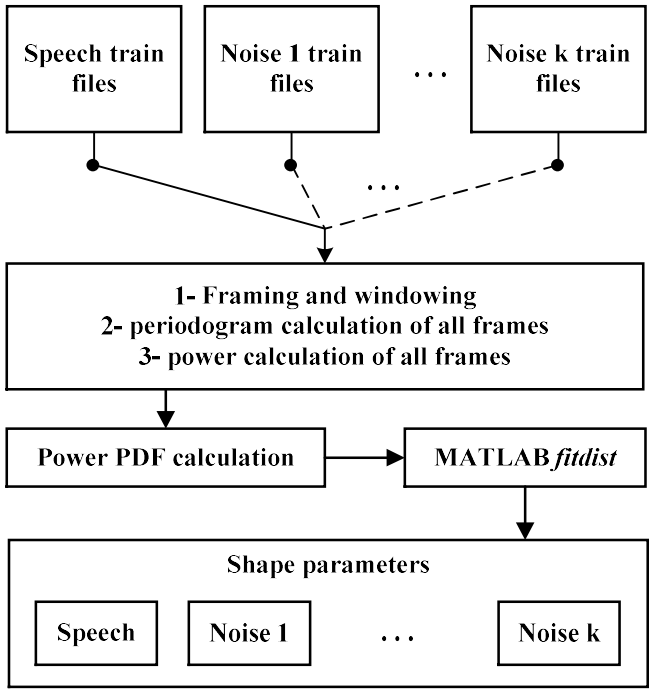

Figure 4.7: Fitting Gamma distribution on the power periodograms of speech and noise and extracting their shape parameters

One of the properties of the Gamma distribution is that the mean of the distribution can be calculated by multiplication of the shape and rate parameters as ab. Since it is considered that the noisy speech is the sum of the clean speech and noise, after doing some practical experiments and observations, it turned out that it can be assumed empirically that the mean of noisy speech power distribution is roughly equal to the sum of the mean of speech and noise power distributions.

$$a_x b_x \cong a_s b_s + a_n b_n \tag{4.32}$$

where the x, s and n indexes represent noisy speech, clean speech and noise and a and b are the corresponding shape and rate parameters of their Gamma distribution. In this way after finding the parameters of the distributions of the power PDF of the observed noisy speech (a_x and b_x) and also replacing a_s and a_n from their corresponding models created from the large power PDF datasets shown in Table 4.2, there is a relationship between b_s and b_n. The next step is finding a way to estimate the proper values for these two rate parameters to be able to get to the power distribution of speech and noise in a noisy speech observation. It is considered that for different noisy speech files, the shapes of the power PDF of the consisting speech and noise is the same as the ones in Table 4.1, and their different rates will cause different shapes and rates for the distribution of the power PDF of the noisy speech. Hence, the power PDF of some sample noisy speech files of different noises with different input SNRs were drawn as shown in Figure 4.8 to find a relationship between the SNR and the shape of the power PDF of the noisy speech.

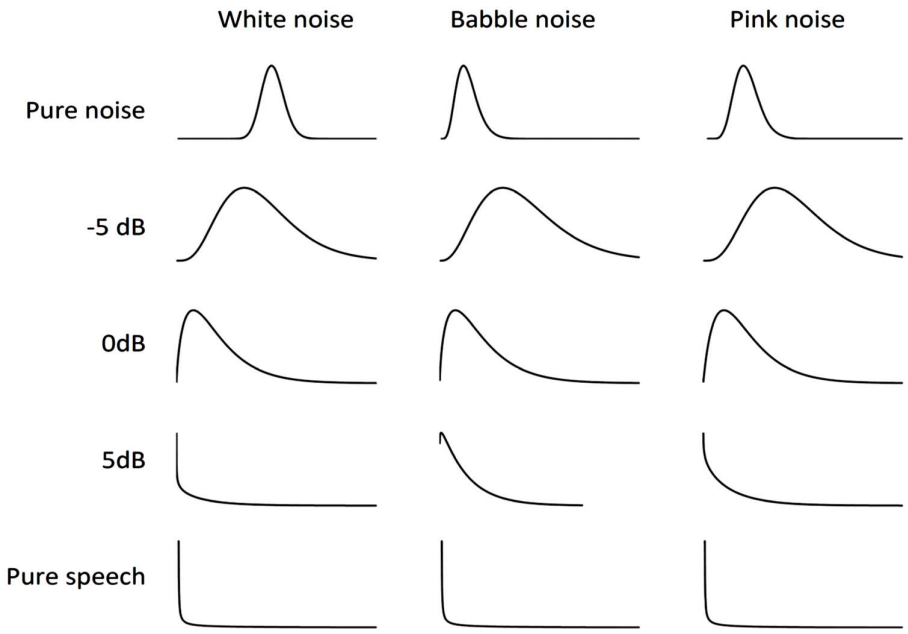

Figure 4.8: The power PDF of 3 different clean speech signals mixed with White, Babble and Pink noises with -5, 0 and 5 dB input SNRs.

As can be seen from Figure 4.8, by increasing the SNR (moving from the first row to the last row), the tendency of the shape of the noisy speech power PDF migrates towards the left and gets close to the power PDF of clean speech. However, the decrement of the SNR results in the less tendency of noisy speech power PDF towards left, and mostly getting close to the noise power PDF. The tendency of the distribution towards the left is actually the asymmetry of the power PDF and in terms of statistic criterions, it can be considered as the Skewness of the distribution [77]. In Gamma distributions, the Skewness can be calculated as $2/\sqrt{a}$ where a is the shape parameter of the distribution [78]. To find the relationship between the SNR and the Skewness, 50 test speech files were used to create noisy speech files for each noise type with -5, 0, 5 and 10 dB input SNRs. In this way a total number of 200 noisy files were created for each noise type. For each noisy speech signal the power PDF and its corresponding Skewness were calculated, and for each noise type the average Skewness versus input SNR was drawn as in Figure 4.9.

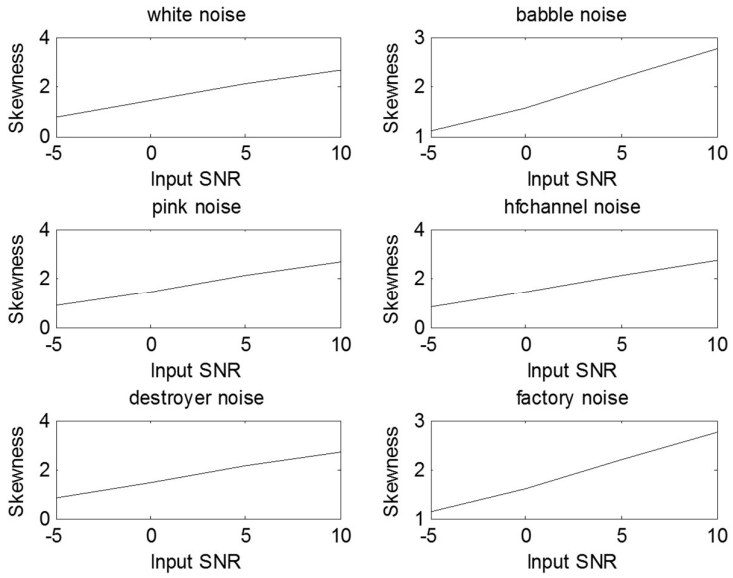

Figure 4.9: Skewness of noisy speech power distribution averaged over 50 noisy speech files

As can be seen in Figure 4.9, the Skewness vs. SNR diagrams are almost linear and almost have the same values. Hence all diagrams of Figure 4.9 were averaged and came up with a final Skewness vs. SNR diagram as in Figure 4.10.

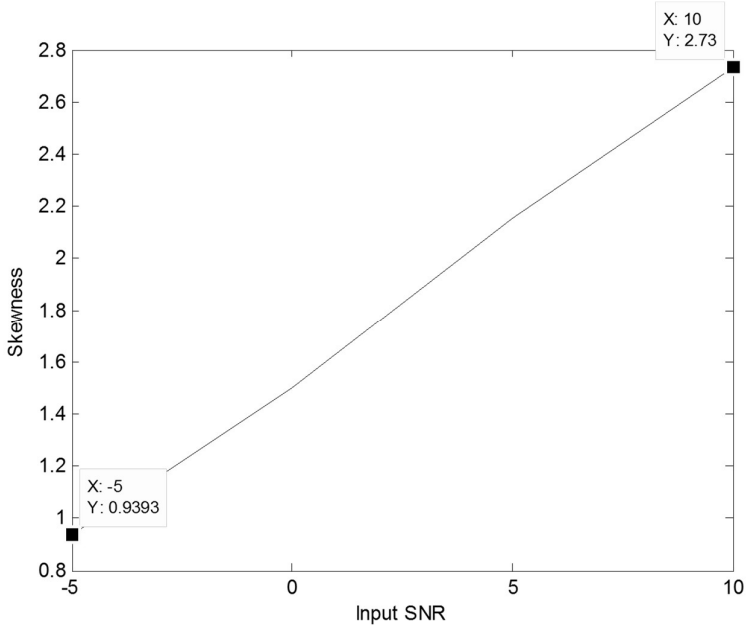

Figure 4.10: Skewness of noisy speech power PDF averaged over 50 noisy speech files and 6 noise types

Using the two points shown in Figure 4.10, a relationship between Skewness and the SNR of the noisy speech can be written as (4.33).

$$SNR = 8.375\lambda - 12.864 \tag{4.33}$$

and by replacing λ with its relationship with the shape parameter (4.33) can be rewritten as:

$$SNR = \frac{16.75}{\sqrt{a_x}} - 12.864 \tag{4.34}$$

So, using (4.34) it will be possible to estimate the input SNR using the calculated a_x from the power distribution of the observed noisy speech. To test the accuracy of this method, for each noise type of Babble, White, Pink, HF channel, Destroyer Engine and Factory we created 3 different noisy speeches with -5, 0 and 5 dB input SNR. The estimated speech and noise powers using this method can be seen in the following figures.

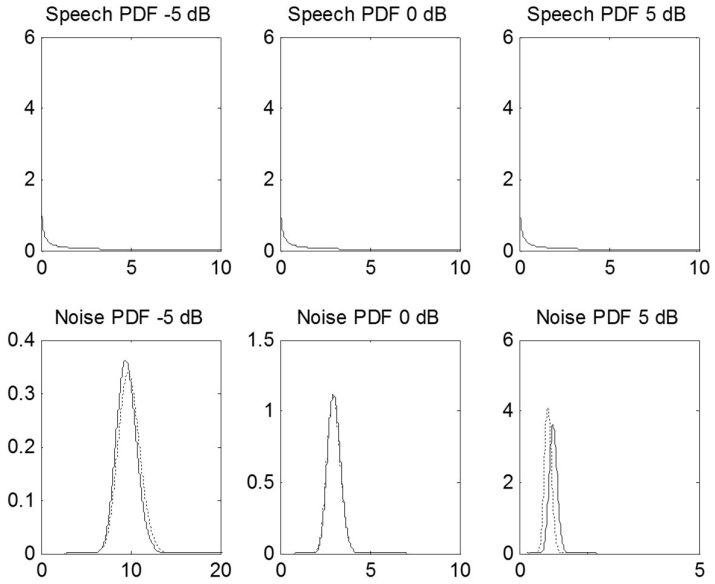

Figure 4.11: The real (solid **line) and estimated (dashed line) speech and White noise power distribution PDF of a sample noisy speech with different input SNRs (vertical and horizontal axis are the power and the input SNR, respectively)**

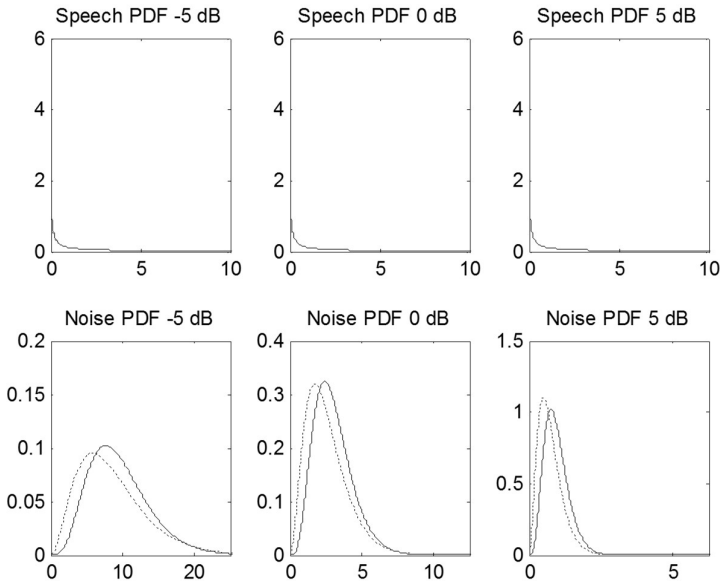

Figure 4.12: The real (solid line) and estimated (dashed line) speech and Babble noise power distribution PDF of a sample noisy speech with different input SNRs (vertical and horizontal axis are the power and the input SNR, respectively)

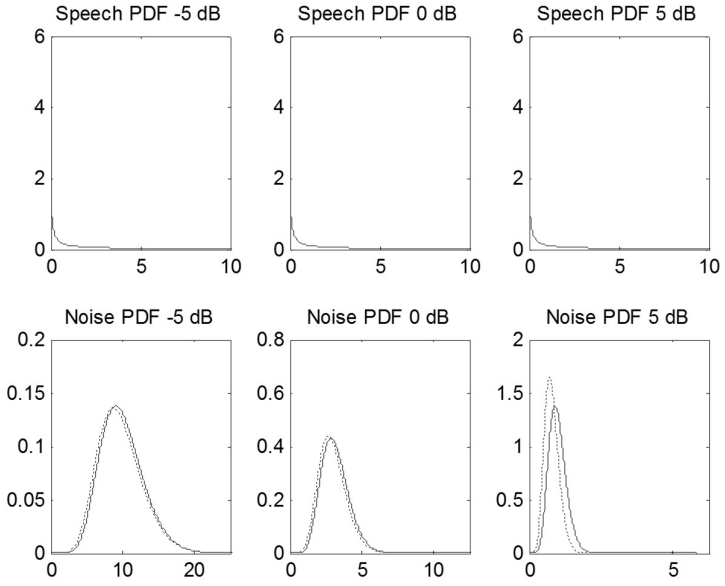

Figure 4.13: The real (solid line) and estimated (dashed line) speech and Pink noise power distribution PDF of a sample noisy speech with different input SNRs (vertical and horizontal axis are the power and the input SNR, respectively)

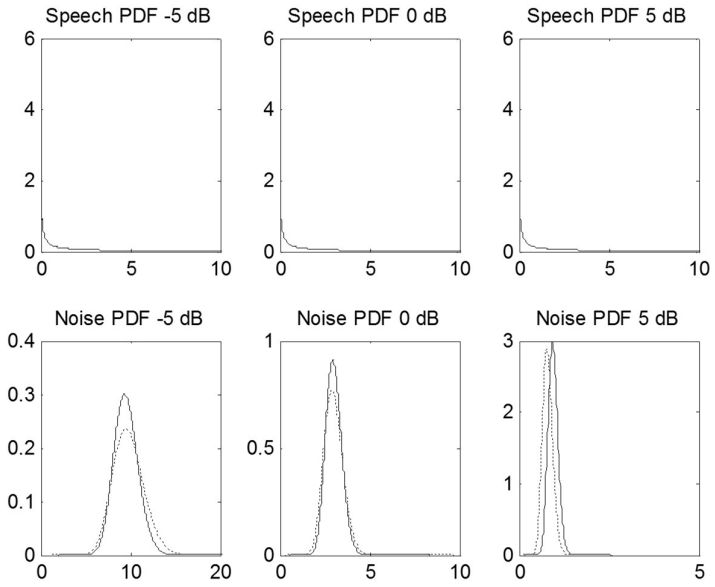

Figure 4.14: The real (solid line) and estimated (dashed line) speech and HF Channel noise power distribution PDF of a sample noisy speech with different input SNRs (vertical and horizontal axis are the power and the input SNR, respectively)

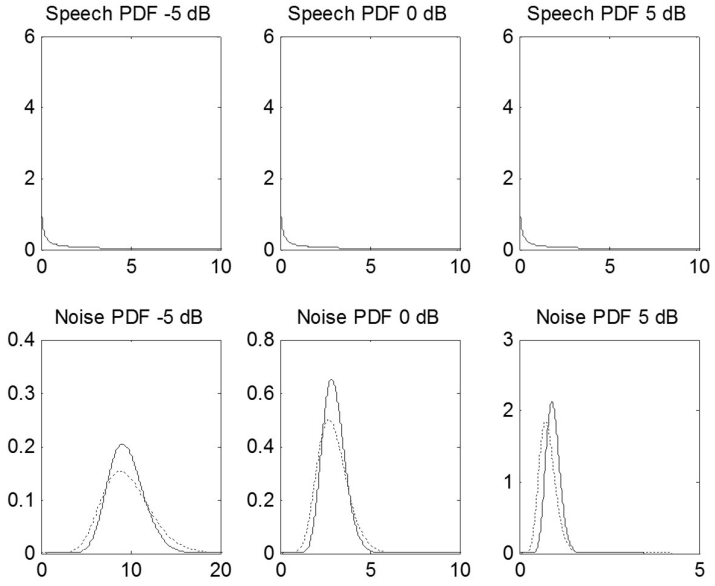

Figure 4.15: The real (solid line) and estimated (dashed line) speech and Destroyer Engine noise power distribution PDF of a sample noisy speech with different input SNRs (vertical and horizontal axis are the power and the input SNR, respectively)

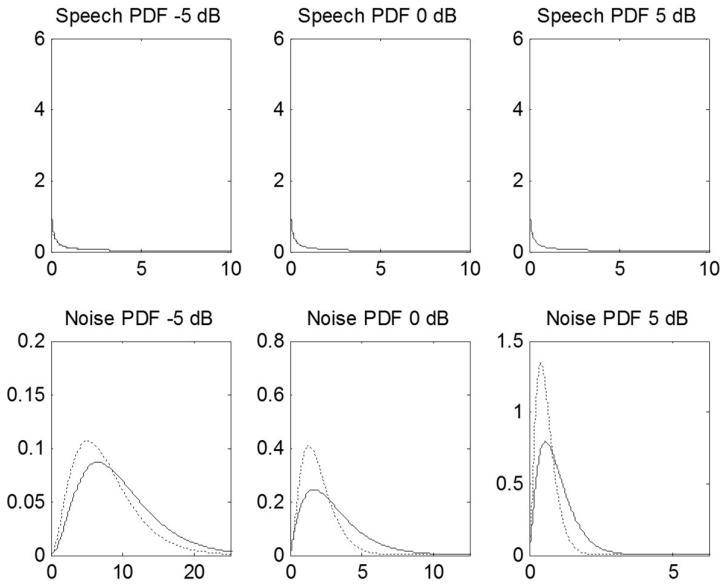

Figure 4.16: The real (solid line) and estimated (dashed line) speech and Factory noise power distribution PDF of a sample noisy speech with different input SNRs (vertical and horizontal axis are the power and the input SNR, respectively)

As can be seen from Figure 4.11 to Figure 4.16 using the mentioned Gamma modelling method the power distributions of the clean speech and different noises with different input SNRs can be estimated with high accuracy.

4.5.2 Power estimation using the MAP criterion

By considering the first frame of the noisy speech as pure noise (silent speech) and comparing its normalized periodogram with all the periodograms in different noise GMMs (each GMM contains 10 normalized periodogram as the mean vectors) as introduced in [73], the GMM containing the closest periodogram to the noisy speech frame could be considered as the proper noise model for the current noisy speech file. For the selected noise model the corresponding shape parameters (a_s and a_n) are available from the Gamma models shown in the "training data" column of Table 4.2. Based on (4.32) we can write the SNR of a frame as:

$$SNR = 10\log\left(\frac{a_s b_s}{a_n b_n}\right) \tag{4.35}$$

where $a_s b_s$ is the power of the speech periodogram and $a_n b_n$ is the power of the noise periodogram and by the assumption of (4.30), we can rewrite (4.35) as:

$$SNR = 10\log\left(\frac{a_s b_s}{a_x b_x - a_s b_s}\right) \tag{4.36}$$

In this equation the SNR is known from (4.34), a_s and a_n are known from the model, a_x and b_x are known from the Gamma distribution of the noisy speech power and the only unknown is b_s which could be calculated as:

$$b_s = \frac{a_x b_x}{a_s\left(1 + 10^{-\frac{SNR}{10}}\right)} \tag{4.37}$$

By replacing the calculated b_s in (4.32) the proper value for b_n could be calculated. Now by having all the shape and rate parameters the power PDF of speech and noise periodograms forming the observed noisy speech periodogram can be calculated. Now the right speech and noise powers can be found using the MAP criterion on these power distributions. The Gamma distributions of speech and noise power ($\bar{P}_s$ and $\bar{P}_n$) using equation (4.31) by replacing y with power can be written as (4.38).

$$\begin{aligned} f_s(\bar{P}_s) &= c_s \bar{P}_s{}^{a_s - 1} e^{-\frac{\bar{P}_s}{b_s}} \quad , \quad c_s = \frac{1}{b_s{}^{a_s}\Gamma(a_s)} \\ f_n(\bar{P}_n) &= c_n \bar{P}_n{}^{a_n - 1} e^{-\frac{\bar{P}_n}{b_n}} \quad , \quad c_n = \frac{1}{b_n{}^{a_n}\Gamma(a_n)} \end{aligned} \tag{4.38}$$

where $\bar{P}_s$ and $\bar{P}_n$ are the powers of speech and noise, and f_s and f_n are the PDFs of speech and noise powers respectively. As discussed in [73], the MAP criterion between the two PDFs is written as (4.39).

$$\bar{P}_s^{MAP}, \bar{P}_n^{MAP} = \arg\max_{\bar{P}_s, \bar{P}_n} [f_s(\bar{P}_s) f_n(\bar{P}_n)] \tag{4.39}$$

Since both PDFs result in positive values, maximization of the multiplication of the two PDFs is like maximization of the logarithm of this multiplication.

$$\begin{aligned} \bar{P}_s^{MAP}, \bar{P}_n^{MAP} &= \arg\max_{\bar{P}_s, \bar{P}_n} [f_s(\bar{P}_s) f_n(\bar{P}_n)] = \arg\max_{\bar{P}_s, \bar{P}_n} \left[\ln\left(f_s(\bar{P}_s) f_n(\bar{P}_n)\right)\right] \\ &= \arg\max_{\bar{P}_s, \bar{P}_n} \left[\ln\left(f_s(\bar{P}_s)\right) + \ln\left(f_n(\bar{P}_n)\right)\right] \end{aligned} \tag{4.40}$$

After replacing the f_s and f_n in (4.40) with their equivalents from (4.38) there will be an equation with $\bar{P}_s$ and $\bar{P}_n$ that should be maximized to find the $\bar{P}_s^{MAP}$ and $\bar{P}_n^{MAP}$.

$$\begin{aligned} \bar{P}_s^{MAP}, \bar{P}_n^{MAP} = \arg\max_{\bar{P}_s, \bar{P}_n} \Bigg[& \ln(c_s) + (a_s - 1)\ln(\bar{P}_s) - \frac{\bar{P}_s}{b_s} + \ln(c_n) \\ & + (a_n - 1)\ln(\bar{P}_n) - \frac{\bar{P}_n}{b_n} \Bigg] \end{aligned} \tag{4.41}$$

By replacing $\bar{P}_n$ with $\bar{P}_x - \bar{P}_s$, the whole equation would be just in terms of $\bar{P}_s$ as:

$$\begin{aligned} \bar{P}_s^{MAP} = \arg\max_{\bar{P}_s} \Bigg[& \ln(c_s) + (a_s - 1)\ln(\bar{P}_s) - \frac{\bar{P}_s}{b_s} + \ln(c_n) + (a_n - 1)\ln(\bar{P}_x - \bar{P}_s) \\ & - \frac{\bar{P}_x - \bar{P}_s}{b_n} \Bigg] \end{aligned} \tag{4.42}$$

Now to find the $\bar{P}_s^{MAP}$ the first derivative of the right hand side term of (4.42) should be taken equal to zero and be solved to have the $\bar{P}_s^{MAP}$. The resulting equation is shown below

$$\frac{a_s - 1}{\bar{P}_s} - \frac{1}{b_s} - \frac{a_n - 1}{\bar{P}_x - \bar{P}_s} + \frac{1}{b_n} = 0 \tag{4.43}$$

In (4.43) $\bar{P}_s$ is actually $\bar{P}_s^{MAP}$. By solving (4.43) the resulting speech power using the MAP criterion will be:

$$\bar{P}_s^{MAP} = \frac{-M \pm \sqrt{M^2 - 4LN}}{2L}$$

$$L = b_n - b_s$$

$$M = (b_s - b_n)\bar{P}_x + (2 - a_s - a_n)b_s b_n \tag{4.44}$$

$$N = (a_s - 1)b_s b_n \bar{P}_x$$

From the two calculated values for $\bar{P}_s^{MAP}$, the positive one and smaller than $\bar{P}_x$ will be selected, since the power cannot be negative and cannot go beyond the total power of the frame. In this way the value of $\bar{P}_n^{MAP}$ can be calculated as $\bar{P}_x - \bar{P}_s^{MAP}$.

4.5.3 MAP periodogram estimation and Wiener filtering

As discussed in chapter 3, we created GMMs on normalized periodograms of speech and different noise types. To estimate the periodograms of the speech and noise that form the observed noisy speech periodogram, a MAP criterion is calculated between the PDFs of speech and noise periodograms using their GMMs. Since the GMMs are created on normalized periodograms (with their powers equal to 1), we need an estimate of speech and noise power to bias these to the level of the speech and noise periodograms of the observed noisy frame. In (4.20), these power estimates were shown as $\bar{P}_s^{MS}$ and $\bar{P}_n^{MS}$ which were calculated using Minimum Statistics method. After finding the proper values for $\bar{P}_s^{MAP}$ and $\bar{P}_n^{MAP}$ from (4.44), they can replace the $\bar{P}_s^{MS}$ and $\bar{P}_n^{MS}$ values in equation (4.20) and in this way the MAP estimate of speech periodogram can be shown as:

$$P_s^{MAP}(\omega) = \frac{\sum_{k=1}^{K}\left(\frac{\mu_{s_k}(\omega)}{\sigma_{s_k}(\omega)} - \frac{\mu_{n_k}(\omega)}{\sigma_{n_k}(\omega)}\right) + \frac{P_x(\omega)}{\bar{P}_n^{MAP}}\sum_{k=1}^{K}\left(\frac{1}{\sigma_{n_k}(\omega)}\right)}{\frac{1}{\bar{P}_s^{MAP}}\sum_{k=1}^{K}\left(\frac{1}{\sigma_{s_k}(\omega)}\right) + \frac{1}{\bar{P}_n^{MAP}}\sum_{k=1}^{K}\left(\frac{1}{\sigma_{n_k}(\omega)}\right)} \tag{4.45}$$

where K is the total number of mixtures in the GMMs which in our experments is taken as 10. $\mu_{s_k}(\omega)$ and $\mu_{n_k}(\omega)$ are the k-th mean vector (periodogram) of speech and noise GMMs respectively. $\sigma_{s_k}(\omega)$ and $\sigma_{n_k}(\omega)$ are the k-th mean vector (periodogram) of speech and noise GMMs respectively. $\boldsymbol{P}_x$ is the periodogram of the observed noisy speech frame. The MAP estimate of noise periodogram can be calculated as $P_n^{MAP}(\omega) = P_x(\omega) - P_s^{MAP}(\omega)$ and as it might result in some negative frequency components which is of no meaning for periodograms, they will be zeroed. Using these MAP estimates of speech and noise periodograms a Wiener filter can be constructed to de-noise the noisy frame as:

$$W(\omega) = \frac{P_s^{MAP}(\omega)}{P_s^{MAP}(\omega) + P_n^{MAP}(\omega)}$$
$$\hat{S}(\omega) = W(\omega)X(\omega) \tag{4.46}$$

where $W(\omega)$, $\hat{S}(\omega)$ and $X(\omega)$ are the Wiener filter, the estimated spectrum of clean speech and the observed noisy speech spectrum, respectively. Using inverse DFT, $\hat{S}(\omega)$ can be converted to $\hat{s}_m(t)$ which is the estimate of the m-th frame of the clean speech in time domain. It can be put together with the other neighbor frames using overlap-add method and form the whole clean speech file. This procedure is illustrated in Figure 4.17.

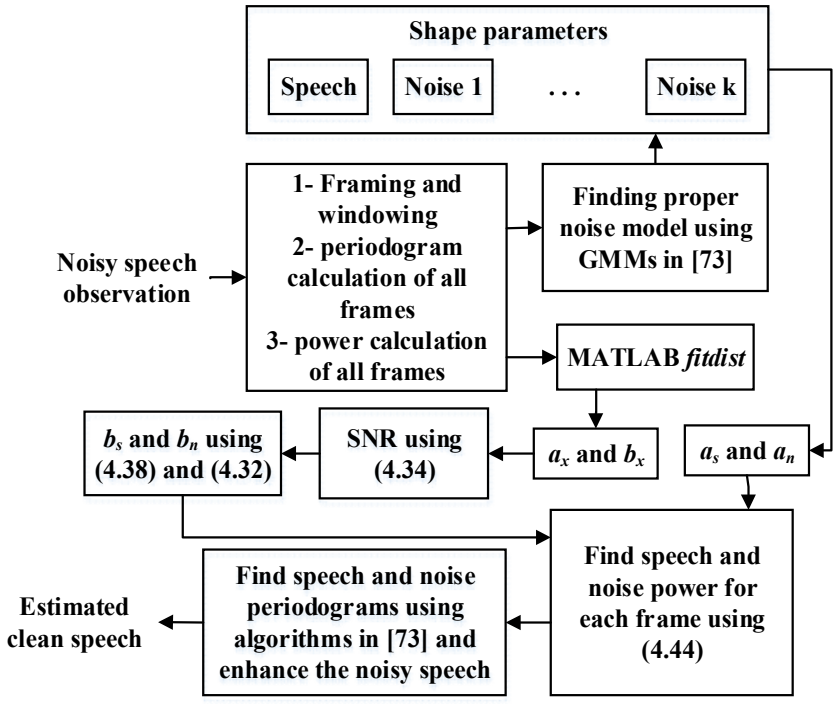

Figure 4.17: Power estimation and enhancement procedure using the introduced power estimation and the enhancement method introduced in [73]

The enhancement process and filtering is exactly the same as the simple MAP method discussed in [73] and just the power estimation method is changed the way shown in Figure 4.17. The enhancement results using this method are discussed in section 5 in Figure 5.11.

4.6 Improved Wiener filter

We have used the Wiener filter in its frequency domain form with different periodogram estimation methods. Using all these methods we tried to extract an estimate of clean speech periodogram and noise periodogram from the observed noisy speech periodogram and construct a proper Wiener filter for the enhancement of the observed frame. The performance of such a Wiener filter is highly dependent on the accuracy of the estimated periodograms. Most of the discussed periodogram estimation methods are of good performance for frames with high speech activity where the speech

level is reasonably higher than the noise level. The critical situations is when the speech power is at the noise level or lower as the possibility of wrong frequency components in estimated periodograms will increase. Subsequently some components that are basically for the noise periodogram may appear in the speech periodogram which can cause some residual background noise and hence lower the quality of the enhanced speech. Moreover, some components that are basically for the speech periodogram may appear in the noise periodogram which can lower the intelligibility of the enhanced speech.

In methods like Minimum Statistics that the noise periodogram is estimated through tracking the minima of a smoothed noisy speech periodogram as in [71], the estimated speech periodogram will be too smooth to track the sharp peaks of the original speech periodogram. In codebook based methods either full search codebooks or GMM based methods as in [59, 60, 64], since there is a limited number of centroids representing all possible periodogram shapes, the estimated speech periodogram cannot accurately simulate the whole variety of speech periodogram shapes. Such errors are quite probable in frames with low SNR values and hence we need to come up with a Wiener filter that is more robust all the estimation errors. Having more variety of periodogram shapes is more related to the codebook or GMM design and the optimal number of clusters and classification algorithm and is not the scope of this topic, but removing as many unwanted components as possible from the enhanced speech, can be attained by improving the Wiener filter formula. The Wiener filter is constructed by dividing the speech periodogram by the noisy speech periodogram and since the estimated speech periodogram in the numerator is not accurate especially for low SNRs, we can compensate it with more attenuation at lower SNRs. For higher SNRs we assume that the estimation is accurate enough and we do not need any more attenuation. Such an extra attenuation for low SNRs and none for high SNRs is like increasing the denominator of the Wiener filter with a factor that has an inverse relationship with the

frame SNR. Such a factor can be like multiplying the nomerator by a factor or adding a constant to it. Multiplication can increase distortion since it will mess with the relationship between periodogram components and will change the shape of the periodogram which is not desirable. Adding a constant to the denominator can preserve the shape of the noisy speech periodogram whilst increasing all the components at the same level and in this way all the components in the noisy speech will be attenuated. Such attenuation will not affect the large speech components which are higher than the average noise level but those ones that are below the average noise level and have the potential of being mistaken with noise components We should mention again that in this way the noise components will be attenuated to the highest degree at the cost of losing some low level speech components. This will increase the quality and hence the SNR of enhanced speech but lower the intelligibility. With such a constant in the denominator, the Wiener filter becomes:

$$\widehat{W}(\omega) = \frac{\hat{P}_s(\omega)}{P_x(\omega) + cu(\omega)} \tag{4.47}$$

where $\widehat{W}(\omega)$ is the estimated Wiener filter, $\hat{P}_s(\omega)$ is the estimated speech periodogram, $u(\omega)$ is a unity vector with Ω frequency bins and all components are equal to one and c the constant that is going to be added to all the components of the noisy speech spectrum. Since $\widehat{W}(\omega)$ should be as close as possible to $W(\omega)$ in (4.47) we should have:

$$\frac{\hat{P}_s(\omega)}{P_x(\omega) + cu(\omega)} = \frac{P_s(\omega)}{P_x(\omega)} \tag{4.48}$$

We know that the noisy speech periodogram is the sum of speech and noise periodograms and hence replacing the nominator of the right hand side fraction with $P_x(\omega) - P_n(\omega)$ gives:

$$\frac{\hat{P}_s(\omega)}{P_x(\omega)+cu(\omega)} = 1 - \frac{P_n(\omega)}{P_x(\omega)} \tag{4.49}$$

Since we can say $\hat{P}_s(\omega) = \left[\hat{P}_{s_1}, \hat{P}_{s_2}, \dots, \hat{P}_{s_\Omega}\right]$, $P_x(\omega) = \left[P_{x_1}, P_{x_2}, \dots, P_{x_\Omega}\right]$ and $P_s(\omega) = \left[P_{s_1}, P_{s_2}, \dots, P_{s_\Omega}\right]$ which are the vector representation of the periodograms, writing (4.49) for each frequency component and simplifying it we have:

$$\frac{\hat{P}_{s_\omega}}{P_{x_\omega}+c} = 1 - \frac{P_{n_\omega}}{P_{x_\omega}} \rightarrow c\left(1 - \frac{P_{n_\omega}}{P_{x_\omega}}\right) = \hat{P}_{s_\omega} + P_{n_\omega} - P_{x_\omega} \tag{4.50}$$

If we rewrite this equation for all frequency components we will have:

$$\begin{cases} c\left(1 - \frac{P_{n_1}}{P_{x_1}}\right) = \hat{P}_{s_1} + P_{n_1} - P_{x_1} \\ c\left(1 - \frac{P_{n_2}}{P_{x_2}}\right) = \hat{P}_{s_2} + P_{n_2} - P_{x_2} \\ \quad\vdots \qquad\quad \vdots \qquad \vdots \qquad \vdots \\ c\left(1 - \frac{P_{n_\Omega}}{P_{x_\Omega}}\right) = \hat{P}_{s_\Omega} + P_{n_\Omega} - P_{x_\Omega} \end{cases} \rightarrow cA(\omega) = B(\omega) \tag{4.51}$$

where the left hand side of all these equations can be replaced with a vector as $A(\omega)$ and the right hand sides with $B(\omega)$ both of size $\Omega\times 1$ (Ω rows and 1 column). In this way we will have one unknown variable of c and a total number of Ω equations that should be solved to find this variable. Since the value of $\frac{P_n(\omega)}{P_x(\omega)}$ in $A(\omega)$ cannot be larger than 1 (because $P_x(\omega) \geq P_n(\omega)$), then all the negative values of $A(\omega)$ will be zeroed. (4.51) can be considered as an over-determined equation and hence we can solve it the way discussed in section 3.3.3. The final value for c can be calculated by multiplying the pseudo-inverse of $A(\omega)$ and multiply it by $B(\omega)$. Such a pseudo-inverse can be calculated using the *pinv* function in MATLAB. This calculation is:

$$c = \text{Pinv}\big(A(\omega)\big)\times B(\omega) \tag{4.52}$$

where $\times$ represents the matrix multiplication. In this equation $\text{Pinv}\big(A(\omega)\big)$ is of size $1\times\Omega$ and multiplying it by $B(\omega)$ of size $\Omega\times 1$ will result in the scalar c. This calculated value for c together with the estimated speech and noisy speech periodogram can be used in (4.47) to construct an improved Wiener filter for the enhancement of the current noisy frame. Using this constant in the denominator of the Wiener filter can result in adaptive attenuation for different frames, more attenuation for frames with more noise and vice versa, that can lead to a high degree of noise reduction. The results of this method are discussed in section 5 in Figure 5.14.

4.7 Summary

Some new speech enhancement methods based on GMM, MAP estimation and Wiener filtering introduced and discussed mathematically. In each method we tried to resolve the shortcomings of the other existing similar methods. In the next chapter we are going to analyze the performance of the introduced methods in terms of quality and intelligibility.

5 Experiments

In this chapter we are going to discuss the performance of all the introduced algorithms from chapter 4. This analysis is in terms of quality and intelligibility of the enhanced speech using these methods.

We have formed 6 noise Periodogram datasets using long files of each noise type babble, white, pink, destroyer engine, factory and HF channel and each noise file contains more than 1.8 million samples. The noise files are converted from originally 16 kHz to 8 kHz from the NOISEX dataset available on the Rice University Digital Signal Processing (DSP) group home page [79]. We have also created a speech Periodogram dataset using training data obtained from the TIMIT dataset [80] with each clean speech file containing around 30000 to 60000 samples. The speech files are also quantized with 16 bit and converted from 16 kHz originally to 8 kHz. The test files of speech and noise (TIMIT has some test speech files and from each noise file some samples are not used in the Periodogram dataset collection and are used instead as test files) are used to create noisy observations at -5, 0, 5 and 10dB input SNRs. All these training files of speech different noisy type are divided to some overlapping windowed frames. These frames have 75% overlap and each has 256 samples which for 8 kHz sampling rate is equal to 32 ms length and each of them will be multiplied by a Hamming window with the same number of samples as discussed in section 01 and illustrated in Figure 2.1. In this way for each windowed frame the periodogram will be calculated and inserted into the corresponding periodogram dataset. For calculating the periodogram we need to find the DFT or the Discrete Fourier transform of these frames and this DFT can be implemented using Fast Fourier Transform (FFT). We applied FFT of length 512

(frequency components) on each frame of 256 time samples. Since the Fourier transform of a signal is symmetric to the zero frequency (DC component), we will take the half band of each Fourier transform plus the DC component, totally 257 frequency components, as the Fourier transform of the frame. In this way in all the equations that we had so far we can take $\Omega = 256$ and hence $\omega \in \{0,1,\dots,256\}$. These Fourier transforms contain complex components and to find the periodogram of these Fourier transforms, we need to find the squared value of the amplitude of each frequency component. Using this procedure for speech and different noise training files, we will have collections of periodogram vectors with 257 frequency components which are called periodogram datasets. The resulted datasets had almost 800000 periodograms for speech and 50000 periodograms for each noise type.

To create these noisy speech signals with any desirable SNR, to test the performance of the proposed algorithms, we will use the test files of each signal type and these files are not used in the periodogram dataset creation. To have a noisy speech file, we will select one of the test files from the TIMIT dataset and from the test file of the proper noise type, we will select a part with the same length of the speech file. To have the desired SNR in the noisy speech file, the proper level of the selected part of the specific noise signal (a number multiplied by noise), will be added to the speech signal. If the speech file and the selected noise part are of length L, the power of these files can be calculated as:

$$\bar{P}_s^t = \sum_{m=1}^{L} s_L^2(m)$$
$$\bar{P}_n^t = \sum_{m=1}^{L} n_L^2(m) \qquad (5.1)$$

where $s_L(m)$ and $n_L(m)$ are the time domain speech and noise signals of length L and $\bar{P}_s^t$ and $\bar{P}_n^t$ are the power of the $s_L(m)$ and $n_L(m)$ in time domain, respectively. Using these power values, the proper coefficient for the noise signal can be calculated as:

$$c_n = \sqrt{\frac{\bar{P}_s^t}{\bar{P}_n^t} \times 10^{-\frac{SNR}{10}}} \tag{5.2}$$

where SNR is the expected Signal to Noise Ratio after multiplying this coefficient by the noise signal. The final noisy speech can be created as:

$$x_L(m) = s_L(m) + c_n n_L(m) \tag{5.3}$$

Now these noisy speech signals can be used for testing the proposed algorithms.

The next stage is to create the GMMs for the speech and different noise types. This procedure is discussed in section 3.4. Earlier we mentioned that we created vary large periodogram datasets for speech and each noise type and now using the EM algorithm we are going to create some GMMs on them. As discussed in section 4.1 we will use 10 Gaussians in each GMM and hence in the EM algorithm implemented in *gmdistribution.fit* command in MATLAB, we will set 10 as the number of classes and at the output we will come up with 10 probabilities (1 for each Gaussian), 10 mean vectors and 10 covariance matrices. Since we are neglecting the cross-correlation of frequency components, we just use the diagonal of the covariance matrices as the variance vector for the calculations. In this way we can create one GMM for speech and 6 GMMs for the 6 different noisy types of White, Babble, Pink, Factory, HF Channel and Destroyer Engine.

All the tests are performed on a PC with Intel Core i5 3.2 GHz processor and 16 GB RAM. We used MATLAB 2013b on 64 bit Windows 7 OS to implement the algorithms.

5.1 Performance measurement

To measure the performance of each speech enhancement algorithm and to be able to compare them we need to use some criteria. In speech enhancement algorithms the criteria which can represent the algorithm performance are the quality and the intelligibility of the enhanced speech. Quality can be considered as the level of noise in the enhanced speech file. The less noise remaining in the enhanced speech the higher the quality will be. The intelligibility means how good the enhanced speech can be understood by a listener. The more words that could be recognized by the listener the more intelligible the enhanced speech is. To determine these criteria in for enhanced speech we can use subjective and objective methods. In subjective measurements of quality and intelligibility, we can use a group of human listeners to rate the outcome of the enhancement algorithms. In this way to measure the quality of the played speech they can rate it for example from very good to very bad [27]. To measure the intelligibility they could be asked to determine the number of recognized words in the played enhanced speech. Such subjective measurement methods are expensive and time consuming since we need a group of listeners that should be trained. To overcome these problems we will use objective measurement methods. In objective methods we use some numeric criteria to measure the quality and the intelligibility of the enhanced speech and hence it will be easier to compare different speech enhancement algorithms. Some of these objective criteria are discussed in the following sections. All these mentioned criteria will be calculated on both the enhanced speech and the noisy speech and then the difference of these two values will be considered as the improvement resulted from the speech enhancement algorithm.

5.1.1 **Segmental SNR**

In this criterion, we can divide the enhanced speech to some segments and in each segment calculate the power of noise and the power of speech. Dividing the power of speech to the power of noise will give the value of SNR in that segment and averaging it over all the segments of the enhanced speech will give the value of the segmental SNR [27]. Since we have access to the clean speeches that are used for the creation of the noisy speeches, the speech power in each segment of the noisy speech is considered as the power of corresponding segment from the clean speech. In this way, the power of noise is considered as the power of the difference of the clean speech and the enhanced speech. The segmental SNR can be calculated as:

$$SNR_{seg} = \frac{10}{L}\sum_{l=1}^{L}\log_{10}\left(\frac{\sum_{t=(l-1)\times T+1}^{l\times T} s^2(t)}{\sum_{t=(l-1)\times T+1}^{l\times T}\left(s(t)-\hat{s}(t)\right)^2}\right) \tag{5.4}$$

where $s(t)$ and $\hat{s}(t)$ are the original and the enhanced clean speeches, respectively. The total number of segments in these signals are L and each segment is of length T time samples. The segmental SNR is shown in dB and its higher values represents the higher quality of the enhanced speech.

5.1.2 **Perceptual Evaluation of Speech Quality (PESQ)**

In this criterion, we have a pre-processing stage in which the input signals are normalized and time-aligned and in this way we can overcome the problems caused by delay and non-matching gains. After this stage, the signals are compared using a perceptually motivated distance measure. The signals are divided into frames of length 32 ms. A bark scale filter bank with 42 bands is then applied to the power spectrum of each frame and hence a loudness spectrum is produced. The next stage is calculating a difference between the signals. The negative differences are related to the components

being added to the signal, like noise. Positive differences mean that the signal is attenuated. Using these differences between the signals, the disturbance values are calculated which are then used to form a single score by averaging these disturbances between the frames and then between the files. The overall disturbance score is then scaled to within the range of 1.0 and 4.5 to produce a score. The PESQ provides a good correlation with subjective tests. PESQ is a quantitative psycho-acoustic measure that is used to evaluate how the enhanced speech is appreciated. To calculate PESQ as explained in [14], the routine available in [81] is used.

5.1.3 BSS-Eval toolbox

This is a toolbox to evaluate some Blind Source Separation (BSS) criteria which is introduced in [82]. Using these criteria we can say how good the clean speech is separated from the noise in the observed noisy speech. These criteria are Source to Distortion Ratio (SDR), Source to Interference Ratio (SIR) and Source to Artefact Ratio (SAR). SDR measures the amount of distortion in the output enhanced speech signal and is defined as the ratio of the energy of the clean signal, and the energy of the distortion [83]. SIR is defined as the ratio of the target enhanced clean speech power to the power of the interference signal and measures the amount of undesired interference signal still remained [83]. SAR measures the quality in terms of absence of the artificial noise [83]. SDR represents the overall quality of the enhanced speech while SIR and SAR represent the amount of noise reduction and are proportional to the inverse of the distortion [3].

5.2 Optimization MAP and simple explicit MAP

Here we are going to implement the “explicit MAP using an optimization algorithm” as discussed in section 4.2 and the “simple explicit MAP” as discussed in section 4.3 and

measure their performance and compare them with each other. We set the parameters of *fmincon* function as $TolFun = 10$, $TolCon = 10^{-3}$ and $TolX = 10^{-3}$ which are calculated by trial and error. As we decrease these values, the number of iterations in the optimization algorithm will increase and will result in higher accuracy and hence higher processing time. We found that for values lower than these, the accuracy will not increase significantly. To analyze the performance of the simple MAP algorithm we compared our results with the Nonnegative Matrix Factorization (NMF) method discussed in [3] which has reported outstanding speech enhancement results, recently in which a MAP estimation of the speech periodogram has been used. To do so, we used the MATLAB files of the author of [3] which is available in [84] and trained it with the same training datasets that we used for our GMM procedure. We used all different test sentences of different speakers from the TIMIT dataset (192 sentences) and all the measured performance criteria are averaged on these 192 files. To perform experiments, each noisy speech file is divided into some 75% overlapping Hamming windowed frames of length of 256 samples which for 8kHz sampling rate is equal to 32ms. Using a FFT with 512 frequency samples, the Fourier transform of the noisy speech frame is calculated. The amplitude of all these frequency components are calculated and then squared to find the periodogram of noisy speech frame. Since the periodogram is symmetric with respect to the DC component (frequency of zero) half band plus DC component, totally 257 frequency components are considered as the periodogram of the noisy speech frame. Using the defined algorithms in sections 4.2 and 4.3, the MAP estimates of speech and noise periodograms presenting in the noisy speech periodogram frame are calculated. The appropriate Wiener filter is then constructed using these periodograms and multiplied by the noisy speech spectrum (Fourier transform) to suppress the noise. These resulted spectrums are recovered in the time-domain by using the inverse FFT and are put together with the neighbor overlapping frames in the same

order of the input noisy speech through overlap-add method. Hence the enhanced speech in the time domain is created. As discussed previously, we are using the MS method to have an initial estimate of speech and noise Periodograms and their power and also finding the right GMM of noise. We have also calculated all the performance criterions for the case that we just use the MS method as discussed in [25] for the estimation of noise and speech periodograms and use them in Wiener filter construction. This method is compared to other results as an unsupervised speech enhancement method versus MAP and NMF which are supervised methods.

For each noise type (totally 6), and each input SNR (-5, 0, 5 and 10 dB), the performance criterions are averaged on the test files. All the criterions are averaged on the 6 noises and are shown with respect to the input SNRs in Figure 5.1. In terms of segmental SNR, PESQ and SIR improvement, the simple MAP algorithm performs significantly better than the NMF method and the other ones too. This means that we have a higher degree of enhancement with the simple MAP algorithm than the optimization, NMF and MS methods. But in terms of SDR improvement and SAR we can see that NMF method has better performance. Since we are using limited GMMs for speech and noise and there are 10 mixtures (Periodograms) in each GMM, the resulting Wiener filters for enhancement of noisy frames will definitely cancel some frequency components of speech or keep some frequency components of noise within them and hence increase the distortion in the enhanced speech which results in less SDR improvement and SAR for the simple MAP method rather than the NMF method.

We also compared the two optimization MAP and simple MAP algorithms in terms of processing time. For input speech files of average length of 3.5 secs, the processing time for the optimization method was about 200 secs while it was just 1.9 secs for the simple MAP method and 2.5 secs for the NMF method. Hence, the optimization method is of no practical use for real-time applications as compared with the simple MAP technique.

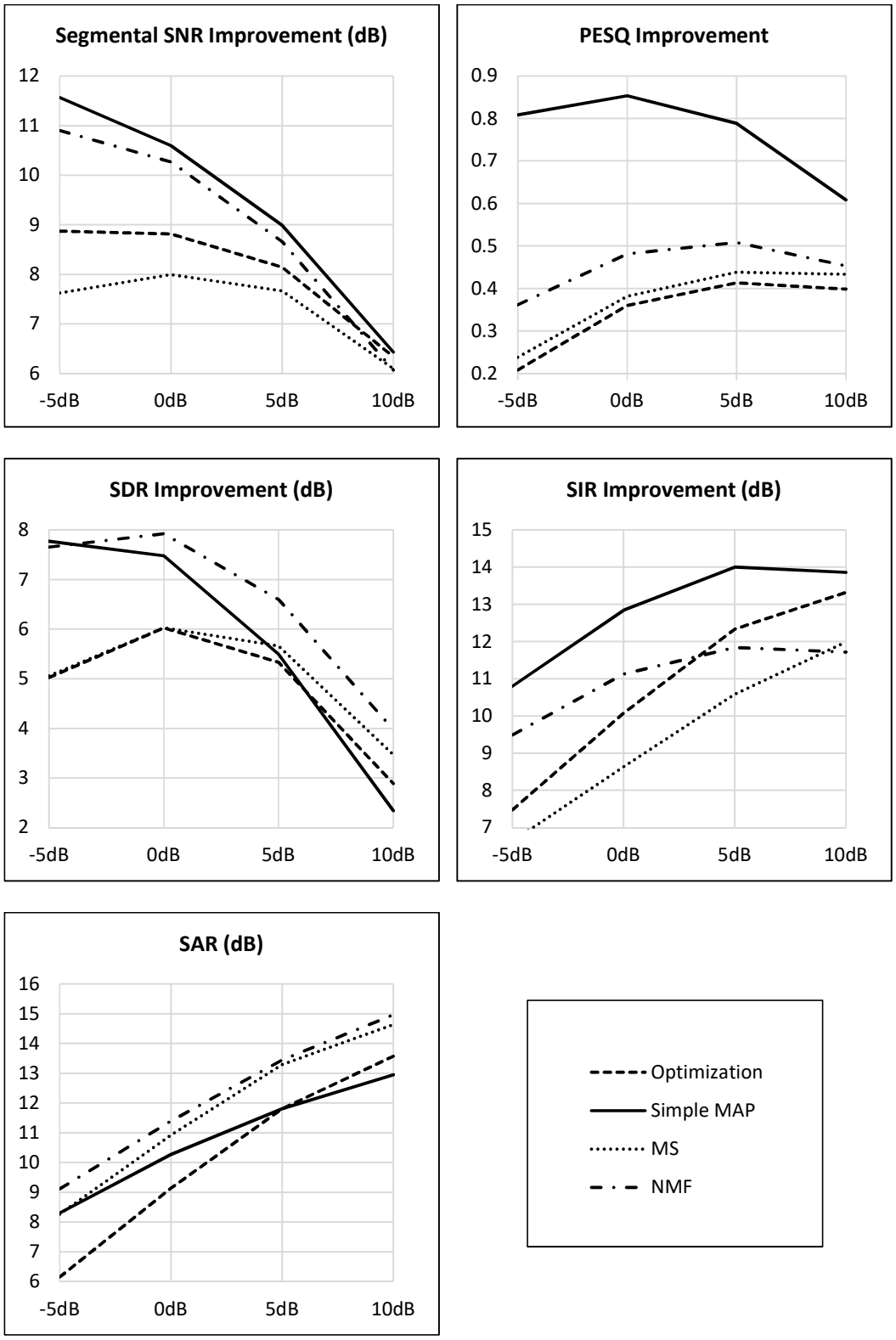

Figure 5.1: Comparison the performance of Optimization MAP, Simple MAP, MS and NMF algorithms. The horizontal axis shows the input SNRs

In Figure 5.1 we have shown the performance of the MS algorithm which is an unsupervised speech enhancement method. A detailed comparison of the proposed Simple MAP method and the NMF algorithm in terms of Segmental SNR improvement and PESQ improvement can be shown in Figure 5.2 and Figure 5.3.

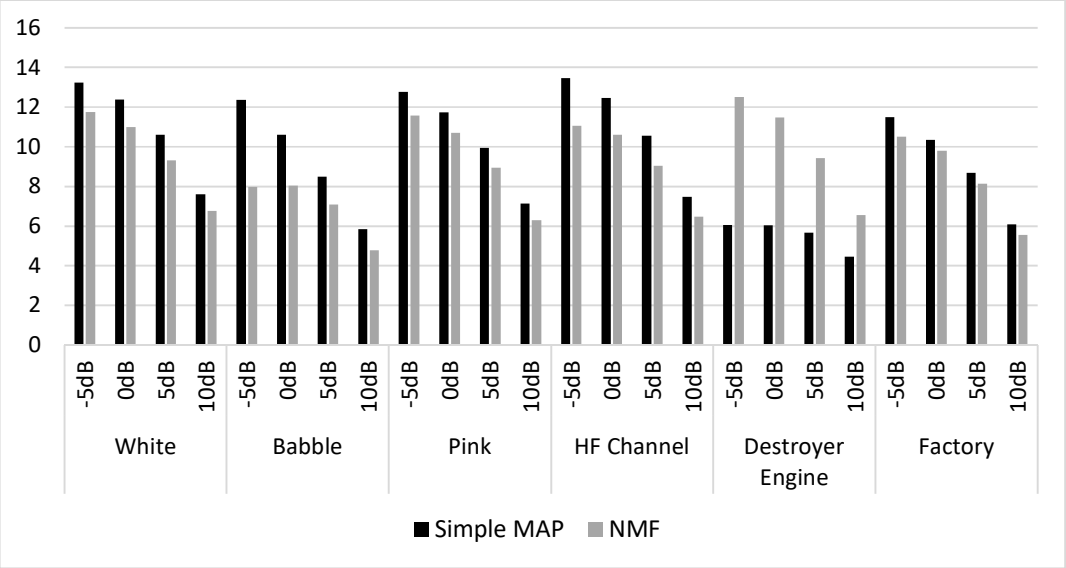

Figure 5.2: Segmental SNR improvement comparison for different noise types and different input SNRs.

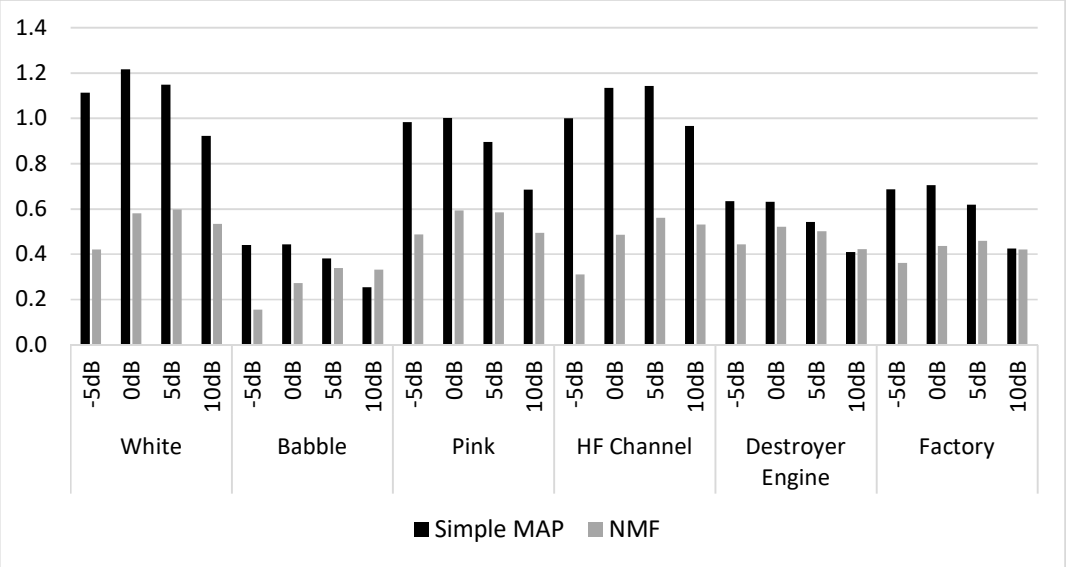

Figure 5.3: PESQ improvement comparison for different noise types and different input SNRs.

It is seen from Figure 5.2, for all input SNRs and for all noise except for destroyer engine noise, the simple MAP method exhibits better segmental SNR improvement than the NMF method. Such poor performance of this algorithm for destroyer engine noise can be due to its high level of non-stationarity and very sharp changes of frequency components from one frame to another that will result in inaccurate GMM that cannot represent all variety of periodogram shapes. Also from Figure 5.3 we can see that almost for every noise with every input SNR the simple MAP exhibits higher PESQ improvement. The PESQ improvement level for babble noise is lower than the other noise types and that is due to the similarity of this noise type to speech signal which makes it harder to easily distinguish the utterances of the speech.

We used the periodogram power estimates resulting from MS with our MAP estimation method and proposed a supervised method which attained very good enhancement results regardless of just using MS, which has poor performance. The MS algorithm has also some error in the estimation of power. In this way we tried to implement the simple MAP algorithm using the true power of speech and noise in the analyzing frame. Since we are creating the noisy observations from pure speech and pure noise files, in each analyzing frame we can replace $\bar{P}_s^{MS}$ and $\bar{P}_n^{MS}$ in (4.20) with the real power of original speech and noise in that frame. The comparison of this method with the simple MAP method is shown in Figure 5.4.

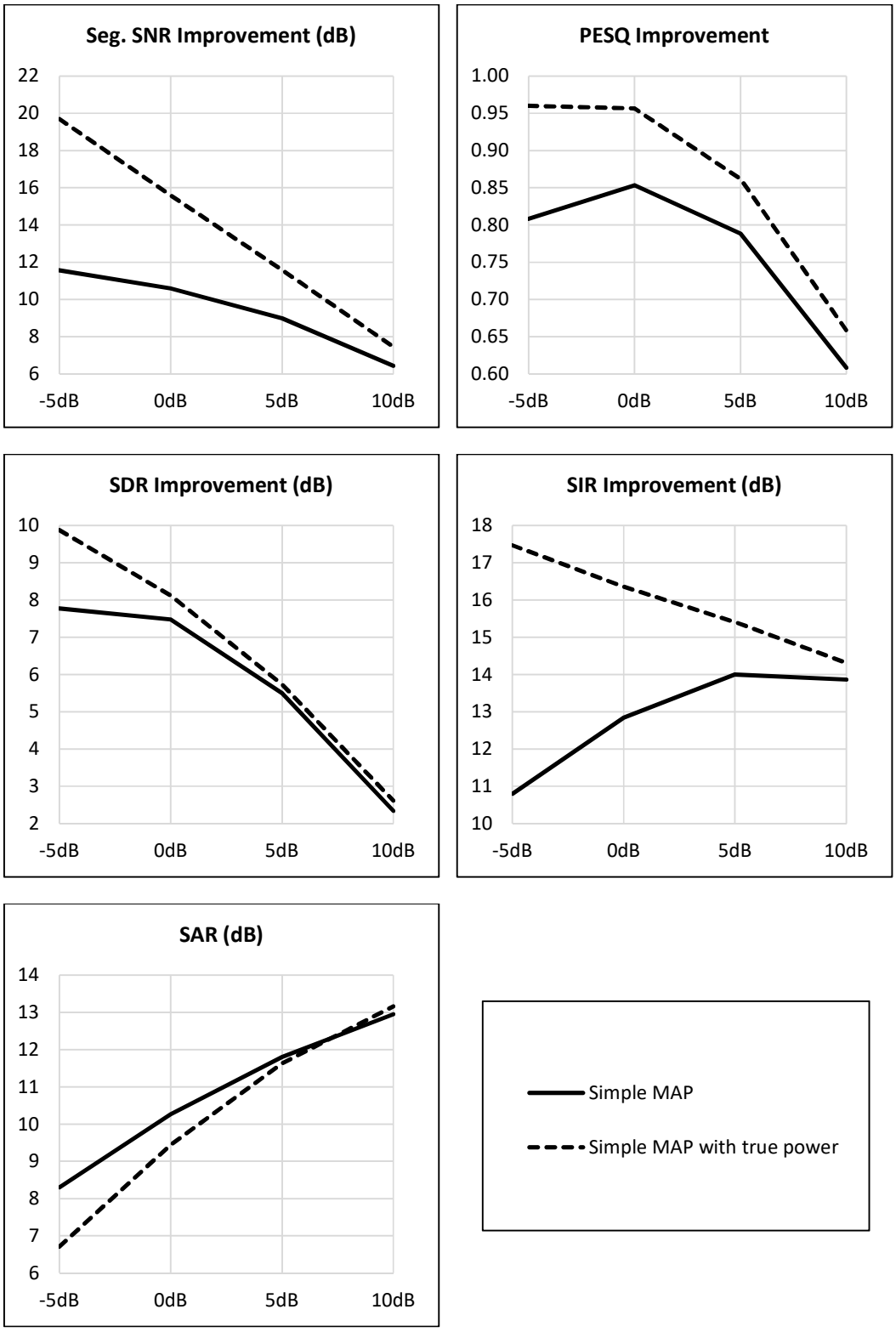

Figure 5.4: Comparison of proposed Simple MAP with the one with the true power of speech and noise applied to it.

As can be seen from Figure 5.4, if we were able to correctly estimate the power of speech and noise in the analyzing frame, we would have significantly increased the segmental SNR, PESQ and SIR. This is an ideal assumption and in practical use we are never able to estimate the true power of speech and noise only from the noisy observation. This experiment can give us the idea about the improvement of simple MAP results by improving the used power estimation method which was the focus of section 4.5. In terms of the SDR and SAR criterion we can see that there will not be a big difference for the case of improvement of the power estimation method, which can justify the poor SDR improvement and SAR criteria when using the simple MAP method in Figure 5.1. This is due to the nature of the Wiener filter and the high degree of enhancement in which we may suppress some frequency components of speech while removing the noise components. In terms of a SIR improvement criterion we can see that the simple MAP has a reverse relation with the one with true power. This is because of the poor performance of the MS algorithm in low input SNRs where the estimated speech periodograms are not accurate and hence the constructed Wiener filter will remove most of the frequency components of the speech as well as noise.

Some sample noisy speeches and their clean and enhanced versions using the “Simple MAP” method for different noise types and different input SNRs are shown in Figure B.1 to Figure B.6 of the Appendix. From the spectrograms in Figure B.1 to Figure B.6 we can see that for the noisy speeches of higher input SNR, more frequency components can be extracted and hence we will have higher resolution in the enhanced speech.

We have also performed another test in which the noisy speech is created as the sum of clean speech and White and Babble noises together. To do so, we just considered that the power of Babble and White noise in the noisy speech are equal and the different

input SNRs are considering the combination of White and Babble noises as a single noise. The enhancement results are shown in Figure 5.5.

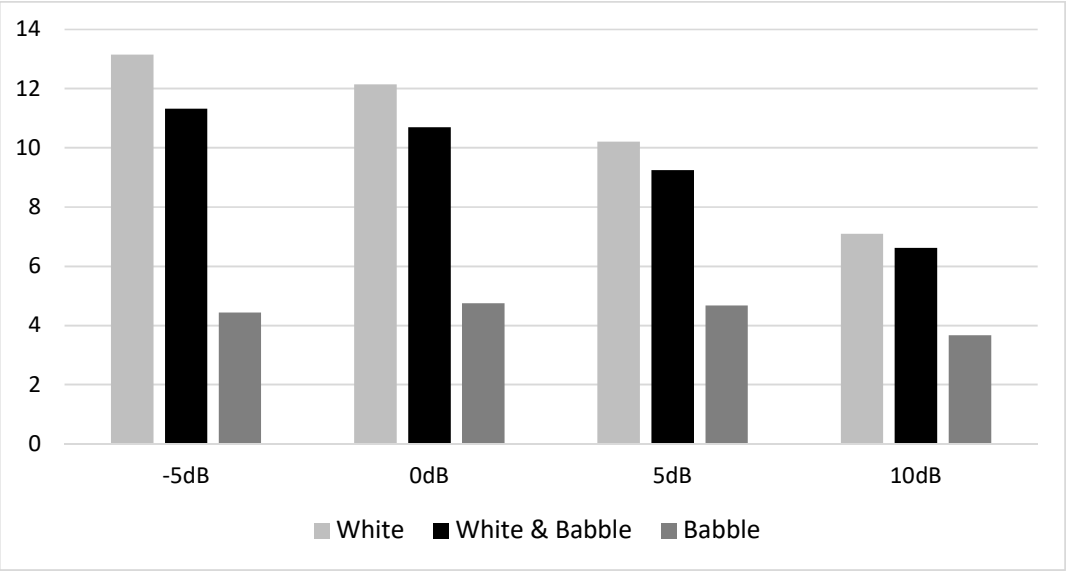

Figure 5.5: Segmental SNR improvement comparison for different input SNRs.

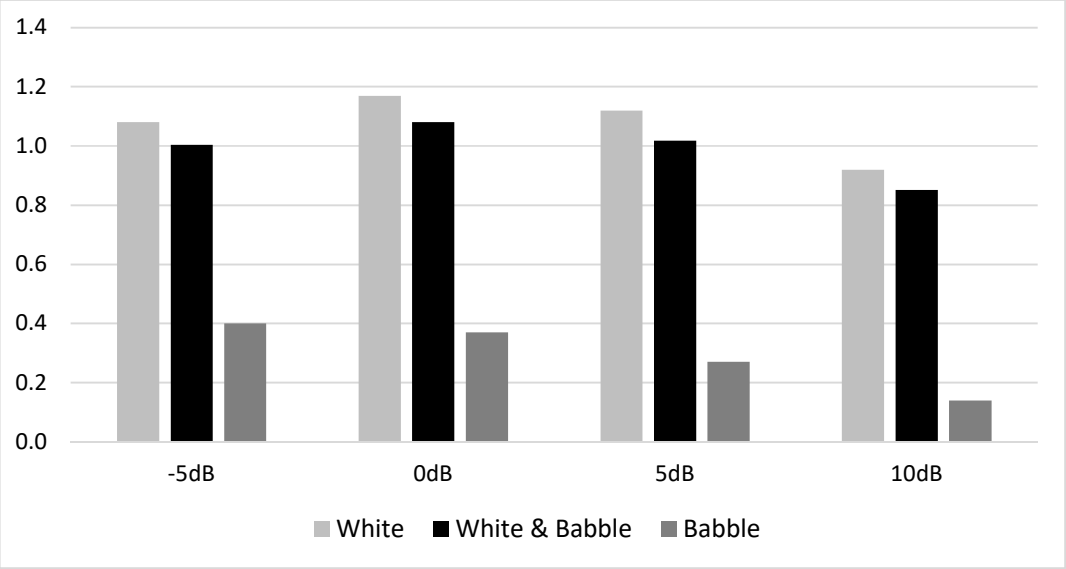

Figure 5.6: PESQ improvement comparison for different input SNRs.

In this case the combination of White and Babble noises can be considered as totally new noise which we do not have its GMM in the collection of noise types GMMs. As can be seen from Figure 5.5 and Figure 5.6, the algorithm has Segmental SNR

improvement and PESQ improvement between those of pure Babble and pure White cases. This algorithm does not care about the noise type or whether it's GMM is available, and just searches among the available GMMs to find the one whose mean vectors are the closes to the current observed noise. In this way lack of some noise types GMMs will not stop the algorithm from working.

5.3 Improved explicit MAP

In this section the performance of the two algorithms discussed in section 4.4 as "MAP periodogram" from (4.22) and "MAP amplitude" from (4.27) will be measured and compared with the performance of the "Simple MAP" method reported in section 5.2. The test procedure for dividing the noisy speech to some overlapping frames, GMMs, filtering and reconstruction of the enhanced speech is like section 5.2 and just an improvement in MAP periodogram estimation is introduced. The enhancement results are shown in Figure 5.7.

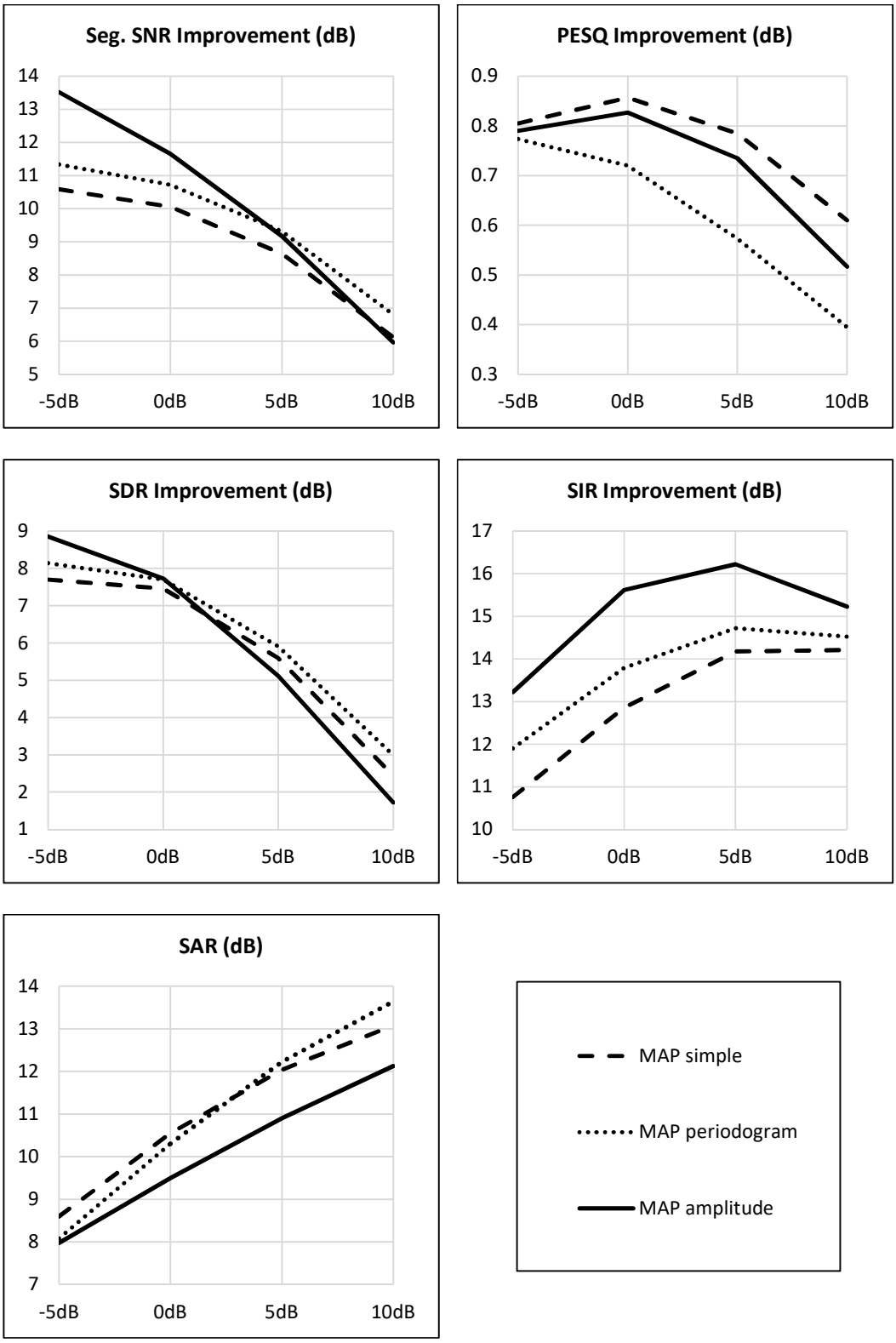

Figure 5.7: Comparison the performance of Simple MAP, Periodogram MAP and Amplitude MAP algorithms. The horizontal axis shows the input SNRs.

As shown in Figure 5.7, in terms of Segmental SNR improvement, the “MAP amplitude” method which could be attained using equations (4.27) and (4.28) is showing very good results with respect the “MAP periodogram” using (4.22) or the “MAP simple” mentioned in [73] which was expected from the assumptions made to get to such a method. This really supports the idea of less residual noise in the enhanced speech frames, using the squared spectral amplitude of speech as the speech periodogram. The Segmental SNR improvement degrades for higher SNRs beyond 5dB since the noisy signal is already of a good quality. In terms of PESQ improvement, the “MAP amplitude” performs way better than the “MAP periodogram” since it does not have those extra frequency components (residual noise) but slightly worse than the “MAP simple” method since in high degree of enhancement, the cancellation of some speech frequency components are inevitable. In low SNRs (-5dB and 0dB) the “MAP amplitude” has better SDR improvement but for higher SNRs due to higher cancellation of speech frequency components, it results in some distortion in the enhanced signal. In terms of SIR improvement, “MAP amplitude” is way better than other algorithms and this represents very low resulted interference of this algorithm which supports its good SAR that shows the low resulted artefact of this algorithm.

A more detailed comparison on the segmental SNR improvement and PESQ improvement for different noise types with different input SNRs are shown below:

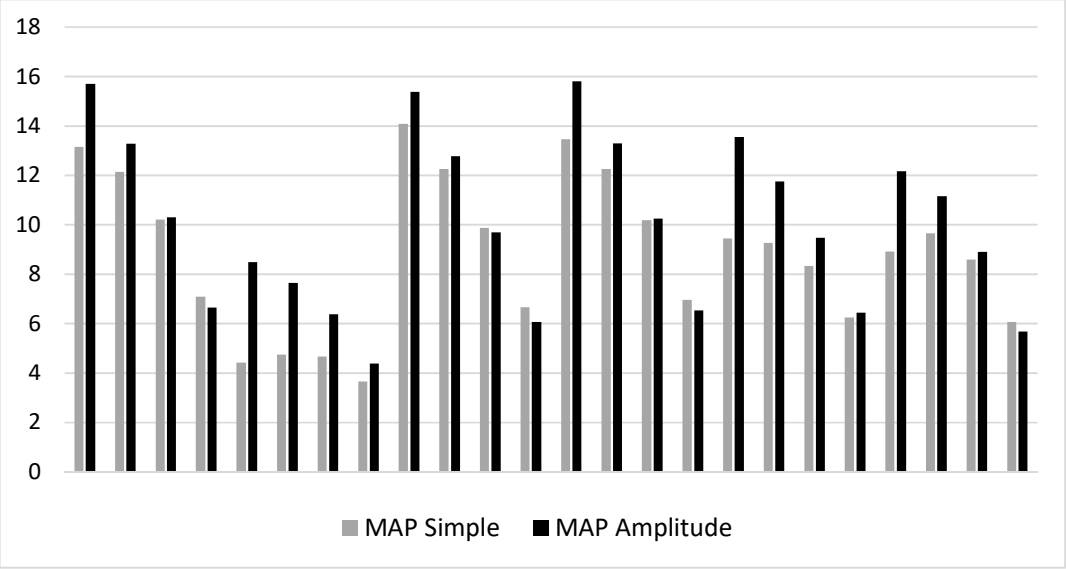

Figure 5.8: Segmental SNR improvement comparison for different noise types and different input SNRs.

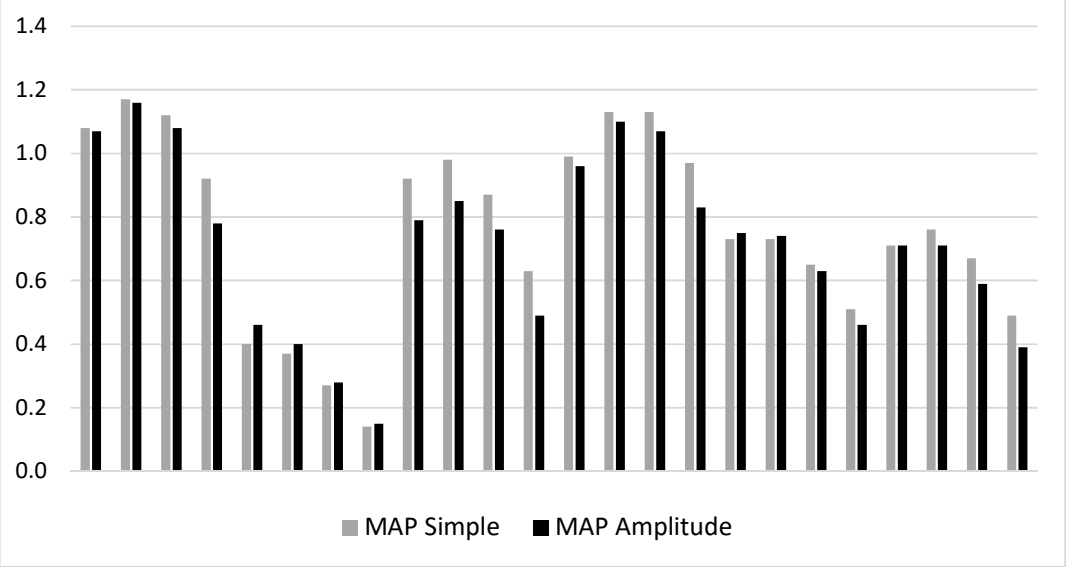

Figure 5.9: PESQ improvement comparison for different noise types and different input SNRs.

As can be seen from Figure 5.8, almost for all noises and all input SNRs the "MAP Amplitude" performs better than "MAP Simple" method which is actually the method discussed as "simple MAP" in section 5.2. Also as was expected from Figure 5.7 it can also be seen from Figure 5.9 that the PESQ improvement for "MAP Amplitude" method

is less than “MAP Simple” method which has already been discussed. Some sample noisy speeches and their clean and enhanced versions using the “MAP Amplitude” method for different noise types and different input SNRs are shown in Figure B.7 to Figure B.12 of the Appendix. By comparing Figure B.7 to Figure B.12 to their corresponding figures among Figure B.1 to Figure B.6 we can see that there are more frequency components present in the spectrogram of the enhanced speech resulted from the “MAP amplitude” method than the “Simple MAP” and this means the higher resolution of the enhanced speech in this method and it can justify the results of Figure 5.7.

5.4 Power estimation using Gamma modelling

Here we are going to analyze the performance of the power estimation method discussed in section 4.5. Since the Minimum Statistic (MS) algorithm is mainly used to estimate the power of noise in the noisy speech signal [71], the power resulted from this method and the Gamma method introduced in section 4.5 are compared. To do this comparison the following formula is used:

$$Error(\%) = 100\times\frac{1}{M}\sum_{m=1}^{M}\frac{\left|\bar{P}_{n_m}-\bar{P}_{n_m}^{est}\right|}{\bar{P}_{n_m}} \tag{5.5}$$

where, M is the total number of frames in a noisy file, $\bar{P}_{n_m}$ is the real noise power in the m-th frame and $\bar{P}_{n_m}^{est}$ is the estimated noise power in the m-th frame that could be a result of either MS or Gamma algorithms. This comparison is shown in Figure 5.10.

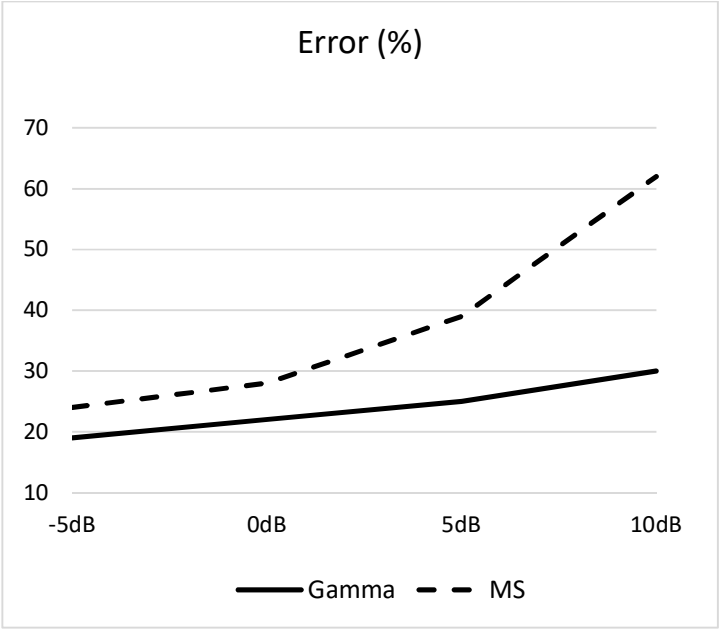

Figure 5.10: Error resulting from the estimated noise power using Gamma and MS algorithms

As can be seen from Figure 5.10, the error resulting from the Gamma method is smaller than MS method. In lower SNRs (-5dB) since the noise level is way higher than speech level the error difference is small but in higher SNRs that the noise is equal or smaller than speech power, the Gamma method exhibits much less error.

The Gamma power estimation method replaced the MS power estimation method in speech enhancement method introduced in section 4.3 and [73], and applied on the same noisy speech files using the same GMMs of speech and noises. All the performance criteria are calculated on these enhanced speech files and compared with those ones using MS algorithm and also the case in which the true power was used in the enhancement procedure. These comparisons are shown in Figure 5.11.

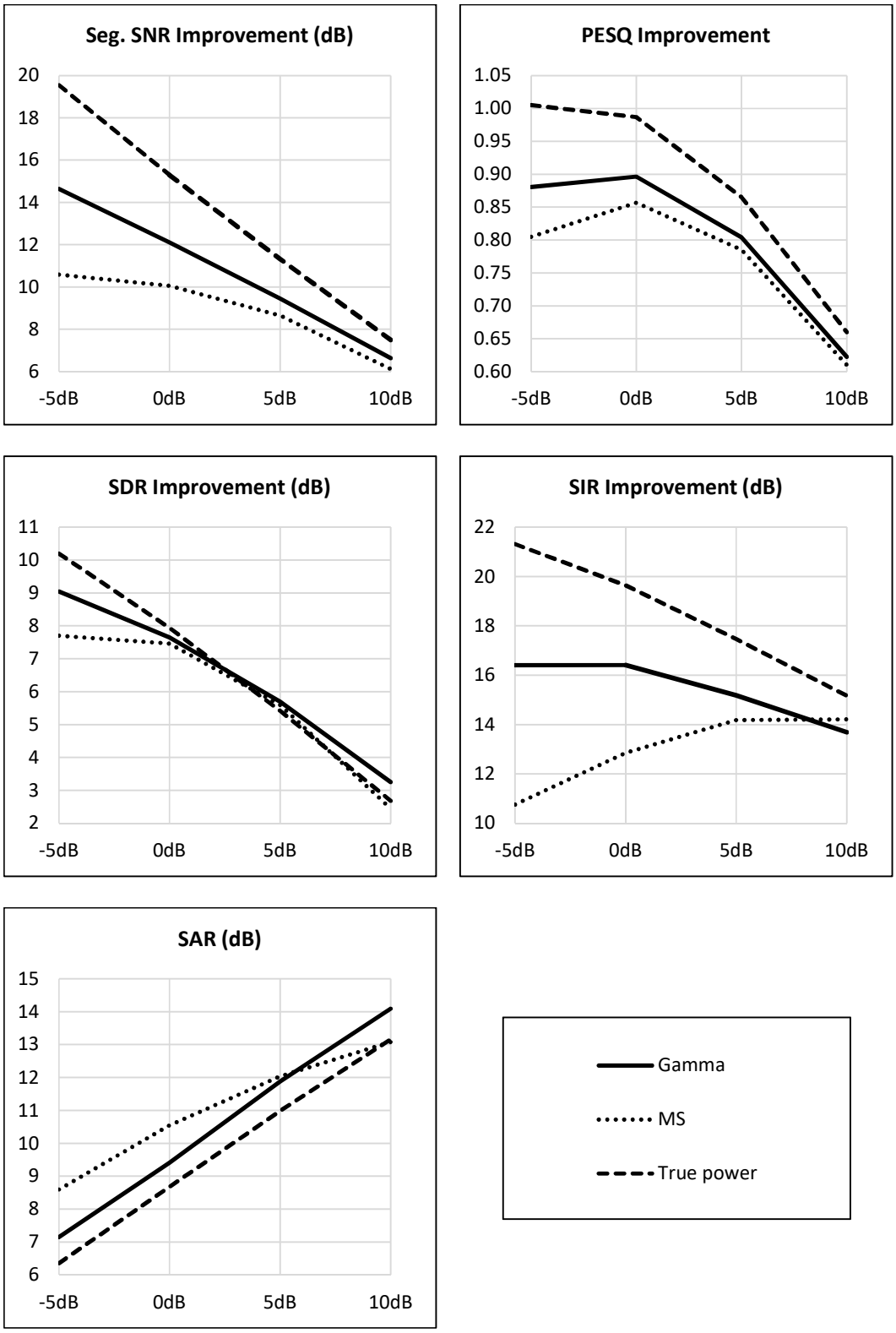

Figure 5.11: Comparison of the performance of speech enhancement algorithms in [73] using Gamma and MS power estimation and also true power. The horizontal axis shows the input SNRs.

As can be seen from Figure 5.11, in terms of all enhancement criterions almost in all input SNRs, the Gamma noise estimation exhibits better performance than the MS algorithm. This performance is still less than the one attained using the true power which is the ideal situation and is shown in Figure 5.4, but is still an improvement to the existing noise estimation methods.

The two power estimation algorithms are also compared while being used for enhancement from a processing time point of view. For input speech files of average length of 3.5 Sec, the processing time for the enhancement algorithm using MS noise estimation was 1.9 sec while it is just 0.75 sec for the one using Gamma power estimation. This difference is due to large amount of calculation used in the implementation of MS as in [71], resulting from the periodogram smoothing procedure while in the Gamma method the power can be calculated very straightforwardly using the model parameters. In the simple MAP method using the MS algorithm during the analysis of each frame, the enhanced version of that frame can be attained at the same time (before getting to the next frame) which makes it applicable for online applications. In the simple MAP using the Gamma method, a proper amount of noisy frames should be captured to extract the proper model parameters and this property makes it offline. But if both algorithms are to be used to enhance a recorded noisy speech, the Gamma power estimation algorithm will work much faster than the MS algorithm.

A detailed comparison of the segmental SNR and PESQ improvement for the two Minimum Statistics and Gamma power estimation methods are shown below:

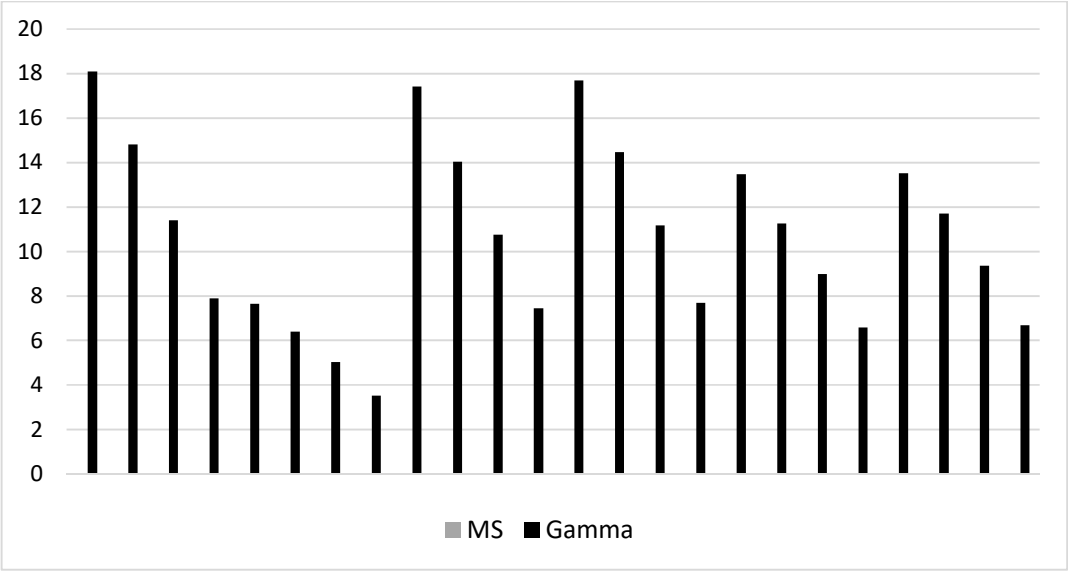

Figure 5.12: Segmental SNR improvement comparison for different noise types and different input SNRs.

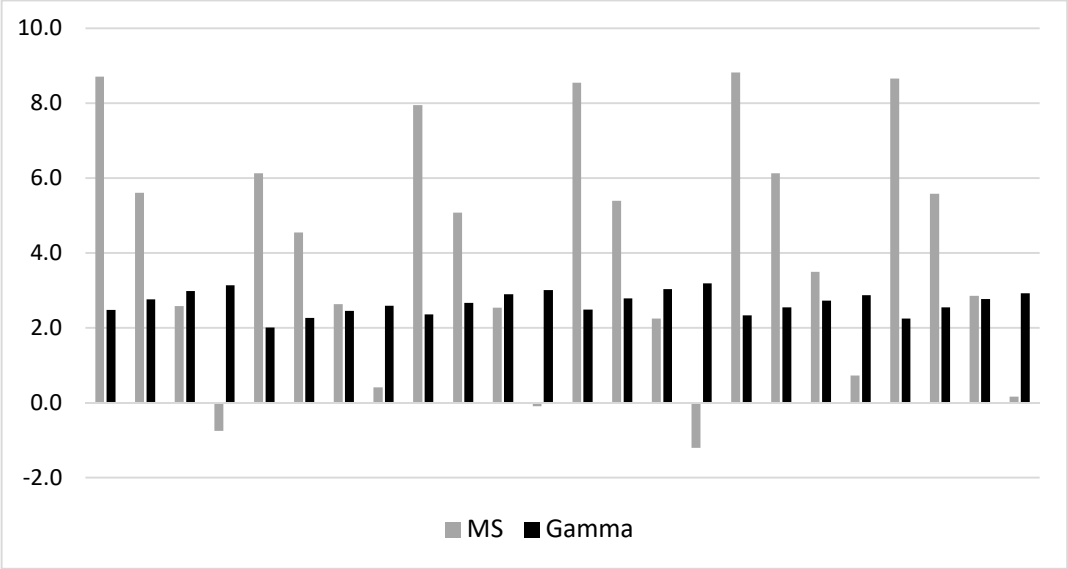

Figure 5.13: PESQ improvement comparison for different noise types and different input SNRs.

As can be seen from Figure 5.12 and Figure 5.13 almost for all noises and for all input SNRs, the Gamma noise estimation works significantly better than the MS method.

5.5 Improved Wiener filter

In this section we use the same speech and noise periodogram estimates resulted from the method introduced in section 4.3 in the new Wiener filter formula introduced in section 4.6 and compare the resulted enhancement criteria with those ones of the classic Wiener filter which used before.

As can be seem from Figure 5.14 for all input SNRs the PESQ improvement in the improved Wiener is slightly higher than the normal Wiener. Since in this method the filter has adaptive attenuation for different frames, more attenuation for frames with more noise and vice versa, it will be easier to distinguish between different types of speech utterances and this will result in higher intelligibility and hence slightly higher PESQ improvement. In terms of Segmental SNR, SDR and SIR improvement, the improved Wiener performs better than the normal Wiener for lower input SNRs (0dB and -5dB). This is exactly expected as in these low input SNRs, the speech is mostly masked by the noise and hence such improved Wiener filter can become of very high performance. For higher input SNRs, since speech is the dominant part of the noisy frame, it is always possible that the lower activity of the speech being treated as noise and hence the corresponding frequency components being removed in the enhanced speech. This will result in less Segmental SNR, SDR and SIR improvement as can be seen in Figure 5.14. Since we are treating different frames with different attenuation, this will slightly affect the original power of the frames and hence generating an artifact in the enhanced speech which results in lower SAR for the improved Wiener with respect to the normal Wiener. Totally for lower input SNRs, which is mostly the focus of speech enhancement methods, the improved Wiener has an outstanding performance over the normal Wiener.

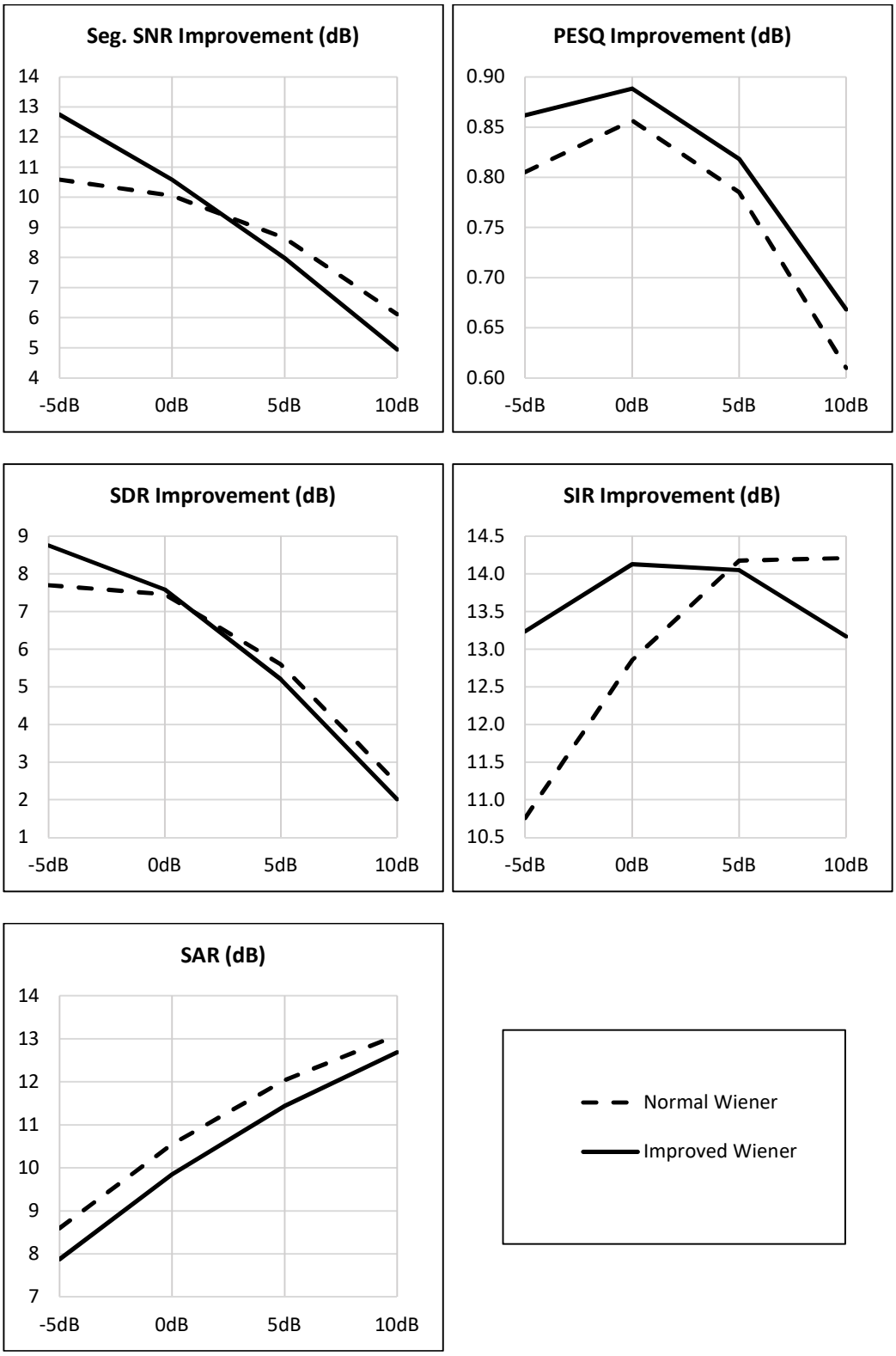

Figure 5.14: Comparison of the performance of normal Wiener and improved Wiener

A detailed comparison of the segmental SNR and PESQ improvement for the two Wiener methods are shown below:

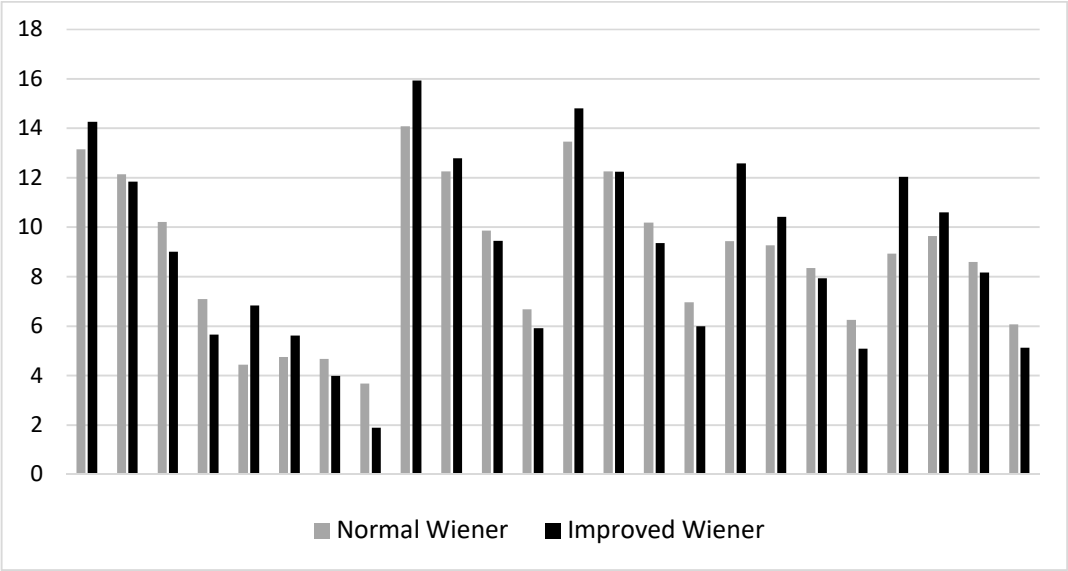

Figure 5.15: Segmental SNR improvement comparison for different noise types and different input SNRs.

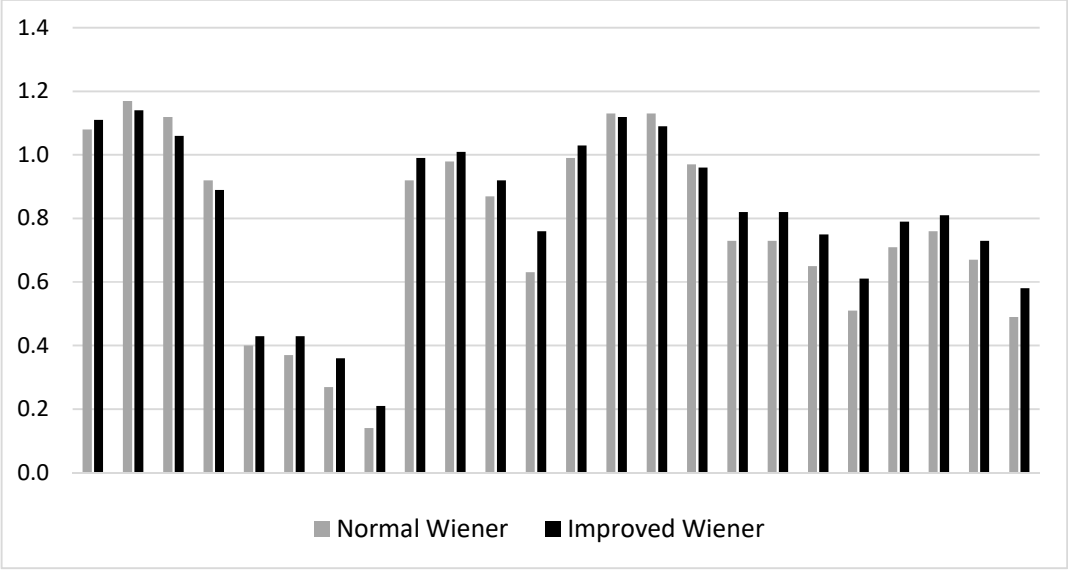

Figure 5.16: PESQ improvement comparison for different noise types and different input SNRs.

As shown in Figure 5.15 and as was expected from Figure 5.14, in Figure 5.15 for most of noises in lower SNRs (-5dB and 0dB) the improved Wiener exhibits higher

segmental SNR improvement than the normal Wiener. From Figure 5.16, we can see that for most noises except for White noise and almost for all input SNRs, the PESQ improvement in improved Wiener is slightly higher than the normal Wiener. The reason for the poor performance of the algorithm for the White noise is the presence of the noise frequency components in almost all frequencies that treating different frames with different attenuation cannot remove it.

5.6 Summary

All the introduced methods of chapter 4 are analyzed and compared in terms of enhancement performance using intelligibility and quality criteria. The mentioned criteria are measured on the noisy speech and enhanced speech and the differences are reported as the improvement of these criteria. They are averaged over a large number of files for each noise type and different input SNR to come up with universal performance measures.

6 Conclusion and future work

A new GMM has been introduced in which the normalized periodograms of each signal type, both speech and different noises are treated as a vector in a 257 dimensional vector space. The periodograms which have similar shapes form a colony in the 257 dimensional vector space and hence the GMM can relate the probability of each colony (PDF of the distribution) to the mean vector and covariance matrix of each colony.

A new method for MAP estimation of the speech periodogram from the observed noisy speech has been introduced in which a MAP criterion is calculated using the PDFs of clean speech and noise based on the mentioned GMMs. This method uses a series of approximations on MAP formula to make it simple enough to implement. Some improvements on these approximations are then introduced which led to more accurate MAP estimates of speech periodogram.

Experiments showed that the accuracy of MAP periodogram estimates is really dependent to the accurate estimation of the speech and noise power within a noisy speech frame. In an earlier MAP estimation method we used Minimum Statistics (for power estimation) which was of low accuracy. We introduced a new power estimation method in which a MAP criterion is used on Gamma models of speech and noise power. These Gamma models were created offline. This power estimation replaced the MS method used in the MAP method discussed previously and exhibited better speech enhancement result.

All the estimated speech and noise periodograms via all the introduced novel methods are used to construct a Wiener filter to enhance the noisy speech frames. In the classic Wiener filter formula, the noisy speech periodogram is considered as the sum of clean

speech periodogram and noise periodogram. Since the estimated speech and noise periodograms are not accurate, this consideration can lead to some residual noise in the enhanced speech. Hence, we introduced a new variable in the denominator of the Wiener filter to suppress the errors resulted from the inaccuracy of periodogram estimation methods. This new Wiener filter formula exhibited relatively better enhancement results with respect to its classic version.

The MAP estimation method discussed in section 4.3 was considered as the reference of the comparison of the other introduced enhancement methods such as improved explicit MAP in section 4.4, Gamma power estimation in section 4.5 and an improved Wiener filter in section 4.6. In this way we could observe the effect of the new introduced methods or the improvements applied on the previous algorithms in terms of enhancement performance.

As a future work the simultaneous use of all the mentioned algorithms can be considered to benefit from the introduced improvements of all of them together.

In the experiments we discussed the use of the enhancement algorithms on noisy speech containing just one noise type. We have also done a test in which a combination of Babble and White noises with the same power are considered as the noise but in the MAP estimation part the combination of speech PDF and one noise PDF is considered. The presence of multiple noises in the noisy speech can highly affect the performance of the mentioned algorithms. In the case of having for example two noises with almost the same power, the MAP estimation can use the speech GMM with the noise GMM that contains the closest mean vectors (centroid periodograms) to the periodogram of the sum of the two noise types. In this case the performance will definitely degrade from the case of one noise but still we can perform the enhancement procedure for the noisy speech. Such case can happen when we are dealing with a noise type whose model does not exist in our collection of noise GMMs. Furthermore the GMM that exhibits

periodograms (mean vectors) that are the closest to the periodogram of the new noise type will be used. This is the strong point of the mentioned MAP algorithm in which the lack of some models will not stop the algorithm from enhancing the noisy file although adding more GMMs of a variety of environmental noise will result in better performance of the algorithm.

Most of the discussed algorithms were implemented in a manner to be of low mathematical complexity and hence of use in real-time and online speech enhancement. The processing times for the introduced algorithms are reported in the test sections.

A method introduced in which we can come up with a reasonable number of Gaussians for the GMMs which is just a smart guess and is not accurate. Further work can be done on coming up with more accurate criteria while modelling high dimensional spaces. For high dimensional spaces in some dimensions the two different Gaussian might look quite similar and hence it will be critical to distinguish between them. The dimensionality of the space can be reduced using Mel frequency analysis and for example the 257 dimensional vector space can be mapped to a 12 dimensional space and in this way it will be easier to distinguish between the two Gaussians and to come up with a more optimum number of Gaussians for the model.

Another improvement on the introduced methods will be considering a case when we show a noisy speech as the sum of clean speech and two or more noises (with different levels). In this way we can have more variety of noise GMMs with respect to the single noise case (since the combination of GMMs can create a new GMM which can represent a new noise type). To have better estimations, we can incorporate more than one noise PDF in the MAP criterion and this can become really complicated mathematically and hence we should incorporate proper simplifications to make it possible to implement. This method can lead to more accurate periodogram estimates for speech and noise.

We can also come up with other variations of parametric Wiener filters in which a power is applied to the numerator and denominator. We can also use the estimated periodograms with other enhancement methods such as spectral subtraction and etc. to find the optimum one in terms of the quality and intelligibility of the resulted enhanced speech.

Some more work can also be done on replacing the used MAP criteria on the PDFs of speech and noise periodograms with some other Bayesian criteria such as ML or MMSE to find the most accurate estimates of the clean speech periodogram.

We have always used the periodogram as the feature to be estimated and enhanced. We can also concentrate on other features of speech and noise signals such as spectral amplitude, LPC and so on. The GMMs can be created upon these new features and then be used to estimate those of clean speech and noise through the corresponding feature observed from the noisy speech.

It has always been considered that human ears are not sensitive to the speech phase and hence the enhanced spectral amplitude was combined with the phase of noisy speech and reconstructed as the enhanced speech. We can do some research on phase enhancement and its effect on the quality and intelligibility of the enhanced speech.

Appendix A

All the multiplications and divisions in the mentioned equations are element-wise. In this way for two vectors as:

$$\boldsymbol{S} = S(\omega) = \{S_0, S_2, \dots, S_\Omega\} \ , \omega \in [0, \Omega]$$

$$\boldsymbol{N} = N(\omega) = \{N_0, N_2, \dots, N_\Omega\} \ , \omega \in [0, \Omega]$$

For the element-wise multiplication of these vectors we have:

$$\boldsymbol{SN} = S(\omega)N(\omega) = \{S_0N_0, S_1N_1, \dots, S_\Omega N_\Omega\}$$

For the element-wise division of these vectors we have:

$$\frac{\boldsymbol{S}}{\boldsymbol{N}} = \frac{S(\omega)}{N(\omega)} = \left\{\frac{S_0}{N_0}, \frac{S_1}{N_1}, \dots, \frac{S_\Omega}{N_\Omega}\right\}$$

Appendix B

Some sample noisy speeches and their clean and enhanced versions using the “Simple MAP” method for different noise types and different input SNRs are shown in Figure B.1 to Figure B.6.

Some sample noisy speeches and their clean and enhanced versions using the “MAP Amplitude” method for different noise types and different input SNRs are shown in Figure B.7 to Figure B.12.

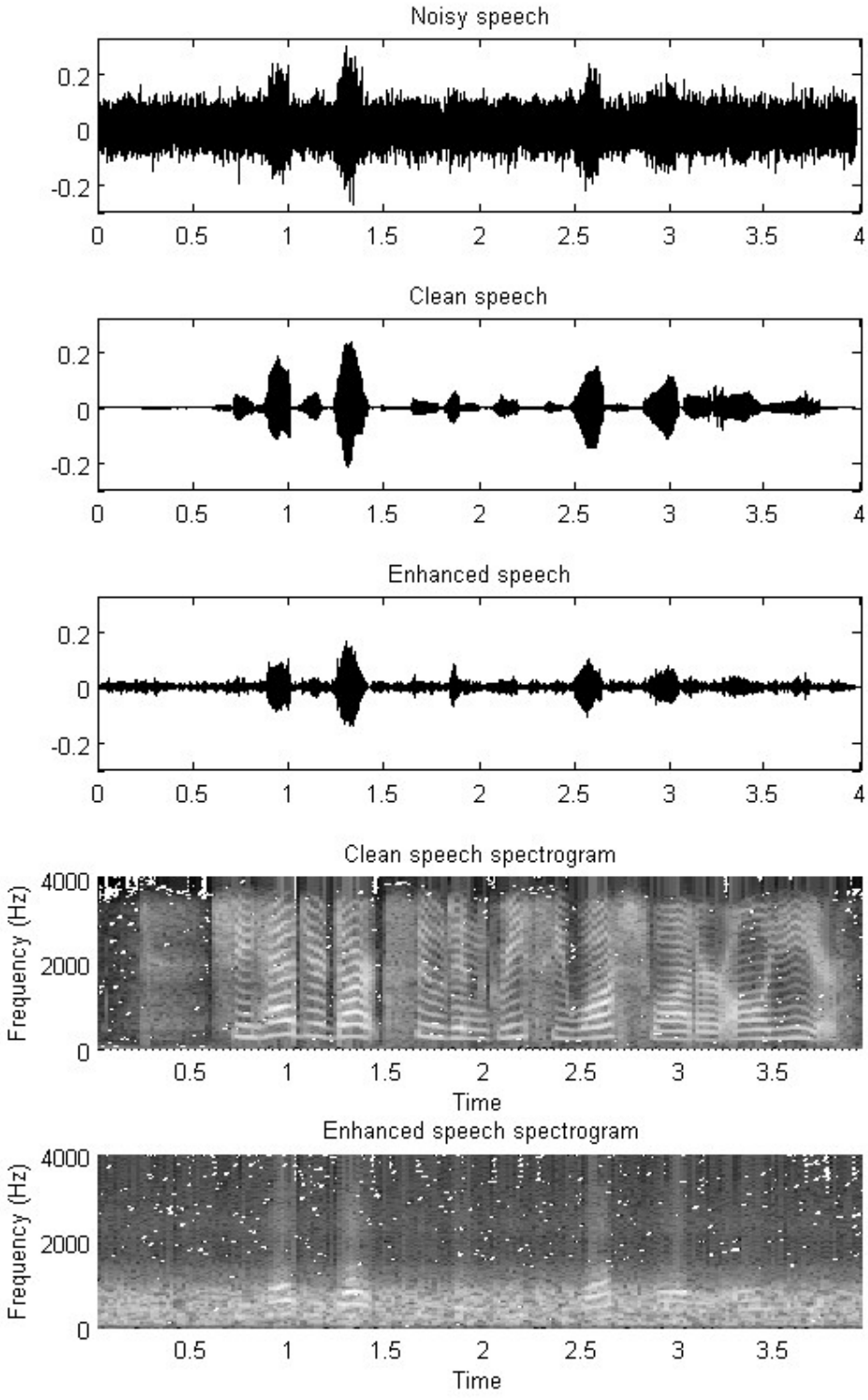

Figure B.1: Noisy speech with White noise and the input SNR of -5dB

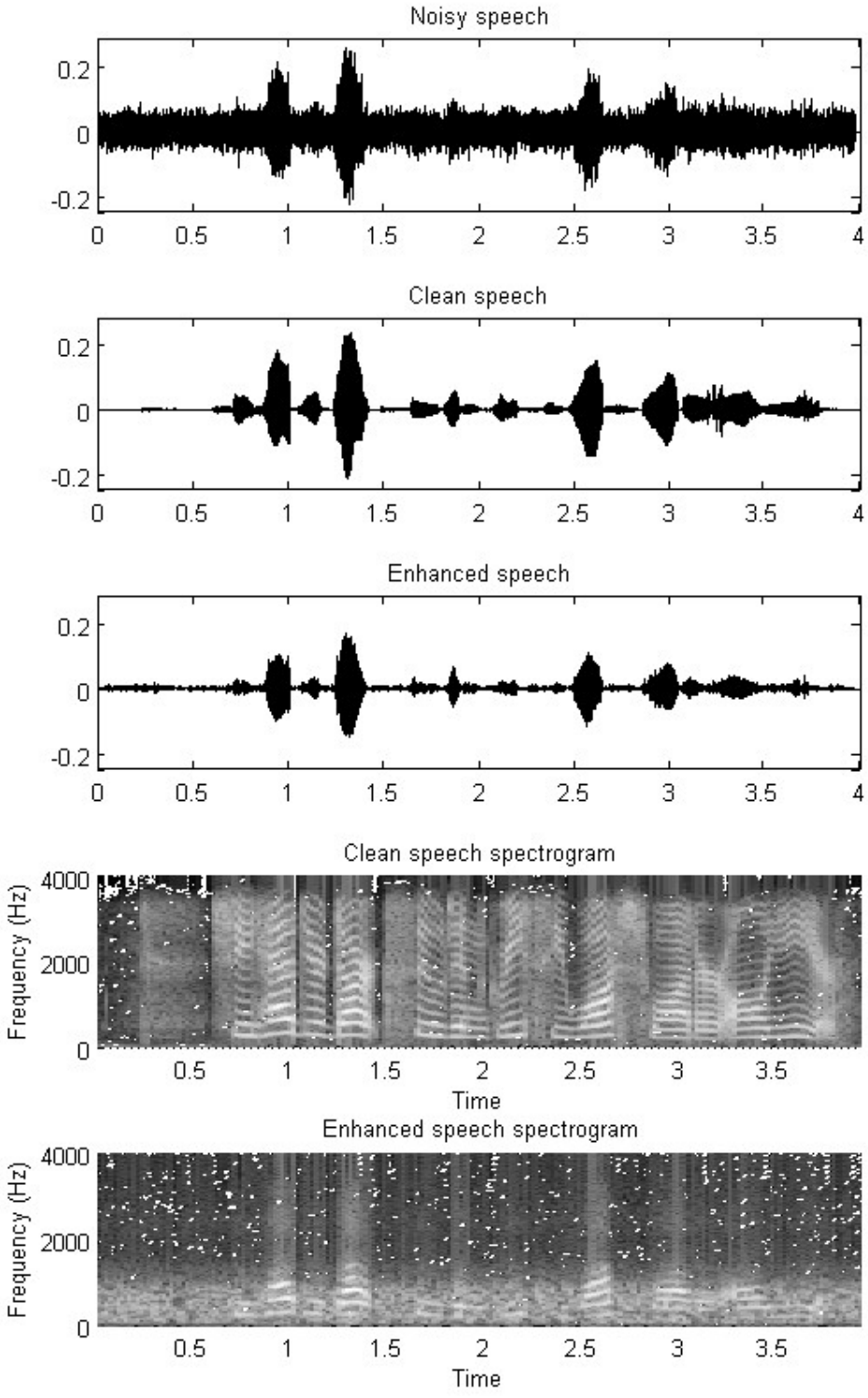

Figure B.2: Noisy speech with White noise and the input SNR of 0dB

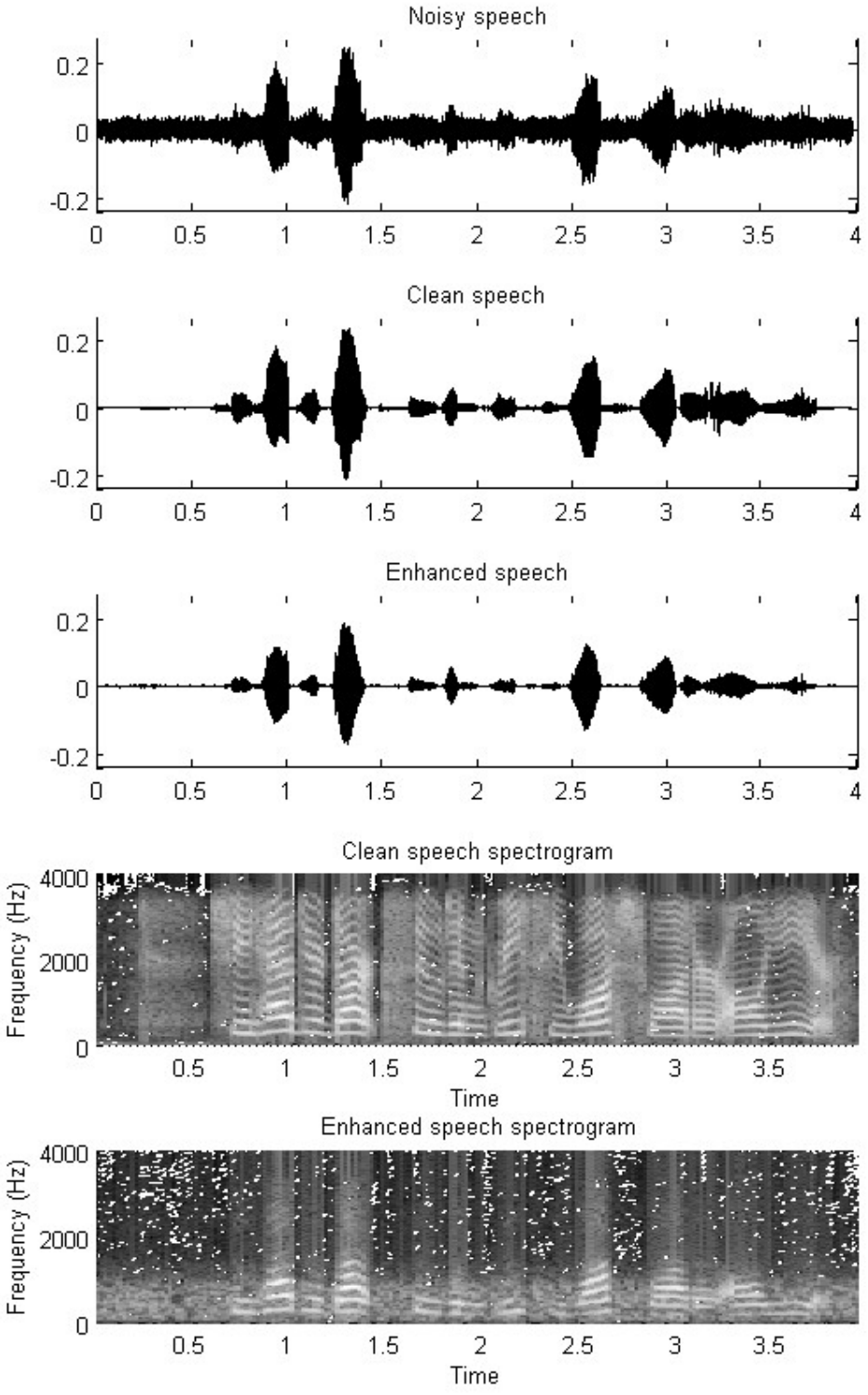

Figure B.3: Noisy speech with White noise and the input SNR of 5dB

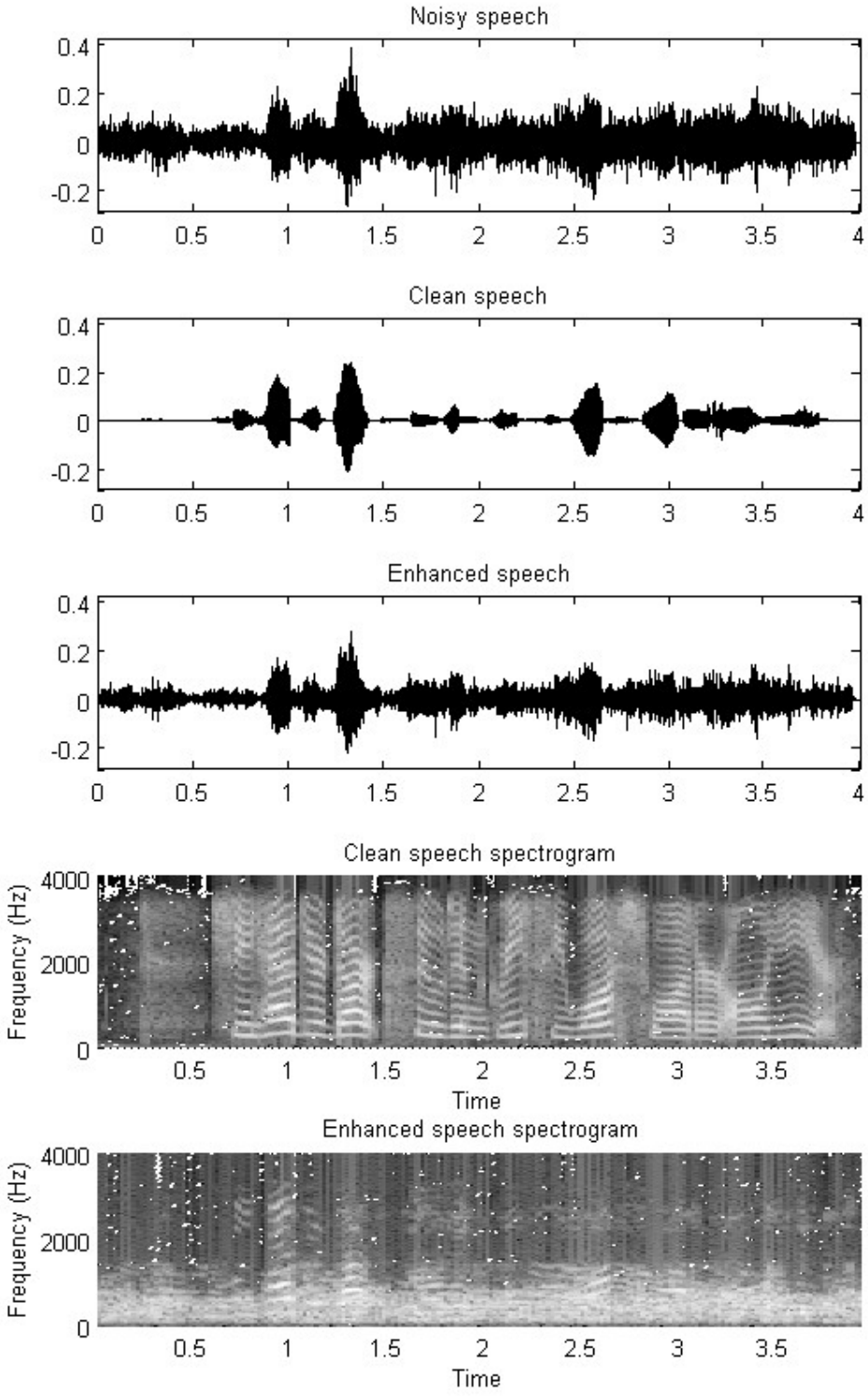

Figure B.4: Noisy speech with Babble noise and the input SNR of -5dB

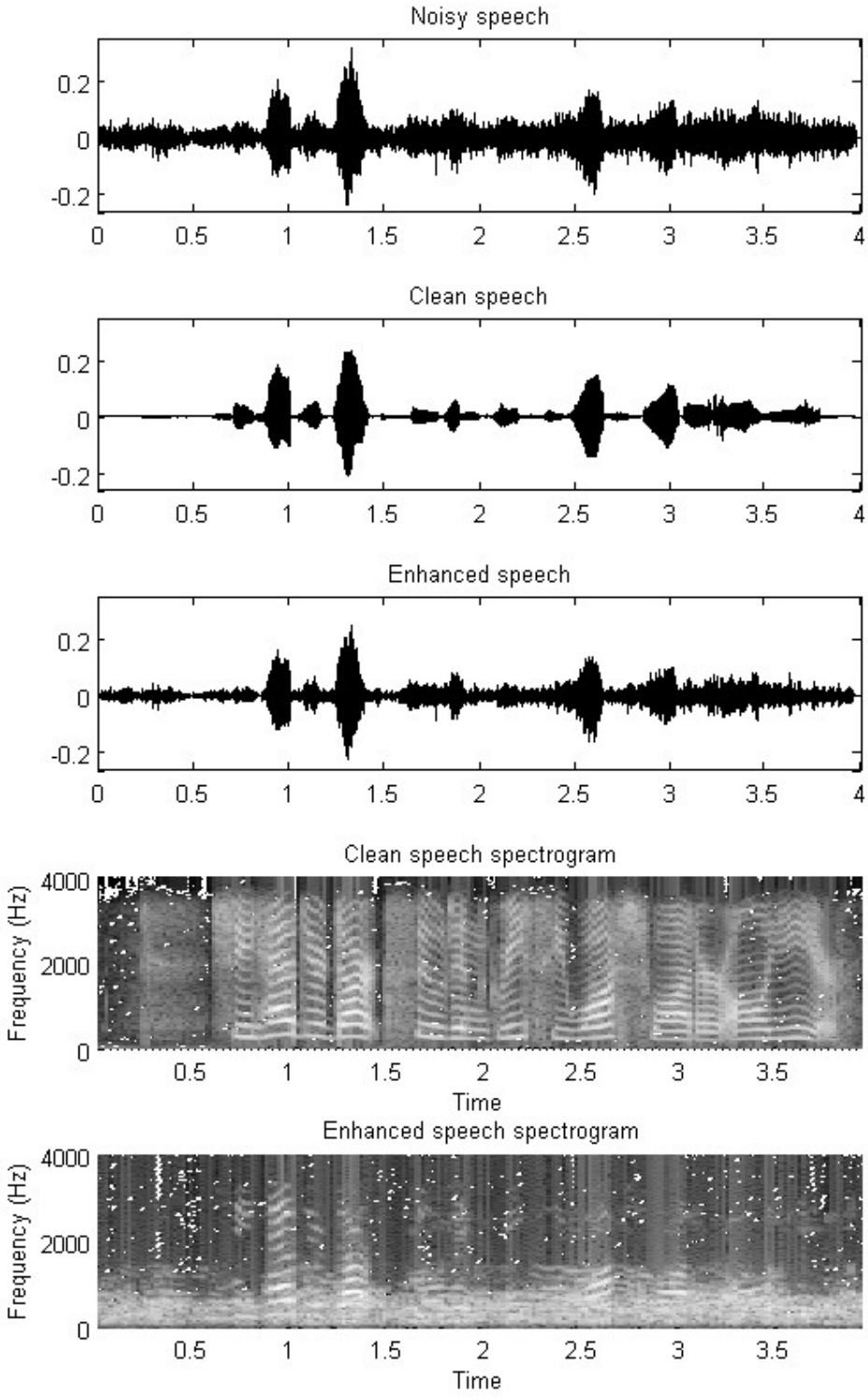

Figure B.5: Noisy speech with Babble noise and the input SNR of 0dB

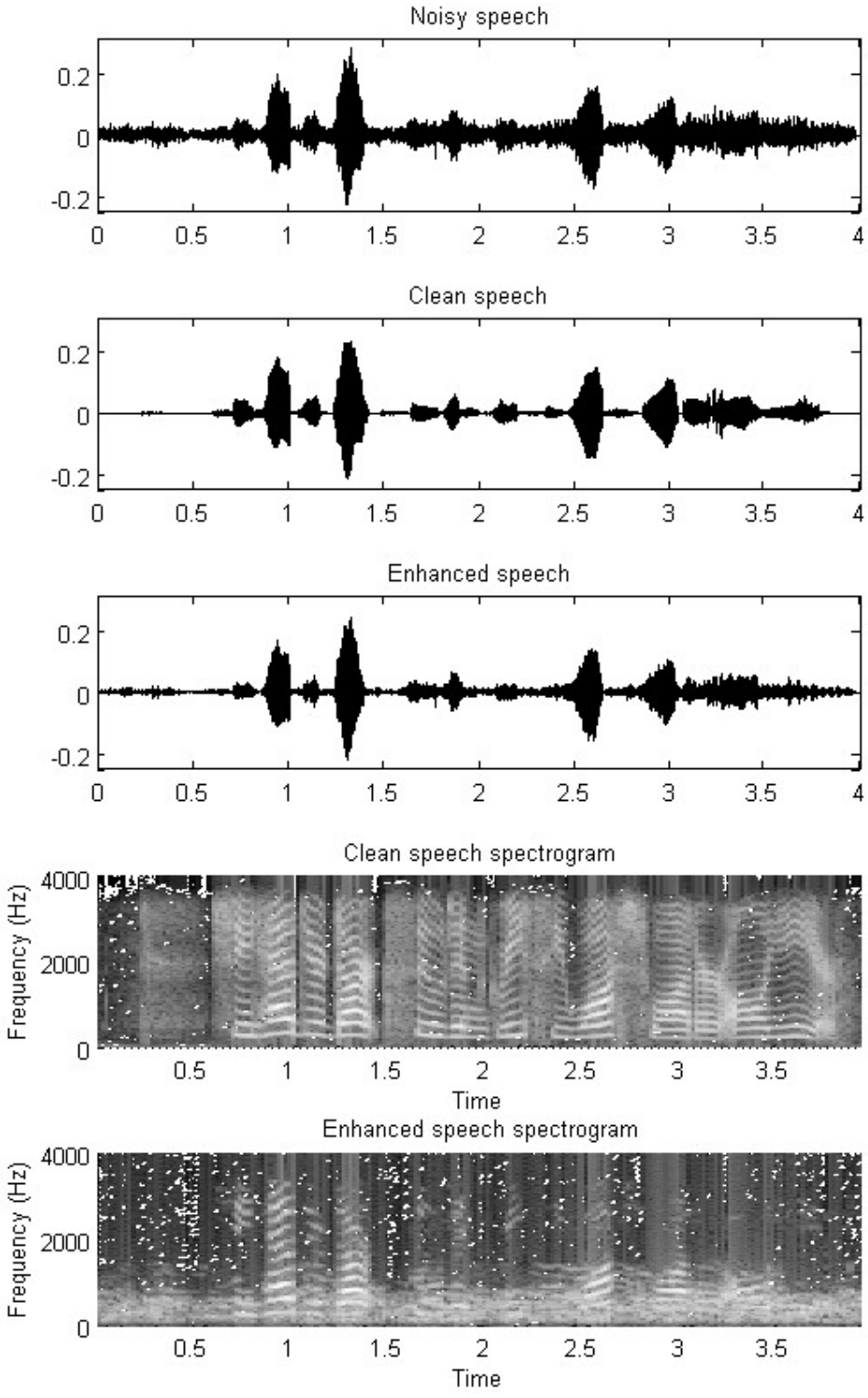

Figure B.6: Noisy speech with Babble noise and the input SNR of 5dB

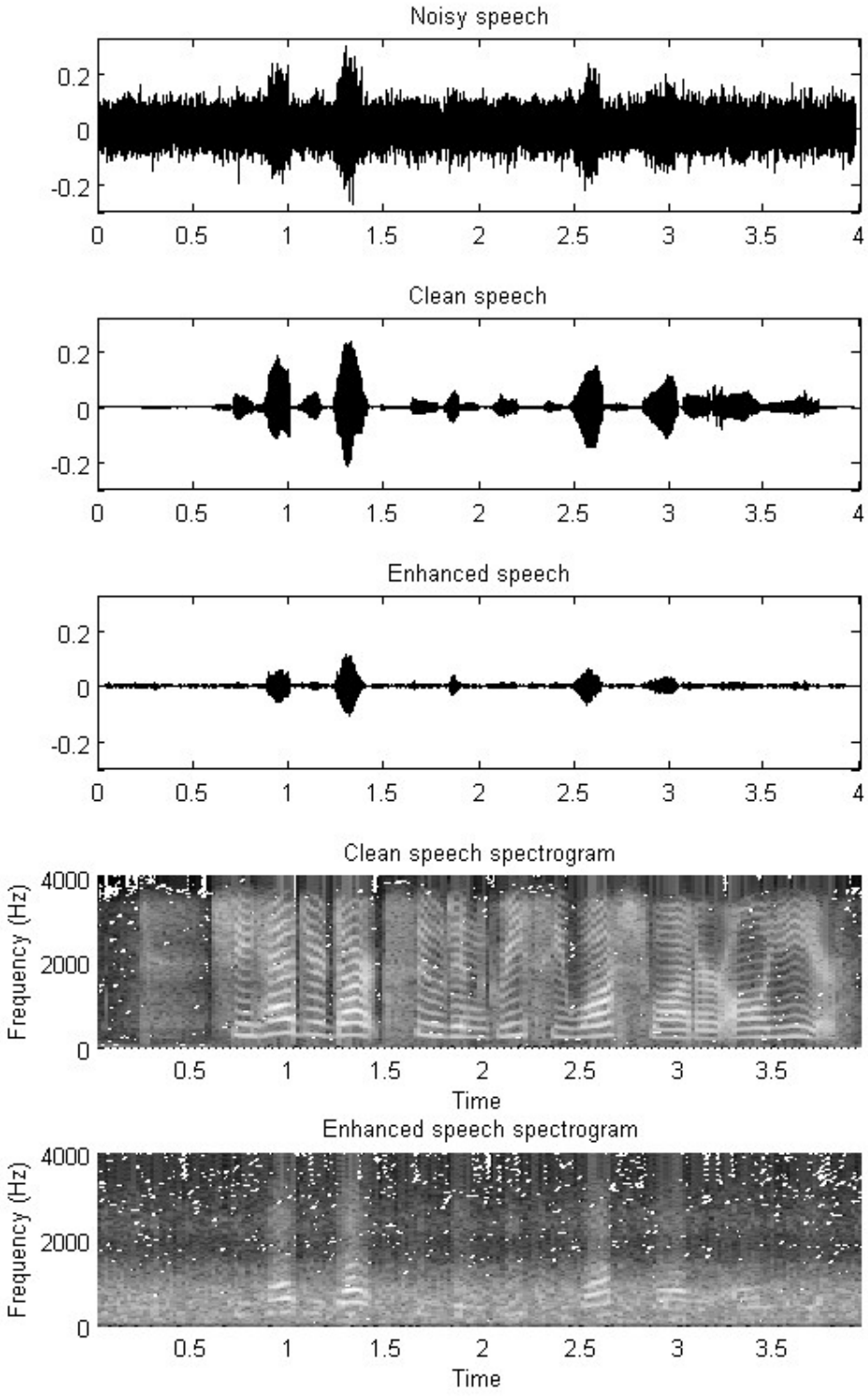

Figure B.7: Noisy speech with White noise and the input SNR of -5dB

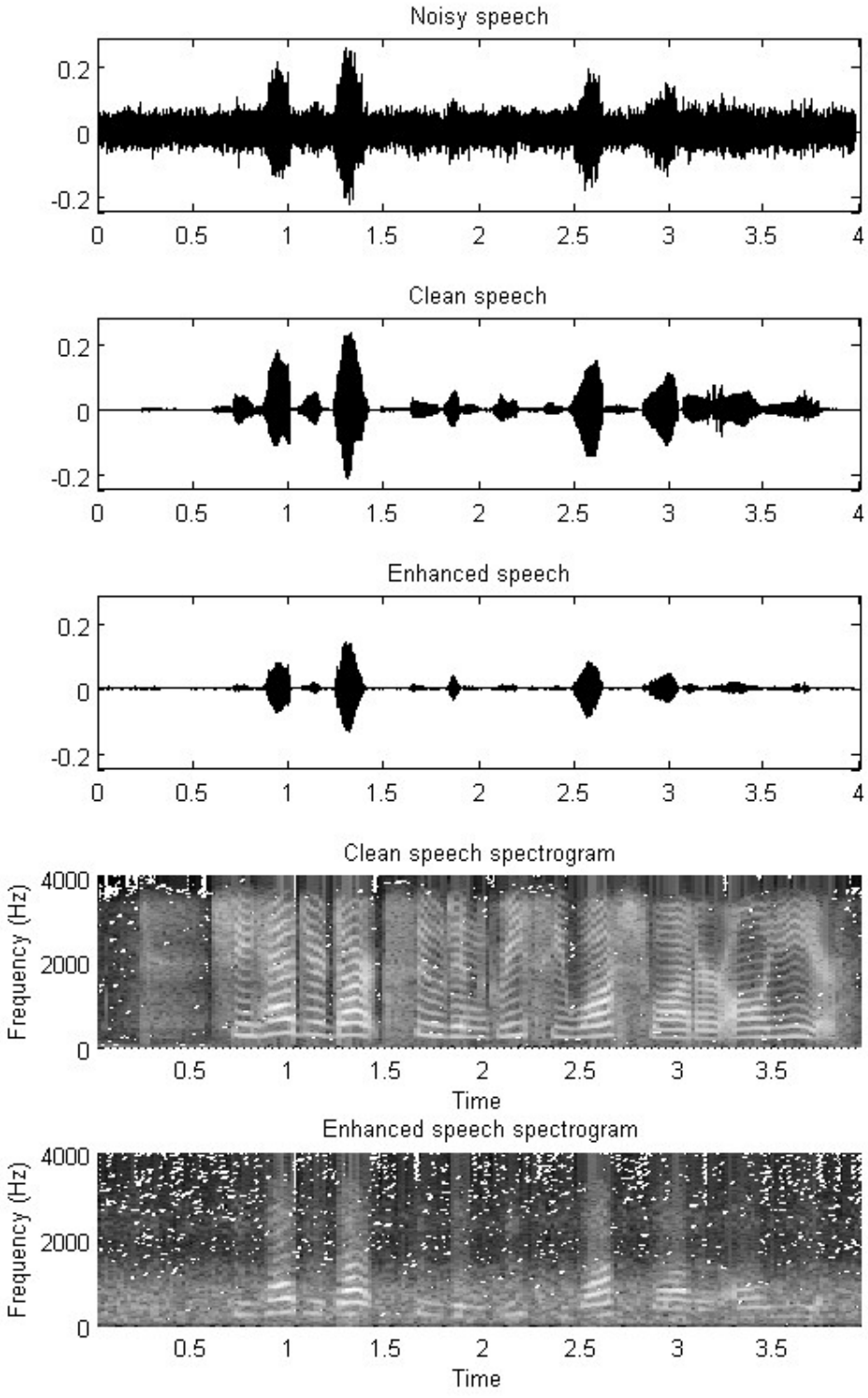

Figure B.8: Noisy speech with White noise and the input SNR of 0dB

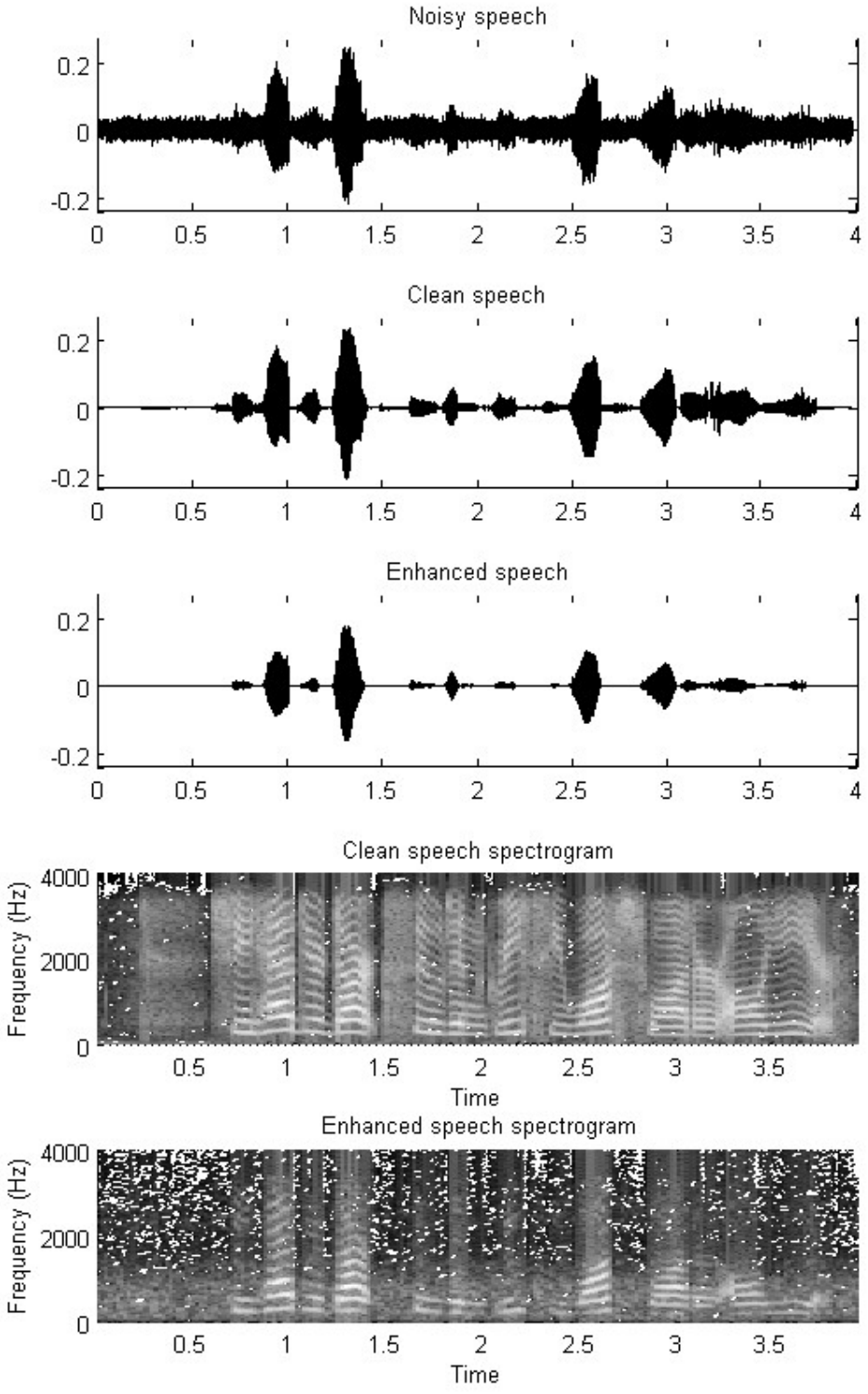

Figure B.9: Noisy speech with White noise and the input SNR of 5dB

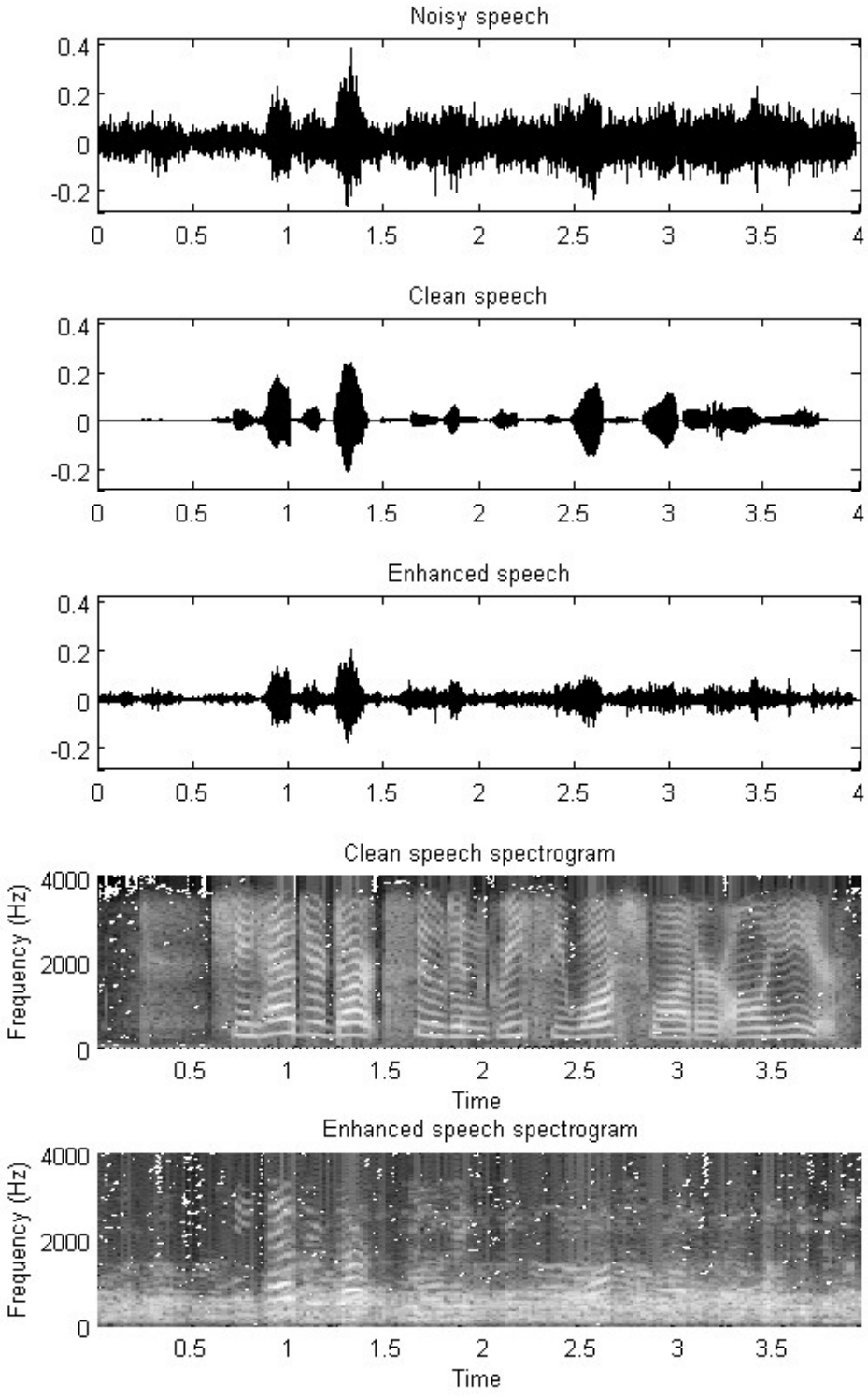

Figure B.10: Noisy speech with Babble noise and the input SNR of -5dB

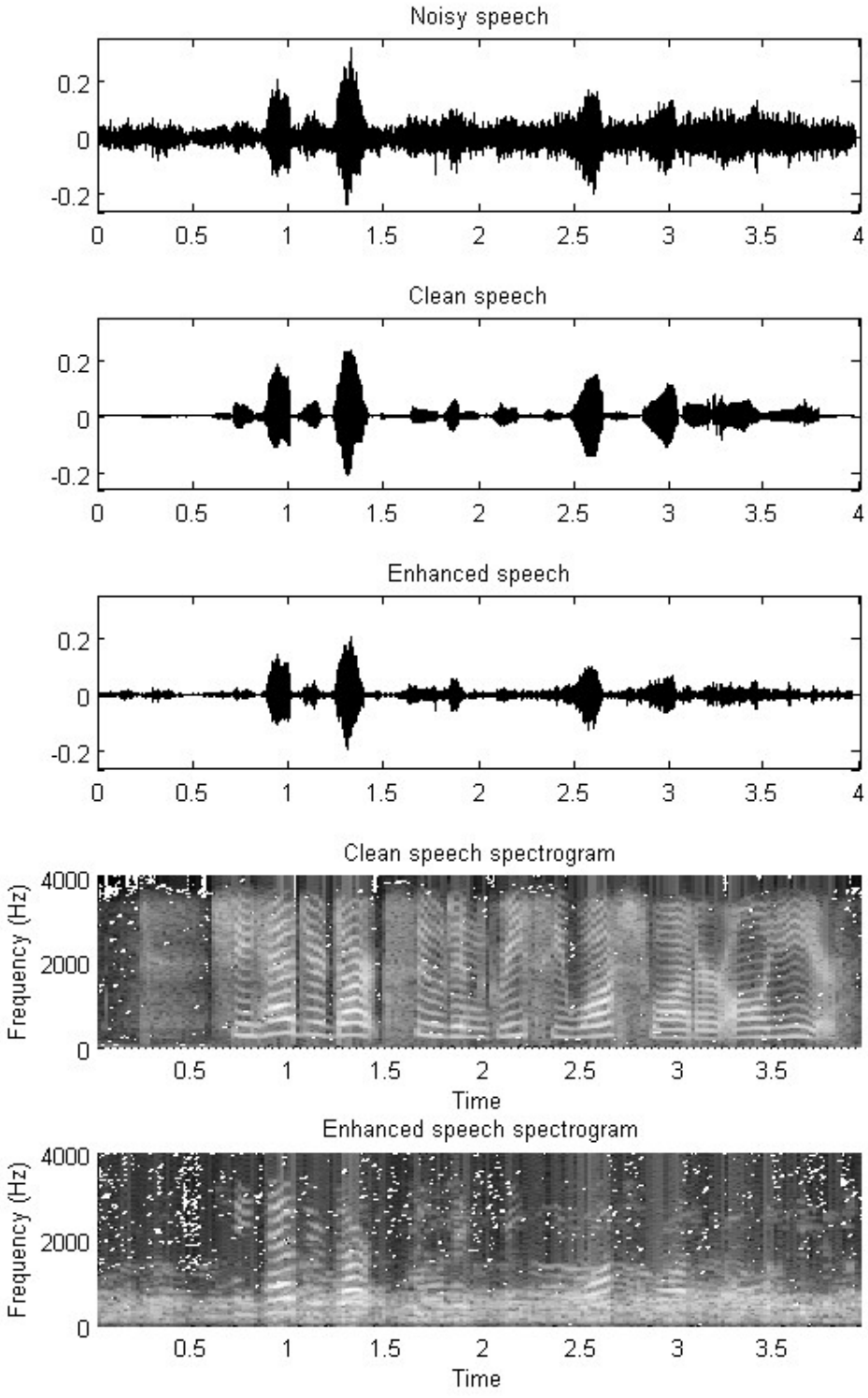

Figure B.11: Noisy speech with Babble noise and the input SNR of 0dB

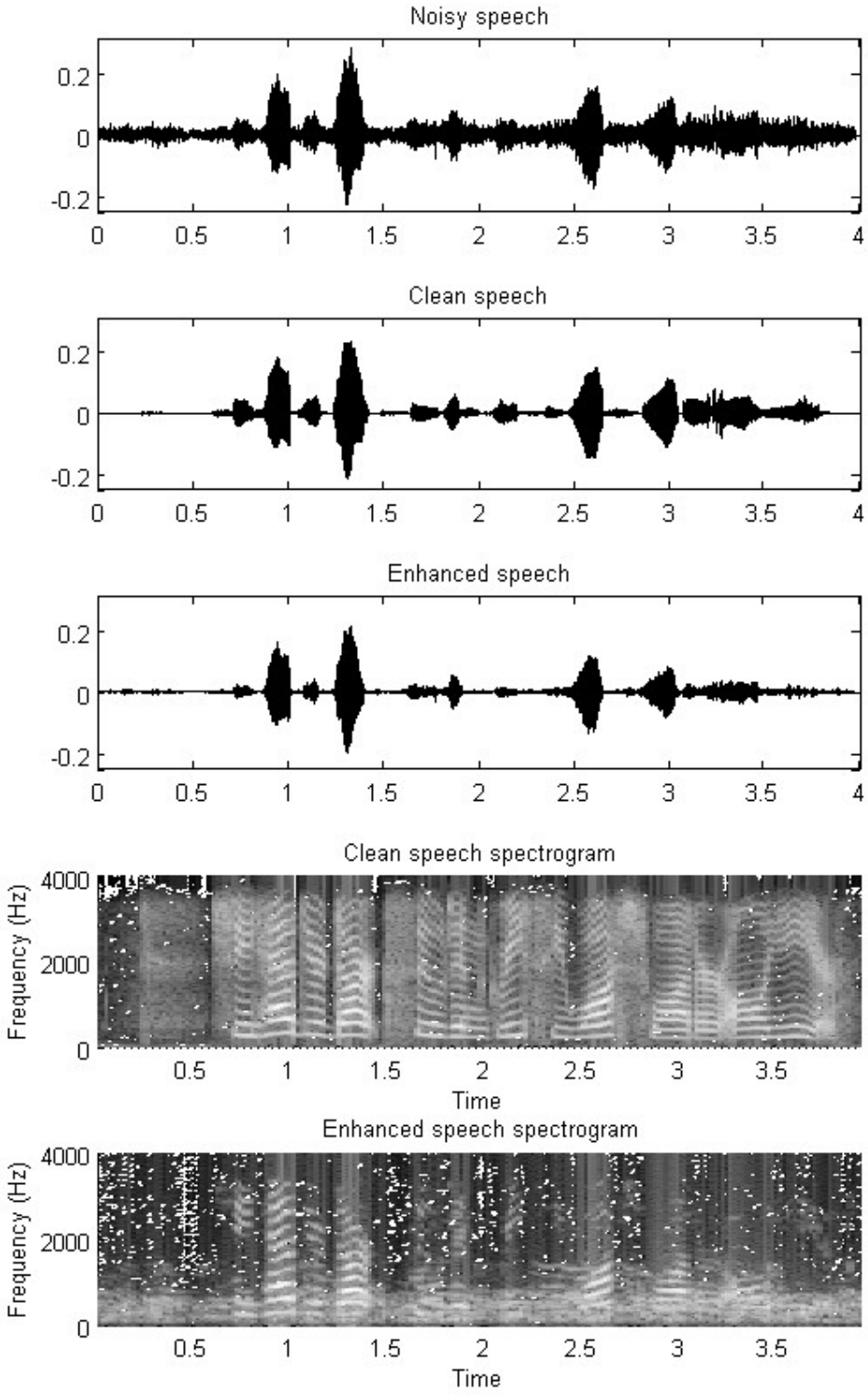

Figure B.12: Noisy speech with Babble noise and the input SNR of 5dB

References

[1] S. Srinivasan, R. Aichner, W. B. Kleijn, and W. Kellermann, "Multichannel parametric speech enhancement," *IEEE SIGNAL PROCESSING LETTERS,* vol. 13, pp. 304-307, May 2006.

[2] J. Meyer and K. Simmer, "Multi-channel speech enhancement in a car environment using Wiener filtering and spectral subtraction," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Munich, Apr. 1997, pp. 1167-117.

[3] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and Unsupervised Speech Enhancement Using Nonnegative Matrix Factorization," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 21, pp. 2140-2151, 2013.

[4] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 27, pp. 113–120, 1979.

[5] B. L. Sim, Y. C. Tong, J. S. Chang, and C. T. Tan, "A parametric formulation of the generalized spectral subtraction method," *IEEE Transaction on Speech and Audio Processing,* vol. 6, pp. 328 - 337, 1998.

[6] Y. Lu and P. C. Loizou, "A geometric approach to spectral subtraction," *Speech Communication,* vol. 50, pp. 453–466, 2008.

[7] B. Widrow, J. G. R. G. Jr., and J. M. Mccool, "Adaptive noise cancelling: principles and applications," *Proceedings of The IEEE,* vol. 63, pp. 1692–1716, 1975.

[8] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean square error short time amplitude estimator," *IEEE Transaction on Acoustic, Speech and Signal Processing,* vol. 32, pp. 1109-1121, Dec. 1984.

[9] P. J. Wolfe and S. J. Godsil, "Efficient alternatives to the Ephraim and Malah suppression rule for audio signal enhancement," *Eurasip Journal on Applied Signal Processing,* vol. 2003, pp. 1043-1051, 2003.

[10] O. Cappe, "Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor," *IEEE Transaction on Speech and Audio Processing,* vol. 2, pp. 345-349, 1994.

[11] S. Suhadi, C. Last, and T. Fingscheidt, "A Data-Driven Approach to A Priori SNR Estimation," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 19, pp. 186-195, 2011.

[12] T. V. Sreenivas and P. Kirnapure, "Codebook Constrained Wiener Filtering for Speech Enhancement," *IEEE Transaction on Speech and Audio Processing,* vol. 4, pp. 383-389, Sep 1996.

[13] S. Srinivasan, J. Samuelsson, and W. B. Kleijn, "Codebook driven short-term predictor parameter estimation for speech enhancement," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 14, pp. 163-176, 2006.

[14] Y. Ephraim, "A Bayesian estimation approach for speech enhancement using hidden Markov models," *IEEE Transactions on Signal Processing,* vol. 40, pp. 725-735, 1992.

[15] H. Sameti, H. Sheikhzadeh, L. Deng, and R. L. Bernan, "HMM-based strategies for enhancement of speech signals embedded in nonstationary noise," *IEEE Transactions on Speech and Audio Processing,* vol. 6, pp. 445-455, 1998.

[16] D. Y. Zhao and W. B. Kleijn, "HMM-Based Gain Modeling for Enhancement of Speech in Noise," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 15, pp. 882-892, 2007.

[17] N. Mohammadiha, R. Martin, and A. Leijon, "Spectral domain speech enhancement using HMM state-dependent super-Gaussian priors," *IEEE SIGNAL PROCESSING LETTERS,* vol. 20, pp. 253-256, 2013.

[18] N. Mohammadiha and A. Leijon, "Nonnegative HMM for Babble Noise Derived From Speech HMM: Application to Speech Enhancement," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 21, pp. 998-1011, 2013.

[19] H. Veisi and H. Sameti, "Speech enhancement using hidden Markov models in Mel-frequency domain," *Speech Communication,* vol. 55, pp. 205-220, 2013.

[20] J. Hao, T. Lee, and T. J. Sejnowski, "Speech Enhancement Using Gaussian Scale Mixture Models," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 18, pp. 1127-1136, Aug. 2010.

[21] J. Hao, H. Attias, S. Nagarajan, T.-W. Lee, and T. J. Sejnowski, "Speech Enhancement, Gain, and Noise Spectrum Adaptation Using Approximate Bayesian Estimation," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 17, pp. 24-37, 2009.

[22] B. Fodor and T. Fingscheidt, "Speech enhancement using a joint map estimator with Gaussian mixture model for (non-)stationary noise," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Prague, 2011, pp. 4768-4771.

[23] D. Burshteinand and S. Gannot, "Speech Enhancement Using a Mixture-Maximum Model," *IEEE Transaction on Speech and Audio Processing,* vol. 10, pp. 341-351, Sep. 2002.

[24] T. Lotter and P. Vary, "Speech enhancement by MAP spectral amplitude estimation using a super-Gaussian speech model," *Eurasip Journal on Applied Signal Processing,* vol. 2005, pp. 1110-1126, 2005.

[25] R. Martin, "Spectral Subtraction Based on Minimum Statistics," in *European Signal Processing Conference (EUSIPCO)*, Edinburgh, 1994, pp. 1182-1185.

[26] D. Wang and J. Lim, "The Unimportance of Phase in Speech Enhancement," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 30, pp. 679-681, 1983.

[27] P. Loizou, *Speech Enhancement: Theory and Practice*: CRC, 2007.

[28] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *nternational Conference on Acoustics, Speech and Signal Processing ICASSP)*, 1979, pp. 208–211.

[29] B. Sim, Y. Tong, J. Chang, and C. Tan, "A Parametric Formulation of the Generalized Spectral Subtraction Method," *IEEE Transactions on Speech and Audio Processing,* vol. 6, pp. 328–337, 1998.

[30] Y. Hu and P. Loizou, "Subjective Comparison of Speech Enhancement Algorithms," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006, pp. 153–156.

[31] P. Vary, "Noise Suppression by Spectral Magnitude Estimation–Mechanism and Theoretical Limits," *Signal Processing,* vol. 8, pp. 387–400, 1985.

[32] P. Scalart, "Speech Enhancement Based on a-priori Signal to Noise Estimation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1996, pp. 629-632.

[33] J. Lim and A. Oppenheim, "All-pole Modeling of Degraded Speech," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 26, pp. 197-210, 1987.

[34] Y. Ephraim, D. Malah, and B. Juang, "On the Application of Hidden Markov Models for Enhancing Noisy Speech," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 37, pp. 1846-1856, 1989.

[35] N. Hadir, F. Faubel, and D. Klakow, "A Model-Based Spectral Envelope Wiener Filter for Perceptually Motivated Speech Enhancement," in *Interspeech*, Florance, Italy, 2011.

[36] R. McAulay and M. Malpass, "Speech Enhancement using a Soft-decision Noise Suppression Filter," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 28, pp. 137-145, 1980.

[37] Y. Ephraim and D. Malah, "Speech Enhancement using a Minimum Meansquare Error Log-spectral Amplitude Estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 33, pp. 443–445, 1985.

[38] O. Capp´e, "Elimination of the Musical Noise Phenomenon with the Ephraim and Malah Noise Suppressor," *IEEE Transactions on Speech and Audio Processing,* vol. 2, pp. 345–349, 1994.

[39] G. Kim, Y. Lu, Y. Hu, and P. Loizou, "An Algorithm that Improves Speech Intelligibility in Noise for Normal-hearing Listeners," *The Journal of the Acoustical Society of America,* vol. 126, p. 1486, 2009.

[40] Y. Hu and P. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 16, pp. 229-238, Jan. 2008.

[41] Y. Hu and P. Loizou, "Techniques for Estimating the Ideal Binary Mask," in *The Eleventh International Workshop on Acoustic Echo Noise Control*, 2008.

[42] J. Jensen and R. C. Hendriks, "Spectral Magnitude Minimum Mean-square Error Binary Masks for DFT-based Speech Enhancement," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4736-4739.

[43] K. Hermus and P. Wambacq, "A Review of Signal Subspace Speech Enhancement and its Application to Noise Robust Speech Recognition," *EURASIP Journal on Advances in Signal Processing,* vol. 2007, 2006.

[44] Y. Hu and P. Loizou, "A Generalized Subspace Approach for Enhancing Speech Corrupted by Colored Noise," *IEEE Transactions on Speech and Audio Processing,* vol. 11, pp. 334-341, 2003.

[45] B. Hanson, D. Wong, and B. Juang, "Speech Enhancement with Harmonic Synthesis," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1983, pp. 1122–1125.

[46] A. Nehorai and B. Porat, "Adaptive Comb Filtering for Harmonic Signal Enhancement," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 34, pp. 1124-1138, 1986.

[47] G. Kang and L. Fransen, "Quality Improvement of LPC-processed Noisy Speech by using Spectral Subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 37, pp. 939–942, 1989.

[48] G. Guilmin, R. Bouquin-Jeannes, and P. Gournay, "Study of the Influence of Noise Preprocessing on the Performance of a Low Bit Rate Parametric Speech Coder," *Proceedings of Eurospeech,* vol. 3, pp. 2367–2370, 1999.

[49] E. Zavarehei, S. Vaseghi, and Q. Yan, "Noisy Speech Enhancement using Harmonic-Noise Model and Codebook-based Post-Processing," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, pp. 1194–1203, 2007.

[50] Y. Stylianou, "Applying the Harmonic plus Noise Model in Concatenative Speech Synthesis," *IEEE Transactions on Speech and Audio Processing,* vol. 9, pp. 21–29, 2001.

[51] M. Krini and G. Schmidt, "Model-Based Speech Enhancement for Automotive Applications," in *International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2009, pp. 632–637.

[52] J. Jensen and J. Hansen, "Speech Enhancement using a Constrained Iterative Sinusoidal Model," *IEEE Transactions on Speech and Audio Processing,* vol. 9, pp. 731-740, 2001.

[53] G. Moharir, P. Patwardhan, and P. Rao, "Spectral Enhancement Preprocessing for the HNM Coding of Noisy Speech," presented at the Seventh International Conference on Spoken Language Processing, 2002.

[54] R. Chen, C. Chan, and H. So, "Model-Based Speech Enhancement With Improved Spectral Envelope Estimation via Dynamics Tracking," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, pp. 1324–1336, 2012.

[55] C. Breithaupt and R. Martin, "Voice activity detection in the DFT domain based on a parametric noise model," presented at the International Workshop on Acoustic Echo and Noise Control (IWAENC), Paris, 2006.

[56] N. Virag, "Single Channel Speech Enhancement Based on Masking Properties of the Human Auditory System," *IEEE Transactions on Speech and Audio Processing,* vol. 7, pp. 126-137, 1999.

[57] R. Martin, D. Malah, R. V. Cox, and A. J. Accardi, "A Noise Reduction Preprocessor for Mobile Voice Communication," *EURASIP Journal on Applied Signal Processing* vol. 2004, pp. 1046–1058., 2004.

[58] N. Moayeri, "Some Issues Related to Fixed-Rate Pruned Tree-Structured Vector Quantizers," *IEEE Transactgions on Information theory,* vol. 41, pp. 1523-1531, 1995.

[59] S. Chehresa and M. H. Savoji, "Codebook Constrained Iterative and Parametric Wiener Filter Speech Enhancement," in *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Kuala Lumpur, Malaysia, Nov. 2009, pp. 548-553.

[60] S. Chehresa and M. H. Savoji, "Improved Codebook Constrained Wiener filter Speech Enhancement," in *5th International Symposium on Telecommunications (IST)*, Tehran, Iran, Dec. 2010, pp. 614-618.
[61] S. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*, 4th ed. Chichester, West Sussex: Wiley, 2008.
[62] S. Chehresa and M. H. Savoji, "MMSE speech enhancement using GMM," in *16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, Shiraz, Iran, May 2012, pp. 266-271.
[63] S. Chehresa and M. H. Savoji, "Speech enhancement based on Gaussian Mixture Modeling and Wiener filtering," *International Journal on Communications Antenna and Propagation (I.Re.C.A.P),* vol. 2, pp. 111-122, Apr. 2012.
[64] S. Chehresa and M. H. Savoji, "MMSE speech enhancement based on GMM and solving an over-determined system of equations," in *IEEE 7th International Symposium on Intelligent Signal Processing (WISP)*, Floriana, Malta, Sep. 2011, pp. 1-5.
[65] I. Potamitis, N. Fakotakis, N. Liolios, and G. K. Kokkinakis, "Speech Enhancement Using Mixtures of Gaussians for Speech and Noise," in *5th Text, Speech and Dialogue Conference*, Czech Republic, 2002, pp. 337-340.
[66] S. Chehrehsa and M. H. Savoji, "Speech Enhancement Using Gaussian Mixture Models, Explicit Bayesian Estimation and Wiener Filtering," *submited for publication in Iranian Journal of Electrical & Electronic Engineering (IJEEE),* 2014.
[67] S. Chehrehsa and M. H. Savoji, "Speech Enhancement Using Gaussian Mixture Models, Explicit Bayesian Estimation and Wiener Filtering," *Iranian Journal of Electrical & Electronic Engineering (IJEEE),* vol. 10, pp. 168-175, 2014.
[68] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis and density estimation," *Journal of the American Statistical Association,* vol. 97, pp. 611-631, 2002.
[69] A. Dasgupta and A. E. Raftery, "Detecting features in spatial point processes with clutter via model-based clustering," *Journal of the American Statistical Association,* vol. 93, pp. 294–302, 1998.
[70] K. L. Nylund, T. Asparouhov, and B. O. Muthén, "Deciding on the Number of Classes in Latent Class Analysis and Growth Mixture Modeling: A Monte Carlo Simulation Study," *Structural Equation Modeling: A Multidisciplinary Journal,* vol. 14, pp. 535-569, 2007.
[71] R. Martin, "Noise Power Spectral Density Estimation Based on Optimal Smoothing and Minimum Statistics," *IEEE Transactions on Speech and Audio Processing,* vol. 9, pp. 504-512, 2001.
[72] M. J. Beal and Z. Ghahramani, *Variational inference for Bayesian mixture of factor analysers*: Oxford University Press, 2003.
[73] S. Chehrehsa and T. J. Moir, "Speech Enhancement Using Maximum A-Posteriori and Gaussian Mixture Models," *Computer Speech & Language,* vol. 36, pp. 58-71, March 2016.
[74] J.-H. Chang, "Noisy speech enhancement based on improved minimum statistics incorporating acoustic environment-awareness," *Digital Signal Processing,* vol. 23, pp. 1233-1238, 2013.
[75] T. Grekmann and R. C. Hendriks, "Unbiased MMSE-Based Noise Power Estimation With Low Complexity and Low Tracking Delay," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 20, pp. 1383-1393, 2012.
[76] I. J. Myung, "Tutorial on maximum likelihood estimation," *Journal of Mathematical Psychology,* vol. 47, pp. 90-100, 2003.
[77] D. P. Doane and L. E. Seward, "Measuring Skewness: A Forgotten Statistic?," *Journal of Statistics Education,* vol. 19, 2011.
[78] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, 5th ed.: Academic Press, 2014.
[79] T. R. University. (1995). *Signal Processing Information Base (SPIB), Noise data*. Available: http://spib.rice.edu/spib/select_noise.html
[80] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, *et al.* (1993). *TIMIT Acoustic-phonetic continuous speech corpus* Available: www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1

[81] S. M. Kim and H. K. Kim, "Noise variance estimation based on dual-channel phase difference for speech enhancement," *Digital Signal Processing,* vol. 26, pp. 169-182, 2014.

[82] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transaction on Audio, Speech and Language Processing,* vol. 14, pp. 1462-1469, 2006.

[83] P. Mowlaee, R. Saeidi, M. G. Christensen, and R. Martin, "Subjective and objective quality assessment of single-channel speech separation algorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 69-72.

[84] N. Mohammadiha. *Matlab Code for "Supervised and Unsupervised Speech Enhancement Using Nonnegative Matrix Factorization"*. Available: http://www.sigproc.uni-oldenburg.de/63331.html