

Comparison of Single Channel Blind Dereverberation Methods for Speech
Signals

by
Deha Deniz Türköz

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
June 2016


Comparison of Single Channel Blind Dereverberation Methods for Speech Signals

APPROVED BY

Assoc. Prof. Dr. Hakan Erdoğan
(Thesis Supervisor)



Prof. Dr. Özgür Erçetin



Assoc. Prof. Dr. İlker Bayram



DATE OF APPROVAL:27.06.2016.....

© Deha Deniz Türköz 2016
All Rights Reserved

Sevgili Ailem Derya, Hakan, Irmak ve Mehdi'ye

Acknowledgments

First, I would like to thank deeply to my supervisor Hakan Erdoğan. He accepted me as his student when I was thinking about quitting my masters and feeling really depressed. He was very kind and sincere to me all the time. He acted with great patience to my simple mistakes, encouraged me to work hard and guided me with a smiling face when I felt lost in the topic. I learned a lot from him and feel really lucky to have a chance to work with such a great supervisor with great knowledge.

I would also like to express my gratitude to my M.S.c oral examination committee members Dr. Özgür Erçetin and Dr. İlker Bayram for giving their precious time to attend my M.Sc thesis presentation. They kindly read my thesis and give their valuable comments.

Additionally, I would like to thank my dad and mom who gave me the opportunity to get a proper education. I am grateful to them since they are always there to support me whatever happens or whatever I decide to do. I feel secure all the time because of knowing that I have such a wonderful parents.

At the end, I would like to express my special thanks to Mehdi for being such a wonderful person. He stayed awake till mornings to support me mentally and was always there for me to help. He fixed my thesis and made it meaningful as my life.

I also want to thank ADP for accepting me as their teaching assistant. Without their financial support, I could not come this far.

Comparison of Single Channel Blind Dereverberation Methods for Speech Signals

Deha Deniz Türköz

EE, M.Sc. Thesis, 2016

Thesis Supervisor: Hakan Erdoğan

Keywords: single channel, blind dereverberation, weighted prediction error (WPE), room impulse response(RIR), delayed linear prediction (DLP), model based signal processing, sparsity, weighted prediction (WP)

Abstract

Reverberation is an effect caused by echoes from objects when an audio wave travels from an audio source to a listener. This channel effect can be modeled by a finite impulse response filter which is called a room impulse response (RIR) in case of speech recordings in a room. Reverberation especially with a long filter causes high degradation in recorded speech signals and may affect applications such as Automatic Speech Recognition (ASR), hands-free teleconferencing and many others significantly. It may even cause ASR performance to decrease even in a system trained using a database with reverberated speech. If the reverberation environment is known, the echoes can be removed using simple methods. However, in most of the cases, it is unknown and the process needs to be done blind, without knowing the reverberation environment. In the literature, this problem is called the blind dereverberation problem. Although, there are several methods proposed to solve the blind dereverberation problem, due to the difficulty caused by not knowing the signal and the filter, the echoes are hard to remove completely from speech signals. This thesis aims to compare some of these existing methods such as Laplacian based weighted prediction error (L-WPE), Gaussian weighted prediction error (G-WPE), NMF based temporal spectral modeling (NMF+N-

CTF), delayed linear prediction (DLP) and proposes a new method that we call sparsity penalized weighted least squares (SPWLS). In our experiments, we obtained the best results with L-WPE followed by G-WPE methods, whereas the new SPWLS method initialized with G-WPE method obtained slightly better signal-to-noise ratio and perceptual quality values when the room impulse responses are long.

Comparison of Single Channel Blind Dereverberation Methods for Speech Signals
Tek Kanallı Ses Sinyallerinin Ekodan Arındırma Yöntemlerinin Karşılaştırması

Deha Deniz Türköz

EE, Yüksek Lisans Tezi, 2016

Tez Danışmanı: Hakan Erdoğan

Anahtar Kelimeler: Tek kanal, Yankılanmadan Arındırma, Ağırlıklı öngörü Hatası, Ertelemeli Lineer öngörü, modele dayalı sinyal işleme

Özet

Yankılanma bir ses dalgasının, ses kaynağından dinleyiciye ulaşırken etraftaki objelerden yansımaları ile oluşur. Bu kanal etkisi ya da diğer ismiyle oda dürtü cevabı (RIR), sonlu dürtü cevablı bir filtre kullanılarak modellenenir. Yankılanma, özellikle uzun bir filtreyle yankılanma, kayıt altına alınmış ses dosyalarında büyük bozulmalara sebep olmaktadır ve otomatik konuşma tanıma (OKT), dokunmasız telekonferans ve benzeri uygulamaları önemli ölçüde etkilemektedir. Hatta, OKT uygulaması, yankılanmış verilerden eğitilmiş olsa bile başarımları kaybı yaşanır. Eğer oda dürtü cevabı biliniyorsa, yankının zarar verici etkisi kolayca kaldırılabilir. Ancak çoğu zaman bu bilgi bilinmemektedir ve işlem kör olarak yapılmak zorundadır. Literatürde bu problem kör yankıdan arındırma problemi olarak bilinmektedir. Bu problemi çözmek amacıyla önerilen bazı metotlar olmasına rağmen, bu metotlar hem temiz sinyal hem de filtrenin bilinmemesi sebebiyle zorlaşan problemi tamamen çözmeyi başaramamışlardır. Bu tez, bu konuyu çözmek amacıyla önerilmiş olan Laplace tabanlı ağırlıklı kestirim hatası (L-WPE), Gauss tabanlı ağırlıklı kestirim hatası (G-WPE), negatif olmayan matris ayrıştırma (NMF) tabanlı zaman-frekans analizi (NMF+N-CTF), gecikmeli doğrusal kestirim yöntemi (DLP) gibi metotları karşılaştırmayı hedeflemekte ve ek olarak seyreklik

düzenlemeli ağırlıklı en küçük kareler (SPWLS) ismiyle yeni bir metot önermektedir. Deneylerimizde görülen en iyi sonuçlar genelde L-WPE metoduna sonrasında da G-WPE metoduna; uzun oda dürtü cevabına sahip sinyaller için ise işaret gürültü oranı (SNR) ve algısal konuşma kalitesi ölçütü açısından yeni önerilen G-WPE metoduyla ilklendirilmiş SPWLS metoduna aittir.

Table of Contents

Acknowledgments	v
Abstract	vi
Özet	viii
1 Introduction	1
1.1 Problem Definition and Motivation	1
1.2 Contributions and Organization of the Thesis	3
2 Background	4
2.1 Speech and reverberation modeling	4
2.1.1 Features of speech	4
2.1.2 Reverberation model	7
2.1.3 Room impulse response	8
2.2 Preliminaries	11
2.2.1 Solving dereverberation as an optimization problem	11
2.2.2 Linear prediction	12
2.2.3 Non-negative matrix factorization	13
3 Blind Dereverberation Methods	16
3.1 Delayed linear prediction (DLP)	16
3.2 Weighted prediction error method (G-WPE)	18
3.2.1 Gaussian model of speech	18
3.2.2 Formulation and algorithm	19
3.3 Laplacian model based weighted prediction (L-WPE)	21
3.3.1 Laplacian model of speech	21
3.3.2 Formulation and algorithm	22
3.4 NMF-based spectral modeling method	24
3.4.1 N-CTF Model Formulation	25
3.4.2 NMF Based Spectral Model	26
3.4.3 Combined Method of N-CTF and NMF	27

3.5	Sparsity penalized weighted least squares method (SPWLS)	29
3.5.1	Introduction to SPWLS method	29
3.5.2	SPWLS problem formulation	29
3.5.3	Proposed algorithm for solution	31
4	Experimental Results	34
4.1	Implementation setup	34
4.1.1	Methods to be compared	34
4.1.2	Test data	34
4.1.3	Analysis conditions and implementation details	35
4.2	Performance measures	36
4.2.1	Computational efficiency of dereverberation	36
4.2.2	Accuracy measures	36
4.3	Experimental results	37
4.3.1	Spectrogram results	37
4.3.2	Numerical evaluations	41
4.3.3	Robustness against RIR size	53
4.3.4	Loss function versus iterations of SPWLS method	55
5	Discussion and Conclusion	59
5.1	Discussion	59
5.2	Conclusion	60

List of Figures

2.1	Human Vocal System	5
2.2	Spectrogram of a Flute Signal	6
2.3	Spectrogram of a Speech Signal	7
2.4	Block Diagram of Reverberation	8
2.5	Reverberation effect on spectrogram	8
2.6	Room impulse response in time domain	10
4.1	Original (anechoic) speech signal	38
4.2	Reverberated speech signal	38
4.3	DLP dereverberation result	39
4.4	Laplacian-WPE method dereverberation result	39
4.5	Gaussian-WPE method dereverberation result	40
4.6	NMF+N-CTF method dereverberation result	40
4.7	SPWLS method dereverberation result	41
4.8	CD Results	43
4.9	STOI Results	43
4.10	SNR Results	44
4.11	Segmental SNR Results	44
4.12	PESQ Result	45
4.13	PESQ2 Result	45
4.14	PESQ3 Result	46
4.15	CD Results for NMF+N-CTF	46
4.16	STOI Results for NMF+N-CTF	47
4.17	SNR Results for NMF+N-CTF	47

4.18	Segmental SNR Results for NMF+N-CTF	48
4.19	PESQ1 Result for NMF+N-CTF	48
4.20	PESQ2 Result for NMF+N-CTF	49
4.21	PESQ3 Result for NMF+N-CTF	49
4.22	SNR for 20 iterations	50
4.23	Segmented SNR for 20 iterations	50
4.24	Cepstral Distance (CD) for 20 iterations	51
4.25	PESQ1 values for 20 iterations	51
4.26	PESQ2 values for 20 iterations	52
4.27	PESQ3 values for 20 iterations	52
4.28	STOI for 20 iterations	53
4.29	Total loss $\ \mathbf{W}(\mathbf{x} - \mathbf{H}\mathbf{s})\ _2^2 + \lambda_s \ \mathbf{s}\ _1 + \lambda_h (\ \mathbf{h}\ _2 - n_h)^2$	56
4.30	Loss function term $\ W(x - Hs)\ _2^2$	56
4.31	Loss function term $\lambda_s \ s\ _1$	57
4.32	Loss function term $\lambda_h (\ h\ _2 - n_h)^2$	57

List of Tables

4.1	Dereverberation Method Results for 20 files	42
4.2	Dereverberation Method Results for 72 files	42
4.3	Dereverberation method results for long RIR	54
4.4	Dereverberation method results for long RIR	55

Chapter 1

Introduction

This thesis compares the test results of several blind-dereverberation methods for single channel speech signals.

1.1 Problem Definition and Motivation

Reverberation is an effect caused by echoes received from blocking objects when an audio wave travels from an audio source to a listener. Reverberation on speech signals degrades applications such as Automatic Speech Recognition (ASR), hands-free teleconferencing and many more significantly. It may cause ASR performance to decrease even in a system trained using a database with reverberated speech [1, 2]. If reverberation environment is known, reverberation problem can be solved with a simple deconvolution operation due to the linear time invariant (LTI) structure of reverberation behaviour. However, if both clean (or anechoic) signal and reverberation environment is unknown, then the problem gets harder to solve. There are some significant approaches that suggest how to remove undesirable and detrimental reverberation effects from a speech signal.

One of the traditional methods is based on using the power spectral domain and spectral modeling [3], [4]. Power spectral techniques are generally based on suppression of the energy of the echo in the power spectral domain. These kind of algorithms are computationally faster as compared to the time-domain algorithms and since they do not make use of the phase information, they may be more robust. However, ignoring

phase information may hurt the accuracy of these algorithms [5], [6].

Another group of methods that are called linear prediction based dereverberation techniques, predicts the current samples of the signal from past samples to estimate the inverse of the room impulse response. Linear prediction (LP) [7], delayed linear prediction (DLP) [8], [9], and variance-normalized delayed linear prediction (NDLP) [10] are some of the examples which operate in the time-domain and in fact they give accurate results for late reverberation reduction. Late reverberant parts are known to be tardy parts of the reverberant components which are the most detrimental parts for ASR applications. However, time domain methods often has a huge computational cost because of having gigantic matrices to solve in their algorithms. To increase run time efficiency, authors in [11] propose direct application of short-time Fourier transform (STFT) to develop algorithms. They work fast and eliminate echo, although they may not be as accurate as time domain methods as mentioned in [10],[12].

Another popular method is utilizing inverse filtering technique to acquire the room impulse response [12], [13], [14]. Some inverse filter techniques use skewness, the scale for the symmetry, or kurtosis, the measure of being heavy-tailed or light-tailed compared to normal distribution, as the design criteria of the prediction residual [15], [16]. Disadvantages of these kind of algorithms are their non-compatibility with real-life noises and also room transfer function fluctuations may occur [17].

There are also methods based on the sparsity of clean speech spectrogram as [18],[19]. These methods model the dereverberation problem as an optimization problem. The optimization problem does not yield a closed form solution and iterative algorithms are applied to find the approximate solution. These algorithms are proven to be fast but their robustness is open to debate.

To summarize, blind-dereverberation on speech signals is a problem that is hard to be solved. Especially for single-channel speech signals, there are few algorithms which work satisfactorily and none of them can solve dereverberation problem completely. Thus, we suggest to compare the existing algorithms for blind speech dereverberation using multiple metrics and propose a new algorithm. We compare delayed linear prediction (DLP), Laplacian based weighted prediction error (L-WPE), Gaussian based

weighted prediction error (G-WPE), non-negative matrix factorization based spectral-temporal modeling (NMF and N-CTF) and we propose a new method that we call sparsity penalized weighted least squares (SPWLS).

1.2 Contributions and Organization of the Thesis

In this thesis we compare the existing single channel blind-dereverberation techniques and we propose a new approach. As discussed, there are very few resources related to the solution of single-channel blind-dereverberation.

Organization of the thesis is as follows: In Chapter 2 background on dereverberation problem and preliminaries are provided. Chapter 3 contains blind-dereverberation methods, their formulations and algorithms. In Chapter 4 we present numerical results and finally in Chapter 5, discussion of the results and suggestion on future works are presented.

Chapter 2

Background

In this chapter, basic background information for the blind-dereverberation problem will be provided. First, general model of reverberation process and statistic nature of speech will be presented. Secondly, room impulse response (RIR) model, features and generating RIR will be explained in details. In preliminaries section, important concepts such as non-negative matrix factorization, linear predictive coding, pseudo inverse, Toeplitz matrix and Tikhonov regularization utilized in this thesis will be introduced briefly.

2.1 Speech and reverberation modeling

2.1.1 Features of speech

Speech is a signal that is created through air and human vocal system which consists of the lungs, trachea, larynx, pharyngeal cavity, oral cavity and nasal cavity as shown in Figure 2.1. Vocal tract can be basically modeled as an all-pole filter in discrete time as given in Equation 2.1 and input to the vocal tract is called a glottal signal which can be approximated as white noise or an impulse train depending on the type of sound produced. Simply, speech is assumed to be produced by filtering the glottal signal with the following all-pole filter:

$$V(z) = \frac{G}{1 - \sum_{k=1}^N \alpha_k z^{-k}}, \quad (2.1)$$

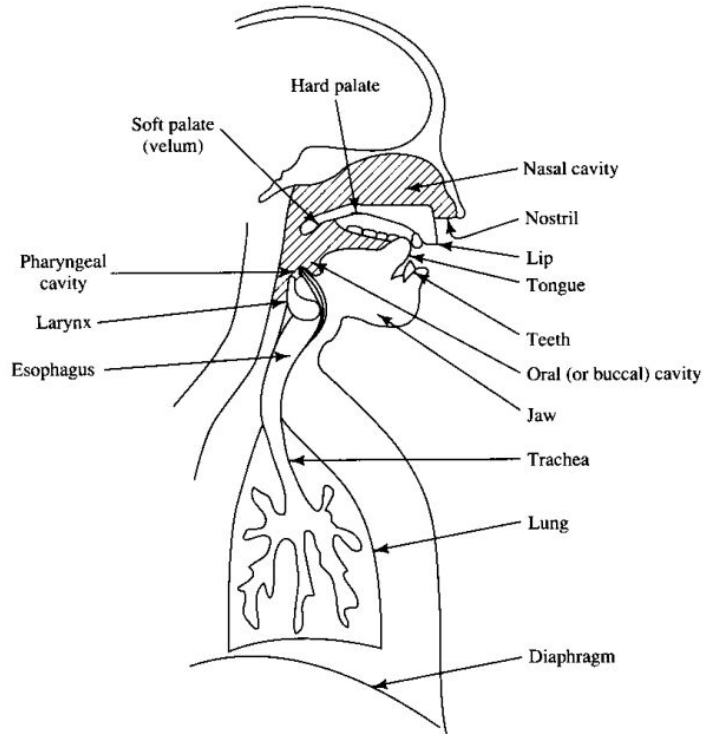


Figure 2.1: Human Vocal System

where G and α_k (reflection coefficients) depend on vocal tract movements. Speech signals have a non-stationary structure due to fast changes in vocal tract which results in time-varying all-pole filters. To form a model and utilize statistical properties, speech signal is divided into small time segments and we assume the signal in each time segment is stationary. Such signals are sometimes said to be quasi-stationary.

To analyze speech, one of the most used tools is short-time Fourier transformation (STFT). This transformation divides speech signal into overlapping segments called frames. It windows each time segment with a “Hamming Window” (other windows such as Hann, Kaiser etc. can be used as well) and calculates discrete Fourier transform (DFT) of these frames.

$$X(n, k) = \sum_{m=0}^N x[nL + m]w[m]e^{-j2\pi k/Nn} \quad (2.2)$$

where L is the frame shift, N is the frame size, and $X(n, k)$ is the discrete time short-time Fourier coefficients of the speech signal $x[m]$ at frame n .

STFT of a given signal is mostly interpreted as a matrix which has complex Discrete Fourier Transform (DFT) coefficients at its columns. Each column represents frequency information of each time segment or frame.

As discussed above, speech is not a stationary signal which means its properties are changing with respect to time. Hence, there is not much meaning to take speech signal as a whole. Thus, we use time dependent Fourier coefficients (i.e., STFT results) to observe the spectro-temporal variations of the speech signal.

Spectrogram is a visual representation of the spectrum of frequencies in the speech as they vary with time. It contains information about frequency content as a function of time, and the signal's time-varying power spectral density (PSD) is shown as intensity values in a 2D-image. Spectrogram matrix, $S(n, k)$, is calculated as follows:

$$S(n, k) = \log |X(n, k)|^2. \quad (2.3)$$

Also, spectrogram might be interpreted as a 3D-image with intensity bars for PSD values. Aim of the spectrogram is to show fast changing harmonics and their intensity values (amplitude values). As the human speech has high energy mostly between 300Hz - 3000Hz, other signals which interfere with the speech can be distinguished easily from the spectrogram if these signals have different frequency content outside this interval.

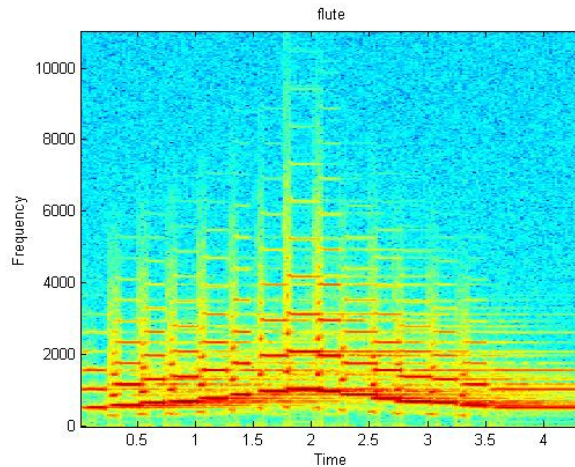


Figure 2.2: Spectrogram of a Flute Signal

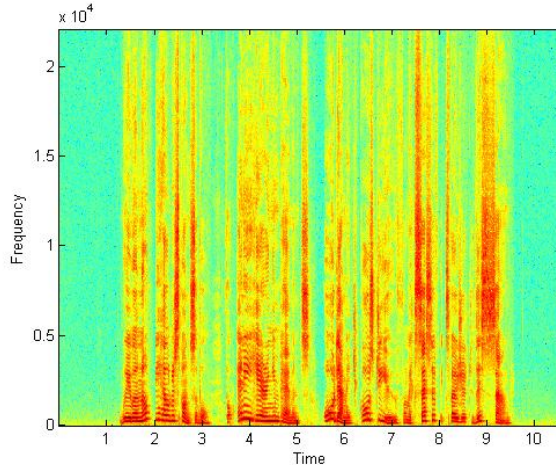


Figure 2.3: Spectrogram of a Speech Signal

2.1.2 Reverberation model

Reverberation is the persistence of sound after a sound is produced [20]. It occurs in consequence of reflections of sound through walls or objects. It can highly reduce the intelligibility of speech, degrade speech quality, and affect the performance of automated systems. Therefore, reverberation effect needs to be removed to improve these kind of applications. Process of removing echo from sound is called dereverberation. Dereverberation can be thought as pre-processing of speech signal. To eliminate echo, reverberation process must be modeled properly. In this case, room can be modeled as a filter called a room impulse response. Anechoic (or clean) speech signal is the input of this filter and as a result of this filtering operation we get the reverberated signal. Reverberation is usually modeled with an FIR filter as

$$x(t) = \sum_{\tau=0}^N h(\tau)s(t - \tau) \quad (2.4)$$

where $x(t)$ is the reverberated signal, $h(t)$ is the reverberation filter which is an FIR filter and $s(t)$ is the anechoic or clean signal. As seen from the Equation 2.4 reverberated signal is equal to the convolution of anechoic signal and a room impulse response filter.

Most of the time, both $s(t)$ and $h(t)$'s are unknowns and they should be estimated from reverberated signal $x(t)$ to eliminate echo. Estimating room impulse response and

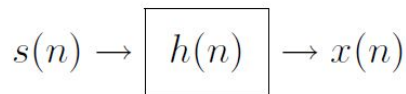


Figure 2.4: Block Diagram of Reverberation

$s(t)$ from known $x(t)$ is called blind dereverberation. It is not an easy task to do due to having one equation and two unknowns. Most of the time, more than one microphone (multi-channel) is used to solve blind dereverberation problems [10]. On the other hand, in this thesis we will focus on single microphone case.

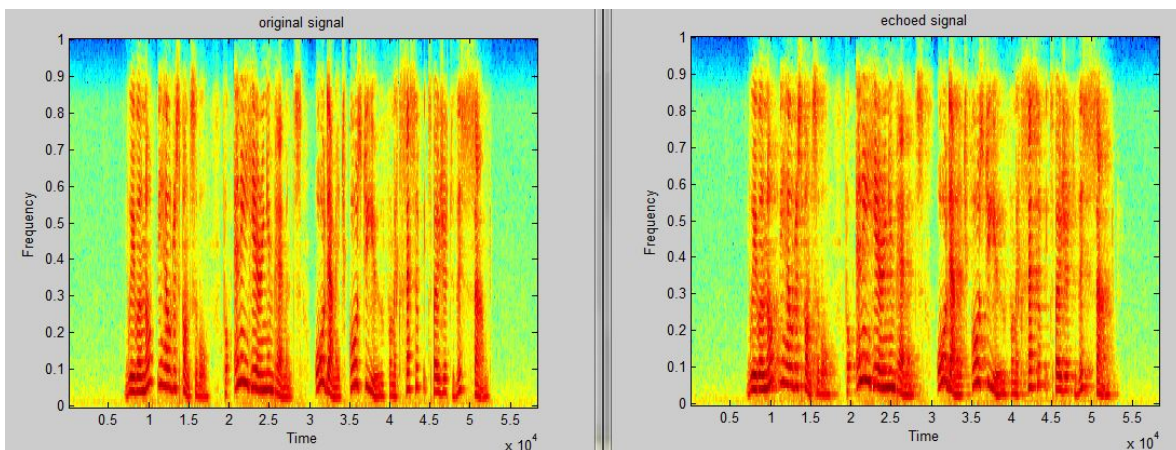


Figure 2.5: Reverberation effect on spectrogram

Reverberation effect on a speech signal can be seen in Figure 2.5 which contains spectrogram of a clean and reverberated (or echoed) signal. It can be seen that, original signal is more sparse as compared to echoed signal. Since speech signal is reflected through walls, a high intensity spectral content of the speech at a time continues to survive longer than the original one.

2.1.3 Room impulse response

In the literature, the FIR filter modeling the reverberations in a room, is called the room impulse response (RIR). The length of the RIR depends on many variables such as room size, room temperature, room shape, microphone's distance to the speech

source, absorption of sound due to materials used in room etc. To measure RIR in a room, a known signal, an impulse for example might be sent and then recorded with a microphone. As a consequence of linear and time-invariant (LTI) nature of RIR, anechoic signal can be estimated with a simple deconvolution operation if RIR is known. However, there is not always an opportunity to measure RIR this way. We may not have enough information about the room or microphone might be moving or room may have temperature fluctuations. Thus, we need more robust solutions to retrieve signal by removing the RIR effect.

One method to predict room impulse response is “inverse filtering” [10], [12]. In this case, inverse RIR is estimated to solve reverberation problem by simply predicting the filter coefficients which will be investigated in detail in Chapter 3. Other methods can be an iterative algorithm, which updates the filter and anechoic signals in each iterations according to well-determined constraints. There are also spectrum enhancement methods as [6],[21]. These kind of methods do not keep the phase information of signal and this process is more robust to microphone movements compared to inverse filtering methods. On the other hand, spectral enhancement method decreases the accuracy of the dereverberation process. Before investigating these algorithms in detail, we need to review important features of a RIR to understand it properly.

One of the significant properties of the RIR is reverberation time, RT_{60} . It is defined as the time required for reflected signal to drop by 60 dB level. It is a significant measure for dereverberation process, since RT_{60} indicates the length of the room impulse response. There are plenty of papers to estimate RT_{60} as in [21],[22],[23],[24]. However, this is not the main subject of this work.

Usually, room impulse response is divided into two as; early reverberation and late reverberation which is shown in Figure 2.6. Since, speech intelligibility is mostly affected by late reverberations, methods based on delayed linear prediction focus on eliminating late reverberations [10], [11] and they represent the reverberation process as:

$$x(t) = \sum_{\tau=0}^{L_h-1} h(\tau)s(t-\tau) + n(t), \quad (2.5)$$

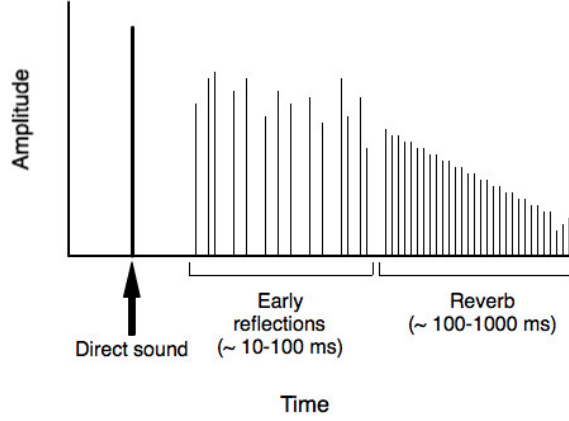


Figure 2.6: Room impulse response in time domain

$$x(t) = d(t) + r(t) + n(t). \quad (2.6)$$

A single-channel reverberation process can be represented as in Equation 2.5 where $x(t)$ is the reverberated signal, $h(t)$ is the room impulse response and L_h the length of room impulse response. In 2.6, $d(t)$ is the desired signal which is equal to the sum of early reverberant and anechoic signal, $n(t)$ is additive noise and $r(t)$ represents late reverberant signal. $d(t)$ and $r(t)$ are represented as:

$$d(t) = \sum_{\tau=0}^{D-1} h(\tau)s(t - \tau), \quad (2.7)$$

and

$$r(t) = \sum_{\tau=D}^{L_h-1} h(\tau)s(t - \tau), \quad (2.8)$$

where D is the sample length which divides room impulse response as early and late in Equation (2.8) and (2.7). First D samples are the early reverberant and the rest until $L_h - 1$ is the late reverberation part. More details are presented in Delayed Linear Prediction (DLP) section of Chapter 3.

2.2 Preliminaries

2.2.1 Solving dereverberation as an optimization problem

In this section, we assume we know the room impulse response and try to estimate the clean signal from the reverberated signal. Consider a reverberated signal without additive noise

$$x(t) = s(t) * h(t) \quad (2.9)$$

where $x(t)$ is the reverberated signal, $h(t)$ is the room impulse response (RIR), $s(t)$ is the anechoic signal in time domain and $*$ symbol represents convolution. We can convert this equation into matrix form as in the following:

$$\mathbf{x} = \mathbf{H}\mathbf{s} \quad (2.10)$$

where \mathbf{H} is a $L_x \times L_s$ matrix, \mathbf{x} is $L_x \times 1$ and \mathbf{s} is $L_s \times 1$ size vector with $L_x = L_s + L_h - 1$. Here \mathbf{H} is called the convolution matrix and \mathbf{x} and \mathbf{s} are the vectors corresponding to the signal samples from beginning to end. The effect of multiplying with \mathbf{H} is the same as convolving with the filter \mathbf{h} . The convolution matrix is the following

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & 0 & \dots & 0 \\ h_1 & h_0 & 0 & \dots & 0 \\ h_2 & h_1 & h_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{L_h-1} & h_{L_h-2} & h_{L_h-3} & \dots & 0 \\ 0 & h_{L_h-1} & h_{L_h-2} & \dots & \vdots \\ 0 & 0 & h_{L_h-1} & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & h_{L_h-1} \end{bmatrix}.$$

One can solve the following “regularized” least-squares optimization problem for a solution:

$$\arg \min_{\mathbf{s}} \|\mathbf{x} - \mathbf{H}\mathbf{s}\|_2^2 + \lambda \|\mathbf{s}\|_2^2. \quad (2.11)$$

This approach is useful when \mathbf{x} is noisy. Solution to the above minimization problem is given explicitly by

$$\mathbf{s} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^T \mathbf{x} \quad (2.12)$$

Note that, If this process is accomplished in complex domain for example after STFT is applied, then instead of transpose, conjugate transpose must be used.

If \mathbf{s} is sparse, then the following optimization problem is more appropriate:

$$\arg \min_{\mathbf{s}} \|\mathbf{x} - \mathbf{H}\mathbf{s}\|_2^2 + \lambda \|\mathbf{s}\|_1 \quad (2.13)$$

where $\|\mathbf{s}\|_1$ is the ℓ_1 norm of s vector. Solution of this problem cannot be explicitly written. This problem is numerically solved with an iterative algorithm. The underlying reason is that $\|\mathbf{s}\|_1$ is not a differentiable function. In the literature this problem is referred to as Lasso Problem (Least Absolute Shrinkage and Selection Operator) [25]. Lasso problem is a large and hard-problem to be solved, but it is convex [26],[27]. There are some fast algorithm suggestions such as [28]; ISTA [29], [30], [31]; FISTA [32]; SALSA [33], [34]. We investigate ISTA further in Chapter 3.

2.2.2 Linear prediction

Linear prediction involves predicting a sample in a signal from its past samples. If we write the linear prediction equation for the whole signal, we obtain:

$$y(t) = \sum_{k=1}^p \alpha_k y(t-k) + e(t) \quad (2.14)$$

where $y(t)$ is the signal to be predicted, α_k are prediction coefficients and $e(t)$ is the prediction error or residual at time t . sample. This formula sums up linear prediction of $y(t)$ from past p samples of $y(t)$ and then, the problem becomes determination of α_k 's to minimize $e(t)$. Denote $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_p]^T$. We convert (2.14) to matrix form as follows:

$$\mathbf{y} = \mathbf{Y}\boldsymbol{\alpha} + \mathbf{e}, \quad (2.15)$$

$$\arg \min_{\alpha} \|\mathbf{y} - \mathbf{Y}\alpha\|_p^p \quad (2.16)$$

where Y is the convolution matrix which consists of y 's past samples and $\|\cdot\|_p$ denotes the p -norm. By setting p to 2, Equation 2.16 becomes a least-squares problem

$$\arg \min_{\alpha} \|\mathbf{y} - \mathbf{Y}\alpha\|_2^2 \quad (2.17)$$

and its explicit solution is

$$\mathbf{Y}^T \mathbf{Y} \alpha = \mathbf{Y}^T \mathbf{y}. \quad (2.18)$$

This form is also known as Yule-Walker Method [35]. $\mathbf{R} = \mathbf{Y}^T \mathbf{Y}$ is an auto-correlation matrix which is a symmetric Toeplitz matrix [36]. A Toeplitz matrix has constant diagonals, so we can re-write Equation 2.18 as

$$\begin{bmatrix} R_0 & R_1 & R_2 & \dots & R_{N-1} \\ R_1 & R_0 & R_1 & \dots & R_{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{N-1} & R_{N-2} & R_{N-3} & \dots & R_0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_N \end{bmatrix} \quad (2.19)$$

Linear systems with Toeplitz matrices can be solved fast and without a need to store the whole matrix in memory. One such algorithm is Levinson-Durbin Algorithm which can be used to solve Toeplitz systems.

2.2.3 Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a common tool which is used for decomposing a nonnegative \mathbf{V} matrix as production of two matrices \mathbf{B} and \mathbf{G} with non-negative entries.

$$\mathbf{V} \approx \mathbf{B}\mathbf{G} \quad (2.20)$$

where \mathbf{B} is called basis or dictionary matrix and \mathbf{G} is called weight or gains matrix. This problem can be perceived as an optimization problem as follows:

$$\min_{\mathbf{B}, \mathbf{G}} C(\mathbf{V}, \mathbf{B}\mathbf{G}) \quad (2.21)$$

where C is the cost function for measuring the distance between \mathbf{V} and \mathbf{BG} . Columns of \mathbf{B} are called basis vectors and generally number of them are smaller than the size of \mathbf{V} in order to create a low-rank approximation of \mathbf{V} .

Iterative algorithms are utilized to solve Equation 2.21 since there is no unique solution in general for this problem. Solution of Equation 2.21 depends on the distance formulation. There are three popular iterative methods to formulate distance function between \mathbf{V} and \mathbf{BG} which are Euclidean Distance, Kullback-Leibler distance and Itakuro-Saito distance methods. Their formulation differs in regularization of \mathbf{B} or \mathbf{G} matrices.

Euclidean Distance Formulation calculates \mathbf{B} and \mathbf{G} as follows:

$$\min_{\mathbf{B}, \mathbf{G}} \|\mathbf{V} - \mathbf{BG}\|_2^2 \quad (2.22)$$

where,

$$\mathbf{B} \leftarrow \mathbf{B} \otimes \frac{\mathbf{VG}^T}{\mathbf{BGG}^T} \quad (2.23)$$

$$\mathbf{G} \leftarrow \mathbf{G} \otimes \frac{\mathbf{B}^T \mathbf{V}}{\mathbf{B}^T \mathbf{BG}} \quad (2.24)$$

where the operation \otimes is element-wise multiplication and division is element-wise divisions. \mathbf{B} and \mathbf{G} matrices are updated until a local minimum is found. Initial values of \mathbf{B} and \mathbf{G} matrices can be given either supervised or unsupervised as positive randomized matrices.

Kullback-Leibler Divergence Formulation calculates \mathbf{B} and \mathbf{G} as follows [37]:

$$\min_{\mathbf{B}, \mathbf{G}} D_{KL}(\mathbf{V} \parallel \mathbf{BG}). \quad (2.25)$$

where,

$$\mathbf{B} \leftarrow \mathbf{B} \otimes \frac{\mathbf{V} \mathbf{G}^T}{\mathbf{1G}^T}, \quad (2.26)$$

$$\mathbf{G} \leftarrow \mathbf{G} \otimes \frac{\mathbf{B}^T \mathbf{V} \mathbf{G}^T}{\mathbf{B}^T \mathbf{1}} \quad (2.27)$$

where “ $\mathbf{1}$ ” is the matrix of ones which has the same size of \mathbf{V} and D_{KL} is the generalized Kullback-Leibler divergence between \mathbf{V} and \mathbf{BG} and defined as:

$$D_{KL}(\mathbf{p}, \mathbf{q}) = \sum_i \{p_i \log \frac{p_i}{q_i} - p_i + q_i\}.$$

Itakura-Saito Divergence Formulation calculates \mathbf{B} and \mathbf{G} as follows [38]:

$$\min_{\mathbf{B}, \mathbf{G}} D_{IS}(\mathbf{V} \parallel \mathbf{BG}), \quad (2.28)$$

where,

$$\mathbf{B} \longleftarrow \mathbf{B} \otimes \frac{\frac{\mathbf{V}}{(\mathbf{BG})^2} \mathbf{G}^T}{\frac{\mathbf{1}}{\mathbf{BG}} \mathbf{G}^T} \quad (2.29)$$

and

$$\mathbf{G} \longleftarrow \mathbf{G} \otimes \frac{\mathbf{B}^T \frac{\mathbf{V}}{(\mathbf{BG})^2}}{\mathbf{B}^T \frac{\mathbf{1}}{\mathbf{BG}}} \quad (2.30)$$

where $(.)^2$ is an element-wise operation and D_{IS} is the Itakura-Saito Divergence between \mathbf{V} and \mathbf{BG} matrices and defined as:

$$D_{IS}(\mathbf{p}, \mathbf{q}) = \sum_i \left\{ \frac{p_i}{q_i} - \log \frac{p_i}{q_i} - 1 \right\}.$$

NMF is a non-convex algorithm and have multiple local minima. As a result of this, multiple \mathbf{B} and \mathbf{G} matrices can be found for the same \mathbf{V} matrix. To acquire better solutions for \mathbf{B} and \mathbf{G} matrices, supervised methods can be utilized.

NMF is a common model used in speech processing, deep learning, clustering, and computer vision. In audio processing, it was used for audio source separation [39, 40], blind-dereverberation [6] and speech denoising.

Chapter 3

Blind Dereverberation Methods

In this chapter, we denote $x(t)$, $s(t)$, $h(t)$ time-domain signals as x_t , s_t , h_t respectively. STFT-domain signals notations will be $x_{n,k}$, $s_{n,k}$ and $h_{n,k}$ instead of $x(n, k)$, $s(n, k)$ and $h(n, k)$ respectively.

3.1 Delayed linear prediction (DLP)

As discussed in 2.1.2 reverberated signal, x_t , can be formulated as convolution of RIR, h_t , and clean signal s_t as

$$x_t = \sum_{\tau=0}^{L_h-1} h_\tau s_{t-\tau} \quad (3.1)$$

where, L_h is the sample length of room impulse response (RIR). Then, L -length vectors \bar{s}_t , \bar{h} and \bar{x}_t are defined as $\bar{s}_t = [s_t, \dots, s_{t-L+1}]^T$ and $\bar{h} = [h_0, h_1, \dots, h_{L_h-1}, 0, \dots, 0]^T$ and $\bar{x}_t = [x_t, x_{t-1}, \dots, x_{t-L+1}]^T$ respectively.

Delayed linear prediction (DLP) is a method based on estimating inverse filter coefficients from reverberated microphone signal. With the inverse filter coefficients and reverberated signal, one can reach dereverberated signal with a simple filtering operation. In matrix form, reverberation can be formulated as

$$x_t = \bar{h}^T \bar{s}_t. \quad (3.2)$$

By using an inverse filter w_t of length L_w , we can approximately obtain a derever-

berated signal \hat{s}_t as:

$$\hat{s}_t = \sum_{k=0}^{L_w-1} w_k x_{t-k}. \quad (3.3)$$

Actual inverse filter is of infinite length since h_t is an FIR filter, but an FIR inverse filter would be an approximate inverse filter.

In Section 2.1.2 we talked about dividing room impulse response (RIR) into early and late reverberation parts. Assume zero noise for calculations, then

$$d_t = \sum_{k=0}^{D-1} h_k s_{t-k}, \quad (3.4)$$

$$r_t = \sum_{k=D}^{L_h-1} h_k s_{t-k} \quad (3.5)$$

where samples from D to $L_h - 1$ of h , is the late reverberation part and samples from 1 to $D - 1$ of h is early reverberation part. d_t is the desired signal which contains early reverberations and original signal. Simply, we are trying to eliminate r_t to remove most detrimental parts of echo. We can write x_t as follows:

$$x_t = d_t + r_t \quad (3.6)$$

In vector form, dereverberated signal can be written as:

$$x_t = d_t + \bar{h}_l^T \bar{s}_{t-D} \quad (3.7)$$

where $\bar{h}_l = [h_D, h_{D+1}, \dots, h_{L_h-1}, 0, \dots, 0]^T$ and $\bar{h}_e = [h_0, h_1, \dots, h_{D-1}, 0, \dots, 0]^T$.

Let's assume we can find a $\bar{c} = [c_1, c_2, \dots, c_{L_e}, 0, \dots, 0]$, such that $r_t \approx \bar{c}^T \bar{x}_{t-D}$. Then,

$$x_t = d_t + \bar{c}^T \bar{x}_{t-D} \quad (3.8)$$

where \bar{c} are called regression coefficients. Then desired signal \bar{d}_t becomes

$$d_t = x_t - \bar{c}^T \bar{x}_{t-D} \quad (3.9)$$

which means desired signal can be estimated by only using reverberated signal and its past samples.

Actually, in the DLP method, \bar{c} is found by self delayed linear prediction of x_t from its delayed samples \bar{x}_{t-D} . So, we find \bar{c} as the prediction coefficients that minimizes the norm of the difference $x_t - \bar{c}^T \bar{x}_{t-D}$. The idea is that the self-prediction that can be achieved by delayed samples will remove the late reverberant components in $x(t)$.

We mentioned in Equation 3.3 about estimating an inverse filter. With the guidance of Equation (3.9), inverse filter can be represented as $\bar{w}_t = [1, 0, 0, \dots, 0, -\bar{c}]^T$. The number of zeros in the inverse filter vector is equal to D , delay.

In conclusion, DLP algorithm is a simple technique to achieve dereverberation. However, it may not work well in most cases. The reason behind this is having an FIR filter as the inverse filter. We know that RIR acts as an FIR filter and expectation would be having an IIR filter as inverse of an FIR filter. In contrary, another FIR filter is approximately found as the inverse filter. As a result, this may be one of the reasons why the DLP method is suboptimal.

3.2 Weighted prediction error method (G-WPE)

Weighted prediction error (WPE) is rooted from DLP to solve dereverberation problem. It can be applied both in time domain and STFT domain. This method is suggested in [10] and to solve WPE problem, maximum likelihood estimation (MLE) is used. In this section, pre-assumptions, formulation and algorithm will be explained in depth.

3.2.1 Gaussian model of speech

For WPE method, speech signal s_t is assumed to have a local Gaussian distribution for small frames with length L_f . As a result, d_t , desired signal, also has Gaussian distribution due to the fact that weighted summation of Gaussians also forms Gaussian distribution. As a result, probability density function of d_t can be formulated as

$$p(d_t) = \mathcal{N}(d_t ; 0, \sigma_t^2). \quad (3.10)$$

Additionally, we assume that samples are mutually uncorrelated after a certain distance i.e.,

$$E\{s_t s_{t'}\} = 0 \quad \text{for } |t - t'| > \delta. \quad (3.11)$$

Third issue is time-varying variance assumption. We assume variance is constant for short-time frames with size L_f for WPE method. Variance of t samples of d_t can be approximated as follows:

$$\alpha_t^2 \approx \frac{1}{L_f} \sum_{t'=t-L_f/2+1}^{t+L_f/2} |d_{t'}|^2. \quad (3.12)$$

3.2.2 Formulation and algorithm

Dereverberation can be done both in time domain and in STFT domain. However, most of the time, using time domain is very costly, because of having quite big matrices to solve. As a result, we will solve dereverberation problem by WPE in STFT domain.

When we apply such algorithms in the STFT domain, we consider each frequency k separately and the signal $x_{n,k}$ as a function of n becomes a much shorter signal than the full time-domain signal x_t . Also, we assume that we can find prediction filters in the STFT domain and they also become much shorter than their time-domain counterparts. Hence, computationally, it becomes advantageous to work in the STFT domain.

Probability density function of desired signal in STFT domain, $p(d_{n,k})$ is as defined as

$$p(d_{n,k}) = \mathcal{N}(d_{n,k}; 0, \rho_{n,k}^2) \quad (3.13)$$

where n is frame number, k is frequency bin, $\rho_{n,k}^2$ is the time-varying variance value for each frequency and frame and defined as $\rho_{n,k}^2 = E\{d_{n,k} d_{n,k}^*\}$. $p(d_{n,k})$ is the probability distribution function of a complex Gaussian process, “ $(.)^*$ ” is conjugate operator and “ $(.)^T$ ” is transpose operator.

Then, our model in STFT domain becomes as

$$\hat{d}_{n,k} = x_{n,k} - (\bar{c}_k^*)^T \bar{x}_{n-D',k} \quad (3.14)$$

where D' is the number of delayed frames, \bar{c}_k is regression coefficient vector defined as $\bar{c}_k = [c_{1,k}, c_{2,k}, \dots, c_{L_c,k}]^T$ for each frequency bin.

We know that $\rho_{n,k}^2$, variance values alter only with respect to time frames. Thus, ρ_k^2 can be illustrated as $\rho_k^2 = \{\rho_{1,k}^2, \rho_{2,k}^2, \dots, \rho_{N,k}^2\}$.

Now, we will apply Likelihood maximization to Gaussian pdf in Equation 3.13. Parameter vector θ_k for likelihood maximization can be set as $\theta_k = \{\bar{c}_k, \rho_k^2\}$. Then, log likelihood function for dereverberation process in STFT domain becomes:

$$\mathcal{L}(\theta_k) = \sum_{n=1}^N \log p(d_{n,k}; \theta_k), \quad (3.15)$$

$$\mathcal{L}(\theta_k) = \sum_{n=1}^N \log p(d_{n,k} = x_{n,k} - (\bar{c}_k^*)^T \bar{x}_{n-D',k}; \theta_k), \quad (3.16)$$

$$\mathcal{L}(\theta_k) = - \sum_{n=1}^N \frac{|x_{n,k} - (\bar{c}_k^*)^T \bar{x}_{n-D',k}|^2}{\rho_{n,k}^2} - \sum_{n=1}^N \log(\rho_{n,k}^2) + const. \quad (3.17)$$

Maximizing the Equation 3.17 with respect to parameter vector θ_k cannot be achieved analytically and there is no closed form solution for this equation. Thus, an iterative algorithm is needed. Two step procedure has been proposed in [10] to solve Likelihood maximization problem as: in the first iteration, we keep $\rho_{n,k}^2$ constant and solve for $d_{n,k}$ by estimating \bar{c}_k regression coefficients to maximize equation; in second step, we keep $d_{n,k}$ constant and update $\rho_{n,k}^2$ and so on until a convergence criterion satisfied or a maximum number of iterations completed. For the first step, a linear least square problem has to be solved for \bar{c}_k and for second step, it is a simple calculation of variance through updated $d_{n,k}$. Algorithm 1 summarises the process.

Algorithm 1 : Gaussian based WPE Algorithm (in STFT domain)

Input: $x_{n,k}$, ϵ

while Condition criteria is not satisfied **do**

1. Initialize $\hat{\rho}_{n,k}^2 = \max\{|x_{n,k}|^2, \epsilon\}$
2. Update \hat{c}_k as : $\hat{c}_k = \left(\sum_{n=1}^N \frac{\bar{x}_{n-D',k} \bar{x}_{n-D',k}^H}{\hat{\rho}_{n,k}^2} \right)^+ \left(\sum_{n=1}^N \frac{\bar{x}_{n-D',k} x_{n,k}^*}{\hat{\rho}_{n,k}^2} \right)$
3. Update $\hat{d}_{n,k}$ as : $\hat{d}_{n,k} = x_{n,k} - \hat{c}_k^H \bar{x}_{n-D',k}$
4. Update $\hat{\rho}_{n,k}^2$ as : $\hat{\rho}_{n,k}^2 = \max\{|\hat{d}_{n,k}|^2, \epsilon\}$.

end while

The sign “ $(.)^H$ ” denotes the Hermitian transpose of a matrix, ϵ is the lower bound of $\hat{\rho}_{n,k}^2$ preventing zero divisions and “ $(.)^+$ ” denotes the Moore-Penrose pseudo-inverse.

In [10], algorithm 1 is suggested to do just one iteration. However, in Chapter 4, results up to 5 iterations have been found to obtain better results.

Frequency domain WPE compared to time domain WPE reduces the number of regression coefficients greatly. It results in fast computation due to small matrix size. For DLP algorithm, there is not much of a time complexity problem, since convolution matrix can be defined as a Toeplitz matrix. However, for time-domain WPE, this is not possible due to fast varying variance weights.

Result of STFT domain WPE algorithm can be summarized as normalization of speech signal samples with variance weights to make silent parts more silent and voiced parts more dominant. Additionally, as stated in [12], using frequency-domain WPE instead of time-domain is more robust to be combined with blind-source separation algorithms and it is much more faster.

3.3 Laplacian model based weighted prediction (L-WPE)

Laplacian model based WPE method (L-WPE) as proposed in [11] suggests that speech can be modeled more precisely with a Laplacian model rather than a Gaussian model in STFT domain. The L-WPE algorithm differs from regular WPE because of this assumption. In this section, Laplacian model based WPE method will be used to achieve single-channel speech dereverberation as written in [11].

3.3.1 Laplacian model of speech

For Gaussian based WPE algorithm, regression coefficients has been found in a closed form solution as a result of linear least square problem’s having exact solution in Section 3.2. On the other hand, STFT coefficients can be modeled more accurately with a Laplacian distributive model or a Gamma distributive model as mentioned in

[41],[42],[43],[11].

A Laplacian model will be proposed to represent STFT coefficients of the desired signal $d_{n,k}$ for each time-frequency bin with an equal variance $\rho_{n,k}^2/2$ for independent imaginary and real parts. Probability distribution function of the Laplacian model of desired signal can be formulated as:

$$p(d_{n,k}) = \frac{1}{\rho_{n,k}^2} e^{-2 \frac{|\Re(\mathbf{d}_{n,k})| + |\Im(\mathbf{d}_{n,k})|}{\rho_{n,k}}} \quad (3.18)$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote real and imaginary part of a complex number and $p(\cdot)$ symbolizes probability density function.

3.3.2 Formulation and algorithm

Likewise to solve Gaussian based WPE method, we will utilize maximum likelihood estimation (ML) to find parameter vector θ_k which is defined as $\theta_k = \{\bar{c}_k, \rho_k^2\}$ in Section 3.2. According to Laplacian pdf assumption of speech signal in frequency domain and Equation 3.18, we can write ML estimate of the parameter vector θ_k . Then, likelihood function of θ_k is

$$\mathcal{L}(\theta_k) = \sum_{n=1}^N -\log(\rho_{n,k}^2) - 2 \frac{|\Re(d_{n,k})| + |\Im(d_{n,k})|}{\rho_{n,k}} \quad (3.19)$$

where N is the number of time frames in STFT domain. As seen there is no closed formulation for 3.19. As a result we will apply an iterative algorithm to solve it numerically. First let's assume variance is fixed and maximize 3.19 with respect to regression coefficients \bar{c}_k and according to estimated \bar{c}_k vector, update desired signal $d_{n,k}$. Equation (3.19) can be rewritten in terms of \bar{c}_k as

$$\mathcal{L}(\bar{c}_k) = \sum_{n=1}^N -\log(\rho_{n,k}^2) - \frac{2}{\rho_{n,k}} \left(|\Re(x_{n,k} - \bar{c}_k^H \bar{x}_{n-D',k})| + |\Im(x_{n,k} - \bar{c}_k^H \bar{x}_{n-D',k})| \right). \quad (3.20)$$

$-\log(\rho_{n,k}^2)$ part of the equation does not depend on \bar{c}_k . As a result, this term will disappear while taking derivative of the equation with respect to \bar{c}_k .

$$\Re(x_{n,k} - \bar{c}_k^H \bar{x}_{n-D',k}) = \Re(x_{n,k}) - \bar{c}_k^T \bar{\bar{x}}_{n-D',k} \quad (3.21)$$

where

$$\bar{\bar{x}}_{n,k} = \begin{bmatrix} \Re(\bar{x}_{n,k}) \\ \Im(\bar{x}_{n,k}) \end{bmatrix}, \quad \bar{\bar{c}}_k = \begin{bmatrix} \Re(\bar{c}_k) \\ \Im(\bar{c}_k) \end{bmatrix} \quad (3.22)$$

likewise,

$$\Im(x_{n,k} - \bar{c}_k^H \bar{x}_{n-D',k}) = \Im(x_{n,k}) - \bar{c}_k^T \tilde{\tilde{x}}_{n-D',k} \quad (3.23)$$

where

$$\tilde{\tilde{x}}_{n,k} = \begin{bmatrix} \Im(\bar{x}_{n,k})^T & -\Re(\bar{x}_{n,k})^T \end{bmatrix}^T \quad (3.24)$$

Thus, likelihood function 3.20 can be rewritten as

$$\mathcal{L}(\bar{c}_k) = \sum_{n=1}^N -\frac{2}{\rho_{n,k}} \left(|\Re(x_{n,k}) - \bar{c}_k^T \bar{\bar{x}}_{n-D',k}| + |\Im(x_{n,k}) - \bar{c}_k^T \tilde{\tilde{x}}_{n-D',k}| \right) \quad (3.25)$$

To clear the minus sign, we can think it as an minimization problem instead of an maximization problem with a minus. Then, problem can be interpreted as a linear programming problem as given in

$$\begin{aligned} & \underset{t, \bar{c}_k}{\text{minimize}} && \|t\|_1 \\ & \text{subject to} && t \geq 0 \\ & && |\Re(x_{n,k}) - \bar{c}_k^T \bar{\bar{x}}_{n-D',k}| \leq \frac{\rho_{n,k}}{2} t_{2n-1} \\ & && |\Im(x_{n,k}) - \bar{c}_k^T \tilde{\tilde{x}}_{n-D',k}| \leq \frac{\rho_{n,k}}{2} t_{2n} \end{aligned} \quad (3.26)$$

where $t \in \mathbb{R}^{2N}$ and t_n represents the n-th element of the vector t and $\|t\|_1$ is ℓ_1 norm of t . $t_{2n} = [\Im(r_{n,k})]$ and $t_{2n-1} = [\Re(r_{n,k})]$ where $(r_{n,k})$ is the error. So, t is defined as upper bound of error for real and imaginary parts.

Second step, fix \bar{c}_k , update $\rho_{n,k}^2$ and rewrite log likelihood function as a function of variance as given in 3.19. To maximize it, take its derivative with respect to $\rho_{n,k}$ and equalize it to zero. Then, closed form solution for variance is

$$\rho_{n,k}^2 = \left(|\Re(d_{n,k})| + |\Im(d_{n,k})| \right)^2. \quad (3.27)$$

These two steps will proceed until a convergence criterion is satisfied or maximum number of iterations has been reached. Summary of the steps are given in Algorithm 2. where ϵ is the lower bound of $\hat{\rho}_{n,k}^2$. It is a very small positive number and prevents

Algorithm 2 : Laplacian based WPE Algorithm

Input: $x_{n,k}$, ϵ

Initialize: $\hat{\rho}_{n,k}^2 = \max\{(|\Re(x_{n,k})| + |\Im(x_{n,k})|)^2, \epsilon\}$

while Condition criteria is not satisfied **do**

1. Update \hat{c}_k as : solve LP in (3.26)
2. Update $\hat{d}_{n,k}$ as : $\hat{d}_{n,k} = x_{n,k} - \hat{c}_k^H \bar{x}_{n-D',k}$
3. Update $\hat{\rho}_{n,k}^2$ as : $\hat{\rho}_{n,k}^2 = \max\{(|\Re(\hat{d}_{n,k})| + |\Im(\hat{d}_{n,k})|)^2, \epsilon\}$

end while

solution from zero divisions.

As seen, this algorithm is much more complex than Gaussian based WPE because of linear programming part. Thus, it works very slow. Paper [11] claims that there may be different algorithms to solve L-WPE very fast. For our case, if the length of room impulse response (RIR) is long (0.5 sec or more), then algorithm takes more than a day to run with CVX, a package for specifying and solving convex programs [44],[45] in Matlab. These issues will be discussed in detail in Chapter 4.

3.4 NMF-based spectral modeling method

NMF-based spectral modeling method is proposed in [6] and [46]. This method does blind-dereverberation for a single-channel speech signal and it is a combined version of non-negative convoluted transfer function (N-CTF) model and non-negative matrix factorization (NMF) method. Assumption here is that for each frequency bin, the power spectrogram of STFT coefficient matrices of clean speech signal and room impulse response's convolution gives us the power spectrogram of STFT coefficient matrix of reverberated signal in (3.29). Note that the suggested method does not keep the phase information of RIR, because of only using power spectral domain.

$$x(n, k) \approx \sum_{\tau=0}^{L_h-1} h(\tau, k) s(n - \tau, k) \quad (3.28)$$

where $x(n, k)$, h and s are all in STFT domain which means they are complex and respectively represent reverberated, room impulse response (RIR) and clean signals. k is the frequency index and n is time-frame index. L_h is the frame length of RIR. It has been suggested in [47] to rewrite the Equation 3.28 for power spectral domain as

$$|x(n, k)|^2 \approx \sum_{\tau=0}^{L_h-1} |h(\tau, k)|^2 |s(n - \tau, k)|^2 \quad (3.29)$$

which is called non-negative convolutive transfer function (N-CTF) model. This model presumes that phase elements of the $h_{\tau,k}$ at different frames are mutually independent zero-mean random variable with Gaussian distribution as mentioned in [46] and additionally depends on the idea that clean signal and RIR spectral coefficients are also mutually independent.

For simplicity of notation, we will set $|x(n, k)|^2$ as $x_{n,k}$ and likewise for $s(n, k)$ and $h(n, k)$ as shown in Equation 3.30. Note that this notation is different than other methods for this thesis.

$$x_{n,k} \approx \sum_{\tau=0}^{L_h-1} h_{\tau,k} s_{n-\tau,k}. \quad (3.30)$$

3.4.1 N-CTF Model Formulation

In order to solve Equation 3.30 and to estimate power spectrogram of $s_{n,k}$, Kullback-Leibler (KL) divergence will be used. KL divergence is a common minimum distance algorithm and it has been investigated in Section 2.2.3.

$$Q = \sum_{k,n} KL\left(x_{n,k} \left| \sum_{\tau=0}^{L_h-1} h_{\tau,k} s_{n-\tau,k} \right.\right) \quad (3.31)$$

where

$$KL(x_{n,k} | \tilde{x}_{n,k}) = x_{n,k} \log\left(\frac{x_{n,k}}{\tilde{x}_{n,k}}\right) + \tilde{x}_{n,k} - x_{n,k} \quad (3.32)$$

and $\tilde{x}_{n,k}$ represents estimated power spectrogram of reverberated signal. To acquire more accurate estimation, we can use the sparsity of clean speech spectrogram and add a regularization term with a weight λ to the optimization problem given in (3.31).

$$Q = \sum_{n,k} KL\left(x_{n,k} \middle| \sum_{\tau=0}^{L_h-1} h_{\tau,k} s_{n-\tau,k}\right) + \lambda \sum_{n,k} s_{n,k}. \quad (3.33)$$

Additionally, as a non-negativity constraint, $s_{n,k}$ and $h_{\tau,k}$ are expected to be greater than zero. Thus, the problem turns into minimizing cost function 3.33 to estimate $h_{\tau,k}$. This model can be solved as an iterative learning method as

$$h_{\tau,k}^{i+1} = h_{\tau,k}^i \otimes \frac{\sum_n x_{n,k} s_{n-\tau,k}^i / \tilde{x}_{n,k}}{\sum_n s_{n-\tau,k}^i}, \quad (3.34)$$

$$s_{n,k}^{i+1} = s_{n,k}^i \otimes \frac{\sum_{\tau} x_{n+\tau,k} h_{\tau,k}^{i+1} / \tilde{x}_{n+\tau,k}}{\sum_{\tau} h_{\tau,k}^{i+1} + \lambda} \quad (3.35)$$

where $(\cdot)^i$ represents the iteration index and $\tilde{x}_{n,k} = \sum_{\tau} h_{\tau,k} s_{n-\tau,k}$. $\tilde{x}_{n,k}$ is computed from latest estimations of $h_{\tau,k}$ and $s_{n,k}$.

3.4.2 NMF Based Spectral Model

As discussed in Section 2.2.3, non-negative matrix factorization (NMF) is an algorithm to factorize a matrix as a product of two new matrices with non-negative entries. The method in [6] suggest to factorize the clean speech magnitude spectrogram \mathbf{S} as a production of a dictionary matrix \mathbf{B} and a weight matrix \mathbf{G} . Bold letters represent matrices.

$$\mathbf{S} \approx \mathbf{B}\mathbf{G} \quad (3.36)$$

where $\mathbf{B} = [b_{r,k}]^T$, $\mathbf{S} = [s_{n,k}]^T$ and $\mathbf{G} = [g_{n,r}]^T$. In other terms, Equation 3.36 can be rewritten as

$$s_{n,k} = \sum_{r=1}^R b_{r,k} g_{n,r}. \quad (3.37)$$

Here, R stands for the number of basis vectors in the dictionary matrix \mathbf{B} . R is smaller than number of frames in $s_{n,k}$, N , as discussed in 2.2.3.

3.4.3 Combined Method of N-CTF and NMF

In [6], it has been proposed to put Equation 3.37 directly into Equation 3.30 in order to create an integrated method. Then, cost function to be minimized becomes

$$Q_1 = \sum_{n,k} KL\left(x_{n,k} \mid \sum_{\tau=0}^{L_h-1} h_{\tau,k} \sum_{r=1}^R b_{r,k} g_{n-\tau,r}\right) + \lambda \sum_{r,n} g_{n,r}. \quad (3.38)$$

According to [46], sparsity constraint is enforced on the weight matrix $g_{n,r}$ and this assumption helps estimated s to be sparse.

To be able to minimize Q_1 , auxiliary function method is suggested which is similar to [37]. As seen, we have three variables to be alternatively updated in iterations: $h_{\tau,k}$, $b_{r,k}$, $g_{n,r}$. Thus, we need to keep two fixed and assume one is to be updated. That means, derivative of function will be calculated when it is zero to find a local minimum with respect to the variable which is going to be updated and other two variable must be keep fixed. This process will be carried on until a convergence criterion has been succeeded or maximum number of iteration has been reached. According to results of this method, update rule can be denoted as

$$h_{\tau,k}^{i+1} = h_{\tau,k}^i \frac{\sum_n x_{n,k} \tilde{s}_{n-\tau,k} / \tilde{x}_{n,k}}{\sum_n \tilde{s}_{n-\tau,k}} \quad (3.39)$$

$$b_{r,k}^{i+1} = b_{r,k}^i \otimes \frac{\sum_{n,r} x_{n,k} h_{\tau,k}^{i+1} x_{n-\tau,r}^i / \tilde{x}_{n,k}}{\sum_{n,\tau} h_{\tau,k}^{i+1} g_{n-\tau,r}^i} \quad (3.40)$$

$$g_{n,r}^{i+1} = g_{n,r}^i \otimes \frac{\sum_{\tau,k} x_{n+\tau,k} h_{\tau,k}^{i+1} b_{r,k}^{i+1} / \tilde{x}_{n+\tau,k}}{\sum_{\tau,k} h_{\tau,k}^{i+1} b_{r,k}^{i+1} + \lambda} \quad (3.41)$$

and

$$\tilde{s}_{n,k} = \sum_r b_{r,k} g_{n,r} \quad (3.42)$$

$$\tilde{x}_{n,k} = \sum_{\tau} h_{\tau,k} \tilde{s}_{n-\tau,k} \quad (3.43)$$

where $\tilde{s}_{n,k}$ and $\tilde{x}_{n,k}$ are updated according to the last estimates of the variables. To remove scale ambiguity, after each iteration each columns of \mathbf{B} is normalized to sum to one and the columns of \mathbf{H} are element-wise divided by the first column of itself as

suggested in [6], [37]. The nature of RIR consists of decaying impulses. Thus, paper [6] suggest to satisfy $h_{\tau,k} < h_{\tau-1,k} \forall \tau$.

Additionally, clean speech spectrogram estimation $\hat{s}_{n,k}$, can be accomplished through variables \mathbf{H} , \mathbf{B} , \mathbf{G} where they model a gain matrix $W_{n,k}$ which can be defined as the mapping coefficient matrix between clean speech signal and reverberated speech signal $x_{n,k}$. Gain matrix can be represented as $\hat{s}_{n,k} = W_{n,k} x_{n,k}$ where $W_{n,k}$ formulation is

$$W_{n,k} = \frac{\sum_r \hat{b}_{r,k} \hat{g}_{n,r}}{\sum_{\tau,r} \hat{h}_{\tau,k} \hat{b}_{r,k} \hat{g}_{n-\tau,r}} \quad (3.44)$$

As a result, the algorithm of the integrated version of N-CTF and NMF can be written as in Algorithm 3 where *Niter* is the total iteration number.

Algorithm 3 : N-CTF and NMF combined Algorithm

Input: $x(n, k)$

Output: $\hat{s}(n, k)$

1. **Initialize:** set parameters: R (number of basis vectors) , λ (sparsity weight), L_h (RIR length)

2. **Initialize:** \mathbf{H} , \mathbf{B} , \mathbf{G} with non-negative numbers (see Chapter 4 for more details)

3. **Compute:** compute power spectrogram of $x_{n,k} = |x(n, k)|^2$

while $i=1$ to *Niter* **do**

4. Update $\mathbf{H}^{i+1} = [h_{\tau,k}^{i+1}]$ by using 3.39

5. Update $\mathbf{B}^{i+1} = [b_{r,k}^{i+1}]$ by using 3.40

6. Update $\mathbf{G}^{i+1} = [g_{n,r}^{i+1}]$ by using 3.41

end while

7. **Compute:** compute gain function $W_{n,k}$ by using 3.44

8. **Compute:** compute dereverberated signal $\hat{s}(k, n)$ by taking inverse STFT of $\hat{s}(n, k) = W_{n,k}^{1/2} x(n, k)$

Initialization basis matrix \mathbf{B} can be learned online or offline. For online case, initializations of basis and weight matrices consists of randomized non-negative numbers as applied for regular NMF. However, as an alternative to online method, it has been proposed that a pre-basis matrix can be learned from a general speech signal database. In this thesis, we only employ the online method.

3.5 Sparsity penalized weighted least squares method (SPWLS)

3.5.1 Introduction to SPWLS method

Sparsity Penalized Weighted Least Squares (SPWLS) method is a novel approach which we introduce for single channel blind dereverberation problems. Different from [19], SPWLS approach combines the idea of variance normalization with a weight matrix and sparsity property of speech spectrogram matrices. In order to provide sparsity of a variable, generally ℓ_1 norm regularization is used as discussed in Section 2.2.1. With ℓ_1 regularization, optimization problem, also known as Lasso (Least absolute shrinkage and selection operator) problem, usually requires an iterative algorithm for solution. There are several popular algorithms to solve Lasso problem as ISTA (iterative shrinkage and threshold algorithm) [29][30][31], FISTA (fast-ISTA) [32] and SALSA (split variable augmented Lagrangian shrinkage algorithm) [33][34]. In our method we are going to use ISTA which will be explained in Section 3.5.3.

3.5.2 SPWLS problem formulation

As seen in NMF and WPE methods, we assume that the STFT of reverberated speech signal is equal to the convolution of RIR spectrogram and clean speech spectrogram which are unknown. This convolution is performed for each frequency k separately. For a fixed frequency, we can express it in a matrix form (dropping dependence on k) as

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{3.45}$$

where \mathbf{s} is the clean speech signal, \mathbf{x} is the reverberated signal, \mathbf{n} is noise signal and \mathbf{H} is the convolution matrix of RIR with complex coefficients all in the STFT domain with fixed frequency k . While solving this problem, we want to add a sparsity constraint because of the sparse nature of clean speech signal spectrogram and it is known that ℓ_1 norm works better with noisy cases as mentioned in [26]. So, we can define an optimization problem as

$$\arg \min_{\mathbf{s}, \mathbf{h}} \|(\mathbf{x} - \mathbf{H}\mathbf{s})\|_2^2 + \lambda_s \|\mathbf{s}\|_1 \quad (3.46)$$

where λ_s is a regularization parameter. In the literature, this type of equations are called Lasso problem or *basis pursuit denoising*. The Equation 3.46 does not have an exact solution, so an iterative numeric solution is needed.

Furthermore, we want to add weights to the problem as in Laplacian based WPE method and Gaussian based WPE method. In addition, we need an extra regularization on the norm of the filter \mathbf{h} which makes sure that we do not get a trivial solution. So, we update our optimization loss function as follows

$$\arg \min_{\mathbf{s}, \mathbf{h}} \|\mathbf{W}(\mathbf{x} - \mathbf{H}\mathbf{s})\|_2^2 + \lambda_s \|\mathbf{s}\|_1 + \lambda_h (\|\mathbf{h}\|_2 - n_h)^2 \quad (3.47)$$

where n_h is the target norm for the filter \mathbf{h} , where weight matrix \mathbf{W} is defined as

$$\mathbf{W} = \text{diag}(1/\hat{\rho}_{n,k}) \quad (3.48)$$

and

$$\hat{\rho}_{n,k}^2 = |\hat{s}_{n,k}|^2. \quad (3.49)$$

$\hat{\rho}_{n,k}^2$ is the precision values vector and $\text{diag}(1/\hat{\rho}_{n,k})$ is a diagonal matrix with reciprocal of the standard deviation entries. Note that, here k is fixed and n is the frame index variable which is indexing the vectors \mathbf{s} and \mathbf{x} . Hence \mathbf{W} is a diagonal weighting matrix which has a weight equal to the reciprocal of the standard deviations for each frame in its diagonal.

Note that the term $\mathbf{H}\mathbf{s}$ can also be written as $\mathbf{S}\mathbf{h}$ since it corresponds to a convolution operation and depending on the variable of interest, we may form a convolution matrix from \mathbf{h} called \mathbf{H} or we may form a convolution matrix from \mathbf{s} called \mathbf{S} each exactly determined from the filter coefficients. In practice, \mathbf{H} is a large matrix and \mathbf{S} is often manageable in size. So, during our implementations, multiplication with \mathbf{H} is implemented as a convolution, but multiplications with \mathbf{S} can be implemented as a matrix multiplication directly. This fact also affects our algorithm choice since we cannot use algorithms which require inverting matrices of large sizes which are derived from \mathbf{H} for example.

In the following, we discuss our proposed solution to this problem which is called the SPWLS algorithm.

3.5.3 Proposed algorithm for solution

As it is discussed in Section 3.5.2, our problem Equation 3.47 is non-differentiable with respect to \mathbf{s} at its local minimum. Thus, we have to calculate \mathbf{s} and \mathbf{h} numerically with an iterative approach. Our approach requires a good initialization for \mathbf{s} and \mathbf{h} which can be obtained from an earlier method such as G-WPE. After obtaining initial values from G-WPE, we perform alternating updates of \mathbf{s} and \mathbf{h} that would minimize the objective function with respect to the corresponding variable. For each update of \mathbf{s} , we fix the value of \mathbf{h} and update \mathbf{s} and vice versa. For updating \mathbf{s} , and \mathbf{h} , we use the ISTA algorithm which is well suited for these kind of problems.

ISTA is used to minimize functions like $f(\mathbf{s}) + g(\mathbf{s})$ where the first function is differentiable and the second function is usually not differentiable but simple. The first step of the ISTA algorithm takes a gradient descent step in the direction of the first function $f(\cdot)$

$$\mathbf{s}^{i+.5} = \mathbf{s}^i - \eta \nabla_{\mathbf{s}} f(\mathbf{s}^i),$$

where i is the iteration index and the result is an intermediate solution. Then a proximal operator of $g(\cdot)$ is performed around that intermediate solution as follows:

$$\mathbf{s}^{i+1} = \arg \min_{\mathbf{s}} g(\mathbf{s}) + \frac{1}{2\eta} \|\mathbf{s} - \mathbf{s}^{i+.5}\|_2^2.$$

This gradient descent followed by a proximal operator is shown to converge to the true solution under certain conditions [29]. Here, η is a positive step size parameter and indicates the amount that we move along the negative gradient.

If we calculate the gradient of the first function in Equation 3.47, we will find it as

$$\nabla_{\mathbf{s}} f(\mathbf{s}^i) = -2\mathbf{H}^H \mathbf{W}^2 (\mathbf{x} - \mathbf{H}\mathbf{s}^i). \quad (3.50)$$

The next proximal step corresponds to a thresholding/shrinkage operation for the ℓ_1 norm penalty. We update \mathbf{s} using the shrinkage operator defined as

$$\tau_a(\mathbf{s}) = (|\mathbf{s}| - a)_+ \frac{\mathbf{s}}{|\mathbf{s}|} \quad (3.51)$$

where $(x)_+ = \max(x, 0)$. Basically, it erases the components that have small energy and shrinks the other parts.

For our problem, the threshold a should be $\eta\lambda_s$ due to the ISTA algorithm as shown in Algorithm 4. In this algorithm, we show the algorithm for a fixed step size η . However, in practice we may want to reduce the step size as the iterations increase as shown in Equation 3.53.

After updating \mathbf{s} , we usually update the matrix \mathbf{W} which contains the variances of \mathbf{s} too. This update may hurt convergence of the algorithm, but since the variances are supposed to be more accurate after the update, it may help to improve the results as well.

To update \mathbf{h} , we need to solve the same optimization function for \mathbf{h} by fixing \mathbf{s} terms. After completing estimation of \mathbf{s} part, we will update weight matrix from estimated \mathbf{s} and update \mathbf{h} according to $\|\mathbf{W}(\mathbf{x} - \mathbf{S}\mathbf{h})\|_2^2 + \lambda_h(\|\mathbf{h}\|_2 - n_h)^2$ part due to derivative of other parts with respect to \mathbf{h} being zero. We can again use the ISTA algorithm for this purpose. But, we make one change. Since it is easy to find the minimizer of $f(\cdot)$ for updating \mathbf{h} , we find the minimizer for $f(\cdot)$ and perform the proximal operator step afterwards. Minimizing $f(\cdot)$ is a simple least squares problem with an exact solution

$$\mathbf{h} = (\mathbf{S}^H \mathbf{W}^H \mathbf{W} \mathbf{S})^{-1} \mathbf{S}^H \mathbf{W}^H \mathbf{W} \mathbf{x}. \quad (3.52)$$

Next step is to perform the proximal step for the regularization of \mathbf{h} . This corresponds to a rescaling of the solution as follows:

$$\mathbf{h}^{i+1} = \frac{1 + \lambda_h \frac{n_h}{\|\mathbf{h}\|}}{1 + \lambda_h} \mathbf{h}.$$

In practice, for the inner gradient descent iteration for \mathbf{s} , the step size parameter η can be made to change for each iteration. We apply a schedule as follows:

$$\eta_{ij} = \alpha^{1-i} \beta^{1-j} \eta_0, \quad (3.53)$$

where α and β are hyperparameters and η_0 is the initial step size, i and j are the inner and outer iteration indices (see Algorithm 4) respectively. At first iteration, we have $\eta = \eta_0$ and the step size keeps decreasing as we increase the iterations.

Note that, the target norm n_h of the filter \mathbf{h} can be obtained as the norm of the initial \mathbf{h} filter.

Algorithm 4 SPWLS Algorithm with ISTA

Input: $\mathbf{x} = \{X(n, k), n = 1 : \dots : N_f\}$ repeat for each k

Output: \mathbf{s}, \mathbf{h} (note \mathbf{H} and \mathbf{S} are convolution matrices of signals \mathbf{h} and \mathbf{s})

Initialize: $\mathbf{s}, \mathbf{h}, \mathbf{W}$

Set parameters: $\lambda_s, \epsilon, \lambda_h, N_{out}, N_{in}$

for $j=1$ to N_{out} **do**

for $i= 1$ to N_{in} **do**

 1. **Update** \mathbf{s} :

 Determine $\eta = \eta_{ij}$ (use Equation 3.53)

$\mathbf{s}^{i+.5} = \mathbf{s}^i - \eta \nabla_{\mathbf{s}} f(\mathbf{s}^i)$ (use Equation 3.50)

$\mathbf{s}^{i+1} = \tau_{\lambda_s \eta}(\mathbf{s}^{i+.5})$ (use Equation 3.51)

end for

 2. **Update new** \mathbf{W} : use Equation 3.48

 3. **Update** \mathbf{h} :

$\mathbf{h}^{i+.5} = (\mathbf{S}^H \mathbf{W}^H \mathbf{W} \mathbf{S})^{-1} \mathbf{S}^H \mathbf{W}^H \mathbf{W} \mathbf{x}$.

$\mathbf{h}^{i+1} = \frac{1+\lambda_h \frac{n_h}{\|\mathbf{h}^{i+.5}\|}}{1+\lambda_h} \mathbf{h}^{i+.5}$

end for

Chapter 4

Experimental Results

4.1 Implementation setup

4.1.1 Methods to be compared

1) DLP: Delayed linear prediction method which is proposed in [9]. It is discussed in Section 3.1.

2) G-WPE: Gaussian based weighted prediction error method. It is based on variance normalization method proposed in [10], discussed in Section 3.2

3) L-WPE: Laplacian-based WPE method which is proposed in [11], discussed in Section 3.3

4) NMF+N-CTF: NMF based N-CTF method which is proposed in [6]. It is discussed in Section 3.4

5) SPWLS: Spectrogram sparsity based weighted optimization method which is discussed in Section 3.5.

4.1.2 Test data

For the first experiment, 3 male and 3 female voices without reverberation has been convolved with 6 different room impulse response samples with 30dB and 60dB additive noises separately for DLP, G-WPE, NMF+N-CTF, SPWLS methods. It means 72 different speech signal has been dereverberated for experiment 1. For the experiment 2, 1 male and 1 female voices without reverberation has been convolved with 5 different

RIR samples and added 30dB and 60dB additive noises separately onto them to test all methods. It means 20 different reverberated and noised speech signals have been dereverberated for experiment 2. Test data has been taken from “Reverb Challenge” [48] data set. Sampling frequency was 16KHz which is same for all the audio data.

Room impulse response times (RT_{60}) were 0.17, 0.11, 0.95, 0.33, 0.54, 0.35s respectively. L-WPE method was not performed with 0.95 sec RT_{60} only due to excessive run time.

The input to all algorithms is the reverberated speech signal and the RT_{60} values for the unknown RIRs. Original signals without echo and noise have been only used while testing them.

As noise data, a “.wav” file with cafe sounds has been used as additive noise with 30dB and 60db levels.

4.1.3 Analysis conditions and implementation details

Sampling rate for all the signals were 16KHz. Number of delayed frame size, D was set to 3 frames for G-WPE, L-WPE and DLP methods. NMF+N-TFC method and SPWLS method do not include D variable. L_f , number of frames used for variance calculations is set to 1 frame for G-WPE, L-WPE and SPWLS methods, “NMF+N-CTF” method and DLP do not contain any variance calculations and L_f variable. Iteration numbers for all methods except NMF+N-CTF have been set to 5. NMF+N-CTF method has 100 iterations due to slow convergence rate as proposed in [6]. STFT parameters are hop size and STFT window size which have been set to 10 ms and 30 ms respectively. Furthermore, Minimum variance to avoid zero divisions, ϵ has been set to 10^{-6} .

Number of STFT frames used to predict signals is obtained from RT_{60} values which is fixed for all methods.

SPWLS parameters specific to this method are step size, η set as 10^{-7} , ISTA regularization parameter λ_s set as 10^5 , inner iteration number for ISTA set as 10, ISTA regularization parameter for filter λ_h set as 10. SPWLS initialization for RIR, \mathbf{h} is obtained using Equation 3.52 using the \mathbf{s} value from the output of the G-WPE method.

NMF+N-CTF method has dictionary matrix size “ndict” as 100 for the experiments. “ndict” is altered for some cases to observe the effect of it over results. NMF+N-CTF method uses online method.

4.2 Performance measures

4.2.1 Computational efficiency of dereverberation

The computational efficiency is an important issue. The algorithm is computationally efficient means that we can apply it in real time applications. All the algorithms are implemented in MATLAB on a computer with an Intel Xeon CPU, 2.5GHz.

When we compare all the methods, the fastest one becomes SPWLS method, then G-WPE, DLP, NMF+N-CTF and L-WPE come respectively. L-WPE is very slow due to linear programming (LP) part inside. We used CVX tool [45] [44] to solve the LP part of the L-WPE algorithm. For a reverberated signal with 0.5s reverberation time, L-WPE is solved in approximately one day. Also, NMF+N-CTF method is quite slow. It takes around 1.5 hour for dereverberating the same data. On the other hand, the code takes approximately 2-3 minutes for G-WPE, SPWLS and DLP. DLP method is implemented with Levinson-Durbin algorithm to accelerate the process. This method does not keep the whole convolution matrix, instead it creates an auto-correlation matrix and just keeps the first row. This kind of solutions are called matrix-free. Due to the fact that G-WPE, L-WPE and SPWLS methods have variance weights, it is not possible to apply Levinson-Durbin algorithm to these methods.

4.2.2 Accuracy measures

Accuracy of the dereverberation process is calculated with average cepstral distortion (CD) over short time frames as suggested in [49]. It is a very common and popular tool to measure speech quality measure between clean signal and reconstructed signal. CD in short time frame can be defined as

$$CD = (10/\ln 10)^2 \sqrt{(\hat{\beta}_0 - \beta_0)^2 + 2 \sum_{k=1}^{12} (\hat{\beta}_k - \beta_k)^2} \quad (4.1)$$

in dB and where β_k is clean speech signal cepstral coefficients from 1th to 12th order and $\hat{\beta}_k$ is estimated speech signal's cepstral coefficients with same order. Zero order coefficient, β_0 defines the power spectrum envelope in dB. CD between similar signals converges to 0. Our aim is to keep CD as small as possible after dereverberation process. CD results can be found in Section 4.3.

Signal to noise ration (SNR) and segmental SNR, SNR obtained from short segments, measurements have been used as a signal level accuracy measure for performance measurement.

We also evaluate algorithms using STOI and PESQ measures as well. Short-time objective intelligibility measure (STOI) is an intelligibility measure which is introduced in [50]. For short-time frames, it compares the temporal envelopes of the clean and degraded speech in terms of correlation coefficients. Perceptual Evaluation of Speech Quality (PESQ) is a common standardized test method for speech quality [51].

4.3 Experimental results

4.3.1 Spectrogram results

In this section, spectrogram effects of the dereverberation methods will be illustrated. To be able to capture the differences, additionally spectrogram figures of clean and reverberated signals are shown. Spectrogram corresponding to a clean signal is provided in Figure 4.1, whereas the spectrogram for the reverberated signal is in Figure 4.2. DLP reconstructed signal's spectrogram is in 4.3, whereas L-WPE, G-WPE, NMF+N-CTF and SPWLS reconstructed signals' spectrograms are in Figures 4.4, 4.5, 4.6, 4.7 respectively. We observe that reverberation causes extension of structures and other effects in the spectrogram and dereverberation methods such as L-WPE, G-PWE and SPWLS manage to remove some of those effects to yield a spectrogram close to the clean signal's spectrogram.

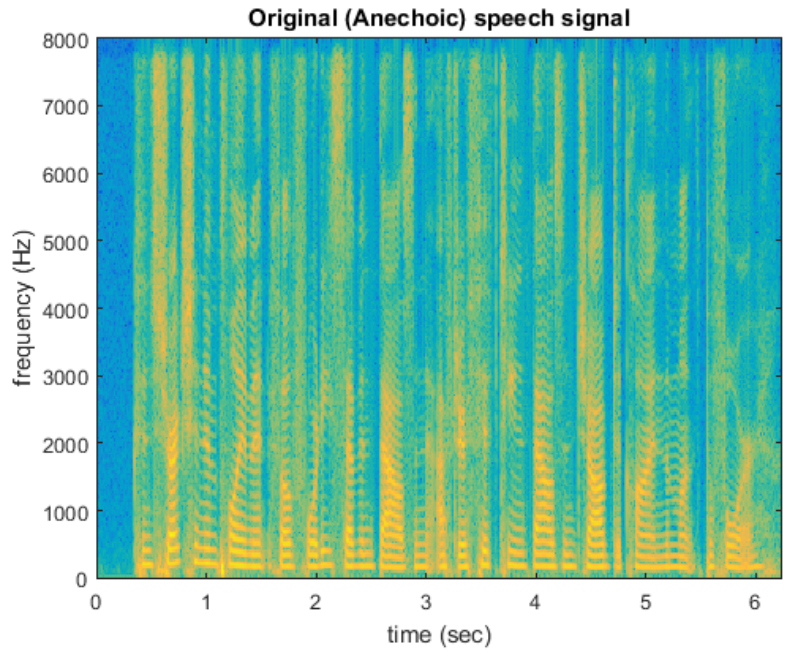


Figure 4.1: Original (anechoic) speech signal

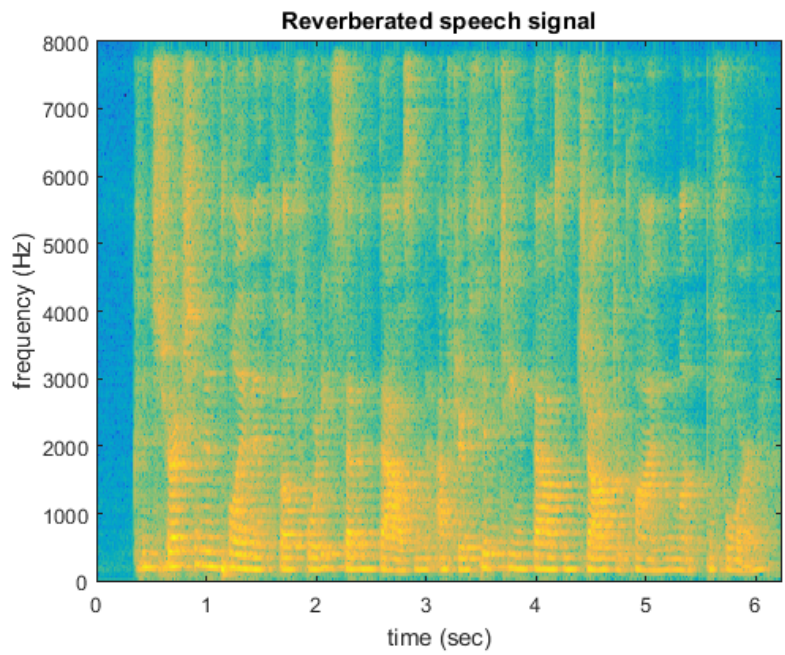


Figure 4.2: Reverberated speech signal

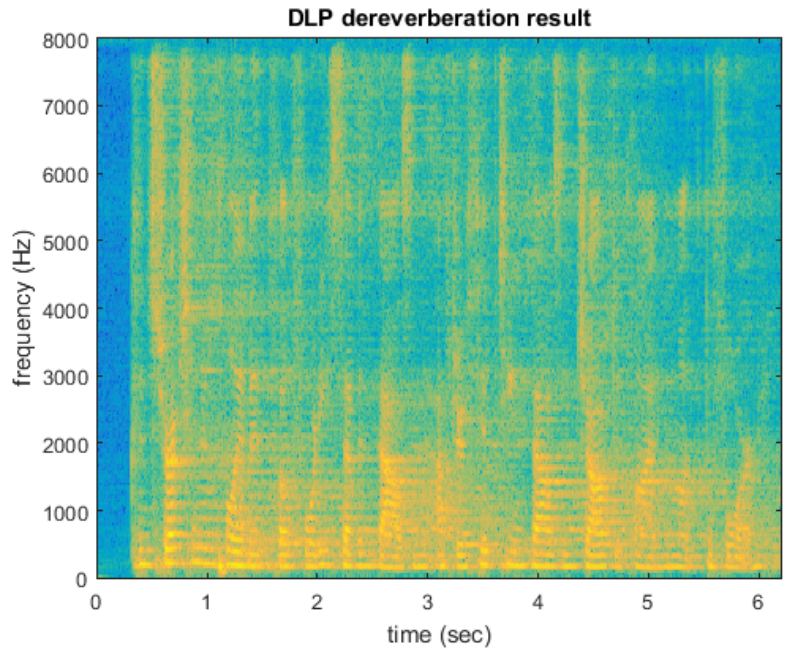


Figure 4.3: DLP dereverberation result

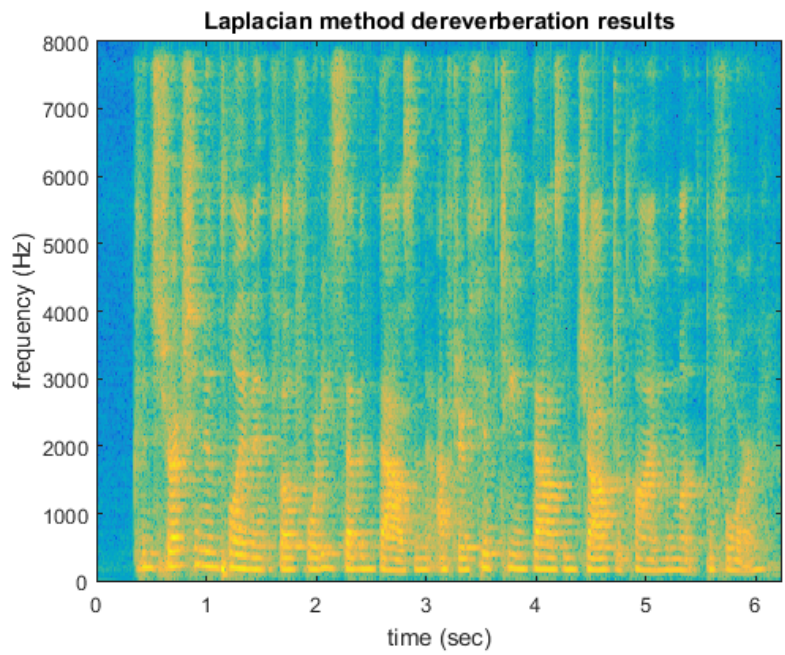


Figure 4.4: Laplacian-WPE method dereverberation result

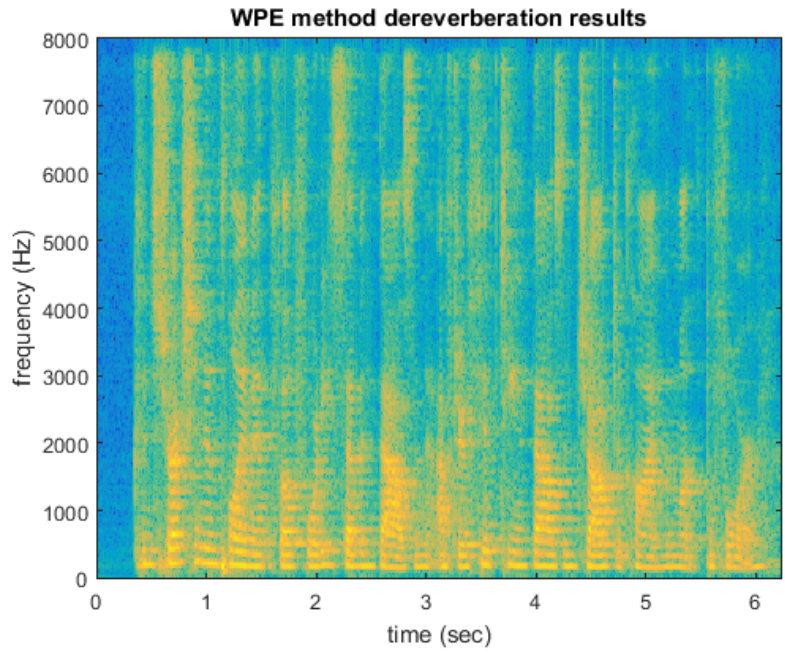


Figure 4.5: Gaussian-WPE method dereverberation result

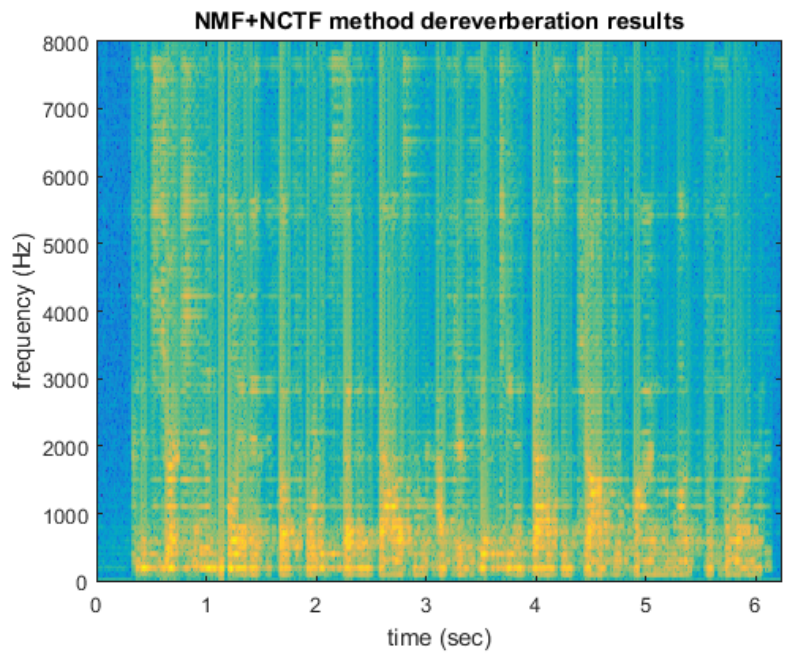


Figure 4.6: NMF+N-CTF method dereverberation result

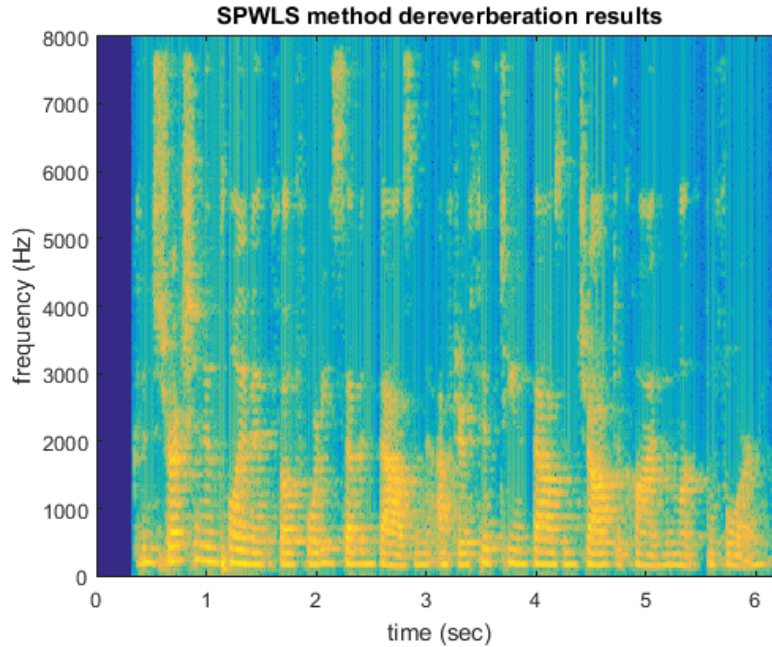


Figure 4.7: SPWLS method dereverberation result

4.3.2 Numerical evaluations

In Table 4.1 and 4.2 below, the results of experiments are presented. The accuracy measures are averaged over all files. “Revb/Clean” in tables refers to accuracy measures between reverberated and clean speech signals to show the difference and alterations in results after dereverberation. Three different types of PESQ results indicated with PESQ1, PESQ2, PESQ3 are wideband MOS-LQO, narrowband PESQ-MOS and narrowband MOS-LQO respectively.

Table 4.1: Dereverberation Method Results for 20 files

Method	SNR	segSNR	CD	PESQ1	PESQ2	PESQ3	STOI
Revb/Clean	-2.0086	0.2080	20.4112	2.6094	2.3403	1.8367	0.8911
L-WPE	-0.8339	0.5731	16.2641	2.9742	2.8165	2.3402	0.9202
G-WPE	-0.8922	0.5532	16.4651	2.9495	2.7826	2.3036	0.9188
SPWLS	-0.4435	0.6569	56.7051	2.4384	2.1024	1.5539	0.8796
NMF+CTF	-0.0379	0.0357	27.2434	1.8837	1.5674	1.2315	0.6839
DLP	-2,049	0,5652	22,1378	2,4976	2,1710	1,6273	0,8955

Table 4.2: Dereverberation Method Results for 72 files

Method	SNR	segSNR	CD	PESQ1	PESQ2	PESQ3	STOI
Revb/Clean	-2.7062	-0.3922	24.1169	2.3945	2.1012	1.5998	0.8687
G-WPE	-1.6003	-0.0970	20.2801	2.6956	2.4624	1.9540	0.8985
SPWLS	-1.1054	0.0006	51.8132	2.4507	2.1380	1.5610	0.8768
NMF+CTF	-0.1233	0.0016	30.4215	1.8197	1.5268	1.1714	0.6725
DLP	-2.7661	-0.0246	25.9241	2.3365	2.0062	1.4707	0.8732

For the Table 4.1 and 4.2, iteration sizes for NN-CTF (NMF+N-CTF method), L-WPE, G-WPE, DLP, SPWLS are 100, 5, 5, 1, 5 respectively. Due to different iteration numbers, we investigated the test results separately for NN-CTF alone and L-WPE, G-WPE, SPWLS group.

We provide plots of test results versus iteration number for the L-WPE, G-WPE and SPWLS methods in Figures 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14. These performance measures have been obtained as an average over 20 files. It is observed that SPWLS method gets worse in certain measures as the iterations are increased. This may be due to the non-convex nature of the loss function used in the problem which does not converge to a desirable solution.

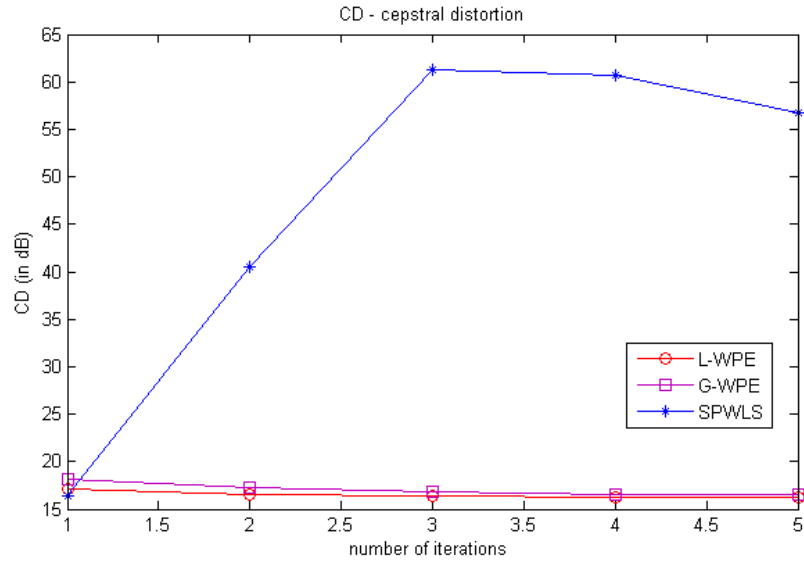


Figure 4.8: CD Results

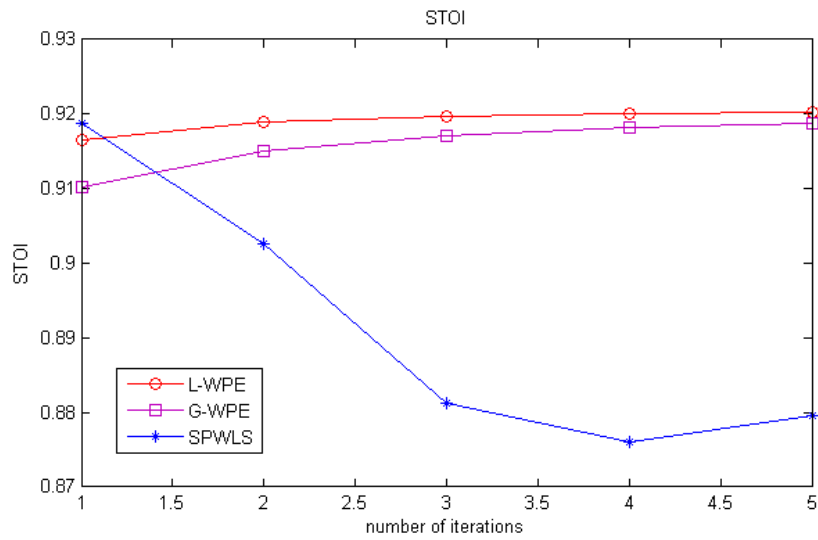


Figure 4.9: STOI Results

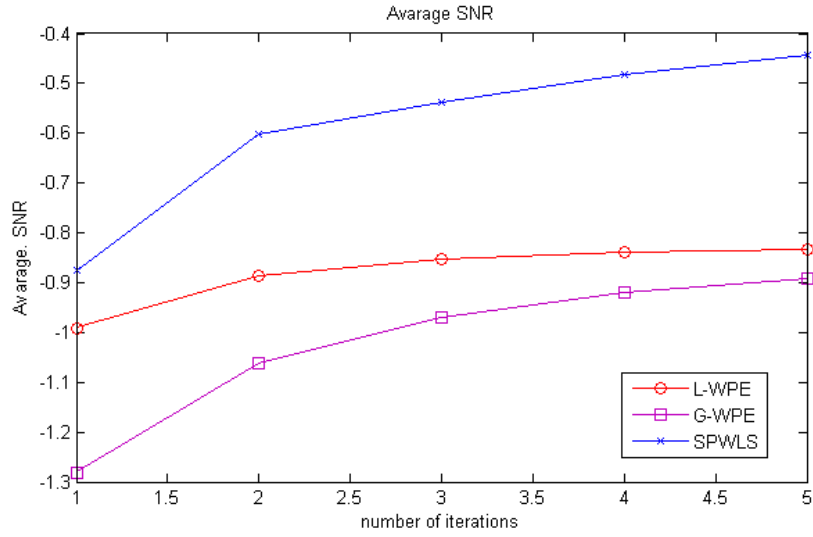


Figure 4.10: SNR Results

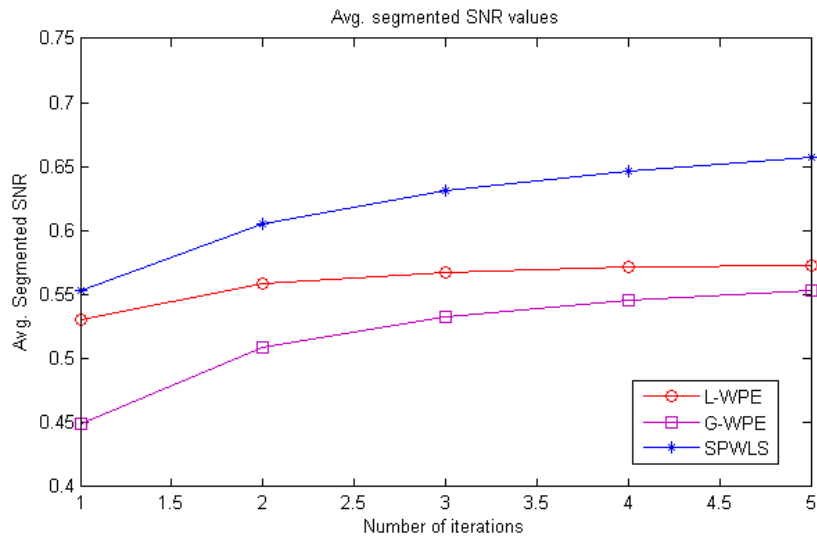


Figure 4.11: Segmental SNR Results

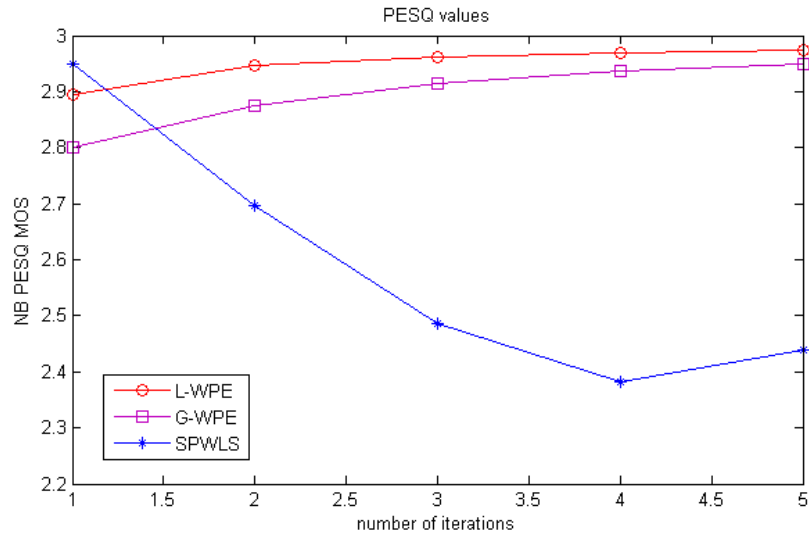


Figure 4.12: PESQ Result

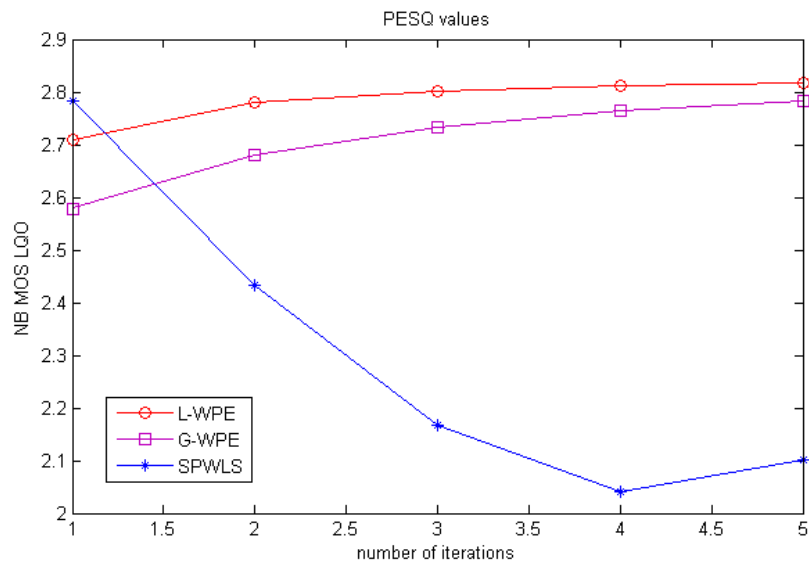


Figure 4.13: PESQ2 Result

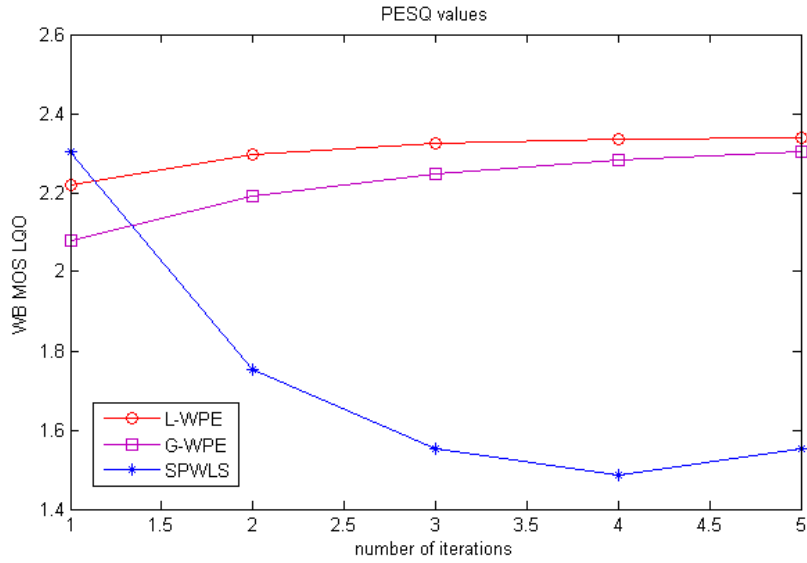


Figure 4.14: PESQ3 Result

We provide NMF+N-CTF test results versus number of iterations plots in Figures 4.15, 4.16, 4.17, 4.18, 4.19, 4.20, 4.21.

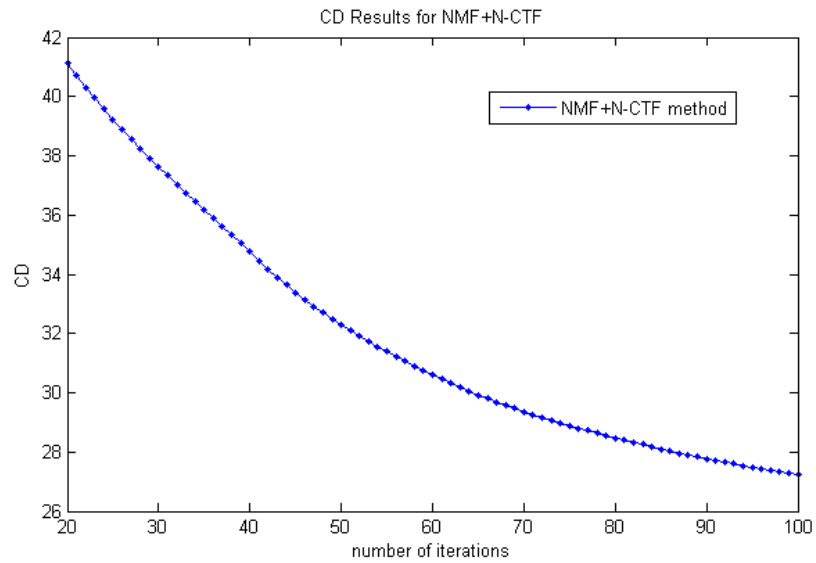


Figure 4.15: CD Results for NMF+N-CTF

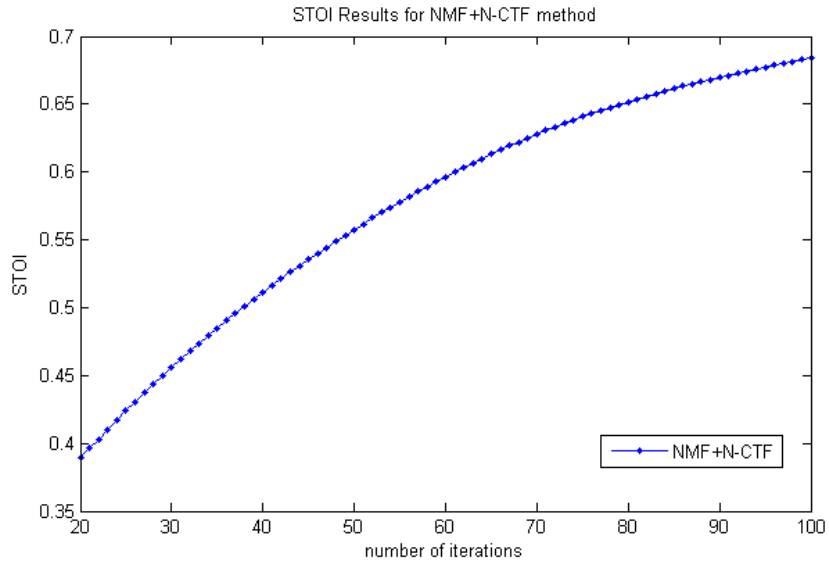


Figure 4.16: STOI Results for NMF+N-CTF

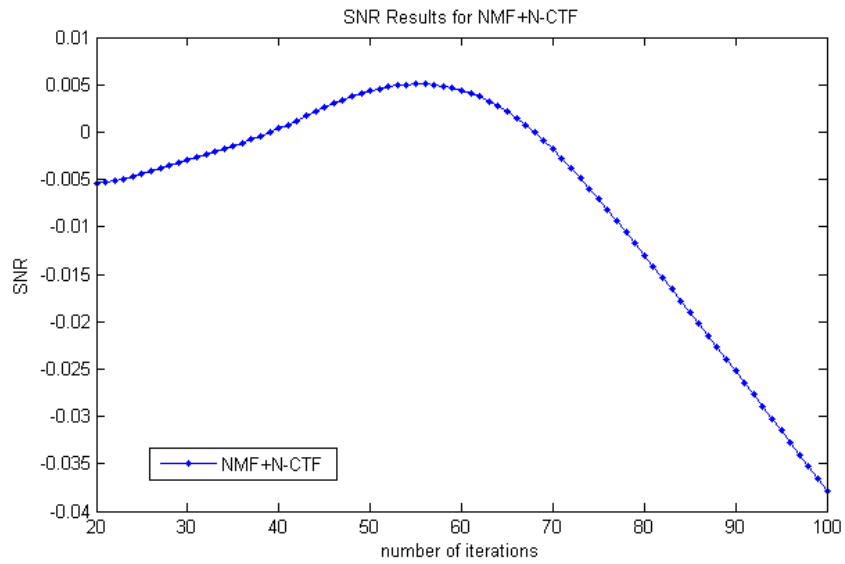


Figure 4.17: SNR Results for NMF+N-CTF

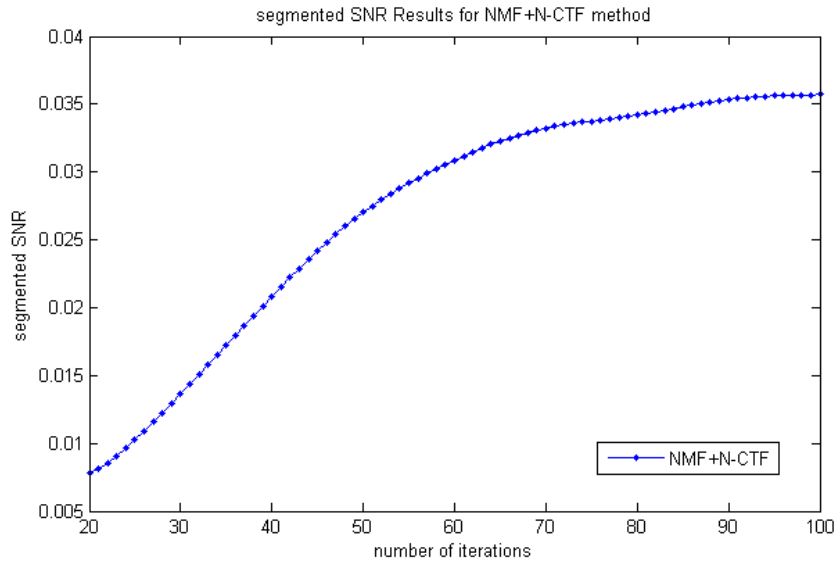


Figure 4.18: Segmental SNR Results for NMF+N-CTF

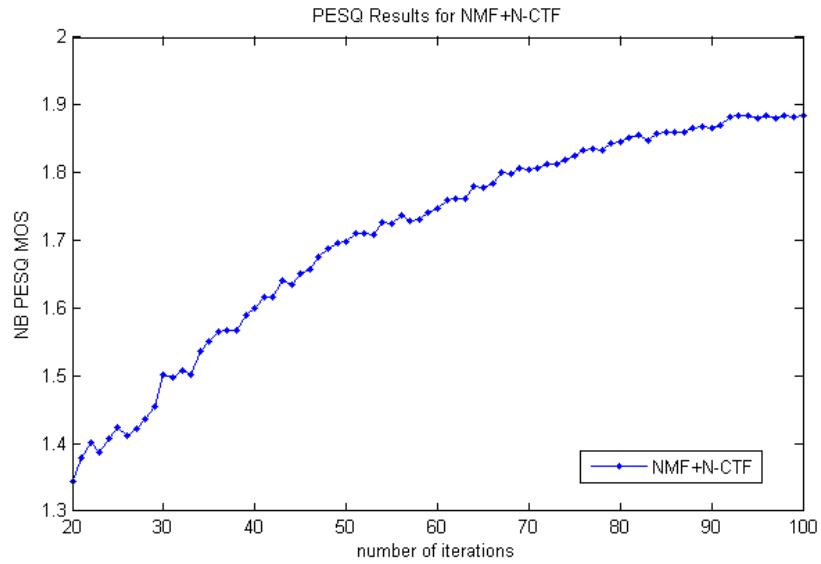


Figure 4.19: PESQ1 Result for NMF+N-CTF

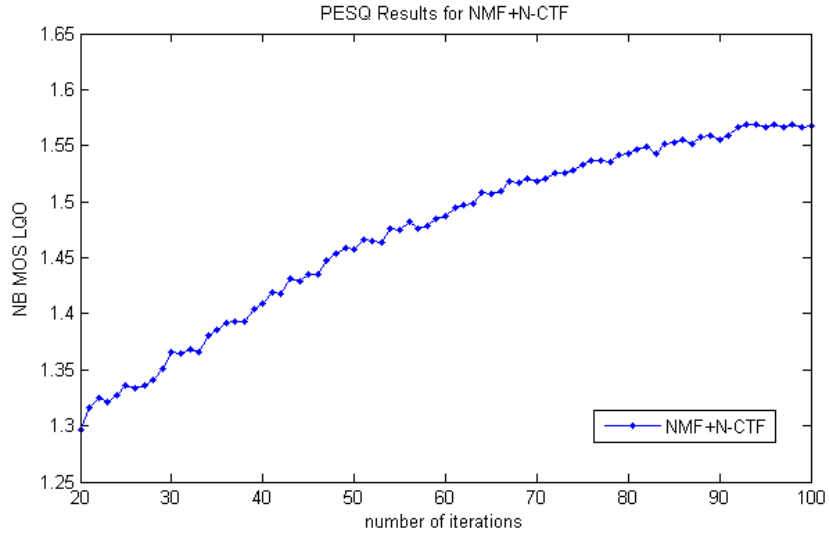


Figure 4.20: PESQ2 Result for NMF+N-CTF

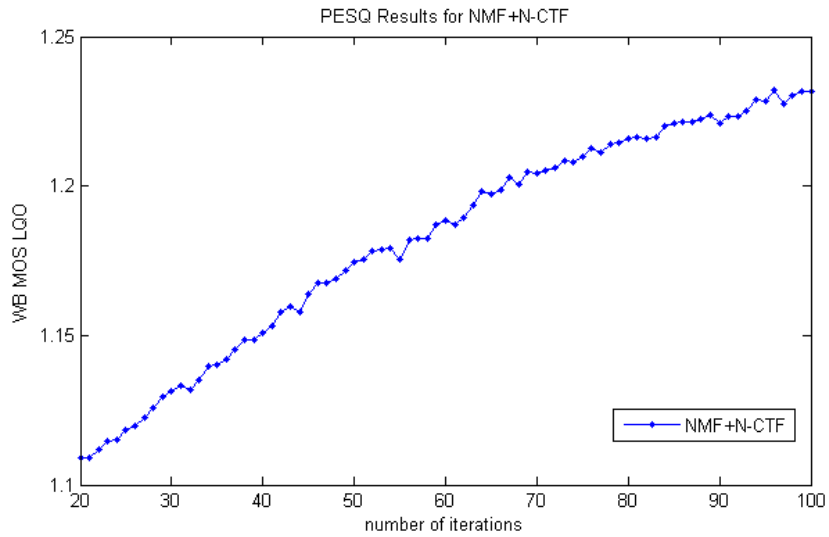


Figure 4.21: PESQ3 Result for NMF+N-CTF

As an addition to these, to pick a proper iteration number for L-WPE and G-WPE methods, we performed an experiment with 20 iterations. The results can be observed below as given in Figures 4.22, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28.

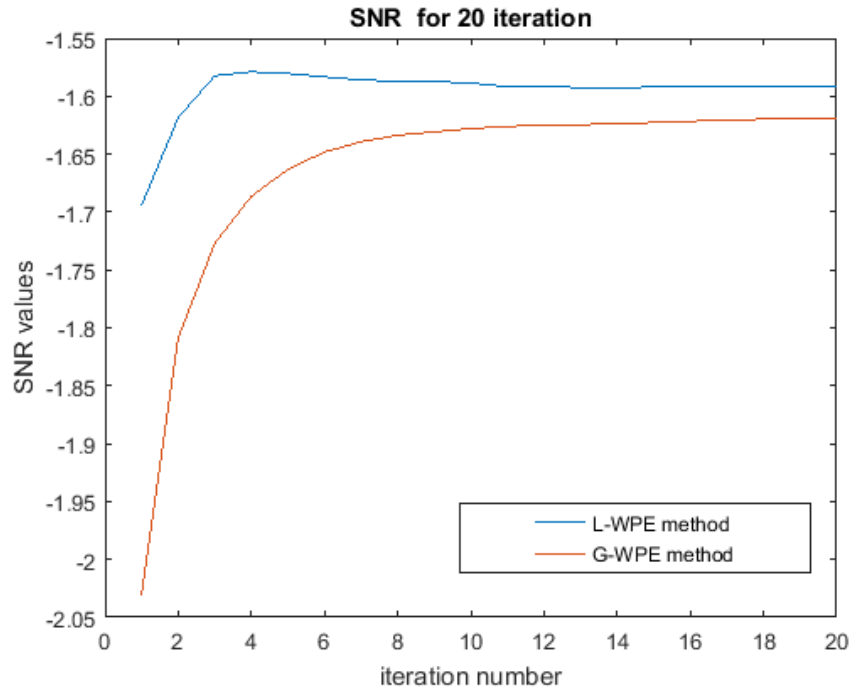


Figure 4.22: SNR for 20 iterations

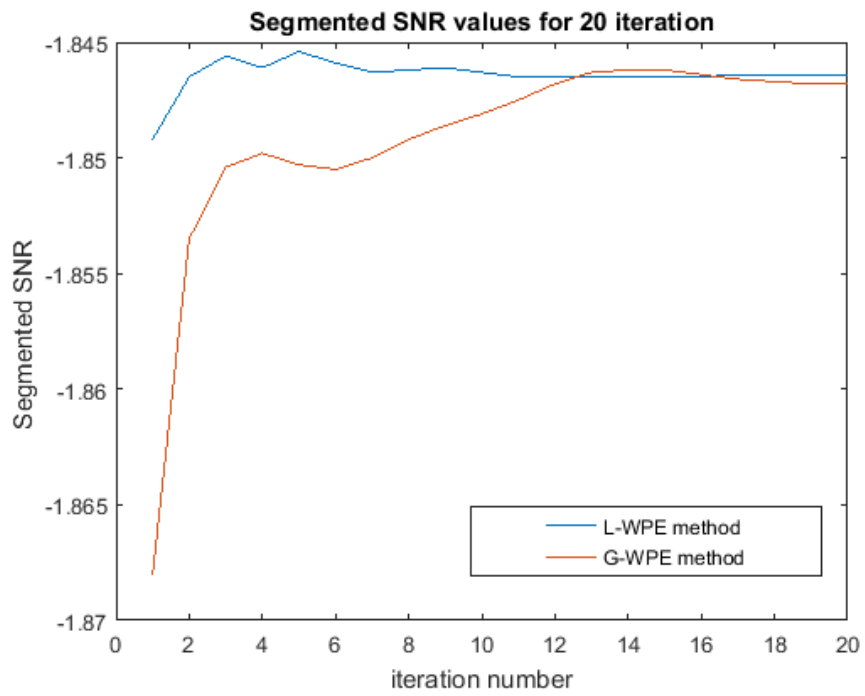


Figure 4.23: Segmented SNR for 20 iterations

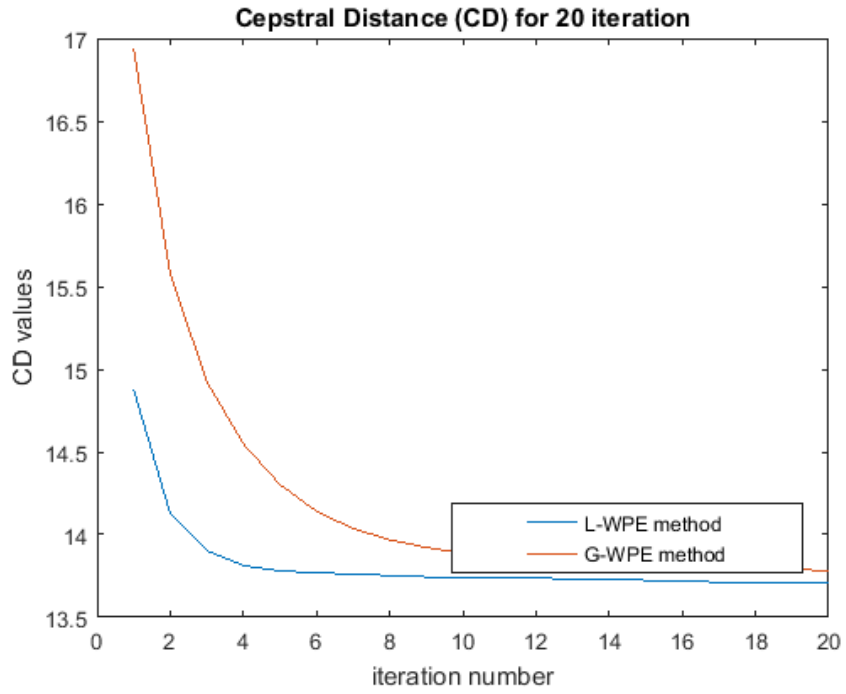


Figure 4.24: Cepstral Distance (CD) for 20 iterations

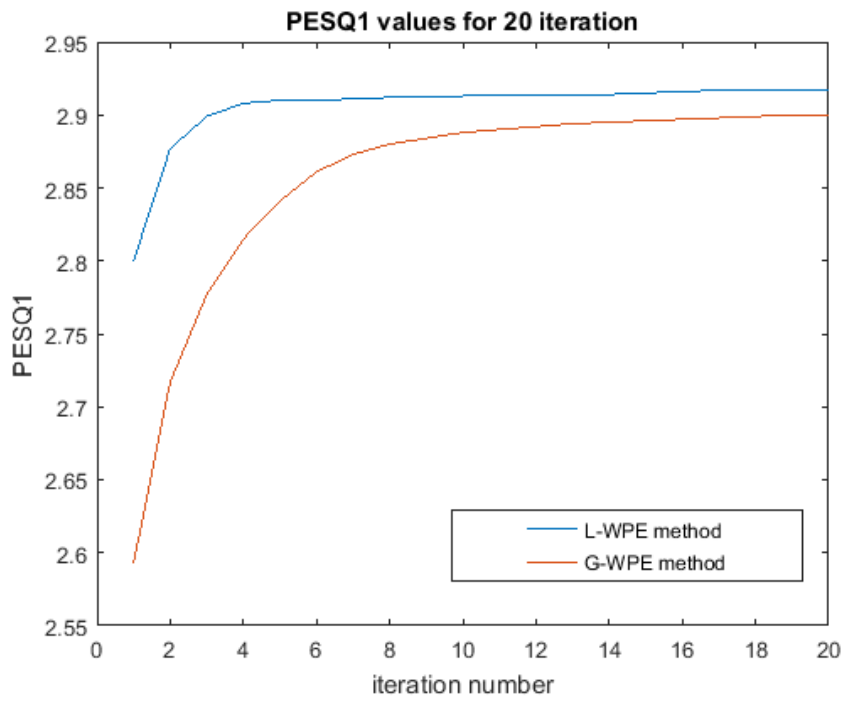


Figure 4.25: PESQ1 values for 20 iterations

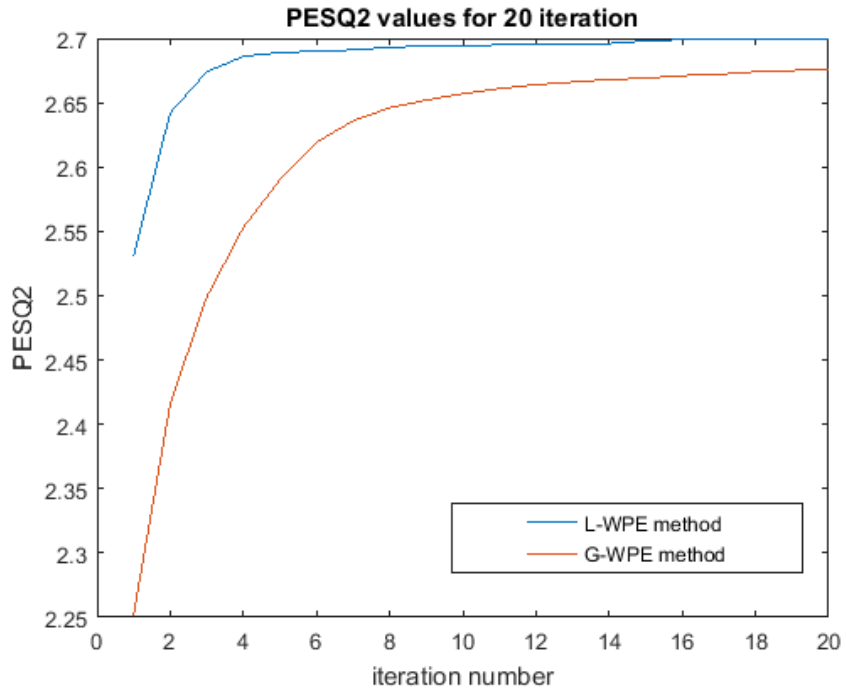


Figure 4.26: PESQ2 values for 20 iterations

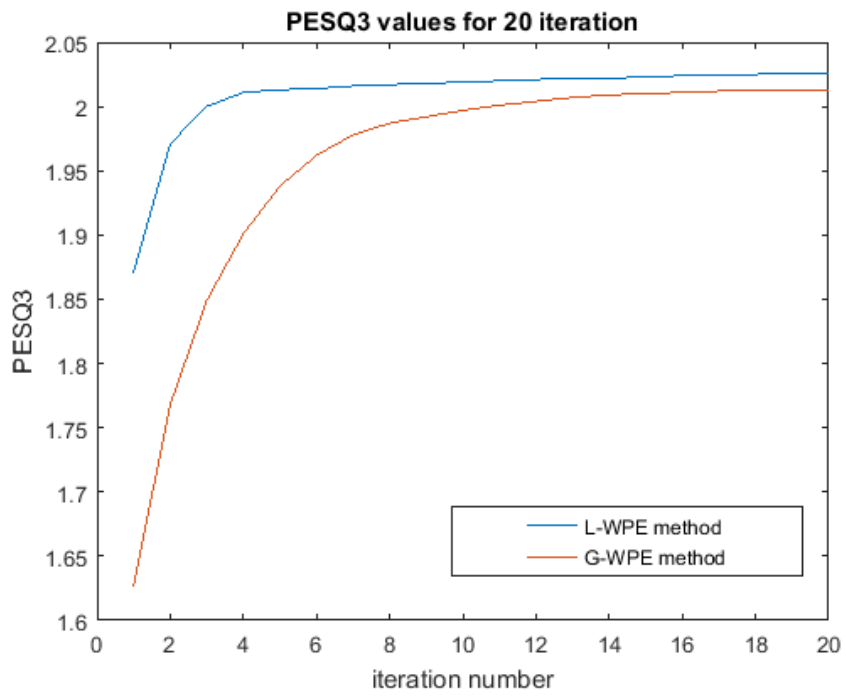


Figure 4.27: PESQ3 values for 20 iterations

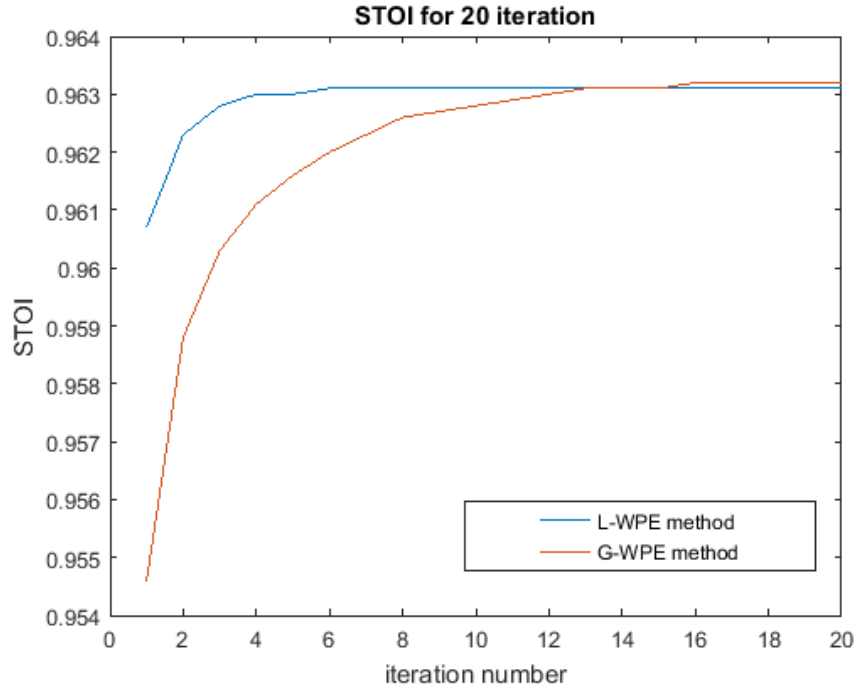


Figure 4.28: STOI for 20 iterations

As seen, 5-6 iterations are usually enough for both methods to stabilize test results.

4.3.3 Robustness against RIR size

For long room impulse with $RT_{60} = 0.54s$ results are as given in Table 4.3. SNR and segmental SNR results are better for SPWLS and NMF+N-CTF methods. On the other hand L-WPE and G-WPE give better results for CD, STOI and PESQ results after 5 iterations to each. For first and second iterations of SPWLS, also we can see some improvements for STOI and PESQ.

Table 4.3: Dereverberation method results for long RIR

Method	SNR	segSNR	CD	PESQ1	PESQ2	PESQ3	STOI
Revb/Clean	-4.386	-1.2268	28.923	1.771	1.47	1.141	0.7915
L-WPE	-2.7747	-0.5622	24.2204	2.0330	1.6590	1.2810	0.8556
G-WPE	-2.8522	-0.5904	24.4234	2.0190	1.6470	1.2710	0.8516
SPWLS	-2.5681	-0.5053	41.6081	2.0400	1.6640	1.2580	0.8391
NMF+CTF	-0.1454	0.0304	33.4180	1.5840	1.3660	1.0890	0.6685
DLP	-3.6398	-0.3382	28.1874	1.8810	1.5430	1.1750	0.8303

When the Table 4.3 and 4.1 are compared, it is clear that long RIR results are worse than the average RIR results as expected. However interestingly, NMF+N-CTF's CD results and SPWLS's CD results are better for long RIR dereverberation case than average CD results. Additionally, SPWLS PESQ results are better than G-WPE different than the average results.

As an addition, five separate experiments are conducted to measure NMF+N-CTF results for different iteration numbers and dictionary sizes. NNCTF1 is the results for NMF+N-CTF with dictionary matrix size 100 and iteration number 100; NNCTF2 is the results for NMF+N-CTF with dictionary matrix size 500 and iteration number 200; NNCTF3 is the results for NMF+N-CTF with dictionary matrix size 1000 and iteration number 200. We made an forth experiment which is called NNCTF4 with dictionary size 1000 and iteration numbers 400 in case of a large dictionary matrix might need more iterations to converge. NNCTF5 is the fifth experiment with iteration number 240 and dictionary size 1000. The results of these experiments can be seen in Table 4.4.

Table 4.4: Dereverberation method results for long RIR

Method	SNR	segSNR	CD	PESQ1	PESQ2	PESQ3	STOI
NNCTF1	-0.1454	0.0304	33.418	1.5840	1.3660	1.0890	0.6685
NNCTF2	-0.6503	-0.0176	30.6301	1.5230	1.3370	1.0840	0.7187
NNCTF3	-0.6517	-0.0084	30.7251	1.4890	1.3210	1.0780	0.7205
NNCTF4	-0.8553	-0.0337	31.0310	1.4850	1.3190	1.0840	0.7070
NNCTF5	-0.5673	0.0036	30.5984	1.5240	1.3370	1.0760	0.7192

As seen from the Table 4.4, CD and STOI results are getting better for NMF+N-CTF when iteration number and dictionary matrix size both are increased in general. On the other hand, SNR, segSNR and PESQ values are getting worse. These four experiments are only conducted for the speech data with $RT_{60} = 0.54s$. Also, as seen in experiment 5 and 4 in Table 4.4, the NMF+N-CTF algorithm does not always converge with respect to iteration number.

4.3.4 Loss function versus iterations of SPWLS method

In this section, loss function will be calculated to analyze the convergence of SPWLS method. To observe convergence behavior, the loss for certain frequency bins are shown in figures below.

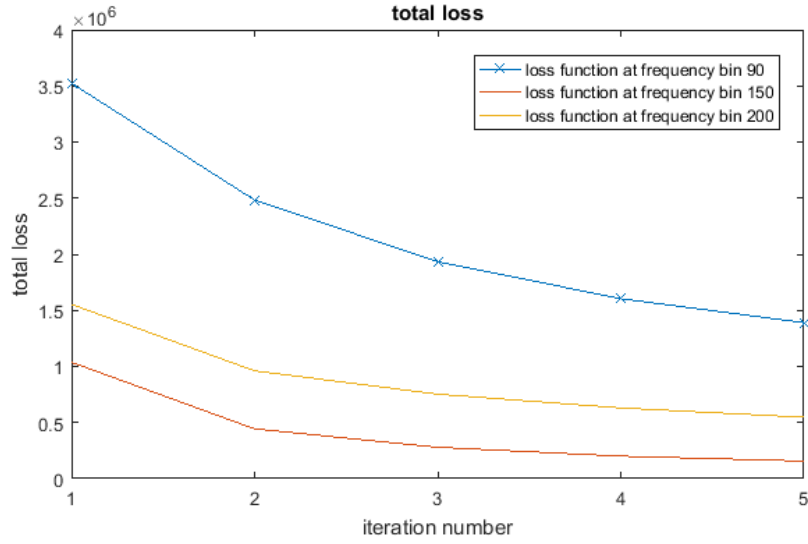


Figure 4.29: Total loss $\|\mathbf{W}(\mathbf{x} - \mathbf{H}\mathbf{s})\|_2^2 + \lambda_s\|\mathbf{s}\|_1 + \lambda_h(\|\mathbf{h}\|_2 - n_h)^2$

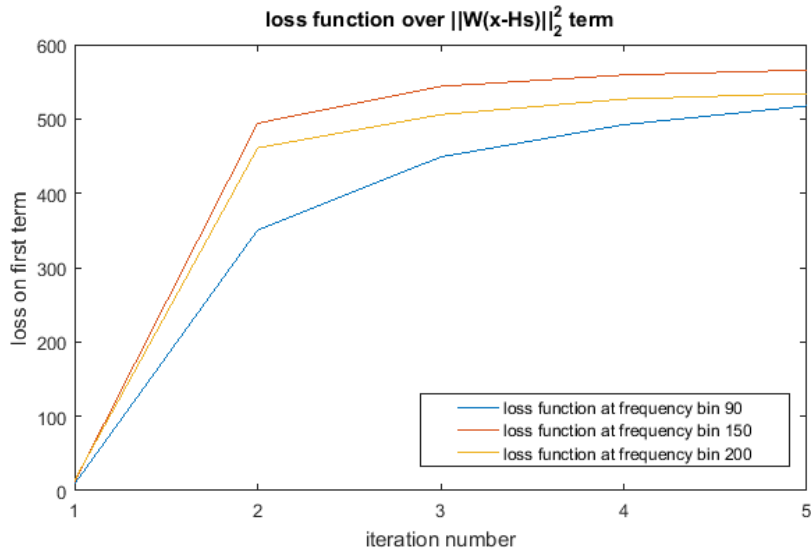


Figure 4.30: Loss function term $\|W(x - Hs)\|_2^2$

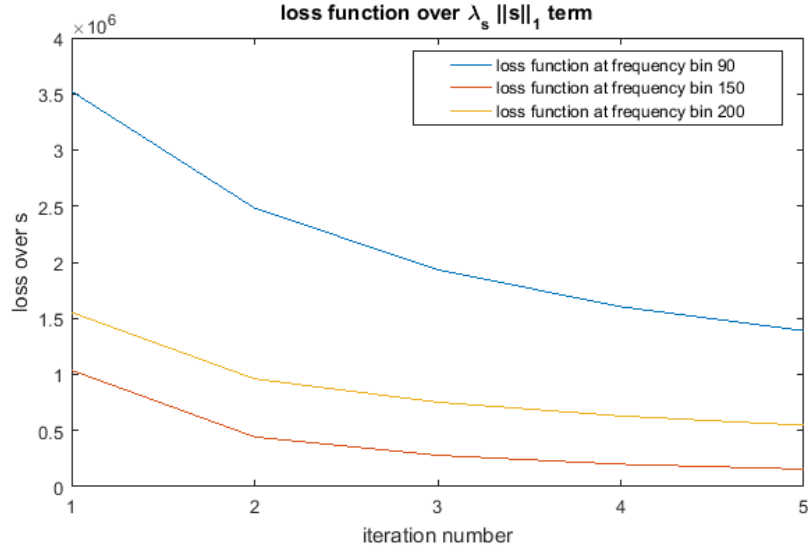


Figure 4.31: Loss function term $\lambda_s ||s||_1$

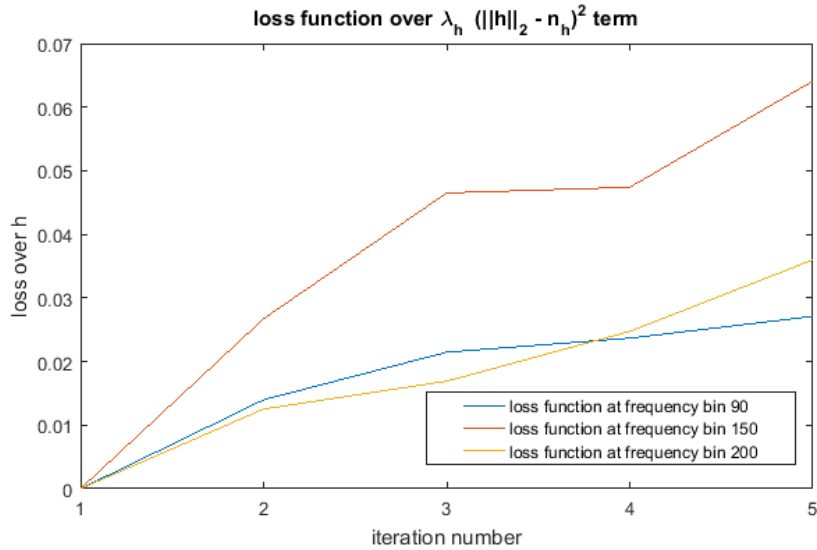


Figure 4.32: Loss function term $\lambda_h (||h||_2 - n_h)^2$

As observed in Figure 4.29, total loss value is decreasing with iterations. So, we can claim that the algorithm works to decrease the total loss value as iterations increase. However, it is not clear whether we are moving towards a more accurate solution. This may be due to the non-convex nature of the problem and also we may need more constraints to obtain more reasonable solutions.

We, also observe the behavior with respect to each term of the loss function. As observed in the figures above, $\|W(x - Hs)\|_2^2$ term is increasing and small; $\lambda_s \|s\|_1$ term is decreasing and large; finally $\lambda_h (\|h\|_2 - n_h)^2$ term is increasing and small. We can see that total loss function is decreasing and it is dominated by the sparsity $\lambda_s \|s\|_1$ term. Decreased loss function indicates that we are solving the defined problem. Sparsity dominating effect can be observed in Figure 4.7, spectrogram results.

Chapter 5

Discussion and Conclusion

5.1 Discussion

As seen from the results in Table 4.1 and 4.2, the best results are obtained by L-WPE method in terms of performance measures. It is the best in terms of SNR, segmental SNR, CD, PESQ values and STOI. As a disadvantage, L-WPE method is really slow. Thus, in terms of time efficiency and test results, G-WPE may be better with real time applications such as ASR.

NMF+N-CTF results are converging as seen from plots in Section 4.3.2. However, the test results are not as good as proposed in paper [6]. To get better results, iteration number can be increased. Also, this method could perform better with a good initialization. In [6], it is proposed to utilize two cases: first one with an online method to learn basis matrix which has dictionary size 100 columns and the 50 iterations, and second one with a supervised basis matrix with dictionary size 4000 columns to start with a good initialization and with same iteration numbers. In our implementation, basis matrix dictionary size is set as 100 without any initialization as proposed for online method in [6] with iteration size 100. Although, we put more iterations, results were not as accurate as the results of other algorithms except for SNR results. On the other hand, increasing the dictionary matrix size and iteration number had positive effects on the results. It must be considered that increasing dictionary size or iteration number increases the computational complexity of the algorithm.

DLP is just utilized to make comparisons with L-WPE and G-WPE methods, since

they are based on DLP method. In comparison results, both gave better results than DLP as expected. L-WPE was slower and G-WPE was faster in comparison to DLP for one iteration.

SPWLS could not show good performance as expected, especially in terms of CD test results. For iterative algorithms, they need to be advanced with well defined constraints. This might be the problem with SPWLS. As seen from Table 4.1 and 4.2, results are not converging all the time. The reason behind it could be getting stuck at local minimum solutions. To improve results, better constraints should be used and coefficients must be set properly. This method needs to be improved. However, it shows promise due to time efficiency, SNR and PESQ results. Also, the intelligibility of the results are still high.

Additionally, spectrogram results show that L-WPE and G-WPE are successfully managing dereverberation since the high similarity between clean and dereverberated signals. Also, listening the results shows that best ones are L-WPE and G-WPE methods, indeed. For NMF+N-CTF, dereverberated sound is nearly impossible to understand due to some background noise and degradation of the speech signal. For SPWLS method, intelligibility of words are good. However, there is a voice like glottal vibrations in the background.

5.2 Conclusion

In conclusion, best test result are obtained with L-WPE method. On the other hand, for efficiency G-WPE is better. If the dereverberation method picked will be combined with real time applications, then we propose to use G-WPE. If there is no need for speed and efficiency, L-WPE is suggested to be utilized.

NMF+N-CTF and SPWLS methods need to be improved. Especially, SPWLS method is promising due to the run-time efficiency and robustness. Also, the results were slightly better for noisy and long RIR cases. However, the algorithm needs to be reevaluated with better constraints and the effect of the sparsity term might be decreased.

Bibliography

- [1] B. E. Kingsbury and N. Morgan, “Recognizing reverberant speech with rasta-plp,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 1259–1262.
- [2] P. A. Naylor and N. D. Gaubitch, *Speech dereverberation*. Springer Science & Business Media, 2010.
- [3] M. Wu and D. Wang, “A two-stage algorithm for one-microphone reverberant speech enhancement,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 774–784, 2006.
- [4] S. Mosayyebpour, M. Esmaeili, and T. A. Gulliver, “Single-microphone early and late reverberation suppression in noisy speech,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 322–335, 2013.
- [5] T. Yoshioka, H. Kameoka, T. Nakatani, and H. G. Okuno, “Statistical models for speech dereverberation,” in *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09. IEEE Workshop on*. IEEE, 2009, pp. 145–148.
- [6] N. Mohammadiha, P. Smaragdis, and S. Doclo, “Joint acoustic and spectral modeling for speech dereverberation using non-negative representations,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4410–4414.
- [7] M. Triki and D. Slock, “Delay and predict equalization for blind speech dereverberation,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5. IEEE, 2006, pp. V–V.

- [8] M. Miyoshi, “Estimating ar parameter-sets for linear-recurrent signals in convolutive mixtures,” *Proc. ICA-03*, pp. 585–589, 2003.
- [9] K. Kinoshita, M. Delcroix, T. Nakatani, and M. Miyoshi, “Suppression of late reverberation effect on speech signal using long-term multiple-step linear prediction,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 4, pp. 534–545, 2009.
- [10] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, “Speech dereverberation based on variance-normalized delayed linear prediction,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 7, pp. 1717–1731, 2010.
- [11] A. Jukic and S. Doclo, “Speech dereverberation using weighted prediction error with laplacian model of the desired signal,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5172–5176.
- [12] T. Nakatani and M. Miyoshi, “Blind dereverberation of single channel speech signal based on harmonic structure,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP’03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–92.
- [13] K. Kinoshita, T. Nakatani, and M. Miyoshi, “Fast estimation of a precise dereverberation filter based on speech harmonicity.” in *ICASSP (1)*, 2005, pp. 1073–1076.
- [14] M. Miyoshi and Y. Kaneda, “Inverse filtering of room acoustics,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 36, no. 2, pp. 145–152, 1988.
- [15] B. Yegnanarayana and P. S. Murthy, “Enhancement of reverberant speech using lp residual signal,” *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 3, pp. 267–281, 2000.
- [16] B. W. Gillespie, H. S. Malvar, and D. A. Florêncio, “Speech dereverberation via maximum-kurtosis subband adaptive filtering,” in *Acoustics, Speech, and Signal*

- Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 6. IEEE, 2001, pp. 3701–3704.
- [17] T. Hikichi, M. Delcroix, and M. Miyoshi, “Inverse filtering for speech dereverberation less sensitive to noise and room transfer function fluctuations,” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 1–12, 2007.
- [18] A. Jukić, T. van Waterschoot, T. Gerkmann, and S. Doclo, “Group sparsity for mimo speech dereverberation,” in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*. IEEE, 2015, pp. 1–5.
- [19] A. Kocanaogullari and I. Bayram, “A dereverberation formulation based on sparsity,” in *Signal Processing and Communications Applications Conference (SIU), 2015 23th*. IEEE, 2015, pp. 1018–1021.
- [20] M. Valente, H. Hosford-Dunn, and R. Roeser, *Audiology: Treatment*, ser. Thieme Publishers Series. Thieme, 2008. [Online]. Available: <https://books.google.com.tr/books?id=9e-kHPXPhhIC>
- [21] E. A. P. Habets, “Single- and multi-microphone speech dereverberation using spectral enhancement,” 2007.
- [22] H. Löllmann, E. Yilmaz, M. Jeub, and P. Vary, “An improved algorithm for blind reverberation time estimation,” in *Proceedings of International Workshop on Acoustic Echo and Noise Control (IWAENC)*, 2010, pp. 1–4.
- [23] N. D. Gaubitch, H. W. Loellmann, M. Jeub, T. H. Falk, P. A. Naylor, P. Vary, and M. Brookes, “Performance comparison of algorithms for blind reverberation time estimation from speech,” in *Acoustic Signal Enhancement; Proceedings of IWAENC 2012; International Workshop on*. VDE, 2012, pp. 1–4.
- [24] R. Ratnam, D. L. Jones, B. C. Wheeler, W. D. O’Brien Jr, C. R. Lansing, and A. S. Feng, “Blind estimation of reverberation time,” *The Journal of the Acoustical Society of America*, vol. 114, no. 5, pp. 2877–2892, 2003.

- [25] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- [26] I. Selesnick, *Introduction to sparsity in signal processing*. NYU-Poly, 2012.
- [27] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [28] T. L. Jensen, D. Giacobello, T. van Waterschoot, and M. G. Christensen, “Fast algorithms for high-order sparse linear prediction with applications to speech processing,” *Speech Communication*, 2015.
- [29] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.
- [30] M. A. Figueiredo and R. D. Nowak, “An em algorithm for wavelet-based image restoration,” *Image Processing, IEEE Transactions on*, vol. 12, no. 8, pp. 906–916, 2003.
- [31] P. L. Combettes and J.-C. Pesquet, “Proximal splitting methods in signal processing,” in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.
- [32] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [33] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, “Fast image recovery using variable splitting and constrained optimization,” *Image Processing, IEEE Transactions on*, vol. 19, no. 9, pp. 2345–2356, 2010.
- [34] —, “An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems,” *Image Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 681–695, 2011.

- [35] P. Stoica and R. L. Moses, *Spectral analysis of signals*. Pearson/Prentice Hall Upper Saddle River, NJ, 2005.
- [36] J. Makhoul, “Linear prediction: A tutorial review,” *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.
- [37] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in neural information processing systems*, 2001, pp. 556–562.
- [38] C. Févotte, N. Bertin, and J.-L. Durrieu, “Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis,” *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.
- [39] E. M. G. Girgis, “Incorporating prior information in nonnegative matrix factorization for audio source separation,” Ph.D. dissertation, SABANCI UNIVERSITY, 2013.
- [40] J.-L. Durrieu, G. Richard, B. David, and C. Févotte, “Source/filter model for unsupervised main melody extraction from polyphonic audio signals,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 3, pp. 564–575, 2010.
- [41] R. Martin, “Speech enhancement based on minimum mean-square error estimation and supergaussian priors,” *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 845–856, 2005.
- [42] J.-H. Chang, “Complex laplacian probability density function for noisy speech enhancement,” *IEICE Electronics Express*, vol. 4, no. 8, pp. 245–250, 2007.
- [43] B. Lee, T. Kalker, and R. W. Schafer, “Maximum-likelihood sound source localization with a multivariate complex laplacian distribution,” in *Proc. Internat. Workshop. Acoust. Echo and Noise Control*. Citeseer, 2008.
- [44] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.

- [45] ———, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.
- [46] N. Mohammadiha and S. Doclo, “Speech dereverberation using non-negative convolutive transfer function and spectro-temporal modeling,” *IEEE/ACM Transactions on Audio, Speech, And Language Processing*, vol. 24, no. 2, pp. 276–289, February 2016.
- [47] H. Kameoka, T. Nakatani, and T. Yoshioka, “Robust speech dereverberation based on non-negativity and sparse nature of speech spectrograms,” in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 45–48.
- [48] K. Kinoshita, M. Delcroix, S. Gannot, E. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, A. Sehr, and T. Yoshioka, “A summary of the reverb challenge: state-of-the-art and remaining challenges in reverberant speech processing research,” *EURASIP Journal on Advances in Signal Processing*, vol. 10, no. 1186, pp. 13 634–016–0306–6, 2016.
- [49] S. Furui, “Digital speech processing, synthesis, and recognition (revised and expanded),” *Digital Speech Processing, Synthesis, and Recognition (Second Edition, Revised and Expanded)*, 2000.
- [50] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “An algorithm for intelligibility prediction of time–frequency weighted noisy speech,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 7, pp. 2125–2136, 2011.
- [51] Y. Hu and P. C. Loizou, “Evaluation of objective quality measures for speech enhancement,” *IEEE Transactions on audio, speech, and language processing*, vol. 16, no. 1, pp. 229–238, 2008.