BENCH-family algorithms for multichannel adaptive speech dereverberation in the frequency domain

Kamal Mustafa

supervised by
Dr. James Hopgood

Masters Thesis

August 19, 2016

Abstract

This document records the efforts of implementing the BENCH-family algorithms proposed by Schmid *et al* in [1]. The BENCH algorithms family comprises of three dereverberation solutions defined in the context of blind speech dereverberation. These algorithms are defined with concordance to basic parameter learning methods namely Maximum Likelihood (ML), Maximum-A-Posteriori (MAP) and finally Variational Bayesian (VB) techniques. In this document, the three algorithms are reviewed. In addition, the algorithms are implemented and experimented on synthetic simulation environments. Several objective measures are applied to measure the performance of the algorithms. In addition, subjective measures are also employed in the form of comparison spectrograms.

Acknowledgements

I would like to express my deepest appreciation to my thesis advisor Dr. James Hopgood for his continuous support. Dr. Hopgood's office was always open to me whenever I had trouble with my research. In addition, he steered me in the right direction whenever he thought I needed it. I am also grateful to Dr. Mike Davies for providing me with the necessary literature at the start of my thesis. I would also like to thank Prof. Dr. Habets for the usage of his RIR generator code and Dr. Loizou for using his code for objective testing. Finally, I would like to take the opportunity to thank my family and friends for being able to provide me with unfailing support and encouragement throughout my life and through this thesis.

August, 2016,
Kamal Mustafa

Contents

List of Figures

Chapter 1

Introduction

Not so long ago, speech/word recognition was a very arduous signal processing problem where research did not yield impressive results. Influential researchers such as John Pierce even compared it to "curing cancer and turning dust to gold" [2]. However, with the advances in acoustic and language modelling and more specifically the development of Hidden Markov Models (HMMs), research in speech recognition bloomed. This led it to be an integral part of our daily lives with plenty of applications in major fields such as healthcare, transportation, telephony, education,etc [3].

In order to improve the performance of speech recognition systems, the objective speech signal should not suffer major attenuations or spectral corruption. However, in reality, speech signals are susceptible to major spoiling factors such as background noise, reverberation, echoes, and amplitude attenuation. *Speech enhancement* encompasses removing all these effects thus improving the overall intelligibility and pleasantness of the speech signal. Signal enhancement is not only beneficial in the speech recognition domain. It facilitates convenient interpersonal communication especially in mobile phones, teleconferencing, etc [4].

In this document, we are particularly interested in speech enhancement using statistical methods. However, it would be beneficial to first outline the noise that we are trying to eliminate. The term "noise" is used to define any unwanted signal which decreases the quality of the speech signal. However, this broad definition might be quite misleading since it does not outline many of the underlying technical aspects of the problem at hand. A more reliable definition can be obtained by dividing the term "noise" into four different subcategories namely, *additive noise*, *interfering signals*, *reverberation* and *echo*. Additive noise encompasses all the sounds originating from neighbouring ambient noise sources such as people speaking, ventilation, car noises, etc. However, interfering signals are produced by other competing speakers. The main difference between both is that additive noise is usually has an approximately flat spectrum thus no emphasis on a particular frequency. The latter however, has a structured spectrum. Reverberation on the other hand is a spectral distortion produced by multichannel propagation due to an enclosure. Finally, an echo is a result of a reflected wave arriving several milliseconds after the direct speech signal [5]. Eliminating each of the mentioned corrupting factors is a vibrant field of audio processing research. In this thesis document, particular interest is paid towards removing the spectral colouration introduced by reverberation without any prior knowledge of the source speech signal. In other words, this is an exercise of *Blind dereverberation* which is an interesting signal processing problem. There are different methods of blind dereverberation. Namely, single channel blind dereverberation where only a single microphone is used in the process. This method is becoming increasingly popular due to the gradual decrease in the size of everyday gadgets starting from mobile phones and ending with hearing aids [6]. Another method of speech dereverberation is via the exploitation of spatial diversity to blindly estimate the source signal. The latter method is going to be implemented in this document. There are several approaches to multichannel dereverberation. One of the established methods is beamforming in which signals from a specific direction are combined together while interfering signals from other directions and noise are suppressed. Examples of famous beamformers are the Delay and Sum Beamformer (DSB), Linearly Constrained Minimum Variance (LCMV) and Minimum Variance Distortionless Response (MVDR) [7]. Other super-resolution direction of arrival (DOA) estimation techniques which are used to enhance beamformers include the MUltiple SIgnal

Classification (MUSIC) algorithm [8] and Estimation of Signal Parameters via Rotation Invariance Techniques (ESPIRIT) algorithm [9].

A model for the reverberant signal using frame based definitions is defined in Chapter 2. In addition, the first algorithm known as MLBENCH algorithm will be discussed in Chapter 3. Furthermore, the MAP extension of this algorithm titled MAPBENCH algorithm is reviewed in chapter 4. This algorithm is also extended to a variational bayesian definition in Chapter 5. Finally, the algorithms are implemented and tested in a simulated environment in Chapter 6.

Chapter 2

Multichannel frame-based frequency domain observation model

The problem of multichannel reverberation could be conveniently conceived by a Single Input Multiple Output (SIMO) model. In order to realise the constantly changing Room Impulse Responses (RIR), each RIR is modelled as linear time-varying system. The clean speech signal s_k is transmitted into a bank of RIRs of length L [10] and background noise is added. This process can be expressed as:

$$\mathbf{y}_{i,k} = \mathbf{h}_{i,k}^T \mathbf{s}'_k + \mathbf{n}_{i,k}, \qquad i = 1 \ ... \ P \tag{2.1}$$

where $\mathbf{h}_{i,k}$ is the ith RIR at discrete time k. In addition, the vector $\mathbf{s}'_k$ constitutes the most recent L samples of the clean signal s_k. The additive noise at the ith channel and discrete time k is represented by $\mathbf{n}_{i,k}$. Since the length of a practical RIR tends to be very long which leads to increased latency and memory problems [11], frame based definitions of the above terms are used for better computational efficiency [12]. To begin with, the frame based source vector of length M is defined as:

$$\mathbf{s}_\tau = [s_{\tau R-M+1} \ s_{\tau R-M+2} \ \dots \ s_{\tau R}]^T \tag{2.2}$$

Where the source vector above is overlapped by R samples. The reason of the overlap is to implement an overlap-save convolution defined in [12] and [11]. Multiplying it by a DFT matrix $\mathbf{F}_M$ will render it to the frequency domain such as:

$$\underline{\mathbf{s}}_\tau = \mathbf{F}_M \mathbf{s}_\tau \tag{2.3}$$

In order to realise time variation in channels, we model L non-zero channel coefficients $\mathbf{w}_{i,\tau}$. Next, the channel coefficients are padded with R zeros such that $L + R = M$. The padded channel frame is then converted to the frequency domain upon multiplication by the DFT matrix as shown below:

$$\underline{\mathbf{w}}_{\mathbf{i},\tau} = \mathbf{F}_M [\mathbf{w}_{i,\tau}^T \ \mathbf{0}_{R\times 1}^T]^T \tag{2.4}$$

We next impose time-variation via incorporating first order Markov property to the frequency domain channel vector in (2.4):

$$\underline{\mathbf{w}}_{i,\tau} = A \cdot \underline{\mathbf{w}}_{i,\tau-1} + \Delta\underline{\mathbf{w}}_{i,\tau} \tag{2.5}$$

Where A is the state transition coefficient which falls in the range $0 < A < 1$. The term $\Delta\underline{\mathbf{w}}_{i,\tau}$ is the process noise which is zero mean and frame-wise uncorrelated. In addition, the process noise is normally distributed with an $M \times M$ covariance matrix $\underline{\mathbf{\Psi}}_{i,\tau}^{\Delta} = \mathbb{E}\{\Delta\underline{\mathbf{w}}_{i,\tau}\Delta\underline{\mathbf{w}}_{i,\tau}^H\}$.
Upon imposing the first order Markov model to the channel, the frame-based version of (2.1) could be written as:

$$\mathbf{y}_{i,\tau} = \mathbf{Q}^T \mathbf{F}_M^{-1} \underline{\mathbf{W}}_{i,\tau} \ \underline{\mathbf{s}}_\tau + \mathbf{n}_{i,\tau}, \tag{2.6}$$

where $\mathbf{Q} = [\mathbf{0}_{R\times L} \ \mathbf{I}_R]^T$ is an overlap save constraint of size $M \times R$ used to remove cyclic convolution effects introduced by the DFT convolution. In addition, $\underline{\mathbf{W}}_{i,\tau} = \text{diag}\{\underline{\mathbf{w}}_{i,\tau}\}$ is defined to enable an

element-wise multiplication with the overlapped source vector $\underline{\mathbf{s}}_\tau$. Finally, $\mathbf{n}_{i,\tau}$, is the time-domain additive noise vector of size R which is defined as:

$$\mathbf{n}_{i,\tau} = [n_{i,\tau R-R+1} \;\; n_{i,\tau R-R+2} \;\cdots\; n_{i,\tau R} \,]^T \tag{2.7}$$

Eventually, we can say that (2.6) encompasses the frame-based time domain observation model. However, we are interested in formulating a frequency domain frame-based observation model. This could be done by transforming (2.6) to the frequency domain. By padding the reverberant signal vector $\mathbf{y}_{i,\tau}$ by L zeroes and applying the DFT matrix we get:

$$\underline{\mathbf{y}}_{i,\tau} = \mathbf{F_M Q} \mathbf{y}_{i,\tau} = \mathbf{F_M Q Q}^T \mathbf{F}_M^{-1} \underline{\mathbf{W}}_{i,\tau} \, \underline{\mathbf{s}}_\tau + \mathbf{F_M Q} \mathbf{n}_{i,\tau}, \tag{2.8}$$

Where $\underline{\mathbf{y}}_{i,\tau}$ denotes the frequency domain single channel observation model. The above equation could be written in a more compact form. If we define $\underline{\boldsymbol{\mathcal{W}}}_{i,\tau} = \mathbf{T} \, \underline{\mathbf{W}}_{i,\tau}$ where $\mathbf{T} = \mathbf{F_M Q Q}^T \mathbf{F}_M^{-1}$, (2.8) could be rewritten as:

$$\underline{\mathbf{y}}_{i,\tau} = \underline{\boldsymbol{\mathcal{W}}}_{i,\tau} \underline{\mathbf{s}}_\tau + \underline{\mathbf{n}}_{i,\tau}, \tag{2.9}$$

where $\underline{\boldsymbol{\mathcal{W}}}_{i,\tau}$ is the overlap-save constrained channel matrix of size $M \times M$ and $\mathbf{T}$ is an $M \times M$ overlap-save constraint. Furthermore, the vector $\underline{\mathbf{n}}_{i,\tau}$ is the frequency domain observation noise defined as $\underline{\mathbf{n}}_{i,\tau} = \mathbf{F_M Q} \mathbf{n}_{i,\tau}$. According to the central limit theorem [13], the observation noise is modelled to be zero-mean, Gaussian distributed with an $M \times M$ diagonal covariance matrix $\underline{\boldsymbol{\Psi}}^{\mathbf{n}}{}_{i,\tau} = \mathbb{E}\{\underline{\mathbf{n}}_{i,\tau} \underline{\mathbf{n}}_{i,\tau}^H\}$. Above equations could be extended to a multichannel representation. Upon the definition of stacked quantities:

$$\underline{\mathbf{y}}_\tau = [\underline{\mathbf{y}}_{1,\tau}^T \, \underline{\mathbf{y}}_{2,\tau}^T \cdots \, \underline{\mathbf{y}}_{P,\tau}^T]^T \tag{2.10}$$

$$\underline{\boldsymbol{\mathcal{W}}}_\tau = [\underline{\boldsymbol{\mathcal{W}}}_{1,\tau} \, \underline{\boldsymbol{\mathcal{W}}}_{2,\tau} \cdots \, \underline{\boldsymbol{\mathcal{W}}}_{P,\tau}]^T \tag{2.11}$$

$$\underline{\mathbf{n}}_\tau = [\underline{\mathbf{n}}_{1,\tau}^T \, \underline{\mathbf{n}}_{2,\tau}^T \cdots \, \underline{\mathbf{n}}_{P,\tau}^T]^T \tag{2.12}$$

The multichannel representation of the frequency domain block model is therefore:

$$\underline{\mathbf{y}}_\tau = \underline{\boldsymbol{\mathcal{W}}}_\tau \, \underline{\mathbf{s}}_\tau + \underline{\mathbf{n}}_\tau \tag{2.13}$$

An alternative representation of (2.13) could be inferred by applying the diagonally stacked overlap-save constraint $\mathbf{T}_{PM} = \mathbf{I}_P \otimes \mathbf{T}$ of size $PM \times PM$ to the diagonally stacked source matrix defined as $\underline{\mathbf{S}}_\tau = \mathbf{I}_P \otimes \text{diag}\{\underline{\mathbf{s}}_\tau\}$ such as $\underline{\boldsymbol{\mathcal{S}}}_\tau = \mathbf{T}_{PM} \underline{\mathbf{S}}_\tau$. This representation would allow an overlap-save convolution between the overlap-save constrained source matrix and each of the P channels defined in the stacked channel vector $\underline{\mathbf{w}}_\tau$:

$$\underline{\mathbf{w}}_\tau = [\underline{\mathbf{w}}_{1,\tau}^T \, \underline{\mathbf{w}}_{2,\tau}^T \cdots \, \underline{\mathbf{w}}_{P,\tau}^T]^T \tag{2.14}$$

Thus (2.13) could be rewritten as:

$$\underline{\mathbf{y}}_\tau = \underline{\boldsymbol{\mathcal{S}}}_\tau \, \underline{\mathbf{w}}_\tau + \underline{\mathbf{n}}_\tau \tag{2.15}$$

In the next section, the equations described above will be the basis of formulating the BENCH family dereverberation algorithms.

Chapter 3

Maximum Likelihood Blind Equalization aNd CHannel identification (MLBENCH) algorithm

In this section, the MLBENCH algorithm will be defined. The detailed derivation of the algorithm's equations will not be addressed here. However, interested readers can check [1]. The MLBENCH algorithm comprises of two stages, namely, a Least Mean Squares (LMS) equalizer coupled with a sequential expectation maximization (EM) algorithm. The EM algorithm stage is used to learn the channel coefficients along with noise parameters. Next, these parameters are fed into the LMS equalizer to remove reverberation and noise effects. The process is repeated for all reverberant signal frames.
To begin with, the algorithm assumes explicit stationarity of the reverberant signal. Since speech signals are non-stationary in general, one of the advantages of the frame-based frequency domain model is that it exploits the block stationarity property of a speech signal. In other words, the statistics of a speech signal experience quasi-stationarity for around 20-50 ms [14]. According to the frequency domain frame based model equations (2.13) and (2.15), if we assume that the corresponding terms $\underline{\boldsymbol{\mathcal{W}}}_\tau \ \underline{\mathbf{s}}_\tau$ and $\underline{\boldsymbol{\mathcal{S}}}_\tau \ \underline{\mathbf{w}}_\tau$ are deterministic variables, the reverberant signal will have the same distribution as the frequency domain observation noise frame $\underline{\mathbf{n}}_\tau$ which has a zero mean complex normal distribution with covariance matrix $\underline{\mathbf{\Psi}}_\tau^n$ as shown below:

$$p(\underline{\mathbf{n}}_\tau) = \mathcal{CN}(\underline{\mathbf{n}}_\tau | \mathbf{0}_{PM\times 1}, \underline{\mathbf{\Psi}}_\tau^n) \tag{3.1}$$

Rewriting (2.13) as $\underline{\mathbf{n}}_\tau = \underline{\mathbf{y}}_\tau - \underline{\boldsymbol{\mathcal{W}}}_\tau \ \underline{\mathbf{s}}_\tau$ and (2.15) as $\underline{\mathbf{n}}_\tau = \underline{\mathbf{y}}_\tau - \underline{\boldsymbol{\mathcal{S}}}_\tau \ \underline{\mathbf{w}}_\tau$ would show that the distribution of $\underline{\mathbf{y}}_\tau$ given $\underline{\mathbf{w}}_\tau$, $\underline{\mathbf{s}}_\tau$ and noise parameters $\underline{\mathbf{\Theta}}_\tau = \{\underline{\mathbf{\Psi}}_{i,\tau}^n, \underline{\mathbf{\Psi}}_{i,\tau}^\Delta\}$ is a complex gaussian distribution as shown below:

$$p(\underline{\mathbf{y}}_\tau \mid \underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau, \widehat{\underline{\mathbf{\Theta}}}_\tau) = \mathcal{CN}(\underline{\mathbf{y}}_\tau | \underline{\boldsymbol{\mathcal{W}}}_\tau \ \underline{\mathbf{s}}_\tau, \widehat{\underline{\mathbf{\Psi}}}_\tau^n) \tag{3.2}$$

As a result, each reverberant signal frame would have nearly constant statistical parameters which could be learned using either iterative or sequential learning algorithms. In order to facilitate real-time dereverberation, a sequential EM algorithm is chosen to perform the parameter learning task. In the next section, parameter leaning using ML techniques will be explained.

3.1 Online ML-parameter learning

Generally, in parameter learning of state space models, the EM algorithm iterates between state estimation (E-step) and parameter estimation (M-step) this is done to maximize the lower bound of the log-likelihood [15]. The log likelihood function that we are interested in maximizing is shown below:

$$\mathcal{L}(\underline{\mathbf{s}}_\tau, \underline{\mathbf{\Theta}}_\tau) = \ln p(\underline{\mathbf{y}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \underline{\mathbf{s}}_\tau, \underline{\mathbf{\Theta}}_\tau) = \ln \int p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \underline{\mathbf{s}}_\tau, \underline{\mathbf{\Theta}}_\tau) \mathrm{d}\underline{\mathbf{w}}_\tau \tag{3.3}$$

where $\underline{\mathbf{s}}_\tau$ is clean signal source frame, and $\underline{\boldsymbol{\Theta}}_\tau$ is the model parameter set consisting of the noise parameters and $\underline{\mathbf{w}}_\tau$ is the channel frame introduced to the log likelihood function via marginalization. The joint distribution under the integral in equation (3.3) is conditioned on all the previous observations. Introducing a random distribution $q_w(\underline{\mathbf{w}}_\tau)$ to the log likelihood function in (3.3) can help formulate a lower bound function via the usage of Jensen's inequality. This step can be shown below:

$$\ln \int q_w(\underline{\mathbf{w}}_\tau) \frac{p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \underline{\mathbf{s}}_\tau, \underline{\boldsymbol{\Theta}}_\tau)}{q_w(\underline{\mathbf{w}}_\tau)} \mathrm{d}\underline{\mathbf{w}}_\tau \geq \int q_w(\underline{\mathbf{w}}_\tau) \ln \frac{p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \underline{\mathbf{s}}_\tau, \underline{\boldsymbol{\Theta}}_\tau)}{q_w(\underline{\mathbf{w}}_\tau)} \mathrm{d}\underline{\mathbf{w}}_\tau \tag{3.4}$$

The function on the right is the lower bound for the log likelihood defined above. The E-step seeks the stationary point on the lower bound function while the M-step finds the parameters that maximize the lower bound. This can be summarized in the equations below:

$$\begin{aligned} \textbf{E-step}: \quad & q_k \longleftarrow \underset{q}{\operatorname{argmax}} \ \mathcal{F}(q, \underline{\boldsymbol{\Theta}}_{\tau-1}) \\ \textbf{M-step}: \quad & \underline{\boldsymbol{\Theta}}_{\tau,k} \longleftarrow \underset{\underline{\boldsymbol{\Theta}}_\tau}{\operatorname{argmax}} \ \mathcal{F}(q_k, \underline{\boldsymbol{\Theta}}_\tau) \end{aligned} \tag{3.5}$$

In the next section, the Kalman filtering (E-Step) will be explained.

3.1.1 E-Step: Frequency domain adaptive Kalman filter

In this step, the parameters $\hat{\underline{\mathbf{s}}}_\tau$ and $\widehat{\underline{\boldsymbol{\Theta}}}_\tau$) are assumed to have been estimated from the previous M-step. Using bayes rule, the joint distribution in (3.3) can be factorized to a posterior and likelihood distributions as shown below:

$$p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \hat{\underline{\mathbf{s}}}_\tau, \widehat{\underline{\boldsymbol{\Theta}}}_\tau) = p(\underline{\mathbf{w}}_\tau | \underline{\mathbf{y}}_{1:\tau}, \hat{\underline{\mathbf{s}}}_\tau, \widehat{\underline{\boldsymbol{\Theta}}}_\tau) p(\underline{\mathbf{y}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \hat{\underline{\mathbf{s}}}_\tau, \widehat{\underline{\boldsymbol{\Theta}}}_\tau) \tag{3.6}$$

Since the reverberant signal is deemed Gaussian, a recursive posterior estimator in the form of Kalman filter equations is used to estimate the parameters of the posterior distribution $p(\underline{\mathbf{w}}_\tau | \underline{\mathbf{y}}_{1:\tau}, \hat{\underline{\mathbf{s}}}_\tau, \widehat{\underline{\boldsymbol{\Theta}}}_\tau)$ in (3.5). The derivation of the Kalman filter equations in the context of speech dereverberation will not be addressed in detail however interested readers can check [16]. The diagonalized Kalman filter prediction equations are listed below:

$$\widehat{\underline{\mathbf{w}}}^+_{\tau-1} = A \widehat{\underline{\mathbf{w}}}_{\tau-1} \tag{3.7}$$

$$\widehat{\underline{\mathbf{P}}}^+_{\tau-1} = A^2 \, \underline{\mathbf{P}}_{\tau-1} + \underline{\boldsymbol{\Psi}}^{\Delta}_\tau \tag{3.8}$$

Where $\widehat{\underline{\mathbf{w}}}_{\tau-1}$ and $\underline{\mathbf{P}}_{\tau-1}$ encompass previous frame frequency domain stacked channels estimates and frequency domain error stacked covariance matrix respectively. The prediction equations above realize an extrapolation of the unknown channel coefficients based on the first order Markov property imposed in (2.5). The terms $\widehat{\underline{\mathbf{w}}}^+_{\tau-1}$ and $\widehat{\underline{\mathbf{P}}}^+_{\tau-1}$ define the predicted stacked channels vector and the predicted error stacked covariance matrix respectively. These predictions are fed into the set of correction equations shown below:

$$\underline{\boldsymbol{\mu}}_\tau = \underline{\mathbf{P}}^+_{\tau-1} \left(\widehat{\underline{\mathbf{S}}}_\tau \underline{\mathbf{P}}^+_{\tau-1} \widehat{\underline{\mathbf{S}}}^H_\tau + \frac{M}{R} \underline{\boldsymbol{\Psi}}^n_\tau \right)^{-1} \tag{3.9}$$

$$\underline{\mathbf{e}}_\tau = \underline{\mathbf{y}}_\tau - \mathbf{T}_{PM} \widehat{\underline{\mathbf{S}}}_\tau \widehat{\underline{\mathbf{w}}}^+_{\tau-1} \tag{3.10}$$

$$\widehat{\underline{\mathbf{w}}}_\tau = \widehat{\underline{\mathbf{w}}}^+_{\tau-1} + \underline{\boldsymbol{\mu}}_\tau \widehat{\underline{\mathbf{S}}}^H_\tau \underline{\mathbf{e}}_\tau \tag{3.11}$$

$$\underline{\mathbf{P}}_\tau = \underline{\mathbf{P}}^+_{\tau-1} - \frac{R}{M} \underline{\boldsymbol{\mu}}_\tau \widehat{\underline{\mathbf{S}}}^H_\tau \widehat{\underline{\mathbf{S}}}_\tau \underline{\mathbf{P}}^+_{\tau-1} \tag{3.12}$$

The term $\underline{\boldsymbol{\mu}}_\tau$ could be thought of as a near optimal $PM \times PM$ step size matrix for the above Kalman filter equations. It controls the adaptation of the above algorithm. In addition, the term $\underline{\mathbf{e}}_\tau$ signifies the error between the latest frequency domain stacked observation frame and the overlap save convolution between the stacked source matrix and the predicted stacked channel vector. Once $\widehat{\underline{\mathbf{S}}}_\tau$ and $\widehat{\underline{\mathbf{w}}}^+_{\tau-1}$ in second term of (3.10) are correct, the recursive posterior estimator reaches a stationary point in the lower bound function defined in (3.4). (5.11) represents the update equation for the channel posterior estimates $\widehat{\underline{\mathbf{w}}}_\tau$. In the second term of the equation, $\underline{\boldsymbol{\mu}}_\tau \widehat{\underline{\mathbf{S}}}^H_\tau$ represents the Kalman gain matrix $\underline{\mathbf{K}}_\tau$. If the Kalman gain is high, the Kalman filter places more emphasis on the stacked reverberant signal observation frame. On the other hand, if the gain is low, more emphasis is placed on the estimated model parameters $\widehat{\underline{\mathbf{w}}}_\tau$ and $\underline{\mathbf{P}}_\tau$. In fact, the values in the Kalman gain matrix fall in the range of zero to one. At gain values equal one, the Kalman filter entirely ignores estimated parameters. At the other extremity, the Kalman filter totally ignores the observations. Finally, (5.12) demonstrates the update rule for the stacked error covariance matrix based on the predictions and the Kalman gain. The above equations encompass the diagonalized Kalman filter which estimates the model parameters of the posterior $p(\underline{\mathbf{w}}_\tau|\underline{\mathbf{y}}_{1:\tau}, \hat{\underline{\mathbf{s}}}_\tau, \widehat{\underline{\boldsymbol{\Theta}}}_\tau)$. In fact, this implementation is an approximation of the exact Kalman filter defined in the original Kalman paper [17]. According to [16], diagonalization is vital since the exact Kalman filter suffers from extremely high memory usage along with potential numerical instability. Figure shows the predictor/corrector structure of the diagonalized Kalman filter for the parameters $\widehat{\underline{\mathbf{w}}}_\tau$ and $\underline{\mathbf{P}}_\tau$:

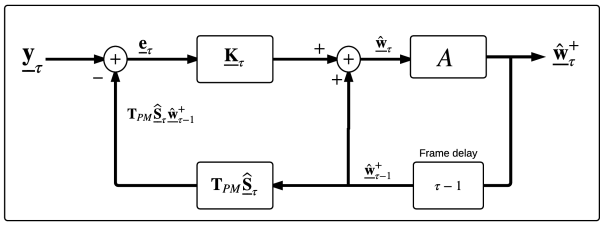

Figure 3.1: Prediction and correction of stacked channel vector

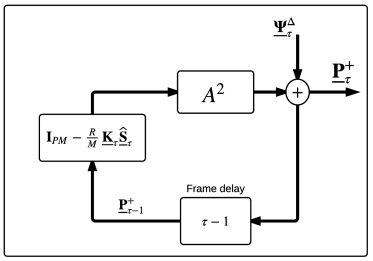

Figure 3.2: Prediction and correction of stacked error covariance matrix

In this section, the model parameter set $\underline{\boldsymbol{\Theta}}_\tau = \{\underline{\boldsymbol{\Psi}}^n_{i,\tau}, \underline{\boldsymbol{\Psi}}^\Delta_{i,\tau}\}$ was assumed to be known from the Maximization step. In the next section the estimation of these noise covariances will be discussed.

3.1.2 M-Step: Covariance estimation

In the M-Step, the parameters $\widehat{\underline{\mathbf{w}}}_\tau$ and $\underline{\mathbf{P}}_\tau$ are assumed to be known beforehand from the previous Expectation step. In addition, the clean signal $\hat{\underline{\mathbf{s}}}_\tau$ is also assumed to be known from the Equalizer step which is discussed in the next section. Once all the parameters are known, the frequency domain observation noise vector estimate $\widehat{\underline{\mathbf{n}}}_\tau$ can be calculated using (2.13) such as:

$$\widehat{\underline{\mathbf{n}}}_\tau = \underline{\mathbf{y}}_\tau - \mathbf{T}_{PM}\widehat{\underline{\mathbf{S}}}_\tau\widehat{\underline{\mathbf{w}}}_{\tau-1} \tag{3.13}$$

If the reverberant signal is in line with the model described in section 2 and the model parameters are estimated correctly then the additive noise can be estimated easily by subtracting the noiseless convolution (second term) from the reverberant signal frame. Therefore the observation noise covariance matrix could be estimated by the following equation:

$$\widehat{\underline{\boldsymbol{\Psi}}}^n_\tau = \left(\hat{\underline{\mathbf{n}}}_\tau\hat{\underline{\mathbf{n}}}^H_\tau + \widehat{\underline{\boldsymbol{\mathcal{S}}}}_\tau\underline{\mathbf{P}}_\tau\widehat{\underline{\boldsymbol{\mathcal{S}}}}^H_\tau\right) \circ \mathbf{I}_{PM} \tag{3.14}$$

Similarly, using the channel vector estimates and the error covariance matrix estimates from the previous E-step, an estimate of the process noise covariance matrix can be calculated by:

$$\underline{\widehat{\mathbf{\Psi}}}_{\tau}^{\Delta} = \left((1-A)^2 \underline{\hat{\mathbf{w}}}_{\tau} \underline{\hat{\mathbf{w}}}_{\tau}^{H} + (1-A^2)\underline{\mathbf{P}}_{\tau} + \gamma \mathbf{I}_{PM}\right) \circ \mathbf{I}_{PM} \tag{3.15}$$

where γ is a regularizer constant used to prevent the matrix elements from reaching zero. In other words, it prevents numerical instability that might arise if $\underline{\mathbf{P}}_{\tau-1}^{+}$ becomes zero. The terms $(1-A^2)$ and $(1-A)^2$ appear after imposing the first order Markov model to the process noise estimate. The full proof for equations 4.17 and 3.15 can be found at [18].
This step combined with the previous Kalman filtering step (E-Step) compose the sequential EM algorithm which is used to learn parameters using ML techniques such that they are passed to the equalizer thus removing reverberation and additive noise effects.

3.2 LMS equalizer: source signal estimation

In this section, the LMS equalizer is explained. Once the channel vector $\underline{\widehat{\mathbf{w}}}_{\tau}$ and the error covariance matrix $\underline{\mathbf{P}}_{\tau}$ are estimated from the sequential EM algorithm, the reverberant signal frame can be successfully equalized. The LMS equalizer takes the form shown below:

$$\underline{\hat{\mathbf{s}}}_{\tau} = \left(\sum_{i=1}^{P} (\underline{\widehat{\mathbf{W}}}_{i,\tau}^{H} \underline{\widehat{\mathbf{W}}}_{i,\tau}^{H} + \underline{\mathbf{P}}_{i,\tau})\right)^{-1} \underline{\widehat{\mathbf{W}}}_{\tau}^{H} \underline{\mathbf{y}}_{\tau} \tag{3.16}$$

The equalizer takes the form of a matched filter array defined in [19]. It is comprised to two fundamental stages. Firstly, the stacked impulse response matrix $\underline{\widehat{\mathbf{W}}}_{\tau}$ which is defined in a manner similar to that of (2.11) is conjugated. Since conjugation in the frequency domain is equivalent to time reversal in the time domain [20], each reversed impulse response acts as a matched filter. Thus $\underline{\widehat{\mathbf{W}}}_{\tau}$ is a cascade of matched filters which are used to equalize the stacked reverberant signals $\underline{\mathbf{y}}_{\tau}$. Secondly, matrix $\underline{\mathbf{D}}_{\tau} = \left(\sum_{i=1}^{P} (\underline{\widehat{\mathbf{W}}}_{i,\tau}^{H} \underline{\widehat{\mathbf{W}}}_{i,\tau}^{H} + \underline{\mathbf{P}}_{i,\tau})\right)^{-1}$ corresponds to a single-channel post-filter stage. Thus it weights each frequency band in effort of equalizing the magnitude. The weighting is controlled by the error covariance matrix $\underline{\mathbf{P}}_{\tau}$. The post-filter suppresses frequency bands which are deemed to have high channel estimation error as signified by the error covariance matrix and vice versa. Finally, the estimated signals for all channels are averaged to produce an clean signal frame to be utilized in the estimation of next frame parameters. Figure 3.3 demonstrates this two stage procedure:

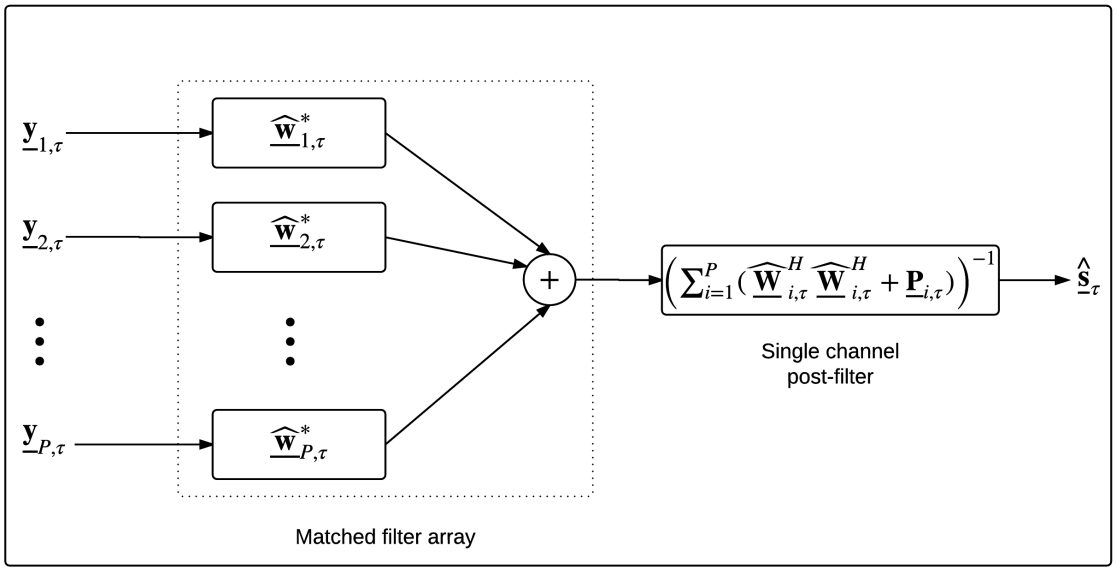

Figure 3.3: Block diagram for the MLBENCH LMS equalizer

More details regarding the derivation of the LMS equalizer can be found in [21]. Finally, this stage coupled with the sequential EM algorithm defined in the last section comprise the MLBENCH

algorithm. Figure 3.4 illustrates the interaction between the coupled systems explained to dereverb the reverberant signal.

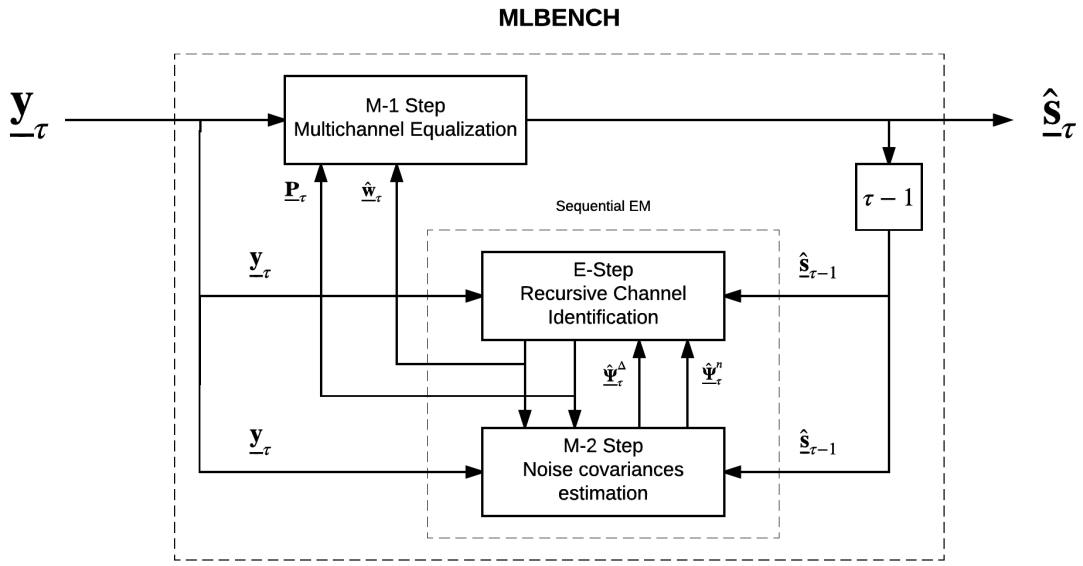

Figure 3.4: MLBENCH schematic outlining individual steps and interactions

In the next section, the implementation details of the algorithm are discussed.

3.3 Algorithm Implementation

In this section, directions for the implementation of this algorithm are given. The summary of the algorithm is shown in algorithm 1. Firstly, the algorithm is initialized with $\underline{\widehat{\mathbf{w}}}_\tau = \underline{\mathbf{1}}$ and $\underline{\mathbf{P}}_\tau = \mathbf{0}_{PM}$ as per rules of [22]. It is critical that the reverberant signal frames $\underline{\mathbf{y}}_\tau$ are correctly synchronized with the estimated parameters $\underline{\widehat{\mathbf{w}}}_\tau$, $\underline{\mathbf{P}}_\tau$, $\{\underline{\mathbf{\Psi}}^n_{i,\tau}$ and $\underline{\mathbf{\Psi}}^\Delta_{i,\tau}\}$. In order to facilitate such synchronization, the algorithm starts with the LMS equalizer step. In this step, it is necessary that the weighting matrix $\underline{\mathbf{D}}_\tau$ is normalized to the mean. In other words, we subtract the mean from the weighting values. This is done to guarantee that the algorithm does not converge to a trivial solution. Since the equalization step is a DFT convolution between the reverberant signals frame and the matched filter array, it is necessary to apply an overlap save constraint $\mathbf{T}_{PM}$ to the estimated output to eliminate circular convolution effects. Finally, the equalizer introduces a delay of $L-1$ samples which should be accounted for in the time domain when implementing the algorithm. Next, the estimated signal is fed into the Maximization step of the EM algorithm where the noise parameters are estimated. The products $\underline{\hat{\mathbf{n}}}_\tau \underline{\hat{\mathbf{n}}}^H_\tau$ and $\underline{\hat{\mathbf{w}}}_\tau \underline{\hat{\mathbf{w}}}^H_\tau$ produce $PM \times PM$ dense matrices which consume lots of memory and eventually increases the computational complexity of the algorithm. However, since these matrices were defined in section 2 as uncorrelated and hence diagonal, it is better to perform an element-wise multiplication of the vectors involved then convert the result into a $PM \times PM$ sparse diagonal matrix. Furthermore, the resulting covariance matrices $\{\underline{\mathbf{\Psi}}^n_{i,\tau}$ and $\underline{\mathbf{\Psi}}^\Delta_{i,\tau}\}$ could be multiplied element-wise with a same size diagonal matrix to ensure stability. That being said, the estimated covariance matrices along with the estimated signal are then passed into the Kalman filter step (E-Step). In this step, it is advisable according to [12] to calculate the error in the time domain for increased efficiency. In order to do so, we need to define a stacked versions of the DFT and the IDFT matrices. These are defined as:

$$\mathbf{F}_{PM} = \mathbf{I}_P \otimes \mathbf{F}_\mathbf{M} \tag{3.17}$$

$$\mathbf{F}_{PM}^{-1} = \mathbf{I}_P \otimes \mathbf{F}_{\mathbf{M}}^{-\mathbf{1}} \tag{3.18}$$

Therefore the error calculation step in equation 3.10 is rewritten as:

$$\underline{\mathbf{e}}_\tau = \mathbf{F}_{PM}(\mathbf{F}_{PM}^{-1}\underline{\mathbf{y}}_\tau - \mathbf{F}_{PM}^{-1}\mathbf{T}_{PM}\widehat{\underline{\mathbf{S}}}_\tau\widehat{\underline{\mathbf{w}}}_{\tau-1}^{+}) \tag{3.19}$$

The introduction of these sparse DFT and IDFT matrices would not greatly increase the computational load since they are sparse matrices and could be efficiently implemented using Fast Fourier Transform algorithm (FFT). Finally, for the stacked channel update step in 5.11, the second term produces a periodic output due to DFT properties. In order to eliminate these aliases, another constraint $\mathbf{C}_{PM}$ is defined which goes in hand with equation 2.4. This stacked constraint saves the first L samples of the individual estimated channel then pads it with R zeros such that it conforms with the model definition. This is a minor correction introduced to the equations in [1, 23]. It is defined in a manner analogous of the stacked overlap save constraint $\mathbf{T}_{PM}$:

$$\mathbf{E} = [\mathbf{I}_L \ \mathbf{0}_{L\times R}]^T \tag{3.20}$$

$$\mathbf{C} = \mathbf{F}_{\mathbf{M}}\mathbf{E}\mathbf{E}^T\mathbf{F}_M^{-1} \tag{3.21}$$

$$\mathbf{C}_{PM} = \mathbf{I}_P \otimes \mathbf{C} \tag{3.22}$$

Thus the update equation in 5.11 can be written as:

$$\widehat{\underline{\mathbf{w}}}_\tau = \mathbf{C}_{PM}(\widehat{\underline{\mathbf{w}}}_{\tau-1}^{+} + \underline{\boldsymbol{\mu}}_\tau\widehat{\underline{\mathbf{S}}}_\tau^H\underline{\mathbf{e}}_\tau) \tag{3.23}$$

Most of the $PM \times PM$ matrices in this algorithm are diagonal. It is very advantageous to convert them into sparse form in order to save memory and increase computational speed. Finally, an important implementation tip is to first test each step alone including the real parameters then try to couple the steps together while gradually removing the real parameters. The initializations mentioned at the start of this section cannot be eliminated by running the algorithm sequentially. For the first couple of frames (≈ 100), the algorithm needs to run until convergence in the recursive posterior estimator and the covariance estimation steps. Thus the equalizer feds the estimate into a coupled EM algorithm. Once the channel is estimated along with the noise parameters another equalization happens and so on. Gradually the likelihood function parameters are learned and one iteration is enough for convergence especially if the channel is slowly varying under the Markov model assumptions.

Algorithm 1 MLBENCH algorithm

1: **for** $\tau = 1, 2, \cdots$ **do**

M1-Step: Source Signal Estimation

2: $\hat{\underline{\mathbf{s}}}_\tau = \mathbf{T}_M \left(\sum_{i=1}^{P} (\widehat{\underline{\mathbf{W}}}_{i,\tau}^H \widehat{\underline{\mathbf{W}}}_{i,\tau}^H + \underline{\mathbf{P}}_{i,\tau}) \right)^{-1} \widehat{\underline{\mathbf{W}}}_\tau^H \underline{\mathbf{y}}_\tau$

M2-Step: Noise Covariance Estimation

3: $\widehat{\underline{\mathbf{n}}}_\tau = \underline{\mathbf{y}}_\tau - \mathbf{T}_{PM} \widehat{\underline{\mathbf{S}}}_\tau \widehat{\underline{\mathbf{w}}}_{\tau-1}$

4: $\widehat{\underline{\mathbf{\Psi}}}_\tau^n = \left(\hat{\underline{\mathbf{n}}}_\tau \hat{\underline{\mathbf{n}}}_\tau^H + \widehat{\underline{\boldsymbol{\mathcal{S}}}}_\tau \underline{\mathbf{P}}_\tau \widehat{\underline{\boldsymbol{\mathcal{S}}}}_\tau^H \right) \circ \mathbf{I}_{PM}$

5: $\widehat{\underline{\mathbf{\Psi}}}_\tau^\Delta = \left((1-A)^2 \hat{\underline{\mathbf{w}}}_\tau \hat{\underline{\mathbf{w}}}_\tau^H + (1-A^2) \underline{\mathbf{P}}_\tau + \gamma \mathbf{I}_{PM} \right) \circ \mathbf{I}_{PM}$

E-Step: Recursive Channel Posterior Estimation

6: $\widehat{\underline{\mathbf{w}}}_{\tau-1}^+ = A \widehat{\underline{\mathbf{w}}}_{\tau-1}$

7: $\widehat{\underline{\mathbf{P}}}_{\tau-1}^+ = A^2 \, \underline{\mathbf{P}}_{\tau-1} + \underline{\mathbf{\Psi}}_\tau^\Delta$

8: $\underline{\boldsymbol{\mu}}_\tau = \underline{\mathbf{P}}_{\tau-1}^+ \left(\widehat{\underline{\mathbf{S}}}_\tau \underline{\mathbf{P}}_{\tau-1}^+ \widehat{\underline{\mathbf{S}}}_\tau^H + \frac{M}{R} \underline{\mathbf{\Psi}}_\tau^n \right)^{-1}$

9: $\underline{\mathbf{e}}_\tau = \mathbf{F}_{PM} (\mathbf{F}_{PM}^{-1} \underline{\mathbf{y}}_\tau - \mathbf{F}_{PM}^{-1} \mathbf{T}_{PM} \widehat{\underline{\mathbf{S}}}_\tau \widehat{\underline{\mathbf{w}}}_{\tau-1}^+)$

10: $\widehat{\underline{\mathbf{w}}}_\tau = \mathbf{C}_{PM} (\widehat{\underline{\mathbf{w}}}_{\tau-1}^+ + \underline{\boldsymbol{\mu}}_\tau \widehat{\underline{\mathbf{S}}}_\tau^H \underline{\mathbf{e}}_\tau)$

11: $\underline{\mathbf{P}}_\tau = \underline{\mathbf{P}}_{\tau-1}^+ - \frac{R}{M} \underline{\boldsymbol{\mu}}_\tau \widehat{\underline{\mathbf{S}}}_\tau^H \widehat{\underline{\mathbf{S}}}_\tau \underline{\mathbf{P}}_{\tau-1}^+$

12: **end for**

Chapter 4

Maximum A-Posteriori Blind Equalization aNd CHannel identification (MAPBENCH) algorithm

Using Maximum Likelihood techniques in the MLBENCH algorithm led us to model the source vector $\underline{\mathbf{s}}_\tau$ as an unknown but deterministic parameter. What if we have prior knowledge (belief) regarding the model parameters we are trying to estimate? We cannot incorporate this knowledge into the MLBENCH algorithm explained in the previous chapter. Adding belief to the estimate is the main characteristic of another class of estimators called the Maximum A-Posteriori (MAP) estimators. Before going into MAP version of the MLBENCH algorithm, it is necessary to identify the types of prior distributions that can be incorporated in the learning process. The introduction of priors to the estimate either maximizes the peak of the posterior distribution (Good prior), doesn't affect the ML estimate (non-informative prior) or wrongly biases the estimate (Wrong prior). Firstly, a non informative prior doesn't have information on the underlying parameters and thus does assign equal probability to all possible parameter values. In other words, it has a uniform structure and could be viewed as allowing the data to speak for itself without the designer's intervention [15]. In the context of speech dereverberation, the primary parameter of interest is the clean signal vector $\underline{\mathbf{s}}_\tau$. Therefore, adding a sensible and informative prior to the it could highly improve its estimate. This informative prior has the structure of:

$$p(\underline{\mathbf{s}}_\tau|\underline{\boldsymbol{\Theta}}_\tau) = \mathcal{CN}(\underline{\mathbf{s}}_\tau|\ \mathbf{0}_{M\times 1},\ \underline{\boldsymbol{\Phi}}^s_\tau) \tag{4.1}$$

where the prior of the parameter $\underline{\mathbf{s}}_\tau$ is a zero mean multivariate complex Gaussian distribution with $M \times M$ source covariance matrix $\underline{\boldsymbol{\Phi}}^s_\tau$. This prior is appropriate since it is well known that frequency domain short speech frames experience and inherent gaussian structure [24]. By adopting the prior above more emphasis will be placed on the estimation of the source signal. We can also understate other parameters $\underline{\boldsymbol{\Phi}}^s_\tau$ since they are of no specific interest by assigning them a non-informative distribution where

$$p(\underline{\boldsymbol{\Theta}}_\tau) = \text{constant} \tag{4.2}$$

Since the above equation states that no information is held regarding the estimation of the parameters $\underline{\boldsymbol{\Phi}}^s_\tau$, the MAP estimates of these parameters will be analogous to those of the MLBENCH algorithm. In concordance to the MAP parameter learning rules discussed above, the algorithm will search for the maxima of the posterior $p(\underline{\mathbf{s}}_\tau, \underline{\boldsymbol{\Theta}}_\tau \mid \mathbf{y}_{1:\tau})$ where $\mathbf{y}_{1:\tau}$ constitutes all the available observations.
The optimization method used in the MAPBENCH algorithm to maximize the posterior distribution is the same as that of the MLBENCH algorithm maximizing the log likelihood function defined in (3.3). The EM algorithm is used to maximize a lower bound for the log posterior distribution as shown below:

$$\underline{\widehat{\mathbf{w}}}^+_{\tau-1} = A\underline{\widehat{\mathbf{w}}}_{\tau-1} \tag{4.3}$$

$$\underline{\widehat{\mathbf{P}}}_{\tau-1}^{+} = A^2\, \underline{\mathbf{P}}_{\tau-1} + \underline{\mathbf{\Psi}}_{\tau}^{\Delta} \tag{4.4}$$

$$\begin{aligned} \ln p(\underline{\mathbf{s}}_\tau, \underline{\mathbf{\Theta}}_\tau \mid \mathbf{y}_{1:\tau}) &= \ln \int q_w(\underline{\mathbf{w}}_\tau) \frac{p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \underline{\mathbf{s}}_\tau, \underline{\mathbf{\Theta}}_\tau) p(\underline{\mathbf{s}}_\tau \mid \underline{\mathbf{\Theta}}_\tau) p(\underline{\mathbf{\Theta}}_\tau)}{p(\underline{\mathbf{y}}_\tau \mid \underline{\mathbf{y}}_{1:\tau-1}) q_w(\underline{\mathbf{w}}_\tau)} \mathrm{d}\underline{\mathbf{w}}_\tau \\ &\geq \int q_w(\underline{\mathbf{w}}_\tau) \ln \frac{p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \underline{\mathbf{s}}_\tau, \underline{\mathbf{\Theta}}_\tau) p(\underline{\mathbf{s}}_\tau \mid \underline{\mathbf{\Theta}}_\tau) p(\underline{\mathbf{\Theta}}_\tau)}{p(\underline{\mathbf{y}}_\tau \mid \underline{\mathbf{y}}_{1:\tau-1}) q_w(\underline{\mathbf{w}}_\tau)} \mathrm{d}\underline{\mathbf{w}}_\tau \end{aligned} \tag{4.5}$$

where the lower bound of the posterior is defined. In the E-step of the algorithm, the lower bound is maximized with respect to the inference of the channel posterior distribution $q_w(\underline{\mathbf{w}}_\tau)$. Next, the M-step maximizes the lower bound and therefore obtains MAP-estimates of the parameters $\underline{\mathbf{s}}_\tau$ and $\underline{\mathbf{\Theta}}_\tau$.

4.1 E-Step: frequency domain adaptive filter

In this section, the lower bound is maximized for the distribution $q_w(\underline{\mathbf{w}}_\tau)$. The MAP BENCH algorithm uses exactly the same recursive posterior estimator as the MLBENCH algorithm described earlier. This could be proven in [1]. Therefore, the channel posterior estimation equations are given as:

$$\underline{\widehat{\mathbf{w}}}_{\tau-1}^{+} = A\underline{\widehat{\mathbf{w}}}_{\tau-1} \tag{4.6}$$

$$\underline{\widehat{\mathbf{P}}}_{\tau-1}^{+} = A^2\, \underline{\mathbf{P}}_{\tau-1} + \underline{\mathbf{\Psi}}_{\tau}^{\Delta} \tag{4.7}$$

$$\underline{\boldsymbol{\mu}}_\tau = \underline{\mathbf{P}}_{\tau-1}^{+} \left(\underline{\widehat{\mathbf{S}}}_\tau \underline{\mathbf{P}}_{\tau-1}^{+} \underline{\widehat{\mathbf{S}}}_\tau^H + \frac{M}{R} \underline{\mathbf{\Psi}}_\tau^n \right)^{-1} \tag{4.8}$$

$$\underline{\mathbf{e}}_\tau = \underline{\mathbf{y}}_\tau - \mathbf{T}_{PM} \underline{\widehat{\mathbf{S}}}_\tau \underline{\widehat{\mathbf{w}}}_{\tau-1}^{+} \tag{4.9}$$

$$\underline{\widehat{\mathbf{w}}}_\tau = \underline{\widehat{\mathbf{w}}}_{\tau-1}^{+} + \underline{\boldsymbol{\mu}}_\tau \underline{\widehat{\mathbf{S}}}_\tau^H \underline{\mathbf{e}}_\tau \tag{4.10}$$

$$\underline{\mathbf{P}}_\tau = \underline{\mathbf{P}}_{\tau-1}^{+} - \frac{R}{M} \underline{\boldsymbol{\mu}}_\tau \underline{\widehat{\mathbf{S}}}_\tau^H \underline{\widehat{\mathbf{S}}}_\tau \underline{\mathbf{P}}_{\tau-1}^{+} \tag{4.11}$$

4.2 M-Step: Covariances Estimation

4.2.1 Source Covariance Estimation

In the MAPBENCH algorithm we are particularly interested in the following covariances:

$$\underline{\mathbf{\Theta}}_\tau = (\underline{\mathbf{\Phi}}_\tau^s, \underline{\mathbf{\Psi}}_{i,\tau}^n, \underline{\mathbf{\Psi}}_{i,\tau}^{\Delta}) \tag{4.12}$$

where $\underline{\mathbf{\Phi}}_\tau^s$ is the $M \times M$ source covariance matrix. The noise covariances comprise the observation noise covariance matrix $\underline{\mathbf{\Psi}}_{i,\tau}^n$ and the process noise covariance matrix $\underline{\mathbf{\Psi}}_{i,\tau}^{\Delta}$.

Firstly, the MAP estimate of the source covariance matrix could be obtained maximizing the lower bound function in (4.5) with respect to $\underline{\mathbf{\Phi}}_\tau^s$:

$$\frac{\delta}{\delta \underline{\mathbf{\Phi}}_\tau^s} \mathcal{L}(q_w^*(\underline{w}_{i,\tau}), \hat{\underline{\mathbf{s}}}_\tau, \underline{\mathbf{\Theta}}_\tau) = \mathbf{0}_M \tag{4.13}$$

where $q_w^*(\underline{w}_{i,\tau}$ is the optimal distribution given by

$$q_w^*(\underline{\mathbf{w}}_{i,\tau}) = p(\underline{\mathbf{w}}_\tau \mid \underline{\mathbf{y}}_\tau, \hat{\underline{\mathbf{s}}}_\tau, \underline{\mathbf{\Theta}}_\tau) = \mathcal{CN}(\underline{\mathbf{w}}_\tau \mid \underline{\hat{\mathbf{w}}}_\tau, \underline{\mathbf{P}}_\tau) \tag{4.14}$$

This results in the MAP-optimal estimate of the source covariance matrix being equal to:

$$\underline{\mathbf{\Phi}}_{\tau}^{s} = \hat{\underline{\mathbf{s}}}_{\tau}\hat{\underline{\mathbf{s}}}_{\tau}^{H} \circ \mathbf{I}_{M} \tag{4.15}$$

where the Schur product ensures the diagonality of the estimates.

4.2.2 Observation Noise Covariance Estimation

Similar to the last section, the lower bound function is maximized with respect to the observation noise covariance matrix $\underline{\mathbf{\Psi}}_{\tau}^{n}$ such as:

$$\frac{\delta}{\delta \underline{\mathbf{\Phi}}_{\tau}^{n}} \mathcal{L}(q_{w}^{*}(\underline{w}_{i,\tau}), \hat{\underline{\mathbf{s}}}_{\tau}, \underline{\mathbf{\Theta}}_{\tau}) = \mathbf{0}_{PM} \tag{4.16}$$

The solution of the above maximization leaves us with the solution below:

$$\widehat{\underline{\mathbf{\Psi}}}_{\tau}^{n} = \left(\hat{\underline{\mathbf{n}}}_{\tau}\hat{\underline{\mathbf{n}}}_{\tau}^{H} + \widehat{\underline{\boldsymbol{\mathcal{S}}}}_{\tau}\underline{\mathbf{P}}_{\tau}\widehat{\underline{\boldsymbol{\mathcal{S}}}}_{\tau}^{H} \right) \circ \mathbf{I}_{PM} \tag{4.17}$$

where

$$\widehat{\underline{\mathbf{n}}}_{\tau} = \underline{\mathbf{y}}_{\tau} - \mathbf{T}_{PM}\widehat{\underline{\mathbf{S}}}_{\tau}\widehat{\underline{\mathbf{w}}}_{\tau} \tag{4.18}$$

Thus proving that adding a non-informative prior to this parameter results in a ML-optimal estimate such as that of the last chapter.
It is beneficial at this point to define the single channel observation noise covariance matrix $\underline{\mathbf{\Phi}}_{\tau}^{n}$which will be used in the source signal estimation step. It is the average of all the channel observation noise covariances. Thus:

$$\underline{\mathbf{\Phi}}_{\tau}^{n} = \frac{1}{P}\sum_{i=1}^{P} \underline{\mathbf{\Psi}}_{i,\tau}^{n} \tag{4.19}$$

4.2.3 Source Signal Estimation

The source signal estimation stage is where the informative source prior affects this algorithm. It is assumed that the covariance matrices parameter set in (4.12) is known from the previous E-step and the M-step algorithm defined in sections 4.1 and 4.2 respectively. Therefore in order to estimate the source signal the lower bound function is maximized with respect to $\underline{\mathbf{s}}_{\tau}$ as shown below:

$$\frac{\delta}{\delta \underline{\mathbf{s}}_{\tau}^{*}} \mathcal{L}(q_{w}^{*}(\underline{w}_{i,\tau}), \hat{\underline{\mathbf{s}}}_{\tau}, \underline{\mathbf{\Theta}}_{\tau}) = \mathbf{0}_{M\times 1} \tag{4.20}$$

where the operator $\frac{\delta}{\delta \underline{\mathbf{s}}_{\tau}^{*}}$ is the complex conjugate differential operator. The solution to this maximization problem takes the form

$$\hat{\underline{\mathbf{s}}}_{\tau} = \left(\sum_{i=1}^{P} \widehat{\underline{\mathbf{\Psi}}}_{i,\tau}^{n^{-1}} (\widehat{\underline{\mathbf{W}}}_{i,\tau}\widehat{\underline{\mathbf{W}}}_{i,\tau}^{H} + \underline{\mathbf{P}}_{i,\tau} + \frac{M}{R}\widehat{\underline{\mathbf{\Phi}}}_{\tau}^{s^{-1}} \right)^{-1} \widehat{\underline{\mathbf{W}}}_{\tau}^{H} \widehat{\underline{\mathbf{\Psi}}}_{\tau}^{n^{-1}} \underline{\mathbf{y}}_{\tau} \tag{4.21}$$

where $\widehat{\underline{\mathbf{W}}}_{i,\tau}$ is the unconstrained single-channel matrix. In addition, $\widehat{\underline{\mathbf{W}}}_{\tau}$ is the stacked unconstrained channel matrix defined as shown below:

$$\underline{\mathbf{W}}_{\tau} = [\underline{\mathbf{W}}_{1,\tau}\ \underline{\mathbf{W}}_{2,\tau} \cdots\ \underline{\mathbf{W}}_{P,\tau}]^{T} \tag{4.22}$$

Since all the matrices in the above equation are strictly diagonal, we could multiply equation (4.21) by the source covariance matrix thus resulting in:

$$\hat{\underline{\mathbf{s}}}_{\tau} = \left(\sum_{i=1}^{P} \underline{\mathbf{\Lambda}}_{i,\tau}(\widehat{\underline{\mathbf{W}}}_{i,\tau}\widehat{\underline{\mathbf{W}}}_{i,\tau}^{H} + \underline{\mathbf{P}}_{i,\tau}) + \frac{M}{R}\mathbf{I}_{M} \right) \widehat{\underline{\mathbf{W}}}_{\tau}^{H} \underline{\mathbf{\Lambda}}_{\tau}\underline{\mathbf{y}}_{\tau} \tag{4.23}$$

where $\underline{\mathbf{\Lambda}}_{i,\tau}$ denotes the single channel frequency domain Source Signal to Noise ratio (SSNR) matrix. The multichannel SSNR matrix is described by $\underline{\mathbf{\Lambda}}_{\tau}$. This term in the MAP Equalization suppresses frequency bands with low SSNRs. This term is channel dependent since it is multiplied by the bank of estimated channels $\underline{\widehat{\mathbf{W}}}_{\tau}$. It is more convenient to define the SSNR term such that it is not dependent on the channel during equalization. This simplification could be incorporated only assuming equal noise power at all microphones. In other words, a homogeneous noise field:

$$\underline{\mathbf{\Phi}}_{\tau}^{n} = \underline{\widehat{\mathbf{\Psi}}}_{i,\tau}^{n} \tag{4.24}$$

As a result, dividing equation (4.23) by the channel-dependent SSNR matrix $\underline{\mathbf{\Lambda}}_{\tau}$ leads to the following interpretation of the MAP equalization:

$$\underline{\hat{\mathbf{s}}}_{\tau} = \left(\sum_{i=1}^{P} (\underline{\widehat{\mathbf{W}}}_{i,\tau} \underline{\widehat{\mathbf{W}}}_{i,\tau}^{H} + \underline{\mathbf{P}}_{i,\tau}) + \frac{M}{R} \underline{\mathbf{\Omega}}_{\tau}^{-1} \right)^{-1} \underline{\widehat{\mathbf{W}}}_{\tau}^{H} \underline{\mathbf{y}}_{\tau} \tag{4.25}$$

where $\underline{\mathbf{\Omega}}_{\tau}$ is the channel independent SSNR matrix of size $M \times M$ and is defined as:

$$\underline{\mathbf{\Omega}}_{\tau} = \underline{\widehat{\mathbf{\Phi}}}_{\tau}^{s} \underline{\widehat{\mathbf{\Phi}}}_{\tau}^{n^{-1}} \tag{4.26}$$

Finally, the MAP equalizer has the two structure shown below:

$$\mathbf{G}_{\tau} = \underline{\widehat{\mathbf{W}}}_{\tau} \mathbf{D}_{\tau}^{-1} \tag{4.27}$$

where $\mathbf{D}_{\tau}^{-1}$ is an $M \times M$ diagonal matrix defined as:

$$\mathbf{D}_{\tau} = \sum_{i=1}^{P} (\underline{\widehat{\mathbf{W}}}_{i,\tau} \underline{\widehat{\mathbf{W}}}_{i,\tau}^{H} + \underline{\mathbf{P}}_{i,\tau}) + \frac{M}{R} \underline{\mathbf{\Omega}}_{\tau}^{-1} \tag{4.28}$$

The two stage equalization process consists of two fundamental stages. Firstly, a bank of channels is conjugated in the frequency domain which is analogous to time reversal in the time domain. In other words, the conjugation operation creates a cascade of matched filters to filter the corresponding reverberant signals. This Matched Filter Array (MFA) structure [19] captured by figure **??** is the same as that of the MLBENCH algorithm. Secondly, the outputs of the matched filter array are weighted such that their magnitudes are equalized. This is where the prior introduced at the start of the chapter distinguishes the MAP-estimate from the previous chapter ML-estimate. Firstly, like the ML-estimate, the posterior error covariance $\underline{\mathbf{P}}_{i,\tau}$ equalizes frequency bins with high channel estimation error. Secondly, the channel-independent SSNR term $\underline{\mathbf{\Omega}}_{\tau}$ equalizes frequency bins with low SSNR values. This two stage equalization is what differentiates the MAPBENCH algorithm from the MLBENCH algorithm. It is worth noting that the channel-independent SSNR term $\underline{\mathbf{\Omega}}_{\tau}$ which converts ML-estimates to MAP-estimates is known as a "regularizer" in the machine learning terminology [15, 25].
In the next section, the implementation details of the MAPBENCH algorithm are discussed.

4.3 MAPBENCH algorithm implementation

In this section, the intricate details for the implementation for the MAPBENCH algorithm are discussed. Firstly the structure of the full algorithm is shown in figure 4.1 below:

While implementing the following algorithm summarized in Algorithm 2, it is advisable to follow the steps outlined. First of all, the equalizer estimates the signal using frequency domain convolution. As a result, even though the convolution is constrained it leads to unwanted cyclic convolution effects. These effects could be easily nulled via using the overlap save constraint defined in section 2. Secondly, also due to circular convolution effects, the estimated signal is delayed by $L-1$ samples in the time domain. In order to circumvent this delay problem, the reverberant signal frame should also be delayed by $L-1$ samples in the time domain. Doing so will guarantee perfect time synchronization which is critical for the functionality of this algorithm. Also in the equalization step, the regularizer SSNR term $\underline{\mathbf{\Omega}}_{\tau}$ should be initialized by small values to avoid the early division by zero which might lead to

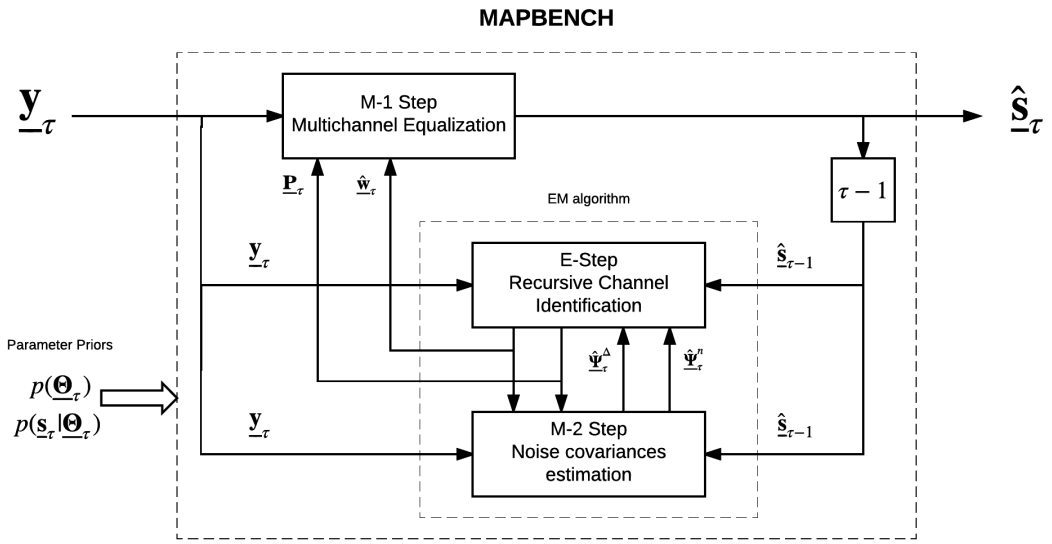

Figure 4.1: Block diagram for the MAPBENCH algorithm

numerical stability. In addition, the regularizer term $\underline{\mathbf{\Omega}}_\tau$ should be normalized before being added to the inverse term. This will prevent excessive weighting of the dereverbed signal such that the speech signal formants are heavily attenuated. Finally, the equalizer experiences fluctuations which produce artifacts in the estimate. This problem could be circumvented by smoothing the channel independent SSNR matrix $\underline{\mathbf{\Omega}}_\tau$ as shown below:

$$\underline{\mathbf{\Omega}}_\tau = \alpha \underline{\mathbf{\Omega}}_{\tau-1} + (1-\alpha)\underline{\mathbf{\Phi}}_\tau^s \widehat{\underline{\mathbf{\Phi}}}_\tau^{n^{-1}} \tag{4.29}$$

Furthermore, the total inverse term $\mathbf{D}_\tau$ should also be normalized to the mean to prevent the algorithm from converging to trivial solutions. For the covariance estimation step, it is highly advisable to estimate the covariances using tips discussed in the MLBENCH section for better computational performance and reduced memory usage.

Algorithm 2 MLBENCH algorithm

1: **for** $\tau = 1, 2, \cdots$ **do**

M1-Step: Source Signal Estimation

2: $\underline{\mathbf{\Omega}}_\tau = \alpha \underline{\mathbf{\Omega}}_{\tau-1} + (1-\alpha) \underline{\mathbf{\Phi}}_\tau^s \widehat{\underline{\mathbf{\Phi}}}_\tau^{n^{-1}}$

3: $\hat{\underline{\mathbf{s}}}_\tau = \mathbf{T}_M \left(\sum_{i=1}^{P} (\widehat{\underline{\mathbf{W}}}_{i,\tau}^H \widehat{\underline{\mathbf{W}}}_{i,\tau}^H + \underline{\mathbf{P}}_{i,\tau}) \right)^{-1} \widehat{\underline{\mathbf{W}}}_\tau^H \underline{\mathbf{y}}_\tau$

M2-Step: Noise Covariance Estimation

4: $\underline{\mathbf{\Phi}}_\tau^s = \hat{\underline{\mathbf{s}}}_\tau \hat{\underline{\mathbf{s}}}_\tau^H \circ \mathbf{I}_M$

5: $\widehat{\underline{\mathbf{n}}}_\tau = \underline{\mathbf{y}}_\tau - \mathbf{T}_{PM} \widehat{\underline{\mathbf{S}}}_\tau \widehat{\underline{\mathbf{w}}}_{\tau-1}$

6: $\widehat{\underline{\mathbf{\Psi}}}_\tau^n = \left(\hat{\underline{\mathbf{n}}}_\tau \hat{\underline{\mathbf{n}}}_\tau^H + \widehat{\underline{\boldsymbol{\mathcal{S}}}}_\tau \underline{\mathbf{P}}_\tau \widehat{\underline{\boldsymbol{\mathcal{S}}}}_\tau^H \right) \circ \mathbf{I}_{PM}$

7: $\underline{\mathbf{\Phi}}_\tau^n = \frac{1}{P} \sum_{i=1}^{P} \underline{\mathbf{\Psi}}_{i,\tau}^n$

8: $\widehat{\underline{\mathbf{\Psi}}}_\tau^\Delta = \left((1-A)^2 \hat{\underline{\mathbf{w}}}_\tau \hat{\underline{\mathbf{w}}}_\tau^H + (1-A^2) \underline{\mathbf{P}}_\tau + \gamma \mathbf{I}_{PM} \right) \circ \mathbf{I}_{PM}$

E-Step: Recursive Channel Posterior Estimation

9: $\widehat{\underline{\mathbf{w}}}_{\tau-1}^+ = A \widehat{\underline{\mathbf{w}}}_{\tau-1}$

10: $\widehat{\underline{\mathbf{P}}}_{\tau-1}^+ = A^2 \, \underline{\mathbf{P}}_{\tau-1} + \underline{\mathbf{\Psi}}_\tau^\Delta$

11: $\underline{\boldsymbol{\mu}}_\tau = \underline{\mathbf{P}}_{\tau-1}^+ \left(\widehat{\underline{\mathbf{S}}}_\tau \underline{\mathbf{P}}_{\tau-1}^+ \widehat{\underline{\mathbf{S}}}_\tau^H + \frac{M}{R} \underline{\mathbf{\Psi}}_\tau^n \right)^{-1}$

12: $\underline{\mathbf{e}}_\tau = \mathbf{F}_{PM} (\mathbf{F}_{PM}^{-1} \underline{\mathbf{y}}_\tau - \mathbf{F}_{PM}^{-1} \mathbf{T}_{PM} \widehat{\underline{\mathbf{S}}}_\tau \widehat{\underline{\mathbf{w}}}_{\tau-1}^+)$

13: $\widehat{\underline{\mathbf{w}}}_\tau = \mathbf{C}_{PM} (\widehat{\underline{\mathbf{w}}}_{\tau-1}^+ + \underline{\boldsymbol{\mu}}_\tau \widehat{\underline{\mathbf{S}}}_\tau^H \underline{\mathbf{e}}_\tau)$

14: $\underline{\mathbf{P}}_\tau = \underline{\mathbf{P}}_{\tau-1}^+ - \frac{R}{M} \underline{\boldsymbol{\mu}}_\tau \widehat{\underline{\mathbf{S}}}_\tau^H \widehat{\underline{\mathbf{S}}}_\tau \underline{\mathbf{P}}_{\tau-1}^+$

15: **end for**

Chapter 5

Variational Bayesian Blind Equalization aNd CHannel identification (VBBENCH) algorithm

As we have seen, the MLBENCH algorithm blindly estimated the source signal using multichannel observations along with the imposed first order Markov Model. However, the MAPBENCH algorithm used the same resources along with an additional source prior. In the MLBENCH algorithm, maximization of the log likelihood function was the basis of estimating the unknown parameters. On the other hand, the log posterior distribution was maximized with respect to unknown parameters. In the MLBENCH, the most sought parameter, $\underline{\mathbf{s}}_\tau$ was modelled as an unknown parameter. However, in the MAPBENCH algorithm, the source vector $\underline{\mathbf{s}}_\tau$ was modelled as a random variable with a specific prior distribution. This coincides with ML and MAP learning rules defined in [15]. Another extension of the above algorithms is estimating the source signal via modelling it as a latent random variable in a manner similar to that of modelling the channel vector. This leads to the variational Bayesian (VB) learning methods. The advantage of these methods is that they do not rely on point estimates such as ML and MAP techniques do. However, they allow the inference of the who posterior distribution of the random variable. Even though, VB methods provide a more accurate insight to the problem at hand, they still suffer from the disadvantage of being computationally expensive [25]. Since the source vector $\underline{\mathbf{s}}_\tau$ is now modelled as a random variable, the new likelihood function $p(\underline{\mathbf{y}}_\tau \mid \underline{\mathbf{y}}_{1:\tau-1}, \underline{\boldsymbol{\Theta}}_\tau)$ is conditioned on the model parameter set $\underline{\boldsymbol{\Theta}}_\tau = (\underline{\boldsymbol{\Phi}}^s_\tau, \underline{\boldsymbol{\Psi}}^n_{i,\tau}, \underline{\boldsymbol{\Psi}}^{\Delta}_{i,\tau})$. Now we have two latent variables $\underline{\mathbf{s}}_\tau$ and $\underline{\mathbf{w}}_\tau$ which are inserted into the likelihood function via double marginalization shown below:

$$p(\underline{\mathbf{y}}_\tau \mid \underline{\mathbf{y}}_{1:\tau-1}, \underline{\boldsymbol{\Theta}}_\tau) = \int\int p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau \mid \underline{\mathbf{y}}_{1:\tau-1}, \underline{\boldsymbol{\Theta}}_\tau) \mathrm{d}\underline{\mathbf{w}}_\tau \mathrm{d}\underline{\mathbf{s}}_\tau \tag{5.1}$$

The above joint density maximization can be solved iteratively via the usage of the EM algorithm. Similar to setting the lower bound in the last two chapters, the lower bound $\mathcal{L}(q(\underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau), \underline{\boldsymbol{\Theta}}_\tau)$ for the log likelihood function can be found using Jensen's inequality such as:

$$\begin{aligned} \ln p(\underline{\mathbf{y}}_\tau \mid \mathbf{y}_{1:\tau-1}, \underline{\boldsymbol{\Theta}}_\tau) &= \ln \int\int q(\underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau) \frac{p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \underline{\boldsymbol{\Theta}}_\tau)}{q_w(\underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau)} \mathrm{d}\underline{\mathbf{w}}_\tau \mathrm{d}\underline{\mathbf{s}}_\tau \\ &\geq \int\int q(\underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau) \ln \frac{p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \underline{\boldsymbol{\Theta}}_\tau)}{q_w(\underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau)} \mathrm{d}\underline{\mathbf{w}}_\tau \mathrm{d}\underline{\mathbf{s}}_\tau \end{aligned} \tag{5.2}$$

where $q(\underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau)$ is an unknown joint distribution over the modelled random variables $\underline{\mathbf{w}}_\tau$ and $\underline{\mathbf{s}}_\tau$). The optimal distribution is the one which tightens the lower bound and thus when substitutes above will turn the inequality into an equality. This optimal distribution is estimated in the Expectation step of the VBBENCH algorithm. The maximization of this distribution function with respect to the model parameter set $\underline{\boldsymbol{\Theta}}_\tau$ above is the M-step of the algorithm. Like most of the Bayesian solutions which are analytically intractable [15], this solution is. In fact, a solution to this problem could be derived

under certain approximations. In other words, if we assume that the joint distribution $q(\underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau)$ can be factorized as

$$q(\underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau) \approx q_w(\underline{\mathbf{w}}_\tau) q_s(\underline{\mathbf{s}}_\tau) \tag{5.3}$$

A closed solution can still be derived. The substitution of the joint density function approximation to equation (5.2) yields the following updated lower bound:

$$\mathcal{L}(q(\underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau), \underline{\boldsymbol{\Theta}}_\tau) = \int\int q_w(\underline{\mathbf{w}}_\tau) q_s(\underline{\mathbf{s}}_\tau) \ln \frac{p(\underline{\mathbf{y}}_\tau, \underline{\mathbf{w}}_\tau, \underline{\mathbf{s}}_\tau | \underline{\mathbf{y}}_{1:\tau-1}, \underline{\boldsymbol{\Theta}}_\tau)}{q_w(\underline{\mathbf{w}}_\tau) q_s(\underline{\mathbf{s}}_\tau)} \mathrm{d}\underline{\mathbf{w}}_\tau \mathrm{d}\underline{\mathbf{s}}_\tau \tag{5.4}$$

The modified lower bound shown above depends now on the individual distributions and on the model parameter set $\underline{\boldsymbol{\Theta}}_\tau$. The lower bound will be maximized separately for each individual distribution in effort to estimate the optimal distributions. Thus the algorithm will comprise of two expectation steps corresponding to each distribution.

5.1 Recursive Channel Posterior Estimation (E-Step(1))

In this step, the lower bound is maximized with respect to the arbitrary channel vector distribution $q_w(\underline{\mathbf{w}}_\tau)$ and the optimal source posterior distribution $q *_s (\underline{\mathbf{s}}_\tau)$ along with model parameter set $\underline{\boldsymbol{\Theta}}_\tau$ are assumed to be known from the other expectation step along with the M-step. The full proof of the solution to this E-step is not addressed here. Interested readers can check [1,23]. The solution to maximizing the lower bound with respect to the channel vector distribution is takes the form of a recursive channel posterior estimator Kalman filter which is then diagonalized for optimum performance. The prediction step is the same as in the other two algorithms such as:

$$\underline{\widehat{\mathbf{w}}}^+_{\tau-1} = A \underline{\widehat{\mathbf{w}}}_{\tau-1} \tag{5.5}$$

$$\underline{\widehat{\mathbf{P}}}^+_{\tau-1} = A^2 \, \underline{\mathbf{P}}_{\tau-1} + \underline{\boldsymbol{\Psi}}^{\Delta}_\tau \tag{5.6}$$

Intermediate steps are then added such that

$$\underline{\tilde{\mathbf{P}}}^+_{\tau-1} = (\underline{\mathbf{P}}^{+-1}_{\tau-1} + \frac{R}{M} \underline{\widehat{\boldsymbol{\Psi}}}^{n^{-1}}_\tau \underline{\mathbf{Q}}_\tau)^{-1} \tag{5.7}$$

$$\underline{U}_\tau = \underline{\tilde{\mathbf{P}}}^+_{\tau-1} \underline{\mathbf{P}}^{+^{-1}}_{\tau-1} \tag{5.8}$$

where $\underline{U}_\tau$ is a $PM \times PM$ weighing matrix. The following equations encompass the diagonalized correction step:

$$\underline{\boldsymbol{\mu}}_\tau = \underline{\tilde{\mathbf{P}}}^+_{\tau-1} \left(\underline{\widehat{\mathbf{S}}}_\tau \underline{\mathbf{P}}^+_{\tau-1} \underline{\widehat{\mathbf{S}}}^H_\tau + \frac{M}{R} \underline{\boldsymbol{\Psi}}^n_\tau \right)^{-1} \tag{5.9}$$

$$\underline{\mathbf{e}}_\tau = \underline{\mathbf{y}}_\tau - \mathbf{T}_{PM} \underline{\widehat{\mathbf{S}}}_\tau \underline{U}_\tau \underline{\widehat{\mathbf{w}}}^+_{\tau-1} \tag{5.10}$$

$$\underline{\widehat{\mathbf{w}}}_\tau = \underline{U}_\tau \underline{\widehat{\mathbf{w}}}^+_{\tau-1} + \underline{\boldsymbol{\mu}}_\tau \underline{\widehat{\mathbf{S}}}^H_\tau \underline{\mathbf{e}}_\tau \tag{5.11}$$

$$\underline{\mathbf{P}}_\tau = \underline{\tilde{\mathbf{P}}}^+_{\tau-1} - \frac{R}{M} \underline{\boldsymbol{\mu}}_\tau \underline{\widehat{\mathbf{S}}}^H_\tau \underline{\widehat{\mathbf{S}}}_\tau \underline{\tilde{\mathbf{P}}}^+_{\tau-1} \tag{5.12}$$

The main difference between this channel recursive estimator and that of the MLBENCH and the MAPBENCH algorithms is that there is modification of the channel error covariance matrix prediction $\underline{\widehat{\mathbf{P}}}^+_{\tau-1}$ as defined in equation (5.7).

5.2 E Step 2: Instantaneous Source Posterior Estimation

In the previous step the lower bound was maximized with respect to the arbitrary channel vector distribution $q_w(\underline{\mathbf{w}}_\tau)$. In this section, the source signal will be estimated via the maximization of the lower bound function with respect to the source vector distribution $q_s(\underline{\mathbf{s}}_\tau)$ assuming the presence of the optimal channel vector $q *_w (\underline{\mathbf{w}}_\tau)$. Again, the full derivation [1, 23] will be not be considered here. In order to understand the concept behind the VB-BENCH equalizer in this section, consider the solution to the likelihood maximization below.

$$\hat{\underline{\mathbf{s}}}_\tau = \left(\sum_{i=1}^{P} (\widehat{\underline{\boldsymbol{\mathcal{W}}}}_\tau^H \widehat{\underline{\boldsymbol{\Psi}}}_\tau^n \widehat{\underline{\boldsymbol{\mathcal{W}}}}_\tau + \mathbf{T}_M^H \widehat{\underline{\boldsymbol{\Psi}}}_\tau^n \mathbf{T}_M \circ \underline{\mathbf{P}}_{i,\tau}) + \widehat{\underline{\boldsymbol{\Phi}}}_\tau^{s^{-1}} \right)^{-1} \widehat{\underline{\boldsymbol{\mathcal{W}}}}_\tau^H \widehat{\underline{\boldsymbol{\Psi}}}_\tau^{n^{-1}} \underline{\mathbf{y}}_\tau \tag{5.13}$$

and

$$\underline{\mathbf{Q}}_{\diamond,\tau} = \left(\sum_{i=1}^{P} (\widehat{\underline{\boldsymbol{\mathcal{W}}}}_\tau^H \widehat{\underline{\boldsymbol{\Psi}}}_\tau^n \widehat{\underline{\boldsymbol{\mathcal{W}}}}_\tau + \mathbf{T}_M^H \widehat{\underline{\boldsymbol{\Psi}}}_\tau^n \mathbf{T}_M \circ \underline{\mathbf{P}}_{i,\tau}) + \widehat{\underline{\boldsymbol{\Phi}}}_\tau^{s^{-1}} \right)^{-1} \tag{5.14}$$

The term $\underline{\mathbf{Q}}_{\diamond,\tau}$ is a $M \times M$ error covariance matrix for the computation of the source signal $\hat{\underline{\mathbf{s}}}_\tau$. Its definition is analogous to that of the channel error covariance matrix $\underline{\mathbf{P}}_\tau$. The above equations are computationally expensive and would lead to impractical memory and processor loads. In order to circumvent this problem, approximations could be done such that:

$$\mathbf{T}_M \approx \frac{R}{M} \mathbf{I}_M \tag{5.15}$$

$$\mathbf{T}_M^H \widehat{\underline{\boldsymbol{\Psi}}}_\tau^{n^{-1}} \mathbf{T}_M^H \approx \frac{R}{M} \widehat{\underline{\boldsymbol{\Psi}}}_\tau^{n^{-1}} \tag{5.16}$$

These approximations above will render equations 5.13 and 5.14 computationally efficient and diagonal. They could be rewritten as:

$$\hat{\underline{\mathbf{s}}}_\tau = \left(\sum_{i=1}^{P} \widehat{\underline{\boldsymbol{\Psi}}}_{i,\tau}^{n^{-1}} (\widehat{\underline{\mathbf{W}}}_{i,\tau} \widehat{\underline{\mathbf{W}}}_{i,\tau}^H + \underline{\mathbf{P}}_{i,\tau} + \frac{M}{R} \widehat{\underline{\boldsymbol{\Phi}}}_\tau^{s^{-1}} \right)^{-1} \widehat{\underline{\mathbf{W}}}_\tau^H \widehat{\underline{\boldsymbol{\Psi}}}_\tau^{n^{-1}} \underline{\mathbf{y}}_\tau \tag{5.17}$$

$$\underline{\mathbf{Q}}_{\diamond,\tau} = \frac{M}{R} \left(\sum_{i=1}^{P} \widehat{\underline{\boldsymbol{\Psi}}}_\tau^{n^{-1}} (\widehat{\underline{\mathbf{W}}}_{i,\tau} \widehat{\underline{\mathbf{W}}}_{i,\tau}^H + \underline{\mathbf{P}}_{i,\tau}) + \frac{M}{R} \widehat{\underline{\boldsymbol{\Phi}}}_\tau^{s^{-1}} \right)^{-1} \tag{5.18}$$

The above equations are very similar to the MAPBENCH equalization equations. Using same techniques, the VBBENCH equalizer could be written as:

$$\underline{\mathbf{G}}_\tau = \underline{\mathbf{W}}_\tau \left(\sum_{i=1}^{P} (\widehat{\underline{\mathbf{W}}}_{i,\tau} \widehat{\underline{\mathbf{W}}}_{i,\tau}^H + \underline{\mathbf{P}}_{i,\tau}) + \frac{M}{R} \underline{\boldsymbol{\Omega}}_\tau^{-1} \right)^{-1} \tag{5.19}$$

where $\underline{\boldsymbol{\Omega}}$ is the channel independent SSNR matrix. The observation noise covariance matrix $\widehat{\underline{\boldsymbol{\Psi}}}_\tau^n$ in equation 4 is replaced by the single channel observation noise covariance matrix defined in 4.19 under the assumption of a homogeneous noise field. The source posterior error covariance matrix therefore has the form:

$$\underline{\mathbf{Q}}_{\diamond,\tau} = \frac{M}{R} \left(\sum_{i=1}^{P} \widehat{\underline{\boldsymbol{\Phi}}}_\tau^{n^{-1}} (\widehat{\underline{\mathbf{W}}}_{i,\tau} \widehat{\underline{\mathbf{W}}}_{i,\tau}^H + \underline{\mathbf{P}}_{i,\tau}) + \frac{M}{R} \widehat{\underline{\boldsymbol{\Phi}}}_\tau^{s^{-1}} \right)^{-1} \tag{5.20}$$

The above two equations summarise the source estimation process in the VBBENCH algorithm.

5.3 M-Step Covariance Parameter estimation

In the previous sections, the channel was estimated using a recursive posterior estimator and the source signal was estimated using an instantaneous equalizer. As mentioned before, the steps maximized the joint likelihood function by estimating the optimal posterior distributions that will render the Jensen's inequality in 5.2 into an equality. Once the optimal posterior distributions were computed in the two Expectation steps, they are fed into the maximization step to estimate the rest of the unknown parameters in the model parameter set $\underline{\mathbf{\Theta}}_\tau = \{\underline{\mathbf{\Phi}}^s_\tau, \underline{\mathbf{\Psi}}^n_\tau, \underline{\mathbf{\Psi}}^\Delta_\tau\}$. As usual this is done by maximizing the joint likelihood with respect to the parameter in desire. The full derivation of the learning rules of the model parameter set could be found in [1, 18]. The parameters are going to be ML-optimal since a non informative posterior was assigned to them. In this section, we are going to list the final update equations for each of the model parameters. Firstly, the source covariance matrix $\underline{\mathbf{\Phi}}^s_\tau$ is given by:

$$\underline{\mathbf{\Phi}}^s_\tau = (\hat{\underline{\mathbf{s}}}_\tau \hat{\underline{\mathbf{s}}}^H_\tau + \underline{\mathbf{Q}}_{\diamond,\tau}) \circ \mathbf{I}_M \tag{5.21}$$

Again here, the estimate is ensured to be diagonal upon the multiplication with $\mathbf{I}_M$. In addition, the source covariance estimation error is added here as a regularizer to convert the ML-optimal estimate to a VB optimal one. Next, the observation noise covariance matrix $\underline{\mathbf{\Psi}}^n_\tau$ estimate update is given by:

$$\widehat{\underline{\mathbf{\Psi}}}^n_\tau = \left(\hat{\underline{\mathbf{n}}}_\tau \hat{\underline{\mathbf{n}}}^H_\tau + \frac{R}{M}(\widehat{\underline{\boldsymbol{\mathcal{S}}}}_\tau \underline{\mathbf{P}}_\tau \widehat{\underline{\boldsymbol{\mathcal{S}}}}^H_\tau + \underline{\mathbf{P}}_\tau \underline{\mathbf{Q}}_\tau + \underline{\mathbf{w}}_\tau \underline{\mathbf{w}}^H_\tau \underline{\mathbf{Q}}_\tau)\right) \circ \mathbf{I}_{PM} \tag{5.22}$$

The VB estimate adds two regularization parameters to the ML and MAP estimates described in the last two chapters. Namely, $\underline{\mathbf{P}}_\tau \underline{\mathbf{Q}}_\tau$ and $\underline{\mathbf{w}}_\tau \underline{\mathbf{w}}^H_\tau \underline{\mathbf{Q}}_\tau$. The stacked noise covariance matrix $\widehat{\underline{\mathbf{\Psi}}}^n_\tau$ is averaged over the independent channels thus producing the single channel noise covariance matrix $\underline{\mathbf{\Phi}}^n_\tau$ in a way similar to equation 4.19. The channel independent SSNR matrix $\underline{\mathbf{\Omega}}_\tau$ has the same form as equation (4.29). Finally, the process noise covariance matrix is similar to the ML and MAP estimation equations. Thus the process noise covariance matrix is estimated in the VBBENCH algorithm as:

$$\widehat{\underline{\mathbf{\Psi}}}^\Delta_\tau = \left((1-A)^2 \hat{\underline{\mathbf{w}}}_\tau \hat{\underline{\mathbf{w}}}^H_\tau + (1-A^2)\underline{\mathbf{P}}_\tau + \gamma \mathbf{I}_{PM}\right) \circ \mathbf{I}_{PM} \tag{5.23}$$

5.4 VBBENCH algorithm implementation

The VBBENCH algorithm is summarized in the schematic in figure 5.1. The individual steps are shown along with their interactions with other parts of the algorithm. All of the implementation details discussed in the last two chapters still hold for this algorithm. In brief, the SSNR matrix should be normalized before the being added as a regularizer to the source estimation stage. In addition, noise covariance estimation could be done more efficiently by multiplying diagonals of individual vectors $\widehat{\underline{\mathbf{n}}}_\tau$ and $\widehat{\underline{\mathbf{w}}}_\tau$. Finally, the inverse term in the second M-step should be normalized by subtracting the mean from it.

It should be noted that the same way, the MAPBENCH algorithm acts like the MLBENCH algorithm when the SSNR matrix inverse is nulled, the VBBENCH algorithm is the most general case of BENCH family algorithms. If the source error covariance $\underline{\mathbf{Q}}_\tau$ is nulled, the algorithm acts like the MAPBENCH algorithm. In addition, if the SSNR inverse is also nulled, the VBBENCH algorithm acts like the MLBENCH algorithm. The algorithm steps are shown in algorithm 3.

Algorithm 3 VBBENCH algorithm

1: **for** $\tau = 1, 2, \cdots$ **do**

M1-Step: Source Signal Estimation

2: $\underline{\mathbf{\Omega}}_\tau = \alpha \underline{\mathbf{\Omega}}_{\tau-1} + (1-\alpha)\underline{\mathbf{\Phi}}_\tau^s \widehat{\underline{\mathbf{\Phi}}}_\tau^{n^{-1}}$

3: $\hat{\underline{\mathbf{s}}}_\tau = \left(\sum_{i=1}^P (\widehat{\underline{\mathbf{W}}}_{i,\tau}^H \widehat{\underline{\mathbf{W}}}_{i,\tau}^H + \underline{\mathbf{P}}_{i,\tau})\right)^{-1} \widehat{\underline{\mathbf{W}}}_\tau^H \underline{\mathbf{y}}_\tau$

4: $\underline{\mathbf{Q}}_{\diamond,\tau} = \frac{R}{M}\left(\sum_{i=1}^P \widehat{\underline{\mathbf{\Psi}}}_\tau^{n^{-1}} (\widehat{\underline{\mathbf{W}}}_{i,\tau} \widehat{\underline{\mathbf{W}}}_{i,\tau}^H + \underline{\mathbf{P}}_{i,\tau}) + \frac{M}{R} \widehat{\underline{\mathbf{\Phi}}}_\tau^{s^{-1}}\right)^{-1}$

M2-Step: Noise Covariance Estimation

5: $\underline{\mathbf{\Phi}}_\tau^s = (\hat{\underline{\mathbf{s}}}_\tau \hat{\underline{\mathbf{s}}}_\tau^H + \underline{\mathbf{Q}}_{\diamond,\tau}) \circ \mathbf{I}_M$

6: $\widehat{\underline{\mathbf{n}}}_\tau = \underline{\mathbf{y}}_\tau - \mathbf{T}_{PM} \widehat{\underline{\mathbf{S}}}_\tau \widehat{\underline{\mathbf{w}}}_{\tau-1}$

7: $\widehat{\underline{\mathbf{\Psi}}}_\tau^n = \left(\hat{\underline{\mathbf{n}}}_\tau \hat{\underline{\mathbf{n}}}_\tau^H + \frac{R}{M}(\widehat{\underline{\boldsymbol{\mathcal{S}}}}_\tau \underline{\mathbf{P}}_\tau \widehat{\underline{\boldsymbol{\mathcal{S}}}}_\tau^H + \underline{\mathbf{P}}_\tau \underline{\mathbf{Q}}_\tau + \underline{\mathbf{w}}_\tau \underline{\mathbf{w}}_\tau^H \underline{\mathbf{Q}}_\tau)\right) \circ \mathbf{I}_{PM}$

8: $\underline{\mathbf{\Phi}}_\tau^n = \frac{1}{P} \sum_{i=1}^P \underline{\mathbf{\Psi}}_{i,\tau}^n$

9: $\widehat{\underline{\mathbf{\Psi}}}_\tau^\Delta = \left((1-A)^2 \hat{\underline{\mathbf{w}}}_\tau \hat{\underline{\mathbf{w}}}_\tau^H + (1-A^2)\underline{\mathbf{P}}_\tau + \gamma \mathbf{I}_{PM}\right) \circ \mathbf{I}_{PM}$

E-Step: Recursive Channel Posterior Estimation

10: $\widehat{\underline{\mathbf{w}}}_{\tau-1}^+ = A \widehat{\underline{\mathbf{w}}}_{\tau-1}$

11: $\widehat{\underline{\mathbf{P}}}_{\tau-1}^+ = A^2 \, \underline{\mathbf{P}}_{\tau-1} + \underline{\mathbf{\Psi}}_\tau^\Delta$

12: $\tilde{\underline{\mathbf{P}}}_{\tau-1}^+ = (\underline{\mathbf{P}}_{\tau-1}^{+^{-1}} + \frac{R}{M} \widehat{\underline{\mathbf{\Psi}}}_\tau^{n^{-1}} \underline{\mathbf{Q}}_\tau)^{-1}$

13: $\underline{U}_\tau = \tilde{\underline{\mathbf{P}}}_{\tau-1}^+ \underline{\mathbf{P}}_{\tau-1}^{+^{-1}}$

14: $\underline{\boldsymbol{\mu}}_\tau = \underline{\mathbf{P}}_{\tau-1}^+ \left(\widehat{\underline{\mathbf{S}}}_\tau \underline{\mathbf{P}}_{\tau-1}^+ \widehat{\underline{\mathbf{S}}}_\tau^H + \frac{M}{R} \underline{\mathbf{\Psi}}_\tau^n\right)^{-1}$

15: $\underline{\mathbf{e}}_\tau = \mathbf{F}_{PM}(\mathbf{F}_{PM}^{-1} \underline{\mathbf{y}}_\tau - \mathbf{F}_{PM}^{-1} \mathbf{T}_{PM} \widehat{\underline{\mathbf{S}}}_\tau \underline{U}_\tau \widehat{\underline{\mathbf{w}}}_{\tau-1}^+)$

16: $\widehat{\underline{\mathbf{w}}}_\tau = \mathbf{C}_{PM}(\underline{U}_\tau \widehat{\underline{\mathbf{w}}}_{\tau-1}^+ + \underline{\boldsymbol{\mu}}_\tau \widehat{\underline{\mathbf{S}}}_\tau^H \underline{\mathbf{e}}_\tau)$

17: $\underline{\mathbf{P}}_\tau = \underline{\mathbf{P}}_{\tau-1}^+ - \frac{R}{M} \underline{\boldsymbol{\mu}}_\tau \widehat{\underline{\mathbf{S}}}_\tau^H \widehat{\underline{\mathbf{S}}}_\tau \underline{\mathbf{P}}_{\tau-1}^+$

18: **end for**

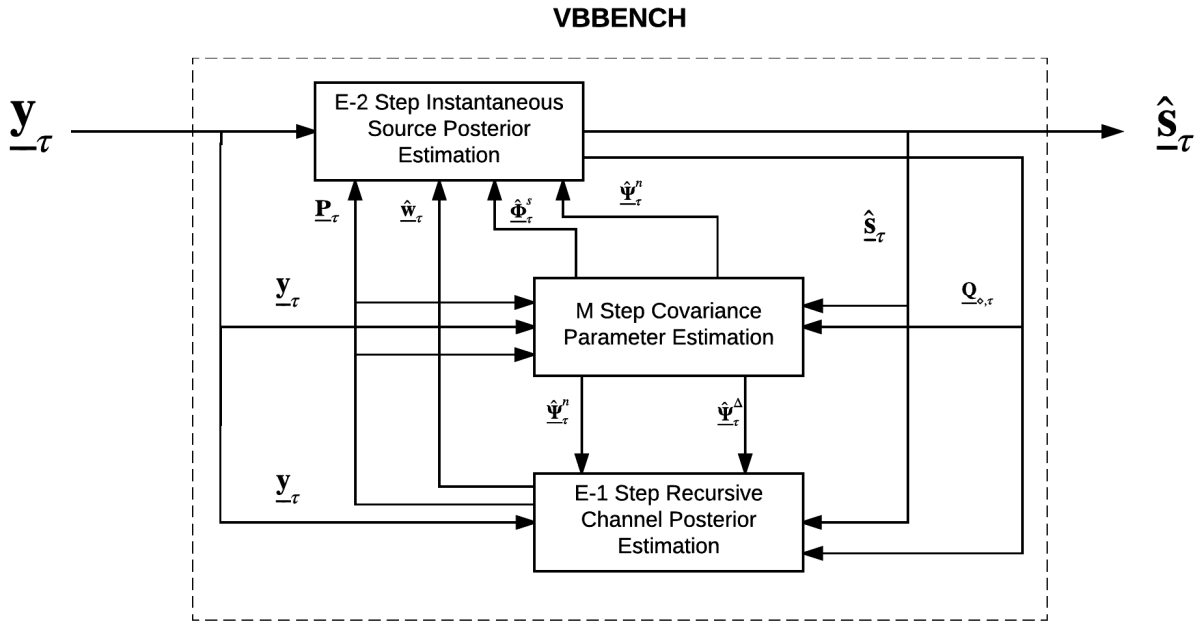

Figure 5.1: Block diagram for the VBBENCH algorithm

Chapter 6

Simulation and results

In this section, the simulation of the three algorithms discussed earlier will be implemented. In the first section, a simulated environment is created according to the model mentioned at the start of this document. The results for the simulations will be analyzed and examined using objective and subjective measures.

6.1 Objective speech quality measures

It is very important to use objective quality measures when assessing the performance of dereverberation and denoising algorithms. Some of the well-known objective measures for assessing speech performance algorithms are the Log-Likelihood-Ratio (LLR), Perceptual Evaluation of Speech Quality (PESQ) and the Speech to Reverberation Modulation energy Ratio tests.

6.1.1 Log likelihood ratio test

This test is one of the Linear Prediction (LP) based tests. It measures the distance between the LPC vector of the both the clean the noisy speech. It is calculated using the following equation:

$$d_{LLR}(\mathbf{a}_n, \mathbf{a}_c) = \log\left(\frac{\mathbf{a}_n \mathbf{R}_c \mathbf{a}_n^T}{\mathbf{a}_c \mathbf{R}_c \mathbf{a}_c^T}\right) \tag{6.1}$$

where $\mathbf{a}_c$ is the LPC vector of the clean speech, $\mathbf{a}_n^T$ is the LPC vector of the noisy speech and $\mathbf{R}_c$ is the autocorrelation matrix for the clean speech [26].

6.1.2 Segmental SNR

Since the classical definition of SNR is not well suited to speech quality, another measure which is reasonable is the segmental SNR. The classical SNR definition averages the ratio over the full signal. Since the speech signal is not stationary, its energy varies with time. As a result, some of the sections of the speech signal have large energy when it is audible and other sections are not. The segmental SNR is calculated in short frames before being averages. This can be expressed as:

$$SNR_{seg} = \frac{10}{M} \sum_{M-1}^{m=0} \log_{10} \frac{\sum_{n=Lm}^{Lm+L-1} x^2(n)}{\sum_{n=Lm}^{Lm+L-1} x^2(n) - \hat{x}^2(n)} \tag{6.2}$$

where L is the frame length, M represents the number of frames. As seen in the above equation, frames with large ratio have low weights applied to them in comparison with frames with low ratio. In other words, the existence of low quality noisy frame will decrease the overall quality measure of the signal since it is not being dominated by the large audible speech frames. This is one of the most popular tests used in IP telephony applications [27].

6.1.3 Perceptual Evaluation of Speech Quality

This is a standard quality test which estimates the Mean Opinion Score (MOS) between both the noisy and the clean signal [28]. This type of quality test time aligns the noisy signal with the original signal after converting them into a certain representation. Next, the delay associated with the corruption is calculated. The PESQ test has five assessing criteria. Namely, 1.0 for poor, 2.0 for bad, 3.0 for fair, 4.0 for very good and 5.0 for excellent.

6.1.4 Speech to Reverberation modulation energy ratio

This is the final objective performance test that we are going to consider in this document. This test is derived in the context of speech dereverberation and this makes it of utmost relatability to our performance analysis. This test is based on the modulation spectral representation of the noisy signal in a way that enables it to assess the intelligibility of the reverberant and enhanced signal. The SRMR measure software was provided by the authors of [29].

6.2 Implementation

To begin with, we started our implementation by creating data in concordance to the model described in section 2. Ten impulse responses of length L=128 were modelled using Alan and Berkeley impulse responses. These trivial impulse responses are shown in the figure below: After modelling the impulse responses, the first order Markov model described in equation (2.5) was imposed to the above impulse responses to model slow time variation. Using overlap save convolution, overlapped frames of the source signal were convolved with these impulse responses to produce outputs alligned with the definition of the model section. The BENCH algorithms were implemented and tested as per sections outlined below:

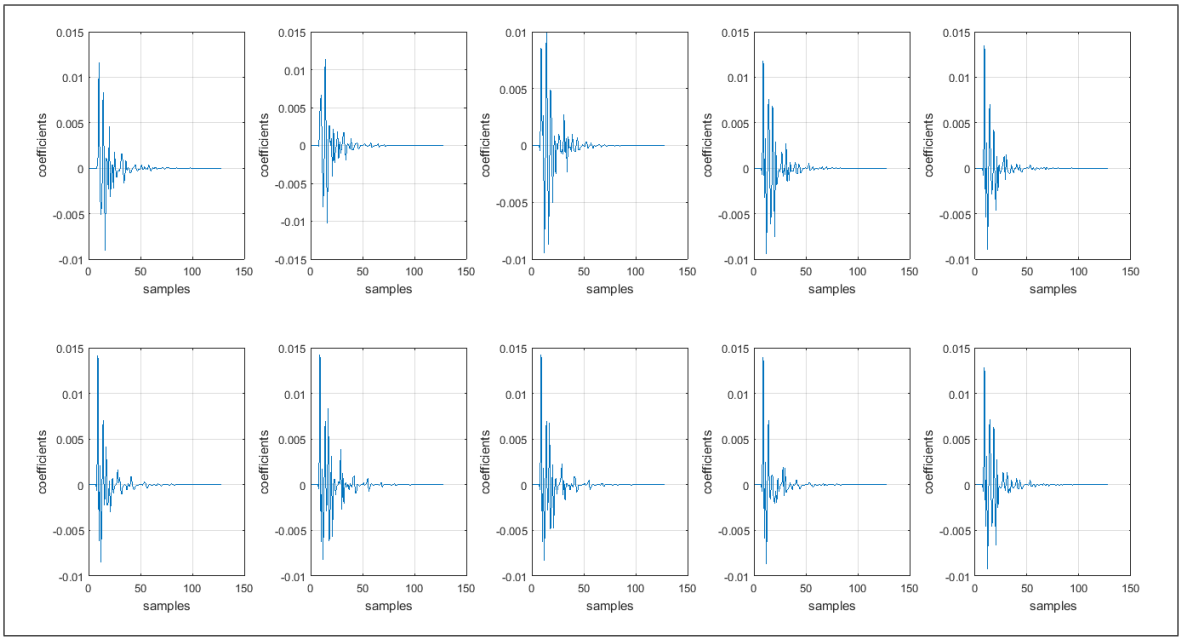

Figure 6.1: trivial impulse responses

6.2.1 Channel estimator and covariance estimator test

In this section, the EM algorithm block in the three algorithms was tested. The recursive channel posterior estimator along with the noise covariance matrices were tested by including the source signal in the EM algorithm along with the reverberant signal $\underline{\mathbf{y}}_\tau$ instead of the estimate $\hat{\underline{\mathbf{s}}}_\tau$ produced from the previous M-step (Equalization stage). The EM algorithm correctly estimated the channel along with small errors as shown in figure 6.2

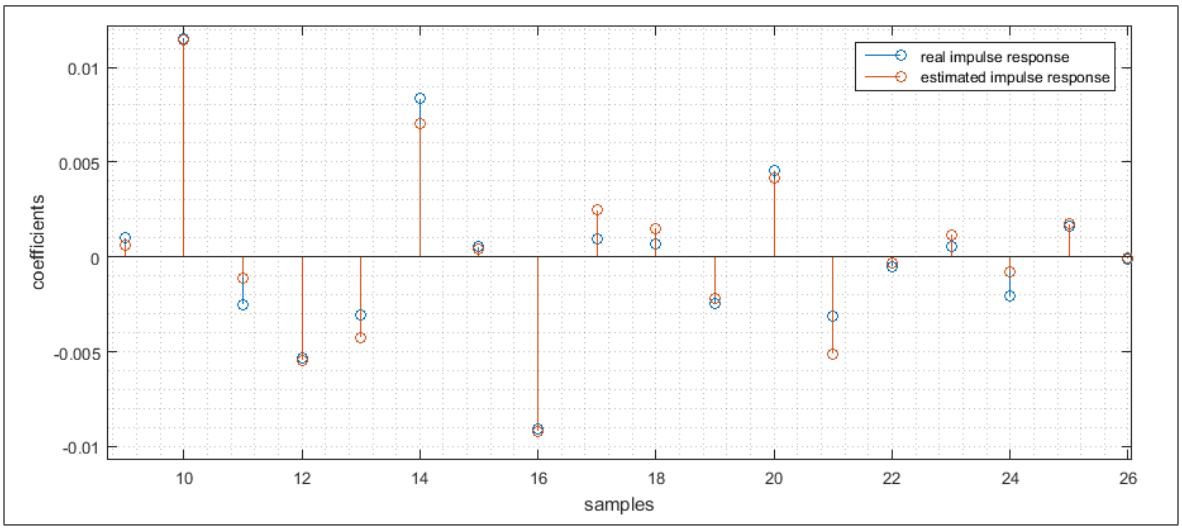

Figure 6.2: channel estimation with clean signal knowledge

As shown above, the channel coefficients were correctly estimated along with minor errors. This proves that the EM algorithm block of two coupled estimators is perfectly functioning and thus can be coupled with the ML or MAP estimator for a fully functioning algorithm.

6.2.2 ML, MAP and VB equalization tests

Inserting the real channel coefficients which were modelled can help us make sure that the algorithm is working. Therefore, in the three equalizers ML, MAP and VB, the real channel coefficients were inserted and the equalizer was tested standalone. In order to view the performance of the equalizer, spectrogram plots for estimated signals were used. These are shown in the figure below along with the reverberant plot:

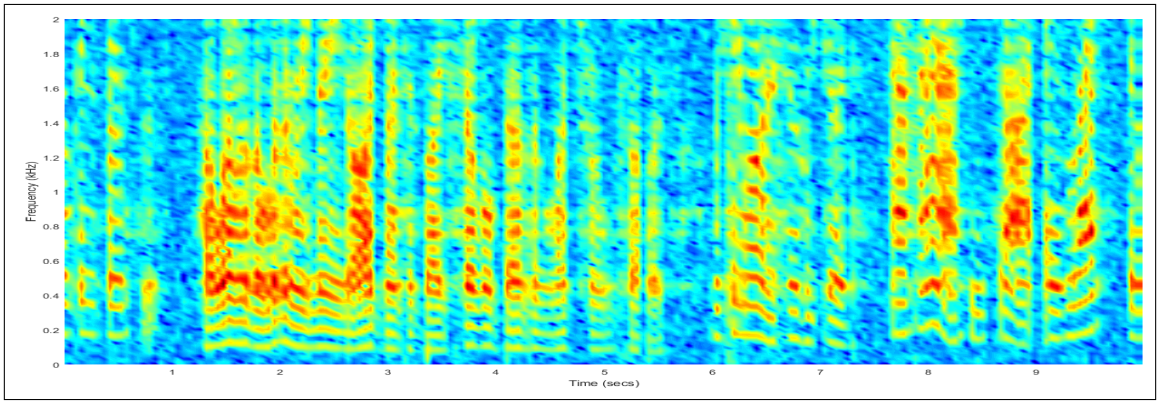

Figure 6.3: noisy and reverberant signal

It can be seen that inputting the real channel coefficients into the equalization step removes all effects from the reverberant and noisy signal. This test proves that the equalizer step is functional. In the MAPBENCH and the VBBENCH case, the prior in the VBBENCH and the source error covariance matrix were nulled thus providing us with the same MLBENCH equalizer whose functionality has just been proved. Since the individual stages of the algorithm have been proved as working, we proceed to coupling stages together.

6.2.3 Algorithm implementation

In this section, the deterministic parameters such as the clean source signal and the channel were removed to test whether the blind performance of the BENCH algorithms. Even though the individual

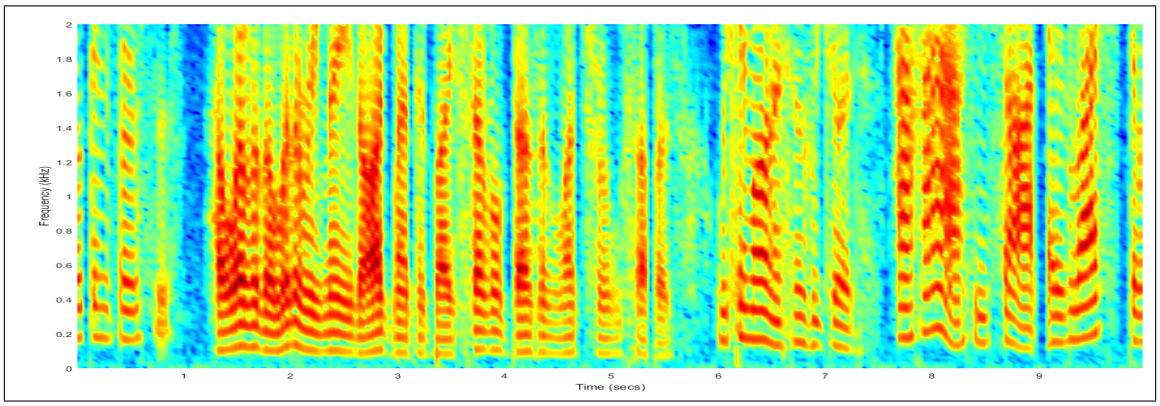

Figure 6.4: Fully equalized speech signal

stages proved working in the last two sections, there seems a synchronization error preventing the correct estimation of the reverberating channels. The estimated channels of the algorithm are shown below:

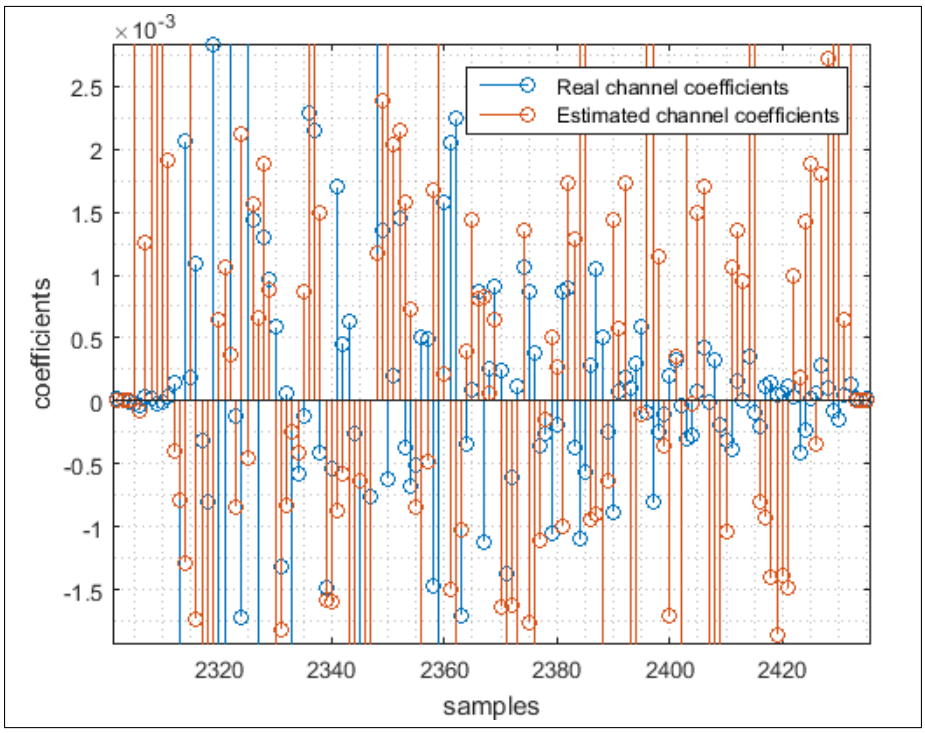

Figure 6.5: BENCH algorithms estimated channel

It can be seen that the channel estimation is erroneous. As a result, the reverberant signal will be reverbed more by the wrongly estimated channel thus deteriorating the performance. The Kalman filter-EM learning algorithm was left to convergence per frame. The convergence condition was the difference of two subsequent states does not exceed 0.001. The possible errors to this problem is that the estimated signal is not synchronized in the time domain with the reverberant signal. Therefore, in order to circumvent any time domain delays, a the cross correlation between the estimated signal frame and the cascade of reverberant frames was computed. No visible delay in the time domain was found as shown in figure below:
It is apparent that the both the estimated signal and the time domain reverberant signal are in full sync in the time domain. That is because each blob in the diagram represents a reverberant signal. Since the estimated signal is highest correlated at the zero lag of the first reverberant signal, therefore the signals are fully synchronized. Another reason why the algorithm might not be converging to th original is that the algorithm cannot eliminate initialization effects. In order to do so blindly, a

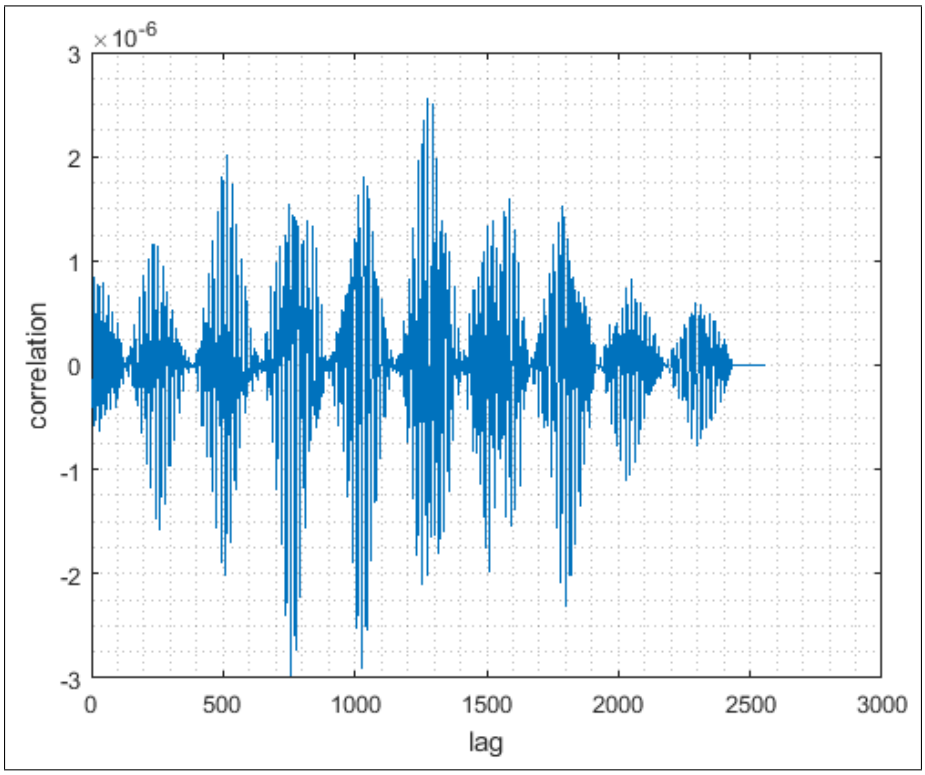

Figure 6.6: Cross correlation between observation frame and estimated source signal

Frequency domain MultiChannel Least Mean Squares (FMCLMS) was implemented with respect to laws given in [5]. The roughly estimated channel was then used to initialize the channel mean to the Kalman filter. However, there was an inherent problem with the channel and the channel error covariance estimation that rendered the channel estimates faulty.

6.3 Simulated Acoustic environment

The algorithms operated at 4 KHz sampling frequency and had a frame size M of 256 samples along with a frame shift R of 128 samples and modelled impulse response length of $M - R$. The values of these frames were chosen to ensure that the signal abides by the model since the algorithm assumes the reverberant signal to have a Gaussian model. In other words, 128/4000 yields 32 ms which falls in the range where the signal has quasi stationary statistic such that its parameters can be estimated. The clean speech frames used in this simulation were a compilation of five different speech samples concatenated with five different female speech samples gathered from the TIMIT corpus database [30]. The state-transition coefficient A was set to 0.9997 thus modelling a slowly time varying impulse response. This equates to being in a surrounding where no room geometry alterations happen. The modelled impulse responses were generated using Alan and Berkeley's impulse response model based on the image method [31]. The number of microphones used in the simulated was 10. In addition the room properties for which the above impulse responses were simulated are shown below:

Table 6.1: Hypothetical Room properties

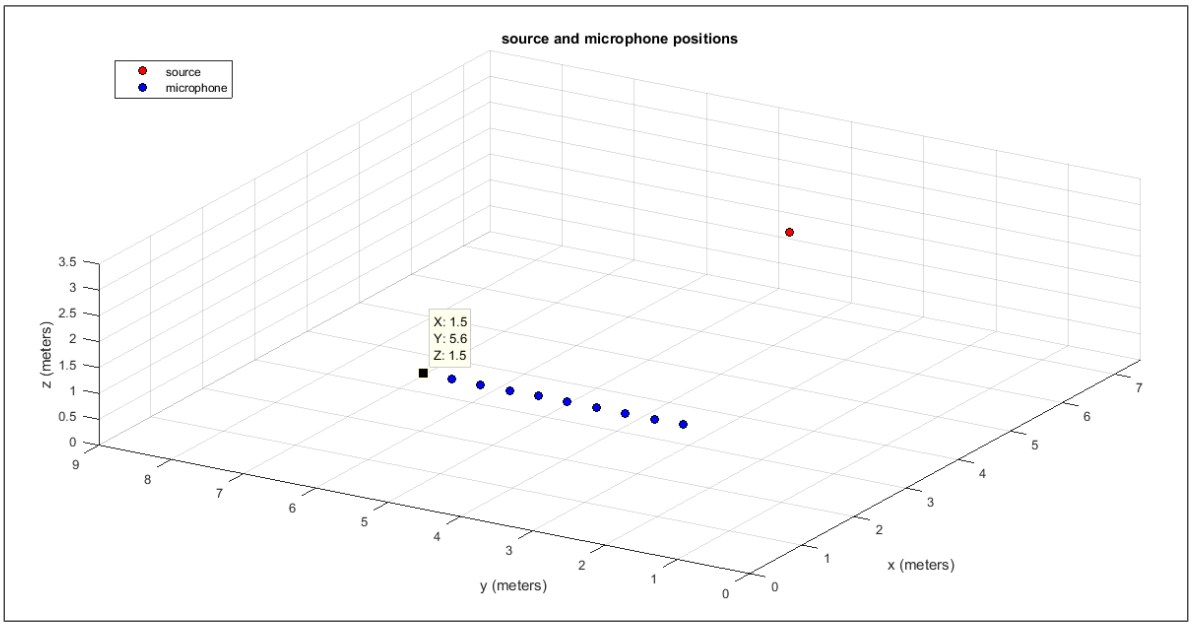

Figure 6.7: Room Plot

Room dimensions	[7.5, 9, 3.5]
Source position	[7, 4.5, 1.5]
Reference microphone	[2.4, 2, 1.5]
Microphones spacing	0.4 m
Reflection coefficients	[0.96, 0.96, 0.96, 0.96, 0.96, 0.96]
Reverberation times (T60)	0.2, 0.4, 0.6, 0.8
Sampling frequency	4000 KHz
Impulse response length	T60*Sampling frequency
No. of microphones	10

A rough schematic of the positions of the microphones in the room with the above properties is shown in figure 6.7. The room shows the relative distances between each microphone and the source. In addition, it also demonstrates that the array of microphones is uniformly spaced with the reference microphone highlighted.
The simulated impulse responses of length $L = 0.4 \times 4000 = 1600$ coefficients are shown in figure **??**. The impulse response set shown has a length which L such that $L \gg L_m$ adopted by the BENCH algorithms. However, this is not a problem since if the BENCH algorithms are successfully implemented, the modelled coefficients are basically the first coefficients of the impulse response which correspond directly to the direct path and early path reflections. As a result, the equalizer will align these components and average them to retrieve the equalized signal. The impulse responses models produced using Alan and Berkeley image method for the same room in table 6.1 are shown in figure 6.8.
It is apparent that the above impulse responses are well structured and therefore could be used in the modelling of the reverberant speech as described earlier in chapter 2. It is very important while simulating the performance of the algorithms that the impulse responses are well-structured and less noisy.

6.4 PESQ tests

The results below summarize the comparison between the various algorithms for different reverberation times and noise values. We are first going to examine the PESQ quality measure for the BENCH

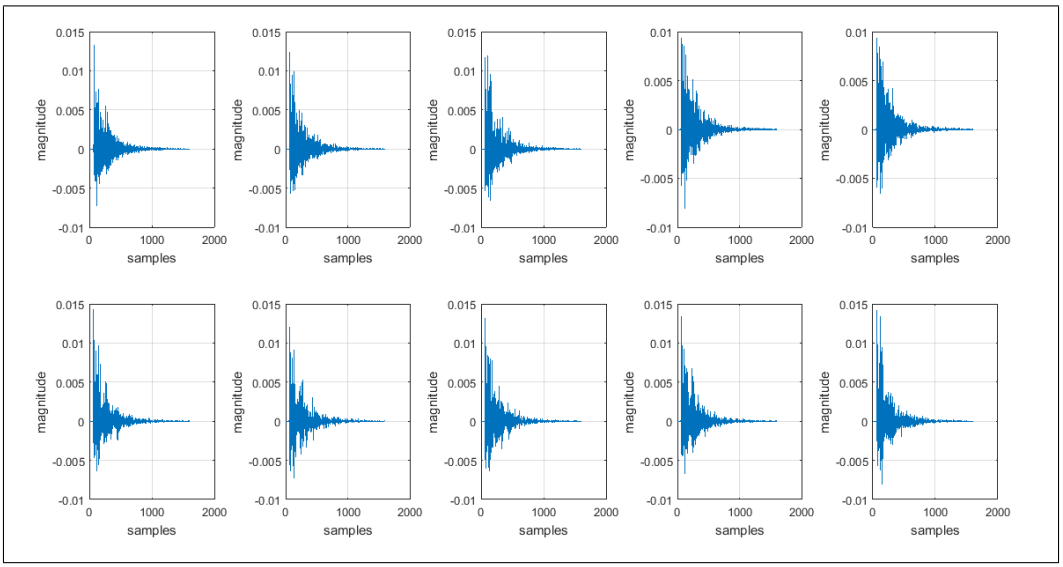

Figure 6.8: Impulse response used in simulation generated by Alan and Berkeley Image method.

algorithms compared with the reverberant signal at reverberation times ranging from 0.2 to 0.8 in steps of 0.2 and additive noises of 5 dB, 10 dB, 20 dB and 40 dB. Figures 6.9-6.11 show the PESQ test results for the BENCH family algorithms versus the PESQ measure for the reverberant signal. In figure **??**, with very high additive noise and less reverberation, the PESQ measure shows big deterioration in the quality of the reverberant signal. As shown, the signal enhanced using the BENCH algorithms has a higher PESQ measure when compared to very noisy, less reverberant signal. As the reverberation time increases, we administer more spectral distortion to our reverberant signal along with the very high additive noise. The worst testing scenario in our tests include a reverberation time of 0.8 and 5 dB additive noise. The reverberant signal falls to a PESQ measure of 1.66 which denote a signal of bad quality compared to the clean signal. It is evident from the high noise PESQ plot, that the algorithms experience a robust denoising performance. As we increase the SNR from 5 dB to 10 dB, the PESQ measure for the reverberant signal improves overall. Similarly are the estimates produced by the BENCH algorithms. Finally, in the 40 dB SNR in figure 6.11, the additive noise is minimum, it can be easily inferred from the figure that the BENCH algorithms produce worse results in less noisy conditions since they wrongly estimate the channel and thus for high reverberation, they further deteriorate the signal.

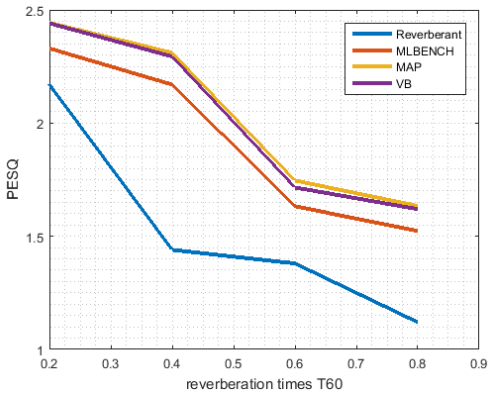

Figure 6.9: PESQ at 5 dB noise

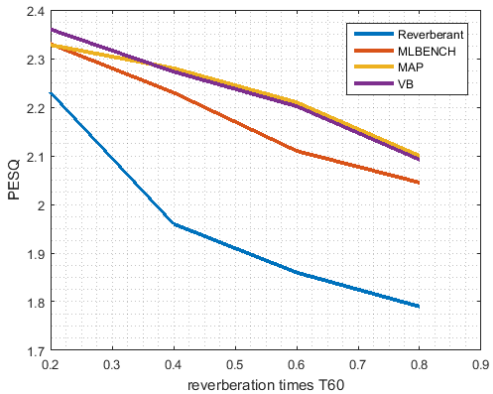

Figure 6.10: PESQ at 10 dB noise

In the next section, we are going to examine segmental SNR performance of the algorithms.

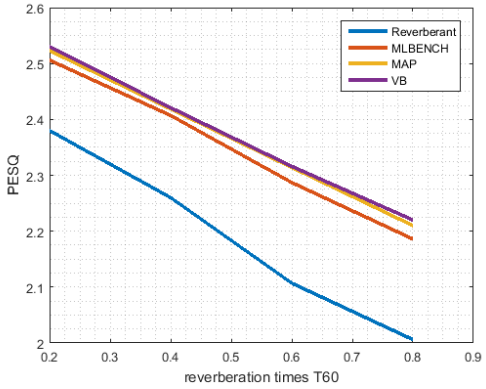

Figure 6.11: PESQ at 20 dB noise

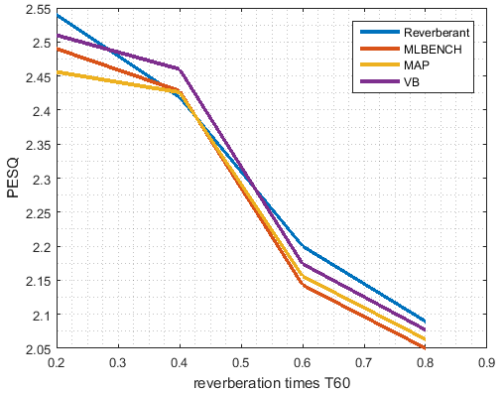

Figure 6.12: PESQ at 40 dB noise

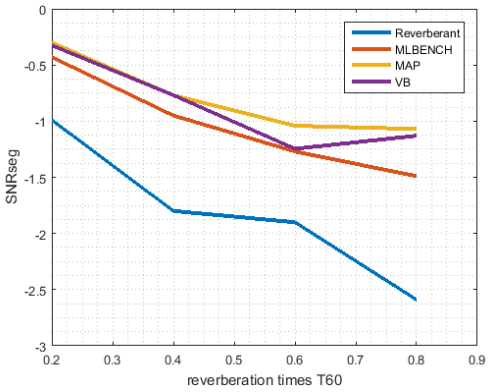

Figure 6.13: SNRseg at 5 dB

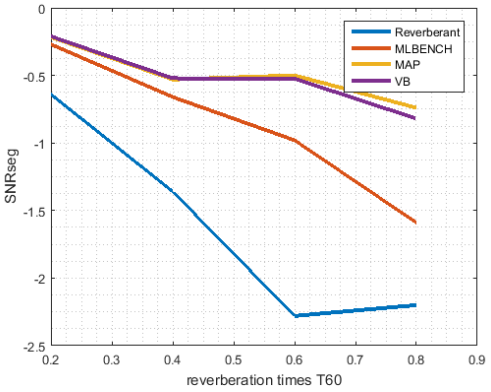

Figure 6.14: SNRseg at 10 dB noise

6.5 Segmental SNR tests

In this section, the segmental SNR test is used to check the performance improvement incurred by the BENCH algorithms in removing additive noise. Figures 6.13-6.16 outline the performance of the algorithms at SNRs 5 dB, 10 dB, 20 dB and 40 dB respectively. Zero SNR marks the clean signal reference. The more noisy is the signal, the more it goes away from the zero mark. Thus the figures demonstrate a sturdy denoising performance. However, it should be noted that the segmental SNR measures the performance with additive noise. Thus, it should not vary with reverberation times which is not the case here. Even though the results show an improvement for the enhanced signal, they are quite questionable. In addition, the SNRseg tests does not capture the extra deterioration in the signal quality as a result of wrong channel estimation as with the case of the PESQ test.

6.6 SRMR measures

In this section, the SRMR measures for the performance of the algorithms is examined. The same noise tests and reverberation times are included here. As it can be seen, the BENCH enhanced signals experience better performance. It can also be noted that the effect of the channel independent SSNR matrix regulizer effect can be clearly visualized when going from a very noisy simulation environment to a less noisy one.
The results of the VBBENCH enhancement in figure **??** must have been prone to a measurement error. In terms of the tests mentioned, it appears that the algorithms positively enhance the reverberant signal. Meanwhile objective measures in this case are enough to assess the performance of the algorithms, subjective tests such as listening to the enhanced signal and visualizing it in spectrograms are equally important. In the next section, the spectrogram plot will be examined.

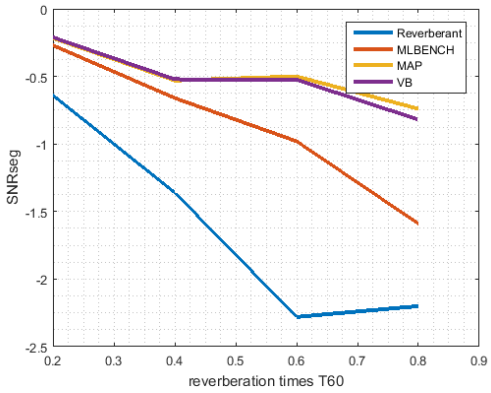

Figure 6.15: SNRseg at 20 dB noise

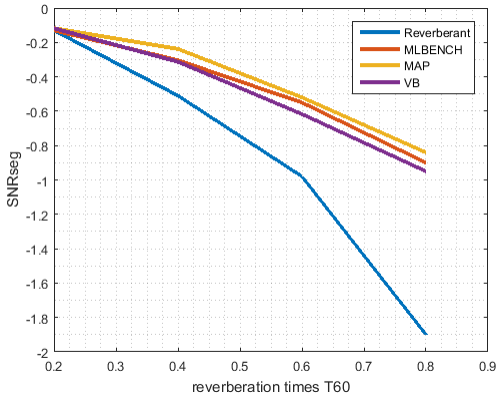

Figure 6.16: SNRseg at 40 dB noise

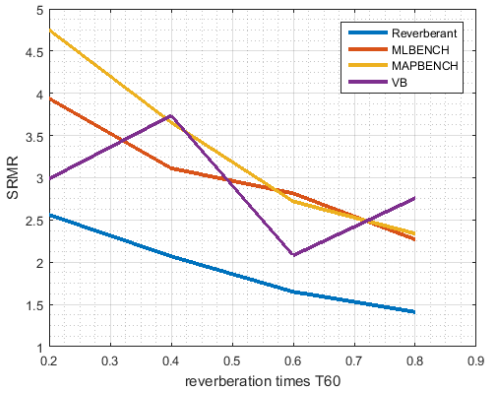

Figure 6.17: SRMR at 5 dB noise

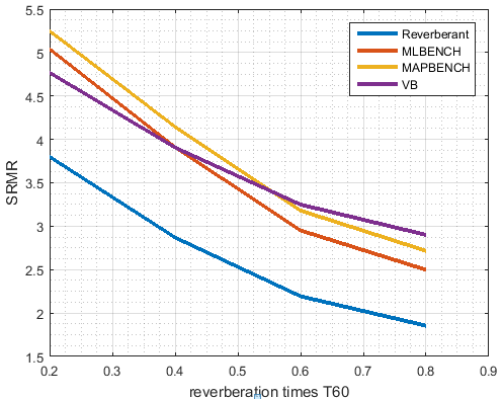

Figure 6.18: SRMR at 10 dB noise

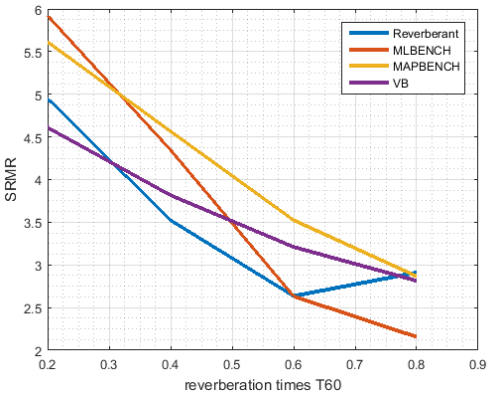

Figure 6.19: SRMR at 20 dB noise

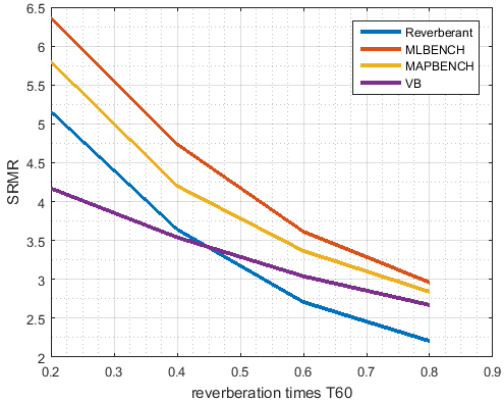

Figure 6.20: SRMR at 40 dB noise

6.7 Spectrogram subjective test

In this section, as a proof of concept, the spectrograms for the dereverberation performance of the algorithms will be shown. The spectrogram simulation was run at a reverberation time of 0.8 seconds and 10 dB additive noise. The results are shown below:

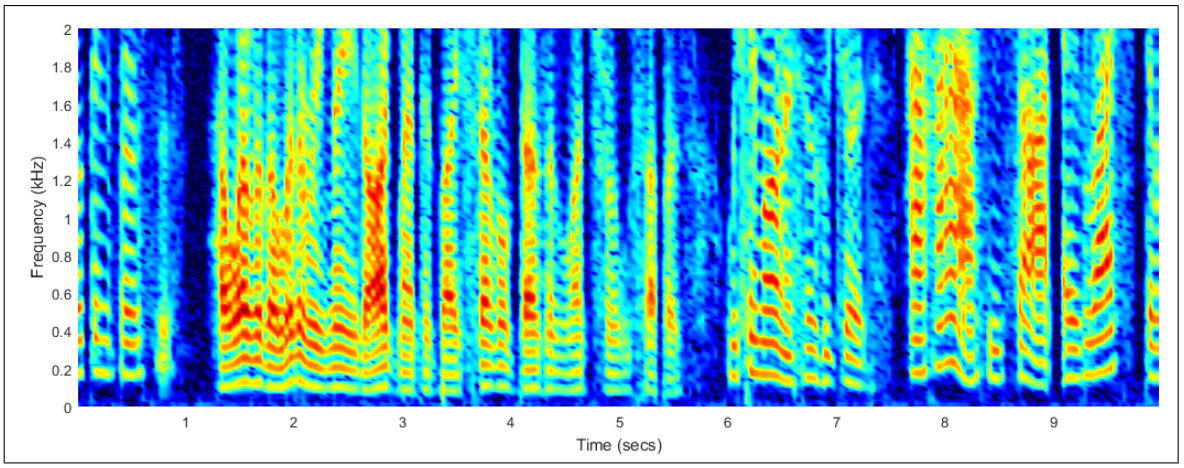

Figure 6.21: Clean Signal

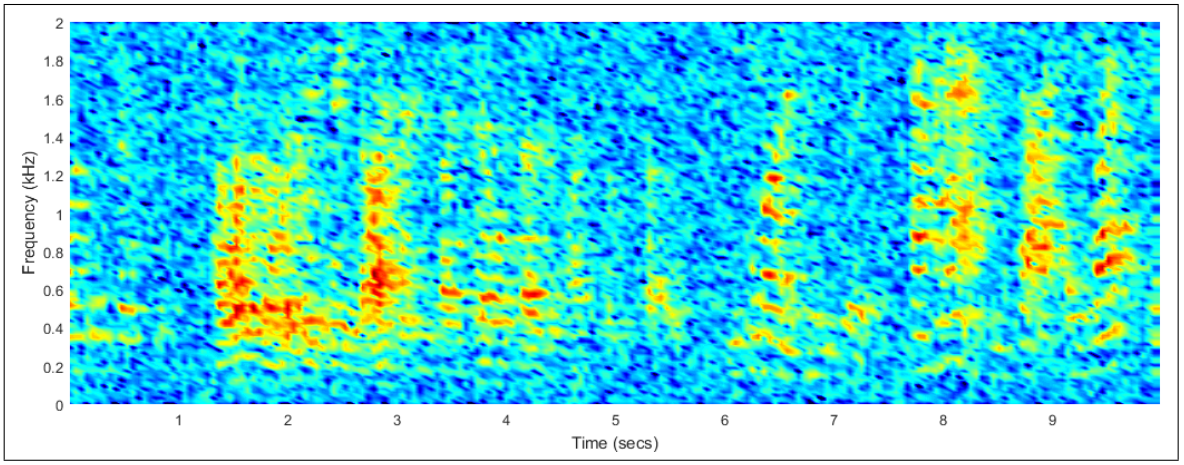

Figure 6.22: Reverberant Signal

The clear denoising and dereverberation of the reverberant speech signal can be seen above. It can also be noted that the noise performance of the MAPBENCH algorithm and the VBBENCH algorithm clearly surpass the performance of the MLBENCH algorithm. The subjective test above can be taken as evidence of the enhancement incurred by the BENCH algorithms. The performance increase was primarily because of the prior included in the MAPBENCH algorithm and the realistic modelling of the source vector as a random variable in the VBBENCH algorithm.

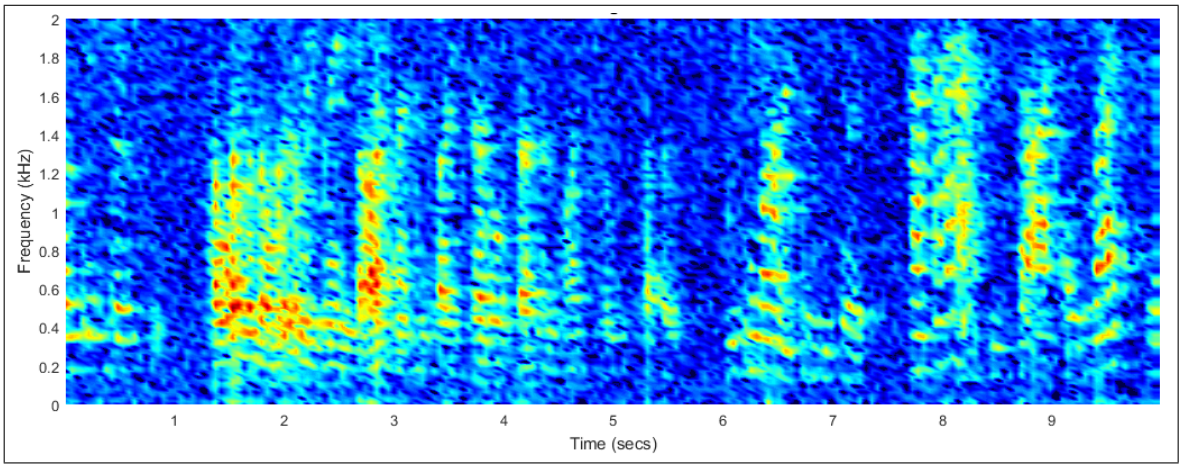

Figure 6.23: MLBENCH enhanced signal

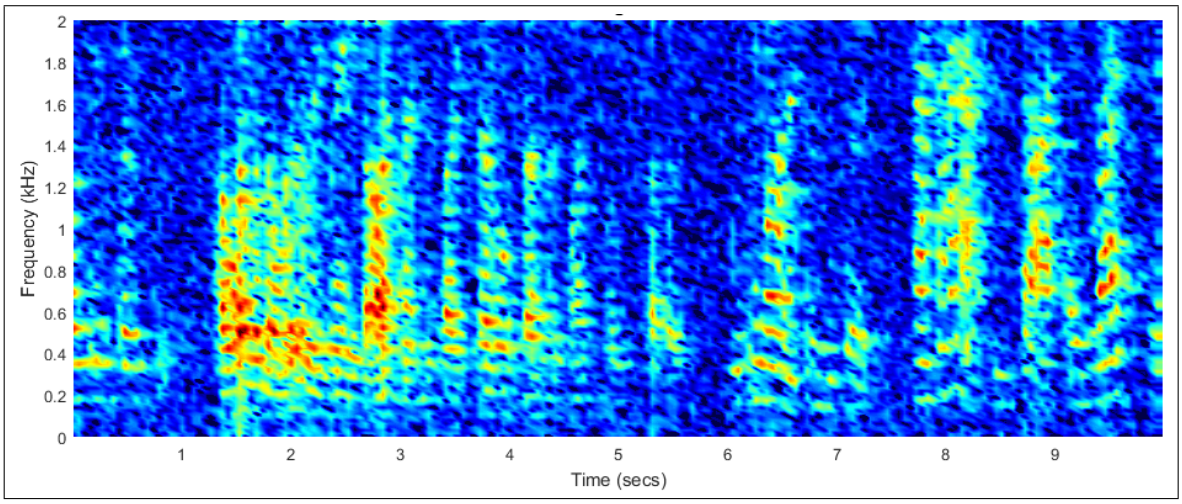

Figure 6.24: MAPBENCH enhanced signal

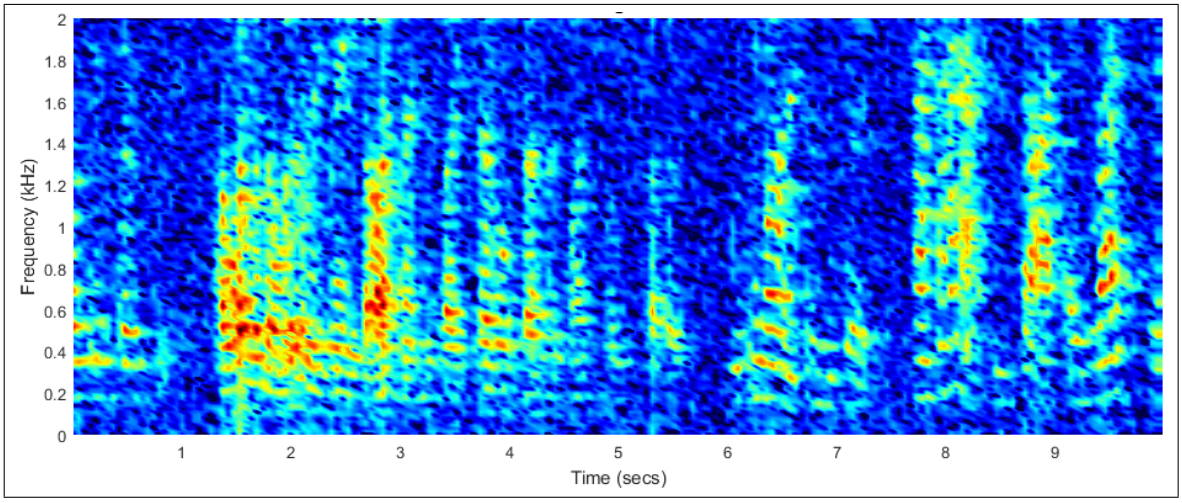

Figure 6.25: VBBENCH enhanced signal

Chapter 7

Conclusion

From the results shown in the previous chapter, it was apparent, that using objective and subjective measures, we were able to improve the reverberant signal just by denoising it. Thus in this document, we were able to examine the different builds of the BENCH algorithms and compare between their performances. In addition, since the MAP and VB BENCH algorithms included a prior to the estimate, a considerable improvement in the denoising process was seen. It is seen that the performance of the MAPBENCH and VBBENCH algorithms surpasses the performance of the MLBENCH algorithm. This is mainly due to the fact that MAP estimates introduces a prior and also that the VB-estimates are a result of full likelihood integration. The results of our simulations prove that the algorithms correctly infer the observation noise parameters while wrongly estimating the channel parameters which unsuccessfully leads to more reverberation. If this problem is circumvented, this family of algorithms can be the basis of various applications since they are computationally efficient and can be implemented in real-time. In addition, further improvement to this class of algorithms can be introduced in the area of modelling. The usage of more realistic speech models for these algorithms has the potential of yielding much appreciated results.

Bibliography

[1] Dominic Schmid. *Multichannel dereverberation and noise reduction for hands-free speech communication systems*. PhD thesis, Ph. D. dissertation, Inst. of Commun. Acoust., Ruhr-Univ. Bochum, Bochum, Germany, 2014.

[2] John R Pierce. Whither speech recognition? *The journal of the acoustical society of america*, 46(4B):1049–1051, 1969.

[3] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[4] Patrick A Naylor and Nikolay D Gaubitch. *Speech dereverberation*. Springer Science & Business Media, 2010.

[5] Jacob Benesty, M Mohan Sondhi, and Yiteng Huang. *Springer handbook of speech processing*. Springer Science & Business Media, 2007.

[6] James R Hopgood and Christine Evers. Towards single-channel blind dereverberation of speech from a moving speaker.

[7] Jian Li and Petre Stoica. *MIMO radar signal processing*. Wiley Online Library, 2009.

[8] A Lee Swindlehurst and Thomas Kailath. A performance analysis of subspace-based methods in the presence of model errors. i. the music algorithm. *IEEE Transactions on signal processing*, 40(7):1758–1774, 1992.

[9] Richard Roy and Thomas Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):984–995, 1989.

[10] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[11] Lawrence R Rabiner and Bernard Gold. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, 1, 1975.

[12] John J Shynk et al. Frequency-domain and multirate adaptive filtering. *IEEE Signal Processing Magazine*, 9(1):14–37, 1992.

[13] Ian Glover and Peter M Grant. *Digital communications*. Pearson Education, 2010.

[14] James R Hopgood, Christine Evers, and Steven Fortune. Bayesian single channel blind dereverberation of speech from a moving talker. In *Speech Dereverberation*, pages 219–270. Springer, 2010.

[15] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[16] Gerald Enzner and Peter Vary. Frequency-domain adaptive kalman filter for acoustic echo control in hands-free telephones. *Signal Processing*, 86(6):1140–1156, 2006.

[17] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[18] Sarmad Malik and Gerald Enzner. Online maximum-likelihood learning of time-varying dynamical models in block-frequency-domain. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3822–3825. IEEE, 2010.

[19] James L Flanagan, Arun C Surendran, and Ea-Ee Jan. Spatially selective sound capture for speech and audio processing. *Speech Communication*, 13(1-2):207–222, 1993.

[20] Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. *Signals and systems*, volume 2. Prentice-Hall Englewood Cliffs, NJ, 1983.

[21] Dominic Schmid, Sarmad Malik, and Gerald Enzner. An expectation-maximization algorithm for multichannel adaptive speech dereverberation in the frequency-domain. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 17–20. IEEE, 2012.

[22] Yiteng Huang, Jacob Benesty, and Jingdong Chen. Optimal step size of the adaptive multichannel lms algorithm for blind simo identification. *IEEE Signal Processing Letters*, 12(3):173–176, 2005.

[23] Dominic Schmid, Gerald Enzner, Sarmad Malik, Dorothea Kolossa, and Rainer Martin. Variational bayesian inference for multichannel dereverberation and noise reduction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(8):1320–1335, 2014.

[24] Thomas Lotter and Peter Vary. Speech enhancement by map spectral amplitude estimation using a super-gaussian speech model. *EURASIP journal on applied signal processing*, 2005:1110–1126, 2005.

[25] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[26] Yi Hu and Philipos C Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 16(1):229–238, 2008.

[27] Gabriella Tognola, Stefano Moriconi, and Emma Chiaramello. On the use of acoustic simulations and pesq measures for the prediction of speech intelligibility in sensorineural hearing loss. In *6th European Conference of the International Federation for Medical and Biological Engineering*, pages 9–12. Springer, 2015.

[28] Kazuhiro Kondo. Speech quality. In *Subjective Quality Measurement of Speech*, pages 7–20. Springer, 2012.

[29] Tiago H Falk, Chenxi Zheng, and Wai-Yip Chan. A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(7):1766–1774, 2010.

[30] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, David S Pallett, Nancy L Dahlgren, and Victor Zue. Timit acoustic-phonetic continuous speech corpus. *Linguistic data consortium, Philadelphia*, 33, 1993.

[31] Jont B Allen and David A Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.