Graz University of Technology

Blind Acoustic Beamforming Based on Generalized Eigenvalue Decomposition

Author: Nikolaus Fankhauser, 1073079

Date: Graz, September 14, 2015
Rev.: alpha 1.0

Contents

1 Introduction

The goal of this "Master Project" is to enhance a speech signal in the case of presence of stationary noise. In reverberated environments signal leakage and insufficient noise suppression decrease the quality of the desired speech signal. With the help of a gradient ascent adaptation algorithm the signal to noise ratio of the speech signal can be increased in the described suboptimal conditions. Two versions of this algorithm are presented: the deterministic gradient ascent and the stochastic gradient ascent. A blocking matrix that is based on this algorithm is introduced and is usually part of a generalized side-lobe canceller. A generalized side-lobe canceller (see Fig.1.1) consists of a fixed beamformer, a blocking matrix and an adaptive interference (noise) canceller. This work deals only with the derivation of the blocking matrix.

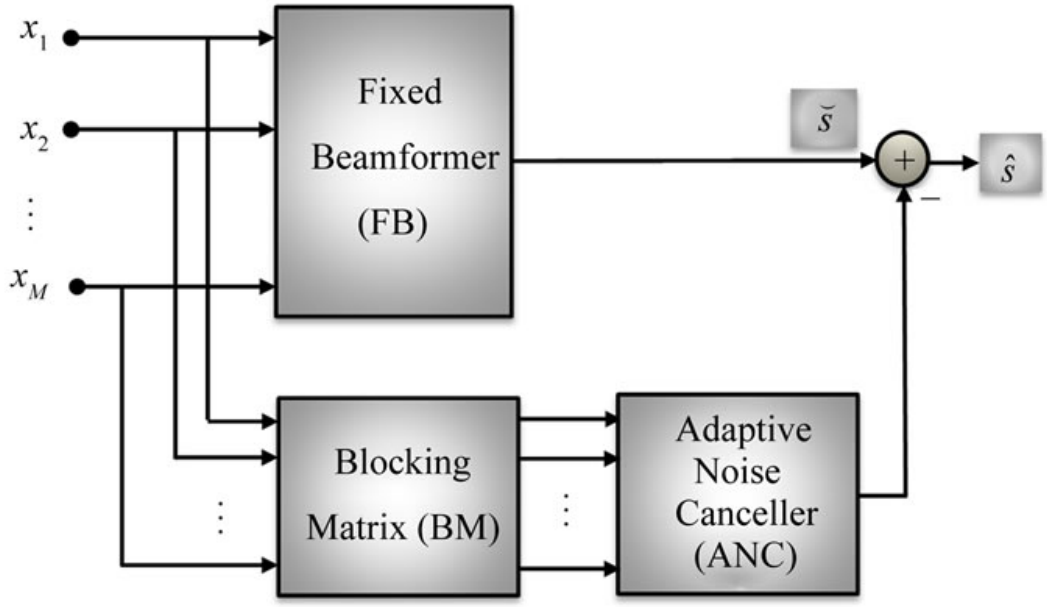

Figure 1.1: Structure of a generalized side-lobe canceller

2 Blocking Matrix

This chapter deals with the derivation of the necessary formulas for calculating the blocking matrix and the optimal transfer functions and is based on [1] and [2].
To capture the sound in a reverberant enclosure, an array of M-microphones is used. Each microphone provides a signal consisting of two components: a signal component $s_i(l)$ and stationary noise $n_i(l)$. The signal component is obtained by the convolution of the channel impulse response $h_i(l)$ with the desired speech source signal $s_0(l)$.

$$x_{\mathrm{i}}(l) = s_{\mathrm{i}}(l) + n_{\mathrm{i}}(l) = (s_0 * h_{\mathrm{i}})(l) + n_{\mathrm{i}}(l), i \geq 1, i \in \mathbb{Z}. \tag{2.1}$$

The optimal transfer function variables are calculated in the frequency domain. Therefore, $x_i(l)$ is Fourier-transformed and results in $X_i[k]$ where k indicates the frequency index. $X_i[k]$ can be represented by a vectorial notation: $\boldsymbol{X}[k] = (X_1[k], ..., X_M[k])^T$ and also the filter coefficients can be written as vectors: $\boldsymbol{F}[k] = (F_1[k], ..., F_M[k])^T$. With these definitions the beamformer output results in

$$Y[k] = \boldsymbol{F}^H \boldsymbol{X}[k]. \tag{2.2}$$

Our goal is to derive a vector of optimal transfer functions $\boldsymbol{F}_{SNR}[k]$ with the help of the Maximum-SNR-criterion:

$$\boldsymbol{F}_{\mathrm{SNR}}[k] := \underset{\boldsymbol{F}[k]}{\mathrm{argmax}} \mathrm{SNR[k]} \tag{2.3}$$

The SNR of the beamformer output can be calculated with the formula:

$$\mathrm{SNR[k]} = \frac{\boldsymbol{F}^H[k]\boldsymbol{\Phi}_{\mathrm{XX}}[k]\boldsymbol{F}[k]}{\boldsymbol{F}^H[k]\boldsymbol{\Phi}_{\mathrm{NN}}[k]\boldsymbol{F}[k]} - 1, \tag{2.4}$$

where $\boldsymbol{\Phi}_{\mathrm{XX}}[k]$ is the cross power spectral density of the microphone signal and $\boldsymbol{\Phi}_{\mathrm{NN}}[k]$ is the cross power spectral density of the noise signal. The desired speech signal and the stationary noise are uncorrelated. Therefore, $\boldsymbol{\Phi}_{\mathrm{XX}}[k]$ can be replaced by $\boldsymbol{\Phi}_{\mathrm{NN}}[k] + \boldsymbol{\Phi}_{\mathrm{SS}}[k]$. Using this relationship eq.(2.4) can be rewritten as

$$\mathrm{SNR[k]} = \frac{\boldsymbol{F}^H[k]\boldsymbol{\Phi}_{\mathrm{SS}}[k]\boldsymbol{F}[k]}{\boldsymbol{F}^H[k]\boldsymbol{\Phi}_{\mathrm{NN}}[k]\boldsymbol{F}[k]}. \tag{2.5}$$

In general, a Rayleigh quotient is defined as [3]:

$$R_{\boldsymbol{A}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^H \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^H \boldsymbol{x}}. \tag{2.6}$$

There is a close relationship between the eigenvalues of a matrix and the Rayleigh quotient. Is $\boldsymbol{v}$ an eigenvector of $\boldsymbol{A}$ and λ the eigenvalue belonging to this eigenvector, then the Rayleigh quotient can be written as:

$$R_{\boldsymbol{A}}(\boldsymbol{v}) = \frac{\boldsymbol{v}^H \boldsymbol{A} \boldsymbol{v}}{\boldsymbol{v}^H \boldsymbol{v}} = \frac{\boldsymbol{v}^H \lambda \boldsymbol{v}}{\boldsymbol{v}^H \boldsymbol{v}} = \lambda. \tag{2.7}$$

Equation (2.5) can be converted to

$$\mathrm{SNR[k]} = \frac{\boldsymbol{F}^H[k]\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\boldsymbol{\Phi}_{\mathrm{SS}}[k]\boldsymbol{F}[k]}{\boldsymbol{F}^H[k]\boldsymbol{F}[k]} \tag{2.8}$$

if $\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]$ is not singular. One can see that eq.(2.8) has the same structure as eq.(2.7) and therefore

$$\mathrm{SNR[k]} = \lambda. \tag{2.9}$$

To achieve the Max-SNR-criterion an eigenvalue problem has to be solved, where the eigenvector of interest is the one belonging to the largest eigenvalue. A common eigenvalue problem has the following structure:

$$(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{x} = 0. \tag{2.10}$$

In the case of calculating the eigenvalues of the Max-SNR-criterion, matrix $\boldsymbol{A}$ is substituted by $\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}\boldsymbol{\Phi}_{\mathrm{XX}}$ and $\boldsymbol{x}$ is replaced by $\boldsymbol{F}$, where $\boldsymbol{\Phi}_{\mathrm{NN}}$ is not singular:

$$\Big(\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\boldsymbol{\Phi}_{\mathrm{XX}}[k] - \lambda\boldsymbol{I}\Big)\boldsymbol{F}[k] = 0. \tag{2.11}$$

Reformulating this equation leads to

$$\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\boldsymbol{\Phi}_{\mathrm{XX}}[k]\boldsymbol{F}[k] = \lambda\boldsymbol{F}[k]. \tag{2.12}$$

$\boldsymbol{\Phi}_{\mathrm{XX}}[k]$ can be expressed by the power spectral density of the source speech signal $S_0[k]$ in combination with the transfer function $\boldsymbol{H}(\boldsymbol{l}_t, k)$ from the target source at position $\boldsymbol{l}_t$ to the ith sensor:

$$\boldsymbol{\Phi}_{\mathrm{XX}}[k] = \Phi_{S_0S_0}[k]\boldsymbol{H}[k]\boldsymbol{H}^H[k] + \boldsymbol{\Phi}_{\mathrm{NN}}[k]. \tag{2.13}$$

Substituting $\boldsymbol{\Phi}_{\mathrm{XX}}$ in eq.(2.12) by the right side of eq.(2.13) results in:

$$\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\Big(\Phi_{S_0S_0}[k]\boldsymbol{H}[k]\boldsymbol{H}^H[k] + \boldsymbol{\Phi}_{\mathrm{NN}}[k]\Big)\boldsymbol{F}[k] = \lambda\boldsymbol{F}[k]. \tag{2.14}$$

Expanding the parenthesis in eq.(2.14) the following equation is obtained:

$$\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\Phi_{S_0S_0}[k]\boldsymbol{H}[k]\boldsymbol{H}^H[k]\boldsymbol{F}[k] + \boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}[k] = \lambda\boldsymbol{F}[k]. \tag{2.15}$$

Equation(2.15) is rewritten as:

$$\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\Phi_{S_0S_0}[k]\boldsymbol{H}[k]\boldsymbol{H}^H[k]\boldsymbol{F}[k] = (\lambda - 1)\boldsymbol{F}[k]. \tag{2.16}$$

If $\Phi_{S_0S_0} \neq 0$ eq.(2.16) can be reformulated as:

$$\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\boldsymbol{H}[k]\boldsymbol{H}^H[k]\boldsymbol{F}[k] = \frac{\lambda[k] - 1}{\Phi_{S_0S_0}[k]}\boldsymbol{F}[k]. \tag{2.17}$$

With this equation and the fact that the rank of the matrix $\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\boldsymbol{H}[k]\boldsymbol{H}^H[k]$ is one, the vector of the optimal transfer functions is:

$$\boldsymbol{F}_{\mathrm{SNR}}[k] = \zeta[k]\boldsymbol{\Phi}_{\mathrm{NN}}^{-1}[k]\boldsymbol{H}[k], \tag{2.18}$$

where ζ is an arbitrary complex constant.
The generalized eigenvector blocking matrix should produce noise reference signals orthogonal

to a speech reference. The optimal filter coefficients are needed to design a projection into the orthogonal complement of $\boldsymbol{H}[k]$. The reference signal that should ideally contain only speech, is denoted as:

$$Y_{\mathrm{SNR}}[k] := \boldsymbol{F}_{\mathrm{SNR}}^{H}[k]\boldsymbol{X}[k]. \tag{2.19}$$

and the noise signal $\boldsymbol{U}[k]$ can be defined with the unknown projection vector $\boldsymbol{P}[k]$ as:

$$\boldsymbol{U}[k] := \boldsymbol{X}[k] - \boldsymbol{P}[k]Y_{\mathrm{SNR}}[k]. \tag{2.20}$$

If signals are orthogonal, their cross-correlation vector should be $\mathbf{0}$. In this case the expectation of the noise signal $\boldsymbol{U}[k]$ and the reference signal $Y_{SNR}[k]$ has to be 0.

$$E[\boldsymbol{U}[k]Y_{\mathrm{SNR}}^{*}[k]] \overset{!}{\approx} \mathbf{0}. \tag{2.21}$$

By inserting eq.(2.19) and eq.(2.20) in eq.(2.21) solving the new equation for $\boldsymbol{P}[k]$ is possible:

$$E\left[\Big(\boldsymbol{X}[k] - \boldsymbol{P}[k]Y_{\mathrm{SNR}}[k]\Big)\boldsymbol{F}_{\mathrm{SNR}}^{H}[k]\boldsymbol{X}[k]\right] = \mathbf{0}. \tag{2.22}$$

Replacing $Y_{SNR}[k]$ by eq.(2.19) leads to:

$$E\left[\Big(\boldsymbol{X}[k] - \boldsymbol{P}[k]\boldsymbol{F}_{\mathrm{SNR}}^{H}[k]\boldsymbol{X}[k]\Big)\boldsymbol{F}_{\mathrm{SNR}}^{H}[k]\boldsymbol{X}[k]\right] = \mathbf{0}. \tag{2.23}$$

Eliminating the parenthesis of the expression gives the following equation:

$$E\left[\boldsymbol{X}[k]\boldsymbol{F}_{\mathrm{SNR}}^{H}[k]\boldsymbol{X}[k] - \boldsymbol{P}[k]\boldsymbol{F}_{\mathrm{SNR}}^{H}[k]\boldsymbol{X}[k]\boldsymbol{F}_{\mathrm{SNR}}^{H}[k]\boldsymbol{X}[k]\right] = \mathbf{0}. \tag{2.24}$$

Because of the linearity of the expectation value and with the knowledge of $E[\boldsymbol{X}[k]\boldsymbol{X}^{H}[k]] = \boldsymbol{\Phi}_{\mathrm{XX}}[k]$, it can be written:

$$\boldsymbol{\Phi}_{\mathrm{XX}}[k]\boldsymbol{F}_{\mathrm{SNR}}[k] - \boldsymbol{P}[k]\boldsymbol{F}_{\mathrm{SNR}}^{H}[k]\boldsymbol{\Phi}_{\mathrm{XX}}[k]\boldsymbol{F}_{\mathrm{SNR}}[k] = \mathbf{0}. \tag{2.25}$$

Equation(2.25) can be written without using the expectation operator because it is only applied to the variable $\boldsymbol{X}[k]$ here. Solving eq.(2.22) for $\boldsymbol{P}[k]$ gives us the projection vector as:

$$\boldsymbol{P}[k] := \frac{\boldsymbol{\Phi}_{\mathrm{XX}}[k]\boldsymbol{F}_{\mathrm{SNR}}[k]}{\boldsymbol{F}_{\mathrm{SNR}}^{H}[k]\boldsymbol{\Phi}_{\mathrm{XX}}[k]\boldsymbol{F}_{\mathrm{SNR}}[k]}. \tag{2.26}$$

With the input signal $\boldsymbol{X}[k]$ and defining the noise reference signal as output of the blocking matrix $\boldsymbol{B}[k]$ the following equation can be found for $\boldsymbol{U}[k]$:

$$\boldsymbol{U}[k] = \boldsymbol{B}^{H}[k]\boldsymbol{X}[k] \tag{2.27}$$

Equation(2.27) is reformulated:

$$\boldsymbol{B}^{H}[k] = \boldsymbol{X}^{-1}[k]\boldsymbol{U}[k]. \tag{2.28}$$

Inserting eq.(2.20) into eq.(2.28) leads to:

$$\boldsymbol{B}^{H}[k] = \boldsymbol{X}^{-1}[k]\Big(\boldsymbol{X}[k] - \boldsymbol{P}[k]Y_{\mathrm{SNR}}[k]\Big). \tag{2.29}$$

Expanding the parenthesis in eq.(2.29) results in: :

$$\boldsymbol{B}^{H}[k] = \boldsymbol{I}_{\mathrm{M}} - \boldsymbol{X}^{-1}[k]\boldsymbol{P}[k]Y_{\mathrm{SNR}}[k], \tag{2.30}$$

where $\boldsymbol{I}_M$ is the identity matrix of dimension M, and M is the number of microphones. Inserting eq.(2.19) in eq.(2.30) gives the following expression:

$$\boldsymbol{B}^H[k] = \boldsymbol{I}_{\mathrm{M}} - \boldsymbol{X}^{-1}[k]\boldsymbol{P}[k]\boldsymbol{F}^H_{\mathrm{SNR}}[k]\boldsymbol{X}[k]. \tag{2.31}$$

Simplifying the last term of eq.(2.32) results in obtaining the blocking matrix.

$$\boldsymbol{B}^H[k] := \boldsymbol{I}_{\mathrm{M}} - \boldsymbol{P}[k]\boldsymbol{F}_{\mathrm{SNR}}[k]. \tag{2.32}$$

For the calculation of $\boldsymbol{F}_{\mathrm{SNR}}[k]$ the cross power spectral density matrices of the noise and the microphone signals are needed. By using a voice activity detector, $\boldsymbol{\Phi}_{\mathrm{NN}}[k]$ can be estimated in periods where only noise is present in the microphone signals, i.e.,

$$\hat{\boldsymbol{\Phi}}_{\mathrm{NN}}[k] := \frac{1}{K_{\mathrm{N}}} \sum_{m=1}^{K_{\mathrm{N}}} (\boldsymbol{X}[k]\boldsymbol{X}^H[k]) \mid_{\boldsymbol{X}=\boldsymbol{N}}, \tag{2.33}$$

where K_N is the number of frames. The estimation of the averaged cross power spectral density matrix $\boldsymbol{\Phi}_{\mathrm{XX}}[k]$ is done in the same way, with the difference that for $\boldsymbol{\Phi}_{\mathrm{XX}}[k]$ the microphone signals have to contain both speech and noise signals:

$$\hat{\boldsymbol{\Phi}}_{\mathrm{XX}}[k] := \frac{1}{K_{\mathrm{X}}} \sum_{m=1}^{K_{\mathrm{X}}} (\boldsymbol{X}[k]\boldsymbol{X}^H[k]). \tag{2.34}$$

3 Adaptive Eigenvector Tracking

This chapter is based on [2]. The goal of adaptive eigenvector tracking is to maximize the SNR of the beamformer output, and this leads to the following constrained optimization problem:

$$\max_{\mathrm{F^H[k]}} \boldsymbol{F}^H[k]\boldsymbol{\Phi}_{\mathrm{XX}}[k]\boldsymbol{F}[k] \tag{3.1}$$

subject to

$$\boldsymbol{F}^H[k]\boldsymbol{\Phi}_{\mathrm{NN}}[k]\boldsymbol{F}[k] = C[k]. \tag{3.2}$$

With the use of Lagrange multipliers we can form the optimization function:

$$\boldsymbol{J}(\boldsymbol{F},\beta) = \boldsymbol{F}^H[k]\boldsymbol{\Phi}_{\mathrm{XX}}[k]\boldsymbol{F}[k] + \beta[k]\Big(\boldsymbol{F}^H[k]\boldsymbol{\Phi}_{\mathrm{NN}}[k]\boldsymbol{F}[k] - C[k]\Big) \tag{3.3}$$

where $\beta[k]$ is a Lagrange multiplier. The adaptation must be done for every frequency index, but for simplicity the frequency index is omitted in the following derivations. Two different algorithms are examined for getting the filter coefficients: the deterministic gradient ascent and the stochastic gradient ascent. The difference between the deterministic and the stochastic gradient ascent [6] is that in the deterministic gradient ascent all the samples of the training data must be used for a single update of the parameters, whereas in the stochastic gradient ascent one single training sample is sufficient to do the update of the parameters. In general, optimization problems can be written as:

$$\max_{\mathrm{x}} f(x) := \frac{1}{N}\sum_{i=1}^{N} f_{\mathrm{i}}(x), \tag{3.4}$$

where f(x) is a convex function and $\frac{df(x)}{dx}$ is Lipschitz-continuous with constant L. Therefore, the following equation is valid for all x and y:

$$\|\frac{df(x)}{dx} - \frac{df(y)}{dy}\| \leq L\|x-y\|. \tag{3.5}$$

This problem can be solved by a deterministic gradient ascent that leads to the following update rule:

$$x_{\mathrm{k+1}} = x_{\mathrm{k}} + a_{\mathrm{k}}\frac{df(x_{\mathrm{k}})}{dx}, \tag{3.6}$$

where a_{k} denotes the step size. In contrast to the deterministic gradient ascent, the stochastic gradient ascent uses a different equation to update the parameters:

$$x_{\mathrm{k+1}} = x_{\mathrm{k}} + a_{\mathrm{k}}\frac{df_{\mathrm{i}}(x_{\mathrm{k}})}{dx}. \tag{3.7}$$

In the stochastic version only one individual data sample is chosen randomly of a data set of N samples. The advantage of the stochastic version in comparison to the deterministic version is that it is independent of N and in cases of large N the stochastic gradient ascent leads to much faster convergence. However, the error function in the stochastic method is not as well

maximized as in the deterministic method.

3.1 Deterministic Gradient Ascent

The generalized eigenvalue problem is solved by an iterative approach,

$$\boldsymbol{F}_{\kappa+1} = \boldsymbol{F}_{\kappa} + \frac{\mu}{2}\nabla_{\boldsymbol{F}}\boldsymbol{J}(\boldsymbol{F},\beta)) \mid_{\boldsymbol{F}=\boldsymbol{F}_{\kappa}}, \tag{3.8}$$

where μ is the step size parameter of the adaptive algorithm and κ is the iteration counter. It is possible to compute the Lagrange multipliers if the constraint C is fulfilled at the next iteration step $\kappa+1$:

$$\boldsymbol{F}_{\kappa+1}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa+1} \stackrel{!}{=} C. \tag{3.9}$$

In the next step the optimization function is deviated with respect to $\boldsymbol{F}$, and this leads to:

$$\nabla_{\boldsymbol{F}}\boldsymbol{J}(\boldsymbol{F},\beta) = 2\boldsymbol{\Phi}_{\mathrm{XX}}\boldsymbol{F}_{\kappa} + 2\beta\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa}. \tag{3.10}$$

The deviation of the optimization function is inserted into the update rule (eq.(3.8)):

$$\boldsymbol{F}_{\kappa+1} = \boldsymbol{F}_{\kappa} + \frac{\mu}{2}\Big(2\boldsymbol{\Phi}_{\mathrm{XX}}\boldsymbol{F} + 2\beta\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}\Big). \tag{3.11}$$

$\boldsymbol{F}_{\kappa+1}$ and $\boldsymbol{F}_{\kappa+1}^{H}$ in eq.(3.9) are replaced by eq.(3.11). This substitution leads to the following equation:

$$\Big[\boldsymbol{F}_{\kappa} + \frac{\mu}{2}\Big(2\boldsymbol{\Phi}_{\mathrm{XX}}\boldsymbol{F}_{\kappa} + 2\beta\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa}\Big)\Big]^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\Big[\boldsymbol{F}_{\kappa} + \frac{\mu}{2}\Big(2\boldsymbol{\Phi}_{\mathrm{XX}}\boldsymbol{F}_{\kappa} + 2\beta\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa}\Big)\Big] = C. \tag{3.12}$$

Expanding the parenthesis in eq.(3.12) and only taking terms of order $O(\mu)$ into account gives the following expression:

$$\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa} + \mu\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{XX}}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa} + 2\mu\beta\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa} + \mu\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{\Phi}_{\mathrm{XX}}\boldsymbol{F}_{\kappa} = C, \tag{3.13}$$

with $\boldsymbol{\Phi}_{\mathrm{NN}}^{H} = \boldsymbol{\Phi}_{\mathrm{NN}}$ and $\boldsymbol{\Phi}_{\mathrm{XX}}^{H} = \boldsymbol{\Phi}_{\mathrm{XX}}$. $\mu\boldsymbol{F}_{\kappa}^{H}\boldsymbol{F}_{\kappa}$ can be lifted out of eq.(3.13) and so eq.(3.14) is gotten:

$$\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa} + \mu\boldsymbol{F}_{\kappa}^{H}\Big(\boldsymbol{\Phi}_{\mathrm{XX}}\boldsymbol{\Phi}_{\mathrm{NN}} + \boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{\Phi}_{\mathrm{XX}}\Big) + 2\mu\beta\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa} = C. \tag{3.14}$$

By introducing

$$\boldsymbol{\Phi}^{(XN)} = \boldsymbol{\Phi}_{\mathrm{XX}}\boldsymbol{\Phi}_{\mathrm{NN}} + \boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{\Phi}_{\mathrm{XX}} \tag{3.15}$$

and rewriting eq.(3.14) in terms of β, the following formula is obtained:

$$\beta \approx \frac{C - \boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa} - \mu\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}^{(XN)}\boldsymbol{F}_{\kappa}}{2\mu\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa}}. \tag{3.16}$$

Inserting eq.(3.16) into (3.8) leads to a calculation rule for the deterministic gradient ascent algorithm:

$$\boldsymbol{F}_{\kappa+1} = \boldsymbol{F}_{\kappa} + \frac{C - \boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa}}{2\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa}}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa} + \mu\left[\boldsymbol{\Phi}_{\mathrm{XX}}\boldsymbol{F}_{\kappa} - \frac{\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}^{(XN)}\boldsymbol{F}_{\kappa}}{2\boldsymbol{F}_{\kappa}^{H}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa}}\boldsymbol{\Phi}_{\mathrm{NN}}\boldsymbol{F}_{\kappa}\right]. \tag{3.17}$$

In the case of spatially white noise $\boldsymbol{\Phi}_{\mathrm{NN}}$ can be written as

$$\boldsymbol{\Phi}_{\mathrm{NN}} = \sigma_{\mathrm{N}}^2 \boldsymbol{I}_{\mathrm{M}} \tag{3.18}$$

where σ_{N}^2 is the variance of the noise term and $\boldsymbol{I}_{\mathrm{M}}$ is the identity matrix of dimension M. It is assumed that the noise is the same for all microphones only depending on the frequency index. With these simplifications the deterministic gradient ascent algorithm can be rewritten as:

$$\boldsymbol{F}_{\kappa+1} = \frac{1 + \boldsymbol{F}_{\kappa}^H \boldsymbol{F}_{\kappa}}{2\boldsymbol{F}_{\kappa}^H \boldsymbol{F}_{\kappa}} \boldsymbol{F}_{\kappa} + \mu \left[\boldsymbol{\Phi}_{\mathrm{XX}} \boldsymbol{F}_{\kappa} - \frac{\boldsymbol{F}_{\kappa}^H \boldsymbol{\Phi}_{\mathrm{XX}} \boldsymbol{F}_{\kappa}}{\boldsymbol{F}_{\kappa}^H \boldsymbol{F}_{\kappa}} \boldsymbol{F}_{\kappa} \right] \tag{3.19}$$

where $C \stackrel{!}{=} \sigma_{\mathrm{N}}^2$.

3.2 Stochastic Gradient Ascent

In a practical scenario the cross power spectral density matrices of the speech-plus-noise signals and of the noise only signals at the microphones have to be estimated from the incoming data. Here, $\boldsymbol{\Phi}_{\mathrm{NN}}[k]$ is estimated in speech pauses and is still a good estimate if the speech signal is present, because the noise is assumed to be stationary, or respectively changes its statistics on a much longer time scale then the speech signal.

$$\hat{\boldsymbol{\Phi}}_{\mathrm{NN},\boldsymbol{\rho}} = \epsilon \cdot \hat{\boldsymbol{\Phi}}_{\mathrm{NN},\boldsymbol{\rho-1}} + (1-\epsilon) \cdot \boldsymbol{X}_{\rho} \boldsymbol{X}_{\rho}^H \tag{3.20}$$

The variable ρ stands for the frame counter and not for the iteration index in the case of the stochastic gradient algorithm; ϵ is a smoothing constant and determines the weighting of the previous samples. The present samples are weighted by $1-\epsilon$; ϵ only takes values between 0 and 1 and is chosen closely to one. In general speech is a highly non-stationary signal and so the power spectral density matrix is calculated by the instantaneous estimate

$$\hat{\boldsymbol{\Phi}}_{\mathrm{XX},\boldsymbol{\rho}} = \boldsymbol{X}_{\rho} \boldsymbol{X}_{\rho}^H. \tag{3.21}$$

By replacing $\boldsymbol{\Phi}_{\mathrm{NN}}[k]$ and $\boldsymbol{\Phi}_{\mathrm{XX}}[k]$ by eq.(3.20) and eq.(3.21) in eq.(3.15) the stochastic version of the gradient ascent algorithm can be written as:

$$\boldsymbol{F}_{\rho+1} = \boldsymbol{F}_{\rho} + \frac{C - \boldsymbol{F}_{\rho}^H \boldsymbol{G}_{\rho}}{2\boldsymbol{G}_{\rho}^H \boldsymbol{G}_{\rho}} \boldsymbol{G}_{\rho} + \mu Y_{\rho}^* [\boldsymbol{X}_{\rho} - \frac{A_{\rho} \boldsymbol{G}_{\rho}}{2\boldsymbol{G}_{\rho}^H \boldsymbol{G}_{\rho}}], \tag{3.22}$$

where

$$\boldsymbol{G}_{\rho} = \hat{\boldsymbol{\Phi}}_{\boldsymbol{NN\rho}} \boldsymbol{F}_{\rho} \tag{3.23}$$

$$A_{\rho} = Y_{\rho} / Y_{\rho}^* \boldsymbol{X}_{\rho}^H \boldsymbol{G}_{\rho} + \boldsymbol{G}_{\rho}^H \boldsymbol{X}_{\rho} \tag{3.24}$$

$$Y[k] = \boldsymbol{F}^H \boldsymbol{X}[k] \tag{3.25}$$

4 Simulation Results

4.1 Synthetic Speech Data

In this section the implementation of the algorithm was controlled by reproducing the scenario "B" in chapter "V. Simulation Results" of [2]. To get simulation results in the case of synthetic speech data a room of dimension (6m) x (5m) x (3m) was created. A five element linear microphone array was used that was located 0.5 m away form the wall (y-axis) in a height of 1.5 m and in the middle of the x-axis (see Fig. 3.1). The speech source was placed 0.8 m away from the centre of the microphone array at an angle of 45° relative to broadside. The noise source was located 0.8 m away from the center of the microphone array at an angle of 110° relative to broadside.

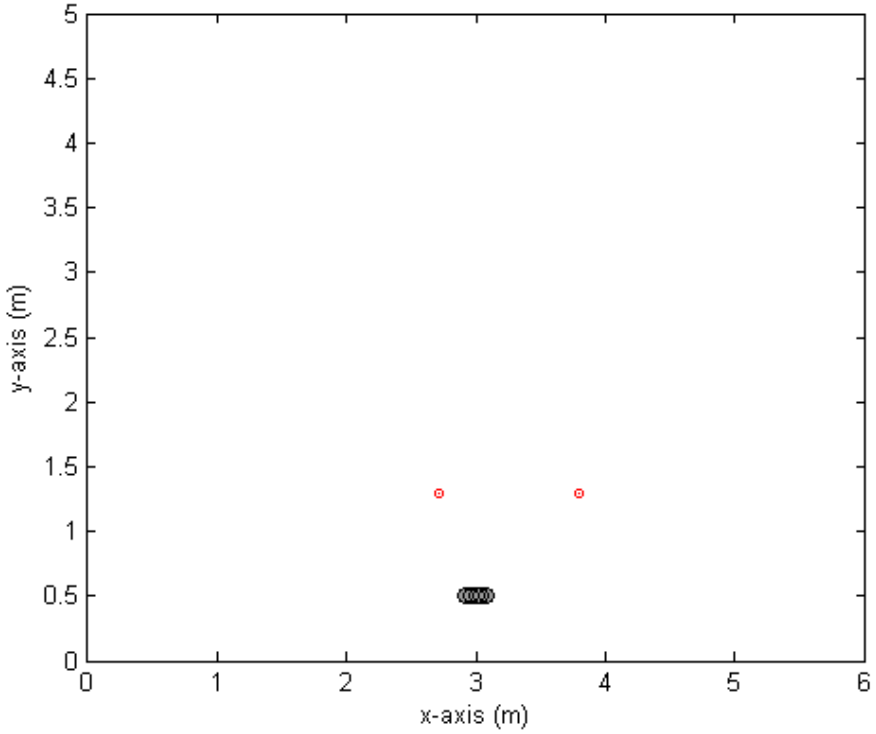

Figure 4.1: Positions of the speech and the noise source

The sampling rate was 12 kHz. The reverberation time T_{60} was calculated for each sensor of the microphone array independently for the case of speech presence and noise presence. In the case of speech only, the speech source was switched on and in the noise scenario only the noise source was switched on. The calculation of the reverberation time was done with the help of the Lehmann Johansson's ISM implementations [5] that are based on the image method. The adaptation of the filter coefficients had to be carried out for every frequency index separately. The index that lies around 1 kHz is presented as a typical example outcome. The simulation was carried out with 1000 iteration steps and in each step the speech and the noise sources were generated as random, complex Gaussian numbers of unit variance and zero mean. These complex

numbers were multiplied with the transfer functions of the microphones that were obtained from the calculation described above. To form the whole microphone signal, the outcome of these multiplications, two vectors, were added up. The ratio of the output SNR to the input SNR is used to analyse the performance of the adaptation algorithm.

$$SNR_\rho = 10log\Big(\frac{\boldsymbol{F}_\rho^H \boldsymbol{\Phi}_{\mathrm{XX},\boldsymbol{\rho}} \boldsymbol{F}_\rho}{\boldsymbol{F}_\rho^H \boldsymbol{\Phi}_{\mathrm{NN},\boldsymbol{\rho}} \boldsymbol{F}_\rho} - 1\Big). \tag{4.1}$$

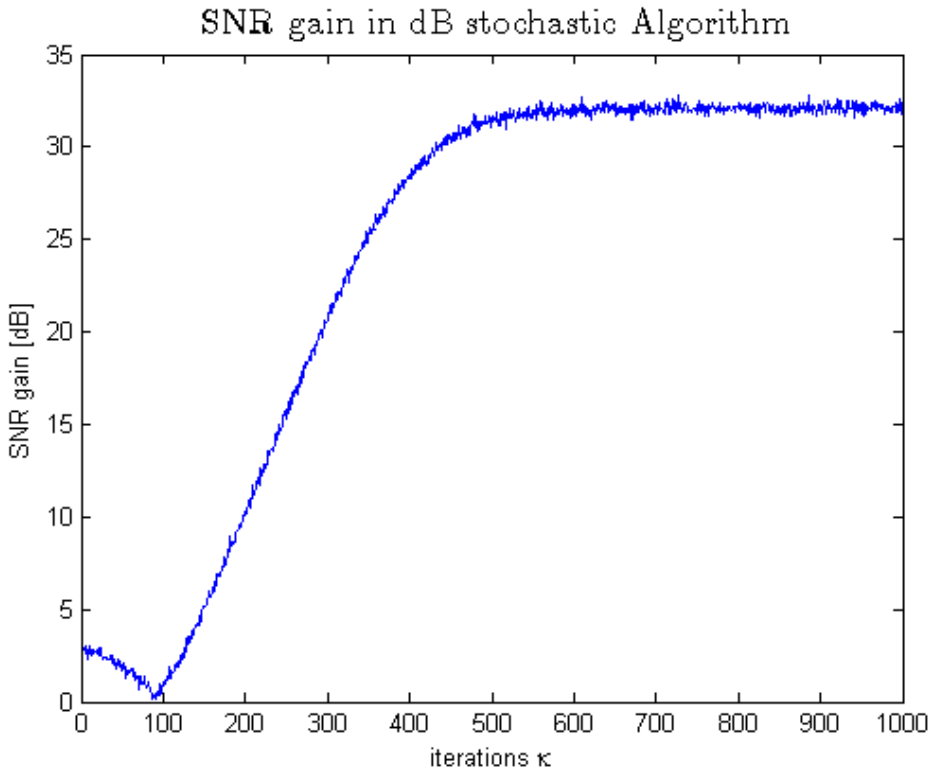

Figure 4.2: SNR gain in dB achieved with the stochastic gradient algorithm

For creating figure 3.2 the value for T_{60} was set to 0.05 s and $\hat{\boldsymbol{\Phi}}_{\mathrm{XX},\boldsymbol{\rho}}$ was calculated according to eq.(2.17). The noisy part of speech is assumed to be known and used for the noise power estimation. $\hat{\boldsymbol{\Phi}}_{\mathrm{NN},\boldsymbol{\rho}}$ was estimated for 1000 iterations only from the noisy parts of speech with eq.(2.16) where ϵ is set to 0.99. In the subsequent adaptation of the filter coefficients with the Stochastic Gradient algorithm, only the fixed value for $\hat{\boldsymbol{\Phi}}_{\mathrm{NN},1000}$ was used. This approach simulates the case in speech enhancement where the noise power spectral density is only calculated during speech pauses and kept constant for the adaptation of the filter coefficients during presence of speech. To find these two cases in practice, a simple voice activity detector was implemented that is based on the power of the speech file. The curve given in the figure shows the average of 1000 Monte Carlo simulations.
The initialization of the parameters and the calculation of the step size are important parts in the usage of adaptation algorithms. The filter coefficients were initialized as complex random variables taken from the standard normal distribution. The stepsize was chosen as $\alpha = 0.01$ normalized on the temporally smoothed input power at every iteration step:

$$\mu_\kappa = \frac{\alpha}{\bar{P}_{\mathrm{X},\rho}} \tag{4.2}$$

4.2 Real Speech Data

For applying the algorithm on real speech data, a database of room impulse responses is needed to get meaningful simulation results. The database was created by varying different parameters of the setup, the room had the same dimensions as in the synthetic case. In the following lines the different values of the parameters are listed:

Impulse response T_{60} = {50, 100, 200, 300, 400, 500} ms
Sampling frequency f_s = {16, 32, 48} kHz
Number of microphone channels N_c = {3, 5, 7}
Distance between microphones d = {4, 8, 12} cm

The database contains a single impulse response for all possible combinations of values, e.g. T_{60} = 50 ms, f_s = 16 kHz, N_c = 3, d = 4 cm, and by using 24 different source positions for every combination. The microphones were placed 0.5 m away from the wall (y-axis), in a height of 1.5 m and in the middle of the x-axis. The sound sources were located in four rows (see Fig.3.1) in a distance of 0.8 m (first row), 1.6 m (second row), 2.4 m (third row) and 3.2 m (fourth row) away from the microphone array in a height of 1.5 m. The sources were placed at the positions 0.5, 1.5, 2.5, 3.5, 4.5 and 5.5 m in x-direction for every row, as you can see in the following figure for a 3-element microphone array.

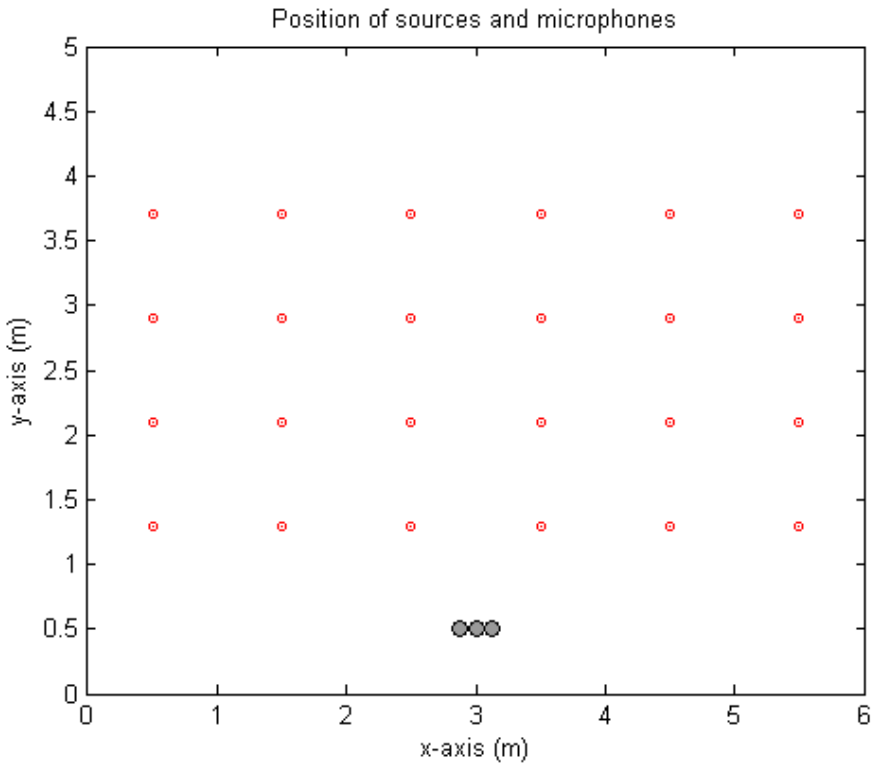

Figure 4.3: Positions of the 24 sources and the microphones in the case of a 3-microphone-array

The database of room impulse responses is needed for the convolution of the real speech signal with the appropriate room impulse response. A corpus of 6975 speech files, with SNR values between -10 and 30 dB, was determined to evaluate the stochastic approximation algorithm. The overall results for 100 iterations of the algorithm are shown in a cumulative distribution function (CDF). One can see (Fig.4.4) that the algorithm leads to a SNR (the SNR was calculated

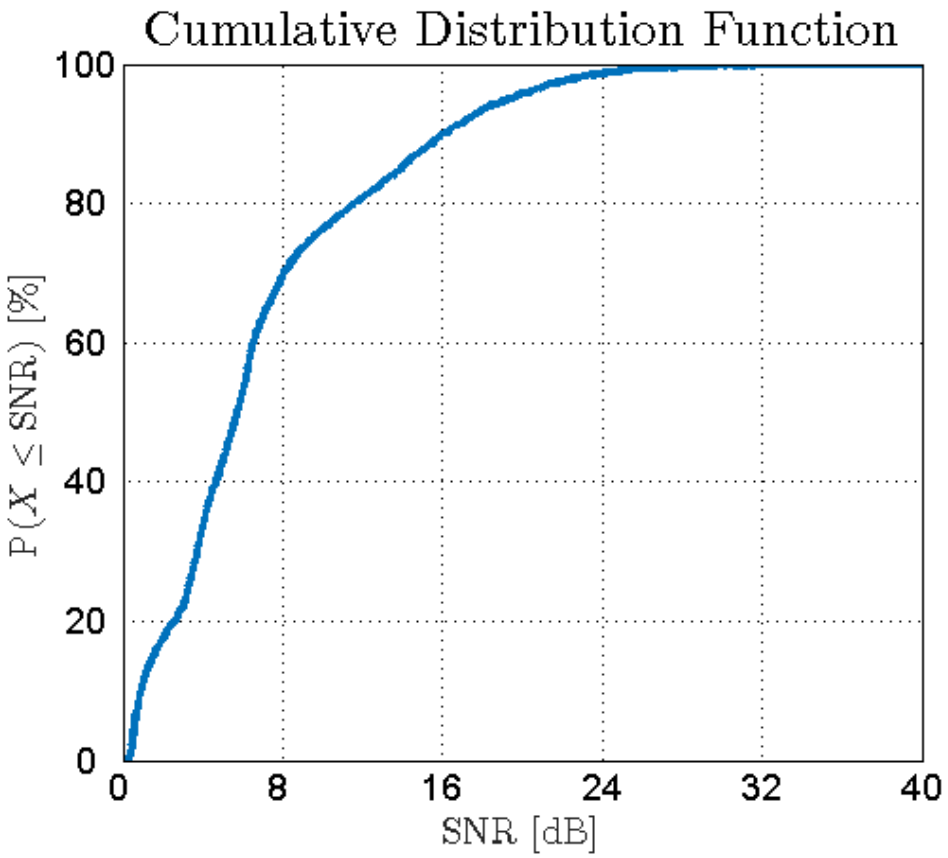

Figure 4.4: CDF of the 6975 analysed speech files

according to [4]) between 0 and 8 dB for 70% of the analysed files, a SNR of 8 to 16 dB is achieved for 20% of the data under test and for the remaining 10% a SNR between 16 and 40 dB can be obtained. In Fig.4.5 to Fig.4.7 the cumulative distribution function is shown separately for three different sampling frequencies. These figures show the same progress with slightly different values. Therefore, the sampling frequency does not strongly affect the performance of the algorithm. Figures 4.8, 4.9 and 4.10 show the cumulative distribution function for a different number of microphones. These figures are very similar too, but the worst result is obtained in the case of 3 microphones (see Fig.4.8). In the case of 5 and 7 microphones the CDF is nearly equal (see. Fig.4.9 and Fig.4.10). It is difficult to evaluate these results because comparable data do not exist.

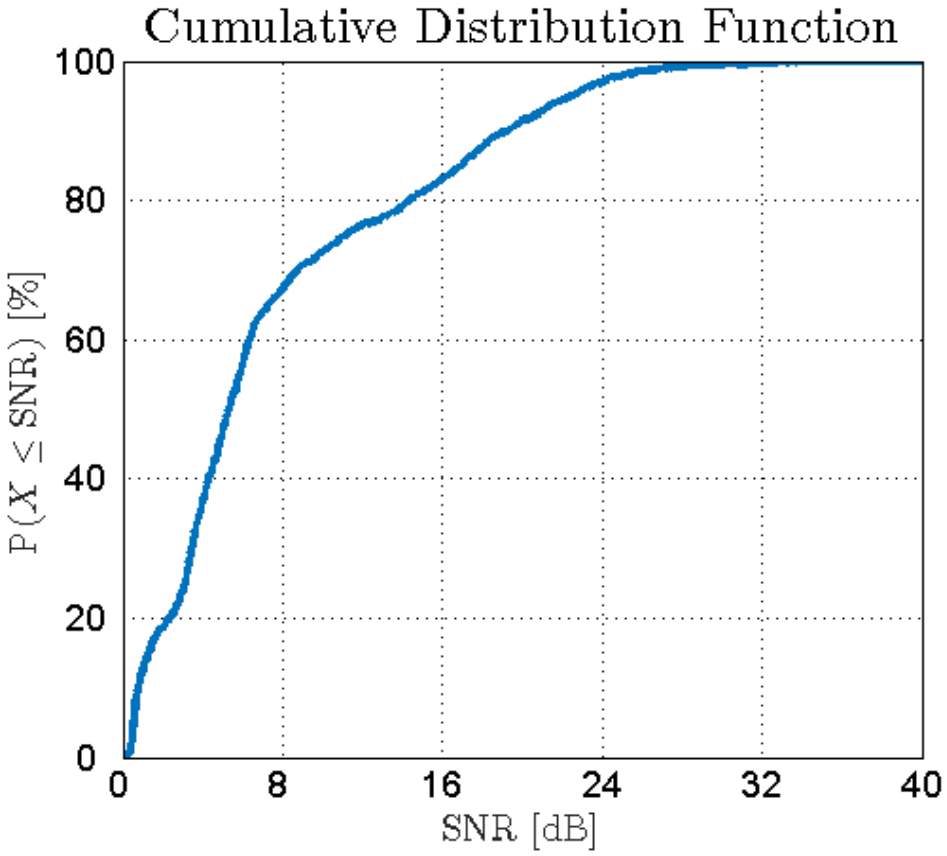

Figure 4.5: CDF of 2408 speech files with f_s = 16kHz

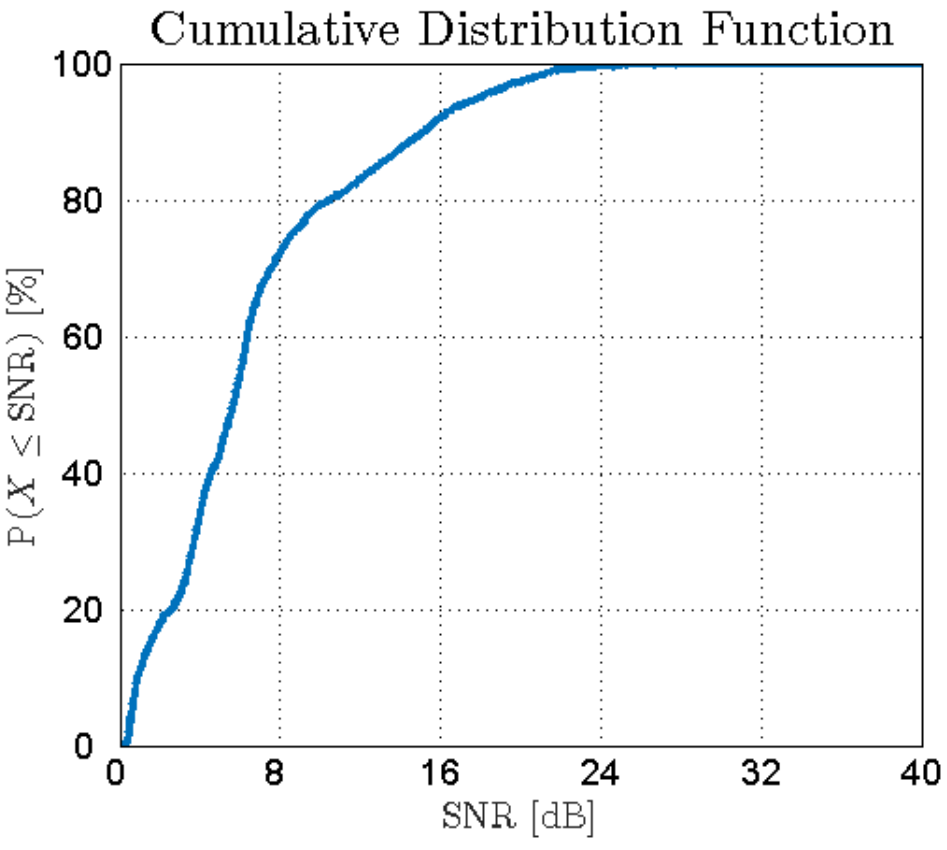

Figure 4.6: CDF of 2415 speech files with f_s = 32kHz

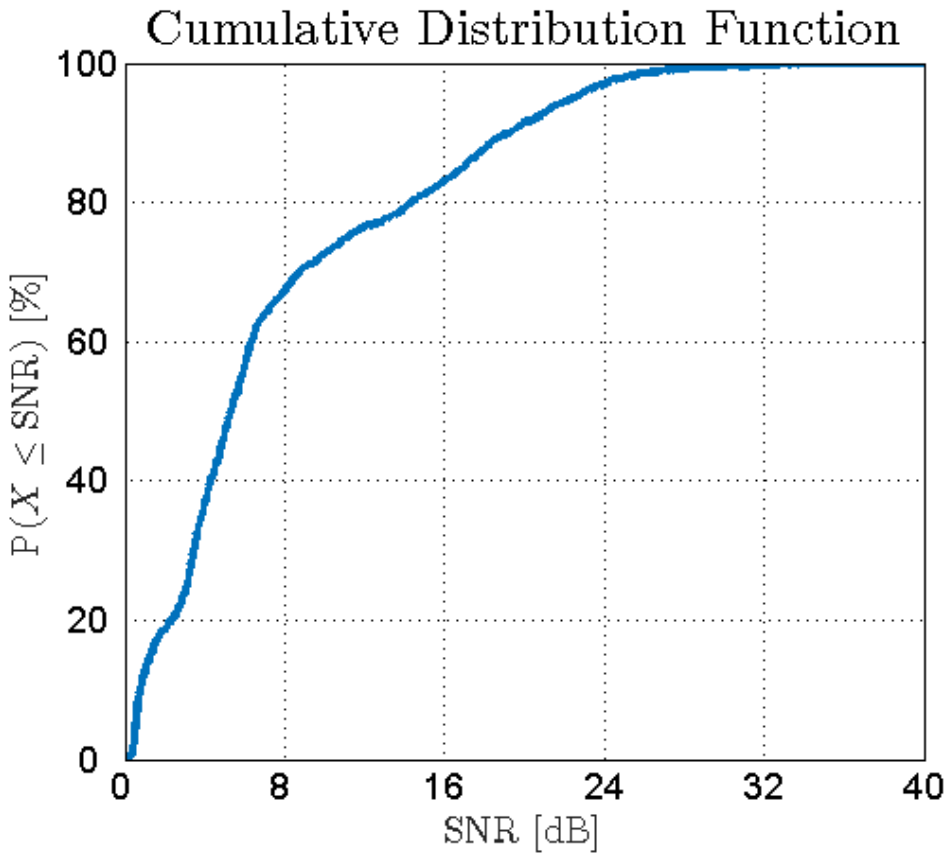

Figure 4.7: CDF of 2152 speech files with $f_s = 48kHz$

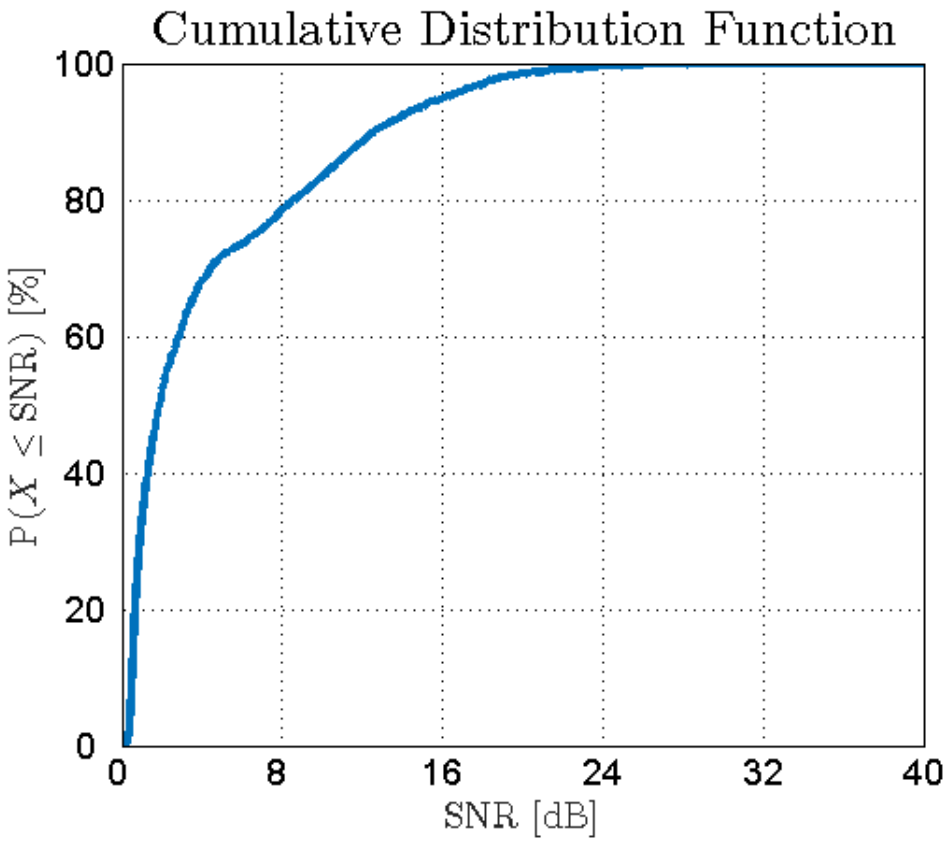

Figure 4.8: CDF of 2311 speech files with $N_c = 3$

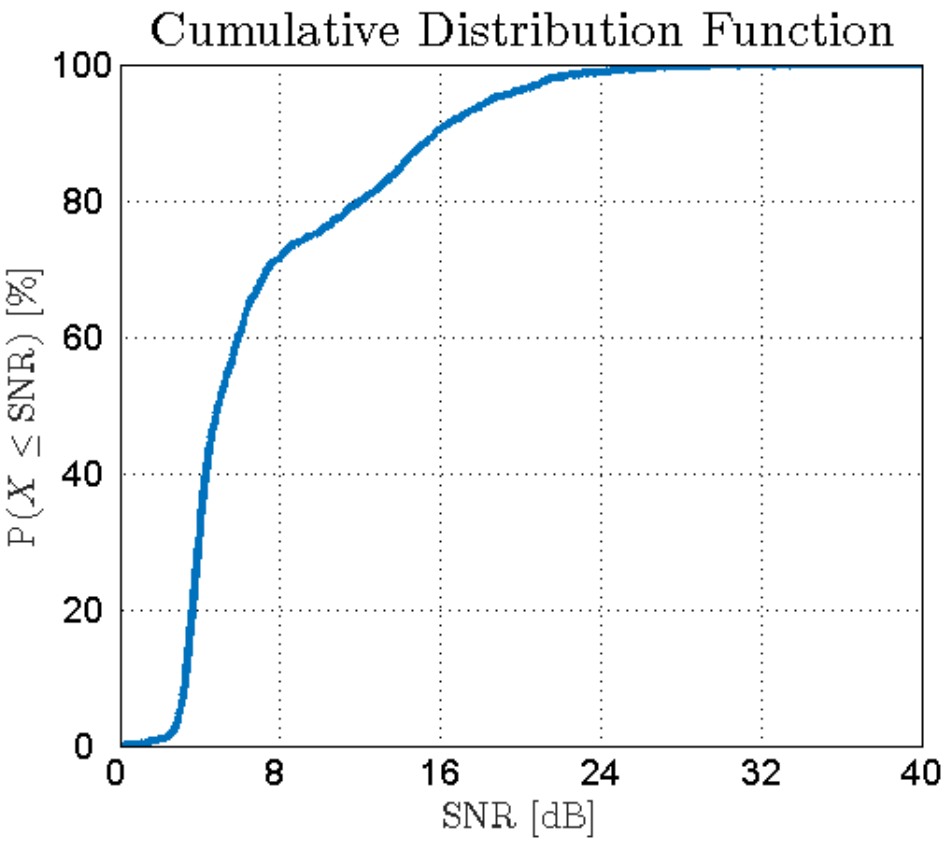

Figure 4.9: CDF of 2311 speech files with $N_c = 5$

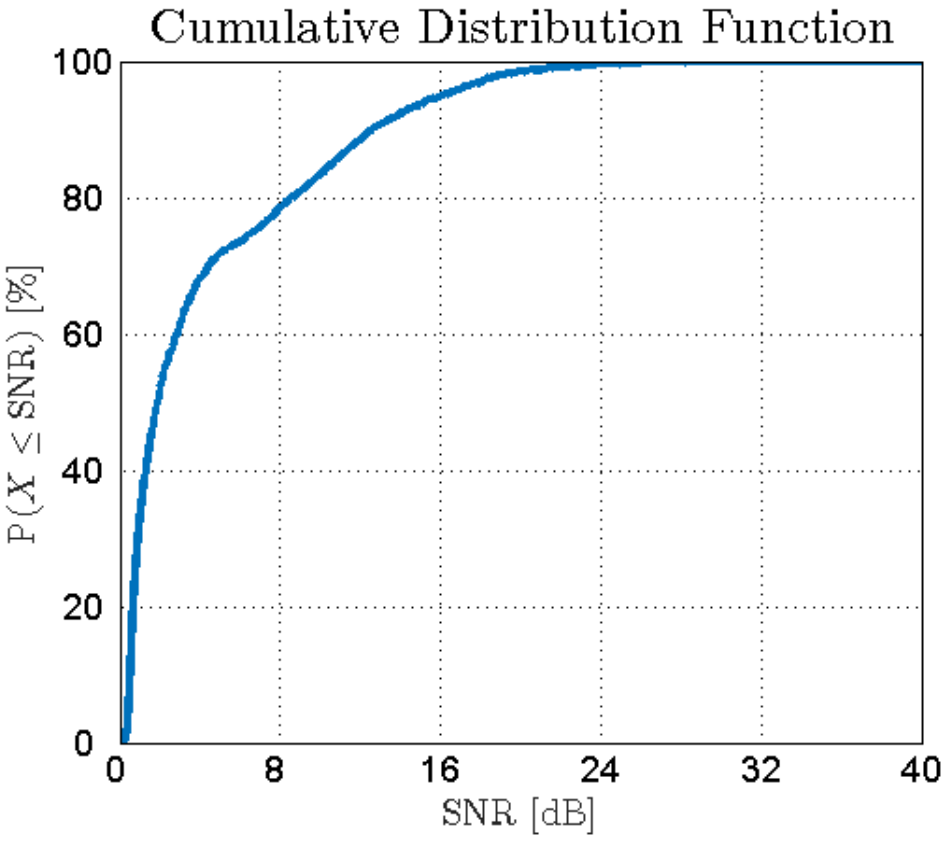

Figure 4.10: CDF of 2311 speech files with $N_c = 7$

Furthermore, the SNR values of four representative speech files were studied in detail with the help of 3D plots. For every chosen speech file the SNR was plotted over iterations and frames for a fixed frequency and in a second version over time and frequency for the last iteration. For speech file 2 the SNR is also calculated without using the algorithm and plotted over time and frequency (see Fig.4.15).

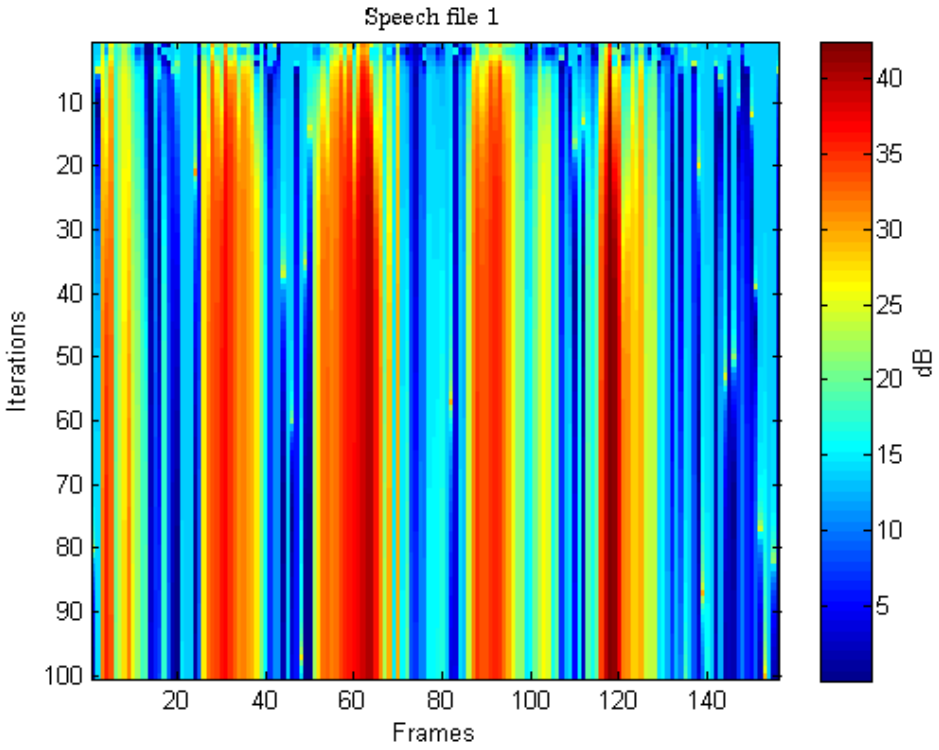

Figure 4.11: SNR of speech file 1 in dB; improvement with rising iterations can be noticed;T_{60} = 500 ms, f_s = 16 kHz, N_c = 5, d = 4 cm;

Filenames of the used speech files:
speech file 1: 08_f_short_1_wireless_001_1_1__16000_500_5_04_22_18_1e-02; T_{60} = 500 ms, f_s = 16 kHz, N_c = 5, d = 4 cm;
speech file 2: 22_m_short_1_wireless_103_15_5__032000_400_3_04_12_09_1e-02; T_{60} = 400 ms, f_s = 32 kHz, N_c = 3, d = 4 cm;
speech file 3: 22_m_short_1_wireless_103_15_5__032000_500_7_08_14_02_1e-03; T_{60} = 500 ms, f_s = 32 kHz, N_c = 7, d = 8 cm;
speech file 4: 08_f_short_1_wireless_056_11_2__048000_400_5_08_03_20_1e-03; T_{60} = 400 ms, f_s = 48 kHz, N_c = 5, d = 8 cm;
SNR values up to 70 dB (Speech File 4) can be achieved after 100 iterations for a speech file with 30 dB SNR. This means that after applying the algorithm to the speech file an improvement of the SNR of about 40 dB can be reached. By looking at the last iteration the harmonic structure of the speech files is clearly noticeable, therefore the algorithm is working in the right way. E.g. for Speech File 3 harmonics with a SNR between 60-70 dB are possible. To sum it up by using the stochastic approximation algorithm the SNR of speech files can be highly improved.

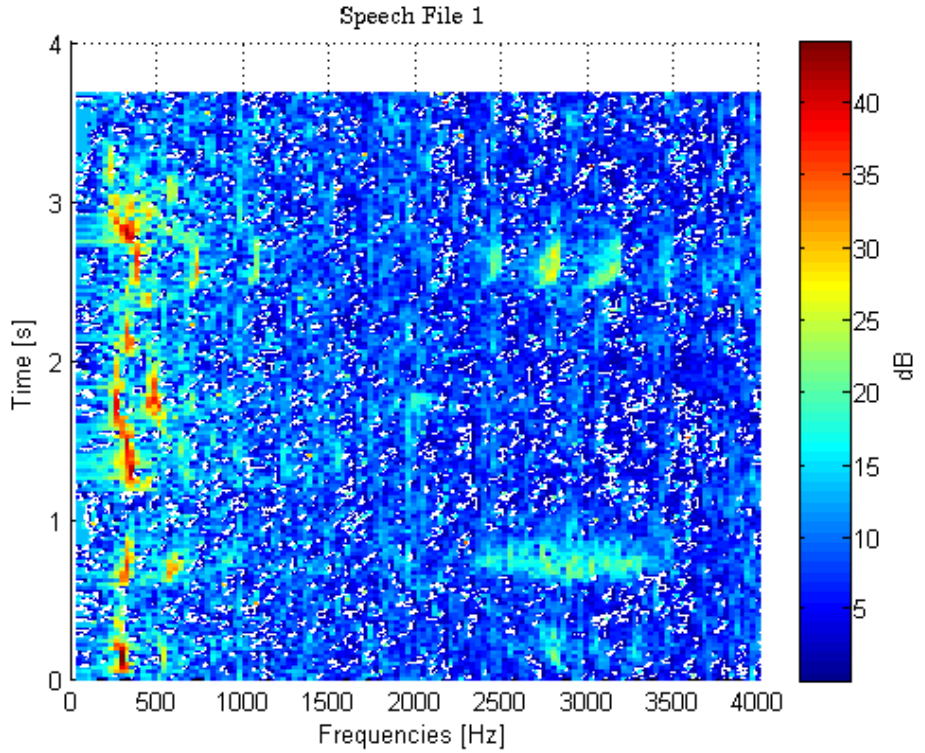

Figure 4.12: SNR of speech file 1 in dB for the last iteration plotted over time and frequency;T_{60} = 500 ms, f_s = 16 kHz, N_c = 5, d = 4 cm;

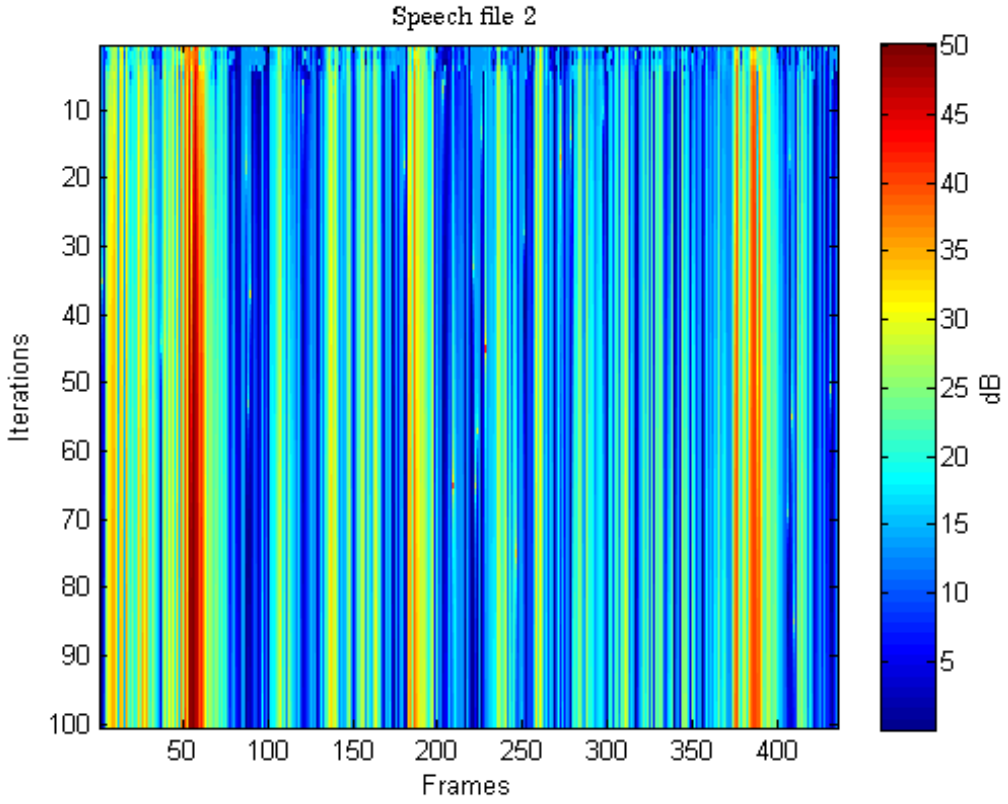

Figure 4.13: SNR of speech file 2 in dB; improvement with rising iterations can be noticed; T_{60} = 400 ms, f_s = 32 kHz, N_c = 3, d = 4 cm;

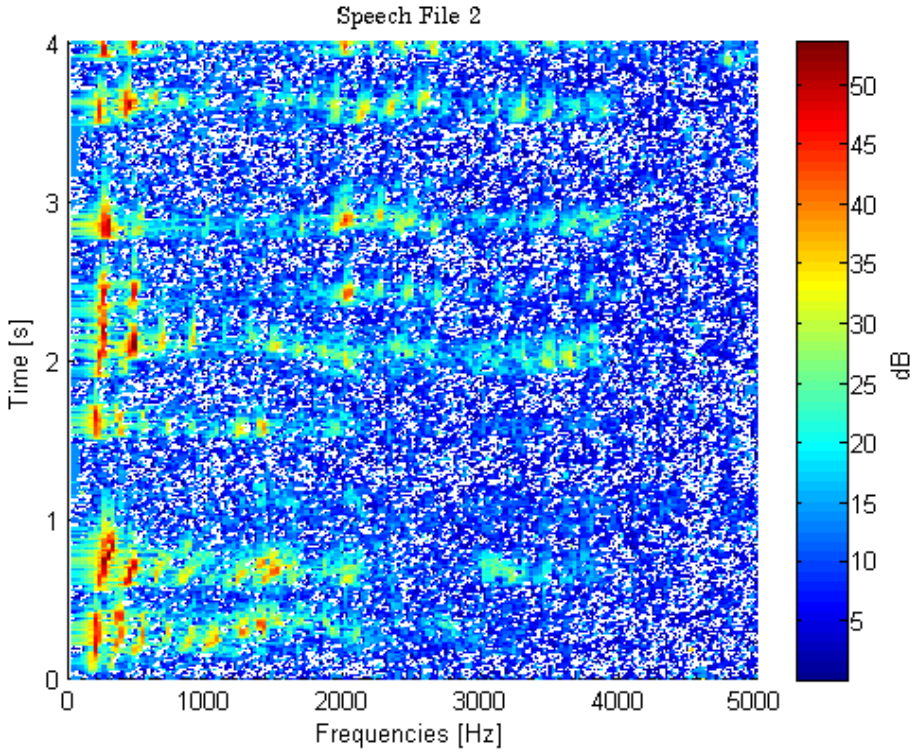

Figure 4.14: SNR of speech file 2 in dB for the last iteration plotted over time and frequency;T_{60} = 400 ms, f_s = 32 kHz, N_c = 3, d = 4 cm;

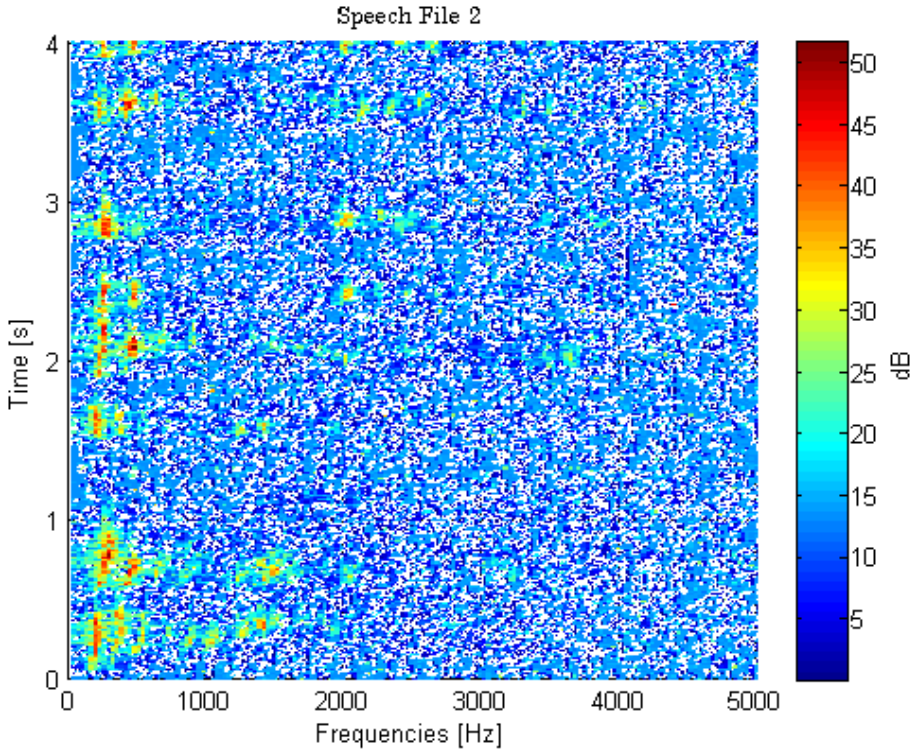

Figure 4.15: SNR of speech file 2 in dB without using the algorithm plotted over time and frequency;T_{60} = 400 ms, f_s = 32 kHz, N_c = 3, d = 4 cm;

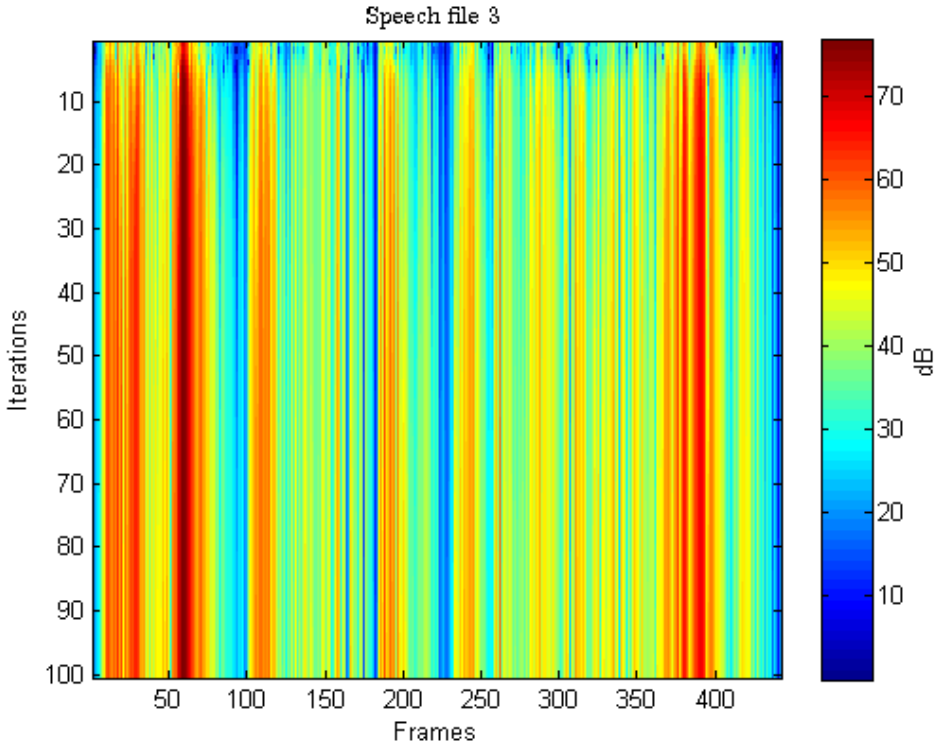

Figure 4.16: SNR of speech file 3 in dB; improvement with rising iterations can be noticed; T_{60} = 500 ms, f_s = 32 kHz, N_c = 7, d = 8 cm;

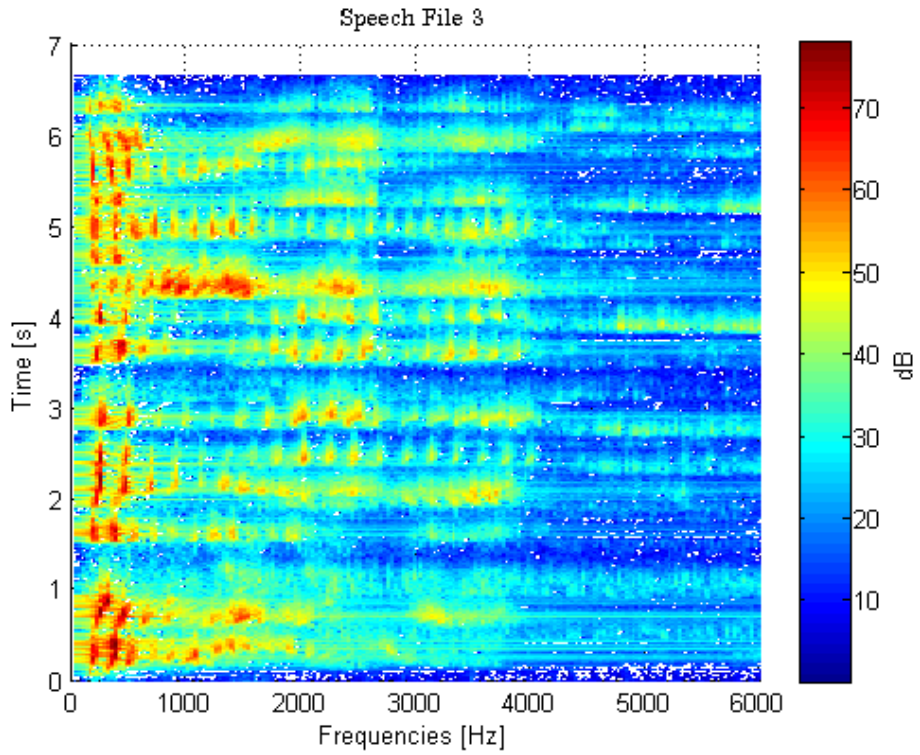

Figure 4.17: SNR of speech file 3 in dB for the last iteration plotted over time and frequency; T_{60} = 500 ms, f_s = 32 kHz, N_c = 7, d = 8 cm;

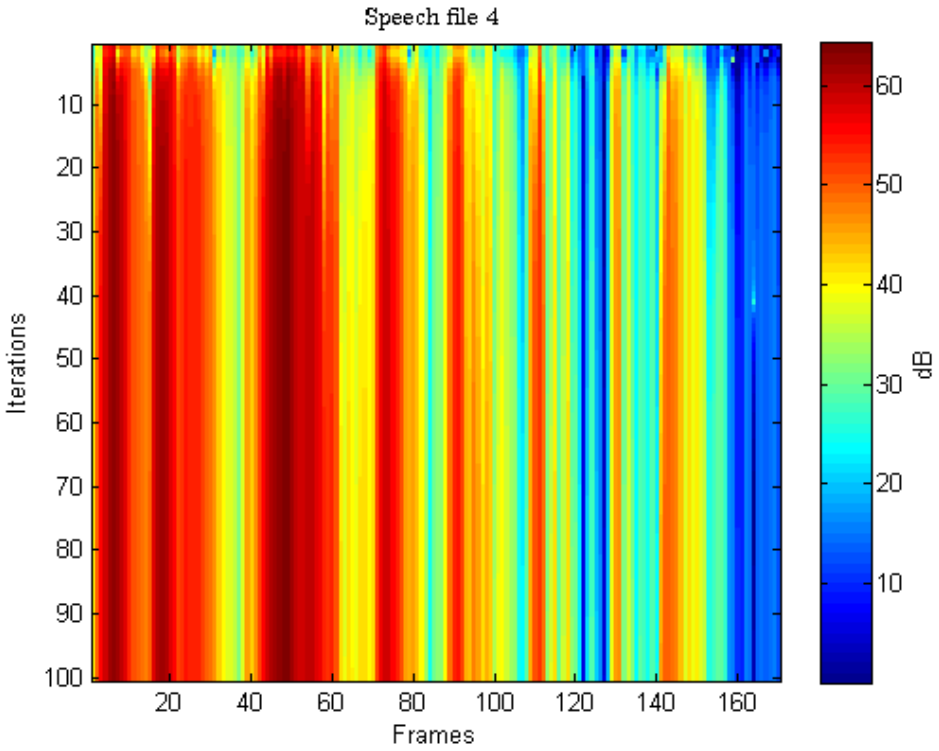

Figure 4.18: SNR of speech file 4 in dB; improvement with rising iterations can be noticed;T_{60} = 400 ms, f_s = 48 kHz, N_c = 5, d = 8 cm;

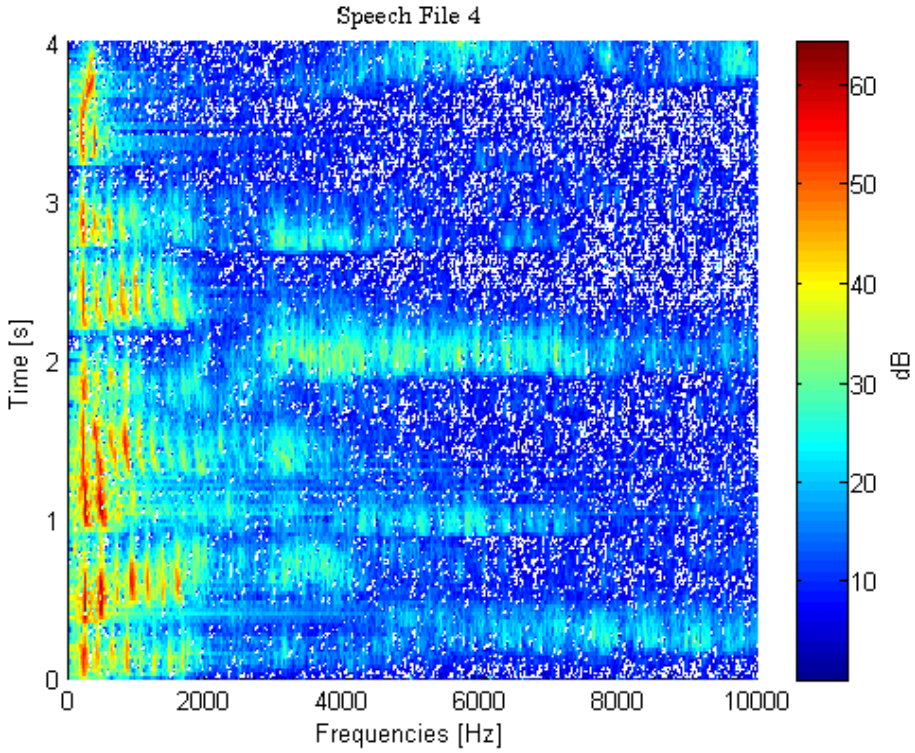

Figure 4.19: SNR of speech file 4 in dB for the last iteration plotted over time and frequency;T_{60} = 400 ms, f_s = 48 kHz, N_c = 5, d = 8 cm;

5 Conclusion

In a first step in this "Master Project" an adaptation algorithm for improving speech quality of speech files was derived. The presented algorithm is based on a gradient ascent and two versions of the algorithm were presented in the theoretical part of the project: the deterministic gradient ascent and the stochastic gradient ascent.
The practical part of this work shows the use of the presented algorithm. The algorithm is applied to synthetic speech data to prove the right implementation and to real speech data to show results. By looking at the results one can see that the algorithm achieves a high improvement in the signal to noise ratio of the speech files under test. For the practical part only the stochastic gradient ascent was used, because the deterministic gradient ascent did not work well, although its filter coefficients perfectly converged.

6 Bibliography

[1] E. Warsitz, A. Krueger and R. Haeb-Umbach, "Speech enhancement with a new generalized eigenvector blocking matrix for application in a generalized sidelobe canceller," ICASSP 2008, pp. 73-76

[2] E.Warsitz and R.Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Trans. on Audio, Speech and Lang. Proc.*, vol 15, no. 5, pp. 1529-1539, July 2007

[3] http://de.wikipedia.org/wiki/Rayleigh-Quotient, 4 Feb 2015, 12:30

[4] Eberhard Hänsler and Gerhard Schmidt, "13.4.1 Short-Term Power Estimation," in *Acoustic Echo and Noise Control: A Practical Approach*,

[5] Fast simulation of acoustic room impulse responses (image-source method) by Eric A. Lehmann, 28 Nov 2009 (Updated 10 Mar 2012), http://www.mathworks.com/matlabcentral/fileexchange/25965-fast-simulation-of-acoustic-room-impulse-responses–image-source-method-

[6] M. Schmidt, N. Le Roux, " Fast Convergence of Stochastic Gradient Descent under a Strong Growth Condition," 16 May 2012, http://nicolas.le-roux.name/publications/Schmidt12_fastcondition.pdf