Friedrich–Alexander–Universität Erlangen–Nürnberg
Lehrstuhl für Multimediakommunikation und Signalverarbeitung

Studienarbeit

Blind Source Separation combined with Noise Reduction

Bearbeiter:	Yuan Liu	**Beginn:**	11.11.2002
Betreuer:	Dipl.–Ing.(FH) Robert Aichner	**Abgabe:**	12.08.2003
	Prof. Dr.–Ing. Walter Kellermann		

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 12.August 2003

Yuan Liu
Äuβere Brucker Str.43 App.015
91052 Erlangen

List of Symbols

$B_{min}(m,\mu)$	the notation for the computation the minimum power estimates
BSS	Blind Source Separation
D	Number of successive short term psd estimates
DFT	Discrete Fourier Transform
$E(.)$	Expectation value
$E_L(\hat{s}_i(t))$	Energy of the separated BSS channel
FFT	Fast Fourier Transform
H(D)	Parameters for the approximation of the mean of the minimum
$H(t)$	Mixing system
$\mathcal{J}(w)$	Cost function
L	FFT length
M	Number of blocks
M(D)	Function for the approximation of the mean of the minimum
MSE	Mean Square Error
N	Signal length
NR	Noise Reduction
P	Mixing filter length
$P(m,\mu)$	Smoothed signal power spectral density
$P_n(m,\mu)$	Estimated noise power spectral density
Q	Unmixing filter length
$Q(m,\mu)$	Subtraction factor
$Q_{eq}(m,\mu)$	Normalized variance
$\hat{R}_x(t)$	Estimated cross-correlation matrix of the sensor signals
SIR	Signal-to-Interference Ratio
SNR	Signal-to-Noise Ratio
$\mathrm{SNR}_x(m,\mu)$	Estimate signal SNR at the sensor signals
SSIR	Segmental SIR
SSNR	Segmental SNR
Tr(.)	Trace of the matrix
U	Number of subwindows
$U(.)$	Unit step function
V	Number of samples in one subwindow
$\mathbf{W}(t)$	Unmixing system in the time-domain
$\underline{\mathbf{W}}(\mu)$	Unmixing system in the frequency-domain
$X(m,\mu)$	Input signal power spectral density
$Y(m,\mu)$	Output signal power spectral density

a	Constant for the $osub\_decaying(\mu)$ computation
b	Constant for the $osub\_decaying(\mu)$ computation
f_s	Sampling frequency
m	Block index in time domain
$m(t, \mu)$	Stepsize normalization factor
$n(t)$	Noise signal
$osub$	Oversubtraction factor
$osub\_decaying(\mu)$	Hyperbolically decaying
psd	Power Spectral Density
$r(t)$	Impulse response of the noise reduction system
$s(t)$	Source signal
sub	Spectral floor constant
$\hat{s}(t)$	Estimated source signal
t	Time index
$x(t)$	Input signal
$y(t)$	Output signal
z_a	Smoothing constant for the noise estimate
z_g	Smoothing constant for the signal estimate
$\Gamma(.)$	Gamma function
$\hat{\Lambda}_s(t)$	Diagonal correlation matrix of the sources $s(t)$
$\alpha(m, \mu)$	Smoothing parameter for the estimated signal psd
β	Threshold for $\lambda(t)$
γ	Forgetting factor
ϵ	Constant for $\lambda(t)$ computation
ζ	”Sharpness” of speech/non speech interval delimitation
η	Forgetting factor for $osub$
$\lambda(t)$	Two-channel energy ratio factor
μ	Frequency index
ν	Learning rate
ξ	Energy change
$\sigma^2(m, \mu)$	Variance
τ	Time index

Abstract

In this document we discuss the combination of the on-line blind source separation (BSS) for non-stationary signals with noise reduction (NR) methods with the purpose of improving the quality of signals. In our experiments we use noisy speech signals obtained by two microphones. The noise background consists of traffic noise which has a diffuse spatial characteristic.

Blind source separation is an ongoing research topic. Here we use the algorithm of Lucas Parra and Clay Spence [1] which generates decorrelated signals by diagonalizing second order statistics at multiple time points. As typical sources are often moving, and the multi-path channel is not static, we use here an on-line gradient algorithm with adaptive step size in the frequency domain based on second derivatives.

Spectral subtraction is a well known noise reduction technique of speech enhancement which has by now developed numerous facets. Here we have compared two methods. One is spectral subtraction based on minimum statistics proposed by Rainer Martin [2,3], another is speech enhancement combining blind source separation and two-channel energy based speaker detection which was presented by Erik Visser and Te-Won Lee [4]. In this work we examine in which order both methods should be combined, i.e. (BSS→NR) or (NR→BSS)?

In our experiments we observed that the two methods are almost the same with (BSS→NR) being slightly better (1dB) than (NR→BSS). Furthermore, for the noise reduction we should use the minimum statistics based algorithm [2,3]. By this combination SNR-improvement is approximately 3-5dB and SIR-improvement is about 5-8dB.

Contents

Chapter 1

Introduction

A growing member of researchers have published in recent years on the problem of blind source separation which are data analysis problems that have received considerable attention in the machine learning community during the last few years. Here we focus on speech signals that can be considered as non-stationary. The problem seems of relevance in various application areas such as speech enhancement with multiple microphones, or crosstalk removal in multichannel communication. The goal of blind source separation is to filter the signals from a microphone array to extract the original sources while reducing interfering signals. Due to the spatial variability of a room impulse response, different microphones receive different convolved versions of each source.

In terms of separation criteria, to our knowledge, there are three types of algorithms for blind source separation of a convolutive mixture of broad-band signals. Algorithms that simultaneously diagonalize second order statistics at multiple time lags, i.e. exploiting nonwhiteness, algorithms that simultaneously diagonalize second order statistics at multiple times exploiting non-stationarity, and algorithms that identify statistically independent signals by considering higher order statistics. In this document we simplify the problem to two channels and two speech signals which are statistically independent.

As Fig. 1.1 shows, for speech enhancements in real environments such as car environment noise reduction is necessary for improving the speech signal quality. Thus, we must think about the combination of the two algorithms, blind source separation and noise reduction.

Spectral subtraction is a well-known noise reduction technique which performs the task very well for non-stationary signals. Frequency-domain speech enhancements systems typically consist of a spectral analysis/synthesis system, a spectral gain computation method, and a background noise power spectral density (psd) estimation algorithm. In this document, we use a noise reduction algorithm which is based on minimum statistics. The psd smoothing algorithm utilizes a first order recursive system with a time- and frequency-dependent smoothing parameter. The smoothing parameter is optimized for tracking nonstationary signals by min-

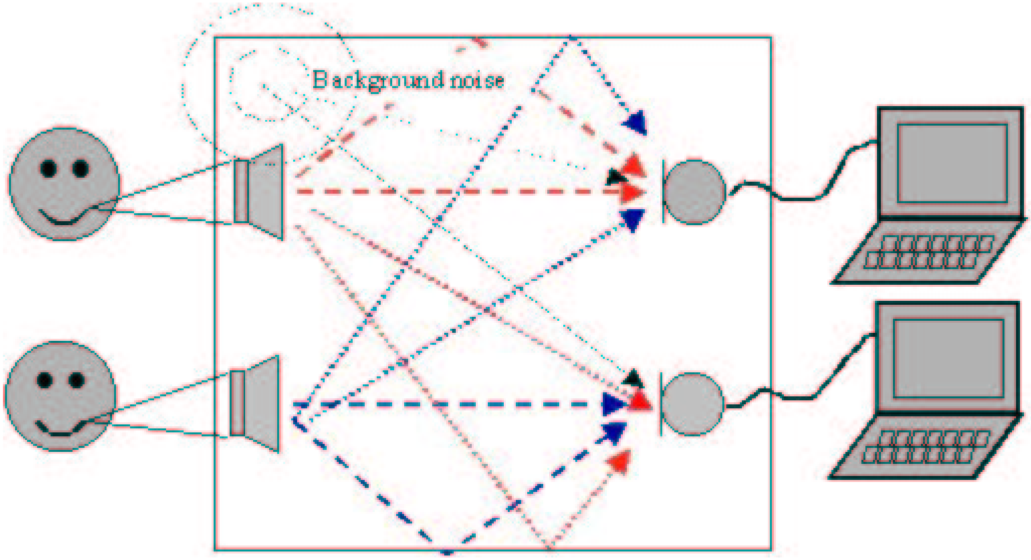

Figure 1.1: Speech in real environments

imizing a conditional mean square error criterion. In contrast to other methods the minimum statistics algorithm does not use any explicit threshold to distinguish between speech activity and speech pause and is therefore more closely related to soft-decision methods than to the traditional voice activity detection methods. Similar to soft-decision methods it can also update the estimated noise psd during speech activity. We here use the minimum of the subband noise power within a finite window to estimate the noise floor. The algorithm is based on the observation that a short-time subband power estimate of a noisy speech signal exhibits distinct peaks and valleys . While the peaks correspond to speech activity the valleys of the smoothed noise estimate can be used to obtain an estimate of subband noise power. To obtain reliable noise power estimates the data window for the minimum search must be large enough to bridge any peak of speech activity.

This document is organized as follows. In chapter 2 we derive Lucas Parra's and Clay Spence's on-line gradient algorithm for blind source separation. In chapter 3 the algorithm for the spectral subtraction with two different methods of noise psd estimation is explained. The combination of the two algorithms is investigated in chapter 4. In chapter 5 the experimental results are presented. At the end of this document we will present some conclusions.

Chapter 2

On-line Blind Source Separation

Recently a number of multiple decorrelation algorithms have been presented that claim good performance for non-stationary signals. Some implicitly exploit non-stationary, while others reduce the number of required constraints by using low dimensional parameterizations of the filters. Here we use the on-line algorithm proposed by Lucas Parra and Clay Spence which exploits multiple decorrelation and visits the signals only once [1].

2.1 On-line Time-domain Decorrelation

Assume non-stationary independent source signals $\mathbf{s}(t) = [s_1(t), ..., s_{d_s}(t)]^T$ where t denotes the time index. These signals are observed in a multi-path environment $\mathbf{H}(\tau)$ of order P as $\mathbf{x}(t) = [x_1(t), ..., x_{d_x}(t)]^T$,

$$\mathbf{x}(t) = \sum_{\tau=0}^{P} \mathbf{H}(\tau)\mathbf{s}(t-\tau) + \mathbf{n}(t). \tag{2.1}$$

Figure 2.1 shows the two-input, two-output convolutive blind source separation diagram which is the easiest instance $d_s = 2; d_x = 2$.

With the purpose of finding an estimate $\hat{\mathbf{s}}(t)$ for the unknown sources $\mathbf{s}(t)$ we formulate an FIR inverse model $\mathbf{W}$ of suitably chosen length Q,

$$\hat{\mathbf{s}}(t) = \sum_{\tau=0}^{Q} \mathbf{W}(\tau)\mathbf{x}(t-\tau). \tag{2.2}$$

The task of convolutive source separation is to find filters $\mathbf{W}$ that separate the mixed signals but not necessarily deconvolve them. By optimizing filter coefficients $\mathbf{W}$ the estimated sources $\hat{\mathbf{s}}(t)$ are equal to the unknown sources $\mathbf{s}(t)$, if $\mathbf{n}(t)$ is equal to 0, and thus the correlation matrix is diagonal,

$$\forall t, \tau : E[\hat{\mathbf{s}}(t)\hat{\mathbf{s}}^H(t-\tau)] = \hat{\mathbf{\Lambda}}_s(t,\tau). \tag{2.3}$$

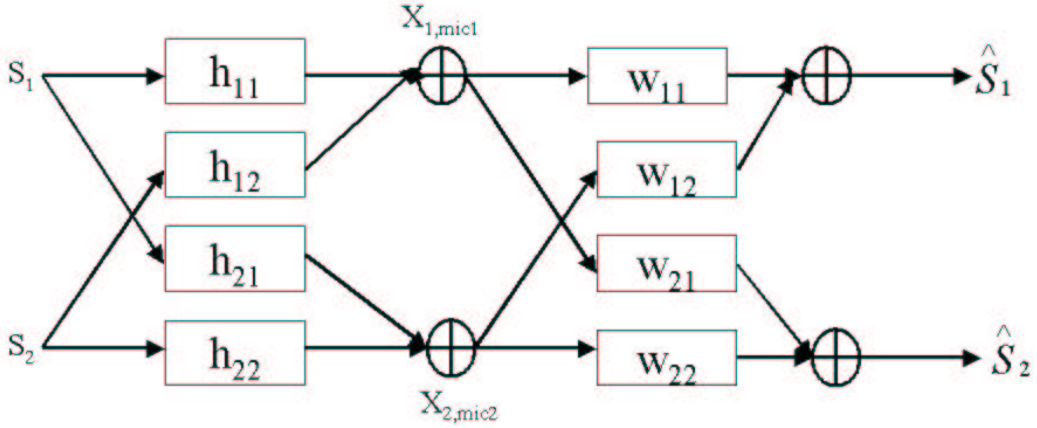

Figure 2.1: BSS system configuration

Here $\hat{\mathbf{\Lambda}}_s(t,\tau) = diag([\hat{\lambda}_1(t,\tau), ..., \hat{\lambda}_{d_s}(t,\tau)])$ represents the autocorrelations of the source signals at times t which can be estimated from the data. For the expectation we will use the time average starting at time t, i.e. $E[f(t)] = \sum_{\tau'} f(t+\tau')$. We can then define a separation criterion for simultaneous diagonalization with,

$$\begin{aligned} \mathcal{J}(\mathbf{W}) &= \sum_t \mathcal{J}(t,\mathbf{W}) = \sum_t \|\mathbf{J}(t,\mathbf{W})\|^2 \\ &= \sum_{t,\tau} \|E[\hat{\mathbf{s}}(t)\hat{\mathbf{s}}^H(t-\tau)] - \hat{\mathbf{\Lambda}}_s(t,\tau)\|^2 \\ &= \sum_{t,\tau} \|\sum_{\tau'} \hat{\mathbf{s}}(t+\tau')\hat{\mathbf{s}}^H(t-\tau+\tau') - \hat{\mathbf{\Lambda}}_s(t,\tau)\|^2, \end{aligned} \tag{2.4}$$

where $\|\mathbf{J}\|^2$ is the trace of the matrix $\mathbf{J}\mathbf{J}^H$, $\|\mathbf{J}\|^2 = \mathrm{Tr}(\mathbf{J}\mathbf{J}^H)$. It is straightforward to compute the stochastic gradient of this optimization criteria with respect to the filter parameters.

$$\begin{aligned} \frac{\partial \mathcal{J}(t,\mathbf{W})}{\partial \mathbf{W}(l)} &= \sum_{\tau} \left(\sum_{\tau'} \hat{\mathbf{s}}(t+\tau^{'})\hat{\mathbf{s}}^{\mathbf{H}}(t+\tau^{'}-\tau) - \hat{\Lambda}_s(t,\tau) \right) \\ &\quad \times \sum_{\tau''} \hat{\mathbf{s}}(t+\tau^{''}-\tau)\hat{\mathbf{x}}^{\mathbf{H}}(t+\tau^{''}-l) \\ &+ \sum_{\tau} \left(\sum_{\tau'} \hat{\mathbf{s}}(t+\tau^{'})\hat{\mathbf{s}}^{\mathbf{H}}(t+\tau^{'}-\tau) - \hat{\Lambda}_s(t,\tau) \right) \\ &\quad \times \sum_{\tau''} \hat{\mathbf{s}}(t+\tau^{''}-\tau)\hat{\mathbf{x}}^{\mathbf{H}}(t+\tau^{''}-\tau-l). \end{aligned} \tag{2.5}$$

To simplify this expression, we show that the first and second sum over τ can be made equal. In the gradient descent procedure we may choose to apply the different gradient terms in these sums at times other than the time t. So we can replace t with $t-\tau$ in the second sum, which effectively uses the value of the sum in the gradient update at time $t^{'} = t-\tau$. In addition, if the sum over τ

runs symmetrically over positive and negative values, we can change the sign of τ in the second sum. It can be argued that the diagonal matrix $\hat{\Lambda}_s(t,\tau)$ remains unchanged by these transformations, at least in a quasistationary approximation. The resulting update at time t for lag l with a step size of ν simplifies to

$$\begin{aligned} \mathbf{\Delta}_t\mathbf{W}(l) &= -2\nu{\textstyle\sum_{\tau}}\Big({\textstyle\sum_{\tau'}}\hat{\mathbf{s}}(t+\tau')\hat{\mathbf{s}}^H(t-\tau+\tau') - \hat{\mathbf{\Lambda}}_s(t,\tau))\Big) \\ &\times \ {\textstyle\sum_{\tau''}}\hat{\mathbf{s}}(t+\tau''-\tau')\mathbf{x}^H(t+\tau''-l). \end{aligned} \tag{2.6}$$

with the estimated cross-correlation of the sensor signal:

$$\hat{\mathbf{R}}_x(t,\tau) = E[\hat{\mathbf{x}}(t) * \hat{\mathbf{x}}^H(t-\tau)]. \tag{2.7}$$

By inserting (2.2) into (2.6) and using (2.7) we obtain for the update at time t,

$$\mathbf{\Delta}_t\mathbf{W} = -2\nu\mathbf{J}(t) * \mathbf{W} * \hat{\mathbf{R}}_x(t), \tag{2.8}$$

with $\mathbf{J}(t) = \mathbf{W} * \hat{\mathbf{R}}_x(t) \bullet \mathbf{W}^H - \hat{\mathbf{\Lambda}}_s(t,\tau)$. In this short hand notation convolutions are represented by $*$, and correlations by $\bullet$, and time lag indices are omitted.

In our experiments it is more assumed that estimated cross-correlations don't change much within the time scale corresponding to one filter length, i.e. $\hat{\mathbf{R}}_x(t,\tau) \approx \hat{\mathbf{R}}_x(t+l,\tau)$ for $0 < l \leq Q$.

2.2 Frequency-Domain Gradient Algorithm

In this section we will present an on-line frequency domain implementation of this basic gradient algorithm with the purpose of reducing the computational cost as well as improving the convergence properties of the gradient updates.

First, we transform this gradient expression into the frequency domain with L frequency bins, because the convolutions in the equation (2.8) are too expensive to compute.

$$\Delta_t\underline{\mathbf{W}}(\mu) = 2\nu\mathbf{J}(\mu)\underline{\mathbf{W}}(\mu)\underline{\hat{\mathbf{R}}}_x(t,\mu) \tag{2.9}$$

with $\mathbf{J}(t,\mu) = \underline{\mathbf{W}}(\mu)\underline{\hat{\mathbf{R}}}_x(t,\mu)\underline{\mathbf{W}}^H(\mu) - \hat{\mathbf{\Lambda}}_s(t,\mu)$ where μ is the frequency bin. Here $\underline{\mathbf{W}}(\mu)$ and $\underline{\hat{\mathbf{R}}}_x(t,\mu)$ are the L point discrete Fourier transform of $\mathbf{W}(\tau)$ and $\hat{\mathbf{R}}_x(t,\tau)$ respectively. It is easy to see that the same expression can also be obtained directly in the frequency domain by using the gradient of $\sum_{t,\tau}\mathcal{J}(t,\mu) = \sum_{t,\tau}\|\mathbf{J}(t,\mu)\|^2$ with respect to $\underline{\mathbf{W}}^*$. In this form the update rule for the complex parameter w with learning rate ν is $\Delta w = -2\nu\frac{\partial}{\partial w*}$.

Secondly, in order to improve the convergence properties of this algorithm it is necessary to consider some second order gradient expressions. Instead of following the original gradient we adapt the step size with a normalization factor $m(t,\mu)$ for different frequencies. This is a power normalization.

$$\Delta_t\underline{\mathbf{W}}(\mu) = -\nu m^{-1}(t,\mu)\frac{\partial\mathcal{J}(t,\mu)}{\partial\underline{\mathbf{W}}^*(\mu)}. \tag{2.10}$$

As a motivation for this step size normalization consider the following. For a real valued square cost, $J(z) = azz^*$ in the complex plane the proper second order gradient step corresponds to $(\partial^2 J(z)/\partial z \partial z^*)^{-1} \partial J(z)/\partial z^* = z$. The corresponding expression in our current cost function can be computed to $\frac{\partial^2 \mathcal{J}}{\partial^2 \underline{\mathbf{W}}^*_{ij} \partial^2 \underline{\mathbf{W}}_{ij}} = 2((\underline{\mathbf{W}}\mathbf{R_X})^{\mathbf{H}}(\underline{\mathbf{W}}\mathbf{R_X}))_{\mathbf{jj}}$. It is real valued and independent of i. In order to have the same step size for all j we sum over j, and use

$$m(t,\mu) = \sum_j \frac{\partial^2 \mathcal{J}}{\partial^2 \underline{\mathbf{W}}^*_{ij} \partial^2 \underline{\mathbf{W}}_{ij}} = \|\underline{\mathbf{W}}(\mu)\hat{\underline{\mathbf{R}}}_x(t,\mu)\|^2, \tag{2.11}$$

which is effectively an adaptive power normalization.

Finally, we implement the multiple adaptive decorrelation as a block processing procedure. The signals are windowed and transformed into the frequency domain. The segment $x_i(t), ..., x_i(t+L-1)$ gives frequency components $x_i(t,\mu)$ for $\mu = 0, ..., L-1$, which are used to compute the estimate cross-correlations directly in the frequency domain. For the cross-correlations of the observations in the frequency domain this reads

$$\hat{\underline{\mathbf{R}}}_x(t,\mu) = (1-\gamma)\hat{\underline{\mathbf{R}}}_x(t,\mu) + \gamma \mathbf{X}(t,\mu)\mathbf{X}^H(t,\mu). \tag{2.12}$$

where γ is the forgetting factor for the on-line blind source separation algorithm which is chosen according to the stationary time of the signal.

Chapter 3

Noise Reduction

In this chapter we present Rainer Martin's algorithm for the enhancement of noisy speech signals by means of spectral subtraction [2,3]. Furthermore, to estimate noise power we introduce another method which estimates noise psd in combination with a blind source separation algorithm [4].

3.1 Description of the Algorithm

A block diagram of the basic spectral subtraction method is shown in Fig. 3.1 where $P(m, \mu)$ is the smoothed signal psd, $P_n(m, \mu)$ is the estimated noise psd and $Q(m, \mu)$ is the subtraction factor. The algorithm appropriately modifies the short time spectral magnitude of the disturbed speech signal such that the synthesized signal is perceptually as close as possible to the undisturbed speech signal. The optimal weighting of spectral magnitudes is computed using a noise psd estimate and a subtraction rule.

At the heart of a spectral subtraction algorithm are a noise psd estimator and a subtraction rule which translates the subband SNR into a spectral weighting factor, such that subbands with low SNR are attenuated and subbands with high SNR are not modified. Both the noise psd estimator and the subtraction rule have significant impact on the audible residual noise. The basic spectral subtraction algorithm requires only one microphone.

We assume that the bandlimited and sampled disturbed signal $x(t)$ is a sum of a zero mean speech signal $s(t)$ and a zero mean noise signal $n(t)$, $x(t) = s(t) + n(t)$.In addition, we further assume that $s(t)$ and $n(t)$ are statistically independent, hence $E\{x^2(t)\} = E\{s^2(t)\} + E\{n^2(t)\}$. Spectral processing is based on a DFT filter bank with the window length L. The phase of the disturbed signal is not modified. We denote the data window by $h(t)$ and the DFT of the windowed disturbed signal $x(t)$ by

$$X(m, \mu) = \sum_{k=0}^{L-1} x(mR - k)h(k)exp\left(-j\frac{2\pi\nu\mu}{L}\right) \tag{3.1}$$

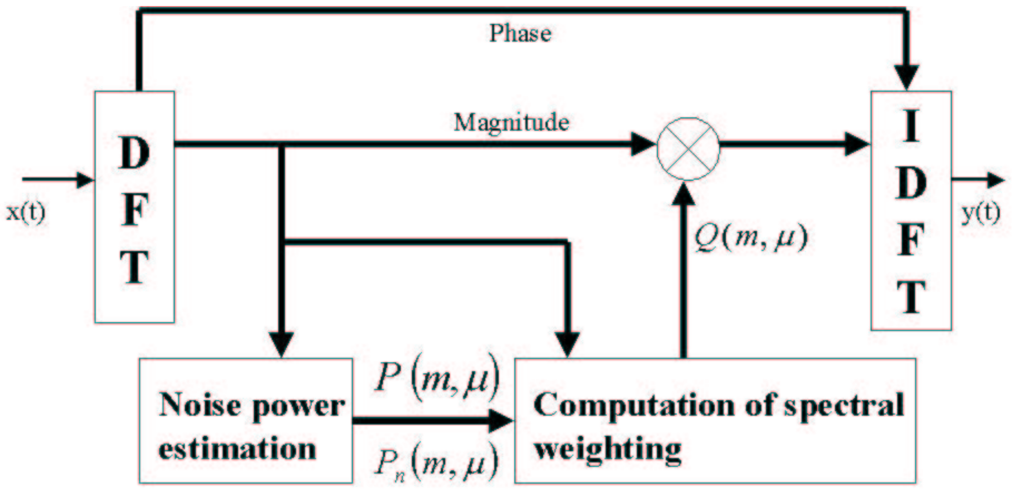

Figure 3.1: Block diagram of spectral processing

where m and μ refer to the decimated time index and the DFT frequency bins. Furthermore, to facilitate our notation and to avoid unnecessary normalization factors we assume $\sum_{k=0}^{L-1} h^2(k) = 1$. The improved subband signals are converted back to the time domain using an inverse DFT. The synthesized improved speech signal is denoted by $y(t)$, the corresponding spectral magnitude by $|Y(m,\mu)|$.

3.2 Noise Power Spectral Density Estimation

3.2.1 Noise Power Spectral Density Estimation based on Optimal Smoothing and Minimum Statistics

First, we consider a method to estimate the psd of nonstationary noise when a noisy speech signal is given [3]. It tracks spectral minima in each frequency band without any distinction between speech activity and speech pause. By minimizing a conditional mean square estimation error criterion in each time step we derive the optimal smoothing parameter for recursive smoothing of the power spectral density of the noisy speech signal. Based on the optimally smoothed power spectral density estimate and the analysis of the statistics of spectral minima an unbiased noise estimator is developed. The estimator is well suited for real time implementations.

It is noted that for all practical purposes the real and imaginary part of a Fourier transform coefficient $X(m,\mu)$ can be considered to be independent and can be modeled as zero mean Gaussian random variables. So each periodogram bin $|X(m,\mu)|^2$ is an exponentially distributed random variable with probability density function (pdf)

$$f_{|X(m,\mu)|^2}(x) = \frac{U(x)}{\sigma_N^2(m,\mu) + \sigma_S^2(m,\mu)} e^{-x/(\sigma_N^2(m,\mu)+\sigma_S^2(m,\mu))} \tag{3.2}$$

where $\sigma_S^2(m,\mu) = E|S(m,\mu)|^2$ and $\sigma_N^2(m,\mu) = E|N(m,\mu)|^2$ are the power spectral densities of the speech and the noise signals, respectively. $U(x)$ denotes the unit step function, i.e. $U(x) = 1$ for $x \geq 0$ and $U(x) = 0$ otherwise. It is easy to see that during speech pause, $\sigma_S^2(m,\mu) \equiv 0$, the mean and the variance of $|X(m,\mu)|^2$ are equal to $\sigma_N^2(m,\mu)$ and $\sigma_N^4(m,\mu)$, respectively.

The minimum statistics noise tracking method is based on the observation that even during speech activity a short term power spectral density estimate of the noise signal frequently decays to values which are representative of the noise power level. The method rests on the fundamental assumption that during speech pause or within brief periods in between words and syllables the speech energy is close or identical to zero. Thus, by tracking the minimum power within a finite window large enough to bridge high power speech segments the noise floor can be estimated.

The main themes of this section are therefore first to find a time varying smoothing periodogram $P(m,\mu)$ and its variance are better balanced, then use the periodogram to develop an algorithm for bias compensation, and to speed up the noise tracking in general.

3.2.1.1 Optimal Time Varying Smoothing

To highlight some of the obstacles which are encountered when implementing such an approach we consider a recursively smoothed periodogram,

$$P(m,\mu) = \alpha P(m-1,\mu) + (1-\alpha)|X(m,\mu)|^2. \tag{3.3}$$

The smoothed signal psd estimated $P(m,\mu)$ from which the noise psd estimate $\hat{\sigma}_N^2(m,\mu)$ is derived has to satisfy conflicting requirements. On one hand the variance should be as small as possible requiring the smoothing parameter α in (3.3) to be close to one. On the other hand, the smoothed psd estimate has to track possibly nonstationary noise and also has to follow the highly nonstationary excursions of the speech signal. Especially when the input signal has a high dynamic range these requirements are impossible to satisfy with a constant smoothing parameter α. So, as we will see below, these problems can be circumvented with a time-varying and possibly frequency dependent smoothing parameter $\alpha(m,\mu)$

$$P(m,\mu) = \alpha(m,\mu)P(m-1,\mu) + (1-\alpha(m,\mu))|X(m,\mu)|^2. \tag{3.4}$$

Because we want $P(m,\mu)$ to be as close as possible to the real noise psd $\hat{\sigma}_N^2(m,\mu)$, our objective is to minimize the conditional mean square error(MSE)

$$E\{(P(m,\mu) - \hat{\sigma}_N^2(m,\mu))^2|P(m-1,\mu)\} \tag{3.5}$$

from one iteration step to the next.

After using $P(m,\mu)$ in (3.5) and using $E\{|X(m,\mu)|^2\} = \hat{\sigma}_N^2(m,\mu)$, $E\{|X(m,\mu)|^4\} = \hat{\sigma}_N^4(m,\mu)$ the mean square error is given by

$$\begin{aligned}&E\{(P(m,\mu) - \hat{\sigma}_N^2(m,\mu))^2 | P(m-1,\mu)\} = \\ &\alpha^2(m,\mu)(P(m,\mu) - \hat{\sigma}_N^2(m,\mu))^2 + \hat{\sigma}_N^4(m,\mu)(1-\alpha(m,\mu))^2.\end{aligned} \tag{3.6}$$

Setting the first derivative with respect to $\alpha(m,\mu)$ to zero yields

$$\alpha_{opt}(m,\mu) = \frac{1}{1 + (P(m-1,\mu)/\hat{\sigma}_N^2(m,\mu) - 1)^2}, \tag{3.7}$$

and the second derivative, being nonnegative, reveals that this is indeed a minimum.

In a practical implementation of the optimal smoothing parameter (3.7) we replace the real noise psd $\sigma_N^2(m,\mu)$ by its latest estimated value $\hat{\sigma}_N^2(m-1,\mu)$ and limit the smoothing parameter to a maximum value α_{max} to avoid dead lock for $P(m-1,\mu)/\hat{\sigma}_N^2(m,\mu) = 1$. Here in our experiments $\alpha_{max} = 0.96$.

In general, the time evolution of the estimated noise psd $\hat{\sigma}_N^2(m,\mu)$ lags behind the time evolution of the true noise psd. Problems may arise when the smoothing parameter is close to one since then the smoothed psd estimate $P(m,\mu)$ cannot react quickly to changes in the noise psd. Given this uncertainty in the noise psd estimate the tracking error in the smoothed short term psd $P(m,\mu)$ must be monitored. Here the monitoring algorithm is to compare the average short-term psd estimate of the previous frame $1/L\sum_{\mu=0}^{L-1}|P(m-1,\mu)|$ to the average periodogram $1/L\sum_{\mu=0}^{L-1}|X(m,\mu)|^2$ and thus detects deviations of the short-term psd estimate from the actual averaged peridogram. The comparison between the average smoothed psd estimate and the average actual periodogram is implemented by means of a "soft" $1/(1+x^2)$ characteristic

$$\tilde{\alpha}_c(m) = \frac{1}{1 + \left(\sum_{\mu=0}^{L-1}|P(m-1,\mu)| / \sum_{\mu=0}^{L-1}|X(m,\mu)|^2 - 1\right)^2}, \tag{3.8}$$

and the resulting correction factor is limited to values larger than 0.7 and smoothed over time

$$\alpha_c(m) = 0.7\alpha_c(m-1) + 0.3max(\tilde{\alpha}_c(m), 0.7). \tag{3.9}$$

The multiplication of the correction factor with the optimal smoothing parameter then yields the final smoothing parameter

$$\hat{\alpha}_c(m,\mu) = \frac{\alpha_{max}\alpha_c(m)}{1 + \left(\sum_{\mu=0}^{L-1}|P(m-1,\mu)| / \sum_{\mu=0}^{L-1}|X(m,\mu)|^2 - 1\right)^2}. \tag{3.10}$$

To improve the performance of the noise estimator in high levels of nonstationary noise it is advantageous to apply also a lower limit α_{min} with a maximum α_{min}

of 0.3, to $\hat{\alpha}_{opt}(m, \mu)$ and thus limit also the variance of the bias correction factor. This lower limit, however, might decrease the performance for high SNR speech. To avoid the attenuation of weak consonants at the end of a word we require that $P(m, \mu)$ can decay from its peak values to the noise level in about 64ms (or four frames at L=2R=256) for example. Then, α_{min} can be computed as

$$\alpha_{min} = min\left(0.3, SNR^{-\frac{R}{0.064 s f_s}}\right). \tag{3.11}$$

3.2.1.2 Statistics of Minimum Power Estimates and Unbiased Noise Estimator

The objective of this section is to derive the bias and the variance of the minimum estimator and the unbiased noise estimator.

We consider the minimum $P_{min}(m, \mu)$ of D successive short term psd estimates $P(m, \mu)$, $m \in \{m_1, ..., m_1 - i, ..., m_1 - D + 1\}$ For an infinite sequence of periodograms $|X(m, \mu)|^2$ the short term psd estimate $P(m, \mu)$ can be written as $(0 \leq \alpha < 1)$

$$P(m, \mu) = (1 - \alpha) \sum_{i=0}^{\infty} \alpha^i |X(m - i, \mu)|^2. \tag{3.12}$$

Since the pdf of $P(m, \mu)$ is scaled by $\sigma_N^2(m, \mu)$ the minimum statistics of the short term psd estimate is also scaled by $\sigma_N^2(m, \mu)$. Therefore, the mean $E\{P_{min}(m, \mu)\}$ is proportional to $\sigma_N^2(m, \mu)$ and the variance is proportional to $\sigma_N^4(m, \mu)$. Without loss of generality, it is sufficient to compute the mean and the variance for $\sigma_N^2(m, \mu) = 1$. We use the notation $B_{min}^{-1} = E\{P_{min}(m, \mu)\}_{|\sigma_N^2(m,\mu)=1}$ and determine the mean B_{min}^{-1} of the minimum of correlated varieties $P(m, \mu)$ as a function of the inverse normalized variance $2\sigma_N^4(m, \mu)/var\{P(m, \mu)\} = Q_{eq}(m, \mu)$ by generating large amounts of exponentially distributed data with variance $\sigma_N^2 = 1$ and by averaging minimum values for various values of D. The inverse normalized variance $Q_{eq}(m, \mu)$ is also called "equivalent degrees of freedom" since nonrecursive (moving average) smoothing of $Q_{eq}(m, \mu)$ independent squared Gaussian varieties would yield an estimate with the same variance. We approximate the inverse mean of the minimum by

$$B_{min}(m, \mu) \approx 1 + (D - 1)\frac{2}{\tilde{Q}_{eq}(m, \mu)}\Gamma\left(1 + \frac{2}{Q_{eq}(m, \mu)}\right)^{H(D)}, \tag{3.13}$$

where $\tilde{Q}_{eq}(m, \mu)$ is a scaled version of $Q_{eq}(m, \mu)$. $M(D)$ and $H(D)$ are functions of D. $\Gamma(.)$ denotes the complete Gamma function.

$$\tilde{Q}_{eq}(m, \mu) = \frac{Q_{eq}(m, \mu) - 2M(D)}{1 - M(D)} \tag{3.14}$$

Table 3.1 lists values for $M(D)$ and $H(D)$ as a function of D. Other values can be obtained by linear interpolation.

D	M(D)	H(D)
1	0	0
2	0.26	0.15
5	0.48	0.48
8	0.58	0.78
10	0.61	0.98
15	0.668	1.55
20	0.705	2.0
30	0.762	2.3
40	0.8	2.52
60	0.841	2.9
80	0.865	3.25
120	0.89	4.0
140	0.9	4.1
160	0.91	4.1

Tabel 3.1:parameters for the approximation of the mean of the minimum

This approximation has a mean square error over the range of values of less than $4 \cdot 10^{-4}$ and a peak relative error of less than 4%. The largest errors are obtained for small values of Q_{eq}. For values $Q_{eq} \geq 8$ the peak errors is always below 2%. In a real-time application with fixed window length D, $M(D)$ and $H(D)$ will be precomputed and (3.13) and (3.14) will be evaluated during runtime.

For software implementations we use the simplified approximation

$$B_{min}(m, \mu) \approx 1 + (D-1)\frac{2}{\tilde{Q}_{eq}(m, \mu)}. \tag{3.15}$$

An unbiased estimator of the noise power spectral density $\sigma_N^2(m, \mu)$ is given by

$$\hat{\sigma}_N^2(m, \mu) = \frac{P_{min}(m, \mu)}{E\{P_{min}(m, \mu)\}|_{\sigma_N^2(m,\mu)=1}} = B_{min}(D, Q_{eq}(m, \mu))P_{min}(m, \mu). \tag{3.16}$$

The unbiased estimator requires the knowledge of the normalized variance $var\{P(m, \mu)\}/(2\sigma_N^4(m, \mu)) = 1/Q_{eq}(m, \mu)$ of the smoothed psd estimate $P(m, \mu)$ at any given time and frequency index.

To estimate the variance of the smoothed psd estimate $P(m, \mu)$ we use a first order smoothing recursion for the approximation of the first moment $E\{P(m, \mu)\}$, and the second moment, $E\{P^2(m, \mu)\}$, of $E\{P(m, \mu)\}$

$$\begin{aligned}
\tilde{P}(m, \mu) &= \beta(m, \mu)\tilde{P}(m-1, \mu) + (1 - \beta(m, \mu))P(m, \mu) && (3.17)\\
\widetilde{P^2}(m, \mu) &= \beta(m, \mu)\widetilde{P^2}(m-1, \mu) + (1 - \beta(m, \mu))P^2(m, \mu) && (3.18)\\
\widehat{var}\{P(m, \mu)\} &= \widetilde{P^2}(m, \mu) - \tilde{P}^2(m, \mu) && (3.19)
\end{aligned}$$

where $\beta(m,\mu) = \alpha^2(m,\mu)$ which is limited to values less or equal to 0.8.

Finally, $1/Q_{eq}(m,\mu)$ is estimated by

$$\frac{1}{Q_{eq}(m,\mu)} \approx \frac{\widehat{var}\{P(m,\mu)\}}{2\hat{\sigma}_N^4(m-1,\mu)} \tag{3.20}$$

and this estimate is limited to a maximum of 0.5 corresponding to $Q_{eq} = 2$. Since an increasing noise power can be tracked only with some delay the minimum statistics estimator has a tendency to underestimate highly nonstationary noise. Furthermore, because the bias compensation(3.13)(3.14) depends on the estimated normalized variance the bias compensation factor is a random variable with a variance depending on the variance of $P(m,\mu)$ It is therefore advantageous to increase the inverse bias $B_{min}(m,\mu)$ by a factor $B_c(m)$ proportional to the normalized standard deviation of the short term estimate $P(m,\mu)$, $B_c(m) = 1 + a_v\sqrt{\overline{Q^{-1}}(m)}$ with the average normalized variance $\overline{Q^{-1}}(m) = (1/L)\sum_{\mu=0}^{L-1} 1/Q_{eq}(m,\mu)$ and a_v typically set to $a_v = 2.12$.

3.2.1.3 Efficient Implementation of the Minimum Search

This algorithm divides the window of D samples into U subwindows of V samples (UV = D). This allows us to update the minimum every V samples while keeping the computational complexity low. Whenever V samples are read the minimum of the current subwindow is determined and stored for later use. The overall minimum is obtained as the minimum of all U subwindow minima. We therefore have 1 + (U - 1)/V compare operations per signal frame and frequency bin.

For less stationary noise the tracking can be improved by looking in each subwindow for local minima with amplitudes in the vicinity of the overall minimum. A minimum of a subwindow is considered to be local if its value was not obtained in the first or the last signal frame of this subwindow. Since we now explicitly consider the minima of the subwindows we also have to compute a bias compensation for these shorter subwindows.

The new algorithm is summarized as follows:

- compute smoothing parameter $\hat{\alpha}(m,\mu)$: (3.10)
- compute smoothed power $P(m,\mu)$: (3.4)
- compute bias correction $B_{min}(m,\mu)$ and $B_{min\_sub}(\lambda,\mu)$: (3.13-15,3.20)
- compute $\overline{Q^{-1}}(m) = \frac{1}{L}\sum_{\mu=0}^{L-1}\frac{1}{Q(m,\mu)}$
- set $\mu\_mod(\mu) = 0$ for all μ
- if $P(m,\mu)B_{min}(m,\mu)B_c(m) < actmin(m,\mu)$

```
    – $actmin(m,\mu) = P(m,\mu)B_{min}(m,\mu)B_c(m)$
    – $actmin\_sub(m,\mu) = P(m,\mu)B_{min\_sub}(m,\mu)B_c(m)$
    – set $\mu\_mod(\mu) = 1$;
• if $subwc == V$
    – if $\mu\_mod(\mu) == 1$
      $lmin\_flag(\mu) = 0$
    – store $actmin(m,\mu)$
    – find $P_{min\_u}$, the minimum of the last $U$ stored values of $actmin$
    – if $\overline{Q^{-1}}(m) < 0.03, noise\_slope\_max = 8$;
    – elseif $\overline{Q^{-1}}(m) < 0.05, noise\_slope\_max = 4$;
    – elseif $\overline{Q^{-1}}(m) < 0.06, noise\_slope\_max = 2$;
    – else $noise\_slope\_max = 1.2$;
    – if $(lmin\_flag(m,\mu))\&(actmin\_sub(m,\mu) < noise\_slope\_maxP_{min\_u}(m,\mu))$
         $\&(actmin\_sub(m,\mu) > P_{min\_u}(m,\mu))$
      $P_{min\_u}(m,\mu) = actmin\_sub(m,\mu)$
      replace all previously stored values of $actmin(m,\mu)$ by $actmin(m,\mu)$
    – $lmin\_flag(\mu) = 0$;
    – set $subwc = 1$, and $actmin(m,\mu)$ and $actmin\_sub(m,\mu)$ to their maximum values
• else
    – if $subwc > 1$
      if $\mu\_mod(\mu) == 1$
      set $lmin\_flag(m,\mu) = 1$
      compute $\widehat{\sigma}_N^2(m,\mu) = min(actmin\_sub(m,\mu), P_{min\_u}(m,\mu))$
      set $P_{min\_u}(m,\mu) = \widehat{\sigma}_N^2(m,\mu)$
    – set $subwc = subwc + 1$
```

All computations are embedded into loops over all frequency indices μ and all time indices m. Subwindow quantities are subscripted by *sub*. In the description of the algorithm we make reference to a subwindow counter *subwc* which counts the signal frames within a subwindow and to the running minimum estimate $actmin(m,\mu)$. At the startup of the program this counter is initialized to $subwc = V$ and $actmin(m,\mu)$ is initialized to a preset maximum value. The vector $P_{min\_u}(m,\mu)$ holds the overall minimum of the length D window. It is updated whenever $subwc == V$, when the current minimum $actmin(m,\mu)$ becomes smaller than $P_{min\_u}(m,\mu)$, or when a local minimum is detected.

The search range *noise_slope_max* for local minima is within 0.8 to 9 dB of the current overall minimum. It depends on the average normalized variance $\overline{Q^{-1}}(m)$ of the short term psd estimate. If the variance is small a local minimum very likely indicates the noise level. It can therefore be accepted even if it is several dB larger than the current overall minimum. An increasing noise level can be therefore tracked on the subwindow level. If the variance is large fluctuations of local minima are not necessarily due to a rising noise floor. Therefore, only minima close to the overall minimum are accepted. The functional dependence of the variance and the search range for local minima was optimized by experiments.

$\mu\_mod(\mu)$ and $lmin\_flag(m,\mu)$ are auxiliary vectors for keeping track of those frequency bins which might contain local minima. If the minimum of a subwindow was determined as the first ($subwc == 1$) or the last ($subwc == V$) value of this subwindow it is not accepted as a local minimum ($lmin\_flag(m,\mu = 0)$). If the minimum was obtained in between the first or the last value of the subwindow it is marked as a local minimum ($lmin\_flag(m,\mu) = 1$). If a local minimum is larger than the overall minimum but still within the search range *noise_slope_max* it replaces all previously stored subwindow minima and thus leads to an increased noise psd estimate.

3.2.2 Noise Estimator using Blind Source Separation and Two-Channel Energy based Speaker Detection

In section 3.2.1 we introduced the single channel noise estimator based on minimum statistics, but it is easy to see, that we will certainly use at least two channels for the blind source separation. Since the slow nonstationary noise information can be exploited by using multiple microphones, in the following a combined spatial/temporal speech enhancement approach based on blind source separation is introduced [4]. Fig. 3.2 is the system diagram for this. Spatial information about interfering point sources is processed in the blind source separation unit while the remaining stage (Background Denoising using Desired Speaker Activity Detection) removes distributed background noise by a mixed temporal/spatial processing approach.

In the following we use the proposed novel robust, model-independent measure based on two channel information to detect desired speaker containing time-intervals. If the energy of separated BSS channel $\hat{s}_i(t)$ over a time frame L is given by

$$E_L(\hat{s}_i(t)) = \sum_{t}^{t+L} \|\hat{s}_i(t)\|^2, \tag{3.21}$$

a two-channel energy ratio factor $\lambda(t)$ can be defined as

$$\lambda(t) = \exp\left(\frac{-\zeta \cdot max[E_L(\hat{s}_2(t)) - \xi\Delta E_L(x_2,\hat{s}_2)(t),\epsilon]}{max[E_L(\hat{s}_1(t)) - E_L(\hat{s}_2(t)) + \xi\Delta E_L(x_2,\hat{s}_2)(t),\epsilon]}\right) \tag{3.22}$$

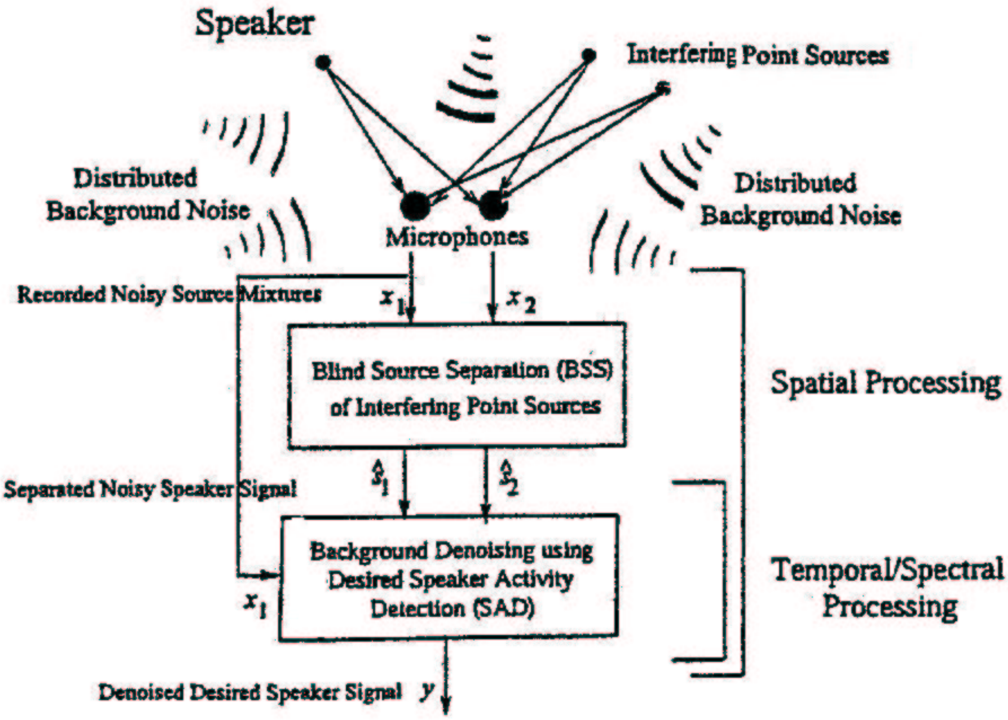
Speaker
Interfering Point Sources
Distributed Background Noise
Distributed Background Noise
Microphones
Recorded Noisy Source Mixtures
x_1
x_2
Blind Source Separation (BSS) of Interfering Point Sources
Spatial Processing
Separated Noisy Speaker Signal
$\hat{s}_1$
$\hat{s}_2$
Background Denoising using Desired Speaker Activity Detection (SAD)
x_1
Temporal/Spectral Processing
Denoised Desired Speaker Signal y

Figure 3.2: Proposed speech enhancement scheme.

for channel 1, where ϵ is a constant threshold. $\lambda(t)$ is computed over the whole signal length in an overlap-add fashion with shifting window of size L.

Indeed, since it is assumed that background noise energies are similar in each recorded mixture when microphones are positioned closed enough and the overall energy is preserved from recorded to separated sources because of the scaling constraint, the denominator in (3.22) is close to zero and hence λ tends to zero when the desired speaker is absent. If it is present, the energy in BSS channel 1 is much larger than in channel 2, the quotient in (3.22) tends to zero and thus λ to 1. In practice, BSS channel 2 may contain an interfering point source eliminated from channel 1. Therefore the ΔE term with

$$\Delta E_L(x_2, \hat{s}_2)(t) = max[\sum_t^{t+L}(\|x_2(t)\|^2 - \|\hat{s}_2(t)\|^2), \epsilon] \tag{3.23}$$

is used in (3.22) to robustify the detection of desired speaker by explicitly tracking energy changes from recorded channel 2 to separated BSS channel 2. Factor $\xi = \frac{\sum_t^{t+L}\|x_1\|^2}{\sum_t^{t+L}\|x_2\|^2}$ scales the energy change in channel 2 to a corresponding energy change in channel 1. The parameter ζ allows to adjust the "sharpness" of speech/non speech interval delimitation.

The resulting $\lambda(t)$ is used to provide a probability measure for the speaker's presence. The noise estimate is given by

$$P_n(m+1, \mu) = z_a \cdot P_n(m, \mu) + (1 - z_a)X_c(m, \mu) \tag{3.24}$$

where z_a is a smoothing constant and $X_c(m, \mu)$ is the auto power spectral density of $(1 - \lambda(t))\hat{s}_1(\mu, t)$, t in time frame L. The current speech plus noise power estimate is obtained from the recurrence

$$P(m+1, \mu) = z_g \cdot P(m, \mu) + (1 - z_g)X(m, \mu) \tag{3.25}$$

where z_g is a smoothing constant and $X(m, \mu)$ is the auto power spectral density of $\hat{s}_1(\mu, t)$, t in time frame L.

It was observed that using $\lambda(t)$ directly from (3.22) resulted in too aggressive denoising performance since the value of $\lambda(t)$ is not necessarily one at each local maximum and may decrease too rapidly near the edges of detected time-intervals, thereby cutting off speech parts. Hence $\lambda(t)$ is refined by replacing it by a sequence of Hanning windows with centers determined by the local maxima of $\lambda(t)$ from (3.22) and widths given by twice the distance between symmetric points around each maximum where $\lambda(t)$ reaches a certain threshold β. But it is easy to see that, if we replace the parameter $\lambda(t)$ with a sequence of Hann window like that, this method introduces an additional delay to detect when $\lambda(t)$ passes the treshold β.

3.3 Computation of Spectral Weighting

In section 3.2 we showed how we can estimate the noise psd and signal psd. In the following we will talk about the subtraction rule which is used for the computation of spectral weighting.

With the estimated subband noise psd and short time signal psd we subtract spectral magnitudes with an oversubtraction factor $osub(\lambda, \mu)$ and a limit the maximum subtraction by a spectral floor constant $subf$,

$$|Y(m,\mu)| = \begin{cases} subf \cdot P_n(m,\mu) & if |X(m,\mu)|Q(m,\mu) \leq subf \cdot P_n(m,\mu) \\ |X(m,\mu)|Q(m,\mu) & else \end{cases} \tag{3.26}$$

where $Q(m,\mu) = \left(1 - \sqrt{osub(m,\mu)\frac{P_n(m,\mu)}{|X(m,\mu)|^2}}\right)$. $subf$ is typically in the range $0.01 \leq subf \leq 0.5$.

While a large oversubtraction factor $osub(m,\mu)$ essentially eliminates residual spectral peaks ('musical noise') it also affects speech quality such that some of the low energy phonemes are suppressed. To limit this undesirable effect the oversubtraction factor is computed as a function of the subband signal-to-noise ratio

$$SNR_x(m,\mu) = 10log_{10}\left(\frac{P(m,\mu) - min(P_n(m,\mu), P(m,\mu))}{P_n(m,\mu)}\right) \tag{3.27}$$

and the frequency bin μ, i.e. $osub(m,\mu) = f(m,\mu,SNR_x(m,\mu))$. In general we use less oversubtraction for high SNR conditions and for high frequencies than for low SNR conditions and for low frequencies. In addition, because the functions for $osub$ should be as easy as possible for the implementation, we examined two main forms,

$$\begin{aligned} osub &= a + b/SNR_x(m,\mu); \\ osub &= c - d \times SNR_x(m,\mu). \end{aligned} \tag{3.28}$$

Here a, b, c, d are parameters for $osub$. a, c are responsible for the stationary noise and b, d will be used to reduce the noise peaks for nonstationary noise. The bigger the parameters are, the more noise will be reduced, but the speech signals will be disturbed too. Comparing with the second form, the first form will disturb the speech signal less. In (3.28) the $osub$ factor is depending on the $SNR(m,\mu)$ which is calculated for each block m and each frequency bin μ. In our experiments we use a hyperbolically decaying oversubtraction factor.

$$osub\_decaying(\mu) = a(1 + \mu f_s/(L \cdot b))^{-1} \tag{3.29}$$

(3.29) is the equation for the decaying in one block, a, b are constants chosen as $a = 4, b = 400$. L is the block length and f_s is the sampling frequency. As Fig.

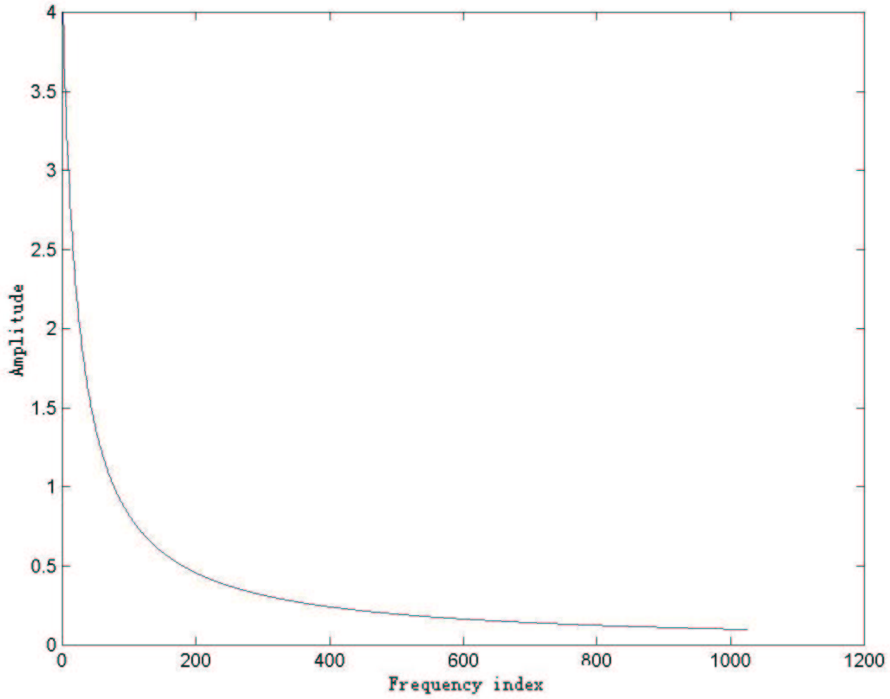

Figure 3.3: Hyperbolically decaying curve for *osub* factor L = 1024.

3.3 shows, for higher frequencies μ the signal will be affected less and less in one block. Additionally, *osub* is calculated iteratively,

$$\begin{aligned} &osub(m+1,\mu) = \\ &\eta \cdot osub(m,\mu) + (1-\eta)(1 + osub\_decaying(\mu)/SNR_x(m,\mu)), \end{aligned} \tag{3.30}$$

where η is the forgetting factor for the oversubtraction factor.

Chapter 4

Combination of BSS and NR

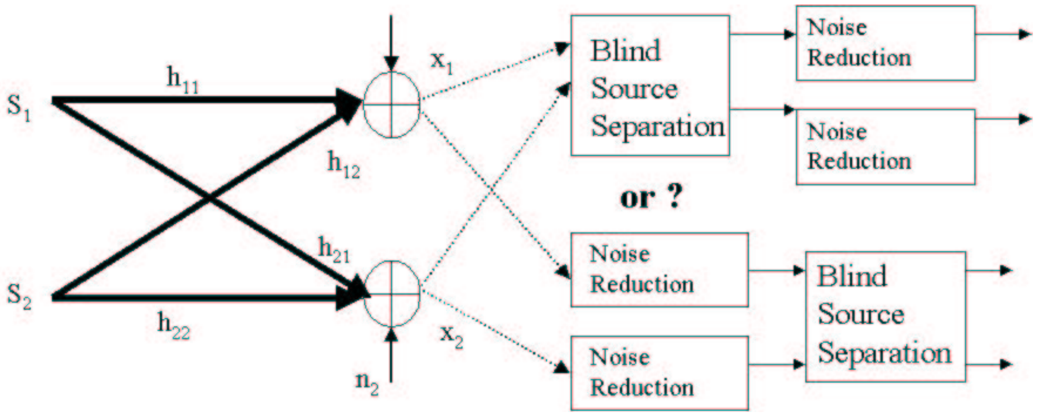

Figure 4.1: The combination of blind source separation and noise reduction.

It is easy to see that there are two methods for the combination between the two algorithms, first blind source separation then noise reduction (BSS → NR) or first noise reduction then blind source separation (NR → BSS), as Fig. 4.1 shows. Furthermore, for the method (BSS → NR) we also need to compare the two algorithms of noise power spectral density estimation: *Noise Power Spectral Density Estimation Based on Optimal Smoothing and Minimum Statistics* [3] and *Noise Estimator Using Blind Source Separation and Two-Channel Energy Based Speaker Detection* [4]. The algorithm in [4] needs to be concatenated after the BSS system, just as Fig. 3.2 shows. To evaluate these methods we will use some parameters which show the ratio of speech signal and interfering signal, respectively.

Chapter 5

Experimental Results

As in the introduction we have said, we use two microphones and two loudspeakers which is the easiest situation to compare the results of the different combination methods. The acoustic signals are sampled with the frequency $f_s = 16\text{kHz}$. The parameters for the two algorithms are selected as follows:

For the blind source separation algorithm we can learn from Martin Burger [15] that the on-line BSS algorithm worked well when $L/Q = 8$ where L is the FFT length and Q is the filter length, so we choose $L = 4096$ and $Q = 512$ according to the stationarity time of the signal. In addition, from [15] we know that the learning rate ν (Eq. 2.6) should be equal to one. The overlap factor is equal to four in our experiments. In other words, we take a signal block of size L and step by $L/4$ samples to obtain the next block of data. For the forgetting factor γ (Eq. 2.12) which is used to smoothen the estimate output signal we use 0.3.

Different from the blind source algorithm we use $L = 1024$ as the FFT length of the noise reduction algorithm and the update rate is not changed. From the experiments we choose the other parameters as shown here:

- smoothing time constant for signal power estimate $z_a = 0.85$ (Eq. 3.25);
- smoothing time constant for noise power estimate $z_g = 0.67$ (Eq. 3.24);
- smoothing constant for oversubtraction factor $\eta = 0.82$ (Eq. 3.30);
- subtraction floor $subf = 0.2$ (Eq. 3.26).

5.1 Condition of Experiments

5.1.1 Recording of Speech Signals in an Office Room

For the first experiment recordings are taken in an 590cm × 580cm × 310cm room with two unidirectional microphones with a distance of 16cm mounted on a desk

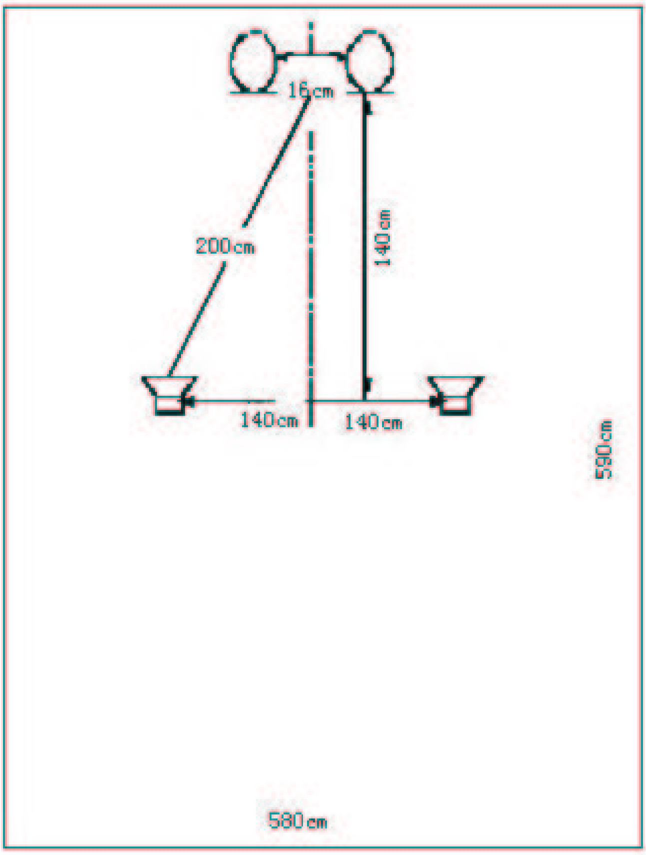

Figure 5.1: Layout of the room used for the recordings.

which is 116.5cm high. The speakers are sitting 140cm from the microphone setup. One speaker has 140cm deviation to left, the other speaker has 140cm deviation to right as Fig. 5.1 shows. The speech signals are two male speakers taken from the TIMIT database. For the background noise we added spatially diffuse noise which was recorded in a car using the same microphone array. The input signal segmental SIR is almost 0dB for the two channels, due to the speakers locations and the small microphone spacing.

5.1.2 Recordings in a Car

For the second experiment we obtained speech and noise recordings from TEMIC, which were performed in a car with different noise scenarios. The layout of the recordings is shown in Fig. 5.2. Now the distance of the two microphones is $70cm$. Each speaker has a distance of $30cm$ to the microphone which was mounted on the sun visor of the driver and co-driver respectively. We now use a female and a male german speaker in our experiments. The spatially distributed background noise is slowly non-stationary. Because of the long distance between the two microphones the average input signal segmental SIR is 11dB and 8dB for the two channels, respectively.

5.2 Evaluation of Experiments

As we explained in chapter 4, to compare the combination of the BSS algorithm and NR algorithm we will use some parameters to show the ratio of speech signal. In this section we will discuss these parameters and explain how we can compute these parameters.

5.2.1 SIR Computation

5.2.1.1 Accurate SIR Computation

In our experiments we measure performance with the Signal-to-Interference Ratio (SIR), which is defined for a signal in a multi-path channel to be the total signal powers of the direct channel versus the signal power stemming from cross channels. If we know the impulse response $h_{ij}(t)$ at all channels and the source signals $s_k(t)$, we can compute the expression directly by using a sample average over the available signal and multiplying the powers with the given direct and cross channel responses. For two channels this computation is simplified to

$$SIR_{output,channel1} = \tag{5.1}$$
$$10\log_{10}\frac{\sum_{t=0}^{N-1}(s_1(t)*h_{11}(t)*w_{11}(t))^2+\sum_{t=0}^{N-1}(s_1(t)*h_{21}(t)*w_{12}(t))^2}{\sum_{t=0}^{N-1}(s_2(t)*h_{12}(t)*w_{11}(t))^2+\sum_{t=0}^{N-1}(s_2(t)*h_{22}(t)*w_{12}(t))^2}$$

$$SIR_{output,channel2} = \tag{5.2}$$
$$10\log_{10}\frac{\sum_{t=0}^{N-1}(s_2(t)*h_{22}(t)*w_{22}(t))^2+\sum_{t=0}^{N-1}(s_2(t)*h_{12}(t)*w_{21}(t))^2}{\sum_{t=0}^{N-1}(s_1(t)*h_{21}(t)*w_{22}(t))^2+\sum_{t=0}^{N-1}(s_1(t)*h_{11}(t)*w_{21}(t))^2}$$

where N is the length of signal s and $*$ refers to the convolution.

Because the mixing system $h_{ij}(t)$ can be time-varying, we should use the Segmental-SIR (SSIR) instead of SIR. We break down all the N desired- and interferer-signal points in M short blocks with the length L, and then compute the SIR-values in every block.

$$SSIR_{output,channel1}(m) = \tag{5.3}$$
$$10\log_{10}\frac{\sum_{t}^{t+L}(s_1(t)*h_{11}(t)*w_{11}(t))^2+\sum_{t}^{t+L}(s_1(t)*h_{21}(t)*w_{12}(t))^2}{\sum_{t}^{t+L}(s_2(t)*h_{12}(t)*w_{11}(t))^2+\sum_{t}^{t+L}(s_2(t)*h_{22}(t)*w_{12}(t))^2}$$

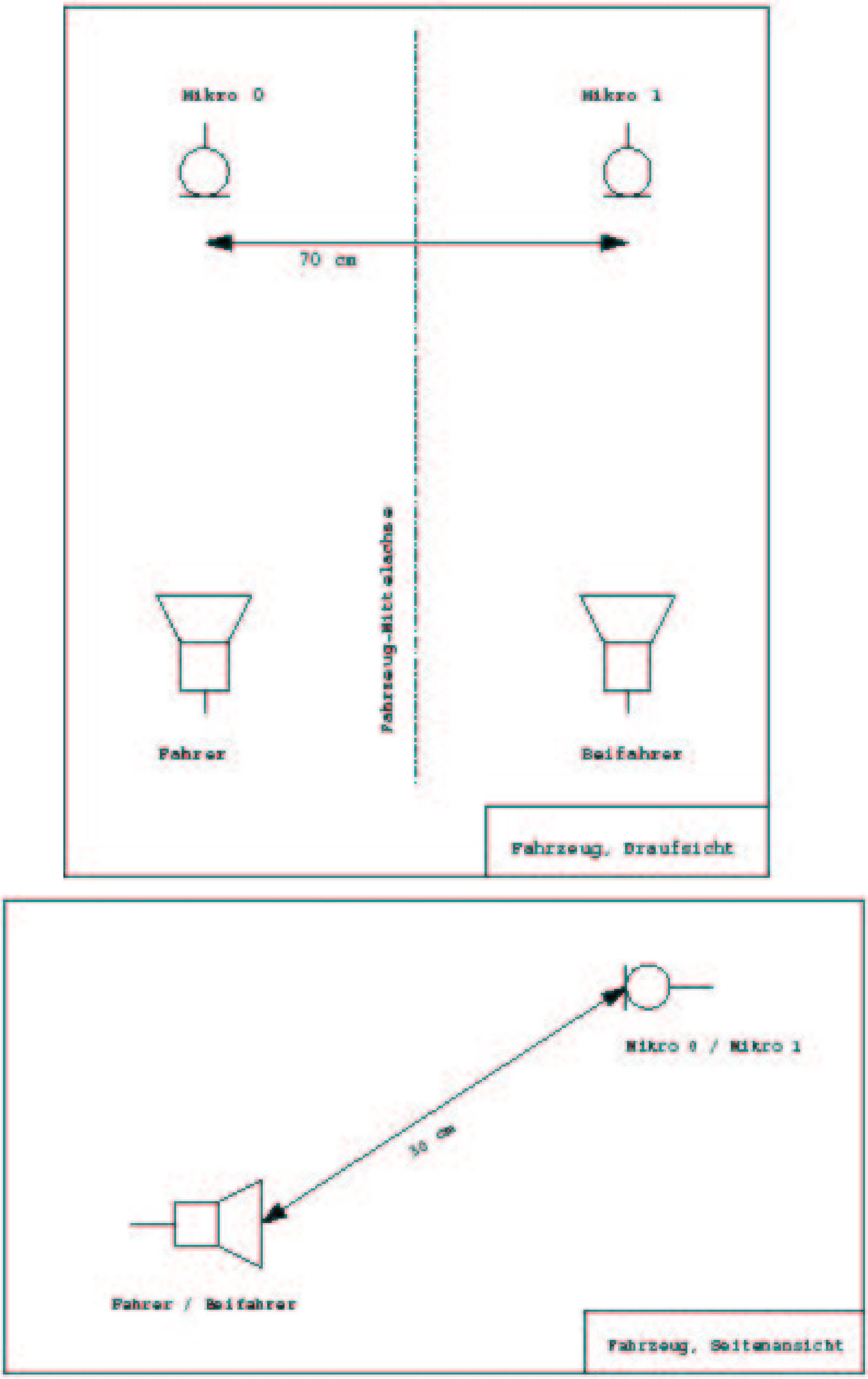

Figure 5.2: Layout for the setup used to record speech and noise signals in a car.

$$SSIR_{output,channel2}(m) = \tag{5.4}$$
$$10\log_{10}\frac{\sum_t^{t+L}(s_2(t)*h_{22}(t)*w_{22}(t))^2+\sum_t^{t+L}(s_2(t)*h_{12}(t)*w_{21}(t))^2}{\sum_t^{t+L}(s_1(t)*h_{21}(t)*w_{22}(t))^2+\sum_t^{t+L}(s_1(t)*h_{11}(t)*w_{21}(t))^2}$$

where m is the block index.

And when blind source separation is combined with noise reduction, we should still compute the impulse response of the noise reduction.

As Fig. 5.3 Fig. 5.4 shows, except the impulse response of mixing system $h(t)$ and the impulse response of blind source separation $w(t)$ we must still know the impulse response of noise reduction $r(t)$ which are also time-varying.

On the grounds of Fig. 5.3, the equation for the sequence (NR → BSS) of the first channel is

$$SSIR_{output,channel1,NR\rightarrow BSS}(m) = \tag{5.5}$$
$$10\log_{10}\frac{\sum_t^{t+L}(s_1(t)*h_{11}(t)*r_1(t)*w_{11}(t))^2+\sum_t^{t+L}(s_1(t)*h_{21}(t)*r_2(t)*w_{12}(t))^2}{\sum_t^{t+L}(s_2(t)*h_{12}(t)*r_1(t)*w_{11}(t))^2+\sum_t^{t+L}(s_2(t)*h_{22}(t)*r_2(t)*w_{12}(t))^2}$$

and on the grounds of Fig. 5.4, the equation for the sequence (BSS → NR) of the first channel is

$$SSIR_{output,channel1,BSS\rightarrow NR}(m) = \tag{5.6}$$
$$10\log_{10}\frac{\sum_t^{t+L}(s_1(t)*h_{11}(t)*w_{11}(t)*r_1(t))^2+\sum_t^{t+L}(s_1(t)*h_{21}(t)*w_{12}(t)*r_1(t))^2}{\sum_t^{t+L}(s_2(t)*h_{12}(t)*w_{11}(t)*r_1(t))^2+\sum_t^{t+L}(s_2(t)*h_{22}(t)*w_{12}(t)*r_1(t))^2}$$

5.2.1.2 SIR Estimation

The foregoing method of SIR computation is based on the condition that we know the impulse response of mixing system and source signals. In the case that we don't know the impulse response of mixing system, we can estimate the direct powers (numerator) and cross-powers (denominator) by using alternating signals. We estimate the contributions of source 1 while source 1 is 'on' and another source 2 is 'off', for example. In this way we can estimate the SIR. In the same way, during periods of silence, i.e. the two sources are 'off' we can estimate background noise powers in all channels to estimate the signal powers for the SNR computation which will be discussed in the next section.

5.2.2 SNR Computation

As we know, Signal-to-Noise Ratio (SNR) is a well known and very important parameter for the measure of the signal quality. Because in our experiments the input signal $x(t)$ is generated by the mix of two speech and noise signals, which is

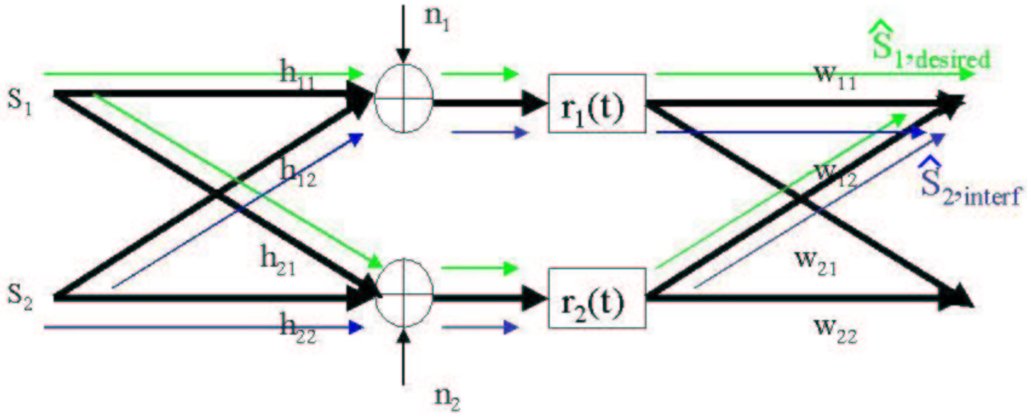

Figure 5.3: SSIR computation for channel 1 for the combination using (NR → BSS)

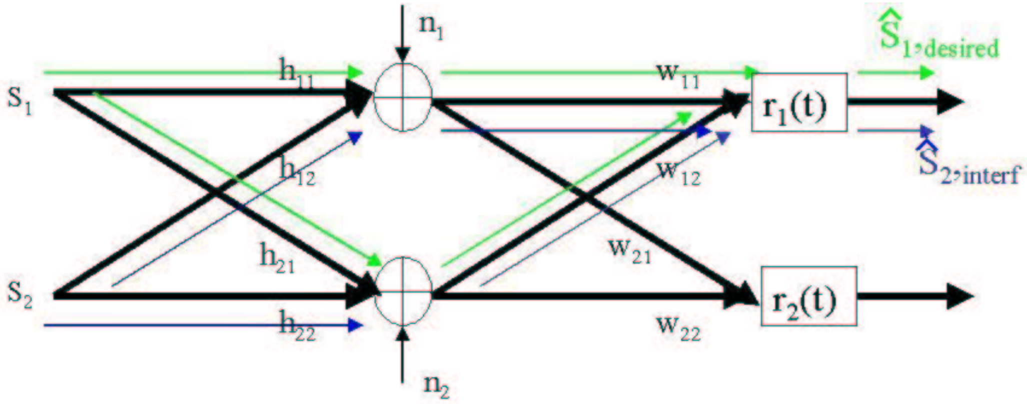

Figure 5.4: SSIR computation for channel 1 for the combination using (BSS → NR)

shown in Fig. 4.1, the input signal SNR can be directly computed by the signal variance,

$$SNR_{input,mic1} = 10\log_{10}\left(\frac{\sum_{t=0}^{N-1}(s_1(t) * h_{11}(t))^2 + \sum_{t=0}^{N-1}(s_2(t) * h_{12}(t))^2}{\sum_{t=0}^{N-1}(n_1(t))^2}\right) \tag{5.7}$$

$$SNR_{input,mic2} = 10\log_{10}\left(\frac{\sum_{t=0}^{N-1}(s_2(t) * h_{22}(t))^2 + \sum_{t=0}^{N-1}(s_1(t) * h_{21}(t))^2}{\sum_{t=0}^{N-1}(n_2(t))^2}\right) \tag{5.8}$$

and because of the same reason as section 5.2.1.1 we compute Segmental-SNR (SSNR) for every block instead of SNR,

$$SSNR_{input,channel1}(m) = 10\log_{10}\frac{\sum_{t}^{t+L}(s_1(t) * h_{11}(t))^2 + (s_2(t) * h_{12}(t))^2}{\sum_{t}^{t+L} n_1(t)^2} \tag{5.9}$$

$$SSNR_{input,channel2}(m) = 10\log_{10}\frac{\sum_{t}^{t+L}(s_2(t) * h_{22}(t))^2 + (s_1(t) * h_{12}(t))^2}{\sum_{t}^{t+L} n_2(t)^2}. \tag{5.10}$$

However, when the BSS algorithm and NR algorithm are combined together, a new problem is that the signal psd for the SNR computation is changed. Although the blind source separation system will not affect the background noise, it will still reduce the signal SNR. For example, when the blind source separation system is optimal, the interference speech signal will be totally filtered. Then the output speech signal power at channel 1 is var($s_1 * h_{11}$) which is of course smaller than the input speech signal psd var($s_1 * h_{11} + s_2 * h_{12}$) where var($s$) is the variance of s. With the smaller output signal psd, we will certainly decrease the SNR enhancement. Therefore, we compute the SNR enhancement between the input and output of the noise reduction system rather than the input signal and the output signal, just as Fig.5.5 and Fig.5.6 show.

The SNR enhancement is equal to

$$SNR_{input} - SNR_{output} = 10\log_{10}\frac{\frac{P_s}{P_n}}{\frac{\hat{P}_s}{\hat{P}_n}} \tag{5.11}$$

where P_s, P_n are the input speech and noise signal psd and $\hat{P}_s, \hat{P}_n$ are the output psd. As the speech signal psd before and after noise reduction system is approximately the same $P_s \approx \hat{P}_s$, (5.11) can be approximated by

$$SNR_{input} - SNR_{output} = 10\log_{10}\frac{\hat{P}_n}{P_n} \tag{5.12}$$

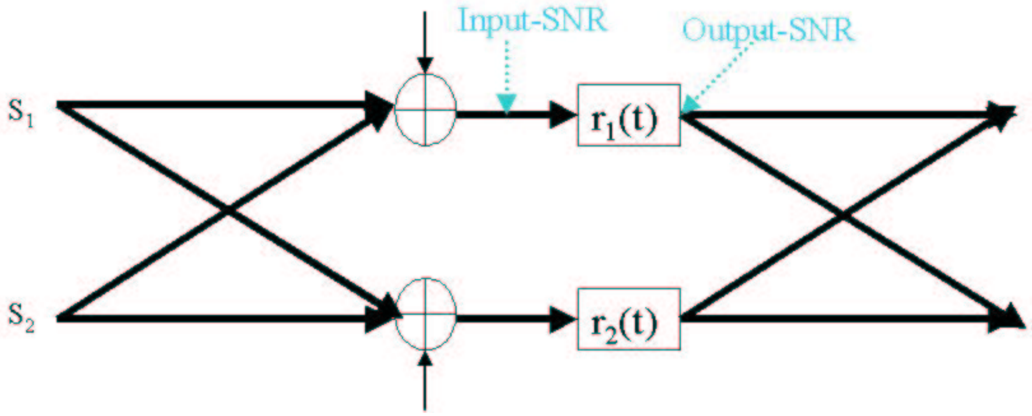

Figure 5.5: SSNR computation for channel 1 (NR → BSS)

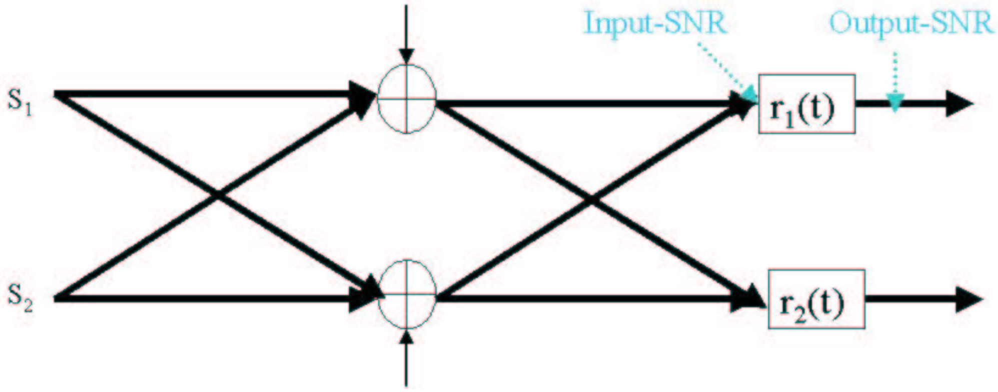

Figure 5.6: SSNR computation for channel 1 (BSS → NR)

which is not related to signal psd anymore. So we can compare accurately the difference of the noise reduction between the two methods.

Of course, with the noise psd estimation explained in section 3.2.1 we can estimate the noise psd directly from the noisy signal and then the SNR computation is very easy. However, with this method the error of SNR computation is sometimes not negligible. So for a precise SNR it is still necessary to compute the accurate SNR.

5.2.3 Threshold for SSNR and SSIR

In section 5.2.1 and 5.2.2 we show how we can compute the SSNR and SSIR, but because of the non-stationarity of the speech signals the SSNR and SSIR will result in negative dB values for speech pauses. So to compute the mean value of the SSNR and SSIR we introduce a threshold to discard the SSNR and SSIR values where the speech signal is not active. We choose the threshold to be equal to the long term average of the whole speech signal power

$$speech\_level = 10 \log_{10} \sum_t (s^2). \tag{5.13}$$

5.3 Experimental Results

5.3.1 Results for the Real Room Recordings

For the experiments we keep the input signal $\mathbf{x}(t)$ at a constant SIR: 1dB and -1dB, respectively. And we use the input noise signal power to change the input signal SSNR. Then with the enhancement of SSIR and SSNR we can compare the difference for the both combination methods.

From Fig. 5.7 we can see the relationship between the different combinations and the input SSNR. Here the mean of the SSNR and SSIR enhancement is shown where we also averaged over both channels.

As Fig. 5.7 a) shows, the SSIR enhancement is increased with the raise of the input signal SSNR. But there is only very small difference between the both combination for the SSIR enhancement. Fig. 5.7 b) depicts the SSNR enhancement. At low input SSNR the spectral subtraction does not work well due to the worse noise psd estimation and at high input SSNR the spectral subtraction can not improve the speech quality much further. So the SSNR enhancement curve is first increased then decreased with the raise of the input signal SSNR. The SNR enhancement is bigger for the combination (NR→BSS). However, the difference between the both combination is smaller than 1 dB as Fig. 5.7 shows.

Fig. 5.8 shows the time-variant SSNR and SSIR enhancement for $SSNR_{input} = 10dB$. Both curves of a) SSIR enhancement, b) SSNR enhancement are averaged

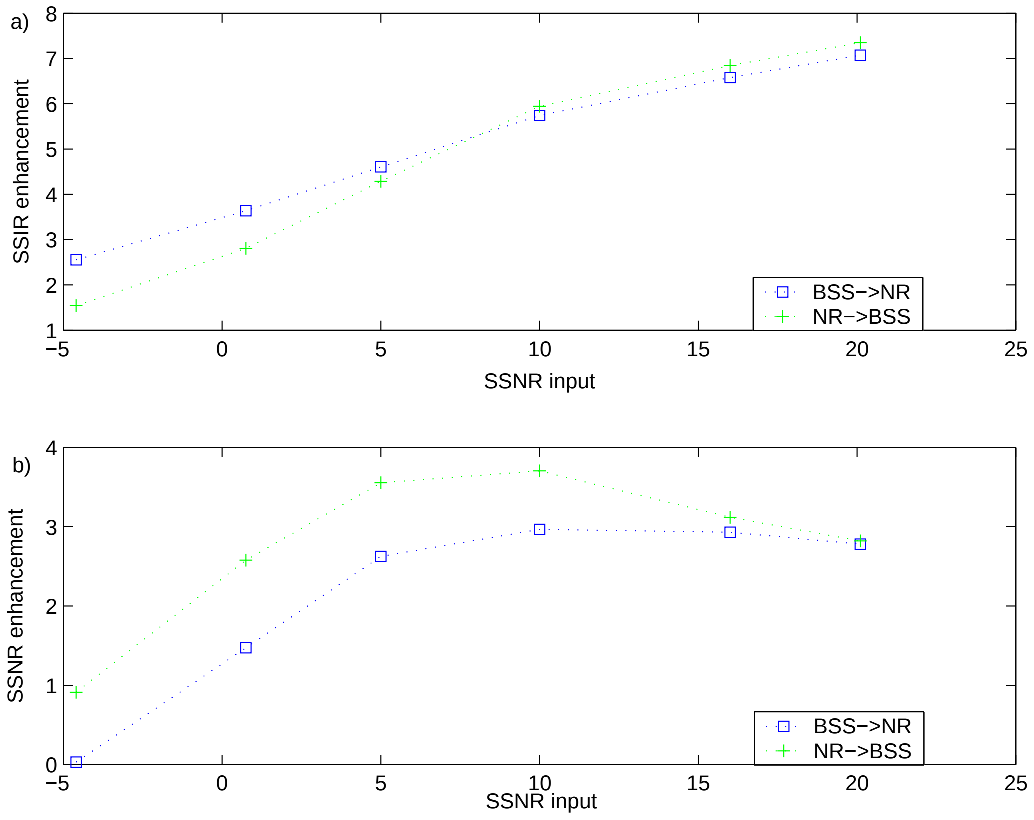

Figure 5.7: the SSNR and SSIR enhancement averaged over all blocks & channels for the real room recordings.

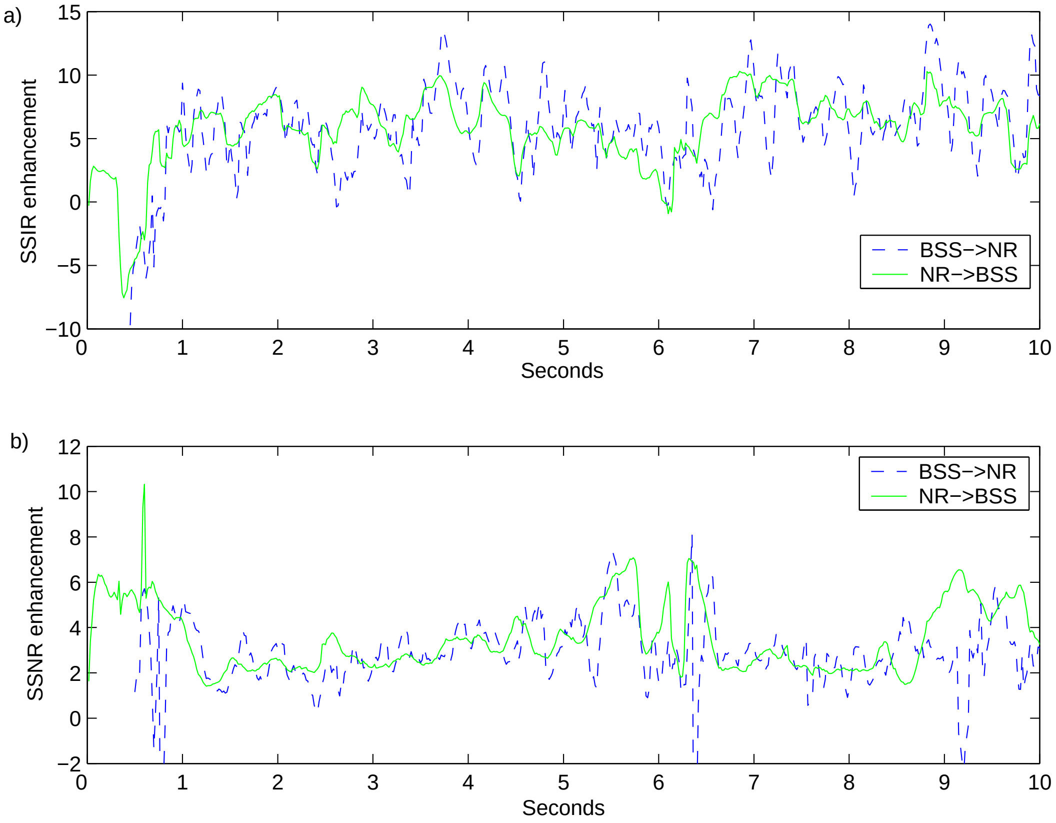

Figure 5.8: SSNR and SSIR enhancement averaged over both channels.

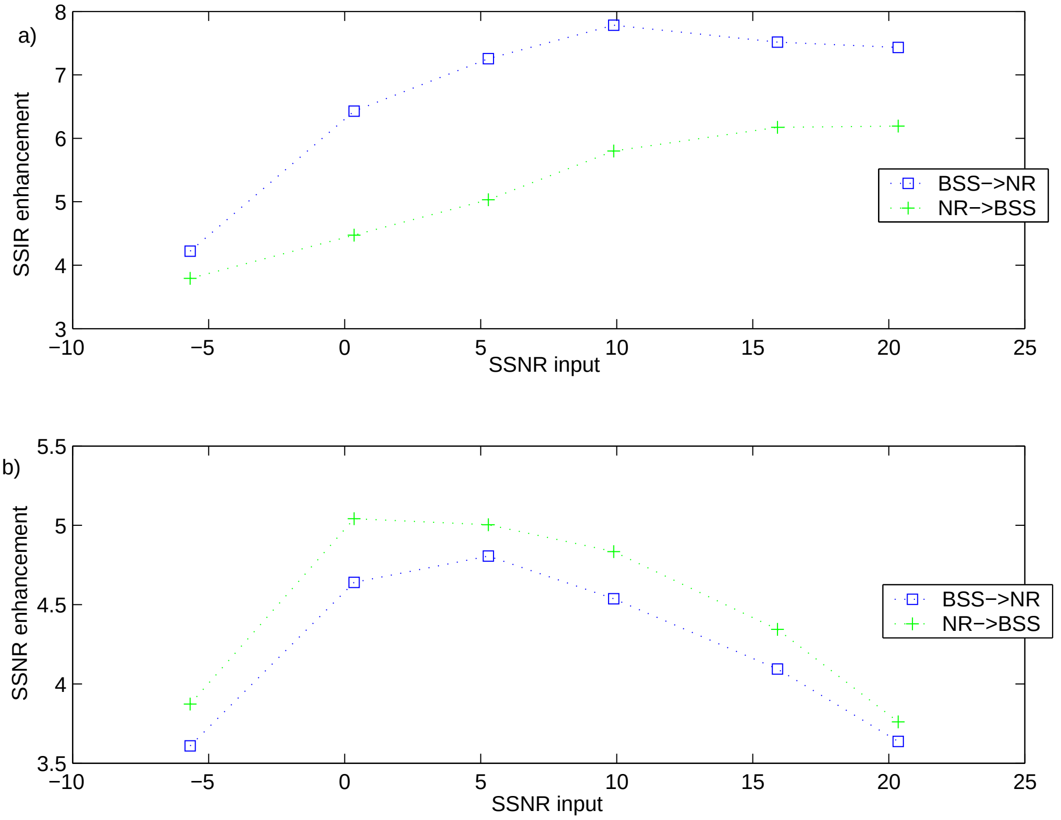

Figure 5.9: SSNR and SSIR enhancement averaged over all blocks & channels in a car.

over the two channels. From this figure we can see that the main difference of the two combinations is the smoothness of the SSNR and SSIR curves. The (NR→BSS) enhancement lines are smoother than the others, i.e. the variance for this combination is smaller. However, this difference will not affect the quality of the output signals.

5.3.2 Results for the Car Recordings

Same as in the first experiment, we keep the input signal $\mathbf{x}(t)$ at a constant SIR (11dB and 8dB) and use the input noise signal power to change the input signal SSNR. The enhancement of SSIR and SSNR will show the difference for the both combination methods. Here in Fig. 5.9 the mean over all blocks and channels of the SSNR and SSIR enhancement is shown similar to Fig. 5.7.

It is known [14] that for a high input SIR the achievable SIR improvement by a BSS system decreases. In this experiment, however, the SSIR enhancement is bigger (Fig. 5.9) compared to a more balanced input SSIR (Fig. 5.7). The surprising fact can be explained by the combination of BSS and NR. The reason for the high SSIR-improvement is that not only the BSS system but also the NR system will reduce the interfering signal power. Because in this simulation the input SSIR is very high, in other words, the interfering signal power is much smaller than the desired speech signal power, the NR system will process the interfering signal as additional noise and restrain it. This is the reason why we get similar results compared to section 5.3.1 where a balanced input SSIR prevails.

From Fig. 5.9 a) we can see that there is about a 1-2dB difference of SSIR enhancement between the two combinations. The combination (BSS→NR) is a little better than the others in this case.

The SSNR enhancement behaves similarly to section 5.3.1 and not much difference between both combinations can be observed.

5.3.3 Comparison of the Two Different Noise Reduction Methods

The section 5.3.1 and 5.3.2 compare the different combinations of the BSS algorithm which was introduced in chapter 2 and the NR algorithm which was introduced in section 3.2.1 and we find that the combination (BSS→NR) is a little bit better. In section 3.2.2 we introduced still another NR algorithm from Erik Visser and Te-Won Lee [4] which can only be used with the combination (BSS→NR), because this NR algorithm need the two-channel energy detection based on blind source separation, just as Fig. 3.2 shows. In this section we will compare the two NR algorithms in practice.

With our experiments we can observe that using the algorithm in [4] resulted in too aggressive reduction performance since the value of the subtraction factor $Q(m, \mu)$ (Eq. 3.26) decrease too fast near the edges of the detected time-intervals. On the contrary, in the speech parts the value of $Q(m, \mu)$ is too big to reduce the noise. Fig. 5.10 shows the SSNR enhancement for the two NR algorithms with $SSNR_{input} = 10dB$. Here we use the same speech and noise signals as in section 5.3.1. We can see that minimum statistics is much better than the algorithm in [4]. The reason is that the algorithm based on minimum statistics estimates the noise psd in the frequency-domain, but the algorithm in [4] estimates the noise power in the time-domain.

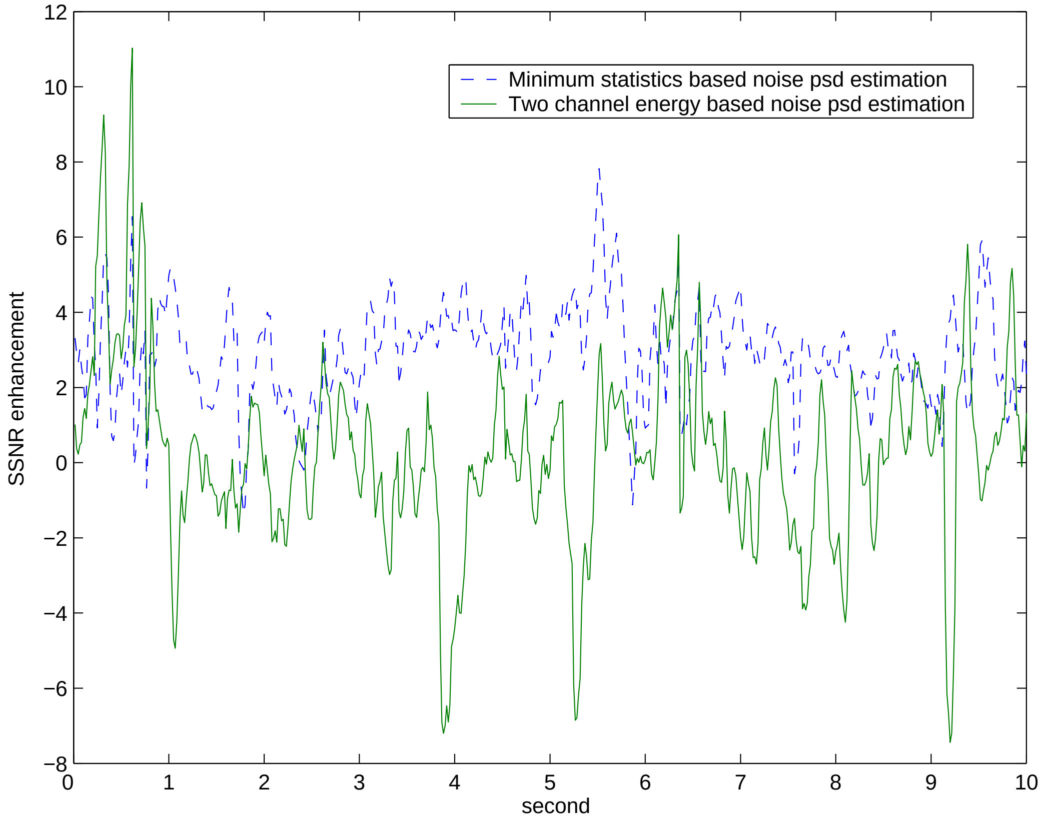

Figure 5.10: SSNR enhancement averaged over both channels.

Chapter 6

Conclusion

In this work we compared the performance of the two combination methods with different signals. We can observe that the NR algorithm using the spectral subtraction will not only reduce the power of the noise signal, but also disturb the speech signal as all denoising algorithms. It is the main reason for the difference between the two combinations (BSS→NR) (NR→BSS).

The on-line BSS algorithm will be affected by the background noise. Due to the loss of phase information in NR systems, it is not possible to increase the BSS performance essenitially by using a channel-wise NR pre-processing. With our experiments we can say that in the common scenarios the performance of both combinations is almost the same.

From our experiments we can also conclude that the SSIR-improvement obtained by the BSS-system decreases with high input SSIR. In the combination with NR methods it was observed that the interfering speaker was treated like additional noise and therefore high SSIR improvements like in the more balanced input SSIR case were achieved.

Furthermore, for the noise psd estimation we should use the algorithm in [3] instead of the algorithm in [4].

Bibliography

[1] Lucas Parra, Clay Spence. *Online Blind Source Separation of Non-Stationary Signals*. Journal of VLSI Signal Processing, vol. 26, no.1/2, pp. 39-46. Aug. 2000.

[2] Rainer Martin. *Spectral Subtraction Based on Minimum Statistics*. EUSIPCO-94, Edinburgh, Scotland. pp. 1182-1185. Sep. 1994.

[3] Rainer Martin. *Noise Power Spectral Density Estimation Based on Optimal Smoothing and Minimum Statistics*. IEEE Trans. Speech and Audio Processing. Vol.9, No.5, Jul. 2001.

[4] Erik Visser, Te-Won Lee. *Speech Enhancement Using Blind Source Separation and Two-Channel Energy Based on Speaker Detection*. ICASSP 2003 Volume I, pp 836-839. International Conference on Acoustics, Speech and Signal Processing, HongKong, China, 2003.

[5] Walter Kellermann. *Signal Processing for Speech and Audio*. Institute for Multimedia Communications and Signal Processing, University Erlangen-Nürnberg. Lecture skript 2002/2003.

[6] Walter Kellermann. *Fundamentals of Digital Signal Prozessing I & II*. FAU Erlangen. Lecture skript 2000/2001.

[7] Rolf Unbehauen. *Systemtheorie I*. 7. Edition. Oldenbourg-Verlag. München, Wien. 1997.

[8] P.Vary, U.Heute, and W.Hess *Digitale Sprachsignalverarbeitung*. Teubner, Stuttgart, Germany, 1998.

[9] Lucas Parra, Clay Spence. *Convolutive Blind Source Separation of Non-Stationary Sources*. IEEE Trans. Speech and Audio Processing, pp. 320-327. May 2000.

[10] Lucas Parra, Clay Spence. *Separation of Non-Stationary Natural Signals*. in Independent Components Analysis, Principles and Practice. R.Everson S.Roberts, Ed., pp. 135-157. Cambridge University Press, Cambridge, 2001.

[11] Lucas Parra, Clay Spence, and Bert De Vries. *Convolutive Blind Source Separation Based on Multiple Decorrelation.* in Proc. Neural Networks for Signal Processing. Sep. 1998.

[12] Aapo Hyvärinen, Erkki Oja *Independent Component Analysis: A Tutorial.* Neural Computing Surveys 2:94–128. 1999.

[13] Martin Burger, *Anwendungen von blinden Quellentrennungsverfahren für akustische Mensch-Maschine Schnittstellen bei Laptop PC's.* Studienarbeit at the Institute for Multimedia Communications and Signal Processing, University Erlangen-Nürnberg. 2002.

[14] Stefan Wehr, *Real-Time Implementation of a Blind Source Separation Algorithm for Convolutive Mixtures.* Diplomarbeit at the the Institute for Multimedia Communications and Signal Processing, University Erlangen-Nürnberg. Dec. 2002.