NTNU
Det skapende universitet

Redusering av akustisk krysstale mellom kanaler i orkesteropptak ved bruk av adaptiv filtrering

Martin Hansen

Master i elektronikk
Oppgaven levert: Juni 2008
Hovedveileder: Peter Svensson, IET
Biveileder(e): Alfred Hanssen, Universitetet i Tromsø

Norges teknisk-naturvitenskapelige universitet
Institutt for elektronikk og telekommunikasjon

Oppgavetekst

Ved opptak av flere instrumenter i samme rom er det ofte et problem at lyden fra lydsterke instrumenter ikke bare blir tatt opp av "sin" mikrofon, men også en eller flere av de andre instrumentenes mikrofoner ("akustisk krysstale"). Hensikten med oppgaven er å studere hvorvidt adaptiv filtrering kan brukes for å redusere slik krysstale. Adaptiv filtrering kan brukes for å modellere impulsresponsen (i rommet) mellom mikrofonen til det lydsterke instumentet og de andre mikrofonene. Det uønskede bidraget fra det lydsterke instrumentet i disse mikrofonene kan da reproduseres, og ved å summere det i motfase med signalene fra mikrofonene vil det kanselleres. Oppgaven skal se på hvilke parametere som er viktige for at en slik redusering av krysstale skal lykkes, og hvorvidt det vil være praktisk mulig å bruke dette i miksebord til bruk ved konserter og i studio.

Oppgaven gitt: 15. januar 2008
Hovedveileder: Peter Svensson, IET

Reduction of acoustic crosstalk in sound mixers by use of adaptive filtering

Martin Hansen

Master's thesis in electronics
Specialization in acoustics

NTNU 2008

Project description

Student:	Martin Hansen
Teacher / supervisor:	Peter Svensson, NTNU/IET
Assistant supervisor:	Alfred Hanssen, University of Tromsø, Department of Physics
Written at:	Department of Electronics and Telecommunications (IET)
Title:	Reduction of acoustic crosstalk in sound mixers by use of adaptive filtering
Assignment given:	15. January 2008
Latest date of delivery:	10. June 2008
Assignment delivered:	June 10, 2008

Assignment text:

When recording several instruments in one room, there is often a problem with loud instruments: The sound from these instruments is recorded not only by "their" microphones, but also by one or more of the other instruments' microphones. This is termed "acoustic crosstalk". The intention of this project is to study whether adaptive filtering can be used to reduce such crosstalk. Adaptive filtering may be used to to model the impulse response (of the room) between the microphone of a loud instrument and the other microphones. The unwanted signal from the loud instrument in these microphones may then be reproduced, and by subtracting it from the other microphone signals, the crosstalk may be cancelled. The student should study which parameters are important for such a cancellation to be achieved, and whether it could be practically possible to implement this in a sound mixer for use at concerts or in a studio.

Abstract

In this work, the main task has been to investigate whether adaptive filtering techniques can be used for the cancellation of acoustic crosstalk between channels in a sound mixer. Emphasis has been placed on applications for musical instruments.

In a setting where one wants to record two or more instruments in the same room, one usually places a microphone close to each instrument, hoping that that instrument will dominate the sound field that is picked up by the microphone. Never the less, there will inevitably be some "leakage" of sound from the other instruments, which is also picked up by the microphone. This is an example of acoustic crosstalk.

In this work, adaptive filtering is investigated as a way to cancel such crosstalk. LMS-type algorithms are used to try to estimate the impulse response between the microphone closest to an instrument (the "reference microphone") and a microphone further away (the "room microphone"). This is nicknamed a "mic-to-mic" impulse response. If this impulse response can be estimated, the crosstalk in the room microphone can be estimated by filtering the signal from the reference microphone with the impulse response estimate. The modelled crosstalk can then be subtracted from the room microphone signal, thus cancelling the crosstalk partly or completely, depending on the accuracy of the estimate.

A theoretical study was done to determine whether a "mic-to-mic" impulse response exists, and whether the corresponding filter would be stable and causal. It was found that such a filter can be guaranteed if the impulse response between sound source and reference microphone is minimum phase. In an experiment, the impulse response between a loudspeaker and the reference microphone in a small, heavily damped room was found not to be minimum phase, even though the reverberation time was only 0,1 seconds and the microphone was standing 30 cm from the speaker. This suggests that few real rooms will fulfill the minimum phase criterion. Never the less, using the method described above, substantial damping of the crosstalk was found to be possible in many different cases.

A block-based LMS (least-mean-squares) adaptive algorithm, implemented in the frequency domain, was chosen as the adaptive algorithm for use in experiments. This algorithm was called the Frequency Block LMS (FBLMS) algorithm. The implementation in the frequency domain had two advantages: The convolution and correlation operations needed in the update of the adaptive filter could be computed very efficiently, and a frequency-dependent step size could be utilized to decorrelate the input signals and speed up convergence.

The method used in most of the work was termed "learn and freeze" – meaning that the adaptive filter first goes through a learning sequence, after which the filter coefficients are held constant. This method has the advantage of the final filter being linear and time-invariant. It is also quite easy to evaluate the performance of the adaptive algorithm when this method is used. An experiment revealed that although a high degree of crosstalk cancellation is possible using this method, even small changes in the actual impulse reponse in the room will result in audible fluctuations of the residue crosstalk.

A second method, termed the "continuous update" method, was also looked into. This method is based on continuously updating the filter coefficients of the crosstalk cancellation filter – equivalent to letting the learning phase of the learn and freeze method continue indefinitely. Several challenges of this method were pointed out. A simulated experiment was also conducted, and this revealed that so-called "doubletalk" (when other sound source are playing in addition to the one producing crosstalk) may give rise to unwanted distortion of the impulse response estimate and corresponding unwanted sound artifacts. The method is still seen to have potential, if a suitable doubletalk detector can be implemented to slow down or stop adaptation during doubletalk segments.

A loudspeaker playing white noise was used as a reference sound source, representing an approximate "best case". In a small heavily damped room, the maximum achieved damping of the crosstalk from this source was approximately 35 dB. This was achieved in the 500-1000 Hz octave bands, while damping was generally somewhat lower in bands above and below this. It was suggested that low-frequency background noise is the cause of less damping in the low-end of the frequency range, while time variance is the cause of less damping in the higher frequencies. Similar experiments in a larger, "ensemble" room resulted in maximal damping of approximately 27 dB, with less damping in the high and low end also in this case. Theoretical calculations of the maximally achievable damping were shown to be much more accurate for the larger ensemble room than for the small, damped room.

Measurements of the achievable crosstalk was also done for several musical instruments in the two rooms mentioned above. In general, in was found that the achievable reduction of crosstalk was quite similar in both the small, damped room and the larger ensemble room, as long as filter length and

adaptation time was adjusted to account for the different reverberation time of the rooms.

For many of the musical instruments, there was substantial crosstalk cancellation only in a few octave bands. These were most often the bands in which the instruments were able to radiate the most energy. It was suggested that the increased signal-to-noise ratio in these bands made a higher degree of crosstalk cancellation possible, and also that such high-energy bands are "prioritized" by the adaptive algorithm, to minimize the overall error. Complex directivities and a high degree of time variance are also suggested as reasons for poor crosstalk reduction in some cases, especially in high frequency bands. It was also pointed out that if there is crosstalk reduction only in a limited frequency range, the spectrum of the remaining crosstalk will be changed, making it sound "unnatural" in some cases. This problem was also illustrated through sound examples.

In some of the experiments using musical instruments as sound sources, measurements were done of both a "calm" and a "fast" musical playing style. Crosstalk cancellation was generally somewhat better when the "fast" style was used. It was suggested that this is due to the "fast" style resulting in an input signal to the adaptive algorithm which is less self-correlated.

The results of the crosstalk cancellation method using adaptive filtering were quite variable, with results depending strongly on what kind of sound source was used. For many of musical instruments, crosstalk was only possible in a few low-frequency octave bands, while substantial damping was possible across most frequency bands for a loudspeaker playing white noise and an electrically amplified guitar. This indicates that the methods investigated in this work may not be usable for any sound source in a practical application (like a sound mixer), but that much can be gained in some cases. The sound sources that seem to be yield the best results are those that are completely stationary, like a loudspeaker or a guitar amplifier.

Preface

Although this project is part of a master's programme at NTNU in Trondheim, I really wanted to do most of the work in Tromsø, since this is where my girlfriend Jorunn lives. My supervisor at NTNU, Peter Svensson, kindly agreed to let me do so, all the while staying in touch through a lot of email correspondence. For this I owe him great thanks.

I would also like to thank Alfred Hanssen at the University in Tromsø, for agreeing to be my assistant supervisor and giving me access to both measuring equipment and a sound lab – all without me even being a student at his university. He has also given me valuable input during my work, and even driven me around for measurements outside the university. Thank you!

I was also fortunate enough to have several students from the Music Conservatory at Tromsø University College help me, by playing their instruments and letting me record them for my experiments. Thank you, Anne Berg Schjønsby, Ruth Hals Karlsen, Jøran Hatten, Mats Roar Sakshaug, Andrei Sorokin, and Håkon Pettersen. I owe an extra thanks to Håkon for reserving the room for the experiments at the conservatory.

Finally I would like to cite the dedication from the book "Adaptive Signal Processing" by Bernard Widrow and Samuel D. Stearns, which has been very useful during my work. These words reflect a sincerity and optimism which sounds strangely naïve today, but which I truly appreciate:

[...] It is also dedicated to the cause of peace on earth. We hope and trust that its contents will be used the improve the lot of mankind everywhere.

Martin Hansen

Tromsø, June 10, 2008

Contents

1
Introduction

Background and basic idea

The use of several microphones to record several musical instruments is common in many situations - most notably in recording studios and on stage. Often one aims to record only one instrument per channel, and to reject "leakage" of sound from other instruments into the microphone - this gives the sound engineer maximum flexibility for mixing all the instruments into a mono, stereo or multichannel output. The leakage of sound "across the channels" is called "acoustic crosstalk". In a recording studio, a very effective way to avoid this is to separate each instrument physically – to put each musician in their own booth, and using headphones to make them hear each other.

Unfortunately, this approach is usually not practically feasible on stage, although screens of plexiglass are sometimes used around for example drums [23]. Here, all the musicians are in the same room and relatively close to each other, and the levels of acoustic crosstalk are naturally quite high. One way to reduce crosstalk in this case is to use directional microphones [9]. By aiming a directional microphone towards the sound source one wants to record, sounds coming from other directions (like sound from other instruments or unwanted reverberation) can be rejected to some degree.

In some cases, the use of directional microphones is not enough. It may not be possible to aim the microphone away from the interfering sound source, or the unwanted sound may be so strong that the microphone rejection off-axis does not provide enough damping of the crosstalk. But are there other possible ways to reduce unwanted crosstalk further?

What is described as an "interfering" sound source above is of course only interfering as crosstalk – usually this sound source is also recorded with its own microphone to become part of the overall mix. If we assume that we are able to record this sound source without too much crosstalk from other instruments, we actually have a separate recording of the sound that we want to remove. So can't we just subtract it from the recording with crosstalk? Unfortunately we can't, because the crosstalk is both scaled, delayed, and mixed with reflections from walls and other objects (reverberation). In addition to this, different

microphones and their positions may also introduce differences between the recorded sound and the crosstalk. In order to remove the unwanted crosstalk, we must filter the direct recording of the interfering sound source in a way which makes it as similar as possible to the crosstalk. Since it is impossible to make a standard filter which can be used for any stage and microphone configuration, the filter has to be adjustable. When a model of the crosstalk has been calculated, it can be subtracted from the signal containing crosstalk, leaving only the wanted signal.

In this work, we will look into the possibility of using adaptive filtering techniques to adjust such a filter to the situation and thereby reduce crosstalk. The adaptation of the filter and the filtering operation itself can be performed in software, built in as part of a digital mixer. In this way, crosstalk levels in "live" sound recordings can be reduced in the sound mixer rather than in the room.

Related work

In telecommunications, and especially concerning teleconferencing systems, the problem of acoustic crosstalk has been an issue for several years. In a telephone system, there is a microphone and a loudspeaker at each end. If the sound from the loudspeaker, conveying a message from the far end, is recorded by the microphone at the near end, the speaker at the far end will hear this as an echo of himself. This is especially a problem in teleconferencing and “speaker mode" in mobile phones. Using adaptive filtering, the loudspeaker sound recorded by the microphone can be estimated and subtracted, so that only the near-end speech is sent back. [8]

The problem of acoustic crosstalk cancellation is also one encountered when trying to use binaural techniques using loudspeakers, and not headphones. One possible use for this is in virtual reality applications. [12] In this case one wants to control which signals arrive at each of the listener's ears, completely independently. To do this, the loudspeaker on one side must both convey the signal intended for the ear on the same side, and also a cancellation signal to cancel the sound from the other speaker. This problem is solved by pre-filtering the loudspeaker signals with crosstalk cancellation (CTC) filters . [12]. For use in a virtual reality environment, these filters must be dynamically updated using both a database of head-related transfer functions (HRTFs) and a head-tracker.

It has already been mentioned that in order to reduce or cancel crosstalk, a filter has to be adjusted to model this crosstalk. One way to do this is to estimate the impulse response of the system using the sound from a musical instrument as the input signal. This may be desirable in other applications as well; for example, if one wants to perform an impulse response

measurement of a sound system immediately before or during a concert, a standard measurement signal like an MLS signal or a sinusoidal sweep will be annoying to the audience. Using a music signal, the measurement may be done without the audience noticing. A modification to the software measurement tool EASERA has been developed to do this. In this approach, the impulse response is estimated directly by transforming input and output signals to the frequency domain, performing elementwise division and transforming the result back to the time domain. Experiments show that this method is able to produce results comparable to those of MLS or sweep measurement signals if the measurement signals are bandpass filtered to remove content below 100 Hz and above 3-9 kHz. Music signals do not seem to have sufficient energy at the low and high end of the audible frequency range (approx. 20 Hz - 20000 Hz) to produce reliable results. So far, only recorded music has been used, but the goal of the developers is to be able to use actual instruments for real-time measurements. [1]

These examples illustrate that both adaptive filtering, acoustic crosstalk cancellation and the use of music signals to estimate impulse responses are all established as fields of study – but in this work, they will be combined in a way which has not been studied before (to the best of the author's knowledge).

Contents of the report

The remainder of this report starts with chapter 2 explaining the theoretical background of adaptive filtering - by introducing the concepts of reference, desired and error signals, the mean square error and its "performance surface", and a gradient search of this surface. The basic LMS (least-mean-squares) algorithm is then described, followed by descriptions of various modifications to this algorithm. The theoretical section concludes with explaining a couple of parameters used to measure the performance of an adaptive algorithm, and also mentioning how reverberation time in rooms and mode distribution in instruments may affect this performance.

Chapter 3 describes two different approaches to use of adaptive filtering for crosstalk cancellation: To first estimate the impulse response between two filters, and then keeping the filter coefficients constant (termed the "learn and freeze" method) – or to continually update the filter coefficients. Both methods have been investigated in this work, although most of the results are produced using the learn and freeze method.

In chapter 4, most of the practical aspects of the measurements are described: The measuring equipment and how it was used, and also what kind of rooms the measurements were done in. General descriptions of the calculations needed to produce the results are also included.

The results of the measurements are presented in chapter 5. These are mainly plots of the performance of the crosstalk cancellation for various instruments (given as damping in octave bands), together with plots of the effect spectra of the instruments. Measurements were done in two different rooms, and the results from these are divided into two different sections, together with measurements of the signal-to-noise ratios for these rooms. Several other measurements concerning the crosstalk cancellation are also presented; A comparison of two different adaptive algorithms, a study of the influence of microphone directivity and type, perceptual effects of different filter lengths, a test of the minimum phase criterion for a measured impulse response, and the results of various simulated experiments. The chapter concludes with descriptions of several sound examples which are presented together with the report.

The results are discussed in chapter 6, and the conclusions are summed up in chapter 7. Relevant MATLAB code has been put in appendix A.

2
Theory

Adaptive filtering techniques have found their use in several areas. As was mentioned in the introduction, they are often used to reduce crosstalk levels in telephony, but they are also used in for example control systems (an example of this is the Kalman filter).

In 1959, Bernard Widrow introduced the least-mean-square (LMS) algorithm. This has become one of the most widely used and well known of the adaptive algorithms, mostly because of its simplicity and robustness[14]. In time, several modifications and additions were made to the original algorithm, yielding algorithms which were more computationally effective or better suited to certain needs. LMS-style algorithms are still the most common in applications where adaptive filtering is used.

In this chapter, we will first introduce the signals and notation involved in a simple use of adaptive filtering. Then, the concept of the mean square error, its corresponding "error surface", and the gradient search of this surface is described, before the basic LMS algorithm is introduced. Following this are descriptions of a couple of modified LMS algorithms. Then, the concept of system distance is explained, together with the "damping parameter" ERLE (Echo Return Loss Enhancement). Both of these are parameters which describe the adaptive filter's performance. Finally, a note is included on how the distribution of modes in a musical instrument may affect the adaptive algorithm.

2.1 Signals: Reference, desired and error

2.1.1 Overview

In figure 2.1 we see a schematic of the signals involved in a simple use of adaptive filtering. A sound source emits sound into the room, and the sound is recorded by two microphones in different positions. One microphone is placed quite close to the sound source, the other further away. The sound that reaches the microphones is a sum of direct sound from the source and

various reflections from the walls. The signal from the close microphone, $x(n)$, is called the reference signal. This microphone will be referred to as the reference microphone. $x(n)$ is filtered with the filter $w(n)$, which outputs the signal $\hat{d}(n)$. This signal is an estimate of the "desired" signal $d(n)$ coming from the other microphone. This microphone is called the room microphone. $\hat{d}(n)$ is subtracted (sample by sample) from $d(n)$, resulting in the error signal e(n). Both $x(n)$ and $e(n)$ are input to an adaptive algorithm which continually updates the filter coefficients. The adaptive algorithm will try to make the estimated signal $\hat{d}(n)$ as similar as possible to $d(n)$ by adjusting the filter weights to minimize the error $e(n)$.

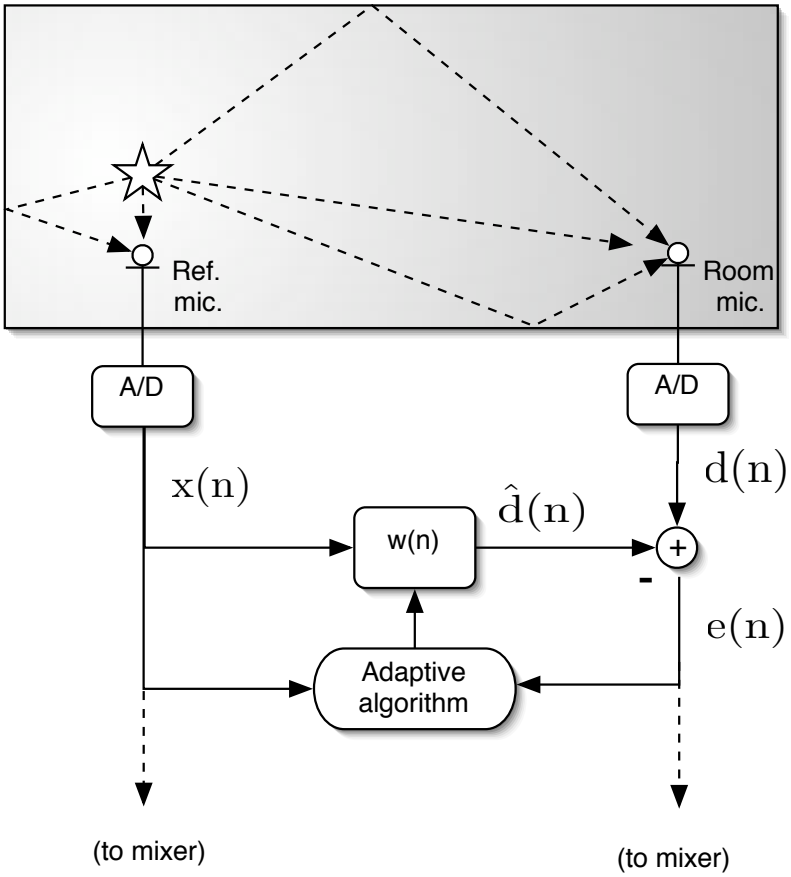


Figure 2.1: Overview of the signals involved in adaptive filtering used for crosstalk cancellation.

2.1.2 Notation

The notation chosen in this report is as similar as possible to the notation found in most of the literature on adaptive filtering:

$\mathbf{x}(n)$ is a column vector consisting of the N last input samples, with the most recent sample as the first element:

$$\mathbf{x}(n) = [x(n) \quad x(n-1) \ \ldots \ x(n-(N-1))]^T \tag{2.1}$$

We will assume that the filter is causal and FIR, meaning that the output is a linear combination of current and past input samples. $\mathbf{w}(n)$ is a length N column vector of the filter weights at sample time n, with the first element being the weight for the most recent sample:

$$\mathbf{w}(n) = [w_0(n) \quad w_1(n) \ \ldots \ w_{N-1}(n)]^T \tag{2.2}$$

Thus, the output $\hat{d}(n)$ from the filter can be written both in the usual "convolution"-style way,

$$\hat{d}(n) = \sum_{l=0}^{N-1} w_l(n) \cdot x(n-l) \tag{2.3}$$

or as a matrix product,

$$\hat{d}(n) = \mathbf{x}^T(n)\mathbf{w}(n) \tag{2.4}$$

We will use the latter notation.

2.2 Mean square error

The error signal $e(n)$ is given by

$$e(n) = d(n) - \hat{d}(n) \tag{2.5}$$

and the *mean square error* is defined as

$$\xi = E[e^2(n)] \tag{2.6}$$

In many applications, the adaptive filter is adjusted to minimize this mean square error, which is the same as minimizing the average power of the error signal [21, chapter 2]. By inserting equations 2.4 and 2.5 into 2.6 we get

$$\xi = E\left[\left(d(n) - \mathbf{x}^T(n)\mathbf{w}(n)\right)^2\right] \tag{2.7}$$

2.2.1 Performance surface

We see that ξ is a quadratic function of the filter weights. Thus, ξ is a surface in N-dimensional space with only one minimum, in the same way that a function $f(x) = ax^2 + bx + c$ also has only one minimum. This minimum is defined by the *optimal* filter weights $\mathbf{w}_0$. If both $\mathbf{x}$ and d are stationary, these weights are found to be (the Wiener-Hopf equation) [21]

$$\mathbf{w}_0 = \mathbf{R}^{-1}\mathbf{P} \tag{2.8}$$

where $\mathbf{R}$ is the autocorrelation matrix of the input signal,

$$\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)] \tag{2.9}$$

and $\mathbf{P}$ is the correlation between the input vector $\mathbf{x}(n)$ and the current sample of the desired signal, $d(n)$.

$$\mathbf{P} = E[d(n)\mathbf{x}(n)] \tag{2.10}$$

2.2.2 Gradient search

In order to find $\mathbf{w}_0$ from some arbitrary starting point, we need an algorithm that will update the filter weights in the direction of $\mathbf{w}_0$, on average. One method for doing so is the "method of steepest descent". As an analogy, we can picture a landscape with a minimum - a valley - that we want to get to. Unfortunately there is zero visibility because of fog. One way of finding the bottom is to just start walking downhill, in the direction of the steepest descent, one step at a time. This will not necessarily bring us to the bottom along the shortest possible path, but we can be sure to get there eventually.

We usually do not know the performance surface of ξ, (it is "covered in fog", to follow the analogy), so in the same way we try to update the filter weights in the direction of steepest descent. We know from calculus [22] that the gradient $\nabla\xi$ at any given point on the surface is a vector which points in the direction of the steepest *ascent*, so naturally we do the update in the direction of the negative gradient, $-\nabla\xi$.

The gradient is defined as

$$\nabla\xi = \frac{\partial\xi}{\partial\mathbf{w}} = \begin{bmatrix} \frac{\partial\xi}{\partial w_0} & \frac{\partial\xi}{\partial w_1} & \cdots & \frac{\partial\xi}{\partial w_{N-1}} \end{bmatrix}^T \tag{2.11}$$

and for each update step, we update the filters with

$$\mathbf{w} = \mathbf{w} - \tilde{\mu} \cdot \nabla\xi(n) \tag{2.12}$$

where $\tilde{\mu}$ is the "step size".

2.2.3 Eigenvalue spread of the correlation matrix and its effect on convergence rate

The nature and speed of the convergence of a gradient search is determined by the specifics of the algorithm and the shape of the performance surface. The latter is again determined by the correlation matrix $\mathbf{R}$ of the input signal (mentioned in section 2.2.1). The rate of convergence is seen to be dependent on the condition number or "eigenvalue spread" of this matrix [14, chapter 10.3].

The condition number $\mathcal{X}(\mathbf{R})$ of the matrix $\mathbf{R}$ is defined as the ratio of the largest eigenvalue of the matrix to the smallest,

$$\mathcal{X}(\mathbf{R}) = \frac{\lambda_{max,\mathbf{R}}}{\lambda_{min,\mathbf{R}}} \tag{2.13}$$

The condition number is a measure of the eigenvalue spread, that is, what kind of magnitude range the eigenvalues span. It can be shown that the convergence rate is approximately inversely proportional to the eigenvalue spread. This means that the larger the eigenvalue spread is, the longer it will take for a gradient search algorithm to converge. The individual convergence rates of each filter weight w_i are also inversely proportional to the corresponding eigenvalue λ_i. Thus, the algorithm will find filter weights corresponding to large eigenvalues relatively fast, and then use a long time to estimate the weights corresponding to smaller eigenvalues [14, chapter 10.3].

Since each sample of white noise is uncorrelated, the corresponding correlation matrix is diagonal, with each entry equal to the power of the signal. This makes all the eigenvalues equal, and the eigenvalue spread is 1, the minimum value. Thus, white noise is the ideal input signal for a steepest descent algorithm. For other, more correlated inputs, the eigenvalue spread is larger, and the convergence rate may be much slower, depending on the initial conditions [7].

2.3 The LMS algorithm

2.3.1 The basic algorithm

As we have seen, one way of finding the minimum squared error is to follow the gradient down in steps into the "bottom of the bowl" of the error surface

$\xi = E[e^2]$. An expression for ξ is usually not available, and it has to be estimated from the data at hand. The LMS (Least Mean Square) algorithm simply sets the estimate of the mean square error equal to the instantaneous error [21]

$$\hat{\xi}(n) = e^2(n) = \left(d(n) - \mathbf{x}(n)^T \hat{\mathbf{w}}\right)^2 \tag{2.14}$$

which gives the gradient estimate

$$\begin{aligned} \hat{\nabla}(n) &= \left[\frac{\partial e^2(n)}{\partial w_0} \quad \frac{\partial e^2(n)}{\partial w_1} \quad \cdots \quad \frac{\partial e^2(n)}{\partial w_{N-1}}\right]^T \\ &= 2 \cdot e(n) \cdot \left[\frac{\partial e(n)}{\partial w_0} \quad \frac{\partial e(n)}{\partial w_1} \quad \cdots \quad \frac{\partial e(n)}{\partial w_{N-1}}\right]^T \\ &= -2 \cdot e(n) \cdot \mathbf{x}(n) \end{aligned} \tag{2.15}$$

where in the last transition we have used that

$$\frac{\partial e(n)}{\partial w_k} = \frac{\partial}{\partial w_k}\left(d(n) - \sum_{l=0}^{N-1} w_l(n) \cdot x(n-l)\right) = -x(n-k) \tag{2.16}$$

The gradient filter update equation becomes

$$\begin{aligned} \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) - \tilde{\mu} \cdot \hat{\nabla}(n) \\ &= \hat{\mathbf{w}}(n) + \mu \cdot e(n) \cdot \mathbf{x}(n) \end{aligned} \tag{2.17}$$

where the "step size" $\mu = 2\tilde{\mu}$ is a gain constant that regulates the speed of adaptation. It should be noted that the estimate of ξ in equation 2.14 is not very accurate, and the algorithm will suffer from substantial "gradient noise". Still, the filter update equation has a low-pass effect on this noise (with cut-off determined by μ) [21, page 100], and on average the filter weights are updated towards the optimum value $\mathbf{w}_0$. The LMS algorithm is quite simple to implement, and has had widespread use because of this.

2.3.2 Stability and the step size parameter μ

The filter update equation 2.17 represents a feedback loop with potential instability problems. In [7, section 9.4] it is shown that the LMS algorithm is "convergent in the mean square" ($\xi \to$ constant as $n \to \infty$) if

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{2.18}$$

where λ_{max} is the largest eigenvalue of the correlation matrix $\mathbf{R}$. Since $\mathbf{R}$ is not available, the tap-input power $\|\mathbf{x}\|^2$ is suggested as a conservative estimate for λ_{max}. Thus the algorithm is expected to converge if μ satisfies

$$0 < \mu < \frac{2}{\|\mathbf{x}\|^2} \tag{2.19}$$

2.3.3 Normalized LMS

We can see from the LMS filter update equation, equation 2.17, that the rate of adaptation will vary with the strength of the input signal $\mathbf{x}(n)$. To assure an approximately constant rate of adaptation, the gradient estimate can be normalized with the tap-input power [7, section 9.12]. The update equation becomes

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\mu \cdot e(n) \cdot \mathbf{x}(n)}{\|\mathbf{x}(n)\|^2} \tag{2.20}$$

This can be seen as a normalization of μ. Whereas μ in the LMS algorithm has the dimensions of inverse power, the normalization makes μ_{norm} dimensionless.

$$\mu_{norm} = \frac{\mu}{\|\mathbf{x}(n)\|^2} \tag{2.21}$$

The normalization makes it a lot easier to pick a step size for which the filter is stable, since equation 2.19 becomes

$$0 < \mu_{norm} < 2 \tag{2.22}$$

This refined algorithm is called the Normalized LMS algorithm (NLMS). The normalization makes it more suitable than LMS for use where the input signal power is unknown and/or time-varying.

2.3.4 Block LMS

The Block LMS algorithm ("BLMS") is similar to the LMS algorithm, but operates on blocks of data of length L. Thus, L samples of the error signal e is calculated with the filter weights held fixed. The gradient is then estimated from an average over the L samples:

$$\hat{\nabla} = \frac{1}{L}\sum_{m=0}^{L-1}\mathbf{x}(kL+m)e(kL+m) \tag{2.23}$$

where k is the block index. Naturally, this averaging gives a gradient estimate which is more accurate than the "one-sample" estimate used in the basic LMS algorithm (equation 2.15). The update equation becomes

$$\begin{aligned}\hat{\mathbf{w}}(k+1) &= \hat{\mathbf{w}}(k) - \tilde{\mu}\cdot\hat{\nabla}(n) \\ &= \hat{\mathbf{w}}(k) + \mu_B \cdot \left(\frac{1}{L}\sum_{m=0}^{L-1}\mathbf{x}(kL+m)e(kL+m)\right)\end{aligned} \tag{2.24}$$

where μ_B is the effective step size. It is shown in [5] that the limits for the effective step size in the block version of LMS are equal to the limits for the basic algorithm,

$$0 < \mu_B < \frac{2}{\lambda_{max}} \tag{2.25}$$

Using the same approximation for λ_{max} as we did before, and letting $\widetilde{\|\mathbf{x}\|}^2$ be the average tap-input power over the block, we get

$$0 < \mu_B < \frac{2}{\widetilde{\|\mathbf{x}\|}^2} \tag{2.26}$$

Even if the upper bound of the step size is equal for LMS and BLMS, BLMS has the disadvantage of only being able to update the filter weights once per L samples. This means that even though BLMS has the advantage of a less noisy gradient estimate, LMS may converge faster. Since the BLMS algorithm only updates the filter weights once for every block, it uses a slightly smaller number of mathematical operations than LMS. The BLMS algorithm is also easier to parallelize, thus making fast implementations on parallel processors possible [5]. But as we shall see in the next section, the BLMS can also become very computationally efficient with a transformation to the frequency domain.

In [5], Clark argues that the filter length N and the block length L should be the same. This recommendation has been followed in this work.

2.3.5 FBLMS: BLMS in the frequency domain

In a very useful article [19], Shynk presents an overview of several ways to implement the BLMS algorithm in the frequency domain. Such algorithms are

termed Frequency Block LMS algorithms (FBLMS). We will describe one of these here - the "linear-convolution overlap-save method".

All frequency-domain vectors are denoted with large caps, and time-domain vectors with small caps. k is the block index. Let $\mathbf{X}(k)$ be the Fourier transform of the $2L$ last input samples (the current block and the most recent block):

$$\mathbf{X}(k) = \text{FFT}\left[x(n-L) \ \ldots \ x(n-1) \quad x(n) \ \ldots \ x(n+(L-1))\right]^T \tag{2.27}$$

$\hat{\mathbf{W}}(k)$ is the zero-padded Fourier transform of the filter weights:

$$\hat{\mathbf{W}}(k) = \text{FFT}\begin{bmatrix}\hat{\mathbf{w}}(k) \\ \mathbf{0}\end{bmatrix} \tag{2.28}$$

where $\mathbf{0}$ is a column vector of length L, so that $W(k)$ also has a length of $2L$. The filtering operation in performed in the frequency domain,

$$\mathbf{D_{est}}(k) = \mathbf{X}(k) \otimes \hat{\mathbf{W}}(k) \tag{2.29}$$

where $\otimes$ denotes an elementwise multiplication in this case. The zero padding of the filter weights is necessary to avoid aliasing in $\mathbf{D_{est}}$. The L output samples for the current block in the time domain are found as the last L elements of

$$\hat{\mathbf{d}}(k) = \text{IFFT}[\mathbf{D_{est}}(k)] \tag{2.30}$$

The error is then calculated in the time domain,

$$\mathbf{e}(k) = \mathbf{d}(k) - \hat{\mathbf{d}}(k) \tag{2.31}$$

and transformed to the frequency domain

$$\mathbf{E}(k) = \text{FFT}\begin{bmatrix}\mathbf{0} \\ \mathbf{e}(k)\end{bmatrix} \tag{2.32}$$

The gradient estimate for a block is equal to the correlation between the input signal and the error signal, as we saw in equation 2.23. Correlation can be seen as a "reversed" convolution, and is also possible to perform in the frequency domain. The gradient estimate is found as

$$\nabla(k) = \text{first L elements of IFFT}[\mathbf{X}^*(k) \otimes \mathbf{E}(k)] \tag{2.33}$$

where $*$ denotes complex conjugate. The filter update equation is

$$\hat{\mathbf{W}}(k+1) = \hat{\mathbf{W}}(k) + \mu \cdot \text{FFT} \begin{bmatrix} \nabla(k) \\ \mathbf{0} \end{bmatrix} \tag{2.34}$$

It should be noted that this implementation of FBLMS is *completely equivalent* to BLMS - the same filter weight updates are done for each block, and the same error signal is produced. The great advantage of FBLMS is that the filtering operation, which is done with convolution in the time domain in BLMS, can be performed as elementwise multiplication in the frequency domain. The same is the case for the correlation between the input and the error signal. Even though the Fourier transformations represent some overhead, the total number of mathematical operations can be greatly reduced when the filter lengths are relatively large. It is shown in [7, section 10.2] that for an adaptive filter with N weights, the complexity ratio CR (ratio of real multiplications needed) between the FBLMS algorithm and the (non-block, time domain) LMS algorithm is

$$CR = \frac{5 \cdot \log_2 N + 13}{N} \tag{2.35}$$

As an example, let us say that the filter is operating at a sampling frequency of 12 kHz and has to model at least 0.3 seconds of the room impulse response. This gives a filter length of 3600, which we round up to $2^{12} = 4096$ (the FFT algorithm is most efficient for vectors with length equal to a power of 2). The CR in this case is about 0.018, meaning that FBLMS is approximately 56 times faster than LMS. In practice, other factors like memory capacity etc. will also affect the performance of the algorithm, but FBLMS is still generally faster than LMS when the number of filter weights is high [19].

In the next section, a modification to the FBLMS algorithm is described, where the step size is controlled independently for each frequency bin. Note that when the abbreviation FBLMS is used in the remainder of this report, it is the modified algorithm that is meant, not the one described in this section.

2.3.6 The RLS algorithm and its FBLMS approximation

When the input signal is non-white and non-stationary (as is often the case with music signals), the convergence rate of LMS algorithms is generally not as good as for white stationary signals. This is due to the signal being correlated, resulting in a large eigenvalue spread of the correlation matrix (see section 2.2.3).

The RLS algorithm (Recursive Least-Squares) [8] is a modification to the LMS algorithm, which solves this problem by decorrelating (or "whitening") the input signal. The decorrelation is done by multiplying the input signal with an estimate of the inverse of its own autocorrelation matrix;

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu \cdot \hat{\mathbf{S}}_{xx}^{-1} \cdot \mathbf{x}(n) \cdot e(n) \tag{2.36}$$

where $\hat{\mathbf{S}}_{xx}$ is the estimate of the autocorrelation matrix. The estimation and inversion of this matrix and the subsequent matrix multiplication in the update equation makes this approach very computationally demanding. The algorithm also has considerable stability problems. Nevertheless, its performance is generally much better than that of the LMS algorithm for correlated input signals [8].

In [18] it is argued that the pre-whitening performed by the RLS algorithm is approximately equivalent to normalizing each frequency component of $\mathbf{x}(n)$ with an estimate of its own power. Using this approximation, the FBLMS algorithm can be modified:

$$\begin{aligned} \nabla(k) &= \text{first L elements of IFFT} \left[\frac{\mathbf{X}^*(k) \otimes \mathbf{E}(k)}{\mathbf{P}(k)} \right] \\ \hat{\mathbf{W}}(k+1) &= \hat{\mathbf{W}}(k) + \mu \cdot \text{FFT} \begin{bmatrix} \nabla(k) \\ \mathbf{0} \end{bmatrix} \end{aligned} \tag{2.37}$$

Here, the frequency representations of the input and error signal are normalized with an estimate of the power in each frequency bin, $\mathbf{P}(k)$. This operation is an approximation to the RLS multiplication with the inverse of the estimated autocorrelation matrix in equation 2.36. One can also see this as introducing a frequency-dependent normalization of the step size [7, section 10.2];

$$\mu_i = \frac{\mu}{\hat{P}_i} \tag{2.38}$$

where μ_i is the step size for frequency bin i, and P_i is an estimate of the power in the same frequency bin. The power spectrum can be estimated in many ways, but one suggested method [19] is a first-order recursion:

$$\hat{\mathbf{P}}(k+1) = \beta \cdot \hat{\mathbf{P}}(k) + (1-\beta) \cdot \|\mathbf{X}(k)\|^2 \tag{2.39}$$

where β is called the "forgetting factor". It should have a value reasonably close to 1 for the recursion to have a smoothing effect of the average power spectrum. This modified FBLMS algorithm is what will be referred to as "the FBLMS algorithm" for the remainder of this report.

2.4 System distance

The mismatch between the actual impulse response $\mathbf{w}(n)$ and the impulse response $\hat{\mathbf{w}}(n)$ estimated by an adaptive algorithm is termed the "mismatch vector" $\mathbf{\Delta w}(n)$ [8]:

$$\mathbf{\Delta w}(n) = \mathbf{w}(n) - \hat{\mathbf{w}}(n) \tag{2.40}$$

The "system distance" is defined as the squared norm of the mismatch vector, $\|\mathbf{\Delta w}(n)\|$ [8]

$$\|\mathbf{\Delta w}(n)\| = \mathbf{\Delta w}^T(n)\mathbf{\Delta w}(n) \tag{2.41}$$

2.5 The damping parameter ERLE

In order to compare the performance of different algorithms and settings, we use the so-called "Echo Return Loss Enhancement" (ERLE) [2] (the name is mostly used in telecommunications, where cancellation of echoes from the loudspeaker(s) is very important). If we use the same notation as in figure 2.1, with $d(n)$ the crosstalk signal and $\hat{d}(n)$ the estimated crosstalk signal, ERLE is defined as

$$\text{ERLE} = 10 \cdot \log_{10} \frac{\|d(n)\|^2}{\|d(n) - \hat{d}(n)\|^2} \tag{2.42}$$

We see that ERLE is a measure (in dB) of the difference in energy of the signal before and after subtraction of the modelled crosstalk. The better the crosstalk cancellation works, the higher ERLE will be. Note that this parameter can only be used for performance evaluation when there is only one sound source. If there are two sound sources (one per microphone, as there will be in the final application), the signals from the microphones will be a mixture of crosstalk and desired signals, and ERLE can not be used to determine how much of the crosstalk is cancelled.

2.6 ERLE and reverberation time

In theory, the impulse response of a room is of infinite length (although it will disappear in the noise floor after a finite time). Therefore, a FIR filter of finite length N can not model the impulse response perfectly. But what is the maximum ERLE for a certain filter length N? In [2], the authors show that if

one assumes a white noise input and and a perfect match of the N first filter weights,

$$\hat{w}_l = w_l, \text{ l} = 0,1, \ldots \text{ , N-1} \tag{2.43}$$

the maximum ERLE can be expressed as

$$\text{ERLE}_{max}(N) = 10 \log_{10} \frac{\sum_{l=0}^{\infty} w_l^2}{\sum_{l=0}^{N} w_l^2} \tag{2.44}$$

If one also assumes an exponential decay of the room impulse response,

$$E\left[|w_l|\right] = e^{-al} \tag{2.45}$$

equation 2.44 simplifies to

$$\text{ERLE}_{max}(N) = 10 \log_{10} \left(e^{2aN}\right) = \frac{10}{\ln 10} \cdot aN \tag{2.46}$$

The reverberation time T_{60} is defined as the time it takes for the sound in a room to decay by 60 dB. Still assuming that the impulse response decays exponentially, the parameter a is related to T_{60} and the sampling frequency F_s:

$$20 \log_{10} \left(e^{-a \cdot T_{60} \cdot F_s}\right) = -60\text{dB} \tag{2.47}$$

which we rearrange to

$$a = \frac{3 \cdot \ln 10}{T_{60} \cdot F_s} \tag{2.48}$$

If we put this into equation 2.46, we get

$$\text{ERLE}_{max}(N) = 60 \cdot \frac{N}{T_{60} \cdot F_s} \tag{2.49}$$

Note that ERLE_{max} is a linear function of the filter length N. As an example, we can assume that a room has a reverberation time $T_{60} = 0.5s$, and that we use a sampling frequency of 44.1 kHz, which is common for music signals. With this sampling frequency, the entire audible range (up to approximately 20 kHz) is covered. With a filter length of 4096, the maximum achievable ERLE is about 11 dB. This shows that for even with a moderate reverberation time, a very high number of filter weights is needed for music signals.

2.7 Random mode distribution

It is well known that in the steady state after an excitation, the displacement of a membrane or plate will be dominated by "modes". These are the result of standing waves. In a circular membrane, the distribution of these modes are a function of both distance from the center, r, and angle, θ. The solutions for the displacement of the (m,n) modes in a circular membrane with a fixed rim are given as [10]

$$\begin{aligned} \mathbf{y}_{mn}(r,\theta,t) &= \mathbf{A}_{mn} J_m(k_{mn} r)\cos(m\theta + \gamma_{mn}) e^{j\omega_{mn} t} \\ k_{mn} a &= j_{mn} \end{aligned} \tag{2.50}$$

where J_m is a Bessel function of the first kind, of order m. a is the radius of the membrane, and j_{mn} is the n'th zero of the Bessel function.

The angular dependence of certain modes poses a problem when trying to describe for example a drum. The drum head is a circular membrane with a fixed rim, so modes like those described above will dominate the sound radiation. Using one reference microphone close to the drum, and one room microphone further away, one might hope to estimate a "mic-to-mic" impulse response to cancel the crosstalk to the room microphone. But unfortunately, the angular distribution (given by γ) of the modes in the drum head is dependent on the initial conditions – that is, how the drum is struck. This dependence makes the directivity of the drum time-dependent, and thus there is probably no single mic-to-mic impulse response which will work for all cases. Figure 2.2 illustrates this for one possible mode of the drum head, with two different values for γ. Considering this, one can not expect crosstalk cancellation to work perfectly for instruments where the mode distribution may change. But in the case of a drum, it should work well for those modes that have a rotational symmetry (no angular dependence).

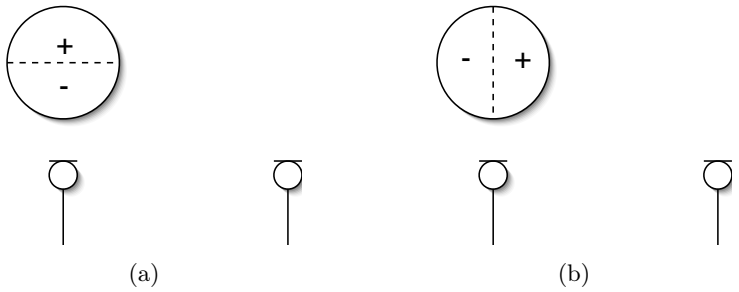


Figure 2.2: Two possible distributions of the (1,1) mode in a drum head. Only the node line has been drawn.

3

Methods utilizing adaptive filters for acoustic crosstalk cancellation

In this section, two different uses of adaptive filtering for crosstalk cancellation are described. The first one is nicknamed "learn and freeze" (section 3.1), with the name referring to an approach of first estimating a crosstalk cancellation filter, and then keeping the filter coefficients constant. A theoretical analysis is done regarding the validity of an impulse response between a reference and a "room" microphone, and an explanation is given as to why an adaptive filtering approach is preferred rather than direct estimation of the impulse response through Fourier transformations. The second method which is described (section 3.2) is based on using adaptive filtering to continually update the filter coefficients. This method is more flexible than the learn and freeze method, but it also faces several challenges that will need to be handled in a practical implementation. Some of these are discussed here. These challenges are also part of the reason why the continuous update method has been given less attention than the learn and freeze method in this work.

3.1 Method: "Learn and freeze"

Figure 3.1 shows a simplified schematic of an asymmetric acoustic crosstalk situation. One instrument ("instrument 1") is very loud, and therefore there is a lot of crosstalk from this instrument into the microphone of another instrument ("instrument 2"). The problem of crosstalk from instrument 2 into the microphone of instrument 1 is negligible, making this an "asymmetric" case.

The learning phase of the learn and freeze method is shown in figure 3.2(a). Here, instrument 2 is silent while instrument 1 plays. The signal from microphone 1, $x(n)$, is filtered through an adaptive filter, and the output of the filter, $\hat{d}(n)$, is subtracted from the signal from microphone 2, $d(n)$. The resulting error signal, $e(n)$, is fed back to the adaptive filter, which tries to minimize this error.

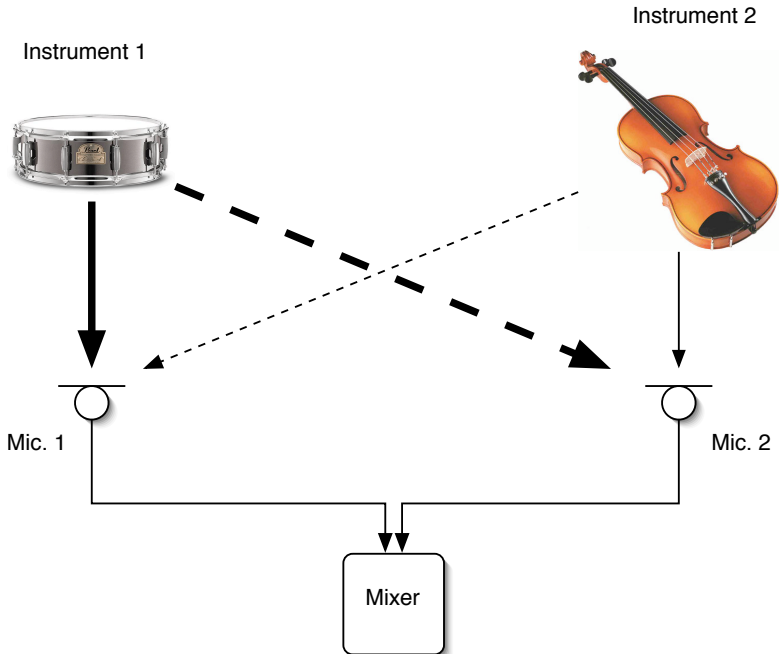


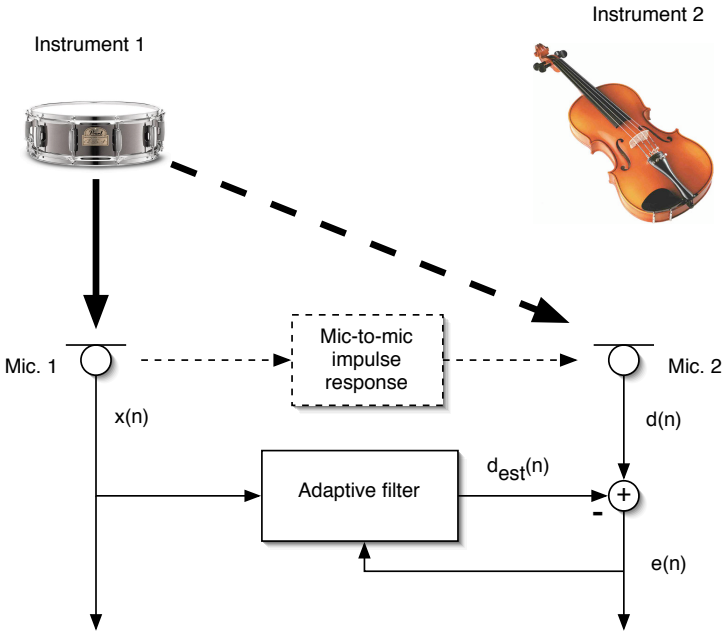
Figure 3.1: Illustration of "asymmetric" crosstalk, where one instrument is much louder than the other.

Although there is only one sound source, the signals coming from the two microphones will be different. This is due to many factors:

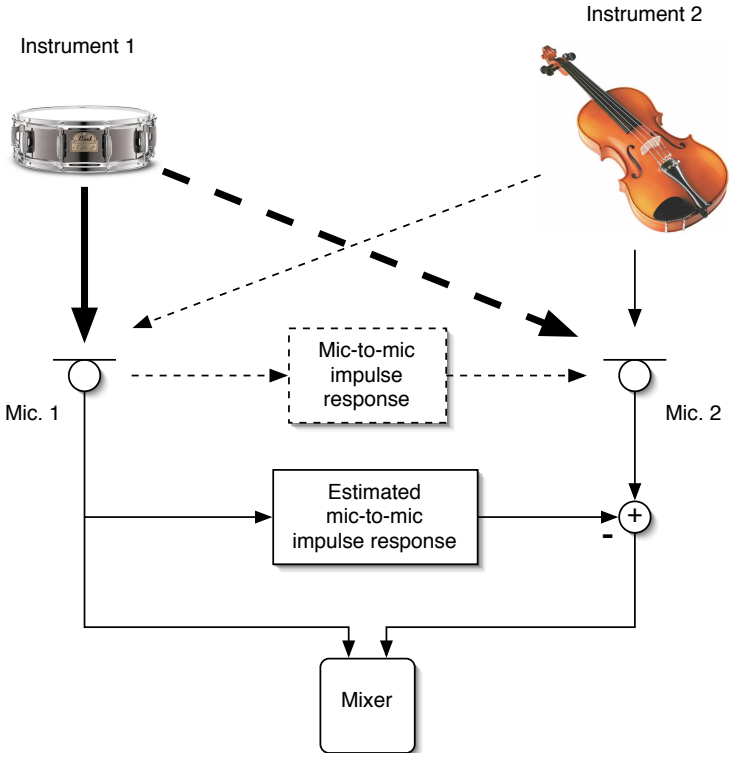
- A difference in distance from the sound source to the microphones will cause a difference in delay and amplitude of the direct sound
- Sound due to reflections from the walls in the room ("reverberation") will be different. If the room is not very reverberant, and instrument 1 is standing fairly close to microphone 1, we may assume that the signal from microphone 1 is dominated by the direct sound. The signal from microphone 2 will contain a mix of direct sound and reverberation.
- The microphones may themselves be different (two different models, or the same model with different settings), influencing how the sound field around the microphones is transformed to an electric signal

If the whole system that influences the microphone signals can be viewed as a linear, time-invariant system, it can be described through an impulse response $w(n)$ that accounts for all effects of the system. The signal $d(n)$ is then seen as a convolution between the input signal x(n) and this impulse response. In this work, this is nicknamed an "mic-to-mic" impulse response, since both the input and the output signal are produced by microphones in the room.

The adaptive filter is of a FIR design, so the filtering operation is equivalent to



(a) "Learning" phase of the filter.



(b) Operation after learning, with constant filter coefficients.

Figure 3.2: Illustration of "Learn and freeze" method.

convolution between the input signal $x(n)$ and the coefficients $\hat{w}(n)$ of the filter. Seeing this, it should be obvious that the filter coefficients should be as similar as possible to the mic-to-mic impulse response to produce the smallest possible error. The adaptive filtering is really a an approach to *system identification*, since the filter weights that produce the minimum error are equal to the impulse response that identifies the system.

If the adaptive algorithm is allowed to run until the estimate of the impulse response is deemed good enough, and then "freeze" the filter coefficients, the crosstalk from instrument 1 can be reduced using the setup shown in figure 3.2(b).

If the estimate of the mic-to-mic impulse response is close enough to the real impulse response, and this does not change significantly with time, this method is a simple and stable way to reduce crosstalk. The method will be referred to as the "learn and freeze" method in the remainder of this report.

Causality and stability of a mic-to-mic impulse response

Figure 3.3 shows a schematic of a sound source and two microphones. The impulse responses from this sound source are h_1 and h_2, denoted $H_1(\omega)$ and $H_2(\omega)$ in the frequency domain. A third impulse response, describing the transfer function between the signal from microphone 1 and microphone 2, is also plotted. It is this kind of response that the learn and freeze method tries to estimate.

In the frequency domain, this mic-to-mic transfer function $H_3(\omega)$ can be described in this way:

$$H_2(\omega) = H_1(\omega) \cdot H_3(\omega) \quad \Longrightarrow \quad H_3(\omega) = \frac{H_2(\omega)}{H_1(\omega)} \tag{3.1}$$

which in the time domain becomes

$$h_3(t) = h_2(t) \otimes h_{1,inv}(t) \tag{3.2}$$

where $\otimes$ denotes convolution and $h_{1,inv}$ is the inverse of the impulse response h_1. So, in order to calculate h_3, one must first invert h_1. In [16], it is stated that the only case in which the inverse of a room impulse response can be guaranteed to be causal and stable is where the original impulse response is minimum phase[1]. This again means that one can only be sure that a stable

[1]A digital filter is said to be minimum phase if all its zeros are inside the unit circle. Minimum phase systems have the useful property that the phase response of the system can be uniquely determined from the magnitude response [16].

and causal h_3 exists as long as h_1 is minimum phase. In [16], several different synthetic impulse responses are analyzed for the minimum phase property, and it is found that for rooms with walls of sufficiently low reflectivity (about 30-40 % in their experiments with a room of 4.5 x 3.8 x 2.4 meters, and a source-receiver distance of 1.8 meters), the impulse response is minimum phase. Although the authors could not say exactly which conditions where necessary for a room impulse response to be minimum phase, their results imply that the level of the direct sound has to be sufficiently high compared to the room reverberation.

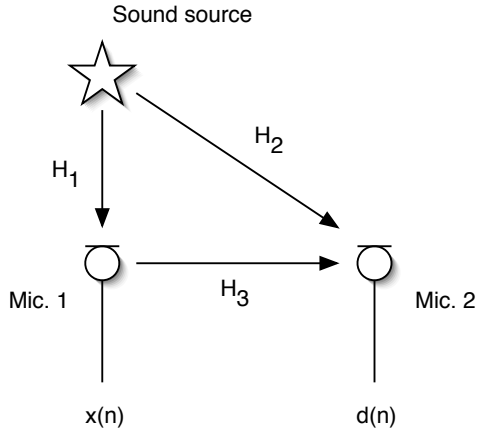


Figure 3.3: Schematic of a sound source with room impulse responses H_1 and H_2 to two different microphones, plus a "mic-to-mic" impulse response H_3.

This suggests that a perfect cancellation of the crosstalk through a mic-to-mic impulse response is only possible for rooms with short reverberation times and/or a very short distance between the sound source and the reference microphone. Although a perfect cancellation may not be possible for cases where these conditions are not fulfilled, the learn and freeze method may still be able to reduce the crosstalk to some degree.

Problems of direct impulse response estimation

Although the learn and freeze method provides one way to estimate the mic-to-mic impulse response, a more direct approach is also possible – but as we shall see, this approach has its limitations.

Still using the notation of figure 3.3, we assume that the room microphone signal d is a result of the reference signal x having been filtered through a system with an impulse response $h_3(t)$ and a corresponding frequency response $H_3(\omega)$

$$d(t) = x(t) \otimes h(t) \quad \Longrightarrow \quad D(\omega) = X(\omega) \cdot H(\omega) \tag{3.3}$$

where $\otimes$ denotes convolution. From this, the impulse response can be estimated directly:

$$\begin{aligned} H_3(\omega) &= \frac{D(\omega)}{X(\omega)} \\ h_3(t) &= \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{d(t)\}}{\mathcal{F}\{x(t)\}}\right\} \end{aligned} \tag{3.4}$$

where $\mathcal{F}$ denotes Fourier transformation. This approach is an effective way to estimate the impulse response, as long as one has access to two suitable signals $x(t)$ and $d(t)$. Such a direct estimation of the impulse response would be a computationally effective alternative to adaptive filtering for use with the learn and freeze method. However, a problem arises if the magnitude response of $x(t)$, $|X(\omega)|$, is very small (or zero) for some frequencies. Inverting the response will cause a magnitude response "blow-up" at these frequencies which is unwanted [11]. Such a magnitude response may both pose problems to other parts of the system (such as loudspeakers and amplifiers) and sound very unnatural to the human ear. Noise in the frequency ranges what have a low signal-to-noise ratio may also be greatly amplified.

The magnitude response $|X(\omega)|$ may have small values for certain frequencies because the sound source is not able to radiate sound efficiently at these frequencies. This will be a typical problem if a musical instrument is used as the sound source - the magnitude spectrum of such an instrument often has several peaks and dips, and does not cover the entire frequency range.

The problems described above are the reason why adaptive estimation of the impulse response has been chosen for this work, rather that the direct approach based on Fourier transformation. An adaptive algorithm will adjust the filter coefficients to produce the smallest possible error, regardless of whether the signal $x(t)$ covers the entire frequency range. Even if both $x(n)$ and $d(n)$ were sinusoids, the approach would still work, without problems occurring in the zero-energy frequency bands. In this way, the problem of the magnitude response "blow-up" can be circumvented.

3.2 Method: "Continuous update"

In a practical application, for example in a concert venue, the learn and freeze method has some obvious drawbacks: After the impulse response has been

estimated, the filter coefficients are held constant, and thus the filter can not adapt to changes in the actual impulse response. Such changes may be caused by several factors; people or objects moving in the room, temperature and humidity changes, and moving the microphones. Although such changes may not be fatal to the result, it may lessen the achieved damping of the crosstalk.

An alternative to the learn and freeze method is to continuously update the filter coefficients, to let the filter continuously adapt to changes in the system. This is equivalent to letting the "learning" phase of the learn and freeze method (see figure 3.2(a)) continue indefinitely. In this way, the adaptive filters' ability to "track" the changes in a system is fully exploited. This is also closer to the "usual" application of adaptive filters for crosstalk cancellation, in which the adaptive filter is constantly updated [8].

A schematic of the continuous update approach is shown in figure 3.4. The filter coefficients are constantly updated, to minimize the power of the error signal. If the estimation of the impulse response is good enough, crosstalk from instrument 1 will be cancelled, and this error signal will only contain the sound of instrument 2. In addition to being fed back into the adaptive algorithm, the error signal is also be sent to the mixer, representing instrument 2. A "doubletalk detector" is also utilized – this is discussed below.

In a system like a concert venue, where the system is constantly changing, the continuous update should be expected to be superior to the learn and freeze method. However, a practical implementation involves a set of serious challenges:

- The adaptive algorithm is based on the assumption that the signal $x(n)$ and $d(n)$ are correlated - the signal $x(n)$ is the "original" signal from the sound source, and $d(n)$ is assumed to be the same signal run through a system which the algorithm tries to identify. This works fine as long as only instrument 1 plays, but if instrument 2 also plays, the signal $d(n)$ will be a mixture of sound from both instruments. This is termed "double-talk" in the literature [3], usually meaning a situation where both parties having a telephone conversation are talking at the same time. Double-talk will interfere with the adaptation process and possibly cause it to diverge. To counteract this, the signals from microphones can be fed to a "double-talk detector", which will slow down or stop the adaptation process (by adjusting the step size parameter μ) when doubletalk is detected [3]. This situation is shown in figure 3.4. The design of a robust double-talk detector is not trivial. Also, while people having a telephone conversation usually speak in turns, people playing in a band more often than not play *at the same time* – double-talk is the norm, rather than an exception. This poses an extra challenge in an application for musical instruments.

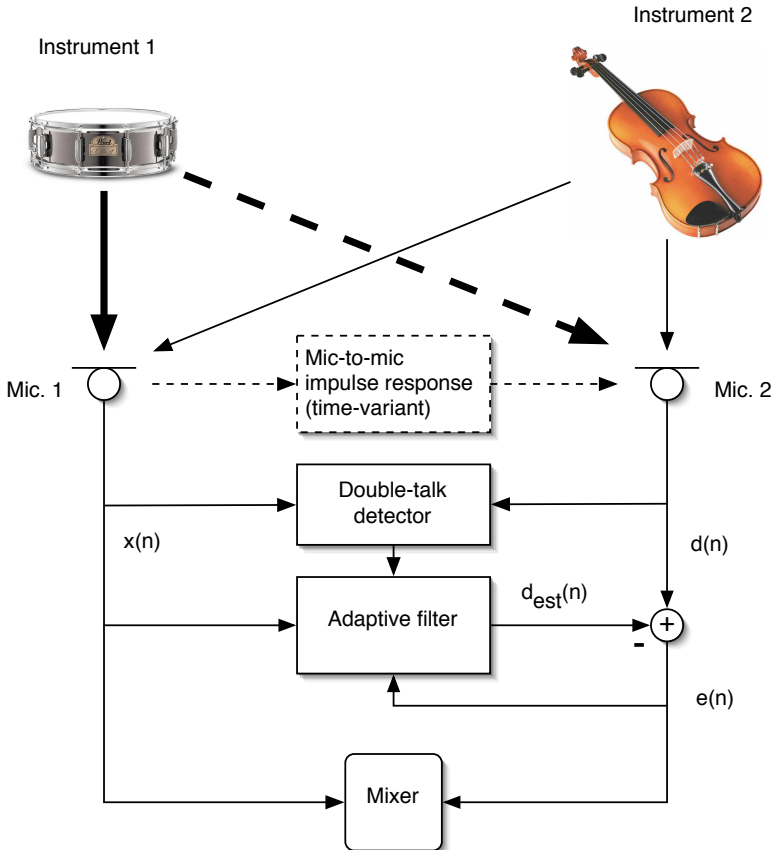


Figure 3.4: Schematic of the "Continuous update" method. Note the double-talk detector necessary to control the adaptation step size.

- The adaptation should also be slowed down or stopped when none of the instruments are playing, to avoid letting background noise misadjust the filter.
- While a "frozen" filter is linear and time independent, a running adaptive filter is both nonlinear and time dependent, since its characteristics constantly change as a function of the input signal. This makes its behavior a lot less predictable – it may, for example, create unwanted sound artifacts as a result of its coefficients changing.
- If a block-based implementation of an adaptive algorithm was to be used to reduce computational complexity (as described in sections 2.3.4 and 2.3.5), the signal from microphone 2 would be delayed for at least the time it takes to record a block for filtering. For a room impulse response, where a block length of several thousand coefficients (or some hundred milliseconds) may be necessary, this delay is probably far too long for a

real-time application like a live performance. The necessary delay may be reduced by using filter banks and performing adaptation in separate frequency bands, since it is possible to operate with a lower sampling frequency and thus also a shorter block length within each frequency band [8].

- When using the learn and freeze method, it is possible to first make a recording of sound from the two microphones, and then do the adaptation "off-line", before the actual performance. In this case, the algorithm may use as much time as it needs to do an adequate adaptation. Thus, the requirements for speed are not as great as for the continuous update method, since the latter will have to do the adaptation in real-time.

In addition to these challenges concerning implementation, the performance of the continuous update method is also not possible to measure using ERLE, as mentioned in section 2.5. The reason for this is that the residue signal, $d(n) - \hat{d}(n)$, contains both the residue crosstalk of instrument 1 and the recorded sound of instrument 2. Because of the contribution from instrument 2, the difference in energy before and after crosstalk cancellation is not a good measure of the actual reduction of crosstalk from instrument 1. This presented a problem in this work, as results from experiments should be based on quantifiable measures, and not only on subjective impressions. Because of this, most experiments were done using only one sound source, which is equivalent to using the learn and freeze method, rather than using the more realistic case of two sound sources, which is equivalent to the continuous update method. In this way, the ERLE values calculated for the experiments could be used as an objective measure of the achieved reduction of crosstalk.

3 METHODS UTILIZING ADAPTIVE FILTERS FOR ACOUSTIC CROSSTALK CANCELLATION

4

Description of experiments and equipment

In this chapter, the practical aspects of experiments and measurements will be described. Section 4.1 lists all hardware and software used in the measurements. The choice of adaptive algorithm for most experiments is also explained. Then, section 4.3 describes the small, damped room in which most experiments were done, and how the experiments here were conducted. Section 4.4 contains similar descriptions for experiments conducted in a larger room at the music conservatory in Tromsø. Finally, sections 4.5.8 and 4.6 describe the specifics of how ERLE and power spectra for the different instruments were calculated.

4.1 Sound recording and processing

Audio recording, signal processing, adaptive filtering, performance analysis and more or less everything else was performed on an Apple Powerbook G4 computer, with a 1.33 GHz processor and 768 MB of memory. The operating system used was OS X 10.5.2.

The microphones used for the measurements were manufactured by AKG, model C 414 B-XLS. These microphones are frequently used for professional sound recording, both on stage and in studio, and cost about 10000,- NOK. They have a large diaphragm (1 inch diameter), an adjustable high-pass filter and the possibility to change directivity. The high-pass filter was used on some recordings, to avoid low frequency background noise. The cutoff frequency was then set to 40 Hz.

A pair of Shure microphones were also used for comparison. These were SM 57 and SM 58 dynamic microphones. The SM 58 is the standard microphone for recording singing or speech anywhere in the world, while the SM 57 is usually used for recording instruments. Both are quite inexpensive, costing about 1500,- NOK.

An Dynaudio Acoustics AIR 15 active loudspeaker was used as a sound source in some experiments. Figure 4.1 shows the setup of both an AKG microphone and the loudspeaker during measurement of a reference signal.

Figure 4.1: AKG C 414 B-XLS microphone and Dynaudio Acoustics AIR 15 loudspeaker.

Preamplification and A/D conversion of the microphone signals was done using an Apogee Ensemble audio interface [25]. Digital audio was transferred to the computer with a sampling frequency of 44.1 kHz and a bit depth of 24 bits.

All audio was recorded using the freeware program Audacity, version 1.3.3 [24]. From here, audio was exported in *.wav format for further processing. All filtering and subsequent analysis was performed using MATLAB, version 7.2.0.283 (R2006a).

Filtering algorithms were implemented as MATLAB functions. The LMS algorithm was implemented as described by Widrow [21], and the NLMS and FBLMS algorithms was implemented as described by Haykin [7]. The MATLAB code for the FBLMS algorithm can be found in appendix A.

Although all the above algorithms were implemented and tested, the FBLMS algorithm was chosen as the main algorithm for use in experiments (see description in sections A and 2.3.6). The reason was mainly that this algorithm was found to be the only one computationally effective enough to produce results within a reasonable time (say, one adaptation for a sound clip of 30 seconds within a couple of minutes). A forgetting factor $\beta = 0.9$ was always used with the FBLMS algorithm.

4.2 Reference sound source: Loudspeaker playing white noise

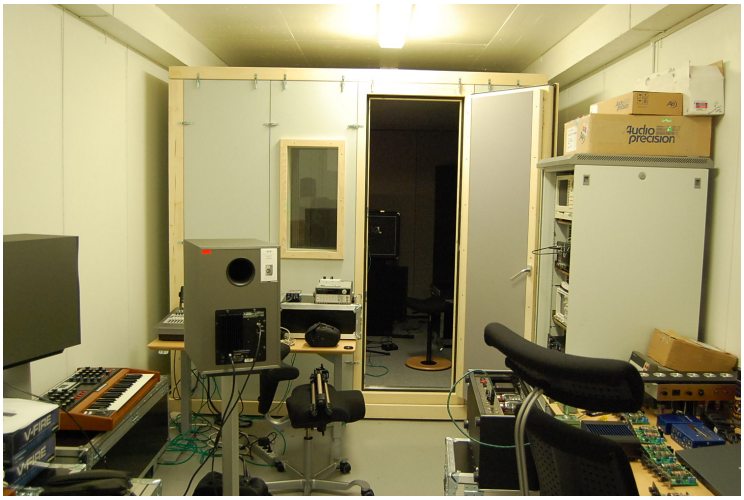
A loudspeaker playing white noise was chosen as a reference sound source for several measurements in this work. There were several reasons for this:

- Repeatablity: The exact same sound could be reproduced at the exact same level, for several different positions and measurements. This rules out any variation between results due to a change in the sound source.
- Immobility: While a hand-held musical instrument is very hard to keep immobile, the loudspeaker does not move at all during measurements. In this way, results are not influenced by movement of the sound source.
- Ideal signal: White noise is an uncorrelated, stationary signal, giving its correlation matrix a minimum eigenvalue spread (see section 2.2.3). This makes it an ideal input signal to an LMS-type algorithm.

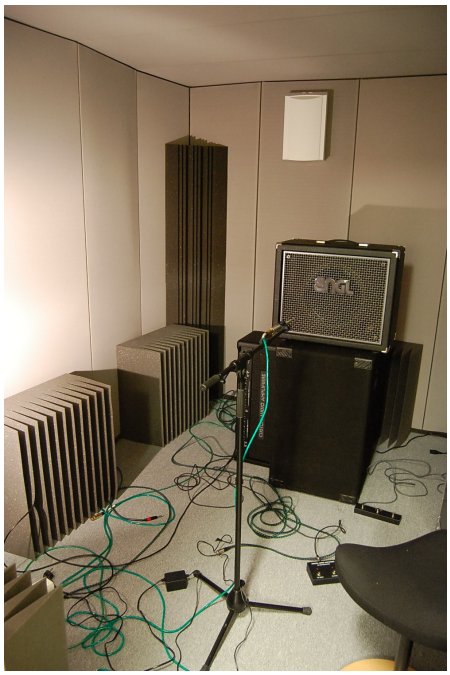
All these factors should make the loudspeaker playing white noise an ideal (or close to ideal) sound source. The white noise used in measurements was generated and played by the Audacity program. The gain of the loudspeaker's internal amplifier could be set digitally, and this was set to -10 dB for all measurements. During measurements, the loudspeaker was put on a stand which was approximately 120 cm tall.

4.3 Measurements in the acoustic booth

Most experiments were conducted in an "acoustic booth" – a small, heavily damped room – assembled at the institute of physics at the University of Tromsø. Two pictures of this booth are shown in figure 4.2. The acoustic booth was of a modular design, the "Premium" model produced by the German company Studiobox [26]. Figure 4.3 shows the reverberation time measurements supplied by Studiobox for a room similar to the one used, with inner dimensions 240 x 180 x 225 cm (length x width x height). The reverberation time is about 0,1 seconds above 250 Hz, and a little higher for the lowest frequencies. The inner dimensions of the acoustic booth that was used were 290 x 230 x 218 cm – a slightly bigger room than the one the measurements were done for. Due to the larger volume, the reverberation times of this room may be slightly longer, but they are probably still comparable to those supplied by the manufacturer.



(a) Outside. The acoustic booth is a "room inside a room". All measuring equipment is operated from here.



(b) Inside. The walls and ceiling are made of sound absorbents covered with a thin fabric. Each panel is slightly angled, to break up and deflect incoming sound waves. Additional sound absorbents were placed in the room to dampen it further.

Figure 4.2: Pictures of the acoustic booth used in many of the measurements. The pictures are only for illustration purposes - the equipment depicted was not used in this work.

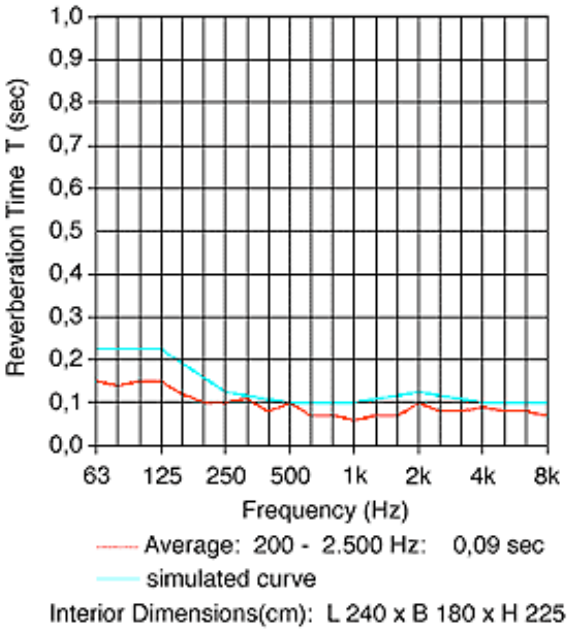


Figure 4.3: Reverberation times for the acoustic booth used for measurements ("Studiobox Premium").

4.3.1 Measurement setup

All measurements for experiments with adaptive filtering were done in the following way: A reference microphone was placed close to the sound source, and one or more room microphones were placed somewhere else in the room, further away from the source. A recording of the signal from each microphone was done while the sound source played. Adaptive filtering could then be performed on these recordings in MATLAB.

Most measurements in the acoustic booth were done for six different microphone positions. The sound source was placed in two different positions in the room, and the room microphone was placed in three different positions for each of these. The distance between the two microphones was held at 120 cm for each measurement. Because of the differences between the sound sources, it was not practically possible to place the microphones in exactly the same place for all measurements. The microphones were placed as similarly as possible to produce comparable measurements.

Recordings were done for six different sound sources: A loudspeaker playing white noise, an acoustic guitar, an electrically amplified guitar, a drum, a male person singing, and a trombone. The different sound sources were chosen on the basis of what was available, all the while trying to represent as different

kinds of sound sources as possible.

2 minute recordings were done in each position, these were each divided into six segments of 20 seconds, and adaptive filtering was performed on each segment. After various preliminary tests, segments of 20 seconds was deemed long enough for the error envelope of most signals to reach a constant value. ERLE values (see definition in section 2.5) for each segment were calculated in octave bands from 63 Hz to 8000 Hz. With six placements and six segments, there were 36 ERLE values in total for each octave band. The mean value and standard deviation were calculated for each band. The reason for using both different placements and several segments for each placement was to account for two different kinds of variation: The statistical variation of results for one given source-microphone placement, and the variation of results caused by different placements in the room. Hopefully, the mean values calculated from this are representative of the typical performance for the sound source in question.

The signal-to-noise ratio (SNR) at each position was also measured, by first recording 10 seconds of the background noise, and then recording 10 seconds the loudspeaker playing white noise. The two segments were filtered in octave bands, and the signal-to-noise ratio in each band was found as the difference in energy between the two segments. Finally, the mean value and standard deviation of the SNR in each band was calculated.

Figure 4.4 shows the two sound source positions and the six microphone positions in the acoustic booth in a "bird's eye" view. Although it can not be seen in the figures, the reference microphone and the room microphone were placed at a slightly different height (a difference of about 30 cm). This was done to avoid having both microphones in a plane parallel to the floor and the ceiling.

4.3.2 Synthetic impulse response

In order to simulate experiments in the acoustic booth, a synthetic impulse response, as similar as possible to a real mic-to-mic impulse response, was generated. It consisted of one perfect pulse accounting for the direct sound, and an exponentially decaying tail of random noise, representing reverberation. A MATLAB function made by Peter Svensson, *creexpir* (appendix A.2), was used to generate this. Figure 4.5 shows an example of this, with the synthetic and the real response plotted together.

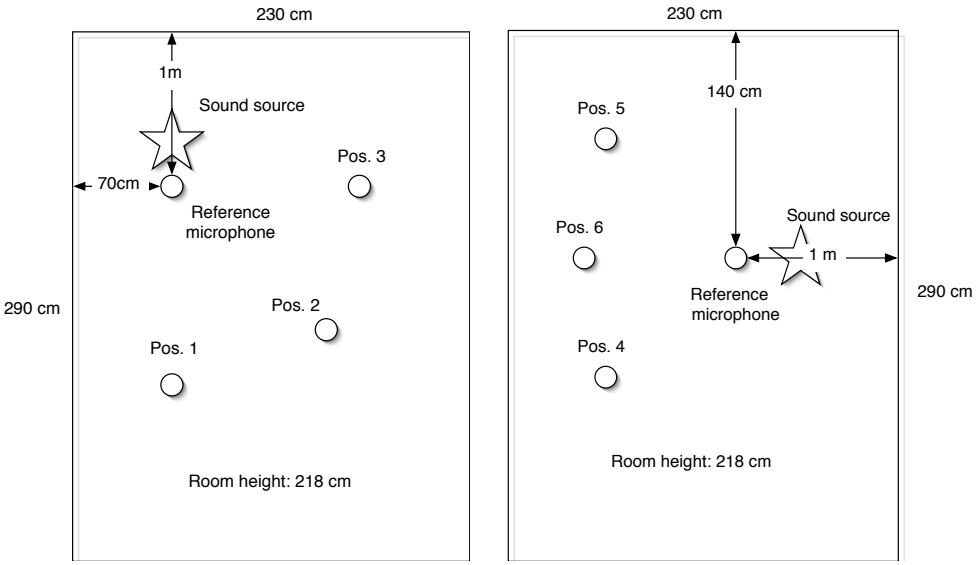


Figure 4.4: Approximate measurement positions in acoustic booth. The figure on the left shows the first sound source position and microphone positions 1-3, and the figure on the right shows the second source position and microphone positions 4-6.

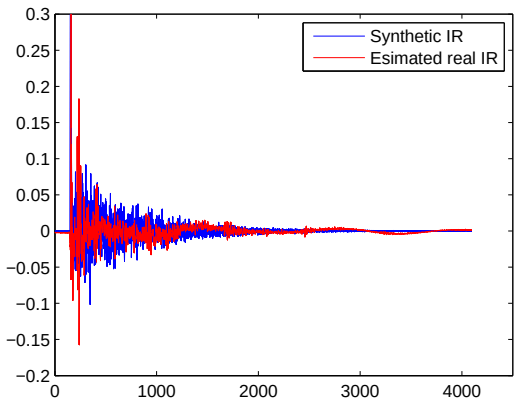


Figure 4.5: Synthetic impulse response, plotted together with an example of an estimated impulse response.

4.4 Recording at the Music Conservatory

A series of recordings of different instruments were done in a medium-sized ensemble room at the Music Conservatory at Tromsø University College. Several students at the conservatory volunteered to be recorded playing their instrument in this room. The room measured 8 x 6.5 meters. The height was not measured, but was estimated to about 6 meters, yielding a total internal volume of about 310 m^3. One of the walls was partly covered with an absorbent, which may explain a reverberation time which was perceived as fairly short for the room's volume.

Four microphones were set up in the room, as seen in figure 4.6. The center microphone was used as the reference microphone, assigned to record the direct sound of the instrument. The exact placement of this microphone had to be adjusted to each player. Three other microphones, used as room microphones, were placed around the room, each at a distance of 2 meters from the reference microphone. These were all set 1.5 meters above the floor. A cardioid directivity was used on all microphones. No high-pass filter was used.

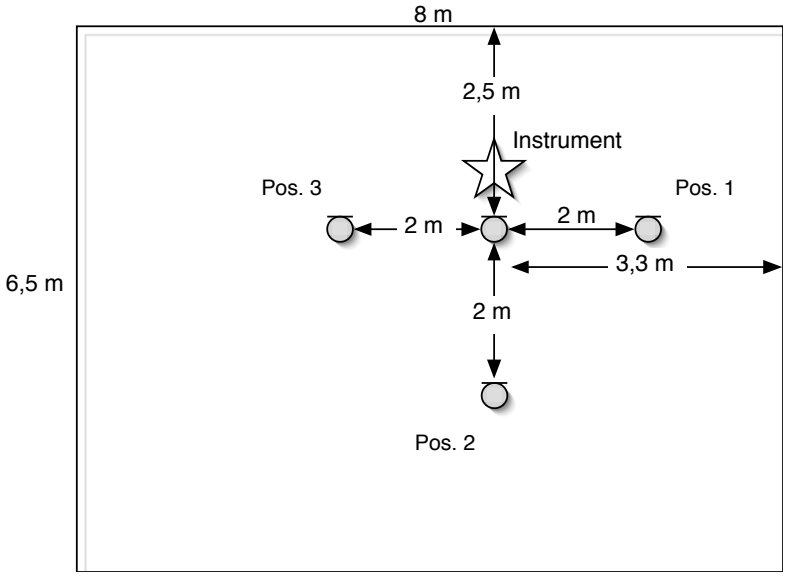


Figure 4.6: Schematic of instrument and microphone placements used for recordings at the conservatory.

Figure 4.7 shows a picture of the room with all four microphones set up. Figure 4.8 shows one of the musicians standing by the reference microphone, playing the clarinet.

The musicians were all asked to play two segments, each two minutes long. The first one should be "calm", with long tones. The second one should be faster,

Figure 4.7: Picture of the microphone set-up in the ensemble room at the conservatory.

Figure 4.8: Student playing clarinet.

with shorter, more staccato tones. The reason for making two such different recordings was to study whether the playing style would affect the adaptive algorithm's performance. Some of the musicians had prepared two different pieces of music, while others improvised or played scales. The instruments that were recorded were: "Classical" (nylon string) guitar, double bass, violin, clarinet and bassoon.

In addition to musical instruments, recordings were also done of the reference sound source, the loudspeaker playing white noise. These recordings were done in the same way as for the musical instruments, with one exception: Recordings were done for two different distances between the reference microphone and the room microphones; 1.2 meters and 2 meters. This was done to make the measurements comparable to both those done in the acoustic booth (where the distance was 1.2 meters), and the other measurements done in the ensemble room (for which the distance was 2 meters, as already mentioned).

An approximate measurement of the reverberation time in the ensemble room was also done, using the measurements of the loudspeaker playing white noise. Using the original signal sent to the loudspeaker, together with the 3 recordings from the room microphones, the impulse responses between the loudspeaker and these microphones were estimated. This was done by Fourier transformation of the input and output signals, elementwise division, and inverse Fourier transformation (the "direct" method mentioned in section 3.1). The Schröder curve (backwards integration of energy) for each of these impulse responses was calculated, and by a fitting a straight line to these curves, the reverberation time[1] of the room could be estimated, as described in [4].

Because of the longer reverberation time in this room, it was found that both a longer filter and a longer adaptation time was needed for impulse response estimation, compared to the acoustic booth. Because of this, each two minute recording was divided into four 30-second segments. The learn and freeze method was used on each of these, yielding a total of 12 results for all three room microphones. Mean and standard deviation values was calculated based on these 12 segments. The filter length used was 16384 (corresponding to 0.37 seconds at a sampling frequency of 44.1 kHz) for each segment. Some preliminary tests were done before all segments were analyzed, to see which value of μ seemed to give the best results. It was found that $\mu = 0.1$ was a suitable value for the clarinet and the double bass, while $\mu = 0.3$ was suitable for the guitar, violin and bassoon.

In the same way as for the acoustic booth, signal-to-noise ratios also were calculated for the ensemble room, using a loudspeaker playing white noise as

[1] The reverberation time is defined as the time it takes for the envelope of a sound in the room to decay with 60 dB.

the sound source (see section 4.3.1).

4.5 Additional experiments

In addition to the experiments conducted to study the crosstalk cancellation of various sound sources in the two different rooms, a few other, smaller experiments were also conducted. These are described in the following sections.

4.5.1 Comparison of the LMS and FBLMS algorithms

In order to compare the NLMS and FBLMS algorithms (described in sections 2.3.3, 2.3.5 and 2.3.6), a simulated experiment was conducted, using a synthetic impulse response. The impulse response is shown in figure 4.9.

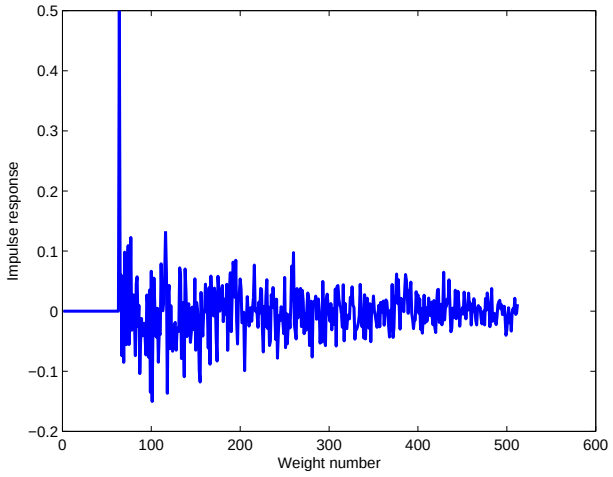


Figure 4.9: Synthetic impulse response used for comparison of NLMS and FBLMS algorithms.

The algorithms were tested for two different input signals; gaussian white noise and a recording of a guitar. The signals were first filtered with the synthetic impulse response to create a room microphone signal, and then both the original signal, $x(n)$, and the filtered signal, $d(n)$, were fed to the algorithms. The length of the adaptive filter was set to 512 coefficients, equal to the length of the synthetic impulse response. The sampling frequency used was 11025 Hz, and the step size (μ_{norm} for NLMS, μ for FBLMS) was set to 0.3 for both algorithms. For a sound clip of 14 seconds, the calculations took about 10 seconds for NLMS and 0.5 seconds for FBLMS.

4.5.2 Simulation of learn and freeze method in practical use

In most experiments in this work, the sound source was set up in the room with a reference and microphone and a room microphone, a mic-to-mic impulse response was calculated, and the resulting ERLE values calculated. This is what is called the "learning phase" of the learn and freeze method (see section 3.1). These values should indicate to what degree crosstalk from the first sound source could be reduced if an additional, second sound source was playing into the room microphone. In most experiments, a second sound source was not used, since the achieved crosstalk cancellation is not possible to measure in this case (as mentioned in section 2.5). However, to illustrate the practical use of the learn and freeze method, a few experiments were done to using two sound sources, to illustrate how the learn and freeze method would work in practice. Sound examples from these experiments were made, making it possible to at least have a subjective impression of the actual achieved crosstalk.

The first of the experiments was set up using a loudspeaker playing white noise as the first, crosstalk-producing sound source. The loudspeaker was set up in the acoustic booth, with one microphone 30 cm from the loudspeaker and the other 1 meter further away. Both microphones were AKG, with directivity set to omnidirectional. First, a 30 second "learning" recording was made with only the loudspeaker playing (white noise) and a person standing (but not singing) close to the second microphone. The algorithm used was FBLMS, with $\mu = 0.2$, $\beta = 0.9$ sampling frequency of 44.1 kHz and a filter length of 8192. Second, a recording was made with the person singing in his microphone simultaneously with the loudspeaker playing white noise. Afterwards, the learning recording was used to estimate the impulse response between the two microphones, and this estimate was subsequently used to reduce the white noise crosstalk in the second recording.

A similar experiment was conducted at the conservatory, but with a double bass as the crosstalk-producing sound source, and a violin as the second sound source. This case is even closer to a "real application" than the experiment with white noise and a person singing, since both sources are musical instruments. The double bass player was placed at the reference microphone, and the violin player was placed at the "Position 1" microphone (see plot of positions at the conservatory in figure 4.6). First a "learning" recording was done, with only the bass playing, and afterwards a recording was done with both instruments playing together. The learning recording was used to estimate the mic-to-mic impulse response, and this was then used to reduce the double bass crosstalk in the second recording.

4.5.3 Perceptual effects of different filter lengths

In section 2.6, it was found that for an exponentially decaying impulse response, the theoretically maximum achievable ERLE was a direct function of the length of the impulse response estimate. This suggests that a longer estimate will always be better. A small experiment was conducted to study the effect of a long estimate length for two different sound sources: A loudspeaker playing white noise, and a drum. Recordings were done of both sound sources in the acoustic booth, with equal distances to reference and room microphones (equivalent to "position 1", plotted in figure 4.4). Example sound files were also made for the drum, illustrating the result of the crosstalk cancellation.

4.5.4 Effects of microphone directivity and type

A small experiment was conducted in the acoustic booth to study the effect of microphone directivity on the achievable crosstalk cancellation. A loudspeaker playing white noise was used as the sound source, and recordings were made with omnidirectional and cardioid microphone directivities. Since it was possible to change the directivity of the AKG microphones that were used for the experiment, measurements could be made of two different directivities (omnidirectional and cardioid) without changing the measurement setup. In this way, any observed changes could be guaranteed to be effect of the changed directivity.

Measurements were made for three different cases: Both reference and room microphone having a omnidirectional directivity, both microphones having a cardioid directivity and facing the loudspeaker, and both microphones having a cardioid directivity, with the reference microphone facing the loudspeaker and the room microphone turned 90° to the side.

A recording of 1 minute was made of each setting, and the recording was divided into six segments of 10 seconds each. ERLE values were calculated for each segment, and then mean values and standard deviations for these were calculated.

4.5.5 Simulated experiment with the continuous update method

Although no "real-life" experiment were conducted with the learn and freeze method, a simulated experiment was conducted, using a synthetic impulse response. The experiment was set up as follows: White noise was used as the signal generating crosstalk. The "raw" noise was used as the reference signal $x(n)$, while a room microphone signal $d(n)$ was made by filtering the noise with a synthetic impulse response. A double-talk signal (see description in section

3.2) was also made, consisting of short segments of a recording of an electric guitar, with periods of silence in between. This double-talk signal was added to the room microphone signal before it was fed to the adaptive algorithm, simulating that a guitar was playing into the room microphone.

The FBLMS algorithm was used for the adaptive filtering, with the step size μ set to 0.5. Since the impulse response was known, it was possible to calculate how the system distance of the filter (see section 2.4) varied during the adaptation, that is, how large a difference there was between the correct impulse response and the estimate of this response.

4.5.6 Testing the minimum phase property of the reference microphone impulse response

As mentioned in section 3.1, the impulse response from the sound source to the reference microphone should be minimum phase in order for the mic-to-mic impulse response to be stable and causal. An experiment was done to see whether an impulse response estimate obtained for "best-case" conditions would fulfill this requirement.

A loudspeaker and a microphone were set up in the acoustic booth, with the microphone facing the loudspeaker and standing approximately 30 cm from it. This is similar to how the loudspeaker and reference microphone were set up during all other experiments with the loudspeaker and white noise. A 10 second recording was made of the loudspeaker playing white noise, and an estimate of the impulse response from loudspeaker to microphone was calculated. This was done by Fourier transformation of 10 seconds of the microphone and loudspeaker signals, elementwise division, and inverse transformation back to the time domain (described as the "direct method" in section 3.1).

Using MATLAB, a Nyquist plot was made of the resulting impulse response. Such a plot is a polar plot of the frequency response, with the radius given by the magnitude response and the angle given by the phase response. A Nyquist plot can be used to indicate whether an impulse response in minimum phase or not (see the description of the results in section 5.7 for further explanation).

4.5.7 Comparing noise input signals and their convergence rates for adaptive algorithms

A simulated experiment was conducted to investigate the performance of the FBLMS algorithm and the learn and freeze method when different kinds of noise were used as input signals. Three noise signals were used; white and pink

noise[2], and white noise played by a loudspeaker and recorded in the acoustic booth. The last signal was included to study the infulence of the loudspeaker, the room and the microphone on the white noise, and what effects this would have on its convergence rate.

Each of the signals were filtered with a synthetic impulse response to create a room microphone signal, and then the FBLMS algorithm was used, with identical parameters, on each signal pair. The synthetic impulse response was similar to a real mic-to-mic impulse response in the acoustic booth (see section 4.3.2). The synthetic impulse response was also longer than the adaptive filter, in order to simulate the "infinite length" of a real impulse response.

Three different length sequences were used for adaptation, one of 10 seconds, one of 30 seconds and one of 60 seconds. With the 10 second sequence, the adaptation process was aborted before it had reached a steady state - the error signal envelope had not become constant for any of the signals. The 60 second sequence was enough for the process to reach such a steady state for all of the signals. The 30 second sequence is somewhere in between – a steady state was reached for the "raw" noise signals, but not for the recorded one. Plots were made of the resulting ERLE values for all the segment lengths.

4.5.8 Calculation of ERLE

For all measurements done using learn and freeze method, ERLE was calculated as described in section 2.5. This means that first, the segment was used to estimate the impulse response, and then the signal $\hat{d}(n)$ was produced by filtering the $x(n)$ signal through this estimate. The $d(n)$ and $\hat{d}(n)$ signals were then each filtered in octave bands, using Chebyshev type 2 bandpass filters with 1 dB passband ripple and 80 dB stopband ripple. The width of the transition bands was set to a tenth of the bandwidths of the corresponding octave bands. ERLE was calculated as the difference in energy of these two signals.

One may argue that using the same segment for doing both the estimate and the calculation of ERLE might give an false impression of how well the method works in general - the estimate might work well for one particular segment, but not necessarily as well for other segments. A small experiment was conducted to investigate this, without any clear conclusion – it seemed that this method overestimated ERLE values somewhat in some cases, while they were underestimated in others. Seeing this, the method of using the same clip was still used, for practical purposes. See section 5.9 for details.

[2]Pink noise is noise with an energy density spectrum which is inversely proportional to the frequency, resulting in a -3 dB decay per octave. The amount of energy in each octave band is equal.

4.6 Estimation of power spectra

The power spectra of the musical instruments were estimated for all the measurements, both in the acoustic booth and in the ensemble room. The estimations were done using Welch's method [17], as it is implemented in the MATLAB function *pwelch()*. Welch's method splits the sequence to be analyzed into smaller segments, uses a windowing function on each segment, and calculates the power spectrum of each of these (the "modified periodogram"). These periodograms are then averaged, producing an estimate of the power spectrum with less variance than the individual periodograms. In all calculations, a Hamming window was used as the windowing function, and a 50% overlap was used between each segment. These are the default settings of the MATLAB function.

5

Results

In this chapter, the results of various experiments are presented. The first sections describe several experiments that were conducted to test the possibility of crosstalk cancellation in "real life". Later sections contain results from experiments that are in some ways more basic in nature, but which have been placed after the main results. This was done partly because the main results are easier to understand to begin with, and partly because they illustrate some of the reasons why the more basic experiments were done.

The first section (section 5.1) describes a small experiment where the performance of the NLMS and the FBLMS algorithms is compared. The results of this experiment contributed to the FBLMS algorithm being chosen as the main algorithm for the rest of the experiments. Sections 5.2 and 5.3 present the results of the main body of experimental work: A comparison of the crosstalk cancellation performance for several different sound sources, both in a small, damped room and in a large ensemble room. During these experiments, it was found that the filter length of the adaptive filter could have various perceptual effects. In section 5.4, an example of such effects is investigated further. An experiment was also conducted to study the effect of different microphones and their directivities. The results are presented in section 5.5. Following these sections, which are all based on the learn and freeze method, is section 5.6, which describes the results of a few simulated experiments with the continuous update method. The minimum phase property, which was mentioned in section 3.1, has been tested for the impulse response of a loudspeaker and a microphone in a small, heavily damped room. The results are presented in section 5.7. In order to further investigate results that were obtained with white noise as the input signal, a simulated experiment was done, comparing the performance for different noise types. ERLE values both during adaptation and after convergence were compared, and the results are shown in section 5.8. An evaluation was also done of the method used for calculating the ERLE values – this is presented in section 5.9.

For some of the experiments, sound examples were made to illustrate the perceptual effects of the results. The sound files for these examples are supplied together with this report. Note that these examples are best heard on a stereo

or a pair of headphones of good quality – effects may not be as easily detectable if the examples are played on, for example, laptop speakers.

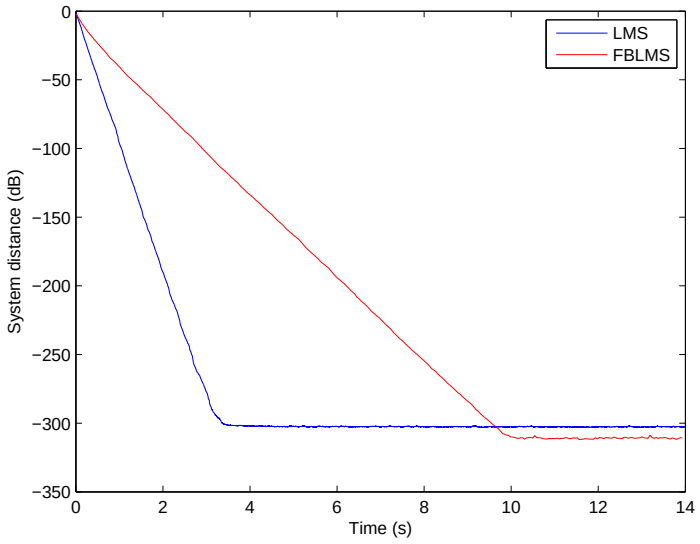
5.1 Comparison of NLMS and FBLMS

As mentioned in section 4.5.1, a small experiment was conducted to compare the performance of the NLMS and FBLMS algorithms. The experiment was a simulated one, using a synthetic impulse response, and both white noise and a recording of a guitar were used as input signals. Plots were made of the system distance (see definition in section 2.4) of the filters during adaptation.

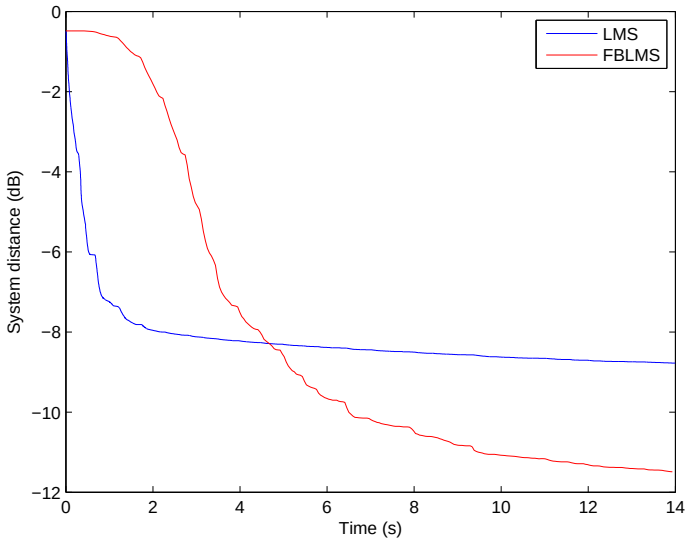
Figure 5.1(a) shows the system distance $\|\mathbf{\Delta w}(n)\|$ when the input signal is gaussian white noise. It is clear that for this signal, the system distance grows smaller linearly (in dB) for both algorithms. The NLMS algorithm converges faster then the FBLMS algorithm. Both algorithms reach a "steady-state" system distance of about -300 dB (which is extremely small), and it is assumed that this is due to limited numerical accuracy.

Figure 5.1(b) shows the corresponding results when a recording of a guitar is used as input. It is clear that with a more "realistic" input signal like this, the convergence rate is much slower for both algorithms. The NLMS algorithm converges faster than FBLMS, but reaches an almost constant system distance of about 8.5 dB. In the start-up phase, the FBLMS algorithm has an almost constant system distance, but after some time it drops below that of the NLMS algorithm and reaches a value about 3 dB under the NLMS system distance. Contrary to the white noise signal, the guitar signal probably has a large degree of self-correlation, and as was mentioned in section 2.2.3, this leads to a slower convergence rate. It may be that the smaller system distance achieved by the FBLMS algorithm is due to its "decorrelation" of the input signal, which is performed by normalizing the input signal with an estimate of the power spectrum (see section 2.3.6).

This small experiment shows that for a fixed impulse response, the performance of both the NLMS and FBLMS algorithms is strongly dependent on the input signal.



(a) Input signal: White noise



(b) Input signal: Recording of acoustic guitar

Figure 5.1: System distance resulting from two different input signals in a simulated experiment comparing the NLMS and FBLMS algorithms. The line labeled 'LMS' actually represents the NLMS results.

5.2 Crosstalk cancellation performance for various sources in a small, damped room using the learn and freeze method

As described in section 4.3, several experiments were done in the "acoustic booth", which is a small, heavily damped room. Although it is not an anechoic room, the level of reverberant sound is relatively low compared to the direct sound, and the reverberation time is very short. The resulting room impulse responses should be both quite short and quite simple, and thus these conditions should be very good for estimation of mic-to-mic impulse responses and therefore also crosstalk cancellation.

Measurements were done of six different sound sources: A loudspeaker playing white noise, an acoustic guitar, an electrically amplified guitar, a drum, a male person singing, and a trombone. The resulting ERLE values of these are presented in sections 5.2.1 to 5.2.6. A plot of the measured signal-to-noise ratios in octave bands is included in section 5.2.7.

5.2.1 Loudspeaker playing white noise

As was described in section 4.2, a loudspeaker playing white noise was chosen as the reference sound source. Because of its favorable qualities, the experiments conducted with this sound source were assumed to be "best case".

Figure 5.2(a) shows the ERLE mean and standard deviation values resulting from the experiments done with the white noise sound source. It is evident that the 500 and 1000 Hz octave bands have the highest mean values, with ERLE about 35 dB. Values in frequency bands below and above these are slightly lower, with the 63 Hz and 8 kHz bands having the lowest mean ERLE values, about 20-25 dB. Note that the standard deviations are also largest for the lowest mean values.

Although the spectrum plotted in figure 5.2(b) is reasonably flat, it clearly has some peaks and dips. Since the input signal to the loudspeaker is perfectly flat, these peaks and dips must be caused by the collective effect of the loudspeaker, the room and the microphone. This indicates that although a perfectly white signal is assumed, in practice this assumption is only approximately fulfilled.

5.2.2 Acoustic guitar

An experiment was also conducted using an acoustic guitar as the sound source. The guitar had steel strings, and only chords were played. For adaptation, a filter length $L = 4096$ and step size $\mu = 0.1$ was found to be suitable values.

Figure 5.2(c) shows the ERLE mean and standard deviation values resulting from this experiment.

We see that the mean values of ERLE are positive only in the octave bands from 63 Hz to 1000 Hz. Above these the mean ERLE is negative, indicating a slight *increase* in crosstalk noise. It is evident that for guitar, the method only works for relatively low frequencies. The ERLE is highest in the 250 Hz octave band, with a mean ERLE of about 15 dB. The standard deviations for the 63 and 125 Hz octave bands are quite large, indicating variable results in these bands.

The magnitude spectrum of the guitar signal is plotted in figure 5.2(d). The guitar signal clearly has most energy in the lowest frequencies, with two peaks at about 100 and 200 Hz. Above this, the spectrum decays almost linearly with frequency (on a log-log plot), with the magnitude being about 60 dB lower at 10000 Hz than at 100 Hz.

In [6, chapter 9], it is claimed that most guitars have three strong resonances in the 100-200 Hz range, due to the coupling between the Helmholtz resonance (resonance caused by spring effect of air cavity and mass of moving air in sound hole) and the (1,1) modes[1] of the top and bottom plate. Measurements made of a Martin D-28 folk guitar, not unlike the one used in these experiments, showed that the resonance frequencies of these modes were 102, 193 and 204 Hz. The radiated sound pressure levels was also high at these frequencies. This is in good agreement with what was found in the magnitude spectrum for the guitar used for this experiment. Measurements of radiation patterns showed that the Martin D-28 guitar was approximately omnidirectional at the same frequencies.

5.2.3 Electrically amplified guitar

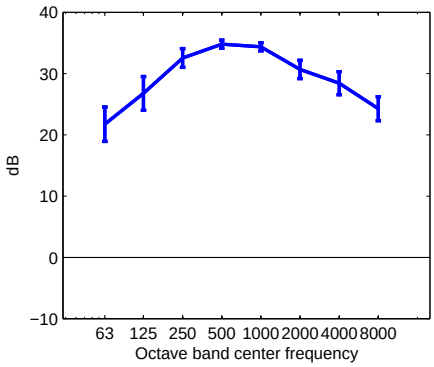
In addition to using an acoustic guitar, an experiment was also made with an electrically amplified guitar. Naturally, in this case the speaker of the amplifier was treated as the sound source, not the guitar itself. The preamplifier was adjusted to give a slightly distorted, "crunchy" sound. Such distortion introduces additional high-frequency components to the original guitar sound. For the adaptive process, a filter length of $L = 4096$ and a step size $\mu = 0.1$ were found to be suitable.

Figure 5.2(e) shows the mean and standard deviation values of the ERLE resulting from this experiment. The highest mean value is found in the 250 Hz band (approx. 23 dB), with gradually lower levels for both lower and higher frequencies. The 250 - 1000 Hz bands also have surprisingly high

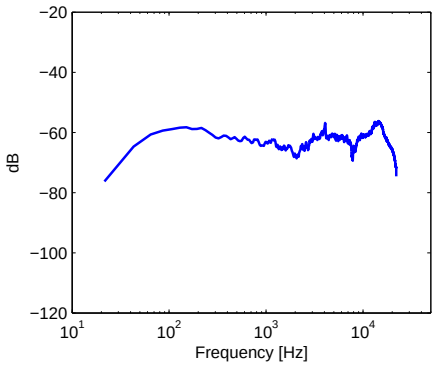
[1] In this case, the (1,1) mode refers to the case where the whole plate is vibrating in phase.

standard deviation values. Although the ERLE values are not as high as in the white noise experiment, a crosstalk reduction of 10-20 dB in most bands is substantial.

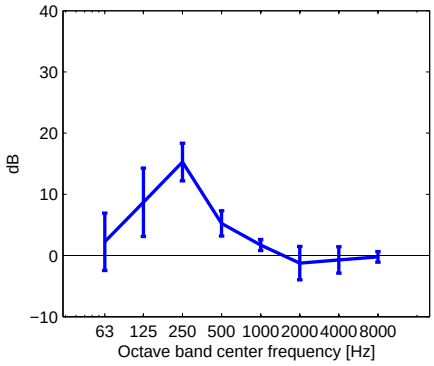
The magnitude spectrum of the electric guitar signal is plotted in figure 5.2(f). We can see that between about 100 Hz and 3000-4000 Hz, the spectrum is actually very flat. Compared with the acoustic guitar, the spectrum has more high-frequency energy, probably because of the distortion mentioned above. The cut-off at about 3000-4000 Hz is probably due to limitations of the speaker and/or a low-pass filter in the amplifier. This cut-off is probably also the main reason for the mean ERLE being so low in the 8000 Hz octave band.



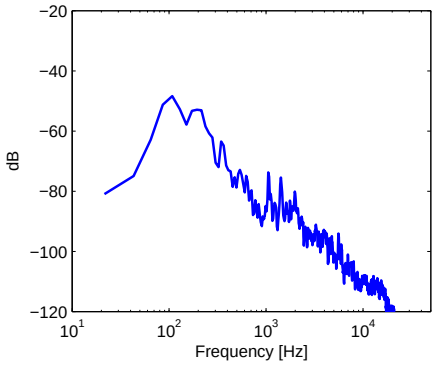
(a) White noise: ERLE values



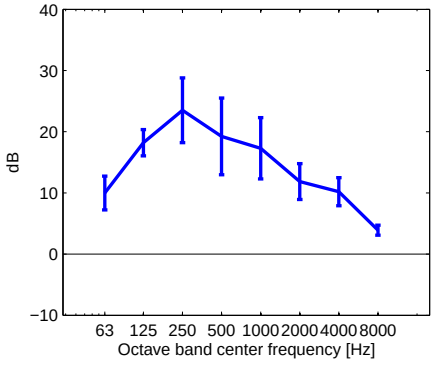
(b) White noise: Magnitude spectrum



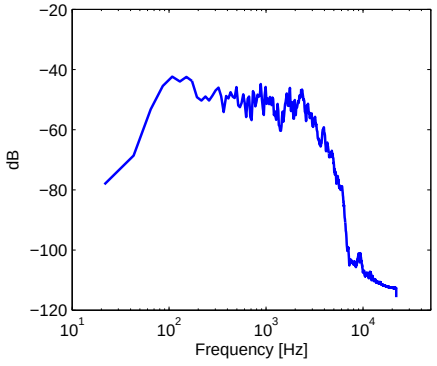
(c) Acoustic guitar: ERLE values



(d) Acoustic guitar: Magnitude spectrum



(e) Electric guitar: ERLE values



(f) Electric guitar: Magnitude spectrum

Figure 5.2: ERLE values, with mean and standard deviation, for a loudspeaker playing white noise, an acoustic guitar, and an electrically ampified guitar. The corresponding magnitude spectrums are also plotted.

5.2.4 Drum

A snare drum without the snares on was also used as a sound source in experiments. The snares were not used because they make the drum a much more complex sound source. Keeping the snares off also makes the drum more similar to other drums which do not have snares, perhaps also making the results more similar to those that could be obtained with other kinds of drums. For the adaptation, it was found that filter length $L = 8192$ and step size $\mu = 0.5$ were suitable.

Figure 5.3(a) shows the mean and standard deviation values of the ERLE resulting from the experiments using the drum as the sound source. Only the 125 octave band has considerable damping of the crosstalk, with a mean ERLE of about 15 dB. The other bands also have some damping, but only a few dB.

The magnitude spectrum of the drum signal is shown in figure 5.3(b). The spectrum has a dominant peak at about 150-200 Hz, and several smaller peaks at higher frequencies. The spectrum seems to decay linearly with higher frequencies (in a log-log plot).

According to [6], a drum similar to the one used in this measurement has two resonances caused by the coupling of the (0,1) modes [2] of the drum heads. The lowest of these, where both the drum heads are moving in the same direction, has a resonance frequency of 182 Hz. The drum is more like a monopole than a dipole in this case, due to the large difference in thickness (and thus weight) of the two heads. The (0,1) modes also radiate sound energy quite effectively, yielding high sound levels. This is in quite good agreement with what was found in the magnitude spectrum - a clear peak at about 150-200 Hz. Comparing with the ERLE plot, one can see that the ERLE has its highest value in approximately the same frequency range - the 125 Hz band.

5.2.5 Male voice

The voice of a male person (the author) singing was used as a sound source in this experiment. The same song was repeated for all measurements. For the adaptive process, a filter length of $L = 2048$ and a step size $\mu = 0.1$ was found suitable.

Figure 5.3(c) shows the mean and standard deviation values of the ERLE resulting from the experiment with this sound source. There is substantial damping in several octave bands, with the bands between 125 Hz and 1000 Hz all having mean ERLE higher than 10 dB.

[2] For a circular membrane like a drum head, the (0,1) mode refers to the case where the entire head is vibrating in phase.

In figure 5.3(d), the magnitude spectrum of the male voice is plotted. The plot shows that much of the energy is in the 100-500 Hz frequency range, but there is substantial energy up to about 3-4 kHz, where there is a kind of cut-off.

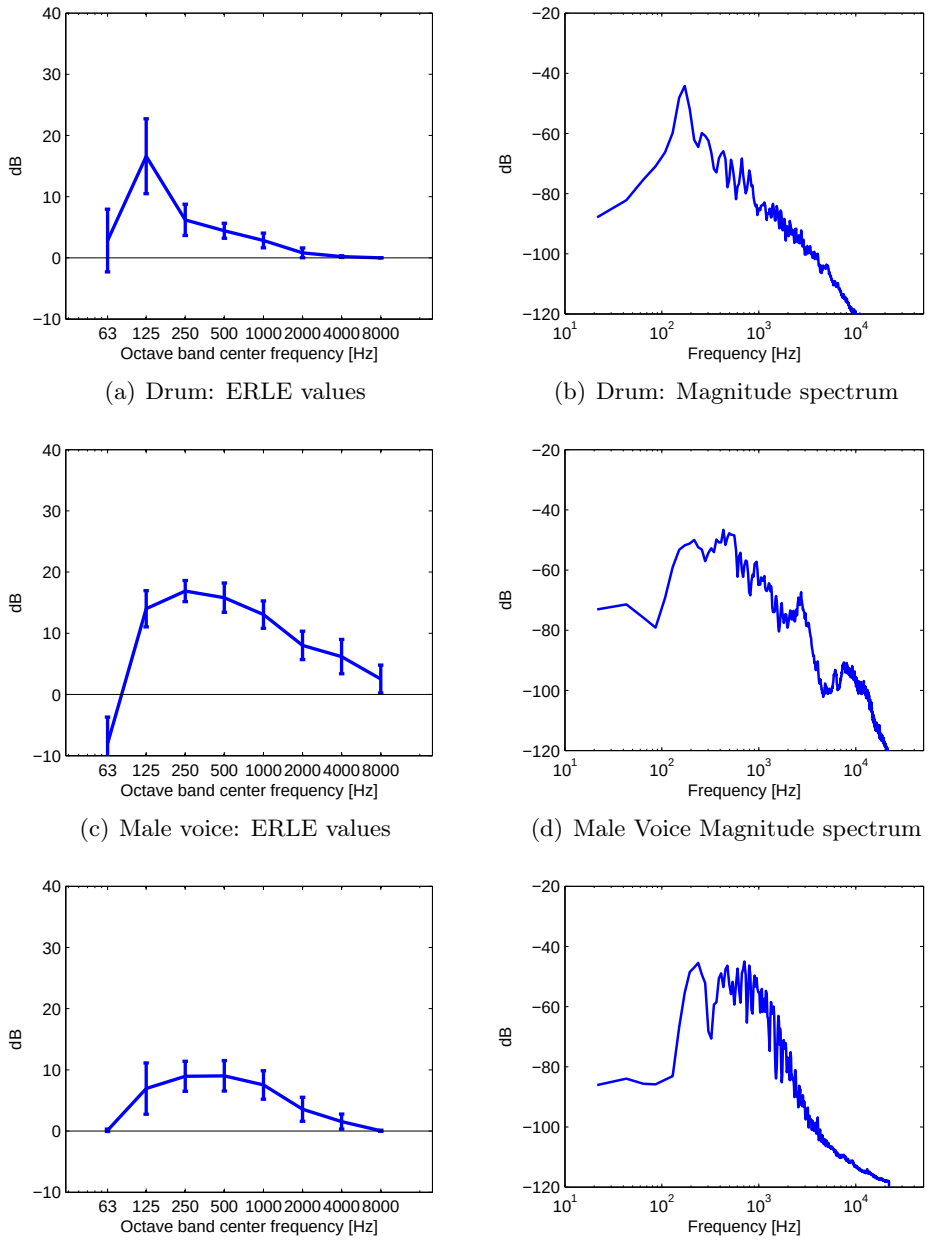
After studying the error signal of some segments, it was found that a so-called "pop noise", resulting from sudden outlets of air from the singer, results in a very large error. This is to be expected, as this noise is only audible in the microphone close to the singer, and not in the rest of the room (and therefore not in the other microphone). Since this is mainly a low frequency noise, it is suspected that pop noise may be the reason for the low mean ERLE values in the 63 Hz band. Pop noise may be avoided by using a "pop filter" - an acoustically transparent membrane placed in front of the microphone. This was unfortunately not used during measurements.

5.2.6 Trombone

A person playing the trombone was the last sound source tested in the acoustic booth. The trombone was played moderately hard, with long tones. For adaptation, a filter length of $L = 4096$ and a step size $\mu = 0.05$ were found to be suitable.

Figure 5.3(e) shows the ERLE mean and standard deviation values resulting from the experiment with the trombone. The mean ERLE value is about 7-9 dB in octave bands between 125 and 1000 Hz , and lower in other bands. These values are surprisingly low, but as we shall see in the next section, the calm playing style may have affected the result.

The magnitude spectrum of the trombone, which is plotted in figure 5.3(f), shows that most energy from a this instrument is contained in a relatively narrow frequency range - from about 150 Hz to 2 kHz.



(a) Drum: ERLE values

(b) Drum: Magnitude spectrum

(c) Male voice: ERLE values

(d) Male Voice Magnitude spectrum

(e) Trombone: ERLE values

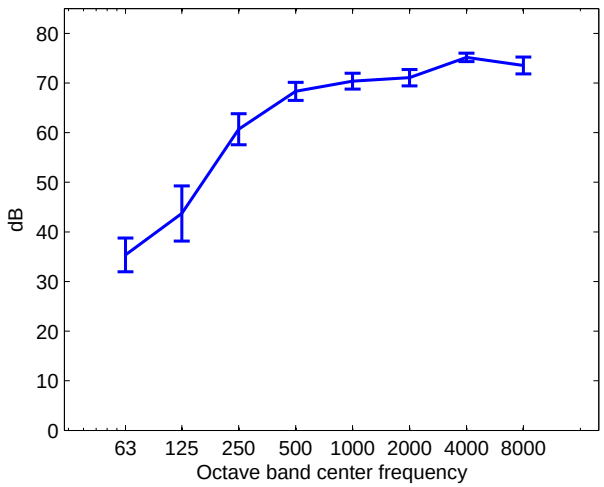
(f) Trombone: Magnitude spectrum

Figure 5.3: ERLE values, with mean and standard deviation, for a drum, an male voice, and trombone. The corresponding magnitude spectrums are also plotted.

5.2.7 SNR values in the acoustic booth

A measurement was made of the signal-to-noise ratio for the loudspeaker playing white noise in the acoustic booth. Measurements were done for the six microphone positions in the room (plotted in figure 4.4), and figure 5.4 shows the mean and standard deviation of the SNR values produced. It is evident that the SNR levels are quite a lot lower for low frequencies than for high, indicating that the background noise is mainly a problem in the low frequency range.

Signal-to-noise ratios were not calculated for each instrument, but by comparing the magnitude spectrum of the instruments and the SNR plot for the white noise, one can get a general impression of the signal-to-noise ratios of each instrument, since the white noise has an approximately flat spectrum. Most of the instruments have quite a lot of energy in approximately the 100-1000 Hz frequency range, and less energy above and below this range. Since the background noise levels seem to be strongest in the low frequency range, a low signal-to-noise level is mainly a problem for the musical instruments in the 63 and 125 Hz octave bands. This may also explain, in part, why ERLE values in these bands are also generally quite low. This is subject to discussion in chapter 6.



(a) ERLE values in octave bands

Figure 5.4: SNR values in octave bands for a loudspeaker playing white noise in the acoustic booth

5.2.8 Sound examples

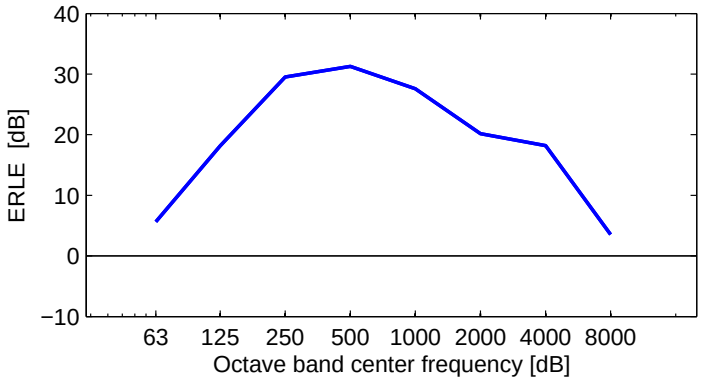
In this section, a few sound examples from experiments in the acoustic booth are described. The sound files for the examples have been supplied together with the report.

Acoustic and electric guitar

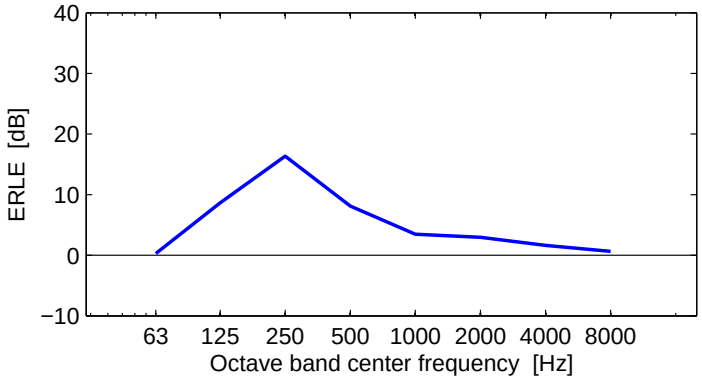
Both of these examples are based on measurements made with only one sound source, done in the exact same way as the experiments which have just been described in the preceding sections (see also section 4.3.1). The mic-to-mic impulse response was estimated based on one recorded segment, and then crosstalk cancellation was done on the same segment, by subtracting the modeled room microphone signal signal $\hat{d}(n)$ from the actual room microphone signal $d(n)$. The "before" sound files contain the signal $d(n)$, before crosstalk cancellation, while the "after" sound files contain the crosstalk-reduced signal $d(n) - \hat{d}(n)$. A speech segment has been artificially added to the sound clips, to simulate how the guitar signals act as crosstalk.

The files named `elguitar_before` and `elguitar_after` are from the example using an electric guitar as the sound source. The ERLE values for the clip used in this example are plotted in figure 5.5(a). As we can see from the plot, the achieved crosstalk cancellation in this case is quite similar to the mean performance for the electric guitar (plotted in figure 5.2(e)), with substantial damping across most frequency bands. Comparing the "before" and "after" sound examples, one can clearly also hear how the guitar crosstalk becomes much lower after crosstalk cancellation, making it much easier to hear the speech which has been added. One can perhaps also hear that there is some fluctuation in the residue of the crosstalk in the "after" signal. This is probably because of the physical impulse response was changing somewhat during the recording, while the impulse response estimate used to reduce crosstalk is kept constant throughout.

The files `guitar_before` and `guitar_after` illustrate an example of the learn and freeze method providing damping in a more narrow frequency range. As we can see in figure 5.5(b), the ERLE values in this case are relativly high in the in the 250 Hz band, but not very high outside this band. Comparing the sound examples, one can hear that the overall sound is perceived as somewhat lower after crosstalk cancellation, but also that it has lost more low frequency than high frequency content.



(a) Electric guitar



(b) Acoustic guitar

Figure 5.5: ERLE values for example clips crosstalk cancellation of acoustic and electric guitar.

Simulation of learn and freeze method in practical use, using white noise and male voice

As described in section 4.5.2, an experiment was also conducted using two sound sources, to illustrate the use of the learn and freeze method in "real life". In the acoustic booth, one microphone was set up close to a loudspeaker playing white noise, and 1 meter further away, a microphone was set up for a person to sing into. First, a learning recording was done, with the person standing by his microphone as he would when he was singing. Then a recording was made of both the loudspeaker playing and the person singing into the microphone. The impulse response estimated from the first recording was then used to reduce the white noise crosstalk of the second recording.

The files `wnoise+song_before` and `wnoise+song_after` contain outtakes of this second recording, before and after crosstalk cancellation. In the "before"

file, we hear the person singing an outtake of the song "I Hung My Head" by Sting, with quite a lot of white noise crosstalk in the background. In the "after" file, we hear that much of the noise has been removed, while the singing is essentially the same. This illustrates that the learn and freeze method may work quite well in practice.

Note that high-frequency noise is damped less than low-frequency noise. Looking back at the mean ERLE values for a loudspeaker playing white noise (figure 5.2(a)), this seems reasonable – the crosstalk cancellation at the highest frequencies becomes gradually lower with increasing frequency. Crosstalk cancellation at the lowest frequencies is also seen to be quite low, but the effect is heard more clearly for the highest frequencies, since the high-frequency octave bands contain more energy than the low-frequency octave bands (due to their wider bandwidth)

Note also that there is a certain fluctuation in the residue of the white noise crosstalk. It seems that also this effect is most easily heard for the higher frequencies. A possible reason for the effect may be that the actual ("physical") impulse response in the room is changing due to movement of the singer, while the impulse response estimate used for crosstalk cancellation is kept constant. This is discussed further in section 6.6.2

5.3 Crosstalk cancellation performance for various sources in a medium-sized ensemble room using the learn and freeze method

The measurements in the small, heavily damped room were useful to study the performance of the adaptive algorithms in a very controlled, best-case environment – but since the final area of application is a room in which several musicians are playing together, measurements were also done in an "ensemble" room, intended for musical performance. As described in section 4.4, this ensemble room was part of the music conservatory in Tromsø, and recordings were done of several different musical instruments: Double bass, violin, clarinet, bassoon, and classical guitar. Again, the musical instruments were chosen based on what was available (volunteering students), but an effort was also made to find different kinds of instruments, to test the method for a wide range of sound sources. In addition to the musical instruments, recordings were also made of the reference sound source; a loudspeaker playing white noise.

Each instrument was recorded by three microphones in different positions in the room, and each musician played both a "calm" and a "fast" piece of music, each two minutes long. As described in section 4.4, each recording was divided into 30 second segments, and the learn and freeze method with the FBLMS algorithm was used on each of these. ERLE mean and standard deviation values were calculated from results from all microphone positions and segments – 12 results from both the "calm" and the "fast" recording. These results are presented in the following sections.

5.3.1 Loudspeaker playing white noise

As described in section 4.2, a recording of the loudspeaker playing white noise was done in the ensemble room. Recordings were done for two different distances between the reference microphone and the room microphones: 1.2 meters, which was the same distance that was used in the acoustic booth, and 2 meters, which was the distance used for recordings of all other instruments in the ensemble room.

Figure 5.6 shows the ERLE values for both distances between reference microphone and room microphones. The values for both cases are somewhat lower than for the experiment in the acoustic booth. In the 1.2 meter case, the distribution of the ERLE values is quite similar to the one found for the acoustic booth, with the values being highest in the 250, 500 and 1000 Hz octave bands.

For the 2 meter case, there seems to be a slight shift downward in frequency – the values in the 63 - 250 Hz bands are slightly higher than in the 1.2 meter

case, while the values above this are slightly lower.

Using the recording of the white noise sound source, an estimate of the reverberation time in the ensemble room could be calculated, as described in section 4.4. The reverberation time was estimated to approximately 0.7 seconds. This is probably quite representative of the reverberation time of a small concert venue.

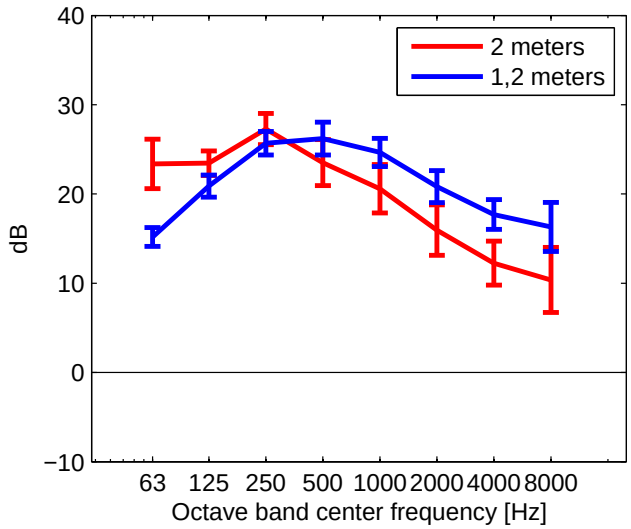


Figure 5.6: ERLE in octave bands, with mean and standard deviation, for recordings of a loudspeaker playing white noise at the music conservatory. Results for room microphones standing 1.2 and 2 meter from the reference microphone are plotted.

5.3.2 Double bass

Figure 5.7(a) shows the ERLE values for the double bass. Clearly, the only octave bands which have any substantial damping are the 63 and 125 Hz bands. It is also evident that the "fast" playing style gives somewhat better results in these bands, while the results are very much the same in the other bands.

In figure 5.7(b), the magnitude response of the double bass is plotted. There is a lot of energy in the low end of the spectrum, around 50-200 Hz, and for the "fast" playing style there seems to be a peak at about 60-70 Hz. There is little difference in the spectra of the calm and the fast playing style, but the fast style seems to have somewhat more energy around the low-frequency resonances, and a slightly less steep decay for the higher frequencies.

According to [6], the two main low-frequency resonances of a double bass are at about 60 Hz ("A_0" - air cavity resonance) and 100 Hz ("T_1" - entire bottom

plate vibration in phase, while the top plate has a slightly more complex mode pattern). The 60 Hz resonance is probably what was was found as a peak in the magnitude spectrum.

5.3.3 Violin

Figure 5.7(c) shows the ERLE values for the violin recordings. Here, the 250 and 500 Hz bands have the highest ERLE values. The "fast" playing style yield better mean values, and also has lower standard deviation values in these bands then the "calm" playing style.

The magnitude spectrum of the violin is plotted in figure 5.7(d). The spectrum has a clear high-pass effect, with an cutoff frequency at about 200 Hz.

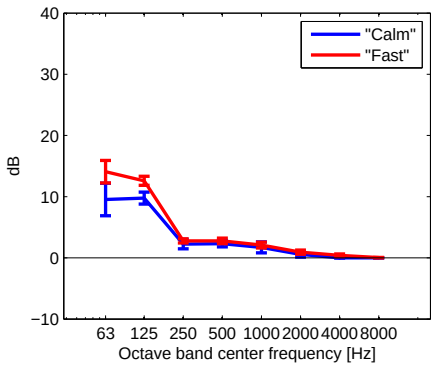
In [6], the two main low-frequency resonances of a violin are measured to 275 Hz (A_0) and 460 Hz (T_1). The radiation patterns of a violin are also presented for a few frequencies. They show that the violin is more or less omnidirectional up to about 400 Hz. For higher frequencies, the radiation pattern soon becomes more complex.

It is interesting to compare the results of the double bass and the violin. The double bass is, after all, a scaled-up version of the violin. We can see that both instruments have two frequency bands in which the ERLE values are much higher than in the other bands, but these bands are shifted upwards in frequency for the violin, compared with the double bass. In the same way, the shape of their spectrums are quite alike, but the spectrum of the violin is shifted upwards in frequency. The bands with the highest ERLE values also coincide quite well with the two lowest resonances of the instruments.

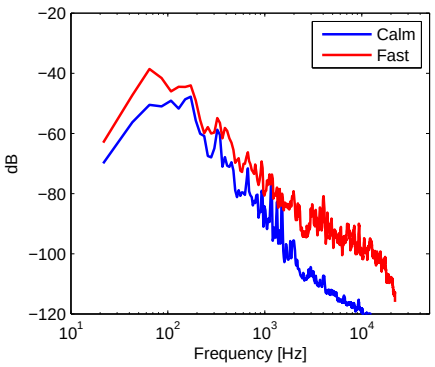
5.3.4 Clarinet

Figure 5.7(e) shows the ERLE values for the recordings of the clarinet. This shows some damping in the octave bands between 125 and 1000 Hz, with the "fast" playing style having significantly higher values in the 125 - 500 Hz octave bands. The fast playing style also yields lower standard deviations.

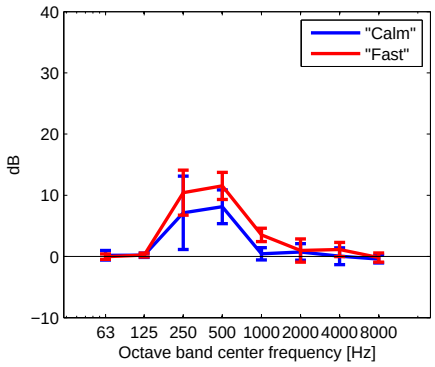
The magnitude spectrum of the clarinet, plotted in figure 5.7(f), shows a very large difference between the calm and fast playing style. While there is almost no energy under 400-500 Hz with the calm playing style, the fast style has energy all the way down to about 100 Hz. This may explain the significant difference in ERLE values in the 125 - 500 Hz octave bands.



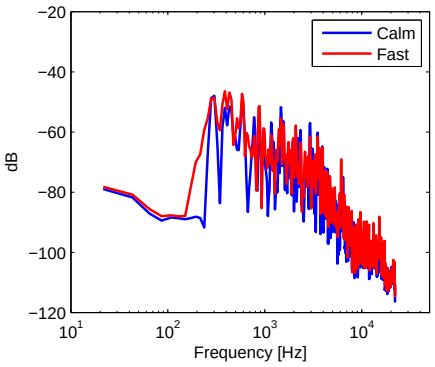
(a) Double bass: ERLE values



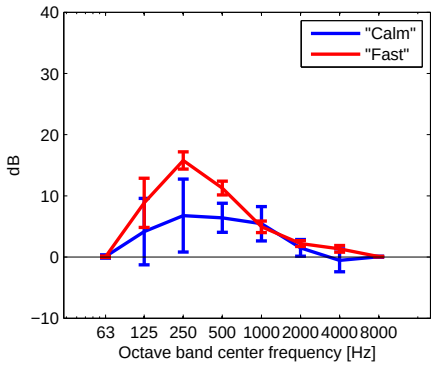
(b) Double bass: Magnitude spectrum



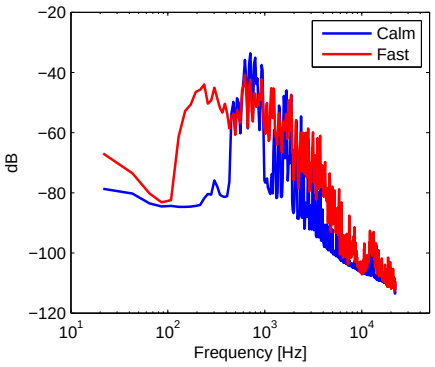
(c) Violin: ERLE values



(d) Violin: Magnitude spectrum



(e) Clarinet: ERLE values



(f) Clarinet: Magnitude spectrum

Figure 5.7: ERLE values, with mean and standard deviation, for a double bass, a violin, and a clarinet. The corresponding magnitude spectrums are also plotted.

5.3.5 Bassoon

Figure 5.8(a) shows the ERLE values for the recordings of the bassoon. Clearly, the attempted damping of the crosstalk has done nothing (or even made things a bit worse) for the "calm" playing style, while some damping has been accomplished in the 125-500 Hz octave bands for the "fast" playing style.

The magnitude spectrum of the bassoon is plotted in figure 5.8(b). We can see that there is little difference between the two playing styles, except for the calm style actually having more energy in the low end on the spectrum, below 500 Hz. There is also a high-frequency cut-off at about 1000-2000 Hz.

5.3.6 Classical guitar

Figure 5.8(c) shows the ERLE values for the classical guitar. There is substantial damping in the 125 - 500 Hz octave bands. Here, the difference between the two playing styles is not so great, but still the "fast" style has a better mean value in the 125 Hz band. The ERLE values are quite similar to those of the steel-stringed guitar being played in the acoustic booth (see figure 5.2(c)), but are slightly higher.

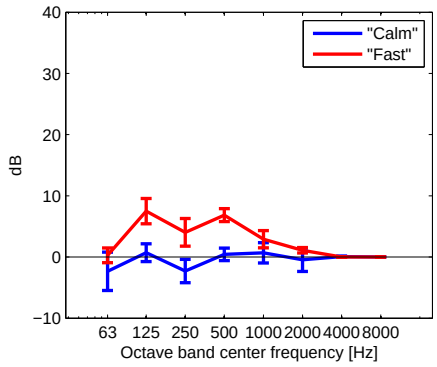
The magnitude spectrum of the classical guitar (figure 5.8(d)) is practically identical for the two playing styles, and also very similar to that measured for the steel string guitar in the acoustic booth.

5.3.7 SNR values at the music conservatory

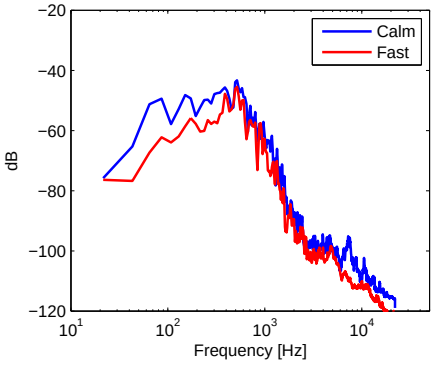
As for the measurements in the acoustic booth, a measurement of the signal-to-noise ratio was done at the music conservatory. A loudspeaker playing white noise was used as the sound source. The results are plotted in figure 5.9. As can be seen from the plot, the situation is very much the same as in the acoustic booth; the SNR values are generally lowest for the lowest frequencies (about 35 dB), and gradually rise to a level of about 70 dB at 500 Hz, from where the values are approximately constant. This means that the background noise level should not represent any systematic difference between the results from the acoustic booth and from the conservatory.

5.3.8 Sound example: Practical use of the learn and freeze method using double bass and violin as sound sources

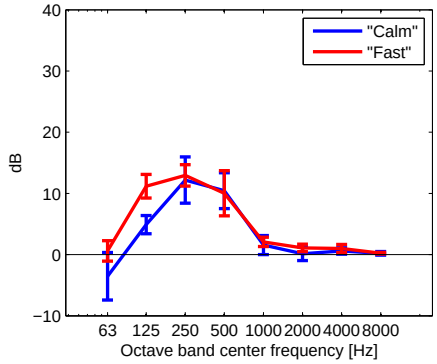
As described in section 4.5.2, a experiment was conducted at the conservatory to study the practical use of the learn and freeze method with two musical



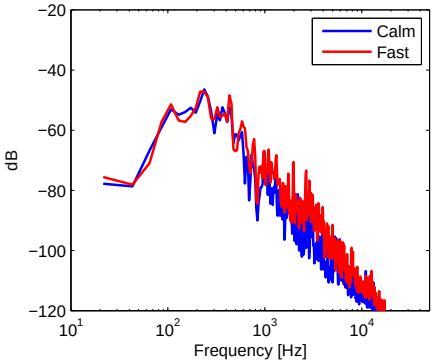
(a) Bassoon: ERLE values



(b) Bassoon: Magnitude spectrum



(c) Classical guitar: ERLE values



(d) Classical guitar: Magnitude spectrum

Figure 5.8: ERLE values, with mean and standard deviation, for a bassoon, and a classical guitar. The corresponding magnitude spectrums are also plotted.

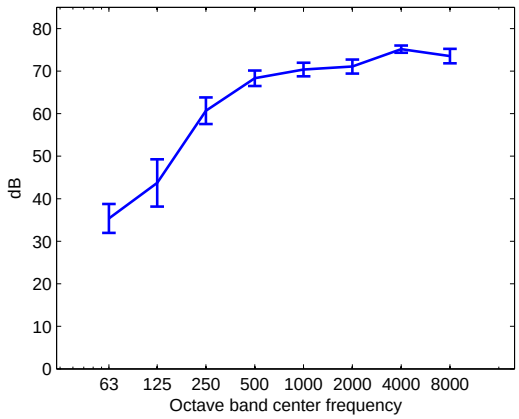


Figure 5.9: SNR values in octave bands for a loudspeaker playing white noise in the ensemble room at the conservatory.

instruments. The double bass was used as the crosstalk-producing sound source, and a violin was used as the second sound source.

First, a learning recording was made with both instruments in place but only the double bass playing. Then a second recording was made with both instruments playing. The mic-to-mic impulse response estimated from the first recording was then used to reduce the double bass crosstalk in the violin microphone in the second recording.

The file `dbass+violin_before` contains the signal from the violin microphone before crosstalk cancellation, and file `dbass+violin_after` contains the signal after cancellation. Since the double bass is not a very loud instrument to begin with, the crosstalk is not very loud either, but nevertheless there is an audible difference between the two tracks. The mean ERLE values of the bass were found to be relatively high in the two lowest frequency bands, and lower in the higher frequency bands (see figure 5.7(a)). This is in agreement with what is heard in the sound examples: The double bass crosstalk is mainly damped in the lower frequencies. The loudness of the crosstalk is not necessarily perceived as lower, but the sound is slightly less "boomy" and defined. Note that a stereo or a pair of headphones with sufficient bass response may be needed to hear this, since the difference is mainly at very low frequencies.

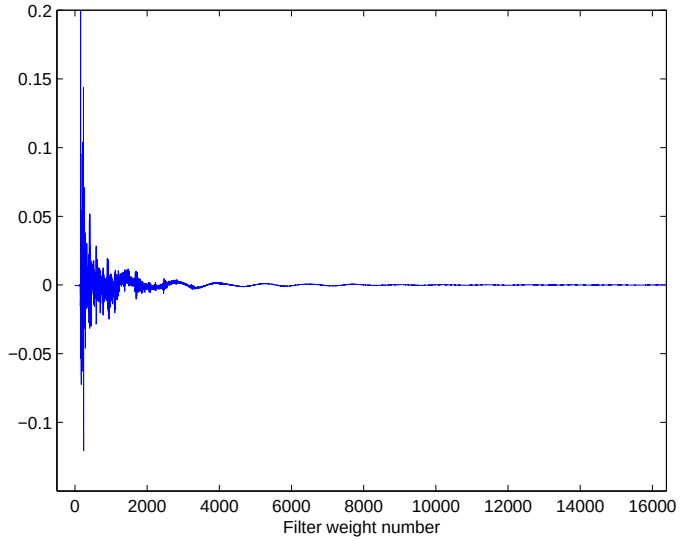
5.4 Perceptual effects of different filter lengths

As described in section 2.6, the theoretically maximum achievable ERLE is a direct function of the length of the impulse response estimate. In practice, the impulse response will hit the "noise floor" of the room after a certain time, and one can not expect to gain additional crosstalk cancellation by making this estimate longer than this.

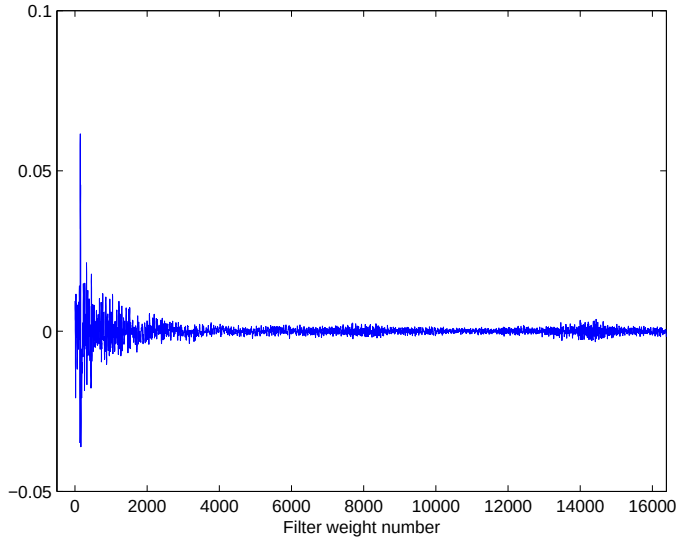
For the best possible damping, the filter length should be at least as long as the impulse response before it hits the noise floor, or there will be a residual "reverberation tail" which is not cancelled. But what will happen if the estimate is longer than necessary? As described in section 4.5.8, a small experiment has been conducted to study the effect of a too-long impulse response estimate for two widely different sound sources; a loudspeaker playing white noise and a drum. The results of this experiment are presented below.

Figure 5.10(a) shows a plot of the estimated impulse response in the acoustic booth, using a loudspeaker playing white noise as the sound source. We see that in this case, the filter length may be longer than necessary, since the last coefficients are very close to zero. However, the fact that they are close to zero also means that there is no harm done in using a too-long filter, except for the waste of filter coefficients.

In comparison to this, figure 5.10(b) shows the estimated impulse response when a snare drum without snares is used as a sound source. The drum and the microphones were placed in approximately the same positions as when a loudspeaker was used. Comparing with figure 5.10(a), we see that the impulse response estimate has a kind of noisy tail which does not decay with time (as a room impulse response always does), but rather has an approximately constant amplitude. Clearly this is a result of an imperfect estimate, and not of the acoustic system's properties. If this estimate is used for crosstalk cancellation, the noise tail at the end is heard as an unnatural reverberation. The files `drum_toolongfilter_before` and `drum_toolongfilter_after` supplied with this report contain before and after versions of crosstalk cancellation where this estimate has been used. Here one can clearly hear that although the sound level is somewhat lower in the after version, there is also a reverberation or echo effect, almost as if the player is standing outside, perhaps near a reflective wall. In general, such an effect is unwanted as long as it can not be controlled. Although this example was made using the learn and freeze method, this kind of effect will probably also be a problem using the continuous update method.



(a) Estimated impulse response for loudspeaker playing white noise



(b) Estimated impulse response for a snaredrum (without snares)

Figure 5.10: Comparison of estimated impulse responses using white noise and a snare drum as sound sources in the acoustic booth. In both cases the filter length used is too long. For the white noise response, this is not a problem, since the last coefficients are very close to zero. For the snare drum, the imperfect estimate of the response combined with a too-long filter length results in an "unnatural", non-decaying tail.

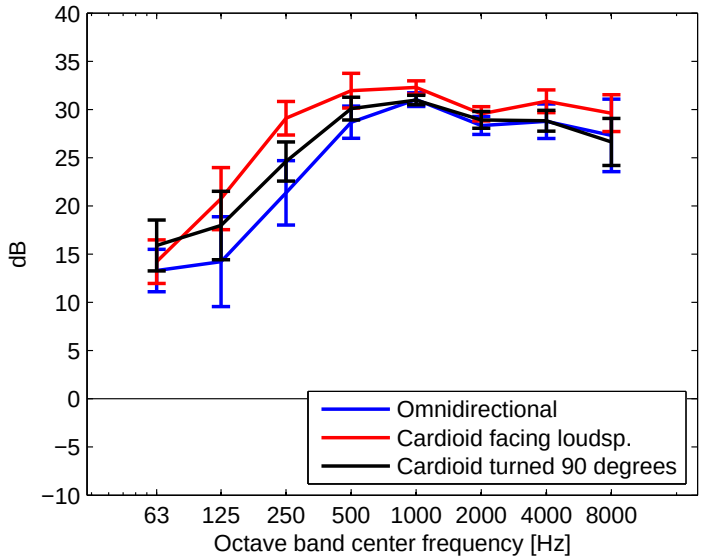
5.5 Effects of microphone directivity and type

As described in section 4.5.4, an experiment was conducted in the acoustic booth to study the effects of microphone directivity on the achievable crosstalk cancellation. A loudspeaker playing white noise was used as the sound source, and measurements were made for three different cases: Both reference and room microphone having a omnidirectional directivity, both microphones having a cardioid directivity and facing the loudspeaker, and both microphones having a cardioid directivity, with the reference microphone facing the loudspeaker and the room microphone turned 90° to the side. Figure 5.11(a) shows the mean and standard deviations of the ERLE results resulting for each of the directivity settings.

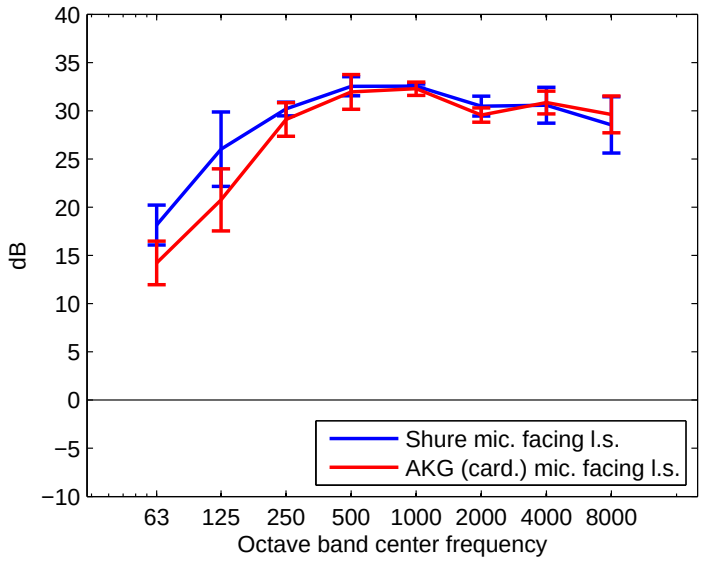
It is evident that the omnidirectional microphone yields the lowest mean ERLE values, and also the highest standard deviations. The cardioid directivity gives a mean ERLE which is several dB higher than that of the omnidirectional microphone – especially in the lower frequency bands. The results of the cardioid microphone which was turned 90° away from the sound source lies between the other two – the mean ERLE values are slightly better than that of the omnidirectional microphone in the lower frequency bands, and more or less the same in higher frequency bands.

A measurement was also conducted to compare the AKG microphones used in all the other measurements with two dynamic microphones produced by Shure. An SM 57 was used as reference microphone, while an SM 58 was used as a room microphone. Both have a cardioid directivity. These are low-budget, standard microphones used on almost any stage in the world. The directivity of the AKG microphones was also set to cardioid, for the conditions to be as similar as possible. The achieved ERLE values for both microphone types are shown in figure 5.11(b).

The results are somewhat better for the Shure microphones for lower frequencies, and more or less the same for higher frequencies. This is in spite of the Shure microphones being dynamic (and therefore having a lower sensitivity) and the AKG microphones costing several times as much as the Shure microphones. Thus it turns out that the microphones that are already in use on many stages are also well suited for the crosstalk cancellation methods presented in this work.



(a) Comparison of ERLE different microphone directivities and placements



(b) Comparison of ERLE for AKG microphones (expensive, large-diaphragm condenser microphones) and Shure microphones (low-cost dynamic microphones)

Figure 5.11: Plots of ERLE, with mean and standard deviation, for different microphone directivities and types. A loudspeaker playing white noise was used as the sound source.

5.6 Simulated experiment with the continuous update method

No real-life experiments with the continuous update method were done, but a few experiments were simulated using a synthetic impulse response in MATLAB, as described in section 4.5.5. The goal of these experiments was to study the effect of doubletalk on the adaptation (see also the discussion on doubletalk in section 3.2).

White noise was used as the input signal for the first experiment. This was filtered with the synthetic impulse response to create a room microphone signal, but sections of electric guitar were added to the room microphone signal to simulate doubletalk. The system distance of the filter and the envelope of the error signal were plotted to investigate the effects of doubletalk on the adaptation process. Example sound files have also been made to illustrate these effects.

Figure 5.12(a) shows some of the plots resulting from the experiment. The top plot shows the room microphone signal, consisting of noise with three periods of guitar doubletalk added. The middle plot shows the error signal, which is what would go on to the sound mixer. Ideally, this signal should only contain the guitar doubletalk, and no noise. The bottom plot shows the system distance.

From the error plot, the method seems to work reasonably well – we can see that the amplitude of the noise in the error signal decays before the first guitar segment, and that the noise level in between the guitar segments is very low. However, the system distance plot reveals that the guitar doubletalk causes system distance to rise to the same level as is was before adaptation began. In between the doubletalk, the system distance quickly sinks back to its minimum value.

Since the error signal is what we would actually hear, it has been included with this report, as file `contud_wnoise+elguitar_mu_05`. Listening to this, one finds that the noise level quickly decreases after a certain start-up period, as can be seen from the plot. During the guitar segments, the sound is kind of noisy, with some annoying clicking sounds. After the guitar has stopped, one can also hear a kind of reverberation, which was not there to begin with (compare with the original guitar sound, file `elguitar+silence`).

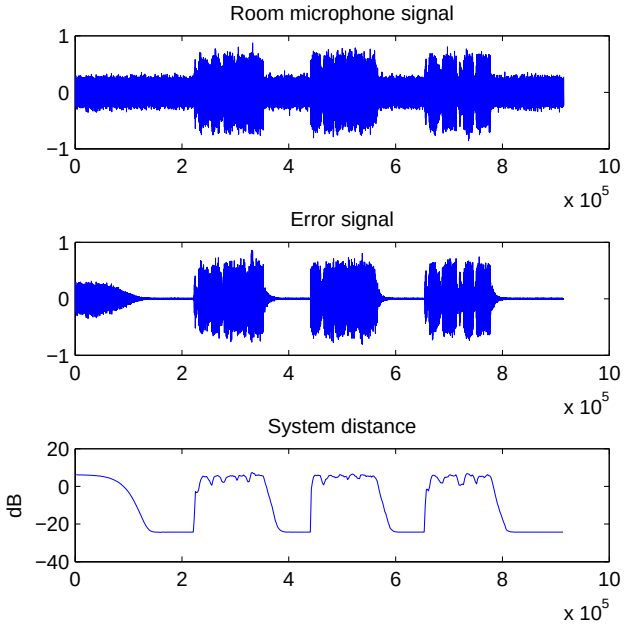
Figure 5.12(b) shows the same kind of plots, but in this case a step size of $\mu = 0.1$ has been used. Here it takes longer for the envelope of the noise to decrease, both before the first guitar segment and in between segments. From the system distance plot one can see that the distance increases during doubletalk segments, but not as high as the case was for $\mu = 0.5$. The system distance also decreases more slowly in between doubletalk segments.

The error signal for this case has been included, as a sound file called `contud_wnoise+elguitar_mu_01`. We can hear that there is no longer any annoying clicking during guitar segments, but the reverberation effect is much more pronounced. This reverberation is probably what we can see as "tails" after guitar segments in the error signal plot in figure 5.12(b).

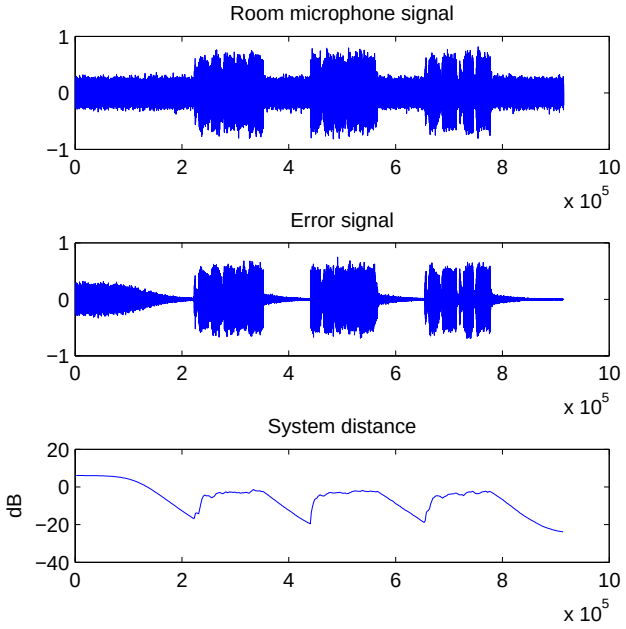
Although is it no surprise that the system distance increases when there is doubletalk, it is very interesting that this "distortion" of the filter segments is not random, but rather shaped by the doubletalk signal. In this way, pure noise which is filtered though the filter actually sounds like the reverberation of the doubletalk sound. Although such reverberation is unwanted for most cases, this is still better than if the filter had been distorted more or less randomly, letting through a lot of noise. This is of course a special case – most musical crosstalk will not be stationary and white like in this case.

A similar experiment was conducted using a guitar signal as the input signal in stead of white noise, and a trombone signal acting as doubletalk. This was done in order to test the effects of doubletalk when two music signals are used, simulating a case more similar to a practical application. The same adaptation parameters were used, with $\mu = 0.5$ and 0.1. The error signals from this experiment are found as files `contud_elguitar+tromb_mu_05` and `contud_elguitar+tromb_mu_01`. For $\mu = 0.5$ we hear that the guitar signal decays quite rapidly, but that it "blows up" again during doubletalk segments, as a result of the doubletalk distorting the filter coefficients. This is not heard as a reverberation effect, but a rather annoying guitar signal (whose amplitude constantly changes) mixed with the trombone.

With $\mu = 0.1$, the envelope of the guitar signal decays a bit more slowly to begin with. During doubletalk segments, the guitar does not "blow up" again in the same way as in the previous example, but, it is possible to hear some of the same effect. When there is doubletalk, the perceived loudness of the guitar increases.



(a) Results with $\mu = 0{,}5$



(b) Results with $\mu = 0{,}1$

Figure 5.12: Plots of room microphone signal, error signal and system distance for an example of the continuous update method with two different values of μ.

5.7 Testing the minimum phase property of the reference microphone impulse response

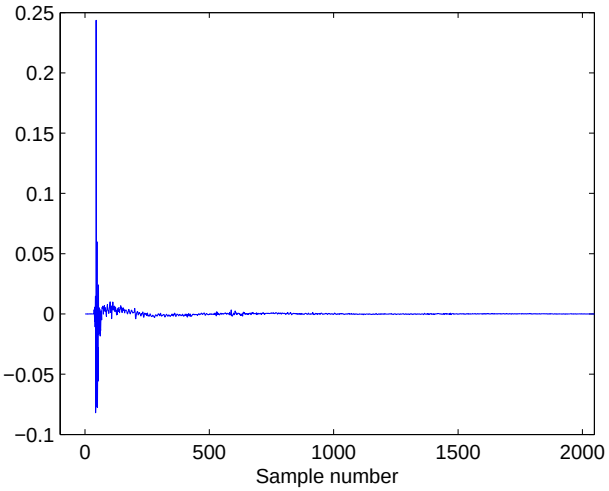
As described in section 4.5.6, an experiment was conducted to test whether the impulse response estimated with "best-case" conditions would fulfill the minimum phase criterion. This is necessary for the inverse of the response to be stable and causal, as described in section 3.1.

A loudspeaker and a microphone were set up with a 30 cm distance, and the impulse response between them was estimated from a 10 second recording of the loudspeaker playing white noise. The resulting impulse response is plotted in figure 5.13(a).

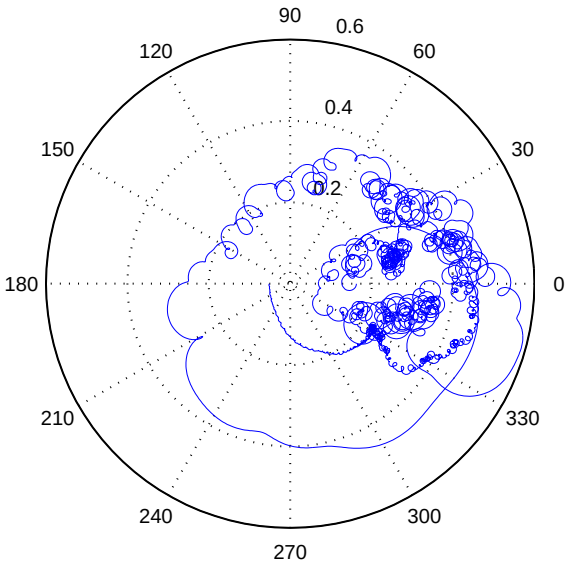
In [16], a Nyquist plot is recommended to determine whether an impulse response is minimum phase or not. Such a plot is a polar plot of the frequency response, with the radius given by the magnitude response and the angle given by the phase response. According to the Nyquist criterion [16], the Nyquist plot will encircle the origin once for each zero of the filter which lies outside the unit circle. Since the definition of a minimum phase system is that no zeros lie outside of the unit circle, the system is identified as minimum phase only if the Nyquist plot does not encircle the origin.

The Nyquist plot of the calculated impulse response is shown in figure 5.13(b). Clearly, the plot encircles the origin, so the impulse response is not minimum phase. As mentioned in section 3.1, this means that the inverse of the impulse response can not be guaranteed to be stable and causal, and thus a mic-to-mic impulse response between a reference microphone and another microphone can not be assumed to be either. This suggests that a *perfect* crosstalk cancellation is not theoretically feasible for the cases that have been investigated in this work, but as experiments have shown, substantial reduction of crosstalk is possible in many cases.

Note also that the impulse response that was analyzed was that of the total system of loudspeaker, room and microphone. Ideally, the effects of the loudspeaker and the microphone should have been included, since it is the room impulse response that should be tested for the minimum phase property. There is a chance that the loudspeaker and microphone contributed to the minimum phase criterion not being fulfilled, but this was not investigated further in this work.



(a) Impulse response including loudspeaker, room and microphone.



(b) Nyquist plot of the impulse response

Figure 5.13: Time domain and Nyquist plot of an impulse response measured in the acoustic booth. The Nyquist plot encircles the origin, indicating that the impulse response is not "minimum phase".

5.8 Comparing noise input signals and their convergence rates for adaptive algorithms

As described in section 4.5.7, an simulated experiment was conducted to study the convergence rates of three different kinds of noise; white noise, pink noise, and white noise which has been played by a loudspeaker and recorded with a microphone in the acoustic booth. Each of these were filtered with a synthetic impulse response to create a room microphone signal, and then the FBLMS algorithm was used, with identical parameters, on each signal pair.

The magnitude specta of each of the noise signals were calculated. These are plotted in figure 5.14. The "raw" white and pink noise spectra are as expected, but the spectrum of the recorded white noise is non-flat. This is probably due to a combination of microphone, loudspeaker and room properties.

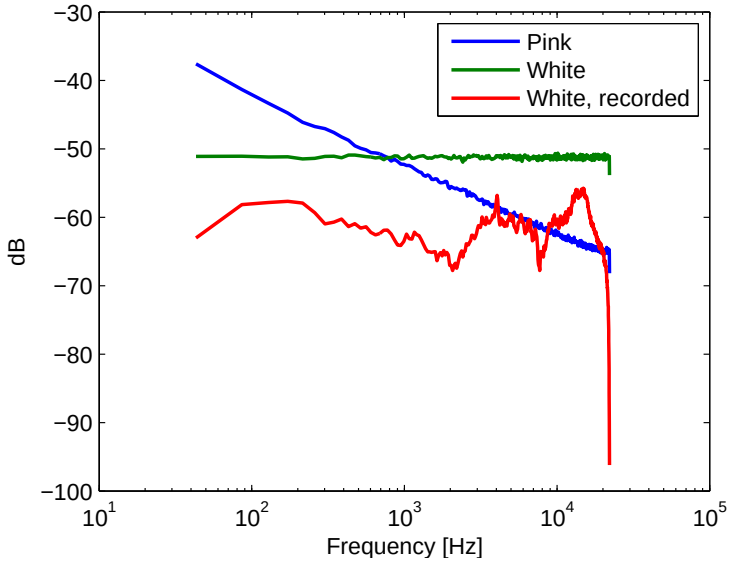


Figure 5.14: Magnitude spectrums of different noise types: Pink, white, and white which has been played through a loudspeaker in the acoustic booth and recorded with a microphone.

Three different length segments were used for adaptation on each kind of noise signal, one of 10 seconds, one of 30 seconds and one of 60 seconds. ERLE values were calculated from the impulse response estimates resulting from each of the segments, and these are all plotted in figure 5.15. This was done to study the achieved degree of convergence for different length input signals for each of the noise types.

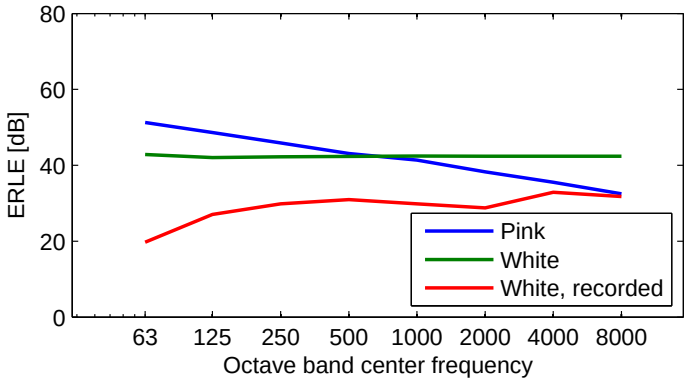
Figure 5.15(a) shows the ERLE values after the 10 second sequence. Here we see that ERLE values are more or less proportional to the magnitude spectra of

the signals - pink noise has higher ERLE values in the lower frequency bands, while the white noise has equal values in all bands. The ERLE values of the recorded white noise are generally lower than those of the raw white noise, and are not as "flat".

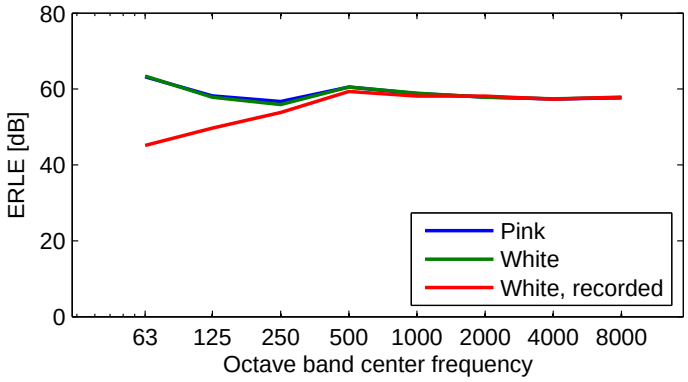
Figure 5.15(b) shows the ERLE values after the 30 second sequence. Here, both the raw noise signals have reached a steady state, and have equal ERLE values in all frequency bands. The recorded noise has a reached the same levels for the highest frequency bands, but has progressively lower values for lower frequencies.

Figure 5.15(c) shows the ERLE values after the 60 second sequence. It is obvious that all three signal have yielded the same results in this case. The reason for the adaptation stopping at about 60-70 dB is probably due to the truncation of the impulse response estimate – the adaptive filtering can not compensate for the last reverberation tail outside the impulse response estimate "window".

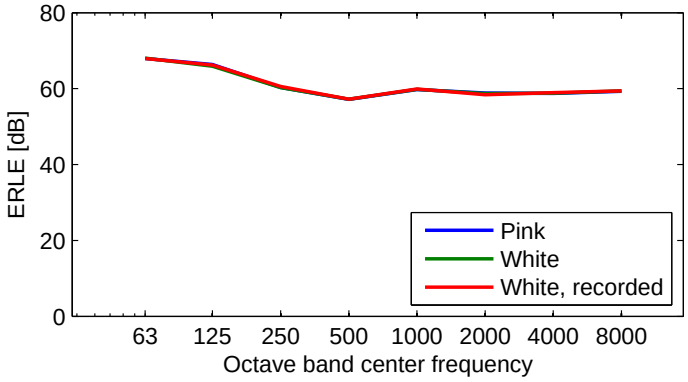
These results show that given enough time, the adaptative process will yield the same result, independent of which input signal is used – but if the process is aborted before it has converged properly, ERLE values are quite different for different input signals. It also seems that in general, the recorded white noise has a lower convergence rate than the other noise types – especially for the lowest frequencies.



(a) ERLE after 10 seconds



(b) ERLE after 30 seconds



(c) ERLE after 60 seconds

Figure 5.15: Comparison of ERLE using different kinds of noise and 10, 30 and 60 second sound clips for adaptation. The first plot illustrates how the adaptive algorithm will reduce the overall error by reducing the most crosstalk in the frequency bands containing most energy. The second plot shows how the convergence rate is slow for the lowest frequencies when recorded white noise is used. The last plot shows that all input signals will yield the same result when the process has converged completely.

5.9 Evaluation of the method used to calculate ERLE

The ERLE values which have been presented in the preceding sections have been calculated in the following way (see also sections 2.5 and 4.5.8): A recording of the reference signal $x(n)$ and the room microphone signals $d(n)$ have been fed to an adaptive algorithm, and used to estimate the impulse response between the two microphones. Then, this impulse response has been used to filter the reference signal, and this filtered signal ($\hat{d}(n)$) has been subtracted from the room microphone signal ($d(n) - \hat{d}(n)$), to cancel out crosstalk. The difference in energy between the crosstalk-reduced signal and the original room microphone signal has been used to calculate the ERLE values.

A small experiment was conducted to investigate if computing ERLE from the same clip that was used for the impulse response estimation would give any bias in the results. Ideally, one should be able to use another recording of the reference and room signals, together with the impulse response estimate, and calculate the same ERLE values. The ERLE values should be a general measure of performance, and not specific to one particular sound clip.

Both approaches mentioned above were tried on recordings which were done in the acoustic booth. This was done for four different sound sources; a loudspeaker playing white noise, a snare drum, an amplified electric guitar and a singing person. Recordings of these sources was done in six different microphone positions, and ERLE values were computed as described above for each position. The difference between ERLE values computed for the same segment and for two different segments was calculated for each position, and the mean of these differences was calculated. These mean values are shown in figure 5.16. Standard deviations were also quite high (about 3-6 dB), but these were not plotted in order to make the plot easier to read.

In the case of the drum and the electric guitar, it is clear that both ways of calculating the ERLE values yield practically the same result. For the white noise, there is quite a large difference – using a different clip for computing ERLE seems to give lower values in general. For the person singing, the opposite is the case: For several frequencies there is a negative difference, meaning that the ERLE values computed for a different clip are actually higher.

The ERLE values that are computed should be representative for the actual damping – not only of the clip that is used for impulse response estimation, but also for all other sounds coming from the same sound source. The results given here are not conclusive – it seems that using the same clip for estimation and calculation of ERLE may overestimate the damping for some cases, and underestimate them for others. As mentioned, variations were also quite high,

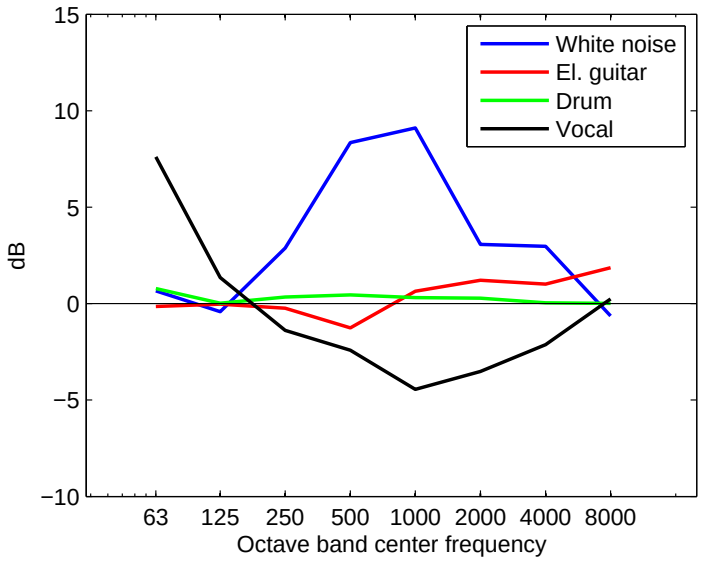


Figure 5.16: Mean difference between ERLE computed from the same clip used for impulse response estimation, and ERLE computed from the clip following that.

making the results less reliable. To get a better impression of what the "real" ERLE values really are, they should be calculated for a large number of segments for each impulse response estimate – but this is beyond the scope of this work. But using the results that have been produced, one must bear in mind that these may not be representative for all possible cases.

6
Discussion

In this chapter, the results of the preceding chapter will be discussed. The discussion is segmented into sections. First, section 6.1 discusses the choice of the FBLMS algorithm for the adaptive filtering, and also some of the characteristics of this algorithm. White noise was chosen as a reference signal for this work, and in section 6.2 the results obtained with this signal are discussed. Following this, in section 6.3, is a discussion of the ERLE values calculated for all other sound sources that have been used in this work. The microphone directivity was found to influence the performance of the crosstalk cancellation method, and the result of the experiments regarding this are discussed in section 6.4. A simulated experiment was also conducted to study the effect of doubletalk on the continuous update method, and the results of this are discussed in section 6.5. Although most of the results in this report are expressed in terms of plots and ERLE values, it is what is actually heard that counts in the final application. In section 6.6, various perceptual effects of adaptive filtering and impulse response estimation are discussed. Finally, section 6.7 lists areas of possible future research.

6.1 Choice of adaptive algorihm

In section 5.1, both the NLMS and the FBLMS algorithms were tested, using a synthetic impulse response in a simulated experiment. The NLMS algorithm was chosen rather than the LMS algorithm, because of its useful modification: The normalization makes it easier to choose a suitable value for the step size, and it also makes the convergence rate less variable in the case of a non-stationary input signal. The FBLMS algorithm was chosen both because of its block processing, its high computational effectivity (due to convolution done by multiplication in the frequency domain), and also its possibility of a frequency-normalized step size (as mentioned in section 2.3.6). These two algorithms were seen as the best possible choice for the application of adaptive techniques in this work, within the non-block and the block domain, respectively. Other, slightly more sophisticated algorithms, like the AP and RLS algorithms [8] were also looked into as alternatives for this application, but they were found

to be too computationally complex - experiments with these algorithms could not be done with the available equipment.

In the first experiment, using white noise as the input signal, the NLMS algorithm was seen to converge about 3 times as fast as the FBLMS algorithm. The rate of convergence was also constant, for both algorithms - this is probably due to the special qualities of the input signal. The results are consistent with what was claimed in section 2.2.3 – since all the eigenvalues of the correlation matrix of white noise are equal, all the corresponding filter weights have an equal convergence rate, and thus the overall convergence rate is constant. The faster convergence rate of the NLMS algorithm is no surprise, as the NLMS algorithm updates its filter coefficients each sample, while the FBLMS algorithm updates them only once for each block (in this case once per 512 samples). The surprise in this case is actually that the convergence rate of the FBLMS algorithm is as high as it is - this is probably due to its more accurate gradient estimate (see section 2.3.4).

In the second experiment, a recording of an acoustic guitar was used as the input signal. The NLMS algorithm also converged faster to begin with, but after a short time the convergence almost stopped, leaving the system distance at an approximately constant value. The system distance of the FBLMS algorithm stayed at an almost constant level at the beginning of the adaptation. This was probably due to the initial values of the power spectrum estimate that normalizes the step size for each frequency bin – if the initial values are too high compared with the signal's actual power spectrum, the effective step size becomes very small. The recursion used in the estimate of the power spectrum causes a delay before the values of the power spectrum are properly adjusted, and the effective step sizes become large enough for the convergence rate to rise. The system distance then drops, and stabilizes at an almost constant level about 3 dB below that of the NLMS algorithm.

The fact that the FBLMS algorithm reaches a smaller system distance is probably due to the decorrelating (or "whitening") effect of the frequency-normalized step size (see section 2.3.6). Although the effect is not very large in this case, it still demonstrates that the FBLMS algorithm may perform better than the NLMS algorithm when correlated input signals are used.

Both algorithms also exhibited a sudden change in convergence rate – the convergence was fast to begin with, but then the system distance stopped at a almost constant level, as mentioned above. This is probably due to a large eigenvalue spread in the correlation matrix of the guitar signal. As mentioned in section 2.2.3, filter weights corresponding to relatively large eigenvalues will have fast convergence rates, while weights corresponding to smaller eigenvalues converge slower. So for both algorithms, the fast drop in system distance is probably an effect of the fast estimation of some filter weights, while the following almost constant system distance is due to the very slow convergence

rate of other weights.

In addition to this, the experiment also showed how computationally effective the FBLMS algorithm is compared to the NLMS algorithm. Since the FBLMS algorithm also seemed to perform better for correlated signals, it was chosen as the "default" algorithm for the rest of the experiments.

6.2 White noise measurements

A loudspeaker playing white noise was chosen as the reference sound source for experiments. This was both because white noise is the ideal input signal to an gradient search algorithm (as stated in section 2.2.3), but also because the loudspeaker is immobile, and experiments with it are easily repeatable, as argued in section 4.2. Experiments were done with this sound source both in the acoustic booth and in the ensemble room at the conservatory.

6.2.1 Comparison with theoretically achievable ERLE values

Knowing the number of filter weights, the sampling frequency, and the reverberation time of the room, it is possible to calculate an approximate value for the maximum achievable ERLE in that room, as described in section 2.5. In the acoustic booth, the sampling frequency was 44,1 kHz, the filter length was $N = 8192$, and the reverberation time was approximately 0,1 seconds. According to equation 2.49, the maximum ERLE value should be

$$\text{ERLE}_{max} = 60 \cdot \frac{N}{T_{60} \cdot F_s} = 60 \cdot \frac{8192}{0{,}1\text{s} \cdot 44100\text{Hz}} \approx 111\text{dB} \tag{6.1}$$

Similarly, for the ensemble room at the conservatory, the sampling frequency was 44,1 kHz, the filter length was $N = 16384$, and the reverberation time was approximately 0,7 seconds. This makes the maximum theoretically achievable ERLE

$$\text{ERLE}_{max} = 60 \cdot \frac{N}{T_{60} \cdot F_s} = 60 \cdot \frac{16384}{0{,}7\text{s} \cdot 44100\text{Hz}} \approx 32\text{dB} \tag{6.2}$$

In reality, the maximum achieved ERLE values for the white noise source were about 35 dB (for the 250-1000 Hz octave bands) in the acoustic booth and 27 dB (for the same bands) in the ensemble room at the conservatory, as shown in figures 5.2(a) and 5.6. Values were somewhat lower to both sides of this frequency range. Comparing with the theoretical values, it seems obvious that the theoretical calculation fits best for the ensemble room, which had a

relatively long reverberation time compared with the acoustic booth. Possible reasons for this include

- An inaccurate value for the reverberation time. When the reverberation time is very short, equation 2.49 is very sensitive to inaccuracies. If the actual reverberation time in the acoustic booth was somewhat higher, the theoretical maximum value for ERLE would have been several dB lower.
- The adaptation process may have been stopped before it had converged properly. If there were filter coefficients which converged significantly slower than others (see section 2.2.3), the adaptation process may have been judged as completely converged, when it was actually just converging very slowly. This is discussed further in section 6.2.5.
- The envelope of the actual impulse response may not have an exponential decay, which is assumed for the theoretical calculation. Also, as it was pointed out in section 3.1, a mic-to-mic impulse response is a convolution of the impulse response from the sound source to the room microphone and the *inverse* of the impulse response to the reference microphone. Although a room impulse response can be assumed to have an exponential decay in general, the same can not be said about the inverse of such a response.
- External mechanisms may limit the maximally achievable ERLE, independent of the room reverberation time and the filter length. Such mechanisms may include background noise and time variance.

6.2.2 Effect of background noise

The signal-to-noise ratio (SNR) was measured for the white noise source in both rooms, and the levels turned out to be approximately the same for both rooms. This is seen by comparing figures 5.4 and 5.9. The 63 Hz octave band had the lowest SNR level, about 35 dB. Higher frequency bands had increasingly higher levels, and above and including the 500 Hz band, the SNR was at a constant level of about 70 dB. The low SNR levels at low frequencies may be a result both of high levels of background noise, and of the loudspeaker radiating sound less effectively in this frequency range.

The ERLE values for the white noise source were generally lower in the lowest octave bands (63 and 125 Hz). This can probably be explained by the low SNR levels at the low end of the frequency range, since a high level of background noise will interfere with the adaptive process, limiting the achievable ERLE. But in addition to this, ERLE values also became gradually lower in the highest octave bands (2000 to 8000 Hz), for both rooms. One possible explanation to this may be time variance.

6.2.3 Effect of time variance

With the learn and freeze method, the impulse response is estimated through a learning sequence. The estimate at the end of this sequence is then used for crosstalk cancellation of the whole sequence, and ERLE values are calculated from this. If there is some time variance during this measurement, the algorithm will adapt to the changes, and thus the final estimate will be more accurate for the end of the sequence than for the beginning. Thus, a time variance within the measurement period may give rise to lower ERLE values.

Time variance in a room may have several causes; for example temperature changes (which affects the sound speed), air movement, movement of microphone or loudspeaker, movement of other objects in the room, heating of the loudspeaker's voice coil, sampling frequency drift in hardware, etc. [20]. Since these measurements were done in a closed room with no people in it, and both the loudspeaker and the microphone were mounted on stands, air movement and temperature changes are assumed to be the most probable causes of time variance in this case.

A change in the system due to time variance will cause an error in an estimation of an impulse response, but the impact of this error is often dependent on frequency. For example, a small displacement of a microphone may be irrelevant as far as low frequencies are concerned, while at the same time it is on the scale of a wavelength at higher frequencies. For this reason, time variance is mainly a problem at higher frequencies. This is illustrated in many articles, for example [15].

The fact that time variance often has a progressively larger impact for higher frequencies makes it a plausible candidate for a mechanism to reduce ERLE at higher frequencies. If the background noise has a "high-pass" effect on ERLE, while the time variance has a "low-pass" effect, this would explain the "band-pass" distribution of the ERLE values across the frequency range, with the maximum ERLE values in the mid-range.

6.2.4 Effect of different distances to room microphones

Measurements in the ensemble room using white noise were also done for two different distances between reference microphone and room microphones. The ERLE values for both cases were plotted in 5.6 for comparison. The results for the different distances seemed to indicate a "shift" effect - for the 63-250 Hz octave bands, the 2 meter distance resulted in the highest ERLE values, while the 1.2 meter distance gave higher ERLE values in the higher frequency bands. One would perhaps expect the 2 meter distance to result in slightly lower ERLE values overall, since the power of the direct sound to the reverberant sound should be somewhat less in this case – but this does not explain the results in

the lower frequency bands. Perhaps this is a result of a near-field vs. far-field effect, but no real explanation has been found for this effect.

6.2.5 Effect of the room on the input signal

In the experiment described in section 5.8, the convergence rates of different noise types were compared for a synthetic impulse response. The noise signals were pink and white noise, and also white noise which had been played through a loudspeaker in the acoustic booth, and recorded by a microphone positioned 30 cm from the speaker. When the adaptation process was stopped before it had converged, the shape of the resulting ERLE values was shown to be approximately equal to the spectrums of the input signal: The white noise signal gave approximately equal values in all octave bands, while the pink noise gave higher values in the lower frequency bands. This is a natural effect of the LMS algorithm trying to reduce the overall (or full-band) error – if there is more energy at some frequencies than others, the algorithm uses more of its resources on reducing the error at these frequencies, since this will minimize the error over the entire frequency range.

The experiment also revealed that the convergence rate of the recorded white noise was considerably slower that that of the "raw" white noise – especially at low frequencies. This seems to indicate that the collective effect of the loudspeaker, the room and the microphone makes the signal less suitable as an input signal. Most probably this is caused by the reverberation of the room, which introduces correlation into the signal. The reverberation time of the acoustic booth was also longer at the lowest frequencies in the acoustic booth (see figure 4.3), and this may be the reason why the convergence rate was extra slow at the lowest frequencies.

During experiments using white noise, the algorithm was assumed to have converged when the envelope of the error signal seemed to have reached a constant value. It may be that the algorithm had not always converged for the lower frequencies, but that this was not visible, because 1) the convergence rate was so slow, and 2) the narrow bandwidth of low-frequency bands makes changes in these bands less visible when the full-band signal is inspected visually. If the adaptive process was actually aborted before the filter was fully converged for the low frequencies, this may account for some for the low ERLE values found at lower frequencies (together with background noise, as mentioned above).

6.3 ERLE values for musical instruments

The learn and freeze method was used on a number of musical instruments, both in the acoustic booth and in the ensemble room at the conservatory. Looking at the results, it seems that several factors may influence how well the crosstalk cancellation works when musical instruments are used as sound sources. In this section, we will discuss some of these, and also suggest possible improvements to the method.

6.3.1 Effect of the room

Unfortunately, the same instruments were not recorded in both the acoustic booth and the ensemble room, making it harder to compare results from the two rooms directly. The only exception is the acoustic guitar, if one ignores the fact that the guitars used in the two different rooms were slightly different, and that the player was not the same in each case. Comparing the results for the guitar (figure 5.2(c) and 5.8(c)), one finds that the ERLE values are actually slightly higher for the experiment in the ensemble room. This is a surprise, as the ensemble room should represent a less ideal case for the adaptive algorithm, with its far longer reverberation time. Comparing some of the other results as well, it does not seem like the results from the ensemble room are considerably worse than those from the acoustic booth. As long as the filter length and adaptation time are adjusted to the reverberation time, it seems that the crosstalk cancellation may work approximately equally well in very different rooms.

6.3.2 Effect of the magnitude spectrum

For most of the musical instruments, there seems to be a correlation between the magnitude spectrum and the ERLE values – the values are high in frequency bands where the magnitude is high. One possible reason for this is that the signal-to-noise ratio is also high in these frequency bands. Another may be that the adaptive algorithm prioritizes the reduction of crosstalk in the frequency bands which contain the most energy, as was seen in the experiment with pink noise as the input signal (described in section 5.8 – the adaptive algorithm uses more of its resources (the filter coefficients) in reducing crosstalk in bands which contain more energy, since this will minimize the total error. See also the discussion in section 6.2.5).

When the ERLE values are not equal in all frequency bands, this means that some frequency bands of the crosstalk are damped more than others. A sound example of this was given for the acoustic guitar (see section 5.2.8), where there was damping mainly in the 125 - 500 Hz bands. Although the magnitude

spectrum tells us that this is the frequency range where the magnitude is highest (thus making this the frequencies where there is most to gain by damping), the residue after damping has an "unnatural" spectrum where the high-frequency components are dominant. See also the discussion in section 6.6.1.

6.3.3 Effect of the playing style

At the conservatory, measurements were done with the musicians using two different playing styles - termed "calm" and "fast". Compared with the results for the calm playing style, the fast style seemed to result in slightly higher ERLE values for all instruments – especially in octave bands where crosstalk cancellation was already substantial. In some cases, the different playing styles also resulted in different magnitude spectra – this was particularly evident in the case of the clarinet (see figure 5.7(e) and 5.7(f)). If the fast playing style brought more energy into some frequency bands, this may in part explain why the ERLE values became higher in these bands - a higher signal-to-noise ratio may improve the impulse response estimation. For other instruments the ERLE values became higher even though there was little change in the spectra. In these cases, it may be that a faster playing style results in a less correlated input signal, thus improving the impulse response estimation.

6.3.4 Effect of directivity and time variance

Time variance has already been mentioned in section 6.2.3 as a possible reason for poor ERLE values, especially at high frequency. When the sound source is a musical instrument, which is handheld by the player, another source of time variance is introduced: The movement of the player. This effect should also be all the more pronounced if the instrument exhibits a large degree of directivity, since a small displacement of the instrument could greatly influence how sound is radiated into the room. The directivity at a given frequency may also change depending on how the instrument is played (fingering, etc.). In general, musical instruments are close to omnidirectional at low frequencies, while they may exhibit complex directivities at higher frequencies [6]. This may be part of the explanation of low ERLE values at higher frequencies for several instruments.

Similarly, the size of the instrument may also affect the degree of time variance. If the instrument is large, a small movement of the instrument may correspond to substantial changes of the impulse response.

For example, the bassoon turned out to have very low ERLE values, particularly for the calm playing style. This may be explained by it being subject to a large degree of time variance – the fact that sound may be radiated

from all holes along the instrument, creating a complex directivity, together with the considerable length of the instrument, suggests that this might be the case.

One possible opposite of this example may be the results obtained from a male person singing. For this source, ERLE values were quite high over several octave bands. A possible reason for this is that the human mouth is a "simpler" source: It is smaller and easier to hold still, thus making it less vulnerable to time variance.

6.4 Microphone directivity

Since it was possible to change the directivity of the AKG microphones that were used for most measurements, an experiment could be conducted to study the effect of the directivity on the performance of the learn and freeze method. A loudspeaker playing white noise was used as the sound source. The results revealed that a cardioid directivity gave mean ERLE values that were a few dB higher in most frequency bands, compared with an omnidirectional directivity. This was the case when the microphone was turned towards the loudspeaker – but the results with the cardioid setting were also slightly better when the room microphone was turned 90 degrees away from the loudspeaker. This was mainly in the low frequency bands.

Using a directional microphone in stead of an omnidirectional one, and pointing it at the sound source, will increase the level of the direct sound to the ambient sound (sound coming from other directions, like reverberation). Correspondingly, the reverberation time is effectively shorter when a directional microphone is used, making a mic-to-mic impulse response simpler. This is probably the reason why the cardioid directivity resulted in higher ERLE values. The sensitivity of a cardioid microphone at 90° from its axis is approximately 6 dB less than on-axis. Even though this means a reduction of the direct sound level when the room microphone is turned 90° from the sound source, the results seem to indicate that the overall effect of the directivity is still positive.

Microphones designed for use on a stage are usually directional, most often with a cardioid or supercardioid directivity. This helps reduce crosstalk and reverberation levels. The fact that the active crosstalk cancellation also works better with directional microphones indicates a "win-win" situation: The directivity helps reduce crosstalk both passively and actively (through making the impulse response easier to estimate). The fact that relatively cheap, standard stage microphones (Shure SM 57 and 58) turned out to yield slightly better results than expensive cardioid microphones (AKG 414) is also promising for the practical application of crosstalk cancellation.

6.5 Simulated experiment with the continuous update method

A simulated experiment was done to study the effects of doubletalk on the continuous update method. The input signal was filtered with a synthetic impulse response to create reference and "desired" (or "room microphone") signals for the adaptive algorithm, but short segments of another signal was added to the desired signal, simulating doubletalk. The results of the experiments, which are described in section 5.6, illustrated some interesting effects:

When white noise was used as the main input signal, with an electric guitar as the doubletalk signal, the doubletalk distorted the filter in such a way that it produced a reverberation-like effect for the guitar. The noise was "shaped" in such a way that the output of the filter sounded harmonic and "guitar-like". The effect was more pronounced for a small step size ($\mu = 0{,}1$) than a larger one ($\mu = 0{,}5$). The system distance was also seen to rise during doubletalk segments.

This is probably an effect of how the filter coefficients are updated (see equation 2.17). The filter update uses both the reference signal $x(n)$ and the error signal $e(n)$. Since the $x(n)$ signal is pure noise in this case, there is no doubt that it is the error signal which causes the effect.

In periods of doubletalk, the error signal will consist of two parts: The residue of crosstalk that has not been cancelled (white noise in this experiment), and the doubletalk signal (the guitar). In these cases, the doubletalk dominates the error signal, and interferes with the filter update process. The filter update then represents a kind of feedback path for this doubletalk – not a direct feedback path, but an indirect one, by letting the filter coefficients become similar to the doubletalk. When white noise is fed through the filter, the output is perceived as reverberation. The impression is also enhanced by the effect that when the doubletalk stops, the filter is able to gradually adjust back to its correct values, thus also gradually reducing the guitar-like sound. This is perceived as a "reverberation tail".

When a guitar signal was used as the input signal, and a recording of a trombone was used as doubletalk, it was not possible to perceive the same kind of reverberation-like effect. In stead, it seemed like the filter distortion caused by the doubletalk only degraded the crosstalk cancellation, so that the perceived level of crosstalk rose. This problem was smaller with a smaller step size μ. The reason why there was no perceived reverberation effect for the last case may be the special nature of white noise: It is stationary, with a constant amplitude and a flat spectrum. Such a signal may be much easier to "shape" to create a reverberation-like effect than a guitar signal, which is nonstationary,

with a constantly changing amplitude and spectrum.

In the first experiment, with white noise as the input signal, a large step size made the system distance go back to approximately its initial value during doubletalk periods - but the fast convergence rate made the "reverberation tail" after these segments quite short. With a smaller step size, the system distance did rise during doubletalk segments, but not as high as for the larger step size. The relatively slow convergence rate also made the reverberation effect after doubletalk segments more pronounced.

In the second experiment, where a guitar signal was used as the input signal, there was no perceived reverberation effect. In this case, the smaller step size lessened the impact of the filter distortion, making the overall result better than when a larger step size was used.

This second experiment is closer to a "real" application of the continuous method, since two musical instruments are used for the input and doubletalk signals. Although the first experiment illustrated the possibility of a pronounced reverberation effect for small step sizes, it seems that a small step size will yield the best results in "real life". As was mentioned in section 3.2, the continuous method method would also benefit greatly from applying a double-talk detector, which can slow down or stop the adaptation in case of doubletalk. Using this, this method may well be an alternative to the learn and freeze method in a real application.

6.6 Perceptual effects

6.6.1 Frequency-dependent damping

In section 5.2.8, two sound examples were presented, illustrating the use of the learn and freeze method on the recordings of two different instruments: An acoustic and an electric guitar. A speech sample was also added to the results, to illustrate how the guitar signals acted as crosstalk, and how this changed when the crosstalk cancellation was used.

The example with the electric guitar illustrated a quite successful crosstalk cancellation, with substantial damping in most frequency bands. While the speech was more or less unintelligible before the crosstalk cancellation, it is clearly heard afterwards. In the example with the acoustic guitar, the crosstalk cancellation worked mainly for lower frequencies. The perceived result was mostly that of having "turned down the bass", rather than reducing the overall level of the crosstalk. This raises a question: If one has to accept some crosstalk, what sounds best to the listener: The original (natural-sounding) crosstalk, or a "high-pass" version? This will be one of many challenges in a practical implementation for crosstalk reduction.

6.6.2 Frozen filter coefficients leading to fluctuation in crosstalk cancellation

A special effect was observed in two of the sound examples that were presented in section 5.2.8: There was a fluctuation in the residue after crosstalk cancellation. This was perhaps most easily heard in the sound example illustrating the use if the learn and freeze method "in real life", using a loudspeaker playing white noise as the crosstalk-generating sound source, and a male voice as the second source. The suggested explanation was that the actual impulse response in the room was changing while the impulse response estimate used for crosstalk cancellation was kept constant (or "frozen"). Such a change of the impulse response was most likely caused by movement of the person, perhaps combined with movement of the air in the room etc. In this way, time variance in the room not only interferes with the learning process, as discussed in sections 6.2.3 and 6.3.4, but also reduces the effect of the crosstalk cancellation after the filter is "frozen".

The crosstalk cancellation method basically relies on adding a opposite-phase model of the crosstalk to the already existing crosstalk. The difference in phase between the crosstalk and the model must be minimal for the crosstalk cancellation to work. If the model has a phase shift, for example caused by a time delay, the effect of the crosstalk cancellation is severely reduced. Because of the shorter wavelength in the air, the crosstalk reduction at high frequencies will be more vulnerable to movements in the room than it is at low frequencies, since a small change will represent a relatively larger change in phase for a high-frequency sound than for a low-frequency sound. If, for example, there is a change in the room (for example a person moving) representing a shift of 3.4 cm, this is equal to half a wavelength at 5 kHz but only a hundredth of a wavelength at 100 Hz. The higher sensitivity to changes is probably the reason why the fluctuation in the crosstalk residue was mainly heard for higher frequencies.

During the experiment, the person was trying not to move at all. Still, the fluctuations in crosstalk residue could be heard, and this is to be expected, since even movements on the scale of centimeters may severely reduce the effect of the crosstalk cancellation. This illustrates the weakness of the learn and freeze method – since there will always be some degree of movement in the room, the effect of the crosstalk cancellation will vary with time. A method based on continuous update in the crosstalk cancellation filter may be able to track such movements and keep the reduction of crosstalk at a more constant level.

6.6.3 Filter length

Intuitively, the mic-to-mic impulse response estimate used in the crosstalk cancellation should be as long as possible, to include as much of the reverberation tail as possible. But as was illustrated in section 5.4, in some cases of poor impulse response estimation, the estimate will contain a noisy tail which does not decay with time. The result is an unnatural-sounding reverberation effect. In this case a truncation of the estimate will yield a better result. The consequence of this is that the filter length should be carefully chosen – long enough to provide substantial damping, short enough to avoid the effect just mentioned. The most important parameter when choosing filter length will be the reverberation time of the room. In a practical application, a way to estimate the reverberation independently of the adaptive algorithm would probably be useful for making a suitable choice of filter length.

6.7 Future considerations

6.7.1 Low complexity implementation: log-log LMS

Many approaches have been used to try to reduce the complexity of LMS-style algorithms. Some have tried to just use the sign of the input or the error signals, or both, in the filter update equation ("sign-data", "sign-error" and "sign-sign" algorithms, respectively). This reduces both memory requirements and computational load during adaptation, but also reduces filter performance drastically. In [13], the authors suggest quantizing both input and error signals to the nearest power of 2, so that these signals can be represented in their log_2 form, yielding much shorter word lengths. This algorithm is called the "log-log LMS" algorithm, and has the same kind of advantages as the sign-type algorithms – but the experiments performed in [13] suggest that its performance is very close to that of the original LMS algorithm. Although the signals are quantized, information about their dynamic range is still retained, and the authors claim that this is because important information about the dynamic range of the signals is retained. The complexity of this algorithm is actually also less than that of the sign-data and sign-error algorithms, and the chip area requirements for ASIC implementation are also lower. For an implementation in a sound mixer, reduced complexity and chip area are both very interesting qualities in an algorithm. Although this algorithm has not been tested in this work, it may be interesting for future research in related areas.

6.7.2 Blind signal separation

In this work, only asymmetrical crosstalk case have been studied. If there is only one-way crosstalk, as described in section 3.1, adaptive filtering may be applied. If, on the other hand, there is symmetrical crosstalk – that each in the instruments are more or less equally strong, so that there is substantial crosstalk both ways – adaptive filtering can not be used, as one does not have access to the original signals from both instruments, only a mixture of these. In this case, Blind Signal Separation (BSS) may turn out to be a useful tool.

Blind signal separation techniques are based on separating signals from two independent sources with different probability distributions. Several approaches and implementations exist.

Blind signal separation works very well for a simple mix of two signals. Unfortunately, this is not the case for two instruments playing in the same room, being recorded by two microphones. The sound from each instrument will hit both microphones, along with several reflections from the walls. This is termed a "convolutive mix", and has turned out to be a much harder case for BSS.

6.7.3 Subband processing

All experiments in this work was done at a sampling frequency of 44.1 kHz, in order to span the entire frequency range audible to the human ear. This means that the filters needed for crosstalk cancellation have to be several thousand coefficients long. The filter may be made considerably shorter by performing the adaptive filtering in subbands, since the sampling rate in each band can be reduced [8]. Also, for several of the instruments that were tested, there was little or no crosstalk cancellation in the higher frequencies. If one still wished to reduce the low-frequency crosstalk, it would be possible to perform adaptive filtering in only a low-frequency band, and thus reduce both sampling frequency and filter length.

6.7.4 Measuring input signal quality

During this work, it was obvious that some instruments and input signals were a better "raw material" for the adaptive algorithm than others, yielding better impulse response estimates and faster convergence rates. This is probably partly a result of the signals themselves being less correlated, resulting in a smaller eigenvalue spread of the correlation matrix (see section 2.2.3) – and also partly a result of the physical properties of the instrument and its interaction with the room, resulting in a smaller degree of time variance (as discussed in section 6.3.4).

From this sprang an idea: Can the input signal "quality" be measured in some quantifiable way, without actually performing the adaptive filtering? The effect of physical aspects of the instrument and the room are probably hard to quantify from a recorded microphone signal, but it should be possible to find a measure for the degree of "self-correlation". One possible way to measure this would be to estimate the correlation matrix, and calculate the eigenvalues and the eigenvalue spread from this estimate. This approach would probably be well suited to stationary signals (whose correlation matrices have constant eigenvalues), but if this method were to be used on musical signals, their highly transient and nonstationary nature may present a problem.

Another, and somewhat simpler measure was also looked into – that of the "spectrum flatness". It can be shown [14, section 3.4.5] that the eigenvalue spread of a stationary signal is upper bounded by the dynamic range of the power spectrum;

$$\mathcal{X}(\mathbf{R}) \leq \frac{max\{P_x(\omega)\}}{min\{P_x(\omega)\}} \tag{6.3}$$

where $P_x(\omega)$ is the power spectrum of the signal. The larger the spread in eigenvalues, the larger the dynamic range of the power spectrum. Since a large dynamic range indicates a "non-flat" power spectrum, the "flatness" of the spectrum was suggested as a measure of input signal quality. A flat spectrum should represent a signal with small eigenvalue spread, making it a "good" input signal. This is in agreement with the classification of white noise (which has a completely flat spectrum) as the ideal input signal (see section 2.2.3). Similarly, a spectrum with large peaks and dips should indicate a "bad" input signal.

The flatness of the spectrum can be measured with what is (quite appropriately) called the Spectral Flatness Measure (SFM) [14, section 4.1.1]. This is based on the ratio of the geometric mean of the power spectrum to the arithmetic mean. If the spectrum is completely flat, both mean values will be equal and the ratio will be 1. If the spectrum is shaped in any way (for example with sharp peaks), the ratio will be less than 1. Since both mean values must be positive, the SFM is always greater than 0.

To test the SFM as a measure of input signal quality, a preliminary experiment was conducted: Adaptive filtering was done on a segment, and then the SFM was calculated for the same segment. The SFM and fullband ERLE values were then compared for a number of segments. This was done for both real and simulated measurements (with a synthetic impulse response). The SFM measure was successful in classifying signals as having non-flat spectrums (for example instruments with harmonics resulting in several peaks in the spectrum), but the correlation with the ERLE value was poor, especially for

the real-world measurements. For the real-world measurements, this may be explained in part by effects like time variance in the room, but the fact that correlation was poor also for the simulated case seems to indicate that the flatness of the spectrum alone is not enough to determine what ERLE value can be expected.

Due to the inconclusive result of the preliminary experiment, this was not investigated further, but classification of input signal quality still remains an interesting problem. Finding alternative ways to measure such quality, or looking further into the use of SFM may both be items for future research.

7
Conclusions

In this work, a method to reduce acoustic crosstalk has been investigated. This has been based on estimation of an impulse response between microphones by adaptive filtering. By using the signal from the microphone closest to the instrument producing crosstalk, and filtering it with the estimated impulse response, the crosstalk in the other microphone may be modeled and subtracted.

A version of the Block LMS algorithm was chosen as the main adaptive algorithm for use in experiments. With this algorithm, many of the operations involved in the updating of the filter were performed in the frequency domain, and thus this algorithm is called the Frequency Block LMS (FBLMS) algorithm. A decorrelation of the input signals, by normalization with an estimate of the power spectrum, was also implemented in this algorithm. An experiment was conducted comparing this algorithm with the normalized LMS (NLMS) algorithm. The experiment illustrated that although the convergence rate of the FBLMS algorithm was somewhat slower than that of the NLMS, the FBLMS algorithm was much more computationally effective, and also seemed to yield a smaller final system distance.

Two different methods were also suggested for crosstalk cancellation. The first was termed "learn and freeze", with the name meaning that the adaptive algorithm should first estimate the impulse response with only the crosstalk-producing sound source playing, and then "freeze" the filter coefficients. The resulting crosstalk cancellation filter is then kept constant during performance, when both instruments are playing. The learn and freeze method was used for most experiments, both because it allowed for simpler and more easily repeatable experiments, and because the achieved crosstalk cancellation was easier to measure. Sound examples illustrated that if the crosstalk cancellation filter is kept constant while the actual impulse response in the room is changing, the "residue crosstalk" will fluctuate. Experiments also showed that even very small changes may affect the crosstalk reduction at high frequencies.

The second method that was suggested was called "continuous update", meaning that the adaptive filter should continuously update the filter coefficients while both sound sources are playing. Several challenges of this

method were pointed out. A simulated experiment was also conducted, and this revealed that so-called "doubletalk" (when other sound source are playing in addition to the one producing crosstalk) may give rise to unwanted distortion of the impulse response estimate and corresponding unwanted sound artifacts. The method is still seen to have potential, if a suitable doubletalk detector can be implemented to slow down or stop adaptation during doubletalk segments.

A loudspeaker playing white noise was used as a reference sound source, representing an approximate "best case". In a small heavily damped room, the maximum achieved damping of the crosstalk from this source was approximately 35 dB. This was achieved in the 500-1000 Hz octave bands, while damping was generally somewhat lower in bands above and below this. It is suggested that low-frequency background noise is the cause of less damping in the low-end of the frequency range, while time variance is the cause of less damping in the higher frequencies. Similar experiments in a larger, "ensemble" room resulted in maximal damping of approximately 27 dB, with less damping in the high and low end also in this case. Theoretical calculations of the maximally achievable damping were shown to be much more accurate for the larger ensemble room than for the small, damped room.

An experiment was also conducted to investigate the influence of microphone directivity on the achieved crosstalk cancellation. Results showed that for a white noise source, a microphone with cardioid directivity gave slightly better damping than an omnidirectional microphone – even when it was not aimed directly at the sound source. It is suggested that this is because the ratio of the direct sound to the reverberation is increased. It was also found that two inexpensive, dynamic microphones (Shure SM 57 and SM 58) resulted in slightly better damping than AKG C 414 microphones, which are more expensive condenser microphones.

Using the learn and freeze method, experiments were done for several musical instruments, in both the small, heavily damped room, and the larger ensemble room. Although the same instruments were not tested for both rooms, the results indicate that the achievable damping was quite similar in these rooms. This is consistent with what was found in the experiment using white noise, and seems to suggest that substantial crosstalk cancellation may be possible in a wide range of rooms.

Some of the experiments were done with the players using both a "calm" and a "fast" playing style. The fast style resulted in slightly better crosstalk cancellation in all cases. It is suggested that this may be due to a "fast" musical signal having less correlation, thus making it a better input signal to a adaptive algorithm.

For several of the musical instruments, there was substantial crosstalk cancellation only in a few octave bands. These were most often the bands in

which the instruments were able to radiate the most energy. It was suggested that the increased signal-to-noise ratio in these bands made a higher degree of crosstalk cancellation possible, and also that such high-energy bands are "prioritized" by the adaptive algorithm, to minimize the overall error. Complex directivities and a high degree of time variance are also suggested as reasons for poor crosstalk reduction, especially in high frequency bands.

It was also pointed out that if there is crosstalk reduction in only a few octave bands, the spectrum of the remaining crosstalk will be changed, making it sound "unnatural" in some cases. This problem was also illustrated through sound examples. This poses a possible challenge for a practical application of crosstalk cancellation: What is actually most pleasing to the human listener – loud, but natural-sounding crosstalk, or partly cancelled crosstalk with an unnatural spectrum?

In all, the results of the experiments with crosstalk cancellation were quite variable, depending of the sound source. Experiments with a loudspeaker playing white noise and an electric guitar amplifier both yielded substantial damping, while the damping of some musical instruments was on the scale of just a few dB, often in only two or three octave bands. This indicates that the methods investigated in this work may not be usable for any sound source in a practical application (like a sound mixer), but that much can be gained in some cases. The sound sources that seem to be yield the best results are those that are completely stationary, like a loudspeaker or a guitar amplifier.

A

MATLAB code

A.1 FBLMS implemented as MATLAB function

```
function varargout = flms(x,d,L,mu,varargin)
%FLMS LMS filtering performed in the frequency domain
%
%   H = FLMS(X,D,L,MU) returns an estimate of length L of the system
%   impulse response defined by the input signal X and the output signal D.
%   Block LMS adaptive filtering, performed in the frequency domain and
%   with a block size of L, is used to estimate H. MU is the step size
%   parameter. MU is normalized by the power of the input signal in each
%   block.
%
%   H = FLMS(X,D,L,MU,GAMMA,'fnorm') will in stead normalize MU by en
%   esimate of the power in each frequency bin. Thus, the step size is a
%   function of frequency. GAMMA is the "forgetting factor" of the power
%   spectrum estimate.
%
%   [H,E] = FLMS(X,D,L,MU,...) will also return the error signal E =
%   D-D_EST, where D_EST is found by filtering X with the filter estimates
%   H during adaption.
%
%   Martin Hansen 04.02.2008
%
%   2008, 1. April: Changed from zero-padding in the back to truncation of
%   x and d

%% Check that x and d have the same length
N = length(x);
if N ~= length(d)
    error('x and d vectors must be the same length')
end

%% Set the 'normalized' switch
normalized = false;
if nargin > 5
    if strcmp(varargin{2}, 'fnorm')
        normalized = true;
        gamma = varargin{1};
    end
end

%% Make sure that x and d are column vectors
x = x(:);
d = d(:);
```

```
%% Vector initialization
% x and d are truncated to have a length equal to an integer times L.
r = mod(N,L);
if r                        % If r is not zero, truncation is needed
    N = N - r;              % Update N
    x = x(1:(end-r));
    d = d(1:(end-r));
end

% x is padded in front with a block of zeros to account for "time -1",
x = [zeros(L,1) ; x];

% Preallocate the rest of the vectors
e = zeros(N,1);
y = zeros(N,1);
H_est = zeros(2*L,1);

% Initial power spectrum esimate. The initial value is not set to zero, to
% avoid division by zero. Since the average value of a power spectrum is
% proportional to the FFT length, the initial estimate is also scaled by L.
P = L*ones(2*L,1);

%% For loop
% Index vectors and window vector
x_index = 1:(2*L);
index = 1:L;
first_L_win = [ones(L,1);zeros(L,1)];

if not(normalized)
    for ii = 1:(N/L)
        % Frequency domain representation of X
        X = fft(x(x_index));

        % Filtering done in frequency domain, and transformed to time domain
        y_temp = real(ifft(X.*H_est));

        % Only the L last samples of y_temp are usable bacause of aliasing
        y(index) = y_temp(L+1:end);

        % Calculate error in time domain
        e(index) = d(index) - y(index);

        % Frequency representation of error
        E = fft([zeros(L,1);e(index)]);

        % Correlation between x end e done in frequency domain
        % Only L first samples are used bacause of aliasing
        Phi = first_L_win.*real(ifft(conj(X).*E));

        % Calculate mu normalized by estimate of signal power
        mu_norm = mu/(x(index+L)'*x(index+L));

        % Update filter coefficients in frequency domain
        H_est = H_est + mu_norm*fft(Phi);

        % Update indices
        x_index = x_index + L;
        index = index + L;
    end

else    % "Normalized"
```

```
    for ii = 1:(N/L)
        % Frequency domain representation of X
        X = fft(x(x_index));

        % Filtering done in frequency domain, and transformed to time domain
        y_temp = real(ifft(X.*H_est));

        % Only the L last samples of y_temp are usable bacause of aliasing
        y(index) = y_temp(L+1:end);

        % Calculate error in time domain
        e(index) = d(index) - y(index);

        % Frequency representation of error
        E = fft([zeros(L,1);e(index)]);

        % Update power estimate
        P = gamma*P + (1-gamma)*real((X.*conj(X)));

        % Correlation between x end e done in frequency domain
        % Only L first samples are used bacause of aliasing
        Phi = first_L_win.*real(ifft( (conj(X).*E)./P ));

        % Update filter coefficients in frequency domain
        H_est = H_est + mu*fft(Phi);

        % Update indices
        x_index = x_index + L;
        index = index + L;
    end
end

% Tranform filter coeffs to time domain
h_est_temp = real(ifft(H_est));

% Only L first samples are not equal to zero
h_est = h_est_temp(1:L);

varargout{1} = h_est;
varargout{2} = e;
```

A.2 Function for generating synthetic impulse responses - creexpir()

```
function varargout = creexpir(fs,Nsamp,T60,V,dist,Q)
%
% Creates an impulse response with an ideally exponential decay.
% Relates to the direct sound at 1 m distance.
% If the optional parameter dist is given, a direct sound is added
% as a perfect pulse and the exponential part starts after this.
%
% impres = creexpir(fs,Nsamp,T60,V,dist,Q);

% 971208 Added the directivityfactor Q.
% 11.02.2008    Added possibility to return direct sound delay

if nargin < 6,
    Q = 1;
end
c = 344;
% impres = randn(Nsamp,1);
% t = [0:Nsamp-1]/fs;
% tau = T60/6.91;
expwin = exp(-(0:Nsamp-1)*3*log(10)/fs/T60)';
% clear t
impres = randn(Nsamp,1).*expwin;
clear expwin
scale = sum(impres.^2)/(100*pi*T60/V);
impres = impres/sqrt(scale);
if nargin ≥ 5,
    if dist > 0,
       ncut = floor(dist/c*fs);
        if ncut ≥ 1,
           impres(1:ncut) = zeros(ncut,1);
            impres(ncut) = sqrt(Q)/dist;
        end
    end
end

varargout{1} = impres;
varargout{2} = ncut;
```

Bibliography

[1] Audio Engineering Society. *Software based live sound measurements*, 2006. Convention Paper 6988, presented by Wolfgang Ahnert at the 121th convention of the AES.

[2] Christina Breining, Pia Dreiseitel, Eberhard Hänsler, Andreas Mader, Bernhard Nitsch, Henning Puder, Tomas Schertler, Gerhard Schmidt, and Jan Tilp. Acoustic echo control - an application of very-high-order adaptive filters. *IEEE Signal Processing Magazine*, 16(4), July 1999.

[3] Jun H. Cho, Dennis R. Morgan, and Jacob Benesty. An objective technique for evaluating doubletalk detectors in acoustic echo cancelers. *IEEE Transactions on Speech and Audio Processing*, 7(6), November 1999.

[4] W. T. Chu. Comparison of reverberation measurements using schroeder's impulse method and decay-curve averaging method. *Journal of the Acoustical Society of America*, 63(5), May 1978.

[5] Gregory A. Clark, Sanjit K. Mitra, and Sydney R.Parker. Block implementation of adaptive digital filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(3), 1981.

[6] Neville H. Fletcher and Thomas D. Rossing. *The Physics of Musical Instruments*. Springer-Verlag New York Inc., 1991.

[7] Simon Haykin. *Adaptive Filter Theory*. Prentice-Hall Inc., third edition, 1996.

[8] Eberhard Hänsler and Gerhard Schmidt. *Adaptive Signal Processing: Applications to Real-World Problems*, chapter 3: Single-Channel Acoustic Echo Cancellation. Springer-Verlag Berlin Heidelberg, 2003. Editors: Jacob Benesty and Yiteng Huang.

[9] Shure Inc. Microphone techniques – live sound reinforcement, 2007. http://www.shure.com/ProAudio/TechLibrary/EducationalArticles/ssLINK/us_pro_mics_for_music_sound_ea.

[10] Lawrence E. Kinsler, Austin R. Frey, Alan B. Coppers, and James B. Sanders. *Fundamentals of Acoustics*. John Wiley & Sons, Inc., fourth edition, 2000.

[11] Ole Kirkeby and Phillip A. Nelson. Digital filter design for inversion problems in sound reproduction. *Journal of the Audio Engineering Society*, 47(7/8), July 1999.

[12] Tobias Lentz. Dynamic crosstalk cancellation for binaural synthesis in virtual reality environments. *Journal of the Audio Engineering Society*, 54(4), April 2006.

[13] S. Mahant-Shetti, S. Hosur, and A. Gatherer. The log-log lms algorithm. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 3, 1997.

[14] Dimitris G. Manolakis, Vinay K. Ingle, and Stephen M. Kogon. *Statistical and Adaptive Signal Processing*. Artech House, Inc., 2005.

[15] Sven Müller and Paulo Massarani. Transfer-function measurement with sweeps. *Journal of the Audio Engineering Society*, 49(6), June 2001.

[16] Stephen T Neely and J. B. Allen. Invertibility of a room impulse response. *The Journal of the Acoustical Society of America*, 66(1), July 1979.

[17] John G. Proakis and Dimitris G Manolakis. *Digital Signal Processing*. Prentice-Hall, Inc., third edition, 1996.

[18] Daniël W.E. Schobben. *Real-time Adaptive Concepts in Acoustics*. Kluwer Academic Publishers, 2001.

[19] John J. Shynk. Frequency-domain and multirate adaptive filtering. *IEEE Signal Processing Magazine*, 9(1), 1991.

[20] U. P. Svensson and J. L. Nielsen. Errors in mls measurements caused by time-variance in acoustic systems. *Journal of the Audio Engineering Society*, 47(11), November 1999.

[21] Bernard Widrow and Samuel D. Stearns. *Adaptive Signal Processing*. Prentice Hall Inc, 1985.

[22] Wikipedia. Gradient. Website, May 2008. http://en.wikipedia.org/w/index.php?title=Gradient&oldid=212647663.

[23] Wikipedia. Orchestral enhancement. Website, January 2008. http://en.wikipedia.org/w/index.php?title=Orchestral_enhancement&oldid=185288672.

[24] Audacity homepage. Website. http://audacity.sourceforge.net.

[25] Apogee ensemble homepage. Website. http://www.apogeedigital.com/products/ensemble.php.

[26] Studiobox homepage. Website. http://www.acousticbooth-studiobox.com/.