UNIVERSITY OF CALIFORNIA, SAN DIEGO

A Prototype System for Remote Collaborative Recording

A dissertation submitted in partial satisfaction of the requirements for the degree

Doctor of Philosophy

in

Music

by

Pei Xiang

Committee in charge:

Miller Puckette, Chair
Shlomo Dubnov
Gert Lanckriet
F.Richard Moore
Roger Reynolds
Shahrokh Yadegari

2007

The dissertation of Pei Xiang is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2007

To Jingrong and Nengcai

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

At the moment of completing a PhD program in computer music, I would like to first thank my dearest parents Nengcai Xiang and Jingrong Li for bringing me to planet earth, training me as a musician when I was really young, suggesting me to study signal processing when I was less young, and supporting me while offering their warmest friendship to me when I put my passion on computer music.

I would like to gratefully acknowledge my Chair Professor Miller Puckette for his kind and insightful guidance throughout the years of my PhD program. His patience, flexibility and enlightening advice have always kept my exploration in this field exciting and rewarding. In my preliminary studies for qualify exam, we had many good discussions that strengthened my math foundation, knowledge in realtime audio systems, and echo control. During the dissertation year, many of his extra hours were used to help me realize the final project and refine dissertation writing. Without his hand, my dissertation completion wouldn't haven been possible.

I sincerely thank Professor Shlomo Dubnov for the great time we worked together on signal processing and source separation topics; I feel grateful to Professor F.Richard Moore for his always encouraging smile and resourceful influence on acoustics and sound spatialization; I also feel indebted to Professor Roger Reynolds for his continuous trust, caring, and wholehearted guidance on my composition and artistic sensibility. My gratitude is extended to Music Department faculties who

have designed this unique program, and accepted me as one of the first group of students to experience this nourishing learning process. I am also very grateful to the remaining members of my dissertation committee, Shahrokh Yadegari, Gert Lanckriet, and Joseph Goguen. Their academic support, input and personal cheering are greatly appreciated. Thank you.

During my dissertation year, I was supported by my current employer Qualcomm, Inc. with tuition assistance and flexible working hours. I owe many thanks to my management team, Prajakt Kulkarni, Eddie Choy, Samir Gupta, and Sharath Manjunath, whose understanding and encouragements have made my dissertation writing in parallel with full-time work duties possible.

I also wish to give thanks to folks in Center for Research in Computing and the Arts (CRCA), Peter Otto, Carol Hobson, Ted Apel, David Camargo Cristyn Magnus and Grace Leslie. It has been such an enjoyable time working in this research community. Finally, my gratefulness goes to my friends who offered a lot of generous help during my dissertation writing: Eric Yan, Yinian Mao, Junwen Wu, Song Wang, and Dinesh Ramackrishnan.

VITA

2001	B.S., Electronic Engineering Electronic Engineering Department Tsinghua University (Beijing, China)
2003	M.A., Computer Music Music Department, University of California, San Diego
2001–2007	Researcher Center for Research in Computing and the Arts University of California, San Diego
2005–2007	Audio System Engineer QCT Audio CMX Qualcomm Incorporated
2007	Ph.D., Computer Music Music Department, University of California, San Diego.

ABSTRACT OF THE DISSERTATION

A Prototype System for Remote Collaborative Recording

by

Pei Xiang

Doctor of Philosophy in Music

University of California, San Diego, 2007

Professor Miller Puckette, Chair

This dissertation presents a prototype system to treat the problems of conducting remote collaborative recording without musicians traveling to be local to each other. Two main issues are explored and studied: realtime loudspeaker-microphone feedback control, and the removal of far-end reference signal from recorded audio in the post production. The design of this system includes a smart system integration of existing techniques. Innovative algorithm designs on some other components are also presented. This system is simulated with Matlab and Pd for off-line and realtime analyses, respectively. The results show that it works robustly as an echo canceller against loudspeaker-microphone feedback; it also performs competently in demixing the recorded audio and removing unwanted components generated from the far-end reference speaker.

1

Remote Collaboration: the general context

1.1 Introduction

Collaborative works have always been very attractive in art making. These projects bring talents from different places and areas to achieve goals that are hard to reach without such teamwork. With the development of technology, virtual collaborations become possible in today's world. People don't need to travel physically to participate in some projects. In music making, researchers and musicians have experimented with many ways to present live concerts and conduct recording projects from different locations. In this chapter, the background and some of the challenges involved in remote collaboration systems are briefly reviewed. This is followed by the proposal of a prototype system targeting two of the major issues from a signal processing perspective. In Chapter 2 and Chapter 3, these issues are discussed on a theoretic level. Chapter 4 presents simulations and analyses. The dissertation concludes in Chapter 5 with summary and future

research directions.

1.2 History and Challenges

Researchers in computer music community have experimented with networked performances and other collaborations for over a decade. Early in the mid-1990s, a team at Chukyo University of Toyota in Japan conducted some experiments over ISDN (128 kbps) to teleconference musicians. In 1997, a Remote Music Control Protocol (RMCP) that integrates MIDI and UDP protocols was developed and experimented with to allow users at different machines to play as an ensemble [30]. In 1998, University of California, San Diego and University of Southern California collaborated in the Global Visual Music Project and presented improvisatory jams between Greece and United States [54]. In 2000, multichannel networked concerts were presented at Stanford University's CCRMA using *SoundWIRE*, a software that evaluates the reliability of a network by creating a sonar-like ping and "displaying" the network quality to the ear [22]. In 2001, a Disklavier duet concert, with two pianists 100 miles away collaboratively improvising, was realized between Center for Research in Computing and the Arts (CRCA) in UC San Diego, and Claire Trevor School of the Arts at UC Irvine [1]. MIDI signals as well as live video streams were experienced simultaneously by two groups of audiences. Various new communication protocols such as the Open Sound Control (OSC) [63]

[1] http://www.calit2.net/newsroom/article.php?id=108

have been designed to facilitate control and audio transmissions over the internet. Beside these example projects, there are many more network systems for music surveyed in [8], [28] and [61]. As discovered by a team in Princeton University in constructing their GIGAPOPR network framework [36], networked audio/music isn't as technically difficult as it has been. In the audio (as opposed to control and other) aspect of networked systems, major challenges can be identified roughly as below.

1.2.1 Audio Latency

Compared to performers in a traditional concert, networked musicians mostly have to deal with the experience of listening to the sound of their remote partners with a considerable amount of delay. When the delay exceeds a certain range, difficulties in the collaboration will rise. Audio latency could originate from many places, including software, hardware, network, and acoustics. Apparently the largest contributing factor is the network connection speed. For certain kinds of music, there exist some interesting solutions, such as the NINJAM software [2]. It solves the latency problem by forcing more latency, so that each networked performer performs in sync with the last measure of everybody else. This mechanism makes an internet jam feasible in its own sense. On the other hand, the variety of music made in this way and possibilities of collaboration are greatly limited due to the nature of this solution. For a general-purpose collaboration, the latency

[2] http://www.ninjam.com

still needs to be reduced to a minimum acceptable amount. According to a study by a team in CCRMA [21], sensitive ensemble performance can be supported over rather long paths (about 20 ms). To bring latency into this delay range, the existing infrastructures Internet2, CA2Net and other optically based gigabit networks are already sufficient when dedicated.

1.2.2 Network Robustness

This aspect is mainly a concern for a live concert. Internet is inherently a best-effort transmission system. Protocols such as TCP and UDP have frequent possibilities of losing packet deliveries. Even with re-transmission mechanisms, in a live concert, the untimely arrived data cannot compensate the already compromised quality of sound at the moment. However, this dissertation is mainly focused on remote collaborative recording instead of a live performance. As will be discussed later, the far-end audio is only a reference signal in the recording, which is not used in the post production that happens after. Thus, network robustness and sound degradation due to network transmission appear to be less urgent concerns in the prototype system.

1.2.3 Echo Handling

In remote collaborative live performances or recordings, musicians in the far end are usually represented by loudspeakers in the near-end space. There

exist inherent acoustic paths from loudspeakers to microphones in the acoustic space. If these paths are not efficiently eliminated, feedback can occur and ruin the sound experience. This is a canonical problem in teleconferencing and is extensively studied for speech signals. Compared to speech, musical instruments exhibit very different characteristics. Accordingly, echo handling for music signal contents demand different efforts than speech and human voice.

1.2.4 Post Processing

Another challenge arises at the stage of auditorily documenting the collaborative work. A straightforward way is to make a live recording of the whole concert or studio session using local microphones. In this way, part of the source signals appear to be low-quality network-transmitted audio played through loudspeakers. A more desirable way is to obtain each sound source through their respective local recordings. In such a collaborative project, to obtain relatively noise-free recording from each location, sounds generated from other locations need to be removed from the mixtures. After that, clean recordings in each location can be sent to one central studio to produce a high-quality mixdown of the whole project.

1.3 Prototype System Setup

From a signal processing perspective, this dissertation searches for possible solutions for remote collaborative recording sessions. An exemplary setup of a

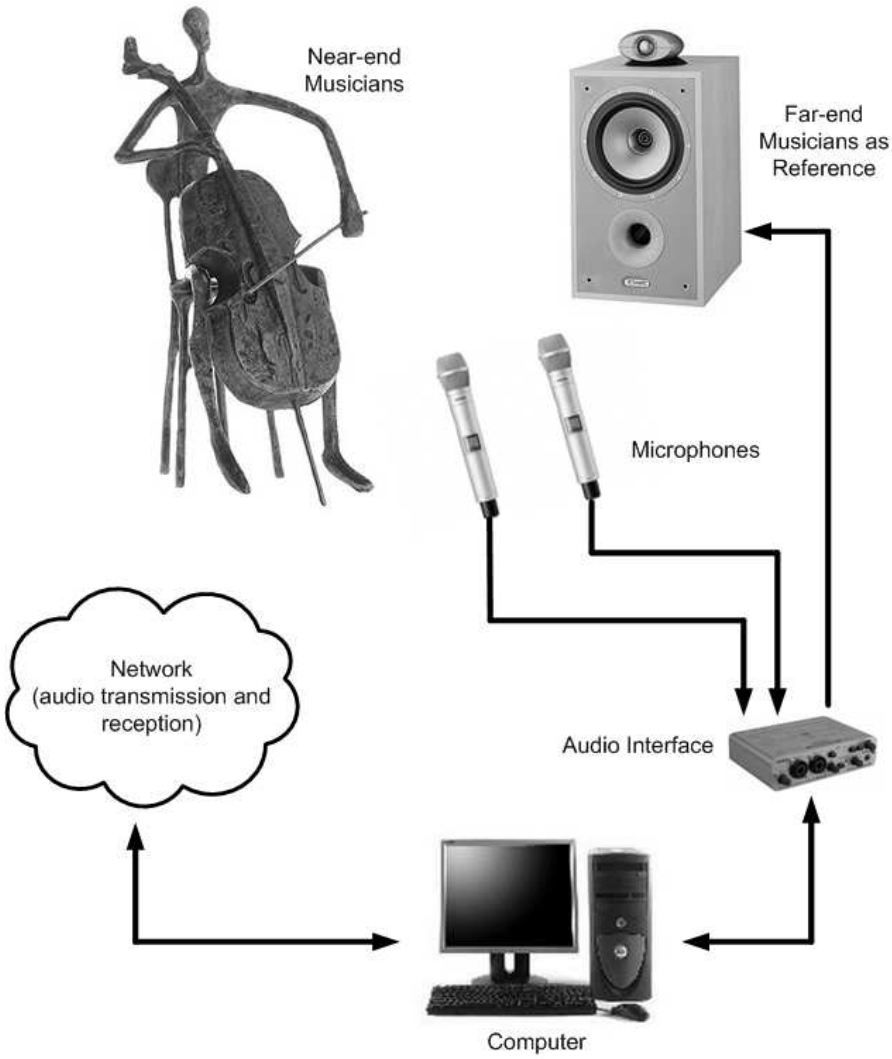


Figure 1.1: A realistic drawing of the prototype system setup.

prototype system is shown in Figure 1.1 with realistic drawings. The system includes a stationary setup of two matched microphones and one loudspeaker. Both microphones perform the task of recording the local musicians. As will be discussed later, one of the microphones functions as the echo cancellation microphone. The loudspeaker lets the local musicians listen to audio from other musicians in remote locations as a reference. Analog and digital audio data from microphones and to the loudspeaker are handled by an audio interface connected with a computer. The computer transmits and receives audio data over certain network channels to and from the far end. Major signal processing tasks are also performed inside this computer, including some post processing after the recording session.

Since the main function of the system is to do recordings instead of to handle live performances, network robustness becomes a less critical requirement, and certain audio artifacts can be tolerated at the same time. The challenge of audio latency can also be reconciled with dedicated fast optical connections between different recording studios. Given these conditions, this system mainly targets two of the usual challenges faced in a networked collaborative recording project: loudspeaker-microphone feedback control during the recording session, and the elimination of unwanted far-end reference signals in the post processing.

2

Echo Cancellation: the online processing

2.1 Background of Echo Cancellation

2.1.1 Definitions

The aforementioned feedback control problem is essentially a well-known topic in telecommunications - *acoustic echo cancellation*. The sound that travels from loudspeakers and arrives at nearby microphones is regarded as echo - "the repetition of a sound caused by reflection of sound waves", or "the sound produced in this manner" [1]. It is basically delayed copies of the original acoustic signal. When the delay is less than 20 ms, the combined effect is usually perceived as comb filter coloration of the original sound. If the delay is greater than 50 ms, discrete echoes can be perceived [53], which may make the musicians very uncomfortable and will greatly interfere with a telephone conversation or teleconference.

[1] First entries for "echo" in http://www.webster.com

There are mainly two types of echoes in communication systems. One is *electric echo*, which is also called hybrid echo or line echo. It exists in public-switched telephone networks (PSTN), mobile and IP phone systems. Due to economic reasons, two-wire systems are normally used to perform full-duplex functions that actually require the performance of a four-wire system. The leaky two-wire / four-wire PSTN conversion points are places where reflections (echoes) are created. Another kind of echo, our main interest here, is *acoustic echo*. At one end of a communication system, if a microphone "hears" the sound production device near it, a copy of the sound from the other end is going to be amplified by the sound production device, picked by the microphone, and transmitted back. In wireless networks, speech processing will introduce delays ranging from 80 ms to 100 ms, thus the total 160 ms to 200 ms end-to-end delay definitely will result in unacceptable echoes. Typical situations for acoustic echoes include teleconferencing systems and the hands-free mode of mobile handsets (such as receiving phone calls with car speakers instead of a Bluetooth headset). Even worse, in a teleconference, if neither end has good echo cancellers, the echo is going to travel back and forth along the transmission channel, continuously make copies of itself and result in infinite feedback. Such feedback can exist in the same form in a remote recording session, making collaboration impossible for musicians in separate locations.

2.1.2 A Brief History of Echo Cancellers

The topic of echo control is addressed almost at the first appearance of telephone systems. In the late 1950s, the first echo-suppression device is essentially voice-activated switches that cut off echoes by not transmitting them. This usually results in choppy first syllables, and the communication is not really full-duplex. In the 1960s, AT&T Bell Labs and COMSAT TeleSystems worked on this and COMSAT implemented their designs across satellite communications networks. In the late 1970s, COMSAT sold their first analog echo cancellers which were mainly digital devices with analog interfaces to the network. In the early 1980s, as signal processing shifted into the digital domain, various new echo cancellers appeared and outperformed the suppression-based techniques significantly. In the 1990s, wireless telecommunications industries thrived and continued to demand good echo cancellers for their digital network infrastructures such as TDMA, CDMA, and GSM. Recently, the growing bandwidth of internet makes teleconferencing much more convenient, and a large-scale teleconference often involves multichannel sound production and microphone arrays. The increased complexity in such systems makes echo cancellation more and more challenging.

2.1.3 A Typical System Model with Solution

Despite the variety of algorithms and implementations to perform echo cancellation [31, 62], mainstream methods are mostly trying to model the problem

with a system shown in Figure 2.1(a) and solve it with a system shown in Figure 2.1(b).

Figure 2.1 illustrates a system that models, for example, a one-speaker-one-microphone teleconference setup. The local end is defined as the *near end*, and the other end across the transmission channel (IP, CDMA, TDMA, GSM, etc.) is defined as the *far end*. For simplicity, system details are only depicted for the near end. In the figure, this near-end system receives a signal $x(n)$ and transmits a signal $y(n)$. $x(n)$ represents the far-end speech signal after a local amplification gain. It is the only signal that an ideal system receives. $s(n)$ represents local speech signal $s'(n)$ after being picked up by the microphone. It is the only signal that an ideal system transmits. In reality, $x(n)$ travels from the loudspeaker, through the conference room, and gets picked up by the same microphone. If the changes $x(n)$ undergoes along this path is modeled with a transfer function of impulse response $h(n)$, what gets really transmitted to the far end is now $x(n) * h(n) + s(n)$ [2], and the echo $x(n) * h(n)$ usually creates undesirable results as described before.

Given this model, a straightforward way to approximate the ideal system is to identify the impulse response $h(n)$, regenerate the echo term $x(n) * h(n)$ by filtering, remove it from the sound mixture picked up by the near-end microphone, and restore the "clean" speech $s(n)$. (Figure 2.1(b))

To obtain $h(n)$, adaptive filters are usually used to perform such a system identification task. If the near-end speech $s'(n)$ is absent, Figure 2.1(b) can be

[2] $*$ denotes convolution

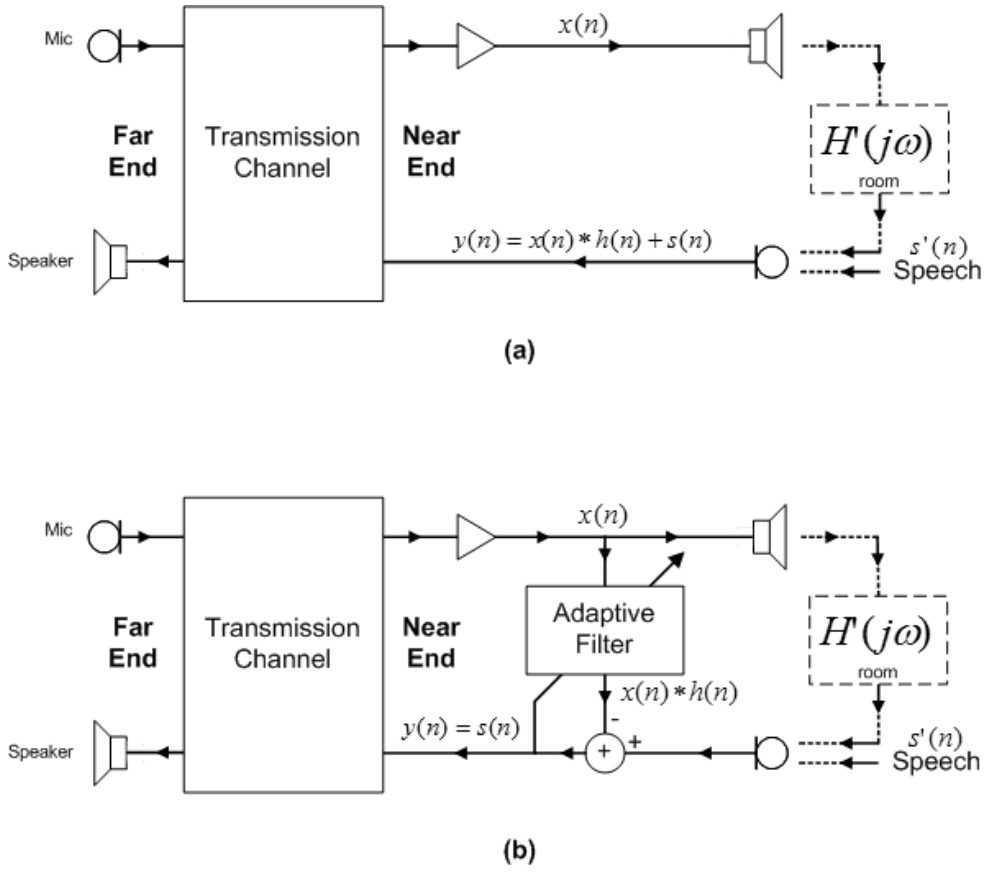


Figure 2.1: System model (a) and typical adaptive filter approach (b) to echo cancellation.

simplified to Figure 2.2: one adaptive filter is put parallel to an unknown system, and both of them receive input signal $x(n)$. The unknown system produces a desired signal $d(n)$ and the difference of adaptive filter output and $d(n)$ is defined as the error signal $e(n)$. Taking $e(n)$ as another input, the adaptive filter adjusts itself over time to minimize the error $e(n)$, thus behaving more and more like the unknown system.

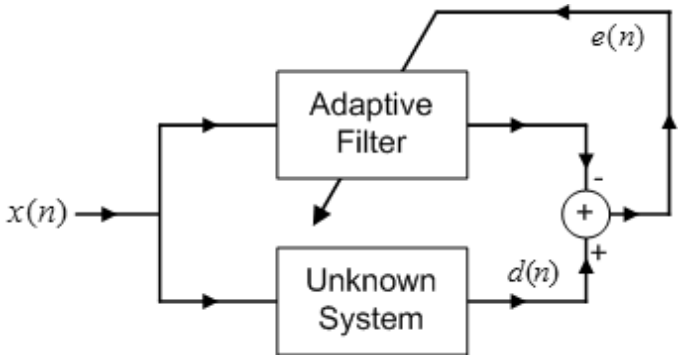


Figure 2.2: System identification with adaptive filter.

In this way, when the far-end person is talking and the near-end speaker is silent, the adaptive filter in Figure 2.1(b) will make efforts to match the unknown impulse response $h(n)$. When the near-end person is talking, the adaptation stops. The filter output should always be subtracted from the transmission path, and the better the filter matches $h(n)$, the less echo will be transmitted. If the adaptive filter is an FIR filter of proper order, two typical implementation methods exist: least mean square (LMS) and recursive least square (RLS).

Least Mean Squares (LMS)

If the adaptive filter is a length-L FIR filter, let

$$\mathbf{h}(n) = [h_0(n), h_1(n), \ldots, h_{L-1}(n)]^T \tag{2.1}$$

be its impulse response and

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T \tag{2.2}$$

be the input signal vector. The error can be calculated as

$$e(n) = d(n) - \mathbf{h}^H(n)\mathbf{x}(n) \tag{2.3}$$

and $\mathbf{h}(n)$ will be updated for every sample as

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu e^*(n)\mathbf{x}(n), \tag{2.4}$$

where $\mu > 0$ is the update *stepsize* and $e^*(n)\mathbf{x}(n)$ is $e(n)$'s direction of change toward the global minimum. The choice of stepsize μ becomes crucial: a small μ creates stable but slowly converging filters; a large μ produces fast converging filters, but might result in unstable oscillating coefficients. LMS will be elaborated in later sections.

Recursive Least Squares (RLS)

Unlike LMS, RLS uses all past samples (with proper weights) instead of just the current tap-input samples to estimate the error. The update doesn't use the gradient descent method in LMS, but directly searches for the minimum of

the cost function by setting its gradient to zero. Skipping details, the resulting equation requires the following recursive update equation to be solved

$$\begin{aligned} \mathbf{P}(n) &= \rho^{-1}\mathbf{P}(n-1) + \rho^{-1}\mathbf{k}(n)\mathbf{x}(n) \\ \text{with} \quad \mathbf{k}(n) &= \frac{\rho^{-1}\mathbf{P}(n-1)\mathbf{x}(n)}{1+\rho^{-1}\mathbf{x}^H(n)\mathbf{P}(n-1)\mathbf{x}(n)} \end{aligned} \tag{2.5}$$

where $0 < \rho < 1$ is the decay factor that weights previous samples. Finally the update equation for the adaptive filter is

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mathbf{k}(n)(d^*(n) - \mathbf{x}^H(n)\mathbf{h}(n-1)). \tag{2.6}$$

This algorithm shows faster convergence than the LMS, while it is computationally more costly.

2.1.4 Problems of Existing Techniques

The approach described in 2.1.3 is canonical and widely used. There are many improvements over the years in different aspects including using multiple sensors and working with subbands [37, 18, 20, 19, 16, 17]. Compared with the research efforts in theory, most practical results are still not as satisfying as expected. Some constraints are described as follows.

Algorithm Constraints

Taking LMS and RLS as time domain adaptive filter examples, there are always inherent trade-offs. Fast convergence, stability, and computational efficiency cannot be achieved at the same time. As an example, reverberant rooms

usually need very long filters to match their room impulse responses. This will immediately incur very high computational cost. Also, it is very hard to make a long filter converge quickly yet stably. New algorithms have been proposed and tested. Jorge Agüero and others [1] experimented with wavelet packet filter banks and low order transversal filters which offer a better trade-off between performance and computational cost. Athanasios and others [43] compared FIR and IIR filters with the same number of free parameters, and did not observe any significant gain from the use of IIR models. They concluded that both polynomial and rational transfer functions are "inadequate" for the application to comparable degrees. Apparently, significant gaps remain between improved algorithms and the reality. One major reason is that room acoustics usually plays an important role in practical situations.

Over-Simplified Room Acoustics

Real-life acoustics are normally hard to model with simple math, and echoes do work with real air vibrations in a room. Naturally, modeling it with a simple impulse response will result in a lot of errors. In room acoustics [41], according to the room dimensions, the transfer function from one point to another (from loudspeaker to microphone, in this case) can be calculated based on room modes. However, this is only true for low frequencies. For frequencies above the *Schroeder frequency* (typically about 200 Hz, depending on volume), room modes become so dense that the room can only be described statistically. As a result,

the energy peaks and valleys are constantly shifting their locations in a room, and the transfer function for high frequencies is highly time-varying. This is why in the echo cancellation community the "20 dB rule" [38] exists which reflects the observation that, typically, one can obtain only about 20 to 30 dB of acoustic echo cancellation in actual physical settings. Cary W. Elko and others [26] also studied the influence of thermal fluctuation on room impulse response and corresponding impact on echo cancellation. Their results show that thermal variations on the order of one tenth of a degree Centigrade can lead to surprisingly large variations in the room impulse response. This is measured and visualized in Figure 2.3. The misalignment error measure J (dB in the figure) is defined as

$$J = \frac{[\mathbf{h} - \hat{\mathbf{h}}]^T[\mathbf{h} - \hat{\mathbf{h}}]}{\mathbf{h}^T\mathbf{h}}, \tag{2.7}$$

where $\mathbf{h} = [h_0, h_1, \cdots, h_{L-1}]^T$ is the impulse response of length-L filter at some reference temperature, and $\hat{\mathbf{h}}$ is the computed impulse response at some other temperature. The effect of temperature change on the impulse response is mainly a result of sound speed change, thus in our research, we assume that an inaccurate phase response of the transfer function suffers more from this and contributes more in the insufficient performance of an echo canceller. Refer to Figure 2.1(b), the estimated echo term is directly subtracted from the sound mixture in the time domain. In this way, any incorrect phase response will cause the canceller to completely fail for high frequency contents.

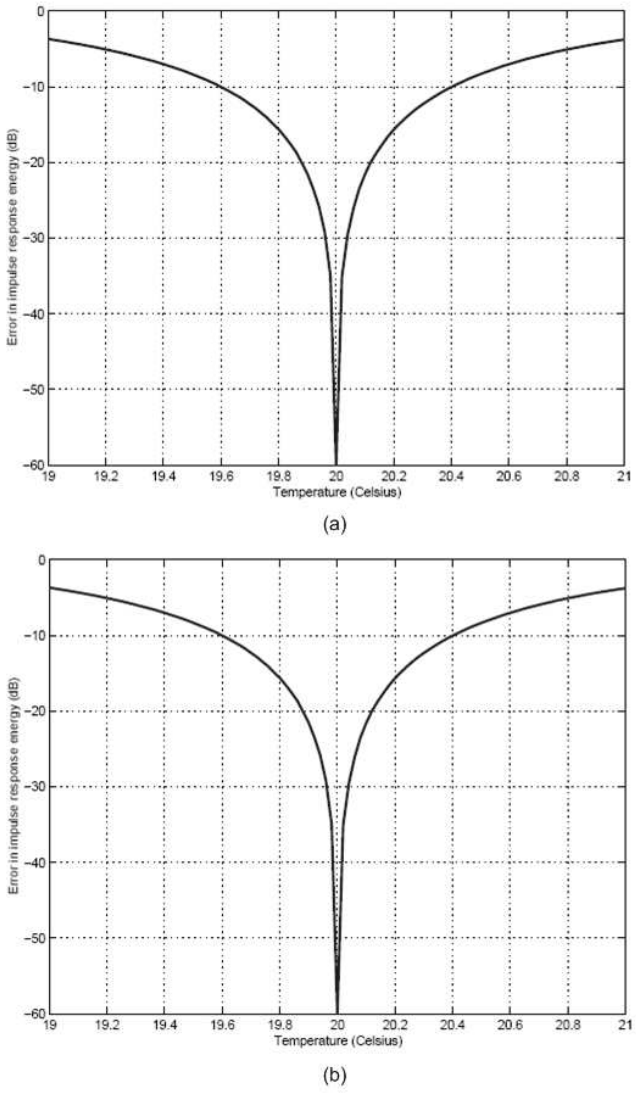


Figure 2.3: Predicted misalignment error as a function of temperature for 10 cm (a) and 2 m (b) microphone-to-loudspeaker spacing [26].

Full-duplex and Double-talk

Full-duplex communication allows users at the near end and far end to talk at the same time. It is called a *double talk* when this happens. The only time for an adaptive filter to correctly update its coefficients is when the far-end user is talking but the near-end is quiet. During double-talk time periods, the adaptive filter should be frozen. Otherwise, the signal from the near-end talker will alter the filter undesirably. To identify time periods of this nature and to help echo cancellation, *double-talk detectors* are built to decide the moments to start and stop adaptive processes. Distinguishing echo path changes is often challenging, and conventional methods [9, 66] and recent improvements [35] usually make the double-talk judgement based on some tunable thresholds. When a computed detection statistic is less than (or greater than, depending on specific algorithms) the threshold, double talk is declared. The threshold value is crucial. If it is too sensitive, the filter will constantly be frozen and not enough adaptations are performed; if it is too insensitive, false filter updates will be performed frequently, causing the filter to diverge. So, the proper value of the double-talk threshold is another trade-off that is hard to find in conventional approaches.

2.2 A Novel System Design Approach

In the prototype system proposed in this dissertation, a new approach for echo cancellation is presented. It is fundamentally different from modeling

the whole system with a linear transfer function. This approach, which actually originates from a noise reduction method, works by keeping the phase of the sound mixture picked up by the near-end microphone, and only attenuating energy in power spectrum that belongs to the echo.

2.2.1 Spectral Subtraction

In the topic of background noise reduction, Boll [12, 13] originally proposed the method of spectral subtraction. The method is based on an additive noise model and targets relatively stationary noise floors. Assume the signal $x(n)$ is a sum of speech signal $s(n)$ and noise signal $v(n)$, in the time domain,

$$x(n) = s(n) + v(n), \tag{2.8}$$

and their Fourier domain relationship is

$$X(e^{j\omega}) = S(e^{j\omega}) + V(e^{j\omega}). \tag{2.9}$$

The spectral subtraction output signal is calculated with its magnitude equal to the magnitude difference of summed signal and noise, and its phase equal to the summed signal phase. Suppose the phase of summed signal is Φ_X, the resulting signal after subtraction filter will be

$$\begin{aligned} \hat{S}(e^{j\omega}) &= H(e^{j\omega})X(e^{j\omega}) \\ &= [|X(e^{j\omega})| - |V(e^{j\omega})|]\Phi_X. \end{aligned} \tag{2.10}$$

Suppose the average magnitude of noise can be measured during moments without speech activities, the measured (average) value $\mu(e^{j\omega}) = E\{|V(e^{j\omega})|\}$ can be substituted into Eq.(2.10), so that

$$\begin{aligned}\hat{S}(e^{j\omega}) &= H(e^{j\omega})X(e^{j\omega}) \\ &= [|X(e^{j\omega})| - \mu(e^{j\omega})]\Phi_X,\end{aligned} \tag{2.11}$$

and

$$H(e^{j\omega}) = 1 - \frac{\mu(e^{j\omega})}{|X(e^{j\omega})|} \tag{2.12}$$

becomes the definition of the spectral subtraction filter. The discrepancy between resulting speech signal and the reality is

$$\epsilon(e^{j\omega}) = \hat{S}(e^{j\omega}) - S(e^{j\omega}) = V(e^{j\omega}) - \mu(e^{j\omega})\Phi_X. \tag{2.13}$$

To minimize the error, [12] also proposed three modifications to enhance the spectral subtraction filter:

Magnitude Averaging

If average $|X(e^{j\omega})|$ over M time windows and substitute it with the averaged version $\overline{|X(e^{j\omega})|}$, the filtered speech will be

$$S_{\text{average}}(e^{j\omega}) = [\overline{|X(e^{j\omega})|} - \mu(e^{j\omega})]\Phi_X, \tag{2.14}$$

where

$$\overline{|X(e^{j\omega})|} = \frac{1}{M}\sum_{i=0}^{M-1}|X_i(e^{j\omega})|. \tag{2.15}$$

In this way, the error will become

$$\epsilon(e^{j\omega}) = S_{\text{average}}(e^{j\omega}) - S(e^{j\omega}) \tag{2.16}$$

$$\simeq \overline{|V|} - \mu = \frac{1}{M}\sum_{i=0}^{M-1} |V_i(e^{j\omega})| - \mu, \tag{2.17}$$

so it converges to zero as the average time increases. On the other hand, the averaging cannot be too long, otherwise smeared transient sounds will be heard.

Half-wave Rectification

By doing half-wave rectification, the spectral subtraction filter becomes

$$H_{\text{rectified}}(e^{j\omega}) = \frac{H(e^{j\omega}) + |H(e^{j\omega})|}{2}. \tag{2.18}$$

In this way, when $|X(e^{j\omega})|$ is less than the noise floor, the output is set to zero. As a result, the noise floor is always reduced by $\mu(e^{j\omega})$, and possible low variance coherent noise tones are eliminated.

Residual Noise Reduction

In absence of speech activities, or during speech activities but for frequency bins without speech content, the actual noise energy could still be different from what is subtracted by the filter. the remaining part of noise residual may sound as random tone generators. To reduce these residual noises, half-wave rec-

tified signal amplitude $|\hat{S}_i(e^{j\omega})|$ is modified, so that

$$|\hat{S}_i(e^{j\omega})| = |\hat{S}_i(e^{j\omega})|, \text{ if } |\hat{S}_i(e^{j\omega}) \geq \max|V_R(e^{j\omega})|$$

$$|\hat{S}_i(e^{j\omega})| = \min\{|\hat{S}_{i-1}(e^{j\omega})|, |\hat{S}_i(e^{j\omega})|, |\hat{S}_{i+1}(e^{j\omega})|\},$$

$$\text{if } |\hat{S}_i(e^{j\omega})| < \max|V_R(e^{j\omega})| \tag{2.19}$$

where $|V_R(e^{j\omega})|$ is the maximum value of noise residual measured in absence of speech activity. In this way, when producing the filtered signal, minimum energy is chosen for neighboring time windows when speech activity is lower than noise floor, and the randomly distributed residual noise is likely to be reduced.

The method of spectral subtraction is also mentioned in [57] and [27], which are related to echo suppression. These methods are still based on the model similar to the one described in 2.1.3. They only use noise suppression to remove background noise and echo residues after the adaptive filter process.

2.2.2 The New System Structure

The spectral subtraction method in noise reduction can be advantageously converted to treat echo cancellation (feedback control) problem in remote collaborative recordings. If echo is treated as the "noise floor" in noise reduction, the new system can be constructed so that the output signal keeps phase of the sound mixture and removes the energy portion that belongs to the echo. In theory, this system has the following merits:

- *better model matching*: the adaptive filter based approach tries to match

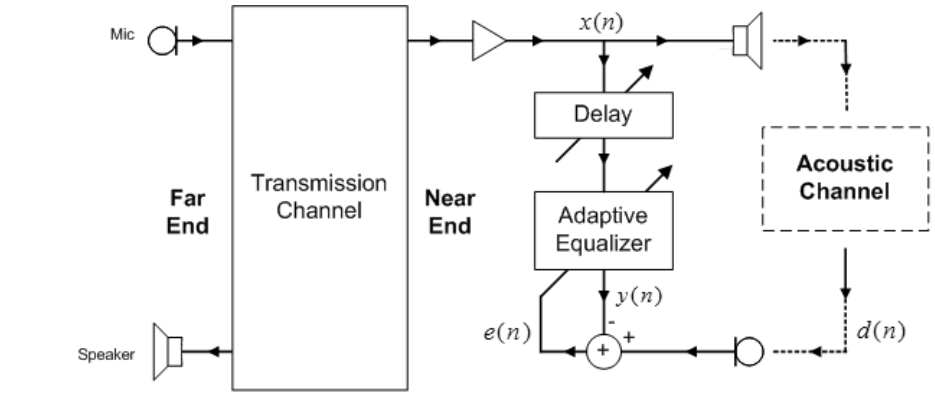
the echo path with a linear filter (or the alike). In reality, influence of the sound production device, sound pickup device and the room could be a fairly complex system that involves non-linear behaviors. The existence of model mismatch may fail the adaptive filter. On the other hand, what is crucial to the proposed new approach is not the real echo path transfer function, but the energy behavior at the end of the path. To match the energy spectrum is more practical than to match the whole system, thus this system exhibits a less aggressive model match, and is likely to be more close to reality.

- *stability*: since this method is totally energy-based, phase information of signals are not as important as in the adaptive filter approach. Hence it can tolerate much more fluctuations in room acoustics and phase errors from measurement or computation.

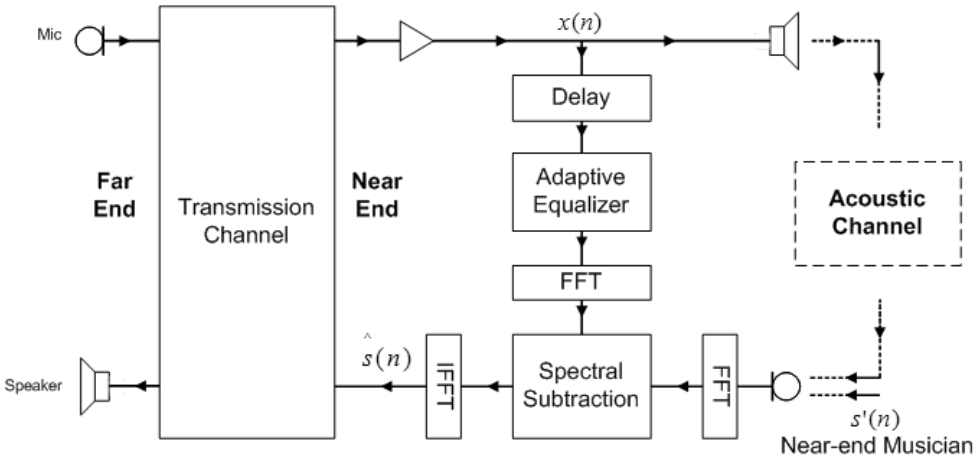
- *readily available "noise" information*: In the spectral subtraction method, the system relies on non-speech activity time periods to estimate average noise energy. If it is applied to echo cancellation, the unwanted item now is the far-end signal's delayed and colored copy. For a communication system, the far-end signal is consistently available, so this "noise reference" is almost ready at any time. Although adjustments to its delay and equalization are still needed to provide its correct power spectrums in the microphone sound mixture, it is a relative more relaxed requirement compared to waiting for time periods without near-end signal to measure new noise floors.

The power spectrum of far-end signal cannot be used directly as the echo's power spectrum. The echo path from before local amplification to after microphone pick up still needs to be modeled. Since we are only interested in the energy of echo signal, the path can be simply modeled with a time delay and an equalizer, which fine-tunes the power spectrum of the echo arriving at a microphone. The equalizer can be adaptively obtained with an FIR adaptive filter short in length. The whole system diagram for the new echo cancellation design is shown in Figure 2.4. The diagram shows one microphone and one loudspeaker in the near end. The loudspeaker corresponds to the better-depicted loudspeaker in Figure 1.1, and the microphone represents one of the two microphones shown in Figure 1.1.

Figure 2.4(a) shows the system structure during equalizer identification mode. The system is switched to this mode either during a tuning period before the recording session, or during a short moment in the recording session when the near-end musician is not playing. Delay can be calibrated with test pulses; if the signal picked up by microphone is used as the desired signal $d(n)$, and the difference between $d(n)$ and equalizer output $y(n)$ is used as error signal $e(n)$, the equalizer can be adaptively identified. Since the equalizer filter is usually short-length FIR, error signal may not be reduced to the same small value as a long filter would do. But if only consider the error in magnitude, the requirement on filter length is relaxed. On the other hand, the echo path in these kind of situations are relatively high-latency routes. Typical PC-based realtime audio systems have



(a) Equalizer identification (training) mode



(b) Spectral subtraction (functioning) mode

Figure 2.4: System diagram of the new approach: (a) training mode to tune delay and adaptive equalizer; (b) functioning mode powered by spectral subtraction.

a system latency of 10 to 100 ms. Even if the near-end microphone is placed very close to the speaker to minimize signal delay due to air transmission, we are still facing a whole system latency of tens of milliseconds. This translates to at least thousands of filter taps if the whole system is modeled by one adaptive filter. This computational cost is significant to even PC-based systems, not to mention embedded system where MIPS and memories are sparse resources. For speech, long filter requirements might be alleviated with lower sampling rate (8 kHz, 16 kHz) due to the band-limited nature of human voice. In music recording, 44.1 kHz, 48 kHz, and up to 192 kHz sampling rates are frequently used. This makes the usage of long filters even more impractical. Consequently, short filters with hundreds of taps are used in my simulations.

Figure 2.4(b) shows the system structure during spectral subtraction mode. The system is switched to this mode when both far-end and near-end musicians are playing in a recording session. The far-end signal is first delayed by the delay unit, filtered by the already updated equalizer, and then transformed into frequency domain. Its power spectrum (magnitude, or similar features) is properly subtracted from that of the sound mixture picked up by the near-end microphone. After this, the resulting frequency domain signal is transformed back to generate signal $\hat{s}(n)$ which is supposed to resemble the near-end signal $s'(n)$.

In noise reduction, stationary noise usually has random phases, so keeping the phase of sound mixture and discarding noise phase is a reasonable choice. It is

a different case for the current system regarding this aspect. The phase of the echo is forcefully eliminated and replaced by that of the sound mixture, which could create colorations to the transmitted sound. Some preliminary experiments and simulations are conducted, which shows that the colorations are acceptable to the ear in general.

2.3 Details of the Integrated System

Without getting into implementation details, some theories regarding important components of the echo cancellation portion of this prototype system are discussed.

2.3.1 NLMS Theory

The training process of equalizer in the new system (see Figure 2.4(a)) is based on normalized least mean square (NLMS) algorithm, due to its simplicity in implementation. Related theories from [31], are briefly described here in steps.

Wiener Filter

Suppose a length-L FIR filter has a transversal structure shown in Figure 2.5, its filter weights (impulse response, complex values) can be represented with vector

$$\mathbf{h}(n) = [h_0(n), h_1(n), \ldots, h_{L-1}(n)]^T. \tag{2.20}$$

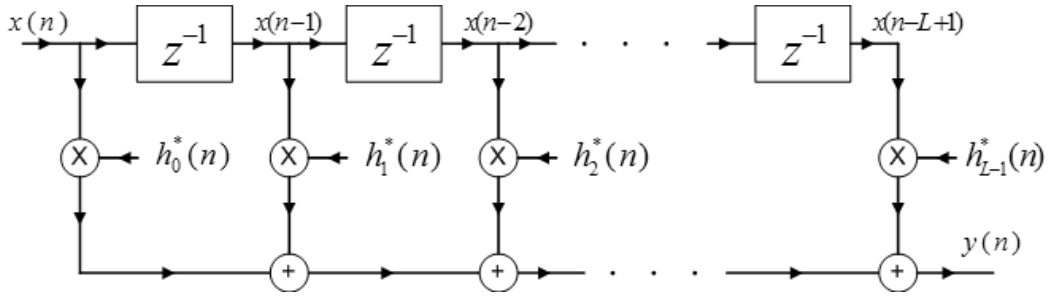


Figure 2.5: Transversal adaptive FIR filter. "*" denotes complex conjugate.

Define another vector $\mathbf{x}(n)$ for the curent and past $L-1$ input samples where

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T, \tag{2.21}$$

the filtered output at time n should be a result of convolution

$$y(n) = \sum_{k=0}^{L-1} h_k^*(n)x(n-k) = \mathbf{h}^H(n)\mathbf{x}(n). \tag{2.22}$$

If a reference signal (desired signal) is given and its value at time n is $d(n)$, then the error signal is

$$e(n) = d(n) - y(n) = d(n) - \mathbf{h}^H\mathbf{x}(n). \tag{2.23}$$

If the input vector $\mathbf{x}(n)$ and the desired signal $d(n)$ are jointly stationary, then the *mean-square error* or *cost function* $J(\mathbf{h}(n))$, or simply $J(n)$, is a quadratic function of the weight vector:

$$\begin{aligned} J(n) &= E[e(n)e^*(n)] \\ &= \sigma_d^2 - \mathbf{h}^H(n)\mathbf{p} - \mathbf{p}^H\mathbf{h}(n) + \mathbf{h}^H(n)\mathbf{R}\mathbf{h}(n) \end{aligned} \tag{2.24}$$

where

σ_d^2 = variance of the desired signal $d(n)$,

$\mathbf{p}$ = cross-correlation vector between input vector $\mathbf{x}(n)$ and desired signal $d(n)$, and $\mathbf{R}$ = correlation matrix of the input vector $\mathbf{x}(n)$.

Given this cost function, an optimum filter $\mathbf{h}(n) = \mathbf{R}^{-1}\mathbf{p}$, the *Wiener filter*, can set it to its minimum

$$\min J(n) = \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p}. \tag{2.25}$$

To find the Wiener solution, matrix inversion is required, which is hard to achieve and sometimes not possible.

Method of Steepest Descent

To avoid matrix inversion, the gradient of the cost function can be obtained by taking its partial derivative with regard of the filter weight vector, resulting in

$$\nabla J(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{h}(n). \tag{2.26}$$

If the filter taps are updated by descending along the gradient of the cost function at each time sample, then with a proper stepsize, the cost function will eventually reach its global minimum. This method is named *method of steepest descent*, and the filter update equation at time n is

$$\begin{aligned} \mathbf{h}(n+1) &= \mathbf{h}(n) - \frac{1}{2}\mu \nabla J(n) \\ &= \mathbf{h}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{h}(n)] \end{aligned} \tag{2.27}$$

where μ is a tunable stepsize parameter.

Least Mean Square (LMS)

The method of steepest descent still requires calculation of correlation matrix $\mathbf{R}$ and cross-correlation vector $\mathbf{p}$. To further simplify this, instantaneous estimates for $\mathbf{R}$ and $\mathbf{p}$ are used, which are defined as

$$\hat{\mathbf{R}} = \mathbf{x}(n)\mathbf{x}^H(n) \tag{2.28}$$

and

$$\hat{\mathbf{p}} = \mathbf{x}(n)d^*(n) \tag{2.29}$$

respectively. Thus, Eq.(2.26), the gradient of mean square error becomes

$$\hat{\nabla} J(n) = -2\mathbf{x}(n)d^*(n) + 2\mathbf{x}(n)\mathbf{x}(n)^H\mathbf{h}(n), \tag{2.30}$$

and the filter update Eq.(2.27) becomes

$$\begin{aligned}\mathbf{h}(n+1) &= \mathbf{h}(n) + \mu\mathbf{x}(n)[d^*(n) - \mathbf{x}^H(n)\mathbf{h}(n)] \\ &= \mathbf{h}(n) + \mu\mathbf{x}(n)e^*(n).\end{aligned} \tag{2.31}$$

This is the *least mean square* (LMS) method. A major merit of this method is its simplicity of implementation.

Normalized Least Mean Square (NLMS)

In LMS, as can be seen from Eq.(2.31), the filter update amount is directly proportional to the input vector $\mathbf{x}(n)$. With a constant update stepsize μ, the actual updated amount varies a lot with input signal, and this will create a *gradient*

noise amplification problem. To overcome this, the second summation term on the right side of Eq.(2.31) can be normalized with the squared Euclidean norm of the input vector:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\tilde{\mu}}{\|\mathbf{x}(n)\|^2}\mathbf{x}(n)e^*(n) \tag{2.32}$$

where

$$\|\mathbf{x}(n)\| = \mathbf{x}^H(n)\mathbf{x}(n), \tag{2.33}$$

and $\tilde{\mu}$ is a tunable update stepsize. When input energy is very small, numerical difficulties may occurs due to division by a very small number. So, adjustments are made to add a small constant $\delta > 0$ to the denominator. The resulting equation

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\tilde{\mu}}{\delta + \|\mathbf{x}(n)\|^2}\mathbf{x}(n)e^*(n) \tag{2.34}$$

is the filter update equation with *normalized least mean square* (NLMS).

Block NLMS

Instead of updating the filter taps sample-by-sample, it could be done in a block-by-block manner instead. The input signal can be sectioned into blocks of samples. For every sample in a block, an error sample is obtained, but there is only one update of the filter taps based on a summation of update vectors. Suppose B is the sample block size, i is the within-block sample index, and let k be the block

index, the relationship of them with original sample time n is

$$n = kB + i, \quad i = 0, 1, \ldots, B-1 \tag{2.35}$$
$$k = 1, 2, \ldots.$$

With the same notation for input vector $\mathbf{x}(n)$ in Eq.(2.21), input data can be represented in a matrix form

$$\mathbf{A}(k) = [\mathbf{x}(kB), \mathbf{x}(kB+1), \ldots, \mathbf{x}(kB+B-1)]^T. \tag{2.36}$$

The output of this block is accordingly

$$\begin{aligned} y(kB+i) &= \mathbf{h}^H(k)\mathbf{x}(kB+i) \\ &= \sum_{j=0}^{L-1} h_j^*(k)x(kB+i-j), \quad i = 0, 1, \ldots, B-1 \end{aligned} \tag{2.37}$$

and there are B error samples in this block:

$$e(kB+i) = d(kB+i) - y(kB+i), \quad i = 0, 1, \ldots, B-1. \tag{2.38}$$

To update the filter taps once per block in an LMS manner, the accumulated cross-correlation between input signal vector and error signal is used. Let $\Phi(k)$ be the cross-correlation that

$$\begin{aligned} \Phi(k) &= \mathbf{A}^T(k)\mathbf{e}^*(k) \\ &= \sum_{i=0}^{B-1} \mathbf{x}(kB+i)e^*(kB+i) \end{aligned} \tag{2.39}$$

where $\mathbf{e}(k)$ is the B-by-1 vector

$$\mathbf{e}(k) = [e(kB), e(kB+1), \ldots, e(kB+B-1)]^T, \tag{2.40}$$

the update equation similar to Eq.(2.31) for LMS then becomes a *block LMS* update equation

$$\begin{aligned}\mathbf{h}(k+1) &= \mathbf{h}(k) + \mu_B \Phi(k) \\ &= \mathbf{h}(k) + \mu_B \sum_{i=0}^{B-1} \mathbf{x}(kB+i) e^*(kB+i). \end{aligned} \tag{2.41}$$

To make an unbiased time average, a factor of $1/B$ should be included in the update increment of Eq.(2.41), so the relationship of block LMS stepsize μ_B and LMS stepsize μ should be

$$\mu_B = B\mu. \tag{2.42}$$

To further transform this block LMS into *block NLMS*, a normalization term should be added to Eq.(2.41), so that not only the cross-correlation vector, but also the input signal energy (squared Euclidean norm) is accumulated. The new update equation for block NLMS should then look like

$$\begin{aligned}\mathbf{h}(k+1) &= \mathbf{h}(k) + \frac{\tilde{\mu}}{\delta + \sum_{i=0}^{B-1} \|\mathbf{x}(kB+i)\|^2} \mathbf{\Phi}(k) \\ &= \mathbf{h}(k) + \frac{\tilde{\mu}}{\delta + \sum_{i=0}^{B-1} \|\mathbf{x}(kB+i)\|^2} \sum_{i=0}^{B-1} \mathbf{x}(kB+i) e^*(kB+i). \end{aligned} \tag{2.43}$$

Since the normalization term is accumulated the same way as the update vector, stepsize $\tilde{\mu}$ is the same as in Eq.(2.34).

Block NLMS is between the method of NLMS and steepest descent. As the block size increases, the averaging inside block NLMS makes the estimation of gradient more and more accurate. At the same time, the adaptation speed

becomes slower and slower. Hence, it is reasonable to have a larger stepsize for block NLMS when compared to LMS.

2.3.2 Stepsize Control

The value of tunable normalized stepsize constant $\tilde{\mu}$ (in Eq.(2.34) and Eq.(2.41)) is very important to the performance of an NLMS adaptive filter. Compared to LMS, NLMS is already performing stepsize control that automatically normalizes μ in LMS (Eq.(2.31)) with input signal energy, leaving another tunable parameter $\tilde{\mu}$. Despite this normalization, too large a fixed value of $\tilde{\mu}$ may still result in a fluctuating adaptation, and too small a fixed value of $\tilde{\mu}$ might result in a very slowly adapting filter. The upper bound and optimum value of $\tilde{\mu}$ are extensively studied in [31].

Suppose the mechanism to generate the desired signal $d(n)$ fits a multiple regression model , so that

$$d(n) = \mathbf{h}_0^H \mathbf{x}(n) + v(n) \tag{2.44}$$

where $\mathbf{h}_0$ is the model's unknown parameter, i.e. the "truth" of filter coefficients, and $v(n)$ is the additive disturbance. The mismatch between estimated $\mathbf{h}(n)$ and the true $\mathbf{h}_0$ is measured by *weight-error vector*

$$\varepsilon(n) = \mathbf{h}_0 - \mathbf{h}(n). \tag{2.45}$$

Subtracting Eq.(2.32) from $\mathbf{h}_0$, we get

$$\varepsilon(n+1) = \varepsilon(n) - \frac{\tilde{\mu}}{\|\mathbf{x}(n)\|^2}\mathbf{x}(n)e^*(n). \tag{2.46}$$

The stability analysis of NLMS filter is based on *mean-square deviation*

$$\mathcal{D}(n) = E[\|\varepsilon(n)\|^2]. \tag{2.47}$$

The filter will be stable only when $\mathcal{D}(n)$ is monotonically decreasing. The increment of $\mathcal{D}$ at time n is obtained by taking Euclidean norms of both sides of Eq.(2.46), resulting in

$$\mathcal{D}(n+1) - \mathcal{D}(n) = \tilde{\mu}^2 E\Big[\frac{|e(n)|^2}{\|\mathbf{x}(n)\|^2}\Big] - 2\tilde{\mu}E\Big\{\mathrm{Re}\Big[\frac{\xi_u(n)e^*(n)}{\|\mathbf{x}(n)\|^2}\Big]\Big\} \tag{2.48}$$

where $\xi_u(n)$ is the *undisturbed error signal* defined by

$$\begin{aligned}\xi_u(n) &= (\mathbf{h}_0 - \mathbf{h}(n))^H\mathbf{x}(n) \\ &= \varepsilon^H(n)\mathbf{x}(n).\end{aligned} \tag{2.49}$$

To make the converging process of filter monotonic, the right side of Eq.(2.48) (second order polynomial of $\tilde{\mu}$) has to be negative, demanding the range of $\tilde{\mu}$ to be

$$0 < \tilde{\mu} < 2\frac{\mathrm{Re}\{E[\xi_u(n)e^*(n)/\|\mathbf{x}(n)\|^2]\}}{E[|e(n)|^2/\|\mathbf{x}(n)\|^2]}. \tag{2.50}$$

Further more, the largest decrease of $\mathcal{D}$ is achieved at the midpoint of the range, thus optimum stepsize should take the value

$$\tilde{\mu}_{\mathrm{opt}} = \frac{\mathrm{Re}\{E[\xi_u(n)e^*(n)/\|\mathbf{x}(n)\|^2]\}}{E[|e(n)|^2/\|\mathbf{x}(n)\|^2]}. \tag{2.51}$$

To make the computation of $\tilde{\mu}_{\text{opt}}$ tractable, three assumptions are introduced. In *Assumption 1*, fluctuation of $\|\mathbf{x}(n)\|^2$ in iterations are small enough, so that the following approximations are made:

$$E\Big[\frac{\xi_u(n)e^*(n)}{\|\mathbf{x}(n)\|^2}\Big] \approx \frac{E[\xi_u(n)e^*(n)]}{E[\|\mathbf{x}(n)\|^2]}, \quad E\Big[\frac{|e(n)|^2}{\|\mathbf{x}(n)\|^2}\Big] \approx \frac{E[e^2(n)]}{E[\|\mathbf{x}(n)\|^2]}, \tag{2.52}$$

and Eq.(2.51) reduces to

$$\tilde{\mu}_{\text{opt}} \approx \frac{\text{Re}\{E[\xi_u(n)e^*(n)]\}}{E[e^2(n)]}. \tag{2.53}$$

In *Assumption 2*, the undisturbed error $\xi_u(n)$ is uncorrelated with noise $v(n)$, so

$$\begin{aligned} E[\xi_u(n)e^*(n)] &= E[\xi_u(n)(\xi_u^*(n) + v^*(n))] \\ &= E[\xi_u^2(n)], \end{aligned} \tag{2.54}$$

and Eq.(2.53) becomes

$$\tilde{\mu}_{\text{opt}} \approx \frac{E[\xi_u^2(n)]}{E[e^2(n)]}. \tag{2.55}$$

In *Assumption 3*, spectral content of input signal $x(n)$ is essentially flat over a frequency band larger than that occupied by each element of the weight-error-vector $\varepsilon(n)$, so

$$\begin{aligned} E[\xi_u^2(n)] &= E[|\varepsilon^H(n)\mathbf{x}(n)|^2] \\ &\approx E[\|\varepsilon(n)\|^2]E[x^2(n)] \\ &= \mathcal{D}(n)E[x^2(n)], \end{aligned} \tag{2.56}$$

and Eq.(2.55) is then approximated by

$$\tilde{\mu}_{\text{opt}} \approx \frac{\mathcal{D}(n)E[x^2(n)]}{E[e^2(n)]}. \tag{2.57}$$

Since input $x(n)$ and error $e(n)$ are readily available signals, $E[x^2(n)]$ and $E[e^2(n)]$ in Eq.(2.57) can be estimated directly using IIR smoothing filters.

$$E[x^2(n+1)] = (1-\gamma_x)x^2(n+1) + \gamma_x E[x^2(n)],$$
$$E[e^2(n+1)] = (1-\gamma_e)e^2(n+1) + \gamma_e E[e^2(n)] \tag{2.58}$$

where γ_x and γ_e are smoothing constant that lie inside the interval $[0.9, 0.999]$.

To estimate the mean-square-deviation $\mathcal{D}(n)$, an artificial delay needs to be inserted into the system [65, 46]. The system diagram from Figure 2.4(a) after inserting the artificial delay will become what's shown in Figure 2.6. The system

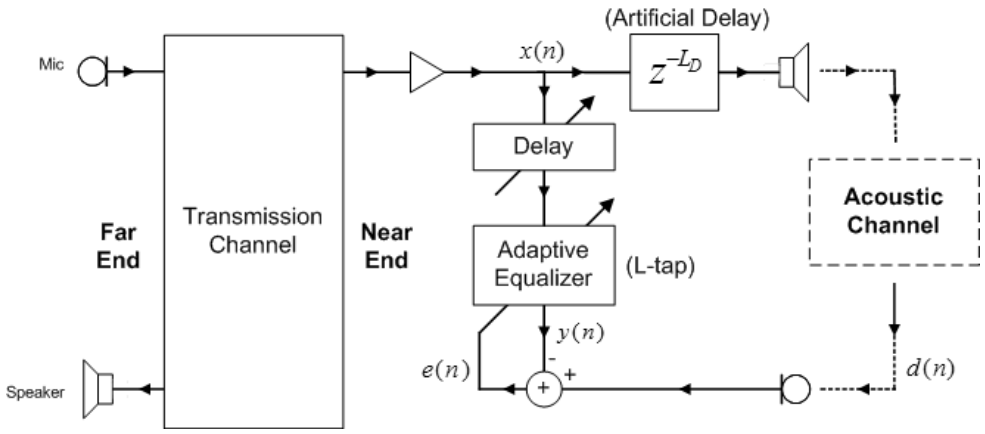


Figure 2.6: Adding an artificial delay to calculate optimum stepsize for NLMS adaptive filters.

latency delay before the equalizer is measured without adding the artificial delay. The equalizer filter will have a total number of L taps, and the first L_D taps model the artificial delay. Since the "truth" is that the first L_D taps in $\mathbf{h}(n)$ are taps with zero weight, the weight-errors will be

$$\varepsilon_k(n) = -h_k(n) \quad \text{for} \quad k = 0, 1, \ldots, L_D - 1. \tag{2.59}$$

In addition, there exists a property that an adaptive filter tends to spread the weight error vector $\varepsilon(n)$ evenly over all the taps [65]. The mean-square deviation can then be approximated with

$$\mathcal{D}(n) \approx \frac{L}{L_D} \sum_{k=0}^{L_D-1} h_k^2(n). \tag{2.60}$$

Bringing Eq.(2.60) and Eq.(2.58) into Eq.(2.57), the optimum stepsize for NLMS filter can then be calculated. However, there is still a possibility that $\tilde{\mu}_{\text{opt}}$ becomes numerically too large. To cope with this, a fixed stepsize $\tilde{\mu}_{\text{fix}}$ is defined which lies in the range $[0, 2]$, and update stepsize $\tilde{\mu}$ is defined as

$$\tilde{\mu} = \begin{cases} \tilde{\mu}_{\text{opt}} & \text{if} \quad \tilde{\mu}_{\text{opt}} < \tilde{\mu}_{\text{fix}}, \\ \tilde{\mu}_{\text{fix}} & \text{otherwise.} \end{cases} \tag{2.61}$$

In later chapters, test results show that this stepsize control produces convincing results. To make adjustments independently for different frequency regions, stepsize control should also be done separately when signal is split into subbands.

2.3.3 Need for Subband Processing

Instead of doing adaptive equalizer identification on the whole signal, a subband approach could be taken so that the input signal and reference signal are broken into different frequency bands, and there is one equalizer for each band. After filtering each band, they are combined again to formulate the output signal. Subband signal processing has taken many different approaches and are applied in different ways to benefit echo cancellation [29, 45, 47, 50, 40, 23, 14, 15]. Most of the efforts are focused on reducing computational complexity.

In the current prototype system, though computational efficiency is still a desirable goal, a more important reason to use the subband adaptive filter is for a better match of reality. The target signal in a digital recording session is usually high-sampling rate and high-resolution data. This scale of precision can host a wide frequency and dynamic range of audio. Thus, single acoustic instruments or vocal sound may appear as sparse signals in the frequency spectrum. During adaptation, it is reasonable to only adapt filters in frequency bands that are "excited" by near-end musician's acoustic signals. For those frequency bands where not much acoustic energy resides, it is better to leave the equalizer as is, rather than keep updating them with some random (noise) input and output. In short, each frequency band should be treated separately and then combined to form the desired output signal. Here, discrete Fourier transform (DFT) analysis and synthesis filter banks are used to achieve subband adaptive filtering.

An interesting analysis is done in [3], where the relationship of Fourier transform (FT) and subband signals can be revealed. For a continuous signal, one popular way to analyze its frequency behavior is the short-time Fourier transform (STFT). STFT translates the signal into a function of frequency f and time t. If the input signal is $x(t)$ and a time window is $w(t)$, the FT of a windowed signal can be expressed as

$$X(f,t) = \int_{-\infty}^{\infty} w(t-\tau)x(\tau)e^{j2\pi f\tau}d\tau. \tag{2.62}$$

This equation can be understood in two ways. In one way, $w(t-\tau)x(\tau)$ is the input

signal scaled by the shifted window function, $e^{j2\pi f\tau}$ is the complex exponential, and $X(f,t)$ is the spectrum (FT) of input signal at time t, with a frequency index f. In the other way, $x(\tau)e^{j2\pi f\tau}$ can be regarded as frequency shifting the frequency band of $x(t)$ centered at f down to zero frequency with the complex exponential $e^{j2\pi ft}$, and the equation is a convolution of impulse response $w(t)$ (usually a low-pass filter) and the frequency-shifted signal. This can be illustrated in Figure 2.7. Thus, STFT when looked along the time axis, represents several parallel complex

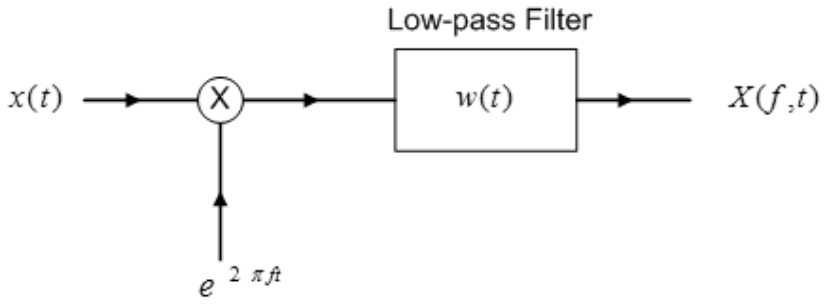


Figure 2.7: Alternative view of STFT.

time signals centered around different frequency bands.

The translation of continuous STFT into DFT sequences is a process of sampling on both time and frequency axis. To make the discrete version preserving enough significant information, certain sampling rates on both axes should be maintained. Two characteristic lengths are defined. The *characteristic time length* is the time period T over which $w(t)$ is significant, and the *characteristic frequency length* is the frequency range F over which $W(f)$, the FT of window $w(t)$, is

significant. For a Hamming window,

$$w(t) = \begin{cases} 0.54 + 0.46cos(2\pi t/T_0), & -T_0/2 \leq t \leq T_0/2 \\ 0, & |t| > T_0/2 \end{cases} \tag{2.63}$$

reasonable T and F are

$$T = T_0 \quad and \quad F = 4/T_0. \tag{2.64}$$

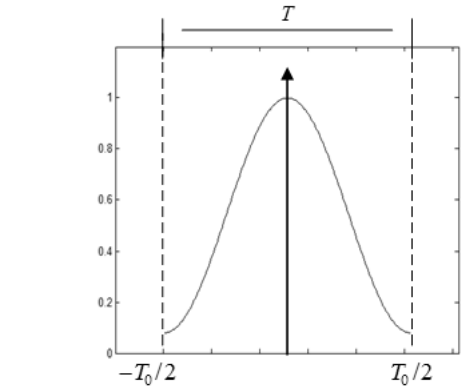
In the time domain (see Figure 2.8(a)), $w(t)$ is non-zero only when t is in the range $[-T_0/2, T_0/2]$, thus the characteristic time length is naturally T_0. In the frequency domain (see Figure 2.8(b)), Hamming window's magnitude is above -42 dB only when the frequency is in the range $[-2/T_0, 2/T_0]$, thus the characteristic frequency length is approximated with $4/T_0$ (counting both positive and negative frequency). According to the Nyquist theorem, the rate of sampling must be greater than or equal to twice the highest frequency. Apply it on the frequency to sample continuous time signals, we get

$$\text{sampling rate on time} \geq 2 \cdot \frac{F}{2} = F, \tag{2.65}$$

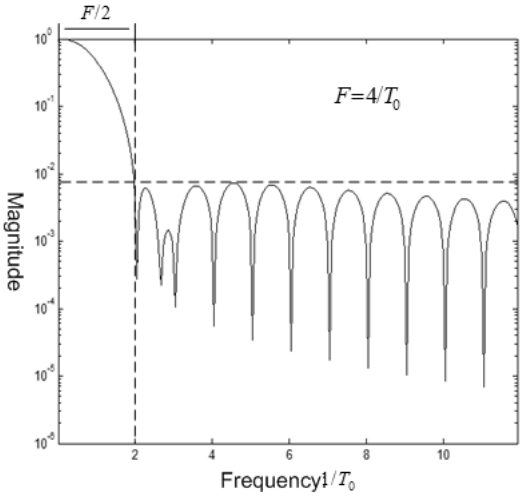
and similar, apply it on time to sample continuous frequency data, we get

$$\text{sampling rate on frequency} \geq 2 \cdot \frac{T}{2} = T. \tag{2.66}$$

Thus, time samples are spaced with $1/F$ intervals and frequency samples are spaced $1/T$ samples. For the case of Hamming window, time spacing of $T_0/4$ indicates that the window hop should be at least one quarter of the window length. In this



(a) Characteristic time length



(b) Characteristic frequency length

Figure 2.8: Characteristic time and frequency length of Hamming window [3].

way, the sampled version of STFT in Eq.(2.62) should be expressed as

$$\begin{aligned} X_{nm} &= X(m/T, n/F) \\ &= \sum_{k=0}^{T-1} w(n/F - k)x(k)e^{j2\pi km/T} \end{aligned} \tag{2.67}$$

where k is the time index after sampling, and n and m are integer values of subband time index and frequency index, respectively. There are totally three different sampling periods indicated in Eq.(2.67): T_s, the sampling period for input signal $x(k)$; 1/F, the frame period for bandlimited signals at each frequency band; and $1/T$, the frequency sampling period.

With a symbol $\mathcal{F}$ for DFT, Eq.(2.67) represents the DFT on windowed discrete signal $w(n/F - k)x(k)$:

$$X_{nm} = \mathcal{F}\{w(n/F - k)x(k)\}. \tag{2.68}$$

Its inverse transform can be expressed as

$$\begin{aligned} w(n/F - k)x(k) &= \mathcal{F}^{-1}\{X_{nm}\} \\ &= \frac{1}{T}\sum_{m=0}^{T-1} X_{nm}e^{-j2\pi km/T}. \end{aligned} \tag{2.69}$$

Eq.(2.69) can be viewed as the synthesis filter bank.

If no modifications are made to the discrete STFT spectrum, simply overlap and add time samples produced by Eq.(2.69) can perfectly reconstruct the original signal $x(n)$ [3]. When there are modifications to the discrete STFT spectrum, more complicated analysis are required to describe the changes and

effects. In the implementation, we just window the results of the inverse DFT with the same window again, overlap-add, normalize, and directly use the real part of the final result as output.

Working on the discrete STFT spectrum, the adaptive equalizer should not be the same form of transversal filter as defined in Eq.(2.20). It should now be an $L \times M$ matrix

$$\mathbf{H} = [\mathbf{h}_0(n), \mathbf{h}_1(n), \ldots, \mathbf{h}_{M-1}(n)] \tag{2.70}$$

where M is the FFT (DFT) size, each column is a length-L transversal filter like the one in Eq.(2.20), and the length L here doesn't have to be the same L defined in Eq.(2.20). Similar to X_{nm} defined in Eq.(2.67), the desired signal $d(t)$ is also transformed into its STFT spectrum D_{nm} where

$$D_{nm} = \sum_{k=0}^{T-1} w(n/F - k)d(k)e^{j2\pi km/T}, \tag{2.71}$$

and the error signal for each subband is calculated as

$$e_m(n) = D_{nm} - \mathbf{h}_m^H(n)\mathbf{X}_m(n), \quad m = 0, 1, \ldots, M. \tag{2.72}$$

where

$$\mathbf{X}_m(n) = [X_{nm}, X_{n-1\ m}, \ldots, X_{n-L+1\ m}]^T. \tag{2.73}$$

The filter adaptation equation should also be similar to Eq.(2.34):

$$\mathbf{h}_m(n+1) = \mathbf{h}_m(n) + \frac{\tilde{\mu}}{\delta + \|\mathbf{X}_m(n)\|^2}\mathbf{X}_m(n)e_m^*(n), \quad m = 0, 1, \ldots, M. \tag{2.74}$$

If using the block NLMS method, the equation for each subband should be

$$\mathbf{h}_m(k+1) = \mathbf{h}_m(k) + \frac{\tilde{\mu}}{\delta + \sum_{i=0}^{B-1} \|\mathbf{X}_m(kB+i)\|^2} \sum_{i=0}^{B-1} \mathbf{X_m}(kB+i) e_m^*(kB+i).$$

$$m = 0, 1, \ldots, M \tag{2.75}$$

where k is the block index and B is the number of subband samples in each block.

2.3.4 Double-talk Detector

When the near-end musician is active, an event of *double-talk* (DT) begins. During DT, if the equalizer in this system is adapted in the same way as when there is no DT, the adaptive filter will regard the near-end musician signal as part of the error. Adaptations based on this false error signal will diverge the filter taps destructively, which should definitely be avoided. To serve this purpose, *double-talk detectors* (DTD) are built to declare DT and freeze filter taps from adaptation.

Determining the onset moment of DT remains a very hard task. Most DTDs are targeting one-microphone-one-speaker systems. In general, most DTD algorithms follow the common principles:

- First, form a detection statistics $g(t)$.
- Compare $g(t)$ to a tunable threshold T, and declare DT if $g(t) \leq T$ (or $g(t) \geq T$, for different algoirhtms).
- During DT, freeze filter taps, and hold DT for a minimum time period t_{hold}.

- if $g(t) > T$ (or $g(t) < T$) for more than t_{hold}, allow filter adaptation again.

Introduced below are several major approaches to DTD design. In reality, none of them works robustly.

Geigel Algorithm

This is a simple energy-based approach. In principle, far-end signals are usually damped by the room when it travels from loudspeaker to microphones. If the near-end musician (or talker, in speech communication) is active, the received energy in microphone should be larger than when there is just the echo. Based on this assumption, the Geigel DTD [25] forms a decision variable as

$$d_{\text{G}} = \frac{|d(t)|}{\max\{|x(t)|, \ldots, |x(t-n+1)|\}} \tag{2.76}$$

where $x(t)$ is the far-end (input) signal, $d(t)$ is the microphone captured signal, and n is a chosen length of time to find the maximum of input magnitude. DT will cause microphone signal to become abnormally large in magnitude, thus, for a threshold T_{G},

$$\text{Double-talk ?} = \begin{cases} d_{\text{G}}(t) \geq T_G & \text{Yes,} \\ d_{\text{G}}(t) < T_G & \text{No.} \end{cases} \tag{2.77}$$

In this approach, the threshold T_{G} is critical to the DT decision and trade-offs often exist in practical situations.

Normalized Cross-correlation Algorithm

This normalized cross-correlation (NCR) based approach is introduced in [10]. It starts with the fact that the power of the measured signal can be expressed as

$$\sigma_d^2(t) = \mathbf{h}_t^H \mathbf{R}_x(t) \mathbf{h}_t + \sigma_v^2(t) \tag{2.78}$$

where $\sigma_d^2(t)$ and $\sigma_v^2(t)$ are powers of the measured signal and near-end signal respectively. $\mathbf{R}_x(t)$ is input signal's correlation matrix at time t (assuming the far-end signal is piece-wise wide-sense-stationary, which validates the matrix). Define the cross-correlation between input signal vector $\mathbf{x}(t)$ and filter output $y(t)$ as

$$\begin{aligned} \mathbf{p}_{\mathbf{x}y}(t) &= E[\mathbf{x}_t y^*(t)] \\ &= E[\mathbf{x}_t (\mathbf{h}_t^H \mathbf{x}_t)^H] = \mathbf{R}_x(t) \mathbf{h}_t, \end{aligned} \tag{2.79}$$

the filter response at time t can then be expressed as

$$\mathbf{h}_t = \mathbf{R}_x^{-1}(t) \mathbf{p}_{\mathbf{x}y}(t). \tag{2.80}$$

Thus, Eq.(2.78) can be expressed alternatively as

$$\sigma_d^2(t) = \mathbf{p}_{\mathbf{x}y}^H(t) \mathbf{R}_x^{-1}(t) \mathbf{p}_{\mathbf{x}y}(t) + \sigma_v^2(t). \tag{2.81}$$

When DT is not present, $v(t) = 0$. The measured signal ideally should be the same as the filter output: $d(t) = y(t)$. With $\mathbf{p}_{\mathbf{x}d}$ denoting the cross-correlation between the input signal vector and the measured signal, measured signal power in this case should be

$$\sigma_d^2(t) = \mathbf{p}_{\mathbf{x}d}^H(t) \mathbf{R}_x^{-1}(t) \mathbf{p}_{\mathbf{x}d}(t). \tag{2.82}$$

Comparing Eq.(2.78) and Eq.(2.82), the decision variable can be formed as

$$d_{\mathrm{NCR}}(t) = \frac{\mathbf{p}_{\mathbf{x}d}^H(t)\mathbf{R}_x^{-1}(t)\mathbf{p}_{\mathbf{x}d}(t)}{\sigma_d^2(t)}. \tag{2.83}$$

During DT, $d_{\mathrm{NCR}}(t)$ should be less than 1, and when there is no DT, $d_{\mathrm{NCR}}(t)$ should be approximately 1. Thus, with a threshold $T_{\mathrm{NCR}} < 1$, the DT decision can be based on

$$\text{Double-talk ?} = \begin{cases} d_{\mathrm{NCR}}(t) \leq T_{NCR} & \text{Yes,} \\ d_{\mathrm{NCR}}(t) > T_{NCR} & \text{No.} \end{cases} \tag{2.84}$$

This method also has a computationally tractable version named *cheap-NCR* [2]. A major problem of the cross-correlation approach is that the decision becomes very sensitive to delay. Some simple experiments in Matlab shows that, if the signal delay (phase) is not estimated correctly, the decision variable can have very unreliable and unstable values, causing wrong DT judgements.

Variable Impulse Response Algorithm

Another approach introduced in [2] treats the near-end signal as corrupting noise that induces large variations in filter taps. This variable impulse response (VIRE) algorithm uses maximum values of adaptive filter taps as the measure for fluctuations. The decision variable is formed as

$$d_{\mathrm{VIRE}}(t) = \lambda d_{\mathrm{VIRE}}(t-1) + (1-\lambda)[\gamma - \overline{\gamma}(t)]^2, \tag{2.85}$$

where

$$\overline{\gamma}(t) = \lambda\overline{\gamma}(t-1) + (1-\lambda)\gamma,$$
$$\gamma = \max\{h_t(0), h_t(1), \ldots, h_t(L-1)\} \tag{2.86}$$

and λ is an IIR smoothing factor. For a tunable threshold T_{VIRE}, the DT decision can be made as

$$\text{Double-talk ?} = \begin{cases} d_{\text{VIRE}}(t) \geq T_{VIRE} & \text{Yes,} \\ d_{\text{VIRE}}(t) < T_{VIRE} & \text{No.} \end{cases} \tag{2.87}$$

There are also other approaches that not only detect DT, but also search for more subtle events such as loudspeaker-microphone setup changes [32, 58]. Since the current prototype system is not going to physically alter its setup over time, it won't benefit much from these approaches.

In this prototype system, there are two microphones (see Figure 1.1). To take advantage of the dual-microphone setup, DTD can be built differently than conventional methods. Two microphones can form a simple *beamformer* [60] or perform low-complexity *blind source separations* (BSS) on the recorded mixtures. In both approaches, algorithms can be developed so that major energy of the sound from the loudspeaker is eliminated from the stereo mixture. Once this is done successfully, the signal that remains in the mixture should not have much energy from the far-end signal. Based on this energy alone, DT can be detected with a much better accuracy. With the same principle but for a different purpose, a more sophisticated BSS can be utilized to remove the far-end reference signal in post

processing. In this way, a clean recording of the local musician can be obtained. If each studio has a similar system, clean samples of musicians in each location, after their remote recording sessions, can be transferred over to one central studio and mixed together for the final production.

BSS will be discussed in the following chapter.

3

Blind Source Separation: the post processing

After a remote recording session, two microphone signals are obtained, with each containing a signal mixture of the near-end musician and far-end sound from the reference speaker, and some other reflections from the room. The first step of postprocessing is to remove the far-end contents from the recording and obtain a relatively clean audio track of the near-end. After that, conventional mixing techniques and postprocessing effects can be applied to create the final production. This process can be generally regarded as *blind source separation* (BSS), whose schematic is shown in Figure 3.1.

3.1 Background

The topic of removing one sound content from a stereo recording has been speculated years before. It is worthwhile to first mention a trivial solution to this problem in karaoke applications. To remove the vocal part and do karaoke, a method simply subtracts two channels (left and right) of the stereo recording and

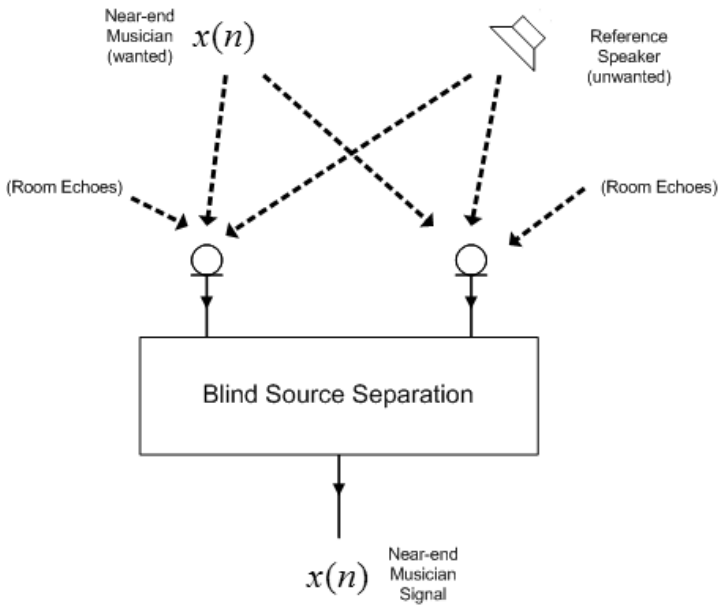


Figure 3.1: Schematic of blind source separation in the prototype system.

hopes the resulting mono sound is free of vocal sound. It is patented in 1996 [52] and is applied in many software tools such as *AnalogX Vocal Remover*, *WinOKE*, and *YoGen Vocal Remover*. Obviously this super-straightforward method cannot work robustly. Its strong assumption is that the unwanted signal (such as the vocal track in a pop music) is located in the center of a stereo sound field and have exactly the same copies in each channel of the stereo signal mixture.

More realistic models to describe the problem are introduced by the topic of BSS, which is usually used to address signal separation problems in both acoustics and wireless communications. A typical model for BSS of an N-channel sensor signal $\mathbf{x}(t)$ arising from M unknown scalar source signals $\mathbf{s}(t)$, is described by (in the case of instantaneous mixture)

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{v}(t) \tag{3.1}$$

where

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix}, \mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ \vdots \\ s_M(t) \end{bmatrix}, \tag{3.2}$$

$\mathbf{A}$ is an $N \times M$ linear mixing matrix, and $\mathbf{v}(t)$ is a zero-mean, white additive noise. If the space is nearly noise-free and $N \geq M$, then matrix $\mathbf{A}$ or part of $\mathbf{A}$ can be inverted to obtain the original sources:

$$\mathbf{s}(t) = \mathbf{A}^{-1}\mathbf{x}(t). \tag{3.3}$$

If $N < M$, i.e. there are less sensors than sources, the demixing process can never be performed by matrix inversion, and the problem becomes *degenerate blind source separation*. Determining the separation blindly from only one mixture (worst degenerate case) is still an open problem, but if given a second mixture, there are many approaches to achieve the separation [44, 42, 33, 11, 5, 4, 51]. A two-mixture case is usually the target case in communications, since mobile handsets or PDA boards are not big enough to make feasible the use of more than two microphones. Similarly, the signal removal problem for postprocessing stage of this prototype system is also a two-sensor case (stereophonic recording).

Two main approaches are explored in theory to solve this signal removal problem in the following sections.

3.2 BSS with Time-frequency Masking

This is the approach summarized by Özgür Yılmaz and Scott Rickard [67], after a sequence of research focusing on the *degenerate unmixing estimation technique* (DUET) [34]. One major assumption of this approach is the *W-disjoint orthogonality* (W-DO).

3.2.1 Assumptions on the Signal

W-DO basically states that the time-frequency representations of the sources do not overlap. In a STFT of a signal mixture, this means in each time-frequency cell, there exists at most one of the original sources.

This is an assumption on the property of signal contents. In reality, not all signals conform to it. As an example, in [56], the W-DO of two speech signals are compared with two independent random noises (Figure 3.2). The figure shows their spectrograms and joint histograms. Although noise is statistically orthogonal, it can be seen that almost all values in the speech joint histogram are close to zero (bottom row, left) while there are a significant number of non-zero values in the noise histogram (bottom row, right). This means W-DO is different and in general a stronger condition than statistical orthogonality.

In reality, speech and some communication signals such as M-ary FSK possess W-DO. Although speech signals are not perfect, but approximately W-disjoint orthogonal, they are already good candidates to be separated this way.

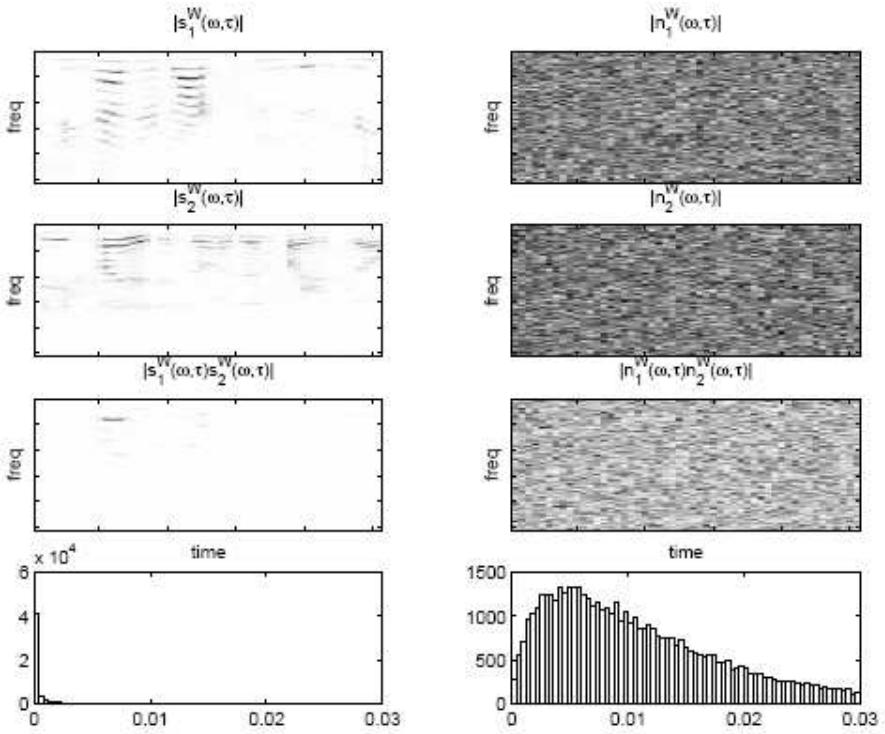


Figure 3.2: Example of W-disjoint orthogonality from [56]. The top three left column figures are grey scale images of $|\hat{s}_1(\omega,\tau)|, |\hat{s}_2(\omega,\tau)|$, and $|\hat{s}_1(\omega,\tau)\hat{s}_2(\omega,\tau)|$ for two speech signals $s_1(t)$ and $s_2(t)$ normalized to have unit energy. The top three right column figures are grey scale images of $|\hat{n}_1(\omega,\tau)|, |\hat{n}_2(\omega,\tau)|$, and $|\hat{n}_1(\omega,\tau)\hat{n}_2(\omega,\tau)|$ for two independent white noise signals $n_1(t)$ and $n_2(t)$ normalized to have unity energy. A Hamming window of length 32 ms was used as $W(t)$ and all signals had length 1.5 seconds. $|\hat{s}_1(\omega,\tau)\hat{s}_2(\omega,\tau)|$ contains far fewer large componetns than $|\hat{n}_1(\omega,\tau)\hat{n}_2(\omega,\tau)|$. This is confirmed in the bottom row which contains histograms of the values in $|\hat{s}_1(\omega,\tau)\hat{s}_2(\omega,\tau)|$ and $|\hat{n}_1(\omega,\tau)\hat{n}_2(\omega,\tau)|$ respectively. Note, almost all values in the voice histogram are close to zero, while there are a significant number of non-zero values in the noise histogram. Thus the speech signals approximately satisfy the W-disjoint orthogonality condition while the independent white noise signals do not.

Whether different (wide-band) musical instruments possess similar W-DO is yet to be studied. A numerical method is given in the literature to assess W-DO quality of a signal in general. Define the summation of the sources interfering with source k as

$$y_k(t) = \sum_{\substack{i=1 \\ i \neq k}}^{N} s_i(t). \tag{3.4}$$

Then, define a time-frequency mask

$$M_{(k,x)}(t,\omega) = \begin{cases} 1, & 20\log(|S_k(t,\omega)|/|Y_k(t,\omega)|) > x, \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

With the notation of $\| \ \|^2$ defined as

$$\|f(x,y)\|^2 := \int\int |f(x,y)|^2 dx\ dy, \tag{3.6}$$

a preserved-energy ratio (PSR) after masking can be expressed as

$$\mathrm{PSR}_k(x) = \frac{\|M_{(k,x)}(t,\omega)S_k(t,\omega)\|^2}{\|S_k(t,\omega)\|^2}. \tag{3.7}$$

$\mathrm{PSR}_k(x)$ reveals the percentage of energy of source k (preserved after the mask $M_{(k,x)}(t,\omega)$) for time-frequency points where it dominates other sources by x dB. It is proposed in [56] to be an approximate measure of W-DO. Some measurements for different numbers of sources are illustrated in Figure 3.3 where $\mathrm{PSR}_k(x)$ values for different xs are plotted. For example, compare one source to the sum of other three ($N = 4$), we have 80% W-DO at 5 dB. In a similar manner, a signal-to-interference ratio (SIR) can be defined as

$$\mathrm{SIR}_k(x) = \frac{\|M_{(k,x)}(t,\omega)S_k(t,\omega)\|^2}{\|M_{(k,x)}(t,\omega)Y_k(t,\omega)\|^2}. \tag{3.8}$$

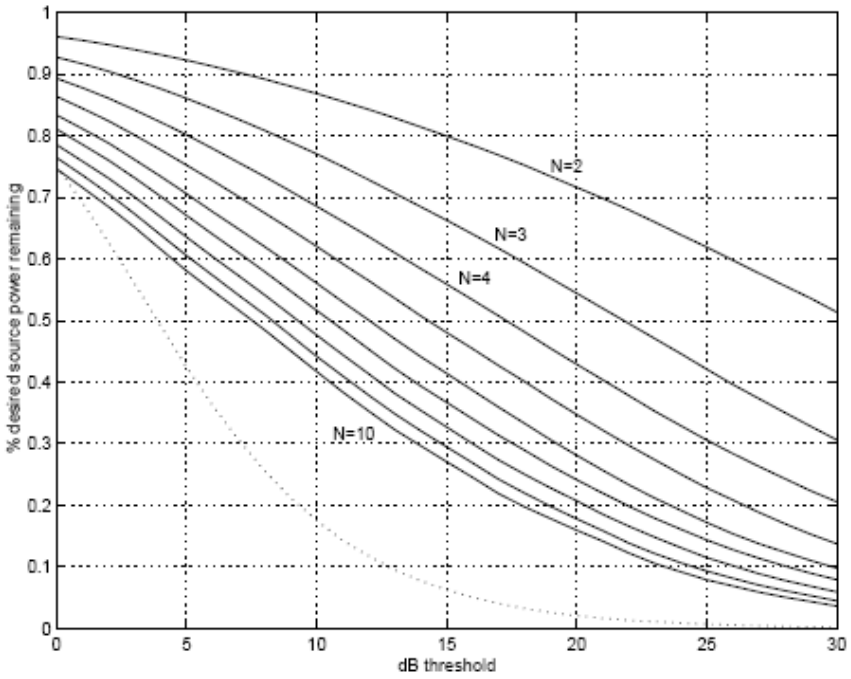


Figure 3.3: Approximate W-disjoint orthogonality. Plot of $\mathrm{PSR}_k(x)$ for $x = 0, 1, \ldots, 30$ and $N = 1, 2, \ldots, 10$. N is the number of sources. [56]

It reveals the overall signal-to-noise ratio for time-frequency points where source k dominates the sum of the other source by x dB. Figure 3.4 shows a plot of $\mathrm{PSR}_k(x)$ versus $\mathrm{SIR}_k(x)$ for mixtures of various orders.

3.2.2 Estimation and Demixing

The estimation process starts by assuming an anechoic and noise-free environment, i.e. only direct paths of the signals to the sensors are present. Based on the model in Eq.(3.1), with $M = 2$, mixture signals are actually $x_1(t)$ and $x_2(t)$. Without loss of generality, attenuation and delay parameters of the first mixture $x_1(t)$ can be absorbed into the definition of the sources themselves. The

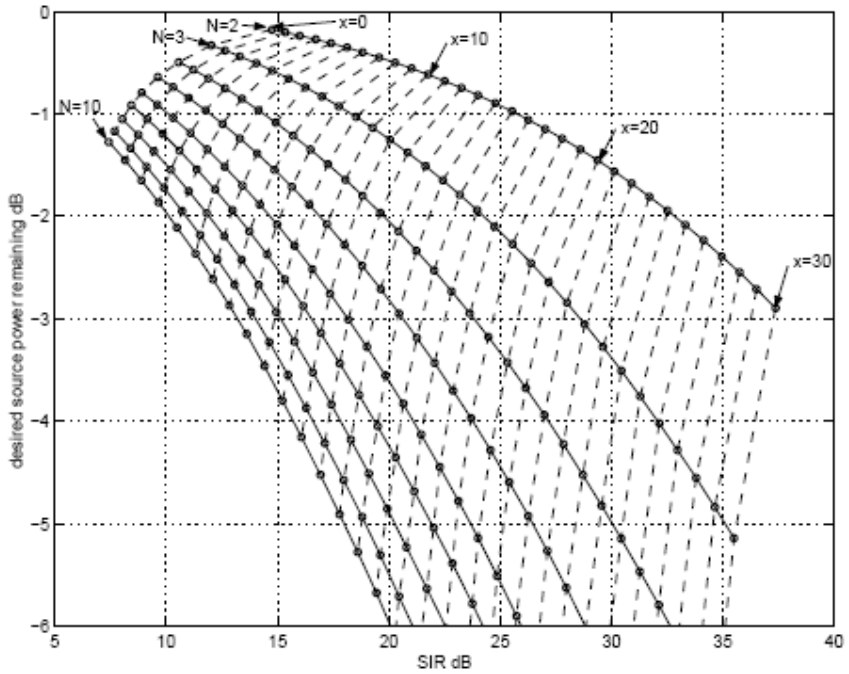


Figure 3.4: Demixing time-frequency mask performance. The plot contains $\mathrm{PSR}_k(x)$ (in dB) versus $\mathrm{SIR}_k(x)$ for $x = 0, 1, \ldots, 30$ and $N = 1, 2, \ldots, 10$. N is the number of sources. [56]

two mixtures in Eq.(3.1) can be expressed in detail as

$$\begin{aligned} x_1(t) &= \sum_{j=1}^{N} s_j(t) \\ x_2(t) &= \sum_{j=1}^{N} a_j s_j(t - \delta_j) \end{aligned} \tag{3.9}$$

Then, under the *narrowband assumption* which will be discussed later, the STFT version of the above equation is

$$\begin{bmatrix} X_1(t,\omega) \\ X_2(t,\omega) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ a_1 e^{-j\omega\delta_1} & \cdots & a_N e^{-j\omega\delta_N} \end{bmatrix} \begin{bmatrix} S_1(t,\omega) \\ \vdots \\ S_N(t,\omega) \end{bmatrix}. \tag{3.10}$$

Recollecting the definition of W-DO, that the time-frequency cells of different sources don't overlap, we have

$$S_k(t,\omega)S_l(t,\omega) = 0, \forall(t,\omega), \forall k \neq l. \tag{3.11}$$

If time-frequency masks M_k can be found so that

$$M_k(t,\omega) = \begin{cases} 1, & S_k(t,\omega) \neq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{3.12}$$

then the time-frequency plane of a mixture can be partitioned and cells that belong to each source can be grouped as

$$S_k = M_k X_1. \tag{3.13}$$

After that, ISTFT of S_k can restore the k^{th} source s_k, fulfilling the task of demixing.

To obtain M_k, the following steps should be taken. From Eq.(3.10), for W-DO sources, at most one of the N sources is non-zero for a given (t_0, ω_0) cell, thus

$$\begin{bmatrix} X_1(t_0,\omega_0) \\ X_2(t_0,\omega_0) \end{bmatrix} = \begin{bmatrix} 1 \\ a_k e^{-j\omega_0\delta_k} \end{bmatrix} S_k(t_0,\omega_0), \quad \text{for some } \ k. \tag{3.14}$$

Therefore, the relative amplitude and delay parameters associated with one source can be calculated with

$$(a_k, \delta_k) = \left(\left\| \frac{X_2(t_0,\omega_0)}{X_1(t_0,\omega_0)} \right\|, -\frac{1}{\omega_0} \angle \frac{X_2(t_0,\omega_0)}{X_1(t_0,\omega_0)} \right). \tag{3.15}$$

Apply this to all time-frequency cells by substituting (t,ω) for (t_0,ω_0) in Eq.(3.15) and observe all the (a,δ) pairs, there are going to be clusters around the true center

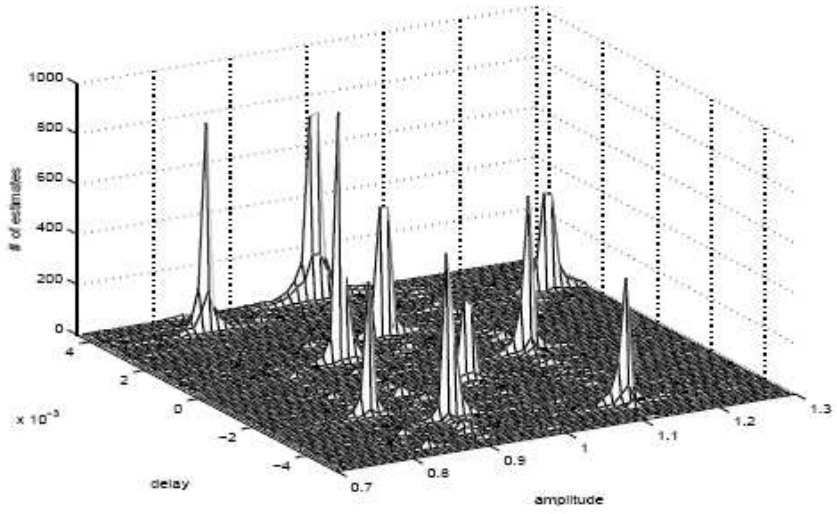


Figure 3.5: Two-dimensional histogram of DUET estimates for delay / amplitude mixing parameters for ten sources (M-ary wireless signals) obtained using two mixtures [55].

$(\tilde{a}_k, \tilde{\delta}_k)$ for each source k, as showed in Figure 3.5. Skipping details, the mask for each source k can be calculated as

$$M_k(t,\omega) = \begin{cases} 1, & (a,\delta)_{(t,\omega)} \approx (\tilde{a}_k, \tilde{\delta}_k) \\ 0, & \text{otherwise} \end{cases} \tag{3.16}$$

$$M_k = \bigcup_{\text{all } (t,\omega)} M_k(t,\omega) \tag{3.17}$$

3.2.3 Discussions about Other Conditions

The above models and algorithms are simplified versions of the reality. For them to work robustly, conditions in different aspects need to be satisfied.

The Narrowband Assumption

When the physical separation of the sensors is small enough relative to the carrier and bandwidth of the signal, such that the relative delay between the sensors can be expressed as a phase shift of the signal. This assumption is known as the *narrowband assumption* in array signal processing [39]. Only under this assumption will Eq.(3.15) be valid. The mathematical expression of this condition is

$$F^W(s_i(\cdot - \delta))(t, \omega) = e^{-j\omega\delta} F^W(s_i(\cdot))(t, \omega), \forall |\delta| < \Delta \tag{3.18}$$

where F^W denotes the windowed FT. Let ω_{max} be the maximum frequency present in the sources, and $\delta_{kmax} := \max_k |\delta_j|$,

$$\omega_{max}\delta_{kmax} < \pi \tag{3.19}$$

is an equivalent expression of the narrowband assumption. In [67], this is discussed in detail, which shows as long as the delay between the two microphone readings is less than a sample, the estimated phase will be accurate. For example, for a sampling rate of $\omega_s/(2\pi) = 16$ kHz, assume $\omega_{max} = \omega_s/2$ and sound speed $c = 344$ m/s, Eq.(3.19) will be satisfied as long as microphone spacing d is equal to or less than 2.15 cm.

Influence of Windowing

Since this method starts with STFT representation, the influence of windowing also is worth studying. Numerical investigations of Balan and others [7]

show that the bias is very small for reasonably wide windows, concluding that the influence of windowing on the delay estimation problem is negligible for a reasonable large class of windows and signals. For speech signals, a sufficiently large window size is needed to make the delay estimation valid. However, too large a window size will result in reduced W-DO. After comparing different types of windows and window sizes, Yılmaz and Rickard [67] found that Hamming window of size 1024 had the best performance.

Echoic Environments and the Uncertainty of Source Location

The real-world spaces for source separation problem are seldom anechoic. Some study about the influence of echoic environment are done by Balan and others [6]. In their experiments, a two-source demixing problem is studied, and the reference room transfer function is obtained by calculating echoes of various orders based on the room dimension. These echoes are then simulated by FIR filters. Demixing matrices are truncated to simulate up to 10 multipaths of echoes. Numerical results show that high order matrices do not sensibly improve the SNR performance, compared to the direct path only or other lower order demixing schemes. They also studied the influence of position change of signal sources by setting up a 10 cm-spacing microphone pair and change the position of one of the signal sources from 0 to 1 m, in increments of 5 cm. (Setup diagram in Figure 3.6.) The results show an important fact that the performance degrades dramatically even when the position change is as little as 5 cm. In this study, when sound source

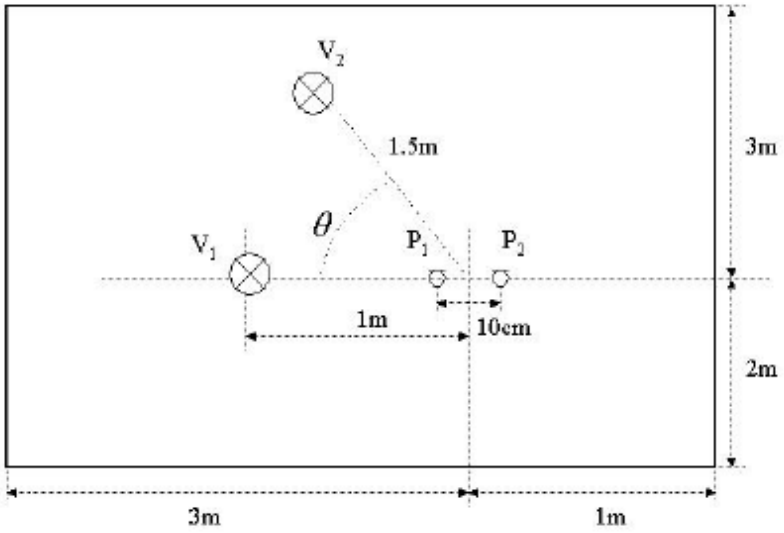


Figure 3.6: Experiment setup, study of the influence of source location change on blind source separation [6].

location is uncertain, SNR for different orders of demixing suffers almost equally.

In the time-frequency masking approach, some recent studies have found ways to address the sound source location uncertainty problem. For example, in [49], efforts have been made to unwrap the phases of frequency domain signals and search for the most likely amplitude-delay pairs, achieving a refined estimation of a signal source's direction of arrival. [49] also utilizes other techniques such as the expectation maximization and Gaussian spatial filters to construct a complete system for auditory scene analysis.

3.3 The Spatial Acoustics Estimation Approach

With a different approach from spatial acoustics estimation, Dubnov and Xiang [64] also studied signal separation techniques that separate sound sources from a stereo mixture. The study was initially intended to remove vocal component in a stereo recording (karaoke). For convenience in notation, the word "vocal" is used in this section to refer to the sound source to be removed from signal mixtures.

This approach assumes that, in a music recording, passages where the vocal is soloing or dominating most of the energy can be used to estimate its spatial characteristics. A prototype system of the whole removal procedure is depicted in Figure 3.7. First, an algorithm identifies the solo segments (done manually in the simulation). These segments are analyzed to estimate the transfer function from the vocal's location to each of the microphones. The method resembles the attenuation-delay pair model in Eq.(3.15), but targets convolutive environments instead. After the analysis, sound in the estimated location can be cancelled or suppressed by projecting the stereo signal on appropriate directions for different frequency bands. The resulting sound is a mono, vocal-erased or vocal-suppressed sound track. Finally, to compensate the bass, low frequency contents of the original recording that don't overlap with the vocal in frequency spectrum will be added back to the mono file.

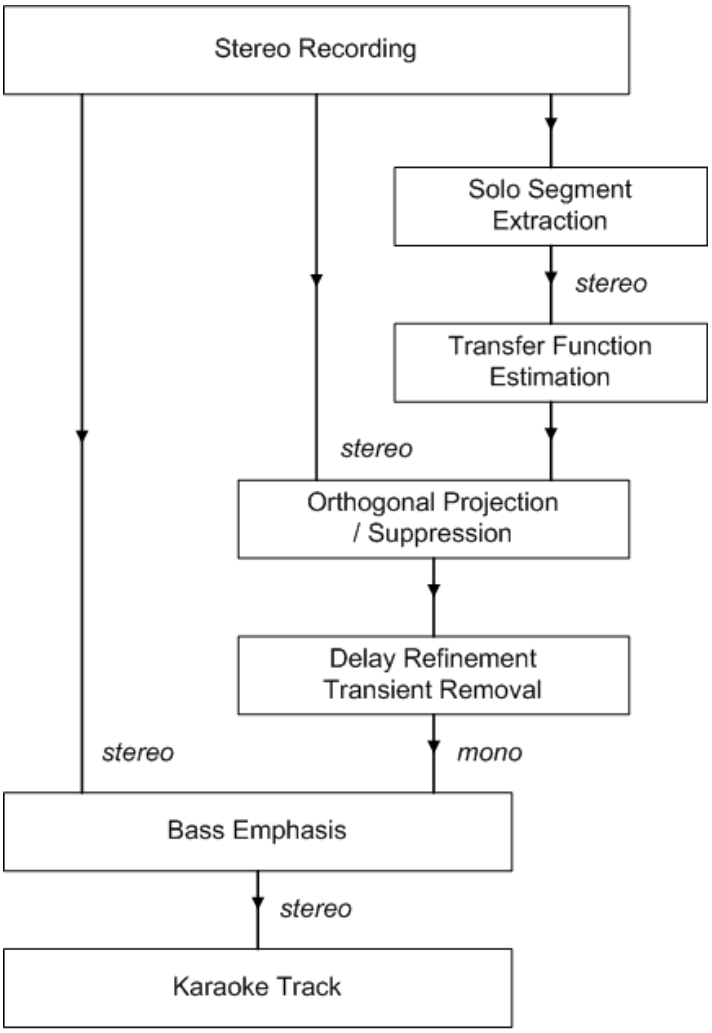


Figure 3.7: System diagram of a vocal removal prototype system based on spatial acoustics estimation.

3.3.1 Assumptions on the Signals

The signal mixtures are assumed to match that of a live stereo recording session where the vocalist and multiple instruments are stabilized in different locations. Reverberation and stationary room noise are assumed as well. With a reasonable reverb decay time, the model is robust to additive noise and room reflections.

Let's first rewrite Eq.(3.1) from BSS basics

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{v}(t), \tag{3.20}$$

the general mixing model. In convolutive environments such as a room with reflections, signals at different locations will have different transfer functions to each microphone. The new expression of a signal mixture will be

$$x_n(t) = \sum_{m=1}^{M} \int a_{nm}(\tau) s_m(t-\tau) d\tau + v_n(t)$$
$$n = 1, \ldots, N \tag{3.21}$$

where $a_{mn}(\tau)$ is the impulse response of the transfer function from the mth source signal to the nth microphone. STFT of Eq.(3.21) turns this into instantaneous mixtures for each frequency:

$$X_n(t,\omega) = \sum_{m=1}^{M} A_{nm}(\omega) S_m(t,\omega) + V_n(t,\omega),$$
$$n = 1, \ldots, N. \tag{3.22}$$

Here $S_m(t,\omega)$ and $V_n(t,\omega)$ denote the STFTs of $s_m(t)$ and $v_n(t)$ respectively, and t denotes the STFT window position. The temporal transfer function of the mth source signal to the nth microphone at frequency ω is defined as

$$A_{nm}(\omega) = \int a_{nm}(\tau)e^{-j\omega\tau}d\tau \doteq \widehat{a}_{nm}(\omega)e^{-j\omega\widehat{\delta}_{nm}(\omega)} \tag{3.23}$$

where we define

$$\begin{aligned} \widehat{a}_{nm}(\omega) &= \|A_{nm}(\omega)\| \\ \widehat{\delta}_{nm}(\omega) &= \angle A_{nm}(\omega). \end{aligned} \tag{3.24}$$

Adding the frequency index ω, Eq.(3.20) can be translated into

$$\mathbf{X}(t,\omega) = \mathbf{A}(\omega)\mathbf{S}(t,\omega) + \mathbf{V}(t,\omega). \tag{3.25}$$

In the stereo case, $N = 2$, so that (3.25) in detail looks like

$$\begin{aligned} \begin{bmatrix} X_1(t,\omega) \\ X_2(t,\omega) \end{bmatrix} = &\begin{bmatrix} \widehat{a}_{11}(\omega)e^{-j\omega\widehat{\delta}_{11}(\omega)} & \cdots & \widehat{a}_{1M}(\omega)e^{-j\omega\widehat{\delta}_{1M}(\omega)} \\ \widehat{a}_{21}(\omega)e^{-j\omega\widehat{\delta}_{21}(\omega)} & \cdots & \widehat{a}_{2M}(\omega)e^{-j\omega\widehat{\delta}_{2M}(\omega)} \end{bmatrix} \\ &\cdot \begin{bmatrix} S_1(t,\omega) \\ \vdots \\ S_M(t,\omega) \end{bmatrix} + \begin{bmatrix} V_1(t,\omega) \\ V_2(t,\omega) \end{bmatrix}. \end{aligned} \tag{3.26}$$

As mentioned before, the attenuation and delay parameters for each frequency bin in the first microphone signal $x_1(t)$ can be absorbed into the definition of the

source itself. In this way, Eq.(3.26) can be rewritten as

$$\begin{bmatrix} X_1(t,\omega) \\ X_2(t,\omega) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ a_1(\omega)e^{-j\omega\delta_1(\omega)} & \cdots & a_M(\omega)e^{-j\omega\delta_M(\omega)} \end{bmatrix} \cdot \begin{bmatrix} S_1(t,\omega) \\ \vdots \\ S_M(t,\omega) \end{bmatrix} + \begin{bmatrix} V_1(t,\omega) \\ V_2(t,\omega) \end{bmatrix}. \tag{3.27}$$

In Eq.(3.27), suppose the vocal component is the kth source, which is associated with STFT $S_k(t,\omega)$, if we can estimate the kth vector $\begin{bmatrix} 1 & a_k(\omega)e^{-j\omega\delta_k(\omega)} \end{bmatrix}^T$ in $\mathbf{A}(\omega)$, then left multiplying a vector that is orthogonal to it will completely remove the kth source, achieving the goal.

3.3.2 Estimation

With solo segments of the vocal extracted and their time-frequency cells ready, it is possible to estimate the unstructured spatial transfer function for each frequency [24]. A variety of methods to estimate the transfer functions are explored. Consider Eq.(3.27) in the case where only a single source exists, one way to obtain the transfer function is the division method of Eq.(3.15) mentioned before, which gives

$$(a_k(\omega), \delta_k(\omega)) = \left(\left\| \frac{X_2(\omega)}{X_1(\omega)} \right\|, -\frac{1}{\omega} \angle \frac{X_2(\omega)}{X_1(\omega)} \right). \tag{3.28}$$

This method assumes anechoic environments, so it's not robust to noise and will not work properly in convolutive spaces. Here an autocorrelation method that is robust

to uncorrelated additive noise will be described in detail. The autocorrelation matrix of Eq.(3.25) at a given time-frequency cell can be obtained by averaging the following equation:

$$\begin{aligned}\mathbf{X}(t,\omega)\mathbf{X}(t,\omega)^H &= \mathbf{A}(\omega)\mathbf{S}(t,\omega)\mathbf{S}(t,\omega)^H\mathbf{A}(\omega)^H \\ &+ \mathbf{A}(\omega)\mathbf{S}(t,\omega)\mathbf{V}(t,\omega)^H \\ &+ \mathbf{V}(t,\omega)\mathbf{S}(t,\omega)^H\mathbf{A}(\omega)^H \\ &+ \mathbf{V}(t,\omega)\mathbf{V}(t,\omega)^H. \end{aligned} \tag{3.29}$$

The signals and noise are assumed to be uncorrelated. Denoting $\mathbf{R}_x$, $\mathbf{R}_s$ and $\mathbf{R}_v$ as the correlation matrices of the microphone signals, source signals and noise respectively, and considering signals as stationary, after averaging over time windows, the following can be obtained:

$$\mathbf{R}_x(\omega) = \mathbf{A}(\omega)\mathbf{R}_s(\omega)\mathbf{A}(\omega)^H + \mathbf{R}_v(\omega). \tag{3.30}$$

Assume the noise is white, so that matrix $\mathbf{R}_v(\omega)$ is diagonal and $\mathbf{R}_v(\omega) = \sigma_v^2\mathbf{I}_N$, $\forall\omega$ where $\mathbf{I}_N$ is an identity matrix of size N. In segments where only one component s_k exists, Eq.(3.30) becomes

$$\mathbf{R}_x(\omega) = \sigma_{s_k}^2\vec{\mathbf{A}}_k(\omega)\vec{\mathbf{A}}_k(\omega)^H + \sigma_v^2\mathbf{I_N} \tag{3.31}$$

where

$$\vec{\mathbf{A}}_k(\omega) = \begin{bmatrix} 1 \\ a_k(\omega)e^{-j\omega\delta_k(\omega)} \end{bmatrix}. \tag{3.32}$$

Right multiply Eq.(3.31) with $\vec{\mathbf{A}}_k(\omega)$, we get

$$\mathbf{R}_x(\omega)\vec{\mathbf{A}}_k(\omega) = (\sigma_{s_k}^2 \vec{\mathbf{A}}_k(\omega)^H \vec{\mathbf{A}}_k(\omega) + \sigma_v^2)\vec{\mathbf{A}}_k(\omega). \tag{3.33}$$

This means that $\vec{\mathbf{A}}_k(\omega)$ can be estimated from eigenvalues of the rank-1 matrix $\mathbf{R}_x(\omega)$. The corresponding eigenvalue is

$$\lambda = \sigma_{s_k}^2 \vec{\mathbf{A}}_k(\omega)^H \vec{\mathbf{A}}_k(\omega) + \sigma_v^2. \tag{3.34}$$

One can note that in principle the transfer function can be estimated by dividing the two components of the eigenvector. As will be discussed later, an advantage of this approach is its robustness to additive noise. The correlation method can be easily extended for the case of higher order statistics [48] by generalizing Eq.(3.29). For example, for the case of 4th order cumulants, a matrix can be constructed as $X(\omega)X^3(\omega)^H - 3X(\omega)X(\omega)^H$. It can be shown that for the case of Gaussian signal, this matrix equals zero since 4th cumulant of a Gaussian signal equals three times the 2nd cumulant (correlation). This effectively eliminates the additional noise matrix from Eq.(3.30). The eigenvectors of the resulting matrix are the same as for the correlation case.

3.3.3 Demixing (Vocal Removal)

After obtaining $\vec{\mathbf{A}}_k(\omega)$, we find a vector

$$\vec{\mathbf{A}}_k^{\perp} = \begin{bmatrix} -a_k(\omega)e^{-j\omega\delta_k(\omega)} & 1 \end{bmatrix} \tag{3.35}$$

that is orthogonal to it. Left multiply the microphone signal in Eq.(3.27) with $\vec{\mathbf{A}}_k^{\perp}$, the kth component will be removed:

$$\vec{\mathbf{A}}_k^{\perp} \begin{bmatrix} X_1(t,\omega) \\ \\ X_2(t,\omega) \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{A}}_k^{\perp}\vec{\mathbf{A}}_1 & \cdots & 0_{(kth)} & \cdots & \vec{\mathbf{A}}_k^{\perp}\vec{\mathbf{A}}_M \end{bmatrix} \cdot \begin{bmatrix} S_1(t,\omega) \\ \vdots \\ S_M(t,\omega) \end{bmatrix} + \vec{\mathbf{A}}_k^{\perp} \begin{bmatrix} V_1(t,\omega) \\ \\ V_2(t,\omega) \end{bmatrix}. \tag{3.36}$$

One special case of this is when only two components exist in the stereo recording, in which we can extract the two sources individually. Both sides of Eq.(3.36) are one dimensional, so the resulting sound is mono.

3.3.4 Discussions

Robustness

As can be seen from Eq.(3.33), added noise doesn't compromise the accuracy of estimating $\vec{\mathbf{A}}_k(\omega)$ from the eigenvector of $\mathbf{R}_x(\omega)$. So, this algorithm is robust in a noisy environment. The model in the system assigns a linear transfer function to each frequency bin, so it is robust to reverberant vocal sound, and (linear) artificial reverberations. For a typical live recording session, there are usually close mics for individual instruments. Usually, these microphone signals are later added to the stereo recording for the whole ensemble to boost certain instruments. This is still a linear operation which well fits in our assumptions for the model to

work robustly in theory.

Influence from Microphone Spacing and Source Location Change

In this approach, after the estimation process with the solo segments, amplitude-phase pairs ($\widehat{a}_{nm}(\omega)$, $\widehat{\delta}_{nm}(\omega)$) defined in Eq.(3.24) will be obtained for each sound source. This looks similar to the spectrogram division method in [34] but is fundamentally different. The phase term here only denotes the phase of one element in the complex transfer function matrix $A_{nm}(\omega)$ in Eq.(3.23), and has little connection with the actual time delay between two microphones. It is already colored by the convolutive environment. In other words, it should be an accurate description of one characteristic of the environment, and there is no need to unwrap this phase. The narrowband assumption is not necessary, and the microphone spacing doesn't have to be small. This allows more flexibility in the physical setup of the microphones. On the other hand, an accurate estimation of the phase becomes even more crucial, which makes this algorithm sensitive to source location change. It is hard to search proximity of the sound source location and unwrap the phase, due to this different meaning of the phase term.

4

Simulations and Results

Individual parts of the prototype system are investigated with Matlab for non-realtime analysis and Pd for realtime simulations. This chapter describes some of the experiments and corresponding results.

4.1 Spectral Subtraction

In some initial experiments, possible signal and noise energy relationships in spectral subtraction are studied. Spectral subtraction is implemented with a method mentioned in [53]. It is slightly different from Eq.(2.10), but performs equivalent functions. Using the same notations as in Eq.(2.10), the output

frequency domain signal is expressed as

$$\begin{aligned}\hat{S}(e^{j\omega}) &= H(e^{j\omega})X(e^{j\omega}) \\ &= \frac{|X(e^{j\omega})|^2}{|X(e^{j\omega})|^2 + \mu\overline{|V(e^{j\omega})|^2}}X(e^{j\omega}),\end{aligned} \tag{4.1}$$

and the transfer function is:

$$H(e^{j\omega}) = \frac{|X(e^{j\omega})|^2}{|X(e^{j\omega})|^2 + \mu\overline{|V(e^{j\omega})|^2}}. \tag{4.2}$$

Here $\overline{|V(e^{j\omega})|^2}$ is a noise energy mask averaged over time, and $\mu > 0$ is a tunable scaling factor for this mask. In the application of noise suppression, $\overline{|V(e^{j\omega})|^2}$ is measured in time periods where only the noise floor is present. Eq.(4.1) is applied on each time frame of the signal. In time frames where the desired signal is dominant, signal mixture's energy will be much greater than noise energy, i.e. $|X(e^{j\omega})|^2 \gg \mu\overline{|V(e^{j\omega})|^2}$, thus $H(e^{j\omega}) \to 1$, and $\hat{S}(e^{j\omega}) \to X(e^{j\omega})$. In time frames where the desired signal is very weak, the amplification of scaling factor μ will usually cause $|X(e^{j\omega})|^2 \ll \mu\overline{|V(e^{j\omega})|^2}$. In this case, $H(e^{j\omega}) \to 0$, and $\hat{S}(e^{j\omega}) \to 0$. In the prototype system, unwanted signal $V(e^{j\omega})$ in Eq.(4.1) is highly time varying. Consequently, the estimation of noise mask $\overline{|V(e^{j\omega})|^2}$ should follow the change of unwanted signal in a proper speed.

In our experiment, a stand-up comedy live recording of relatively high sampling rate (44.1 kHz) is used as the desired signal, and a synthesized time-varying (randomly changing center frequency, bandwidth, and amplitude) filtered noise is used as the interference signal. A time average of 200 ms is used for

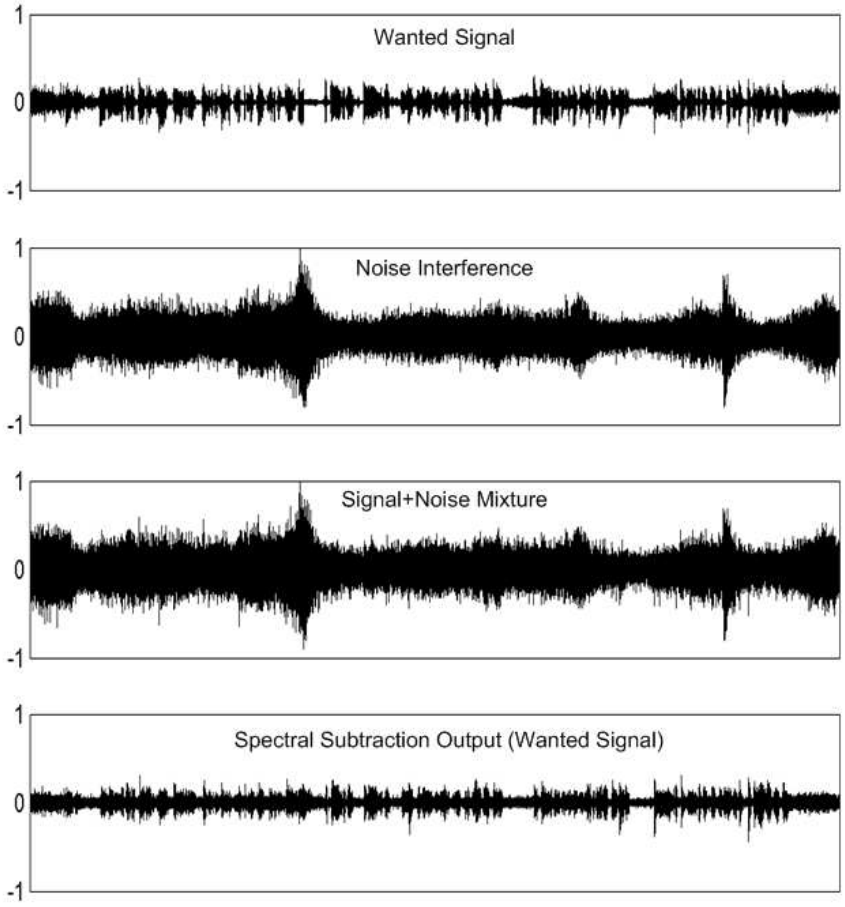


Figure 4.1: Waveforms from experiment on spectral subtraction in severe conditions. A stand-up comedy live recording is used as the wanted signal. A time-varying filtered noise is used as the interference signal. The simulation time is 27.2 seconds, and noise is amplified 10 dB higher in level than the desired signal.

estimating the noise mask over time. This experiment tested with different signal and noise levels. Figure 4.1 shows a severe condition where the filtered noise level is about 10 dB higher than the signal level. Compare the noise waveform and signal mixture waveform, the wanted signal is well buried under noise floor and barely visible. After the spectral subtraction, the original signal is visible again in the last waveform. Although there are still some residual noise in the output, this particular condition represents a very bad noise condition, and practical cases usually perform much better.

Another experiment with the same wanted signal but different interference is conducted. This time, the interference is a rap vocal track (44.1 kHz) of non-stationary intermittent voice. Waveforms from the simulation are shown in Figure 4.2. Again the interference is much louder than the wanted signal, and the output shows a good restoration of the original signal. Because the interference signal here is intermittent, it has less energy than the previous time-varying filtered noise. This allows the interference to be 15 dB higher than the wanted signal in level, while the output is still clear and has little remaining interference.

These two experiments demonstrate the robustness of the method of spectral subtraction in disadvantageous conditions. Using the method of energy masking in these experiments, we can also avoid the implementation of half-wave rectification mentioned in 2.2.1.

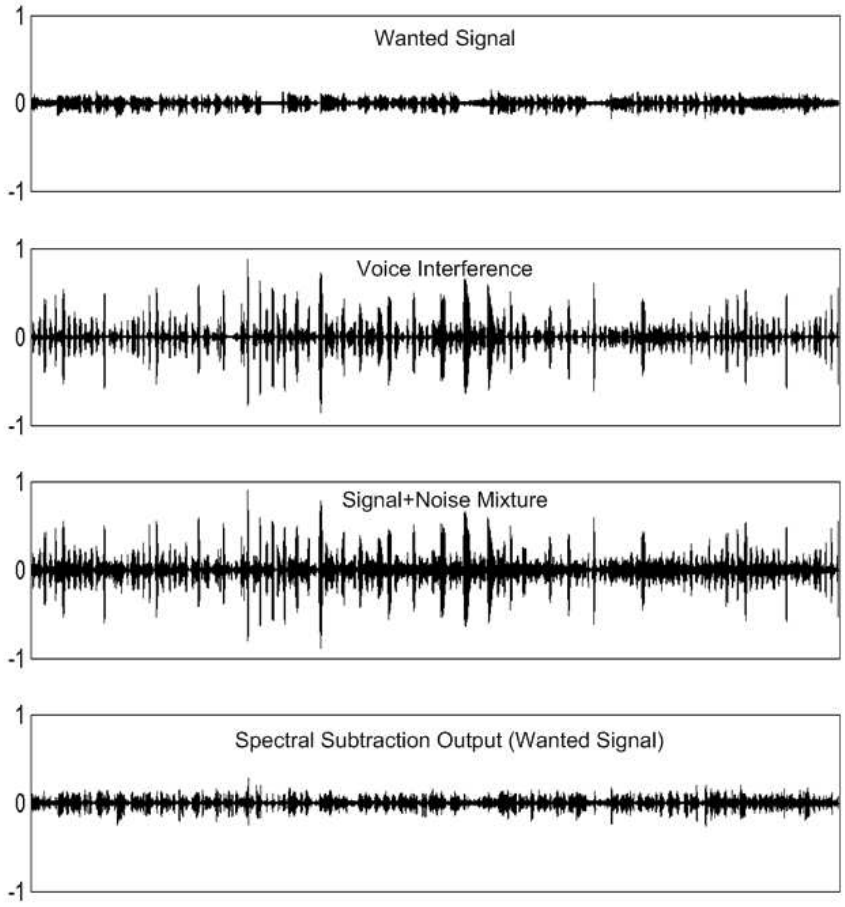


Figure 4.2: Waveforms from an experiment on spectral subtraction in severe conditions. A stand-up comedy live recording is used as the wanted signal. A recoding of a poet rapping is used as the interference signal. The simulation time is 27.2 seconds, and the competing voice is amplified 15 dB higher in level than the wanted signal.

4.2 System with Single-band Equalizer

Another experiment was conducted by adding the room and sound playback/recording devices into the picture. The system is one step more complex than the spectral subtraction experiment. Both wanted and interference sound signals are sound files played back by Pd. Sounds are reproduced via an M-Audio Firewire 410 audio interface and amplified with one Dynaudio BM6A active reference speaker. A consumer quality dynamic omnidirectional microphone is placed 10 cm on-axis in front of the speaker. The close micing is used to minimize the influence of signal delay variations and other room acoustics fluctuations.

4.2.1 Sample-accurate Delay Measurement

As shown in Figure 2.4, one adaptive equalizer and a delay line need to be identified to complete the prototype system. The delay was obtained by sending an impulse and measuring the time elapsed when the microphone picks up the echo. Using Pd object *timer*, this delay can be measured with a precision of one signal frame size (64 samples, 1.45 ms for 44.1 kHz sampling rate). As will be introduced immediately, the maximum adaptive FIR equalizer length is also one signal frame size. If the delay is measured with some error, the least extra (or insufficient) delay will be equal or greater than the whole filter length. This poor precision of delay measurement will greatly compromise the performance of the adaptation process for equalizer identification. Consequently, a Pd external

stimer~ is developed, which allows a sample-accurate measurement of time delay. These two methods are combined to ensure reasonable measurement results. In the experiment, Pd has a set latency of 70 ms. The measured total signal path delay is around 100 ms. Since the acoustic path of 10 cm has a small delay less than 1 ms, the remaining 29 ms delay time should be mostly from the operating system (Windows XP) and possibly from other hardware components.

4.2.2 Adaptive Equalizer

A single-band adaptive filter is used for the equalizer identification. The filter is implemented with a Pd external library *adaptive* developed by [59]. This library limits the maximum filter length to be the signal frame size. Going with the default Pd frame size, a 64-tap FIR filter is used to model the equalizer after the delay line. A moderate constant stepsize is used for NLMS filter adaptation.

First, the adaptive equalizer is initialized to be an all-pass filter, whose first tap is 1 and all others are 0s. It is trained by playing the interference signal alone, and letting the filter adapt according to the microphone signal. The adaptation is disabled when playing both the wanted signal and interference signal together. Then, the spectral subtraction block takes equalizer output's power spectrum as the noise mask to perform echo cancellation.

4.2.3 Observations

For the first group of sounds, the same comedy show clip in the previous experiments was used as the wanted signal, and a pop song containing guitar, drums and female vocal is used as the interference signal (example of a wide-band content). The waveforms of wanted, interference, mixed and processed signals are shown in Figure 4.3. The experiment is conducted many times, each with new calibrations of delay and equalizer filter taps. Other audio contents are also tested to compare the results. Some interesting observations are described below.

Compared to the previous spectral subtraction experiment, signal path no longer stays inside the computer. The additional I/O, hardware and room acoustics in the signal path naturally add more challenges to the interference suppression quality. Compare to that of Figure 4.1 and Figure 4.2, the processed signal (bottom) in Figure 4.3 has less visual similarity to the wanted signal. In this situation, usually a good suppression of interference signal demands an increase of μ in Eq.(4.1). As a result, some parts of the wanted signal are also suppressed. However, the algorithm still performs very robustly in this condition. Figure 4.3 shows a severe case where the interference signal has a 10 dB higher amplification than the wanted signal. In the processed signal, small portions of the wanted signal are attenuated, and there are low-level pops and tones, but there is absolutely no audible fragments of the interference signal. In the case of using two different pop songs as wanted and interference signals, one song can be successfully removed

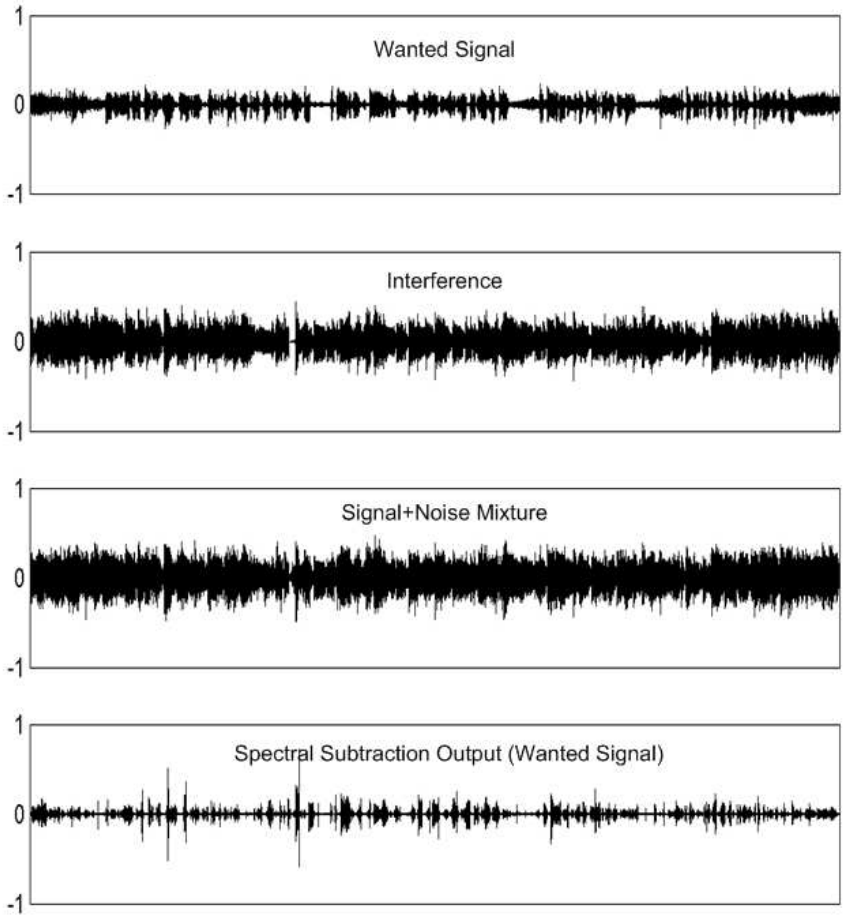


Figure 4.3: Waveforms from an experiment on spectral subtraction with the help of digital delay and single-band adaptive equalizer. A stand-up comedy live recording is used as the wanted signal. A pop song containing guitar, drums and female vocal is used as the interference signal. The simulation time is 27.2 seconds, and the competing soundtrack is amplified 10 dB higher in level than the wanted signal.

from the mixture and only the wanted song is audible. This is very interesting, because both songs have very wide-band contents with a lot of overlapped frequency regions, and the algorithm is purely working on the energy.

As expected, the system is very sensitive to measurement errors in the delay line. Due to the operating system, sometimes the signal latency will fluctuate with a couple of milliseconds or a fraction of a millisecond. Each group of audio contents are tested several times. If the latency is changed during one of the experiments, instead of a complete removal of the interference signal, some onsets of audio events of the interference signal will show up as muffled blips in the processed sound. In this experiment, making a new measurement of the delay line will alleviate this problem. In a practical situation, devices can be easily implemented so that they don't have internal latency fluctuations.

The adaptive equalizer works better with sinusoidal contents (such as melodical instruments, voice) than noise (random noise, drum hits). This is also expected since the filter is a linear FIR filter. In Pd, values of filter taps can be visualized with graphical objects and observed during the adaptation process. Some random movements are observed when signal energy is very small and the adaption is still enabled. This shows the unwanted adaptation due to lack of stepsize control. A crude stepsize control is implemented. It enables adaptation when signal energy is greater than a certain threshold, and disables adaptation for silent periods. More sophisticated stepsize control methods are not experimented

here, but simulated in later experiments with multi-band adaptive equalizer.

Since this is only an intermediate experiment, detail error signals are not captured to analyze.

4.3 System with Multi-band Adaptive Equalizer

To model the system more accurately, further experiments are carried on multi-band adaptive equalizer. Error analysis and various other aspects are studied with Matlab in non-realtime. The same implementation is translated later into a Pd external to simulate the system's realtime performance.

4.3.1 Matlab Simulations

Vast amount of studies in adaptive filter literature are solely focusing on speech signals. For remote collaborative recording sessions, target signals are mostly musical instruments, which have very different signal contents than speech. Even for the case of wide-band speech, it is still band-limited when compared to musical instruments whose family can almost occupy the whole audible frequency range. Musical instruments also don't possess consistent characteristics that most speech signals exhibit. To study the behavior of multi-band adaptive equalizer on various signal contents, several audio clips with different characteristics are chosen as input signals:

- pop music with guitar, vocal and drums (9')

- baritone chant phrase (11')
- soprano chant phrase (10')
- four-part choir phrase (10')
- slow F major arpeggio on harpsichord (6')
- French horn phrase (12')
- trumpet solo excerpt (*Rustiques* by Eugène Bozza, 18')
- Jazz piano excerpt (*All the things you are* by McCoy Tyner, 46')
- contemporary cello piece excerpt (*Focus a beam, emptied of thinking, outward ...* by Roger Reynolds, 20')
- contemporary violin piece excerpt (*Kokoro* by Roger Reynolds, 32')

All these audio clips are played through one Dynaudio BM6A monitor speaker. An omnidirectional dynamic microphone located 10 cm on-axis of the speaker is used to capture the output. To study the adaptive equalizer in non-realtime, system latency needs to be eliminated. In order to do this, each microphone-captured audio clip is manually trimmed, so the delay between original and captured waveforms are negligible. The microphone-captured signals are used as desired signals for the adaptive equalizer output to match, and the adaptive processes are simulated in Matlab. (See Appendix A.)

Influence of Stepsize

Signals are separated into 32 bands with FFT filter banks. The number 32 is chosen, just to follow a common subband number in the front-end of most codecs, such as MP3 and AAC. A good frequency resolution is provided by using

32 bands, and in each band we can still maintain reasonable length of FIR filter without drastically increase the computation burden. In each subband, there is a 15-tap FIR filter where 5 of the taps works with the 5-sample artificial delay inserted into the acoustic path for optimum stepsize calculation. The effective taps for adaptive filters in each band is 10. Stepsize control follows what has been described in 2.3.2.

The fixed stepsize in stepsize control is chosen to be 2.0. In Figure 4.4 and Figure 4.5, changing stepsize values are plotted together with input signal spectrograms. As can be observed, stepsize gray-scale map follows that of the spectrogram in general. This means in regions of weak signal energy, the stepsize control tend to suppress large filter updates, and in more active regions, the filter updates with a maximum of the fixed stepsize. This matches the design goal that filter should not be updating if the only information available is mostly noise.

To demonstrate the influence of stepsize on mean-square error, error energy is plotted for each subband (only lower 16 bands plotted, since band 18 to 32 are symmetrical to band 2 to 16.) From Figure 4.6 to Figure 4.9, with normalization, error magnitude is expressed as percentage of the Euclidean norm of input signal. As examples, plots based on the solo cello audio clip (*Focus a beam, emptied of thinking, outward ...*) are presented here. For other contents, single instruments in general behaves better than percussion or multi-pitch instruments such as the piano.

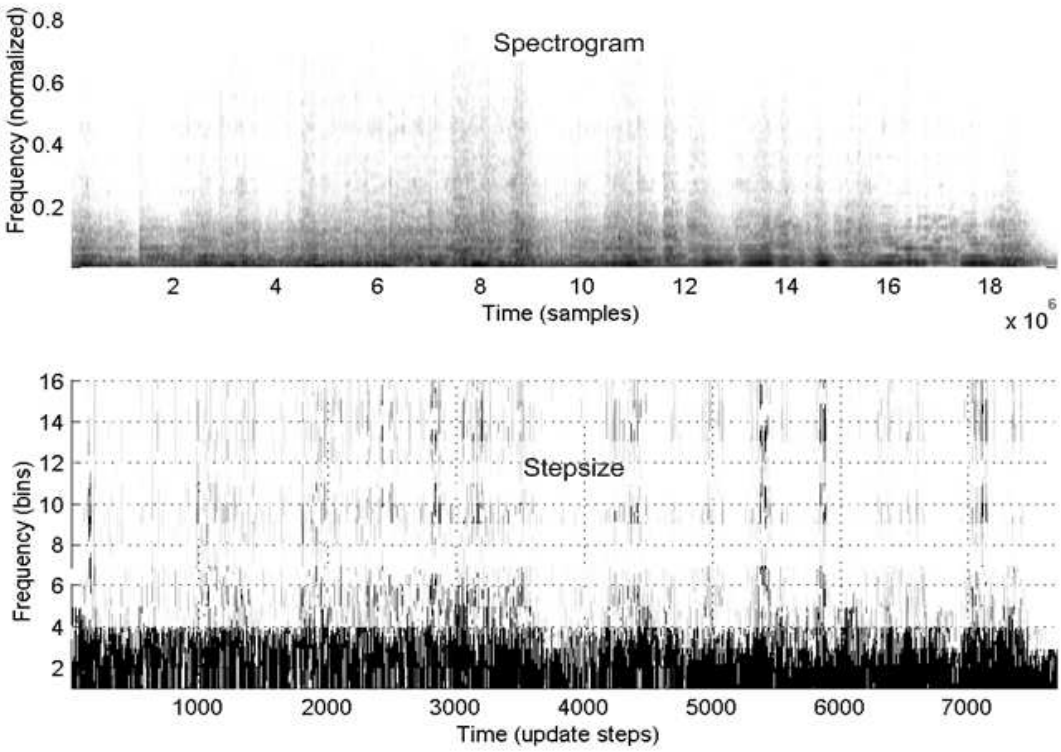


Figure 4.4: Plots of spectrogram and changing stepsize in 32-band adaptive equalizer based on a cello solo excerpt from *Focus a beam, emptied of thinking, outward ...* by Roger Reynolds. The simulation time is 20 seconds. In the lower gray-scale plots, darker portions correspond to larger stepsizes.

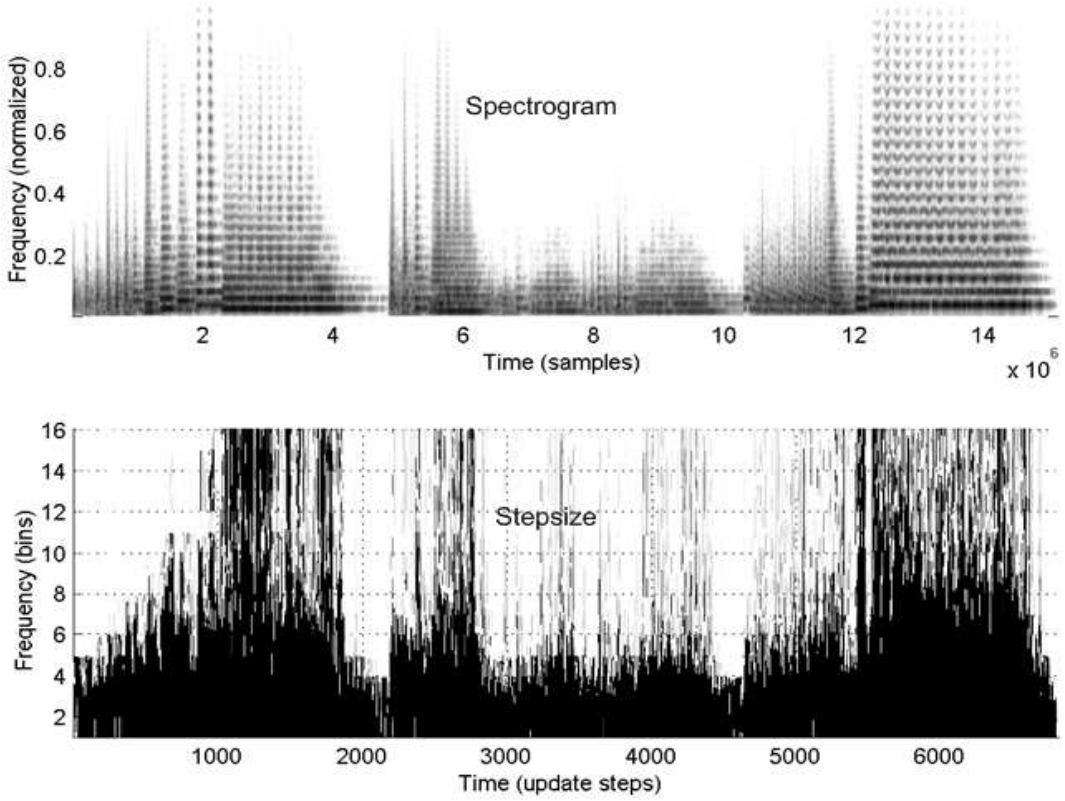


Figure 4.5: Plots of spectrogram and changing stepsize in 32-band adaptive equalizer based on a trumpet solo excerpt from *Rustiques* by Eugène Bozza. The simulation time is 18 seconds. In the lower gray-scale plots, darker portions correspond to larger stepsizes.

Figure 4.6 is the case with the stepsize control method described in 2.3.2. As expected, error energy is seen to be decreasing on a large scale. In band 1 to 4, strong energy in signal content updates the filter quickly from the beginning, and causes error energy to decrease faster. In band 6 to 16, gradual decrease of error energy at the first few seconds can be observed. In these high bands, the input signal has lower energy, allowing the stepsize control to sacrifice adaptation speed for better filter stability. In this audio clip, the cello starts with lower register phrases. In the first 5 seconds, occasional bow noises and other high frequency transients may be explained as reasons for some error peaks shown in band 7, and bands 9 through 15. After that, the average energy keeps decreasing. Around 8 seconds into the audio clip, two prominent peaks are shown in band 3 and 4 respectively. This is the time that the cello plays higher register notes for the first time. These contents reveal more errors in corresponding bands and cause the filter to make new adaptations. After this, more new note attacks can be seen in the spectrogram (Figure 4.4) but not any more peaks in error energy like the ones at 8 second. It can be speculated that the filter has already updated coefficients for those frequency regions and reduced the first peaks, thus the later filter outputs actually match the reference signal well.

Examples of subband NLMS using constant stepsizes are plotted in Figure 4.7 and Figure 4.8. Figure 4.7 shows a case with constant stepsize 0.01. In this case, a gradual decreasing envelop of error energy can be observed in each subband.

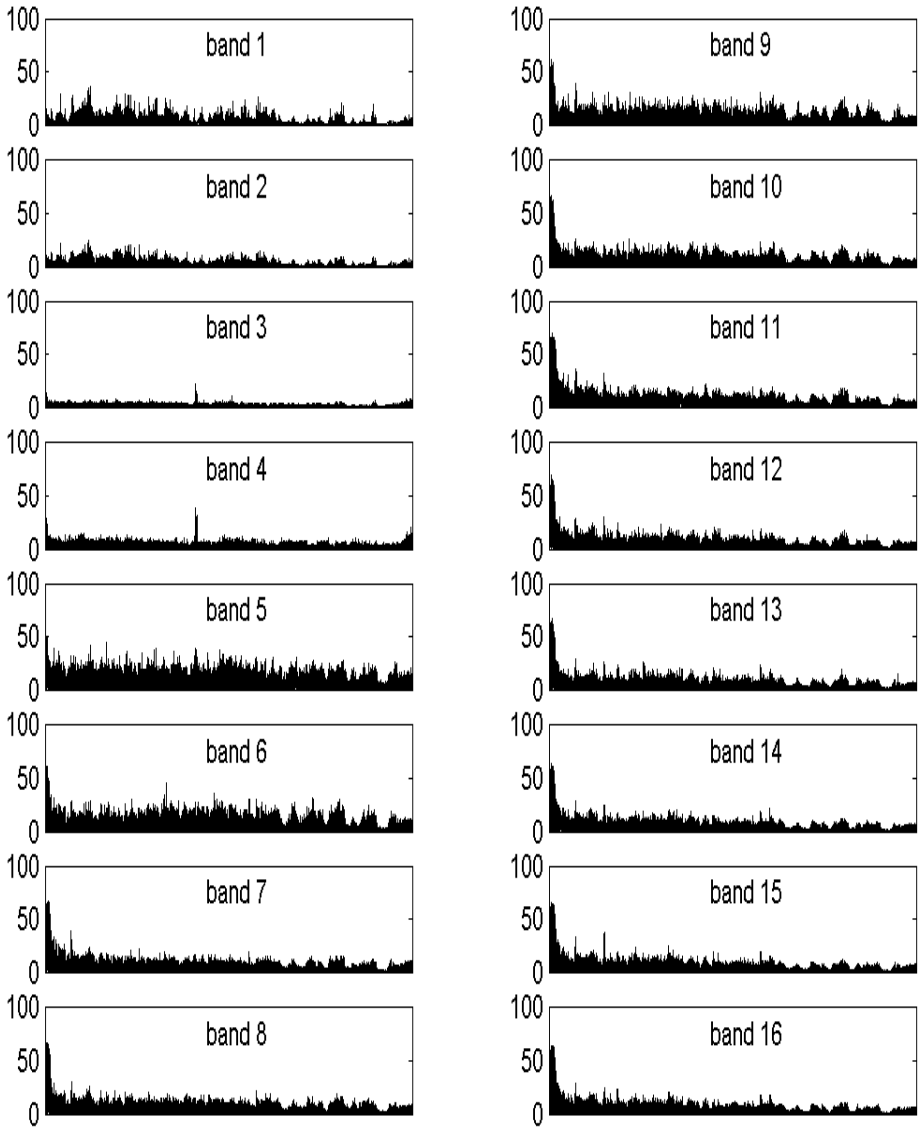


Figure 4.6: Error energy plots over time for a 32-band NLMS adaptive equalizer with stepsize control (varying stepsize). In each subplot, the x-axis is time and y-axis is error magnitude as a percentage of the Euclidean norm of the input signal. The input signal is a 20 second excerpt of a contemporary cello solo piece.

Filter in this case should be very stable, but not efficient enough. Figure 4.7 shows a case with constant stepsize 2.5. In this case, each subband updates very quickly, and the decreasing envelop in error energy cannot be seen at the beginning. This fast adaptation comes with the risk of filter deviation, as error energy in band 2 starts to grow out of control near the end of the simulation. At the same time, these behaviors are very content-dependent. For some other contents, filter deviation can be observed for constant stepsize values less than but very close to 2.0 – the theoretical maximum "safe" stepsize without stepsize control. In general, compared to constant stepsize, adaptive filters with stepsize control should have a good balance between high adaptation efficiency and high stability.

With the current version of stepsize control, additional filter taps have to be added to handle the artificial delay. In some situations, computational cost due to stepsize control may be undesirable. To examine the difference in error energy for designs with stepsize control and those with constant but reasonable stepsize, the error energies between these two cases are subtracted and plotted in Figure 4.9. In this figure, y-axis values are calculated by subtracting error energy of varying stepsize from that of constant stepsize 1.0. It can be seen that, in the beginning, a good choice of constant stepsize can adapt the filter quicker, and later on the error differences are within reasonable fluctuation range. This is true for some other signal contents as well. Thus, when computational power is limited, a reasonable constant stepsize for this system without stepsize control is still acceptable.

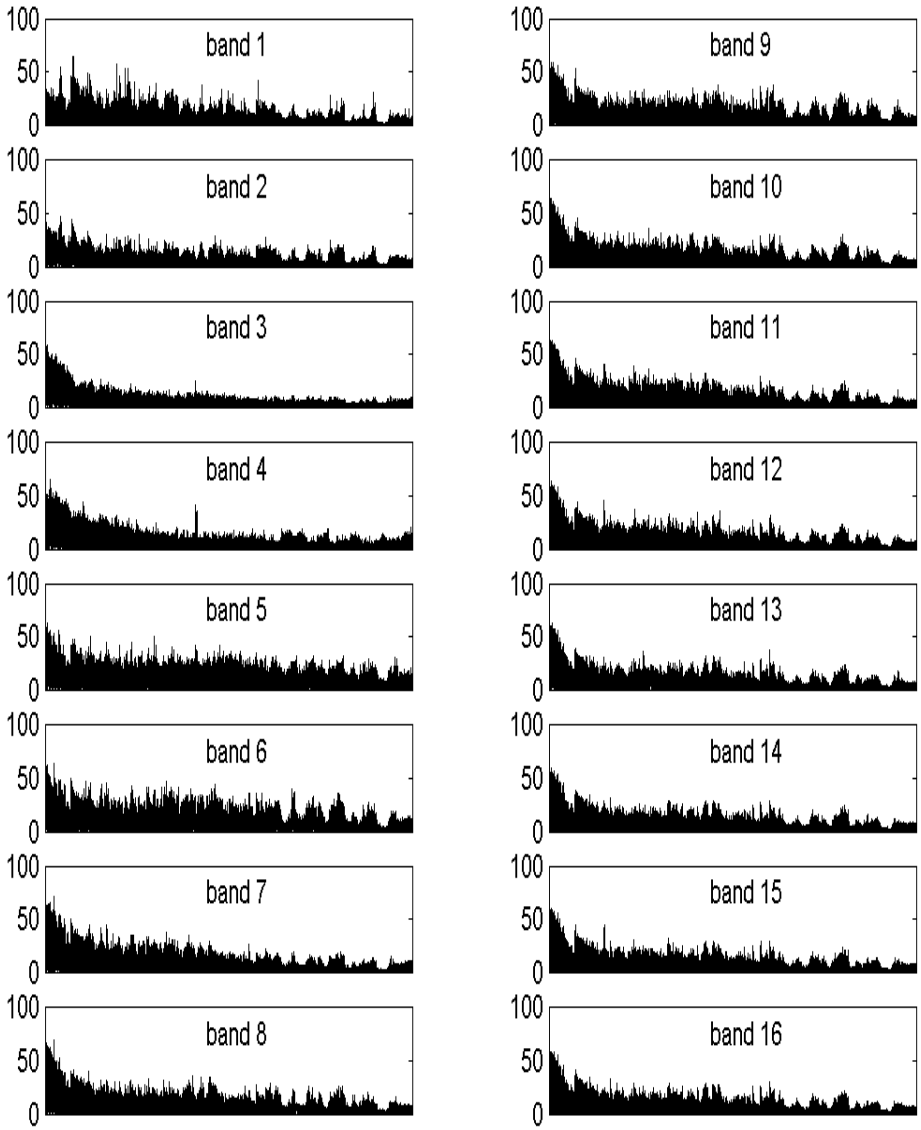


Figure 4.7: Error energy plots over time for a 32-band NLMS adaptive equalizer with constant stepsize 0.01. In each subplot, the x-axis is time and y-axis is error magnitude as a percentage of the Euclidean norm of the input signal. The input signal is a 20 second excerpt of a contemporary cello solo piece.

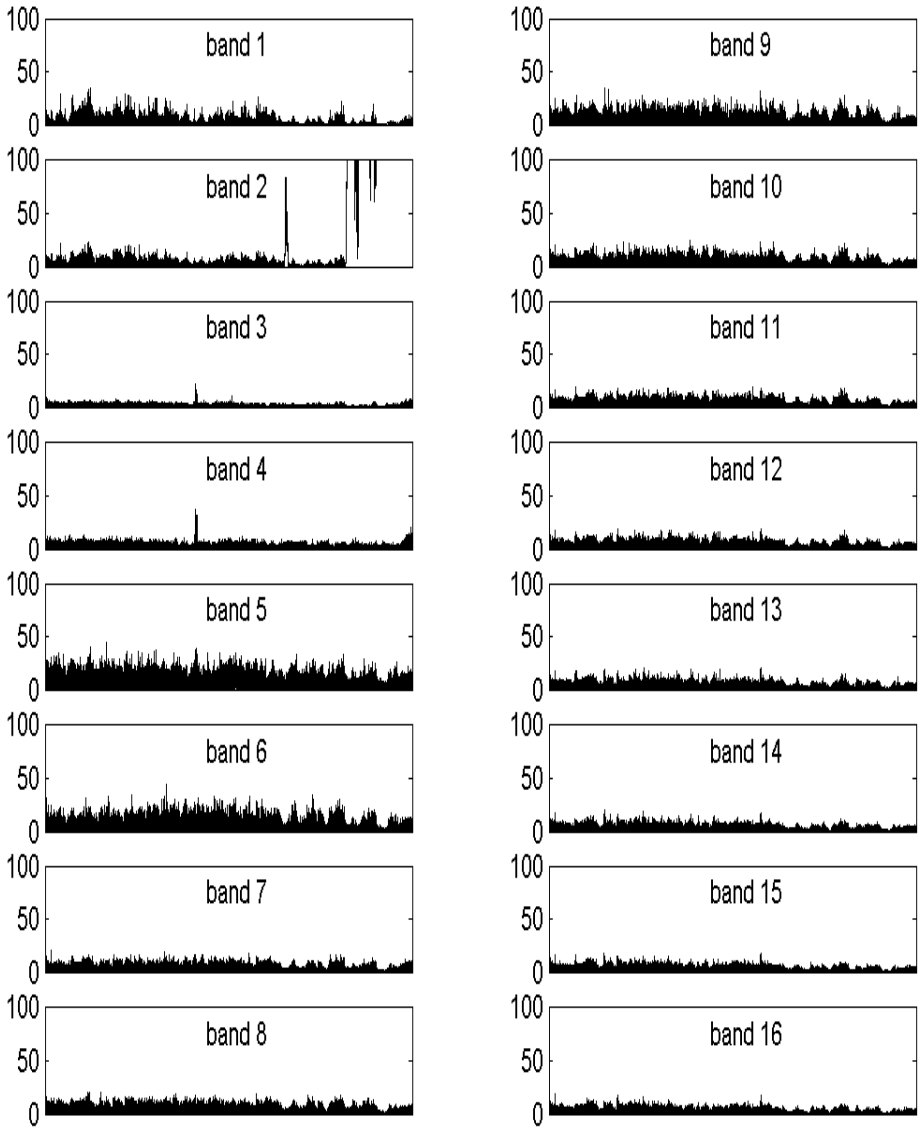


Figure 4.8: Error energy plots over time for a 32-band NLMS adaptive equalizer with constant stepsize 2.5. In each subplot, the x-axis is time and y-axis is error magnitude as a percentage of the Euclidean norm of the input signal. The input signal is a 20 second excerpt of a contemporary cello solo piece.

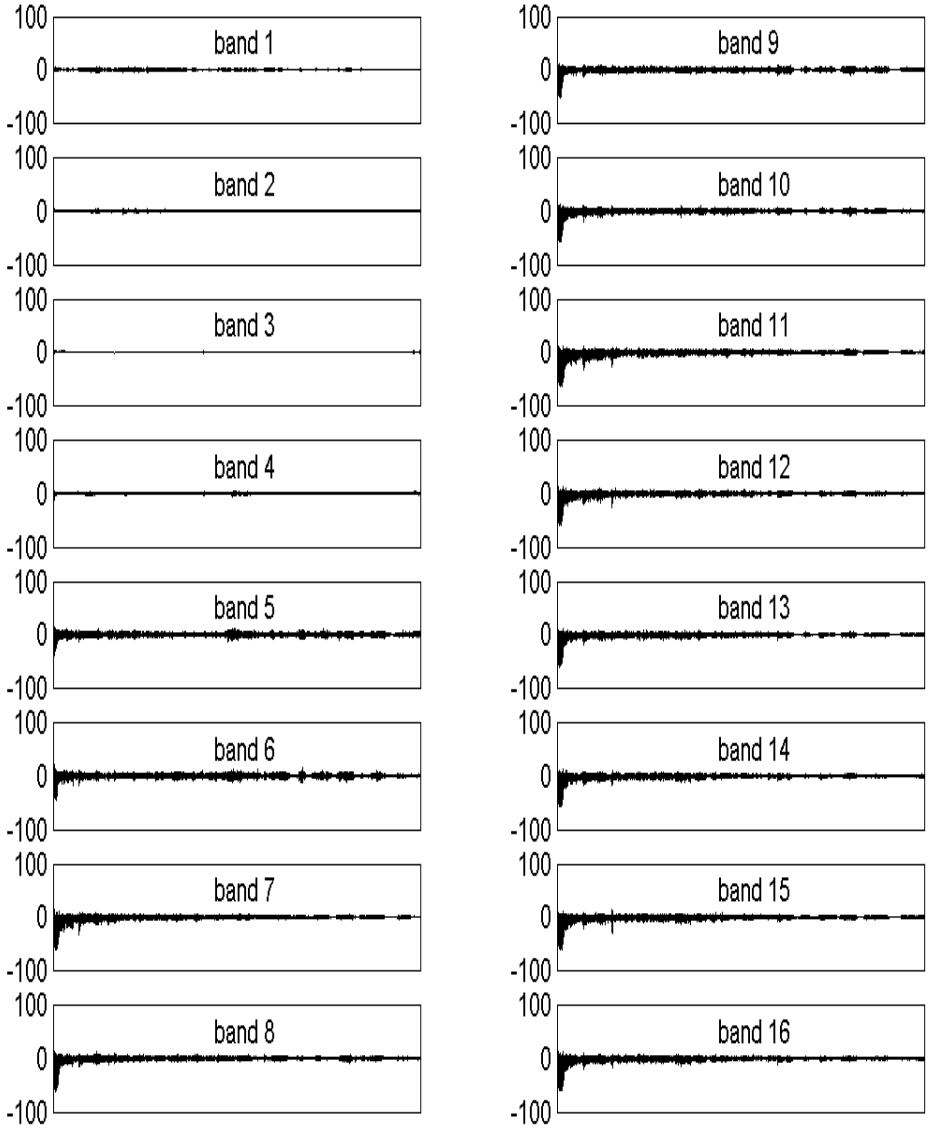


Figure 4.9: Error energy differences between varying stepsize and constant stepsize 1.0 for a 32-band NLMS adaptive equalizer. In each subplot, the x-axis is time and y-axis is error magnitude difference (constant stepsize 1.0 - varying stepsize) as a percentage of the Euclidean norm of the input signal. The input signal is a 20 second excerpt of a contemporary cello solo piece.

Influence of Filter Order

Since the equalizer is realized with a linear adaptive filter with complex coefficients in each band, its performance is greatly influenced by different filter orders.

To study the relationship between error energy and filter order, simulations are performed on several contents for the effective filter lengths of 1 (zero order), 3, 5, 10, 15, 20, 25 and 30. Around 10 seconds after the first adaptation, the filter coefficients are assumed to have reached their best performance on minimizing square errors. From this point, the error magnitudes (as a percentage of input signal Euclidean norm) are averaged over 15000 successive subband samples. By changing effective filter lengths (number of filter taps other than the artificial delay taps for stepsize calculation), the averaged error magnitude values are studied. Some results based on the cello solo audio clip are listed in Table 4.1 and plotted in Figure 4.10. It is obvious that error magnitude as a percentage of Euclidean norm of input signal decreases monotonically with increasing effective filter lengths. The most steep decreases are observed between filters of effective length 1 and 15. For different bands, the decrease curves are also slightly different.

This suggests that effective filter lengths from 10 to 20 should be sufficient to serve the purpose of reducing mean square error. With higher filter orders, the error performance is not improved very noticeably thereafter, but the computational cost grows significantly. Also, for different frequency bands, different filter

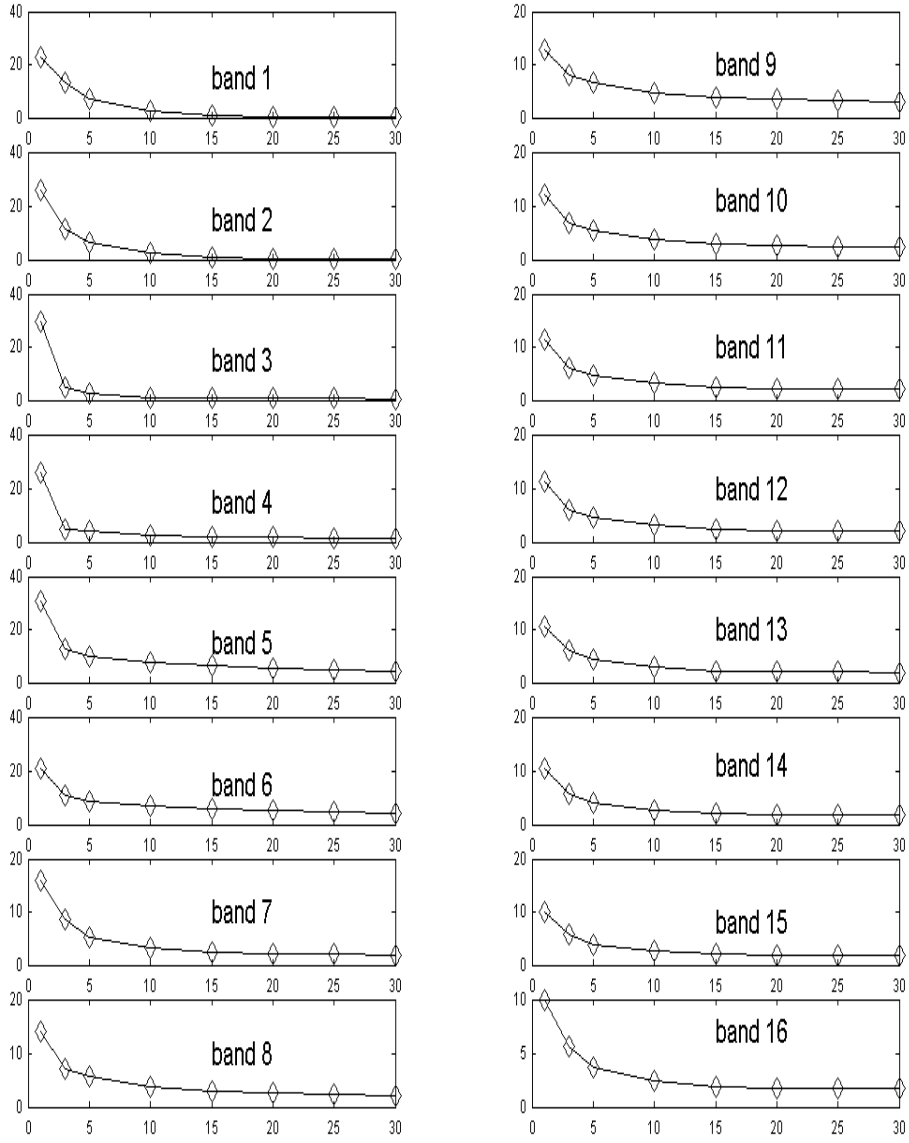


Figure 4.10: A study on the influence of filter order on the minimum error energy that can be achieved in each subband. The error energy is expressed in error magnitude as a percentage of the Euclidean norm of the input signal. The average energy calculated over 15000 subband samples around 10 seconds into the audio clip of a contemporary cello solo piece. Effective filter lengths of 1, 3, 5, 10, 15, 20, 25, 30 are simulated. The values for each data point are listed in Table 4.1.

Table 4.1: Influence of subband FIR filter effective length on minimum error energy after adaptation. Error energy is expressed in error magnitude as a percentage of Euclidean norm of the input signal.

Bands	Effective filter lengths							
	1	**3**	**5**	**10**	**15**	**20**	**25**	**30**
1	22.8315	13.5161	7.1955	2.5963	0.8348	0.4018	0.3387	0.3086
2	25.6897	11.419	6.4143	2.2931	0.8444	0.5148	0.4842	0.5121
3	29.4816	4.9841	2.5509	1.1195	0.8155	0.6596	0.5923	0.552
4	25.9181	4.9368	3.8905	2.3381	1.9755	1.688	1.4719	1.3197
5	30.8508	12.7919	9.7558	7.3741	6.2329	5.3769	4.649	4.2533
6	20.9522	10.6249	8.7309	6.7871	5.916	5.2741	4.525	4.1258
7	15.8254	8.5309	5.1206	3.3075	2.5006	2.1518	2.0132	1.9258
8	13.9374	7.1456	5.5965	3.749	2.9992	2.5852	2.3336	2.1825
9	12.6914	8.1537	6.5629	4.6708	3.9328	3.505	3.1505	2.8954
10	12.0646	6.8589	5.4645	3.6805	3.0018	2.6728	2.4565	2.3015
11	11.39	6.0128	4.7157	3.1966	2.467	2.1889	2.036	2.0034
12	11.1856	6.0288	4.5142	3.0852	2.364	2.1086	1.99	1.9529
13	10.5716	6.134	4.3558	2.8729	2.2457	2.0599	1.976	1.9205
14	10.3409	5.8143	4.0705	2.6525	2.0446	1.8671	1.8215	1.8175
15	10.0531	5.6343	3.7807	2.5496	1.9959	1.8114	1.7799	1.7636
16	9.9641	5.6116	3.7393	2.4396	1.858	1.7068	1.6915	1.7048

order maybe chosen. This could reduce unnecessary computations in bands where lower filter orders are sufficient. Further investigations regarding this matter are yet to be conducted.

4.3.2 Pd Simulations

The system that models the room with a long delay line and an adaptive equalizer is integrated and simulated in real time with Pd. The delay estimation is done with the Pd external *stimer~* as described before (4.2.1). In order to implement the adaptive equalizer, a Pd external named *sbnlms~* is developed based on the Matlab code from Appendix A. *sbnlms~* is basically a subband NLMS adaptive filter that is initialized to be an all-pass filter by default. Its tunable and control parameters include subband numbers, filter order, artificial delay samples (for optimum stepsize calculation), fixed stepsize, filter update blocksize, adaptation enable/disable, optimum stepsize calculation enable/disable. The simulations are performed on the audio clips mentioned in 4.3.1 with 32-band configurations. The filters in each band have an effective length of 10 and the artificial delay is 5 samples. Optimum stepsize calculation and adaptation are enabled through the whole process.

Simulations on Synthetic Transfer Functions

The realtime simulation is first performed with synthetic transfer functions within Pd. Different simple linear filter effects, such as voltage-controlled

band-pass filters, are used to generate the reference signal. Since this setup matches the assumption and filter model almost perfectly, the system behaves very robustly and generates output signals that match the reference exactly. Sound quality and update speed can be easily told from the audible results, so that no detail analyses are performed.

Simulations with Room Acoustics

To study the behavior of the system in a room, the same loudspeaker-microphone setups are constructed. In general, the results also show good approximations of the reference signals from the system outputs. To illustrate some results, different spectrograms are plotted for two of the audio clips. In Figure 4.11, spectrograms regarding the trumpet audio clip are shown. Compare Figure 4.11(a) and Figure 4.11(b), one easily visible difference is that very high frequency portions of the input signal disappear in the microphone-captured reference signal. This is largely due to the frequency response of the microphone in the experiment. To examine how well the spectrogram of system output matches that of the reference signal, their spectrograms are subtracted and plotted. From Figure 4.11(c), it can be seen that most energy cells on the time-frequency map are matched, so that large portion of the spectrogram difference plot are in shallow colors on the gray scale. In high frequency regions, some residuals still exist. This could partly result from insufficient filter order (limited by the computation power of the PC that this simulation is performed on), but largely due to that the real system (room

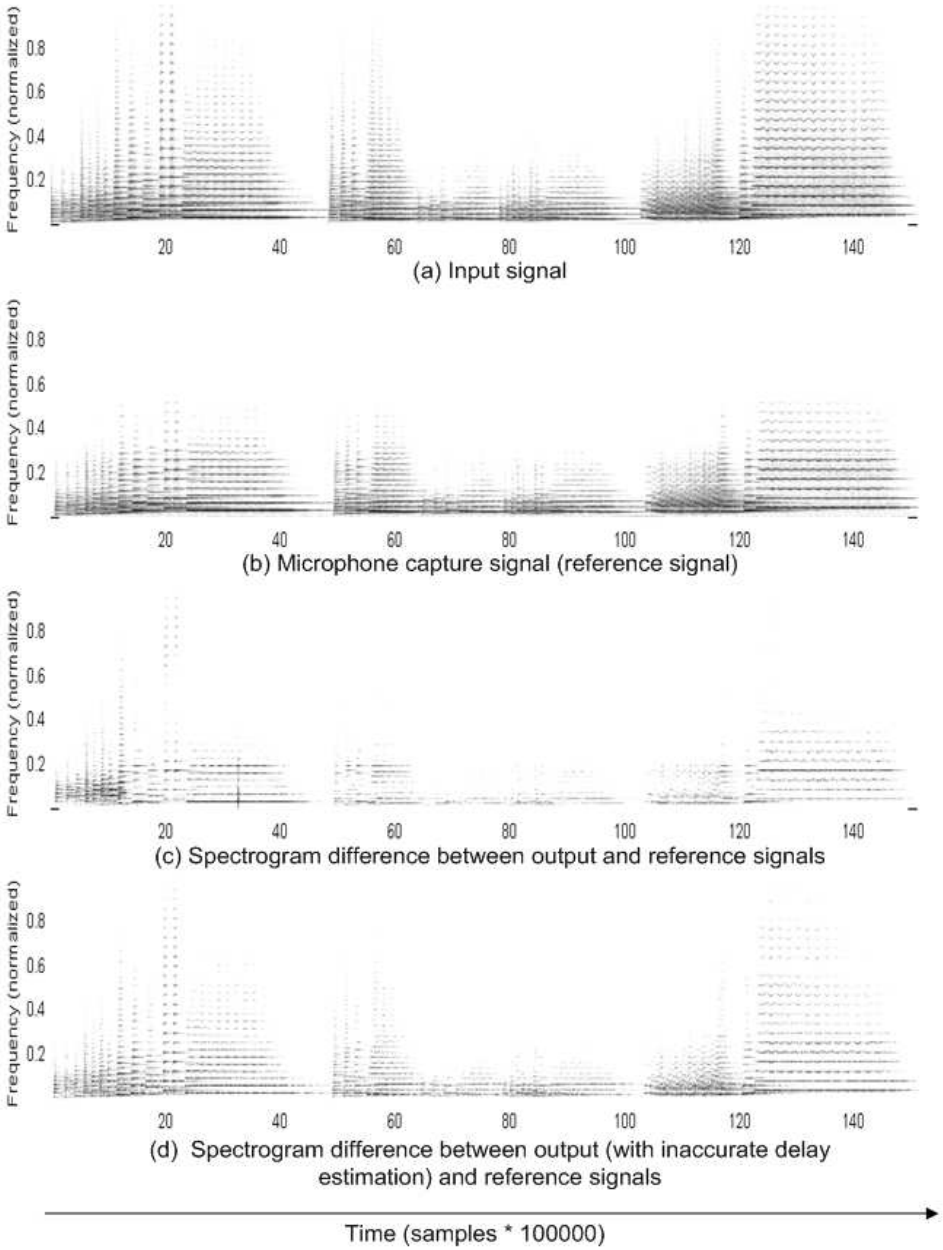


Figure 4.11: Spectrograms from Pd simulation on adaptive equalizer identification after a constant delay line. (a) Input signal; (b) microphone capture signal (reference signal); (c) spectrogram difference between the output signal from equalizer and the reference signal; (d) spectrogram difference between the output signal from equalizer and the reference signal, but the delay estimation in this case is inaccurate. The signal content here is a trumpet solo phrase from a contemporary piece. The simulation time is 18.5 seconds.

acoustics) cannot be modeled solely using linear FIR filters. Nevertheless, comparing only the portion that overlaps with microphone signal energy, the match is considered good. The residuals in high frequency portion of the output signal will result in some additional attenuations in those regions during echo cancellation, and this is not a very destructive effect sonically.

Once again, it is observed that the estimation for the constant long delay in the system (see Figure 2.4) is very crucial. During the simulation on Windows XP, some system operations such as file read/write might cause the overall system latency to fluctuate. To guarantee an accurate estimation of the delay line, two delay measurements are taken before and after a simulation. Only when the delay doesn't change between the two measurements, the simulation results are recorded. Just as an example, Figure 4.11(d) shows the spectrogram difference same as Figure 4.11(c), but the delay estimation is off by around 10 milliseconds. This inaccuracy causes the equalizer to update based on "wrong" reference signal segments. The darker regions in the plot indicate a less desirable match.

The audible results also shows a subjective impression that the algorithm favors single-note instruments over multi-pitch instruments in the simulations. For signal contents of cello, violin, trumpet and other solo instruments, the output sounds all resemble the reference signals. For multi-pitch instruments like the piano, and mixture signal like the pop song, artifacts can be heard when the filter coefficients are adjusted for optimum performance. However, this doesn't compro-

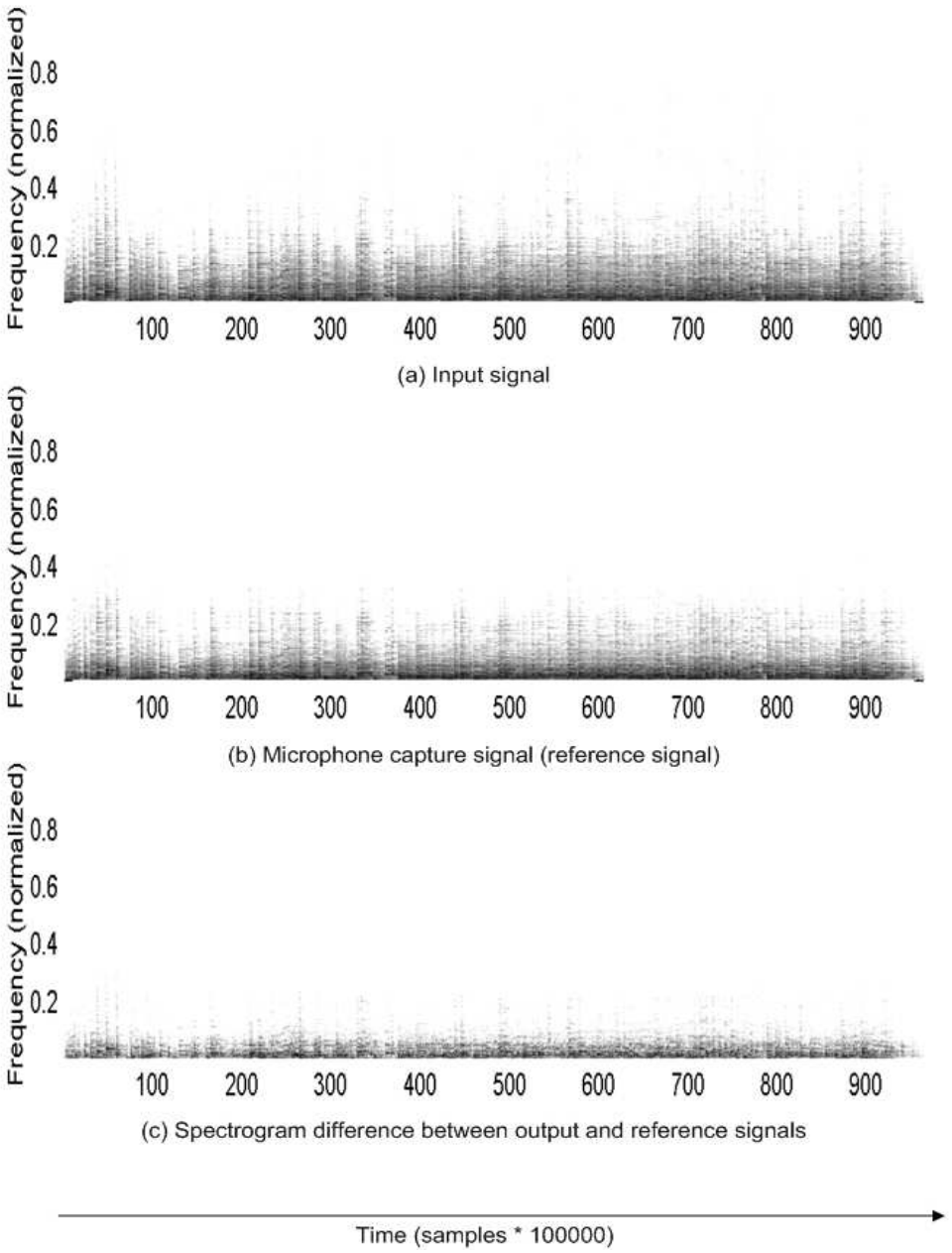


Figure 4.12: Spectrograms from Pd simulation on adaptive equalizer identification after a constant delay line. (a) Input signal; (b) microphone capture signal (reference signal); (c) spectrogram difference between the output signal from equalizer and the reference signal; The signal content here is a jazz piano excerpt. The simulation time is 46.7 seconds.

mise much the quality in matching the spectrogram. The resulting spectrograms based on the piano clip are shown in Figure 4.12. In Figure 4.12(c), it can be seen that despite the audible artifacts, most energy are matched with the reference signal. What's more, the output signal is not used directly, but only used to provide energy mask for spectral subtraction, thus the artifacts for multi-pitch instruments are considered negligible.

Integration with Spectral Subtraction

The delay and equalization portion of the Pd simulations are later integrated with the spectral subtraction processing block to improve the echo cancellation. The results are robust as expected. Since the subtle improvements involves further subjective evaluations, detail analysis are not conducted.

4.4 Signal Separation Simulations

Since BSS doesn't have realtime requirements, sophisticated algorithms can be implemented. Some aspects of BSS are studied in Matlab both for the time-frequency masking approach (with W-DO assumption) and the spatial acoustics estimation approach.

4.4.1 W-DO Measurements

It is previously known (see 3.2) that speech signals show very good W-DO in general. To test the W-DO for musical instruments, some simple measurements are conducted. Four pairs of musical instrument audio clips are mixed, and the preserved-energy ratios (PSR) are calculated for each pair, assuming one of the signals is the wanted source signal. FFT size of 32 and 512 are used to compare the effect of different time-frequency resolutions. The data results are listed in Table 4.2 and plotted in Figure 4.13 and Figure 4.14. In this experiment, all

Table 4.2: Preserved-energy ratio (%) measurements for four musical instrument pairs (detail signal contents explained in Figure 4.13 and Figure 4.14 captions).

Threshold	0 dB	5 dB	10 dB	15 dB	20 dB	25 dB	30 dB
FFT size: 32							
Pair 1	83.1077	65.5029	44.3877	25.6428	13.0086	6.2405	3.3067
Pair 2	80.1371	66.8662	52.3391	36.7138	23.6142	13.9619	8.1554
Pair 3	90.0376	72.7072	51.2449	33.12	20.0691	11.8495	6.9382
Pair 4	71.0919	52.9262	36.4262	23.5201	14.0935	7.3958	3.5342
FFT size: 512							
Pair 1	92.0476	82.2648	64.6066	46.7734	32.2026	21.445	12.1274
Pair 2	93.661	90.0991	84.349	77.6768	70.6567	64.0633	58.5196
Pair 3	93.6041	83.8211	67.5275	48.1804	31.4004	20.7639	12.6556
Pair 4	87.7786	78.5642	66.374	52.9956	39.1451	28.0543	17.6024

measurements are based on two-source audio mixtures. Generally speaking, for each pair, as the dB threshold increases, the PSR decreases rapidly. Compare the four curves in Figure 4.13 to the two-source curve ($N = 2$) in Figure 3.3 for speech, the PSRs for musical instruments show much more rapid decrease. To have 50% of the original signal energy preserved, the signal-to-interference ratio in each

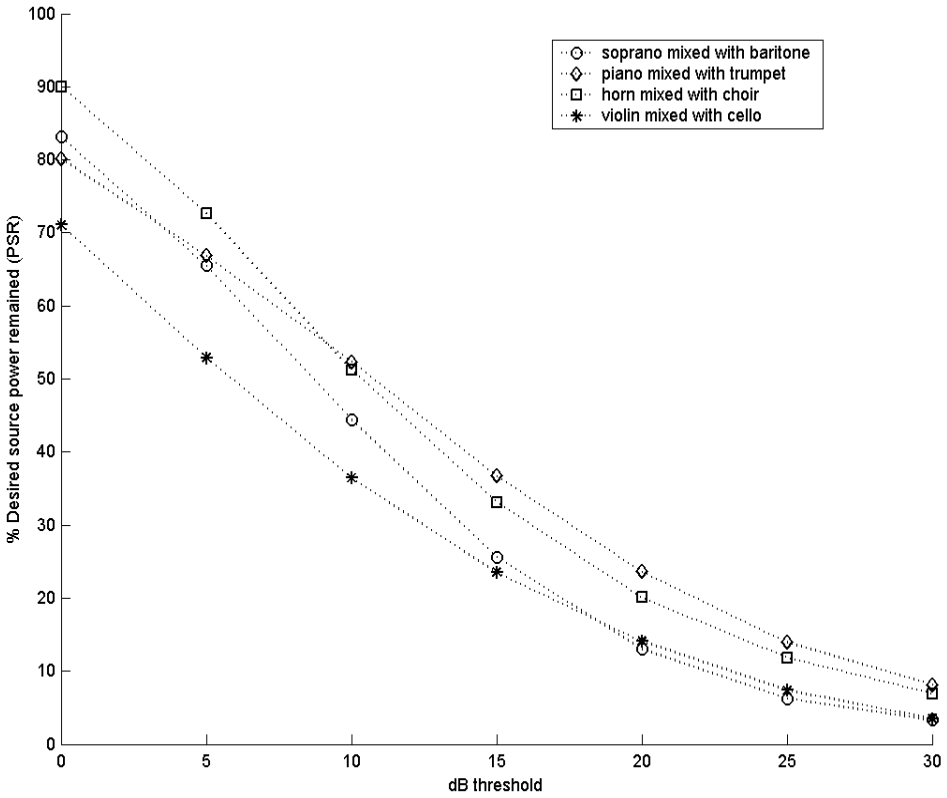


Figure 4.13: Preserved-energy ratio (PSR) measurements for four pairs of musical instrument audio clips: 1. soprano mixed with baritone; 2. piano mixed with trumpet; 3. horn mixed with choir; 4. violin mixed with cello. The FFT size is 32. The audio clips are selected from the same collection mentioned in 4.3.1.

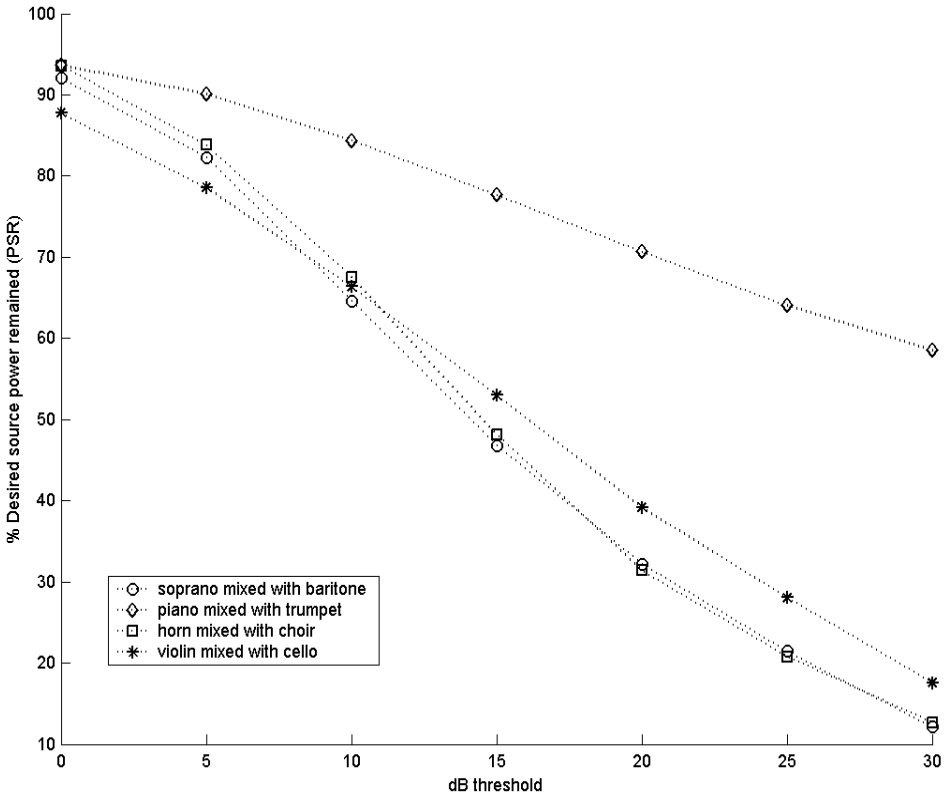


Figure 4.14: Preserved-energy ratio (PSR) measurements for four pairs of musical instrument audio clips: 1. soprano mixed with baritone; 2. piano mixed with trumpet; 3. horn mixed with choir; 4. violin mixed with cello. The FFT size is 512. The audio clips are selected from the same collection mentioned in 4.3.1.

time-frequency cell of musical instrument signals has to be roughly less than 10 dB. For speech, even 30 dB minimum signal-to-interference ratio can still preserve 50% of the original signal energy. When the FFT size increases to have much more frequency resolution and less time resolution, Figure 4.14 shows a better W-DO for the same two-source mixtures. The overall values of PSRs are higher, and they decrease less rapidly as compared to the FFT size of 32. Particularly, one pair (to separate piano from a mixture of piano and trumpet) shows very good W-DO in preserved energy after masking.

The variety of possible signal contents in a remote recording session remains large. In the future, it may be meaningful to do more refined studies on the W-DO for different signal combinations. The current experiment suggests that, compared to speech signals, concurrent musical instruments might not generally be good candidates to apply the W-DO properties and to be processed based on this assumption. But for some particular instrument combinations, W-DO and the time-frequency masking approach might still be a good choice for signal separation. An investigation of the audio content is needed beforehand. This experiment also suggests that, larger FFT size may be good for melodic musical instruments in most situations. In frequency, The resulting fine resolution will help separating different harmonics of instruments into separate bands. In time, since musical instruments usually don't have as rapid temporal changes as speech, the reduced resolution can usually be tolerated.

4.4.2 BSS with Spatial Acoustics Estimation

The signal removal approach based on spatial acoustics estimation is simulated on both synthetic signals and realistic cases. Matlab code for the simulation is provided in Appendix B.

For a mixture of synthetic signals, transfer functions are arbitrarily designed to have certain degree of complexity. Since the model matches the assumption exactly, perfect source removal is always achieved. In the realistic experiments, the stereo signal mixtures are several CD albums including jazz, pop, artistic songs and concerts. Solo segments of the instrument to be removed are selected manually to train the vector defined in Eq.(3.35). After the training, this vector is used to cancel the source in the stereo mixture (Eq.(3.36)). The results turn out to be satisfying and robust to additive noise in general. As an example, some spectrograms for a voice removal on an excerpt from Billie Holiday's *I'm a fool to want you* are shown in Figure 4.15. Figure 4.15(a) shows concatenated spectrograms of some solo voice segments based on which the vocal's spatial transfer functions are calculated. Figure 4.15(b) is the left channel of the voice mixed with a band where the main instruments are basses, cellos, violins, and drum sets. In the signal after voice removal shown in Figure 4.15(c), it can be seen that the harmonics of cello passages around the first half of the clip and harmonics of string ensemble around the second half of the clip are still clearly visible. Voice energy (dark harmonics that used to be in the mixture signal) is reduced significantly, and only a small part

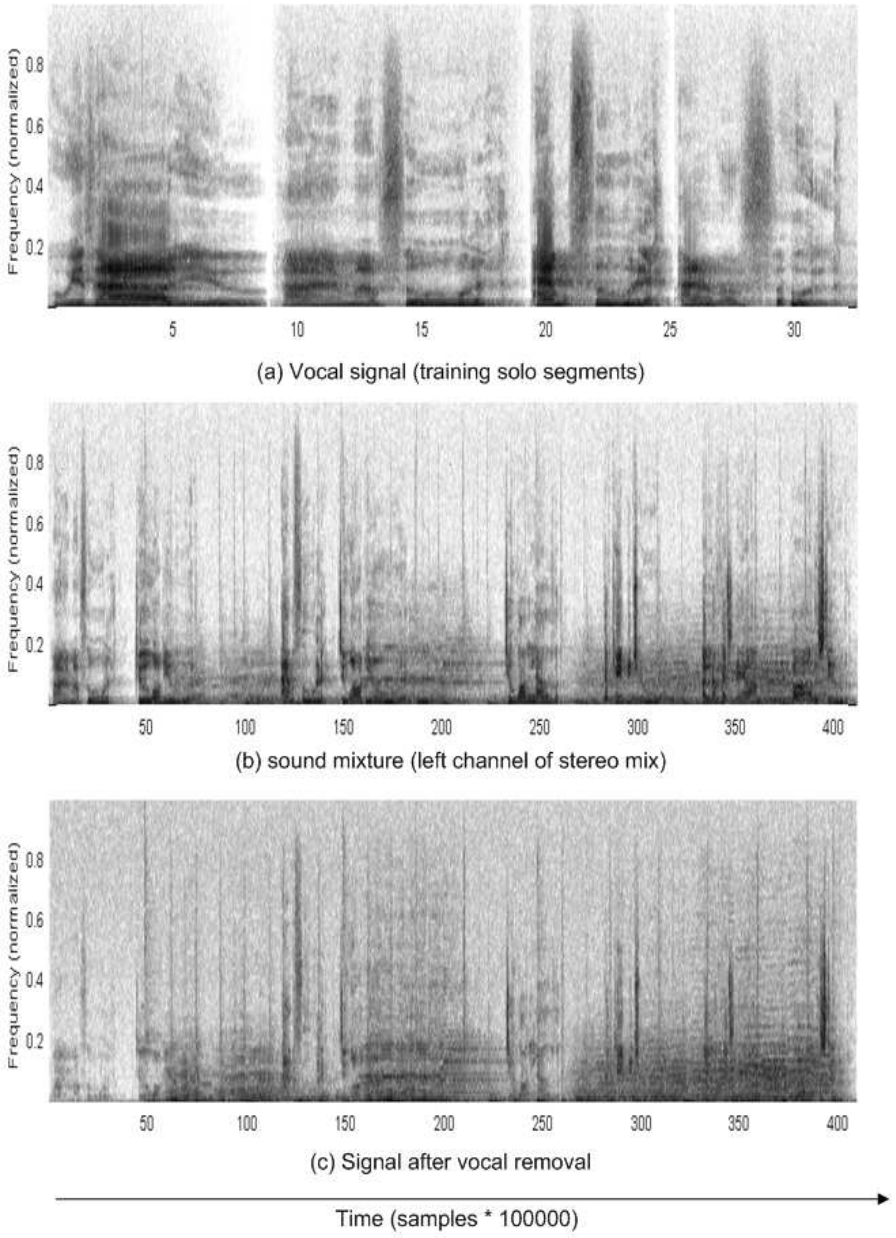


Figure 4.15: Spectrograms from signal removal based on spatial acoustics estimation. (a) Solo vocal signal; (b) stereo signal mixture (left channel); (c) mono signal after vocal removal. The signal contents are from Billie Holiday's song *I'm a fool to want you*.

of high frequency transients are left. Also in Figure 4.15(c), some rhythmic dark lines are still visible as is in Figure 4.15(b). These are hi-hat strikes, which are also wanted signal in the band. They are preserved and possibly reinforced during the voice removal process. The auditory quality is even more convincing than this visual presentation, which was confirmed with several subjective listeners.

4.4.3 Discussion

In the post processing stage of the prototype system, both the time-frequency masking and the acoustics estimation approaches can be very helpful to solve the sound source removal problem. For the time-frequency masking approach, refined delay (phase unwrap) and attenuation tunings can be performed, so that there is a lot of room to improve measurement error performances. The downside of this approach is that W-DO assumption matches the behaviors of some of the musical instruments poorly. Nevertheless, in remote collaborative recordings for speech and some particular "good candidate" musical instruments, this approach can still be used and its degenerative nature can function as one additional merit in certain conditions. For the spatial acoustics approach, the estimation of source removal vector (Eq.(3.35)) is very crucial, but there is no direct physical model, like those in the first approach, to tune the components of the vector. This imposes very critical requirements on the accuracy of acoustic measurements and physical setups. However, in the prototype system, the unwanted signal is a stable

point source (a reference loudspeaker), and almost every component (except the musician himself/herself) can be precisely setup and controlled. These advantages are all benefiting factors to favor this approach. In general, the spatial acoustics estimation approach is suggested as the main algorithm for the post processing stage of the prototype system.

As mentioned before, one additional benefit from the signal removal algorithms is that, a scaled-down version of them (fast computation) can be used to enhance the double-talk detector quality for echo cancellation process. With the reference speaker signal crudely removed from a stereo signal mixture, the resulting mono signal can represent mainly the near-end musician's energy, thus reports double-talk events with ease.

5

Conclusion

5.1 Summary

To sum it up, this dissertation investigates two major problems in making a remote collaborative recording - echo cancellation during the recording, and signal separation during post processing. A prototype system is conceived and simulated to demonstrate the validity of the design in separated steps.

At one site of a remote collaborative recording session, the system consists of two identical microphones as signal capturing devices, one reference speaker as the signal output device, and a computer as the signal processing and recording device. Musicians at different sites listen to other sites through their reference speakers, and recordings are done at each site with their local recording devices. Two microphone signals are going to be captured for the recording. To eliminate echoes traveling from reference loudspeakers to their local microphones, a smart system integration design is proposed, which takes advantage of the spectral subtraction method and subband NLMS adaptive filters to achieve a robust and

stable performance. Either one of the microphone signal mixtures can be used and processed to generate an echo-free transmission signal. BSS with reduced quality but high efficiency can be utilized to help detecting the local musicians' activities in real time, and control the adaptation of simulated echo path from speakers to adjacent microphones. In post processing, from the same two-microphone signal mixtures, high quality BSS with either the time-frequency masking approach or the spatial acoustics estimation approach can be used to perform non-realtime processing and remove reference speaker signals from the mixtures, and obtain a clean recording of the local musicians for further mixing and rendering.

Some characteristics of such remote collaborative recording sessions turn out to be beneficial to the signal processing models. Since the location is usually a recording studio, placement of microphones, speakers and even musicians can be fully controlled and configured to best match the mathematical models. Sometimes, the room characteristics can also be configurable. Microphone spacing and various dimensions can be measured ahead of time to fine-tune some algorithms, so that the source separation is not fully "blind". In echo cancellation, the echo path doesn't need to be estimated blindly, and there is usually plenty of time before a recording session to send test signals and calibrate the system. In post processing, the signal to be removed is from a fixed loudspeaker, thus the errors due to sound source movement in signal removal process is successfully avoided.

This prototype system, for the first time, smartly integrated the method

of spectral subtraction and adaptive filter in a new way to achieve echo cancellation. Spectral subtraction is used as the major component of the echo cancellation system, and an adaptive filter is used as the auxiliary component. The nature of spectral subtraction makes the system stable and robust to room acoustics fluctuations. It also allows a relaxed adaptation of the adaptive filter to fine-adjust and catch changes in acoustic transfer functions. Drawbacks of conventional adaptive filters still exist, but appear to be insignificant within this system. This part of the prototype system is simulated and tested in non-real time with Matlab and real time with Pd, whose results corroborate the benefits brought by the novel system design.

The post processing portion of this prototype system is an original BSS design based on spatial acoustics estimation. Realistic music signals, such as CD records, are used as source materials and simulated with Matlab for the source separation. The audio results, for some good candidate inputs, work surprisingly well, so that the unwanted sound source in a sound mixture is mostly silent after processing, leaving only a small portion of high frequency components. Another BSS approach is also looked at, which assumes the W-DO property of input signal mixtures, and demixes the mixtures with the time-frequency masking method. Music signals, which are rarely studied by the speech community, is explored in this dissertation for the their eligibility to this approach. The Matlab simulation results show that only a few musical instrument signal mixtures tend to favor the

W-DO assumption, so it is in general unrealistic to apply this approach to music signals.

When fully integrated, although still primitive, this system proves its feasibility and suggests a very promising development path toward realistic solutions to help a remote collaborative recording.

5.2 Future Research

The current simulated system proposes a prototype solution to some problems of a remote collaborative recording. To continue investigating details of the solutions, there are many aspects to address in the future.

5.2.1 Online Processing

In echo cancellation, three major processing blocks are involved: constant delay line, adaptive equalizer and spectral subtraction. The constant delay line is estimated with sample-accurate measurement with impulses beforehand. Practically, the delay might change due to software, hardware, or acoustics. Hardware and software can be designed to dedicate to this type of application and avoid delay changes over time. This very costly approach can also be substituted by design algorithms to track delay changes over time when the system is "in session". This might be an interesting future research involving proper statistics and new delay measurement methods with common audio signals.

The adaptive equalizer is the most complicated part in this echo cancellation system. Currently, it is only simulated with 32-point FFT filterbanks, and different orders of subband FIR filters. Many tunable parameters such as the artificial delay amount, window type, window hop size, time constants for input and error signal energy estimation, can all be studied in detail about their influence on the performance of the multi-band NLMS algorithm. Since musical instruments exhibit very different time and frequency characteristics when compared to speech, finding an optimum FFT size, i.e. time-frequency resolution, will also be a very interesting future research topic. Currently, DTD is only theoretically proposed, and an efficient implementation of DTD is yet to be experimented. The influence on musician's minute body movement on the transfer function from loudspeaker to microphone, can also be studied. Many stepsize control designs, which take care of loudspeaker-microphone-enclosure change, may be applied to alleviate problems created by these movements.

In spectral subtraction algorithms, the current experiment only uses 64-point FFT and a time average of 200 ms to generate the energy mask. One optimum FFT size is yet to be found to achieve the best possible performance. Currently, the constant 200 ms noise energy estimation occasionally comes with some undesirable rhythmic artifacts. Better time average designs and smoothing schemes need to be found in the future. Obviously, for the spectral subtraction algorithm itself, there are many optimization methods that are possible to be

explored to obtain further improvements in performance.

5.2.2 Post Processing

In post processing, the conducted simulations are preliminary, and many aspects can be improved and further explored.

In the time-frequency masking approach, so far only approximate W-DO properties of some instruments pairs are studied. It shows that a time-frequency grid of more frequency resolution (and less time resolution accordingly) will better satisfy W-DO for the same instrument pairs. In the future, systematic studies can be carried on more concurrent sound sources and larger variety of musical instruments. Along this direction, a more detailed description for good candidate signals is yet to be defined. Also, a complete separation method based on this approach can be simulated and compared to what is geared toward speech signals in the literature. More exploration can also be focused on one merit of this approach, that when using only two microphones, stereo content can be preserved after the source removal process.

Another major problem that remains for this approach is the sensitivity to signal phases. A robust and powerful method to search for the optimum amplitude ratio / phase pair is yet to be discovered, so that the accuracy of the estimation is not solely based on measurements, but also refinable through algorithms.

Finally, if more microphones can be added to this system, a lot of possi-

bilities can emerge based on the basic approaches mentioned in this dissertation. As a simple example, if a 3 microphone system is available, each pair of microphone can perform BSS to obtain a mono output signal, and the whole system can work and make a decent stereo recording with reference sounds removed from the mixtures. More microphones can also free the system from degenerative situations, so that more sound sources may be separated and multichannel audio contents can be supported in such kind of systems.

A

Matlab Simulation for Multi-band Adaptive Equalizer

<sbnlms_sim.m>

```
% ------------------------- tunable parameters -----------------------------
T0 = 32;                        % hamming window length
fft_size = T0;                  % fft size
T = T0;                         % characteristic time length
D = T0/4;                       % window shift D = 1/F, F = 4/T0, for hamming

fake_delay = 5;                 % fake delay for cal.  optimum stepsize
taps = 10;                      % adaptive filter order for each subband
taps = taps + fake_delay;       % true taps, FIR order+fake delay
B = taps;                       % block NLMS block size

xTC = 0.94;                     % time const for input signal power
eTC = 0.95;                     % time const for error signal power
alpha = 0.0001;                 % constant to avoid division by zero
mu_fix = 2.0;                   % fixed stepsize

% ------------------------- signals and STFT -------------------------------
window = hamming(T0);           % create hamming window
mu = ones(fft_size, 1)*2;       % init stepsize for each band (2.0)

% microphone captured desired signal d(n)
[x, fs, nbits] = wavread('..\inst_test_files\kokoro.wav');

% num of subband samples
size_n = min (floor((length(x)-fft_size) / D) + 1, ...
floor((length(d)-fft_size) / D) + 1 );

% do STFT on x(n) and d(n)
Xmn = zeros(fft_size, size_n);
Dmn = zeros(fft_size, size_n);
for n = 1:size_n
```

```
    Xmn(:, n) = x((n-1)*D + 1 :  (n-1)*D + fft_size) .* window;
    Dmn(:, n) = d((n-1)*D + 1 :  (n-1)*D + fft_size) .* window;
end
Xmn = fft(Xmn, fft_size, 1);
Dmn = fft(Dmn, fft_size, 1);

% add artifitial delay to the captured desired signal
Dmn = [zeros(fft_size, fake_delay) Dmn];

% ------------------------- filter and update ---------------------------
% init subband adaptive filterbank
w = zeros(fft_size, taps);
w(:, 1 + fake_delay) = ones(fft_size, 1);     % allpass with fake delay

% filter:  linear convolution
Ymn = zeros(fft_size, size_n);     % output STFT
Xmn = [zeros(fft_size, taps-1) Xmn]; % pad input STFT with histories

% initialize prameters
inBlockIndex = 0;                  % counter for block NLMS block size
phi_vec = zeros(fft_size, taps);   % vector to accumulate gradient vector
x_2_accum = zeros(fft_size, 1);    % signal energy sum per NLMS block
x_power = zeros(fft_size, 1);      % input signal power estimate
e_power = zeros(fft_size, 1);      % error signal power estimate

% values to visualize
e_display = zeros(fft_size, size_n);             % error signal
mu_display = zeros(fft_size, floor(size_n/B));   % stepsize
muIndex = 0;

% ============================ the main loop ==============================
for n = 1:size_n
    % subband FIR filter
    x_vec = Xmn(:, n+taps-1:-1:n); % x(n), x(n-1), ...  u(n-taps+1)
    Ymn(:, n) = sum(conj(w) .* x_vec, 2); % linear convolution
    % error
    e = Dmn(:, n) - Ymn(:, n);

    % calculate Euclidean norm ||x(n)||^ 2 = x'(n) * x(n), and accumulate
    x_2 = zeros(fft_size, 1);
    for i = 1:fft_size
        x_2(i) = real(x_vec(i, :)  * x_vec(i, :)');
    end
    x_2_accum = x_2_accum + x_2;

    % calculate error as % of Euclidean norm, for visualization
    e_display(:,n) = 100 * abs(e) ./ sqrt(x_2+alpha);

    % accumulate gradient vector (estimation)

    for i = 1:  taps
```

```
    phi_vec(:,i) = phi_vec(:,i) + x_vec(:,i) .* conj(e);
end

% calculate signal power estimation (1st order IIR)
x_power = (1-xTC) * x_power + xTC * (conj(x_vec(:,1)) .* x_vec(:,1));

% calculate error power estimation (1st order IIR)
e_power = (1-eTC) * e_power + eTC * (conj(e) .* e);

% increase counter within block
inBlockIndex = inBlockIndex + 1;

% when it reaches one block, update the filter weights once
if inBlockIndex == B

    % update filter taps (NLMS)
    for i = 1:  taps
        w(:,i) = w(:,i) + (mu ./ (alpha + x_2_accum)) .* phi_vec(:,i);
    end

    % update stepsize, calculate for optimum stepsize and then compare
    %
    %           MSD(n) E[u^ 2(n)]                L   Ld-1 ^
    % mu_opt = ------------------- and MSD(n) = ---- sum  w^ 2k(n)
    %              E[e^ 2(n)]                    Ld  k=0

    MSD = zeros(fft_size, 1);
    for k = 1 :  fake_delay
        MSD = MSD + conj(w(:,k)) .* w(:,k);
    end
    MSD = MSD * taps / fake_delay;

    mu_opt = MSD .* x_power ./ (e_power + alpha);

    for k = 1 :  fft_size
        if mu_opt(k) < 2.0
            mu(k) = mu_opt(k);
        else
            mu(k) = mu_fix;
        end
    end

    % constant stepsize
    % mu = ones(fft_size, 1) * 1.0;

    % prepare to display mu histogram
    muIndex = muIndex + 1;
    mu_display(:, muIndex) = mu;

    % reset necessary parameters
    inBlockIndex = 0;
```

```
        phi_vec = zeros(fft_size, taps);
        x_2_accum = zeros(fft_size, 1);
    end % end of updating filter
end
% ========================= end of the main loop ==========================

% --------------------------- synthesis output y(n) ------------------------
% output buffer
y = zeros(length(x), 1);

Ymn = ifft(Ymn);

% apply synthesis hamming window, then overlap-add
for n = 1:size_n
    Ymn(:, n) = Ymn(:, n) .* window;
end

for n = 1:size_n
    y((n-1)*D+1 :  (n-1)*D+fft_size) = ...
    y((n-1)*D+1 :  (n-1)*D+fft_size) + real(Ymn(:, n))/2;
end

wavwrite(y, fs, nbits, 'output.wav');
```

B

Matlab Simulation for BSS with Spatial Acoustic Estimation

```
<source_removal.m>

HOS = 1; %1 for High Order Statistics (HOS), 0 for second order statistics

[vocal,fs] = wavread('billie_vocal.wav');
stereo = wavread('billie_excerpt.wav');

wholeLength = length(stereo);

nfft = 2*2048;
overlap = 3*nfft/4;

devocal_vec = zeros(nfft, 2);
a21_div_a11 = zeros(nfft, 1);

% Note that here noise is added deliberately to show the use of HOS.
vocal = vocal + 0.1*randn(length(vocal),1)*[1 1];

X11 = stft(vocal(:,1),hamming(nfft, 'periodic'),overlap);
X21 = stft(vocal(:,2),hamming(nfft, 'periodic'),overlap);

Fx1 = stft(stereo(:,1),hamming(nfft, 'periodic'),overlap);
Fx2 = stft(stereo(:,2),hamming(nfft, 'periodic'),overlap);

for index = 1:nfft,
    R1 = [X11(index,:); X21(index,:)]*[X11(index,:); X21(index,:)]';

    if HOS,
        R3 = [X11(index,:); X21(index,:)]*[(X11(index,:).^ 2).* ...
            conj(X11(index,:)); (X21(index,:).^ 2).*conj(X21(index,:))]';
        R = R3-3*R1;
    else
        R = R1;
    end
    [U1,S1,V1] = svd(R);
    a21_div_a11(index)=U1(2,1)/U1(1,1);
```

```
    devocal_vec(index,:)=[-a21_div_a11(index) 1];
end

Faccompaniment = Fx1;

for index = 1:nfft
    Faccompaniment(index, :)  = ...
        0.6 * devocal_vec(index,:)  * [Fx1(index, :); Fx2(index, :)];
end

accompaniment = istft(Faccompaniment, hamming(nfft, 'periodic'), overlap);
s = accompaniment;
```

<stft.m>

```
function [B] = stft (x,window,overlap)
% [B] = stft (a,window,overlap)
% This function calculate the STFT for a samples
% x - sample vector
% nfft - window size
% window - window values
% overlap - overlap hop

nfft = length(window);
step = nfft - overlap;
win_pos = [1:  step:  length(x) - nfft];
B = zeros(nfft,length(win_pos));

for i=1:length(win_pos)
    B(:,i) = x(win_pos(i):win_pos(i)+nfft-1).*window;
end

STFT = fft(B);

%return value
B = STFT;
```

<istft.m>

```
function [x] = istft (B,window,overlap)
% [x] = IStft (B,window,overlap)
% This function calculate the inverse STFT for a STFT matrix
% B - STFT matrix
% nfft - window size
% window - window values
% overlap - overlap hop

nfft = length(window);
STFT = real(ifft(B));
```

```
step = nfft - overlap;

W0 = sum(window(1:step:nfft-1));
[M N] = size(STFT);

x = zeros (1,(N+1)*(step-1));
win_pos = [1:  step:  length(x) - nfft];

for i=1:length(win_pos)
    x(win_pos(i):win_pos(i)+nfft-1) = ...
        x(win_pos(i):win_pos(i)+nfft-1) + STFT(:,i)';
end

x = x / W0;
```

References

[1] Agüero, J., Stefanelli, M. C., de Pérez, T. A., and D'Alvano, F., 1998: Echo reduction performance comparison between a classical adaptive noise canceller and wavelet packet adaptive noise canceller. *International Conference on Signal Processing Applications and Technologies (ICSPAT)*.

[2] Ahgren, P., 2004: *On System Identification and Acoustic Echo Cancellation*. Ph.D. thesis, Uppsala University.

[3] Allen, J. B., 1977: Short term spectral analysis, synthesis, and modification by discrete fourier transform. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, **25**(3), 235–238.

[4] Aoki, M., Okamoto, M., Aoki, S., Matsui, H., Sakurai, T., and Kaneda, Y., 2001: Sound source segregation based on estimating incident angle of each frequency component of input signals acquired by multiple microphones. *Acoustical Science & Technology*, **22**(2), 149–157.

[5] Balan, R., and Rosca, J., 2001: Statistical properties of STFT ratios for two channel systems and applications to blind source separation. *International Workshop on Independent Component Analysis and Blind Source Separation (ICA)*, 429–434.

[6] Balan, R., Rosca, J., and Rickard, S., 2001: Robustness of parametric source demixing in echoic environments. *3rd International Conference on Independent Component Analysis and Blind Source Separation (ICA)*.

[7] Balan, R., Rosca, J., Rickard, S., and O'Ruanaidh, J., 2000: The influence of windowing on time delay estimates. *Proceedings of the 35th Annual Conference on Information Sciences and Systems (CISS)*, **1**, 15–17.

[8] Barbosa, A., 2003: Displaced soundscapes: a survey of networked systems for music and sonic art creation. *Leonardo Music Journal*, **13**, 53–59.

[9] Benesty, J., Morgan, D. R., and Cho, J. H., 2000: A new class of double talk detectors based on cross-correlations. *IEEE Transactions on Speech and Audio Processing*, **8**(2), 168–172.

[10] Benesty, J., Morgan, D. R., and Cho, J. H., 2000: A new class of doubletalk detectors based on cross-correlation. *IEEE Transaction on Speech Audio Processing*, **8**, 168–172.

[11] Bofill, P., and Zibulevsky, M., 2000: Blind separation of more sources than mixtures using sparsity of their short-time fourier transform. *International Workshop on Independent Component Analysis and Blind Signal Separation (ICA)*, 87–92.

[12] Boll, S. F., 1979: Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **27**(2), 113–120.

[13] Boll, S. F., and Pulsipher, D. C., 1979: Suppression of acoustic noise in speech using two microphone adaptive noise cancellation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **27**(2), 113–120.

[14] Buchner, H., Benesty, J., Gänsler, T., and Kellermann, W., 2003: An outlier-robust extended multidelay filter with application to acoustic echo cancellation. *Proceedings of 8th IEEE International Workshop on Acoustic Echo and Noise Control*.

[15] Buchner, H., Benesty, J., and Kellermann, W., 2003: An extended multidelay filter: Fast low-delay algorithms for very high-order adaptive systems. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[16] Buchner, H., Benesty, J., and Kellermann, W., 2003: Multichannel frequency-domain adaptive filtering with application to acoustic echo cancellation. *In J. Benesty and Y. Huang (eds.), Adaptive signal processing: Application to real-world problems*.

[17] Buchner, H., Herbordt, W., and Kellermann, W., 2001: Acoustic echo cancellation for two and more reproduction channels. *Conference Record of IEEE International Workshop on Acoustic Echo and Noise Control (IWAENC)*, 90–102.

[18] Buchner, H., and Kellermann, W., 2001: An efficient combination of multichannel acoustic echo cancellation with a beamforming microphone array. *Conference Record of 1st IEEE International Workshop on Hands-Free Speech Communication (HSC)*, 55–58.

[19] Buchner, H., and Kellermann, W., 2002: Improved kalman gain computation for multichannel frequency-domain adaptive filtering and application to acoustic echo cancellation. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1909–1912.

[20] Buchner, H., Spors, S., Kellermann, W., and Rabenstein, R., 2002: Full-duplex communication systems with loudspeaker arrays and microphone arrays. *Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Special session on coding and transmission formats for 3D audio.*

[21] Chafe, C., Curevich, M., Leslie, G., and Tyan, S., 2004: Effect of time delay on ensemble accuracy. *Proceedings of the International Symposium on Musical Acoustics (ISMA)*, **13**.

[22] Chafe, C., Wilson, S., Leistikow, R., Chisholm, D., and Scavone, G., 2005: A simplified approach to high quality music and sound over IP. *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*, 209–219.

[23] Chen, J., Bes, H., and Vandewalle, J., 1996: A zero-delay FFT-based subband acoustic echo canceller for teleconferencing and hands-free telephone systems. *IEEE Transactions on Circuits and System II Analog Digital Signal Processing*, **43**(10), 713–717.

[24] Dubnov, S., Tabrikian, J., and Arnon-Targan, M., 2004: A method for directionally-disjoint source separation in convolutive environment. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, **5**, 489C492.

[25] Duttweiler, D. L., 1978: A twelve-channel digital echo canceler. *IEEE Transaction on Communication*, **26**, 647–653.

[26] Elko, G. W., Diethorn, E., and Gänsler, T., 2003: Room impulse response variation due to thermal fluctuation and its impact on acoustic echo cancellation. *International Workshop on Acoustic Echo and Noise Control (IWAENC).*

[27] Enzner, G., and Vary, P., to appear in 2005: Frequency-domain adaptive kalman filter for acoustic echo control in handsfree telephones. *Signal Processing, Special Issue on Speech and Audio Processing.*

[28] Föllmer, G., 2002: *Musikmachen im Netz Elektronische, ästhetische und soziale Strukturen einer partizipativen Musik.* Ph.D. thesis, Martin-Luther-Universitä Halle-Wittenberg.

[29] Gilliore, A., and Vetterlli, M., 1992: Adaptive filtering in subbands with critical sampling: analysis, experiments, and application to acoustic echo cancellation. *IEEE Transactions on Signal Processing*, **40**(8), 1862–1875.

[30] Goto, M., Neyama, R., and Muraoka, Y., 1997: RMCP: Remote music control protocol - design and applications. *Proceedings of the 1997 International Computer Music Conference*, 446–449.

[31] Haykin, S., 1996: *Adaptive Filter Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, 3rd edition edition.

[32] Heitkämper, P., and Walker, M., 1995: Adaptive gain control for speech quality improvement and echo suppression. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 3047–3050.

[33] Hulle, M. V., 1999: Clustering approach to square and non-square blind source separation. *IEEE Workshop on Neural Networks for Signal Processing (NNSP)*, 315–323.

[34] Jourjine, A., Rickard, S., and Yılmaz, Ö., 2000: Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, **5**, 2985–2988.

[35] Jung, H. K., Kim, N. S., and Kim, T., 2005: A new double-talk detector using echo path estimation. *Speech Communication*, **45**(11), 41–48.

[36] Kapur, A., Wang, G., Davidson, P., and Cook, P. R., 2005: Interactive network performance: a dream worth dreaming? *Organised Sound*, **10**(3), 209–219.

[37] Kellermann, W., 1991: A self-steering digital microphone array. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 3581–3584.

[38] Knappe, M. E., and Goubran, R. A., 1994: Steady state performance limitations of full-band acoustic echo cancelers. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, **2**, 73–76.

[39] Krim, H., and Viberg, M., 1996: Two decades of array signal processing research, the parametric approach. *IEEE Signal Processing Magazine*, 67–94.

[40] Krukowski, A., and Kale, I., 2003: Polyphase IIR filter banks for subband adaptive echo cancellation applications. *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2003)*, **4**(2), 405–408.

[41] Kuttruff, H., c1973: *Room Acoustics*. Prentice-Hall, New York, Wiley. ISBN 0470511052.

[42] Lee, T.-W., Lewicki, M., Girolami, M., and Sejnowski, T., 1999: Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Signal Processing Letters*, **6**(4), 87–90.

[43] Liavas, A. P., and Regalia, P. A., 1998: Acoustic echo cancellation: IIR models offer better modeling capabilities than their FIR counterparts. *IEEE Transactions on Signal Processing*, **46**, 2499–2504.

[44] Lin, J.-K., Grier, D. G., and Cowan, J. D., 1997: Feature extraction approach to blind source separation. *IEEE Workshop on Neural Networks for Signal Processing (NNSP)*, 398–405.

[45] lu, Y., and Morris, J., 1999: Gabor expansion for adaptive echo cancellation. *IEEE Signal Processing Magazine*, 68–80.

[46] Mader, A., H.puder, and Schmidt, G., 2000: Step-size control for acoustic echo cancellation filters - an overview. *Signal Processing*, **80**, 1697–1719.

[47] Marelli, D., and Fu, M., 2001: System identification using subband signal processing. *Proceedings of 40th IEEE Conference on Decision and Control*, **4**, 3485–3490.

[48] Mendel, J. M., 1991: Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications. *Proceedings of the IEEE*, **79**, 278–305.

[49] Mouba, J., and Marchand, S., 2006: A source localization / separation / respatialization system based on unsupervised classification of internal cues. *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, 233–238.

[50] Naylor, P. A., and Hart, J. E., 1998: Subband adaptive filtering for acoustic echo control using allpass polyphase IIR filter banks. *IEEE Transactions on Speech and Audio Processing*, **6**(2).

[51] Nguyen, L. T., Abed-Meraim, A. B. K., and Boashash, B., 2001: Separating more sources than sensors using time-frequency distributions. *International Symposium on Signal Processing and its Applications (ISSPA)*, 583–586.

[52] Nomura, T., Denki, M., and Kaisha, K., 1996: Voice canceler with simulated stereo output. *U.S. Pat. 05550920*.

[53] Puckette, M., 2007: The online book. *Theory and Techniques of Electronic Music*, http://crca.ucsd.edu/ ˜msp/techniques/latest/book–html/.

[54] Puckette, M., Sorensen, V., and Steiger, R., 1997: *Lemma 1*, performed live at Milos Jazz Club. *International Computer Music Conference*.

[55] Rickard, S., and Dietrich, F., 2000: DOA estimation of many w-disjoint orthogonal sources from two mixtures using duet. *Proceedings of the 10th IEEE Workshop on Statistical Signal and Array Processing (SSAP)*, 311–314.

[56] Rickard, S., and Yılmaz, Ö., 2002: On the approximate w-disjoint orthogonality of speech. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, **5**, 2985–2988.

[57] Sakauchi, S., Haneda, Y., Okamoto, M., Sasaki, J., and Kataoka, A., 2004: Echo canceller with noise reduction provides comfortable hands-free telecommunication in noise environments. *NTT Technical Review*, **2**(3), 59–63.

[58] Schmidt, G., 1999: Step-size control in subband echo cancellation systems. *Proceedings of International Workshop in Acoustic Echo and Noise Control (IWAENC)*, 116–119.

[59] Strobl, G., and Holzmann, G., 2005: Adaptive: a pd-external library for adaptive systems and filters. *http://grh.mur.at/software/adaptive.html*.

[60] Veen, B. D. V., and Buckley, K. M., 1988: Beamforming: a versatile approach to spatial filtering. *IEEE Acoustics, Speech, and Signal Processing Magazine*, 4–24.

[61] Weinberg, G., 2002: *Interconnected Musical Networks: Bringing Expression and Thoughtfulness to Collaborative Music Making*. Ph.D. thesis, MIT Media Lab.

[62] Windrow, B., and Stearns, S. D., 1985: *Adaptive signal processing*. Prentice-Hall, Upper Saddle River, NJ.

[63] Wright, M., Freed, A., and Momeni, A., 2003: Open sound control: state of the art 2003. *Proceedings of the 2003 Conference on New Interface for Musical Expression (NIME-03)*, 153–159.

[64] Xiang, P., and Dubnov, S., 2005: Karaoke system with spatial acoustics estimation for vocal or instrumental removal. *International Computer Music Conference (ICMC)*.

[65] Yamamoto, S., and Kitayama, S., 1982: An adaptive echo canceller with variable step gain method. *Transactions on IECE Japan*, **65**, 1–8.

[66] Ye, H., and Wu, B. X., 1991: A new double-talk algorithm based on the orthogonality theorem. *IEEE Transactions on Communication*, **39**(11), 1542–1545.

[67] Yılmaz, Ö., and Rickard, S., 2004: Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, **52**(7), 1830–1847.