A SOFTWARE STEREO ACOUSTIC ECHO CANCELER FOR MICROSOFT WINDOWS™

V. Fischer[1], T. Gänsler[2], E. J. Diethorn[2], and J. Benesty[3]

[1] Technische Universität Darmstadt, Institut für Nachrichtentechnik
Fachgebiet Theorie der Signale
Merckstr. 25, 64283 Darmstadt

[2]Agere Systems, Research
[3]Bell Laboratories, Lucent Technologies
600 Mountain Avenue, Murray Hill, New Jersey 07974-0636
corrados@gmx.de, {gaensler,diethorn}@agere.com, jbenesty@bell-labs.com

ABSTRACT

A software has been designed that successfully runs a stereophonic acoustic echo canceler natively on a personal computer (PC). This may seem like an easy task considering that echo cancelers have been implemented on VLSI- and DSP-platforms for years. However, there are two features that differentiate this work from all previous implementations: First, stereophonic echo cancellation is significantly more complicated to handle than the monophonic case because of computational complexity, nonuniqueness, and convergence problems. Special care has to be taken in the algorithm design. Second, echo cancellation requires that the soundcard's input and output signals are time-synchronous. The latter is a great challenge to achieve in such an "asynchronous" environment as the operating system of a PC. This work presents the system design and how the input/output audio-streams are synchronized to maintain a stable cancellation performance under Microsoft Windows™ 2000.

1. INTRODUCTION

Real-time echo cancellation requires a significant amount of computational resources. So far, from a computational point of view, real-time implementation has only been possible using custom designed very large scale integration (VLSI) circuits or digital signal processors (DSPs) [8]. These processors are specifically designed for signal processing tasks. They provide parallel processing of operations and optimized pipeline structures. However, because the computation power of regular personal computers (PCs) has increased tremendously in the last few years, it is now possible to perform very demanding real-time signal processing in this environment as well. The objective of this paper is to present a flexible echo cancellation software running natively under the operating system (OS) on a PC.

A block diagram of a two-channel, point-to-point communication link with one[1] echo canceler is shown in Figure 1. We denote the signals picked up by the microphones

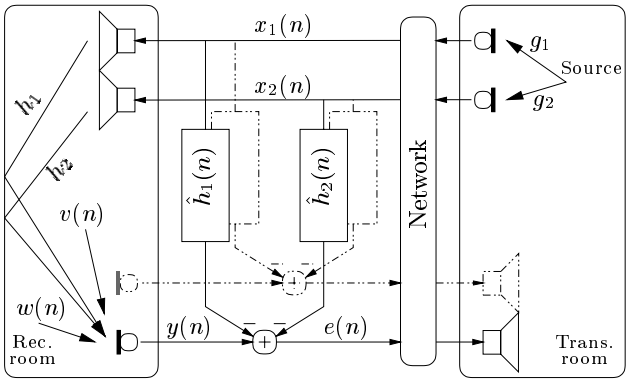

Figure 1: Block diagram of a generic two-channel acoustic echo canceler.

in the transmission room by $x_1(n)$, $x_2(n)$, and the return signal picked up by one of the microphones in the receiving room by $y(n)$. The receiving room signal is in general composed of echo, ambient noise $w(n)$, and possibly receiving room speech $v(n)$. Hence, we have the receiving room signal model: $y(n) = y_e(n) + v(n) + w(n)$, where $y_e(n) = \sum_{p=1}^{2} h_p * x_p(n)$ is the echo, $*$ denotes convolution, and h_p, $p = 1, 2$ are the receiving room echo paths.

Our echo canceler implementation provides the capability of communicating hands-free in single-channel mode (receive one and transmit one audio-stream), synthetic stereo mode (receive two and transmit one stream), or full stereo mode (receive two and transmit two audio-streams). In the full stereo case, natural stereo is transmitted to the receiving side. In the synthetic case, synthesized stereo [2] or 3D-audio [4] is generated from the mono audio stream at an intermediate conference server. The bandwidth of the audio is 8 kHz. To accommodate different choices of environments, the echo canceler can span acoustic paths of lengths 32, 64, 128, or 256 ms.

In Sect. 2, the structure of the software and its components are described. Operation modes, like point-to-point and multi-point, are described in Sect. 3. Sect. 4 shows evaluation of the echo canceler performance and recording features of the software, and a discussion is given in Sect. 5.

[1]In a real-life situation, we need an echo canceler for the "transmission room" as well. However, for simplicity, we chose to exclude it in the figure.

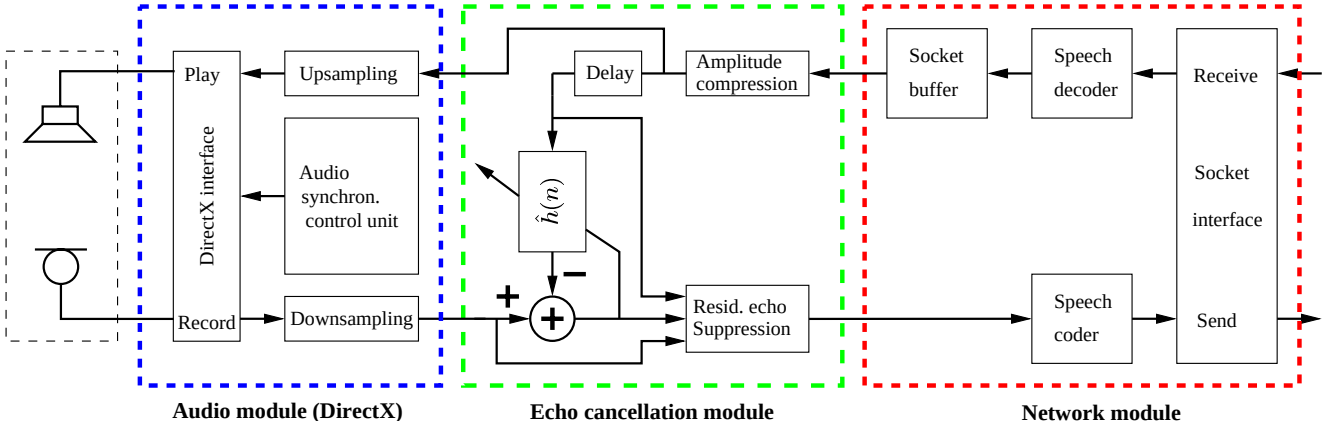

Figure 2: System block diagram for the single-channel case.

2. SYSTEM STRUCTURE

Figure 2 shows a block diagram of the entire software system and acoustic path for the single-channel case. This system primarily consists of three components; the audio module, the echo cancellation module, and the network module. An overview of these modules follows.

2.1. The Audio Module

The audio module is an interface between our software and the Windows DirectX interface [5]. DirectX provides a general interface between the Windows OS and different soundcard drivers. The Windows DirectX interface is well defined and stable. However, a major problem is the so-called device driver that sits between this interface and the actual soundcard hardware. This driver is designed by the manufacturer of the soundcard hardware and it is difficult to predict how it interacts with the Windows OS operations.

The major problem of implementing an echo canceler on a PC is loss synchronization of the audio streams. This causes instantaneous delay variation of the actual echo path and the canceler cannot track these changes fast enough to achieve proper cancellation. There are two primary problems that must be solved: (1) Correcting for synchronization failure after a temporary 100% CPU utilization and, (2) Eliminating the stream synchronization problem observed with Windows sample-rate conversion. It is the objective of the audio delay control unit within the audio module, to keep the audio streams synchronized in spite of these two problems.

2.2. The Echo Canceler Module

The core echo canceler consists of a robust two-channel frequency-domain adaptive algorithm, a pseudo-coherence based double-talk detector, and a residual echo suppression unit. The choice of this solution is based on the facts that we need a low complex algorithm, compared to a time domain solution, that can handle the problem of slow convergence in the two-channel case. This could be achieved with a subband solution as in [7], however, a more complicated adaptive algorithm (fast recursive least-squares) would be required. The frequency-domain solution has been shown to provide a reasonable trade-off between complexity and performance [6]. The specific problem of stereophonic echo cancellation, i.e., the nonuniqueness problem, is handled by nonlinear distortion as described in [3]. Details regarding the adaptive algorithm and double-talk detector can be found in [1]. The suppression algorithm attenuates the residual echo $e(n)$ (Fig. 1) depending on the actual amount of echo cancellation. The adaptation of the attenuation is based on voice activity detection decisions and an echo-return-loss measurement. All the measurements are based on the envelopes of the speech signals and the noise.

If the dynamic range of the receive signal is near full range, unmodeled nonlinearities in the echo path will likely be excited. Therefore, compression of the incoming far-end signal's amplitude is done.

2.3. The Network Module

The network module controls the data transfer between the two connected clients. The main tasks are buffering the different network-side audio streams and compressing (audio/speech coding) the audio data if desired.

The Windows socket interface is used to transmit the data through the network as a user datagram protocol (UDP) packet. This interface deals with all protocol tasks and requires only a port number and the IP-address of the receiving client.

The socket buffer acts as a cache between the clients. It deals with three tasks: synchronization of data blocks, adjustment to different block sizes, and correcting for buffer underflow because of network problems.

The audio compression algorithm [9] we use, is compiled into a Windows dll-file. The provided compression rates at a sample rate of 16 kHz are: 32, 24, or 16 kbits/s for mono transmission. So far, stereo coding has not been considered.

3. MODES OF OPERATION

We have tested and evaluated the system in various environments and operation modes. Point-to-point communication in full stereo or mono has been made in an office environment using a desktop or a laptop computer, and in a larger conference room, where the loudspeakers and microphones are far apart. Multi-point communication has been performed using a mix of laptop and desktop machines.

3.1. Point-to-Point Communication

In point-to-point communication, two clients (PCs) are directly connected to each other through a network. Tests

with typical office environments showed that a sufficient impulse response length is 64 ms for achieving good echo cancellation. It is possible to use 32 ms, but the quality decreases because the suppression has to act more aggressively to reduce residual echo.

In the conference situation, the audio system is physically distributed, hence the distance between the microphone and loudspeakers is increased. Furthermore, the room is usually larger than a regular office. In this case, a 32 ms filter will result in poor echo cancellation, especially in stereo where the tail effects become severe [3]. The minimum length of the estimated filter that worked well was of length 64 ms, but an improvement can be achieved with 128 ms.

What characterizes a laptop audio system is that the microphone and the loudspeakers are very close. Moreover, the loudspeakers are of poor quality. There is also a significantly high noise level originating from the computer components (e.g., motor in the hard-drive) which are very close to the microphone. In this situation, it is not possible to improve the echo cancellation results by simply increasing the impulse response length. A good choice is a short filter of length 32 ms. Other serious problems are the nonlinearity of the loudspeaker and the resonance of the laptop case, both of which cannot be properly modeled with a linear filter. Also, the keyboard is often between the microphone and the loudspeaker and, if it is used, the impulse response rapidly changes all the time. To achieve good performance in this environment, more suppression is required.

3.2. Multi-point Communication

The conference server, also designed by the authors, is an audio bridge that connects many clients and creates synthesized stereo or 3D-audio streams. Each user that connects to the server will hear all other clients connected to the same server. Because returning echoes are summed at the server, this situation places tougher requirements on echo cancellation. Furthermore, because of the nonuniqueness problem, the most difficult situation for the echo canceler to handle is the synthetic stereo/3D-audio case. Therefore, all clients have to adjust the cancellation parameters (i.e. impulse response length, suppression parameters etc.) and the audio system very carefully. With synthesized stereo and four clients, the positions of the different speakers are well distinguished. With 3D-audio, the audio distribution can be improved.

4. PERFORMANCE AND COMPLEXITY

The setup for the measurements is a regular office with a desktop computer audio system. A schematic drawing of this office is shown in Fig. 3. We used two loudspeakers positioned on each side of the monitor and a stereo microphone in front of the monitor. White, mutually uncorrelated Gaussian noise is played through the loudspeakers. For simplicity, we only show the results of one return channel.

The two returning audio streams $y(n)$ (microphone signal) and $e(n)$ (error signal) are recorded (internally in our software). The echo suppressor is switched off in this measurement. The two signals and an estimate of the normalized excess mean-square error (ex. MSE) are plotted in Fig. 4. The excess mean-square error is defined as

$$\text{ex. MSE}(n) = \frac{\text{LPF}\left\{[e(n)]^2\right\} - \sigma_w^2}{\text{LPF}\left\{[y(n)]^2\right\} - \sigma_w^2},$$

where σ_w^2 is the variance of the ambient noise estimated when the office is quiet. With the system parameter settings and the echo-to-ambient-noise ratio ENR $= \sigma_{y_e}^2/\sigma_w^2$ equal to 23.6 dB, the maximum theoretically attainable echo attenuation [1] is ≈ -28.4 dB with our choice of algorithm parameters. This value is almost reached by the canceler as seen in the mean-square error performance in Fig. 4. The average ex. MSE measured from 4 to 7 s is -26.8 dB. At time 8 s there is double-talk (ex. MSE is not presented correctly during this period of time), and from 12 to 14 s the microphone is moved and the algorithm has to readapt. Figure 5 shows the four estimated impulse responses. Responses like these can be saved at any desired time.

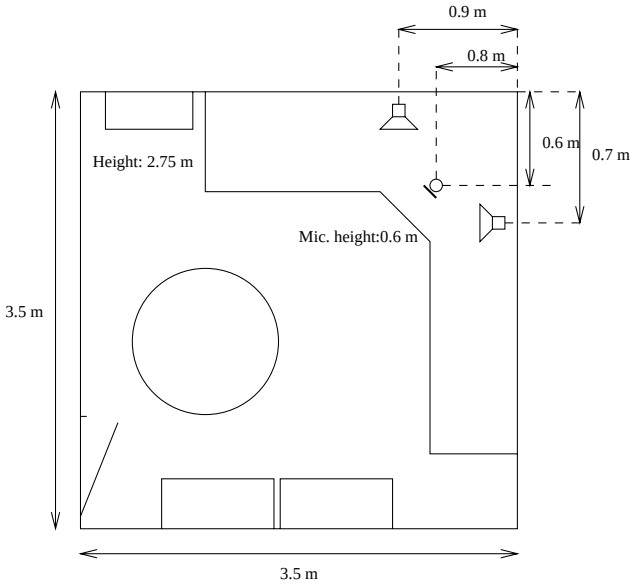

Figure 3: Layout of the office and acoustic setup.

The computer load depends strongly on the choice of the parameter. To give an approximate idea, the CPU usage (measured with the Windows task manager) with an Intel Pentium III 550 MHz processor and the operating system Windows2000 is lower than 80 % with stereo echo cancellation and a filter length of 1024 taps (64 ms).

5. DISCUSSION

In this paper, we have described an implementation of a flexible stereophonic acoustic echo canceler. This implementation runs natively under Microsoft Windows^{TM} on a PC. The major obstacle with such a scheme is the synchronization problem of the input and output audio streams of the soundcard. Without proper synchronization, good cancellation cannot be maintained.

Evaluation of the echo canceler has shown that it achieves the theoretical bounds on performance (echo attenuation) which in general is ≈ 5 dB below the room noise level (algorithm parameter dependent). This performance is valid for an echo-to-noise ratio down to about 35 dB. In practice, we cannot expect more cancellation because of the linear model mismatch, non-stationary room responses, and unmodeled tails of the responses. Attenuation of 20 to 35

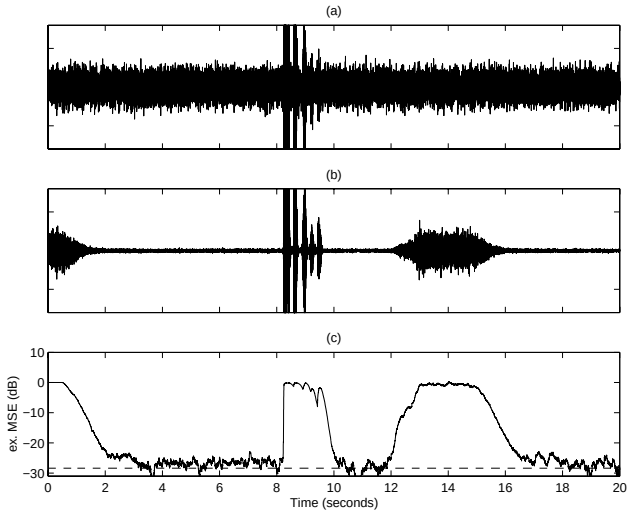

Figure 4: (a) Echo $[y(n)]$. (b) Residual echo $[e(n)]$. (c) Estimate of the excess mean-square error performance (solid) and theoretical ex. MSE (dashed). There is double-talk at 8 s and the microphone is moved between approximately 12 to 14 s.

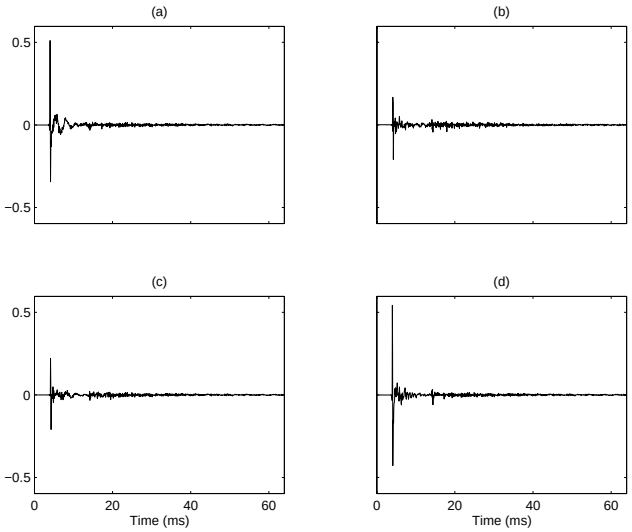

Figure 5: Estimated impulse responses in a office. Loudspeakers and microphones are separated by 20 cm. (a) Left mic. to left loudsp. (b) Left mic. to right loudsp. (c) Right mic. to left loudsp. (d) Right mic. to right loudsp.

dB is not sufficient since the round-trip delay of the system is fairly large, about 350 ms. This delay is mainly due to delay in the soundcard and network interface and is a function of the involved buffers' lengths (assuming insignificant network delay). Because of this, extra residual echo suppression is required and has been implemented.

The software can demonstrate mono, natural full stereo, and synthetic stereo hands-free communication. Multi-point communication can be done in mono or synthetic stereo/3D-audio mode. It is the intention of the authors to have a real-time demonstration system with multi-point capability available at the presentation of this work.

Acknowledgments

This paper is based on a Diplomarbeit done by V. Fischer at Lucent Technologies, Bell Labs/Agere Systems. Supervision and algorithm design were provided by the other authors. The first author would especially like to thank Prof. Dr.-Ing. E. Hänsler, Dr. G. W. Elko and Dr. P. Dreiseitel who made the Diplomarbeit possible.

6. REFERENCES

[1] Benesty J., Gänsler T., Morgan D. R., Sondhi M. M., and Gay S. L.: *Advances in Network and Acoustic Echo cancellation.* Springer-Verlag, Berlin, 2001.

[2] Benesty J., Morgan D. R., Hall J. L., and Sondhi M. M.: *Synthesized stereo combined with acoustic echo cancellation for desktop conferencing. Bell Labs Tech. J.*, 3:148–158, July-Sept. 1998.

[3] Benesty J., Morgan D. R., and Sondhi M. M.: *A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation. IEEE Trans. Speech Audio Processing*, 6:156–165, Mar. 1998.

[4] Chen J.: *3D audio and virtual acoustical environment synthesis.* In S. L. Gay and J. Benesty, editors, *Acoustic Signal Processing for Telecommunication*, chapter 13, pages 283–301. Kluwer Academic Publishers, 2000.

[5] Microsoft Corporation: *Microsoft developer network 6.0.* www.msdn.com, 2000.

[6] Eneroth P., Benesty J., Gänsler T., and Gay S. L.: *Comparison of different adaptive algorithms for stereophonic acoustic echo cancellation.* In *Proc. EUSIPCO*, pages 1835–1837, 2000.

[7] Eneroth P., Gay S. L., Gänsler T., and Benesty J.: *A real-time stereophonic acoustic subband echo canceler.* In S. L. Gay and J. Benesty, editors, *Acoustic Signal Processing for Telecommunications*, chapter 8, pages 135–152. Kluwer Academic Publishers, 2000.

[8] Nitsch B. H:. *Real-time implementation of the exact block NLMS algorithm for acoustic echo control in hands-free telephone systems.* In S. L. Gay and J. Benesty, editors, *Acoustic Signal Processing for Telecommunication*, chapter 4, pages 68–80. Kluwer Academic Publishers, 2000.

[9] Ramprashad S. A.: *A multimode transform predictive coder (MTPC) for speech and audio.* In *IEEE Speech Coding Workshop*, 1999.