FREQUENCY DOMAIN ACOUSTIC ECHO CANCELLER THAT HANDLES ASYNCHRONOUS A/D AND D/A CLOCKS

Mototsugu Abe and Masayuki Nishiguchi

SONY Corporation, Tokyo, JAPAN

ABSTRACT

An implicit premise in using an acoustic echo canceller (AEC) is that the clock for A/D conversion for a microphone and the clock for D/A conversion for a loudspeaker work synchronously. Even a slight difference in sampling rate between the clocks critically degrades the echo cancelling performance. This paper describes a method of making an AEC in the frequency domain that can handle a mismatch in sampling rate between A/D and D/A conversion. The method recursively estimates the sampling-rate offset by a simple extension of the well-known LMS algorithm, and corrects it through two mechanisms, frame-step control and phase rotation, which obviate the need for any explicit resampling operation. Experimental results show that this method provides an echo suppression level comparable to a standard AEC without mismatch for an offset of up to 1000 ppm.

***Index Terms*—** acoustic echo canceller, asynchronous clocks, sampling-rate mismatch, short-term Fourier transform, subband processing.

1. INTRODUCTION

An acoustic echo canceller (AEC) is widely used to eliminate loudspeaker-to-microphone feedback in hands-free telecommunication and automatic speech recognition systems[1]. The most common way to implement an AEC is to use an adaptive filter in the time domain (called a TDAEC) based on a least mean squares (LMS) algorithm[2]. Sometimes, frequency domain implementation is used to reduce computational costs and achieve faster convergence[1, 3, 4, 5] (called FDAEC).

An important restriction on conventional AECs is that the sampling rates of the microphone and loudspeaker must be exactly the same. Robledo-Arnuncio et. al.[6] found through empirical tests that the performance of an AEC is critically degraded by even a slight difference due to manufacturing error, which is generally in the range 10-1000 ppm. That means that we cannot use a standard AEC if the clocks of the A/D and D/A are not made from the same crystal. In modern complex devices (PCs, smartphones, smart TVs, etc.), this condition sometimes cannot be satisfied, because of the use of multiple audio devices, external devices connected by USB or wireless, and so on.

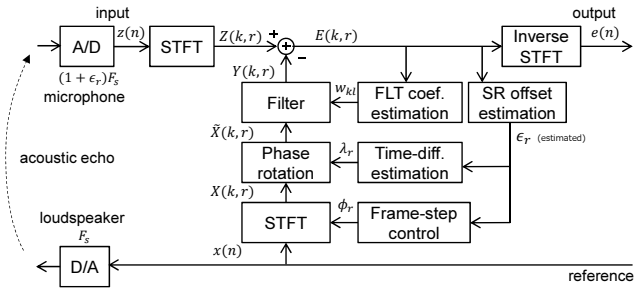

Fig. 1. Block diagram of proposed asynchronous FDAEC

Few attempts have been made to solve the problem of sampling-rate mismatch. Pawig et. al.[7] proposed a method based on an extension of the time-domain LMS algorithm. It recursively estimates the sampling-rate offset and uses an external resampling filter to correct the offset. They mentioned that the resampling filter is the most time-consuming part of their structure. A similar problem can be found in the context of an asynchronous microphone array[8, 9]. Miyabe et. al.[8] proposed method of estimating and correcting sampling-rate mismatch in the frequency domain based on the maximum-likelihood estimation and phase rotation.

In this paper, we address the problem in the frequency domain. Our method recursively estimates the sampling-rate offset by a simple extension of the LMS algorithm, and corrects it through two mechanisms: frame-step control and phase rotation. The estimation and correction are carried out in a single feedback loop without an external resampling filter (Fig.1) .

Section 2 presents a brief explanation of a standard FDAEC, and Section 3 extends it to deal with asynchronous A/D and D/A clocks (called an async-FDAEC). Section 4 presents some experiments, and Section 5 makes some concluding remarks.

2. FREQUENCY DOMAIN AEC

A block diagram of a standard FDAEC is shown in Fig.2, and the symbols used in the equations below are shown in Table 1.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

Constants and their typical values

F_s	(nominal) sampling frequency	16 kHz
M	window length	511 samples
N	FFT size	512 samples
R	(standard) frame step	128 samples
L	filter length	38 frames (300 ms)

Symbols for variables

n	discrete time	k	discrete frequency
r, ρ	frame index	l	filter index
w_{kl}	filter coefficients		
ϵ_r	estimate of sampling-rate offset for frame r		
λ_r	sub-sample time difference for frame r		
ϕ_r	(adaptive) frame-step of reference for frame r		

Table 1. Symbols and their typical values

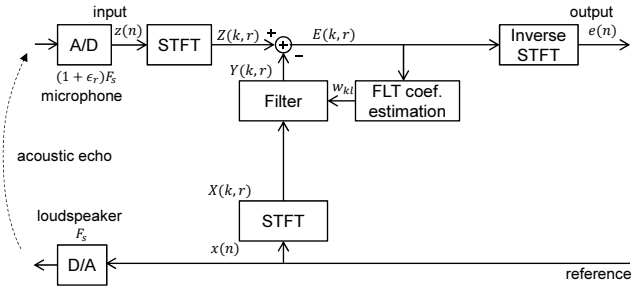

Fig. 2. Block diagram of standard FDAEC

Let $z(n)$ be the input signal from the microphone. Its subband representation based on a short-term Fourier transform (STFT) is

$$Z(k,r) \;=\; \sum_{n=-(M-1)/2}^{(M-1)/2} W(n)\; z(n+rR) e^{-j2\pi kn/N}, \quad (1)$$

where $W(n)$ is a window function for analysis.

Let $x(n)$ be the reference signal from the speaker line. Its subband representation is

$$X(k,r) \;=\; \sum_{n=-(M-1)/2}^{(M-1)/2} W(n)\; x(n+rR) e^{-j2\pi kn/N}. \quad (2)$$

The filtered version of the reference signal is

$$Y(k,r) \;=\; \sum_{l=0}^{L-1} w_{kl} X(k, r-l), \quad (3)$$

where w_{kl} is the filter coefficient for frequency k and frame l. The output signal in the frequency domain is calculated by subtracting the filtered signal from the input signal:

$$E(k,r) \;=\; Z(k,r) - Y(k,r). \quad (4)$$

Applying the inverse STFT yields the output signal in time domain:

$$e(n) \;=\; \sum_{r=0}^{\infty} W_s(n-rR) \sum_{k=-N/2}^{N/2-1} E(k,r) e^{2\pi jk(n-rR)/N}, \quad (5)$$

where $W_s(n)$ is a window function for synthesis. The filter coefficients are recursively estimated based on the LMS algorithm:

$$w_{kl}^{(\mathrm{new})} \;=\; w_{kl}^{(\mathrm{old})} + \mu_k \;\; E(k,r)\; X^*(k, r-l), \quad (6)$$

where μ_k is the learning coefficient for each frequency band, and $*$ denotes complex conjugate.

3. ASYNCHRONOUS FDAEC

3.1. Overview

In the explanatory drawing of the proposed async-FDAEC (Fig.3), the horizontal axis is time, and the top row shows the frame numbers. The second row shows the sampling positions of the input signal, and the bottom row shows those of the reference signal. The sampling rates are slightly different.

For the input signal, the standard STFT in eq.(1) is applied, with M being a fixed window length and R being a fixed frame step. For the reference signal, the same window length is used, but the frame steps are adaptively changed so that the time difference between the window centers of the input and the reference signals does not exceed ± 0.5 sample. Notice that, in the fourth row, the frame steps for frames 2 and 7 are smaller than the standard value, R. The remaining sub-sample time difference between the window centers in the third row is corrected by rotating the phase of the STFT of the reference signal. Note that the adverse effects of the sample jump due to the smaller frame step are adequately compensated for by a change in the direction of phase rotation.

Clearly, a good estimate of the sampling-rate offset is needed for this scheme to work properly. The next section describes the formulation of the scheme, as well as a formula for estimating the offset.

3.2. Formulation

Since the operation of the async-FDAEC (Fig.1) is recursive, we assume that the current frame number is r and that the estimates of the sampling-rate offset (ϵ_ρ), the frame steps (ϕ_ρ), and the time differences between the window centers (λ_ρ) of the preceding frames (i.e. for $\rho = 0, ..., r-1$) have already been calculated properly.

3.2.1. STFT of input signal

The STFT of the input signal is the same as eq.(1).

3.2.2. STFT of reference signal

The STFT of the reference signal is a modification of eq.(2):

$$X(k,r) \;=\; \sum_{n=-(M-1)/2}^{(M-1)/2} W(n)\; x(n+\Phi_r) e^{-j2\pi kn/N}, \quad (7)$$

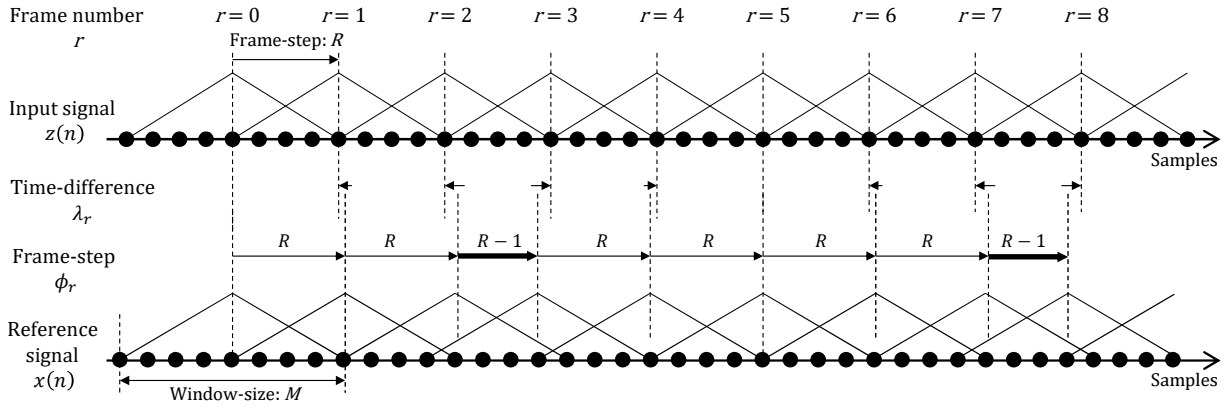

Fig. 3. Operation of async-FDAEC

where

$$\Phi_r \quad = \quad \sum_{\rho=0}^{r-1} \phi_\rho \tag{8}$$

is the position of the window center for frame r. For the first frame, $\Phi_0 = 0$.

3.2.3. Phase rotation

A phase-rotated version of the above STFT is

$$\tilde{X}(k,r) \quad = \quad e^{-j2\pi k\lambda_r/N} X(k,r), \tag{9}$$

where

$$\lambda_r \quad = \quad \sum_{\rho=0}^{r-1} \phi_\rho(1+\epsilon_\rho) - rR \tag{10}$$

is the estimate of the sub-sample time difference between the window centers for frame r. For the first frame, $\lambda_0 = 0$. Note that the first term of eq.(10) is the window center position of the reference signal mapped to the sampling rate of the input signal, and the second term is that of the input signal.

3.2.4. Filter operation, calculation of output signal, and estimation of filter coefficient

These operations are the same as eqs.(3), (4), (5) and (6) except for the replacement of X with $\tilde{X}$.

3.2.5. Estimation of sampling-rate offset

The sampling-rate offset for frame r is recursively estimated by a simple extension of the LMS algorithm:

$$\epsilon_r \quad = \quad \epsilon_{r-1} - \mu_\epsilon \Delta\epsilon_{r-1}, \tag{11}$$

where μ_ϵ is a learning coefficient, and

$$\begin{aligned} \Delta\epsilon_{r-1} \quad &= \quad \frac{\partial}{\partial \epsilon_{r-1}} \sum_{k=0}^{N/2} |E(k,r)|^2 \qquad (12) \\ &= \quad \sum_{k=0}^{N/2} k \,\mathrm{Im}\left(E(k,r) \sum_{l=0}^{L-1} w_{kl}^* \tilde{X}^*(k,r-l) \sum_{\rho=0}^{L-l} \phi_{r-L+\rho} \right) \end{aligned}$$

is a partial derivative of the output signal power, where $\mathrm{Im}()$ denotes the imaginary part. It is a little time-consuming task to get this formula, but the mathematics is straightforward[1].

3.2.6. Frame-step control

The frame step is given by

$$\phi_r = R - \mathrm{round}(\lambda_r + \epsilon_r R), \tag{13}$$

where $\mathrm{round}()$ is a function that rounds off a number to the nearest integer. This means that, if the expected time difference between the window centers for the next frame, $(\lambda_r + \epsilon_r R)$, remains between -0.5 and 0.5, the standard value, R, is used. But if it exceeds 0.5, $R-1$ is used; and if it falls below -0.5, $R+1$ is used.

4. EXPERIMENTS

The parameters in Table 1 were used in the experiments described below. Hann windows were used for both the analysis and synthesis windows; and the filter length for the TDAEC evaluation was 4800 taps, which is equivalent to 300 ms.

A. Computational complexity and delay

Table 2 compares a standard TDAEC, an FDAEC, and our async-FDAEC in terms of computational complexity and algorithmic delay. The complexity is given by the number of multiplications. The results show that the computational cost of the async-FDAEC is about 1.5 times larger than that of the FDAEC. That is mainly due to eqs.(9) and (11). However, it gives no additional delay to the FDAEC, and it still has a great advantage over the TDAEC regarding the computational costs. For a faster implementation, the filter length,

[1] A practical trick is needed here to get this result. In order to update all the estimates at once within the length of the filter, all the estimates should be treated as if they were the same variable, i.e. $\epsilon_{r-L} = \epsilon_{r-L+1} = \ldots = \epsilon_{r-1}$.

Method	Num. of mult. / s	Delay
TDAEC	230 M	0 ms
FDAEC	16 M	32 ms
async-FDAEC	23 M	32 ms

Table 2. Computational complexity and algorithmic delay

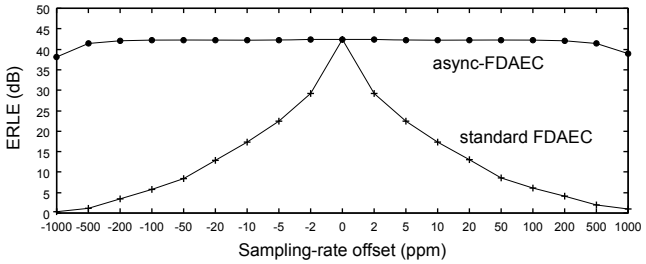

Fig. 4. Performance of proposed async-FDAEC

L, in eq.(11) should be made smaller because a long filter of 300 ms to cover late sound reflection is not needed to estimate the sampling-rate offset.

B. Simulation with Gaussian noise

Figure 4 shows the echo suppression capabilities of the FDAEC and the async-FDAEC with regard to sampling-rate mismatch. The horizontal axis is sampling-rate offset in units of ppm. The vertical axis is echo return loss enhancement (ERLE), which is a standard measure for evaluating an AEC[1]. It is defined as the ratio of the average power of the input to that of the output. The reference signal used here was white Gaussian noise, and the input signal was created by convoluting a room impulse response with a reverberation time (RT60) of about 400 ms, adding room background noise with an SNR of around 50 dB, and shifting the sampling interval by using a 5th-order polynomial interpolation.

When there is no sampling-rate offset, the two methods yield exactly the same ERLE of above 40 dB. With a sampling-rate offset, the async-FDAEC works effectively over a wide range of offsets from -1000 ppm to 1000 ppm, whereas even a slight offset rapidly degrades the ERLE of the FDAEC. For example, it is 20 dB at ± 10 ppm and 6 dB at ± 100 ppm.

C. Test with real-room recording

Figure 5 shows experimental results for the real-world recording of a TV broadcast in a living room. Two different USB audio interfaces were used: one for a loudspeaker and the other for a microphone. The sampling-rate offset between them was measured to be about 106 ppm. The input signal from the microphone is shown in (a). The output waveform of the FDAEC is shown in (b), and that of the async-FDAEC is shown in (c). With the FDAEC, the amplitude is only about -6 dB for the input; whereas, for the async-FDAEC,

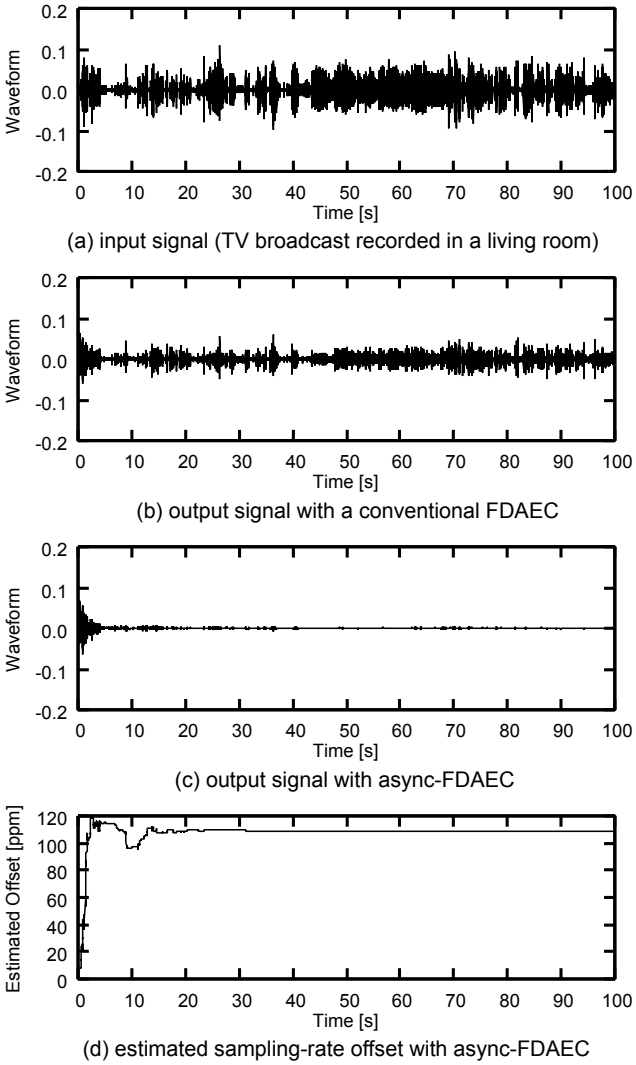

Fig. 5. Results for real-world recording

it reaches about -20 dB within 5 s and maintains an average ERLE suppression of about 30 dB. The estimation results for the sampling-rate offset are shown in (d). After some fluctuations up to 20 s, it converges to about 106, which is the correct offset between the devices used.

5. CONCLUSION

This paper describes a new method of acoustic echo cancellation that handles a sampling-rate mismatch between A/D and D/A conversion. It is a simple extension of the LMS algorithm in the frequency domain and works without an explicit resampling operation. The next step of our research will involve solving some practical problems, such as the handling of double talk, introducing the normalized LMS[10], and adding a multichannel extension to the algorithm.

6. REFERENCES

[1] C. Breining, P. Dreiseitel, E Hansler, A. Mader, B Nitsch, H Puder, T Schertler, G. Schmidt, and J. Tilp, "Acoustic echo control," *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 42–69, July 1999.

[2] G.-O. Glentis, K. Berberidis, and S. Theodoridis, "Efficient least squares adaptive algorithms for fir transversal filtering," *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 13–41, July 1999.

[3] J. W. Stokes and H. S. Malvar, "Acoustic echo cancellation with arbitrary playback sampling rate," *Proc. ICASSP*, pp. 153–156, 2004.

[4] H. Yasukawa, I. Furukawa, and Y. Ishiyama, "Acoustic echo control for high quality audio teleconferencing," *Proc. ICASSP*, pp. 2041–2044, 1989.

[5] K. Steinert, M. Schonle, C. Beaugeant, and T. Fingscheidt, "Hands-free system with low-delay subband acoustic echo control and noise reduction," *Proc. ICASSP*, pp. 1521–1524, 2008.

[6] E. Robledo-Arnuncio, T. S. Wada, and B. H. Juang, "On dealing with sampling rate mismatches in blind source separation and acoustic echo cancellation," *Proc. WASPAA*, pp. 34–37, 2007.

[7] M. Pawig, G. Enzner, and P. Vary, "Adaptive sampling rate correction for acoustic echo control in voice-over-ip," *IEEE Trans. Signal Processing*, vol. 58, no. 1, pp. 189–199, 2010.

[8] S. Miyabe, N. Ono, and S. Makino, "Blind compensation of inter-channel sampling frequency mismatch with maximum likelihood estimation in stft domain," *Proc. ICASSP*, pp. 674–678, 2013.

[9] S. Markovich-Golan, S. Gannot, and I. Cohen, "Blind sampling rate offset estimation and compensation in wireless acoustic sensor networks with application to beamforming," *Proc. IWAENC*, pp. 1–4, 2012.

[10] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE. Trans. Automatic Control*, vol. 12, no. 3, pp. 282–287, 1967.