QUALITY-DELAY-AND-COMPUTATION TRADE-OFF ANALYSIS OF ACOUSTIC ECHO CANCELLATION ON GENERAL-PURPOSE CPU

Justin Song, Jian Li, and Yen-Kuang Chen

Microprocessor Research Laboratories, Intel Corporation

ABSTRACT

While many previous studies have examined acoustic echo cancellation (AEC) in terms of quality, computation complexity, and implementation issues on DSP processors, this work evaluates quality-delay-computation trade-off of unconstrained frequency-domain recursive-least-square AEC algorithm on general purpose microprocessors. Specially, trade-off among echo cancellation quality, sampling delay, and computation time on Intel Pentium 4 systems is analyzed. Our quantitative analysis shows that the effectiveness of echo cancellation does not depend on availability of CPU as long as CPU can provide sufficient computational power for online real-time processing. Today's general-purpose microprocessor-based AEC can deliver satisfactory echo cancellation quality at a computationally acceptable price (less than 5% CPU usage). On other hand, the effectiveness depends on sampling delay. And no matter how fast a microprocessor would be, it is unlikely to guarantee both smaller sampling delay and larger echo-return-loss-enhancement (ERLE) at the same time. Finally, considering possible application of general-purpose processor-based AEC in laptop, office and meeting room environments, we analyzed acoustic channel delay's influence on both ERLE and CPU computation, showing that general-purpose microprocessor AEC's outstanding ability in tolerating various computing environments. Our experimental results can be used to design good configuration to meet specific quality requirements in terms of quality and sampling delay.

1. INTRODUCTION

Previous acoustic echo cancellation (AEC) related work successfully reduces its computational complexity: first adaptive filtering in the frequency-domain by Dentino [1], frequency-domain least mean square by Ferrara [2], improved unconstrained frequency-domain least mean square (UFLMS) by Mansour and Gray [3], and fatherly unconstrained frequency-domain recursive-least-square (UFRLS) by Benesty [4]. With UFRLS, an exact adaptive algorithm from the normal equation is derived and a constraint resulting in the UFLMS algorithm is removed. UFRLS is easy to be applied to multiple channel AEC, so it gains wide application.

AEC will not be successfully used in business multimedia and communication areas before cost is low enough and quality-computation trade-off is best met. Microprocessor centric computing environment can deliver the same echo cancellation quality, and its diluted cost per numerous PC applications is lower than DSP solutions. The primary issue is to identify best balance between quality and computation: both good enough quality and moderate computation requirement. The main quality metrics are echo-return-loss-enhancement (ERLE) and sampling delay: the former reflects how well echo signal is cancelled, and the latter influences acoustic perception. Although there are some work in DSP based AEC [5][6], such as N-point (N<L) multidelay adaptive filter [7] that improves hardware efficiency and gain smaller delay and faster filtering updating speed, and overlap factor α adjustable solutions [8][9] that update filter coefficients every N/α samples (where α >1) and achieve smaller delay and better canceller traceability at the price of more computation, unfortunately these analysis on DSP solutions makes little sense on general-purpose microprocessor, because CPU's flexibility and dynamics in execution cores and memory system are different with DSPs.

This paper evaluates general-purpose microprocessor-based AEC's ERLE, sampling delay, MIPS (Million of Instructions per Second), and their trade-off. Section 2 discusses candidate quality and computation candidate metrics. Section 3 defines a set of configurations, each composed of distance between loud speaker and microphone, block size, sampling frequency, and channel delay. Section 4 shows our experimental results, by which application users can easily choose good configuration and tune microprocessor's computation to meet their specific quality requirements in terms of ERLE and sampling delay. Section 5 discusses MIPS and section 6 is a summary.

2. QUALITY AND COMPUTATION METRICS

We choose the classical overlap save (OLS) method, with overlap factor $\alpha = 2$ and forgetting factor $\lambda = 0.95$. Please find pseudo code description of this algorithm in appendix.

We use ERLE and sampling delay to evaluate quality of general-purpose microprocessor-based AEC. The former reflects how well echo is cancelled, and the latter influences end-to-end perceptible audio quality and interactive applications' timeliness.

ERLE is a standard measurement of the amount of reduction of echo power and is defined as:

$$\text{ERLE} = 10\log_{10}\left(\sigma^2_{input} / \sigma^2_{output}\right) \qquad (1)$$

where ideally σ^2_{input} and σ^2_{output} are the echo signal power and echo residual signal power, respectively. Practically they are estimated via the microphone signal power before and after the echo cancellation algorithm.

Because AEC algorithm may buffer some sample points before processing them, there is a sampling delay defined as:

$$\text{D(sampling)} = \frac{N}{F} \qquad (2)$$

where N is the number of sample points processed by AEC algorithm each time, and F is sampling frequency. For example, D(sampling) for a 512 sample point and 48K sampling frequency AEC is 10.67ms.

Usually people use MIPS as CPU performance indicator. To reflect relationship between real-time AEC processing and CPU performance, we adopt an "online MIPS", which means the minimal MIPS needed for AEC real-time processing. It is defined as follows:

Online MIPS = "# of instructions AEC retired" / "sampling delay" (3)

To better understand analysis results, we'll evaluate some micro-architectural metrics such as IPC (instruction per cycle) and cache misses. IPC is usually considered indicator of practical delivered CPU performance other than theoretical performance. For example the maximal theoretical IPC value of Pentium 4 is 3.0 [10], but actually most media applications can only reach 1.0 or even slightly below. In most cases IPC decreases because its working set exceeds microprocessor cache size, so we are often interested in cache misses.

There is a online real-time problem to point out: if time needed for processing of one block exceeds its sampling delay, this applications can not be running online. We define D(processing) as processing delay. To enable online real-time AEC, processing delay should be less than sampling delay. Actually, because general-purpose microprocessor may serve many concurrent applications, we reasonably expect not too much CPU consumed by AEC, for example not exceeding 10%. Then we have a constraint criterion as follows:

D(processing) / D(sampling) <= 0.1 (4)

3. GENERAL-PURPOSE MICROPROCESSOR-BASED AEC CONFIGURATIONS

We should define baseline configuration before evaluation. AEC parameters include block size L, sampling frequency F and channel delay (CD). Here we use "block size" to refer to number of sample points that is processed and used to update filter by AEC algorithm.

Table1: AEC quality-computation trade-off analysis baseline configurations

D(m)	F (KHz)	CD	L
1	8	24	64
	16	47	128
	44	129	512
	48	141	512
	96	282	1024
	192	564	2048
2	8	47	128
	16	94	256
	44	259	1024
	48	282	1024
	96	564	2048
	192	1129	4096
5	8	118	256
	16	235	512
	44	647	2048
	48	706	2048
	96	1412	4096
	192	2824	8192

Normally the block size ranges from 64 sample points to 16384 sample points. Larger block sizes such as 32768, 65536 sample points or even more, are unlikely to be used. Sampling frequency is chosen from 8KHz through 192KHz, including most frequently used 44.1KHz and 48KHz. CD means number of sample points deferred when echo signal reaches microphone which is determined jointly by sampling frequency and distance D between loud speaker and microphone. Considering typical applications on laptop, office and meeting rooms, we choose the distance to be 1 meter, 2 meters and 5 meters respectively. This is a very loose configuration – actually in many DSP based solutions the distance is less than 1m. Plus assuming sound speed is 340m/s in air, we have

CD = F * D / 340 (5)

Our baseline AEC configuration is shown in Table 1.

Now we are ready to analyze trade-off between AEC quality and computation. We are most interested in two basic questions:

1. Can general-purpose microprocessor-based AEC eliminate echo signal, both effectively and work in real-time?
2. How many CPU MIPS are needed for an ERLE and/or sampling delay?

It is important to check whether existing and emerging general-purpose microprocessor platforms can deliver high qualities at computationally acceptable price.

4. MEASUREMENT AND RESULT ANALYSIS

General-purpose microprocessor-based acoustic echo canceller can surely deliver both satisfactory echo cancellation quality (here we assume that ERLE>40 is good [11]) for human perception. There are always configurations points falling in the ERLE>40 zone with whatever block size ranging from 64 to 16384 as shown in

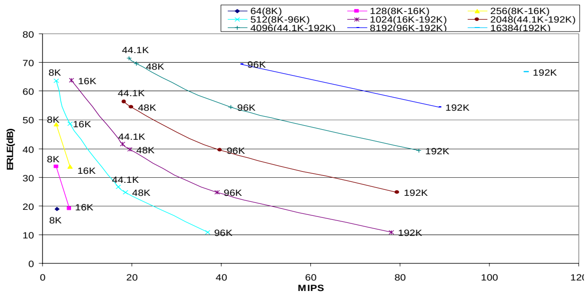

Figure 1a: ERLE vs. online MIPS on 2GHz Pentium 4. Curves are plotted per equal block size.

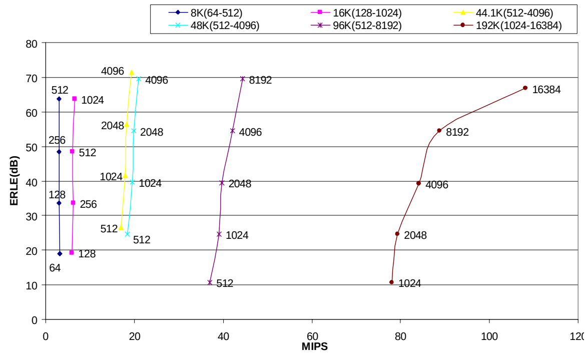

Figure 1b: ERLE vs. one block processing time on 2GHz Pentium 4 2.0G. Curves are plotted per equal sampling frequency.

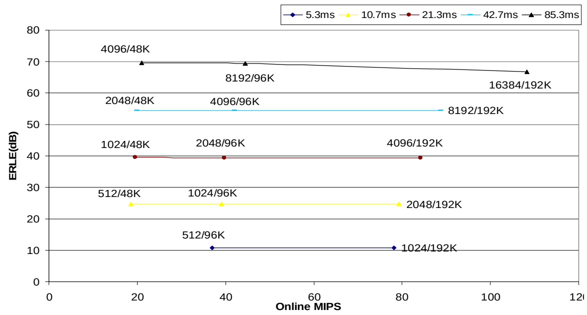

Figure 1c: ERLE vs. online MIPS on 2GHz Pentium 4. Curves are plotted per equal sampling delay.

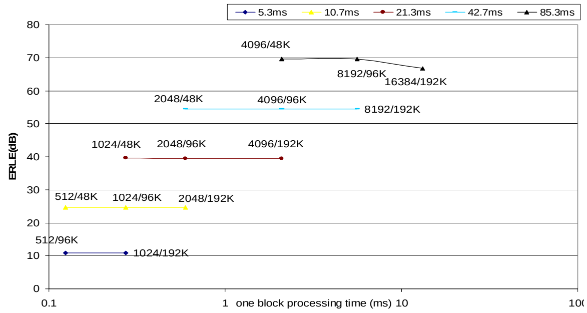

Figure 1d: ERLE vs. one block processing time on 2GHz Pentium 4 2.0G. Curves are plotted per equal sampling delay.

Figure 1a, and whatever frequency ranging from 8KHz to 192KHz as shown in Figure 1b. And small enough sampling delay (here we assume that sampling delay < 50ms is good) can also be achieved, as shown in Figure 1c. The online real-time processing is also of no problem, as shown in Figure 1d, that according to (4) D(processing)/D(sampling) is small. Hence, the first question above is answered: general-purpose microprocessor-based AEC does both well eliminate echo signals and work at a high computational efficiency.

Block size is chosen after channel delay, and channel delay is determined by frequency if distance between loud speaker and microphone is fixed. Consequently it is important to discuss how to choose block size according to sampling frequency. Here we discuss three cases: a) ERLE is more important, b) sampling delay is more important, and c) both are important.

First, for better ERLE with a fixed sampling frequency, as shown in Figure 1b, larger block size gives better ERLE. There are very limited increases in online MIPS needed. However, sampling delay increases noticeably according to (2). This means if designers can tolerate a little bit larger sampling delay, e.g., 42.7ms or 85.3ms, they can obtain excellent ERLE, e.g., 55dB or even 70dB, at nearly the same computational price as low moderate ERLE configurations.

Second, for better sampling delay with a fixed sampling frequency, according to (2), it is obvious that we should decrease block size, although there would result in worse echo cancellation effect, as shown in Figure 1b.

Third, if both are important, we check Figure 1c or Figure 1d and find an interesting fact: as long as general-purpose microprocessor-based AEC can work in real-time, and without changing the sampling delay, ERLE does not depend on availability of CPU any more. For example, with the most frequently used 10.7ms sampling delay, if we increase block size from 512 to 2048, we pay for 4.3x more MIPS, but harvest only the same ERLE. Thus, we can firstly choose the configuration to fulfill the sampling delay requirements, and then can smartly choose the smallest block size, using the lowest sampling frequency, as long as it can cover channel delay. Here note block size must be greater than channel delay, for otherwise AEC doesn't work at all. However, considering possible various distances between loud speaker and microphone while working, we suggest designers make conservative choice of block sizes. For example, at least 2x channel delay.

What if we need tough constraints on both ERLE and sampling delay at the same time? This is the most difficult situation for DSP-based AEC, as we have seen in previous work. General-purpose microprocessor-based AEC has the same difficulty. As shown in Figure 1c or Figure 1d, larger ERLE (better echo cancellation) always results in larger sampling delay (worse for end-to-end acoustic perception), and vice versa. For example, in Figure 1c, only 42.7ms line meets ERLE > 40dB and sampling delay < 50ms. If tougher requirements imposed, e.g. ERLE>60dB and sampling delay < 25ms, no solution.

Till now we have actually answered the second question listed in the last section.

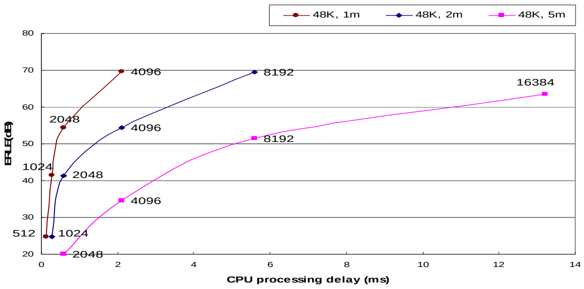

Figure 2: different distances' influence on ERLE and processing delay.

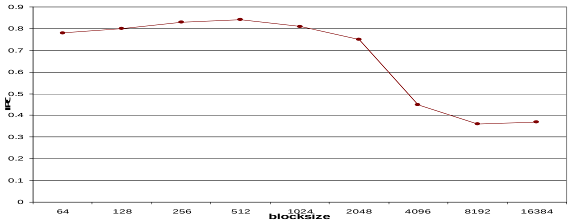

Figure 3: IPC vs. different block size

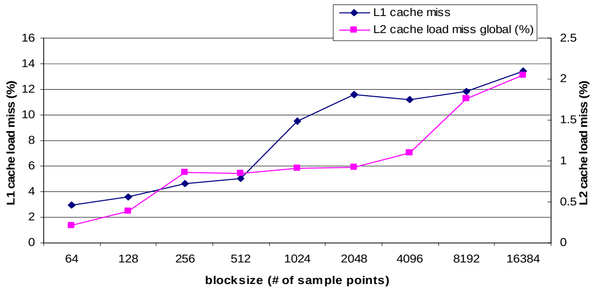

Figure 4: first-level and second-level cache load miss increases with block size

As Table 1 defines different distances between loud speaker and microphone, here we can check different distance's influence on computation and ERLE. Figure 2 chooses the most frequently used 48KHz sampling frequency. To get 40dB ERLE at 1m distance, for example, we can choose block size to be 1024 which results in only 0.27MIPS. However, to get the same ERLE at 2m distance, we have to choose block size to be 2048, which results in 0.6MIPS. If the distance is 5m, things go worse sharply. This is because block size have to be 8192: its needed MIPS is 5.6MIPS, 20x more than that of 1m, and 10x more than that of 2m. Nevertheless, general-purpose microprocessor can guarantee AEC to be workable for either laptop, office, or meeting room environments, whereas this is difficult for DSPs.

5. DISCUSSION

In Section 4, we use MIPS as computational metric, but it is not accurate. Actually MIPS only roughly reflects CPU resources needed, and IPC (as shown in Figure 3) and CPU processing time (as shown in Figure 1d) accurately indicates CPU's delivered performance. The higher IPC, the higher MIPS CPU can deliver. Figure 4 further analyzes cache misses, one factor that greatly influences IPC. All these are special characteristics on general-purpose microprocessor-based AEC.

6. SUMMARY

We analyzed trade-off among ERLE, sampling delay and MIPS of general-purpose microprocessor-based AEC. Our results show that it can effectively eliminate echo signals at high computational efficiency. It is difficult to guarantee both smaller sampling delay and better ERLE.

APPENDIX

Pseudo code description of AEC algorithm:

```
    R: Reference; S: Microphone; E: Error signal
    H: Estimated transmission channel impulse response
BlockProcessing(R, S, E) {
    Convert: from time domain to frequency domain for R
    Convolution: R and H
    Convert: from frequency domain to time domain
    Sub: Achieve E by S sub time domain of convolution result
    Convert: from time domain to frequency domain for E
    Update $S_u$ value and H }
Main {
    Initialization
    For BlockIndex = 0 to N BlockProcessing(R, S, E)
}
```

REFERENCE

[1] M. Dentino, etc, "Adaptive Filtering in the Frequency Domain", Proc. of IEEE, vol. 66, pp. 1658-1659, Dec. 1978.

[2] E. R. Ferrara, "Fast Implementation of LMS Adaptive Filter", IEEE Trans. on Acoustic, Speech, Signal Processing, vol. 28, pp. 474-475, Aug. 1980.

[3] D. Mansour and A. H. Gray, "Unconstrained Frequency-Domain Adaptive Filter", IEEE Trans. on Acoustic, Speech, Signal Processing, vol. 30, pp. 726-734, Oct. 1982.

[4] J. Benesty and D. R. Morgan, "Frequency-Domain Adaptive Filtering Revisited, Generalization to the Multi-Channel Case, And Application To Acoustic Echo Cancellation", Proc. on ICASSP, vol. 2, pp. 789-792, Apr. 2000.

[5] S. L. Gay and R. J. Mammone, "Fast Converging Subband Acoustic Echo Cancellation Using RAP on the WE DSP16A", Proc. on ICASSP, pp. 1141-1144, May 1990.

[6] M. Tahernezhadi and L. Liu, "Real-Time Implementation of an IIR Acoustic Echo Canceller on ADSP21020", Proc. on ICASSP, pp 2723-2726, May 1995.

[7] J.-S. Soo and K. K. Pang, "Multidelay Block Frequency Domain Adaptive Filter", IEEE Trans. on Acoustic, Speech, Signal processing, vol. 38, pp. 373-376, Feb. 1990.

[8] E. Moulines, O. A. Amrane, and Y. Grenier, "The Generalized Multidelay Adaptive Filter: Structure and Convergence Analysis", IEEE Trans. on Signal Processing, vol. 43, pp. 14-28, Jan. 1995.

[9] J. Prado and E. Moulines, "Frequency-Domain Adaptive Filtering With Applications to Acoustic Echo Cancellation", Ann. Telecommun, vol. 49, pp. 414-428, 1994.

[10] Intel Corporation, "Intel® Pentium® 4 and Intel® Xeon™ Processor Optimization", http://www.intel.com/design/pentium4/manuals/248966.htm

[11] ITU-T Recommendation G.167, "Acoustic echo controllers", March 1993.