Implementation Issues for acoustic echo cancellers

Berkeman, Anders; Öwall, Viktor; Torkelson, Mats

Published in:
[Host publication title missing]

DOI:
10.1109/MWSCAS.1999.867218

1999

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Implementation Issues for Acoustic Echo Cancellers

Anders Berkeman, Viktor Öwall and Mats Torkelson
Department of Applied Electronics,
Lund University, Sweden
Email: {abn,vikt,torkel}@tde.lth.se

***Abstract*— The high computational complexity of acoustic echo cancellation algorithms requires application specific implementations to sustain real time signal processing with affordable power consumption. This is especially true for systems where a delayless approach is considered important, e.g. wireless communication systems. The proposed paper presents architectural considerations to reach a feasible hardware solution.**

I. Introduction

This paper presents implementation aspects of an Application Specific Digital Signal Processor (ASDSP) designed to perform acoustic echo cancellation. The need for acoustic echo cancellers arises in systems where two or more people positioned at different locations are having a conversation using loudspeakers and microphones, for example a teleconferencing system. The problem is the acoustic path from loudspeaker to microphone. When the far end talker is speaking, the speech signal is fed to the near end loudspeaker, it enters the room and goes back to the far end through the microphone. Due to the various delays in the signal path, such as the acoustic delay from loudspeaker to microphone, coding delay, transmission delay etc, this returned signal is percepted as an annoying echo. The acoustic impulse responses considered have a duration in the order of hundreds of milliseconds. With a sample rate of 16 kHz this corresponds to roughly 2000-4000 samples. Such long impulse responses make a fullband approach unattractive for both convergence and complexity considerations and a subband approach is investigated [1].

There are two major candidates for implementation, where the first one is performing cancellation in the subband domain [2]. This algorithm is depicted in figure 1. The signals from the far end, $x(n)$, and from the microphone, $y(n)$, are filtered through filterbanks. Impulse response estimation is done on the corresponding subband signals, and the error signals for each subband adaptive filter are put together to a fullband signal using a synthesis filterbank. In this algorithm, there is a delay in the signal path of at least one filterbank delays.

The second candidate performs the actual echo cancellation in the time domain [1], see figure 2. This is achieved by a fullband finite impulse response (FIR) filter that convolves the input signal with the estimate of the room acoustic impulse response. In this algorithm, there is no delay in the signal path, but the estimate of the room acoustic impulse response is lagged. Due to the extra fullband FIR filter, this solution has higher complexity than the algorithm in figure 1.

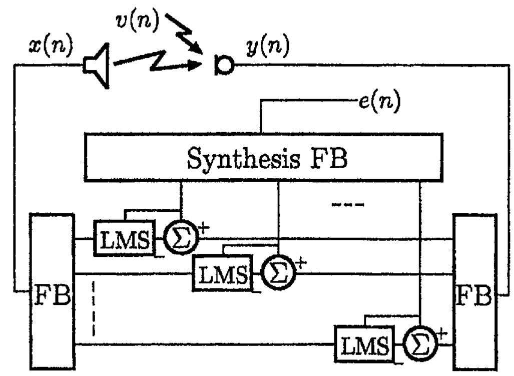


Fig. 1. The echo cancellation algorithm from [1]. The signals to the loudspeaker, $x(n)$, and from the microphone, $y(n)$, are split into subbands, and cancellation is performed in each subband prior to fullband reconstruction.

In wireless and multimedia applications, delay in the signal path is a serious obstacle. Therefore, the delayless cancellation algorithm will be investigated. The computational complexity of the algorithms makes a standard digital signal processor implementation difficult, especially in a mobile terminal scenario were power consumption is a key parameter. Therefore, an application specific solution is investigated to increase the throughput at the same time as the power consumption is reduced.

II. The Cancellation Algorithm

The main parts of the algorithm are shown in figure 2. To the upper left the far end signal $x(n)$ enters and connects to the canceller and to the loudspeaker. The signal from the microphone $y(n)$ is the other input to the canceller. Between the loudspeaker and the microphone is the acoustic signal path, where echo and a near end talker signal $v(n)$ is added. After the microphone, the

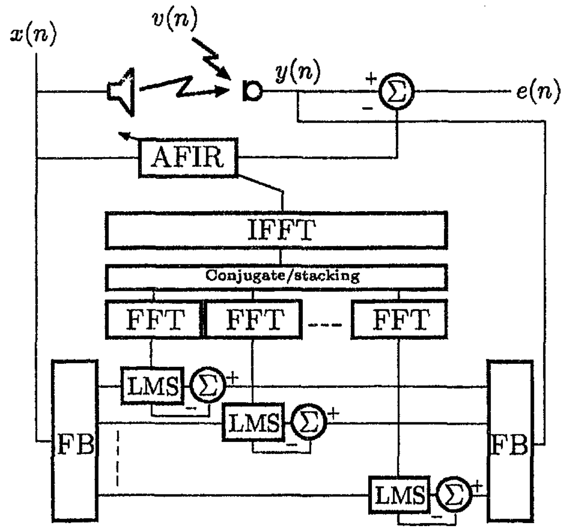


Fig. 2. The subband echo cancellation algorithm, from [2]. Echo cancellation is done in the time domain.

far end signal with estimated echoes are subtracted, optimally leaving nothing but the near end talker. This signal $e(n)$ is then fed back to the far end.

The heart of the algorithm is a number of adaptive Least-Mean-Square (LMS) filters that track the frequency response from speaker to microphone. Each adaptive filter acts on a small frequency band of the x and y signals. If the number of subbands is large, each subband has a narrow bandwidth with near flat frequency response, and therefore convergence of the adaptive filters is fast.

The estimated filter taps from one adaptive filter represent a part of the total impulse response in a certain frequency band. To make a fullband impulse response, these taps have to be Fourier transformed, stacked in frequency and inverse transformed. This fullband impulse response is, due to the delays in the filterbanks, adaptive filters and FFTs, a delayed estimate of the room acoustics. It is used to filter the far end signal to simulate the effect of the sound traveling through the room. The difference between this filtered signal and the microphone signal should be close to zero, except for the sound added in the near end.

III. Analysis of the Algorithm

A. The Filterbanks

An M band filterbank consists of M finite impulse response (FIR) filters of length K, where each filter selects a part of the signal spectrum. For every input sample, KM multiplications are executed in the filters, and M outputs are calculated, one for each subband. This can be seen as an upsampling by a factor M in each band. All these extra samples have to be processed by the following parts of the algorithm, giving a large overhead in the number of operations per sample. To reduce the amount of data, the outputs from the filterbank are downsampled by a factor M/α.

Applying downsampling at the output decreases the computational burden in the filterbanks, since every filter only has to be updated every M/α sample. Furthermore, if the M frequency functions of the filters are chosen as shifted versions of one prototype filter H, the polyphase filterbank approach can be used [1], [3]. The number of taps in each filter can then be decimated from K to K/M. The price for this reduction is an extra IFFT of size M at the outputs of the filters, updated every α/M input sample period, but the total number of operations in the filterbank is reduced considerably.

B. The LMS Filters

For each input sample, the output from one band of the filterbank is α/M samples. Due to symmetry in the frequency plane, only $M/2+1$ bands contain unique information [1], and thus there are $M/2+1$ LMS filters to be updated, each every α/M input sample period. This results in a total of

$$\frac{\alpha}{M}(\frac{M}{2}+1)=\alpha(\frac{1}{2}+\frac{1}{M}) \tag{1}$$

LMS filters to be updated every sample period. If M is reasonably large, the expression is close to $\alpha/2$ and does not vary much with M.

C. Generation of Fullband Filter Taps

The adaptive weights calculated in the $M/2+1$ LMS filters are combined into a fullband impulse response of length N. This is achieved by Fourier transforming the LMS weights, stacking them in frequency into a fullband frequency function, and inverse Fourier transform to get an impulse response. If each adaptive filter is of length T, T/α weights are taken from the Fourier transform of the middle $M/2-1$ filters, and $T/2\alpha$ from the Fourier transform of the left- and rightmost filters. Together this gives

$$(\frac{M}{2}-1)\frac{T}{\alpha}+2\frac{T}{2\alpha}=\frac{M}{2}\frac{T}{\alpha}=\frac{N}{2} \tag{2}$$

bins that are combined into a fullband frequency function of length N. The bins are stacked from position 0 to $N/2-1$, and then the complex conjugate is repeated in reversed order from $N/2$ to $N-1$. From equation (2) the length of each LMS filter can be calculated as

$$T=\alpha\frac{N}{M} \tag{3}$$

D. Complexity Analysis

Combining equation (1) and (3) gives to hand that the number of LMS weights to be processed per input sample

M	Mmul/s	FB(%)	LMS(%)	FIR(%)
2	889	4	90	4
4	366	5	82	9
8	179	5	69	18
16	105	4	53	32
32	72	4	36	46
64	57	3	22	58
128	49	3	13	67
256	45	3	7	72
512	43	3	4	76
1024	42	3	2	78

TABLE I

The number of operations for the echo canceller as a function of the number of subbands, M, in millions of multiplications per second. The three columns to the right shows the percentage of the multiplications spent by the filterbanks (FB), adaptive filters (LMS) and fullband time domain filter (FIR) respectively. The remaining percents are due to the fullband impulse response reconstruction.

clock period is

$$\alpha^2 \frac{N}{M}(\frac{1}{2} + \frac{1}{M}), \tag{4}$$

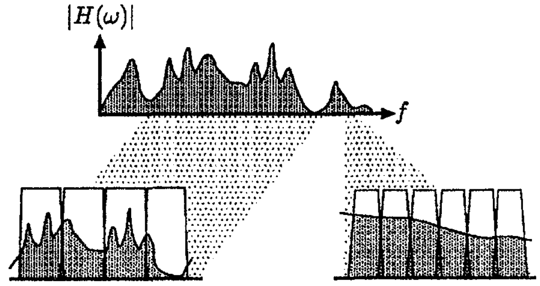
which is inverse proportional to M.

How M affects the total complexity of the algorithm is shown in table I. The table is created assuming a fullband filter length N of 2048 taps and a constant α of two.

The first column is the number of subbands, ranging from two to above a thousand. The second column is an approximation of the required number of real multiplications per second in millions. The number of additions in the LMS and FIR filters are about the same as the number of multiplications, due to the intensive use of the multiply-accumulate (MAC) operation. A multiplication has several times higher complexity than an addition, and therefore only multiplications are considered in this table. It is assumed that a complex multiplier corresponds to four real number multipliers, but their implementation complexity can be reduced to about half that number [4]. Divisions have higher complexity, but are not used as frequent as multiplications in this algorithm.

Column three to five show how much of the total number of calculations that are carried out by the filterbanks, adaptive filters and the fullband FIR filter respectively. In the simulations the filter was updated every 128th sample [1]. This corresponds to the additional percents in table I and goes from 2% for $M = 2$ to 17% for $M = 1024$. If a higher update rate is desired, this will increase the total complexity considerably making an ASDSP solution even more advantageous.

A large number of subbands M gives a low overall complexity of the implementation. It can be seen from the table that as the number of subbands increase the dominating factor becomes the time domain fullband filter as contrary to the adaptation algorithm for a lower number of subbands. The drawback is the increased requirements on the prototype filter in the subband filterbanks, since the frequency response of the filter must have a more high defined selectivity. As the signal is never reconstructed from the subband decomposition, there are no "perfect reconstruction" criteria. Since the frequency response of the filters in a filterbank are abutted, care has to be taken especially in the transition part of the prototype filter to make the group delays between two filters equal.



Few wide subbands Many narrow subbands

Fig. 3. Having many subbands result in a near flat spectrum in each subband.

For large values of M, the passband region for each filter in the filterbank becomes very narrow. Therefore, a signal filtered by such a filterbank can be considered to have a near flat power spectrum, see figure 3. This is beneficial for the convergence of the adaptive filters. So, not only does a large M reduce the number of operations, but the echo canceller will also have a faster convergence time. Simulations have shown that for a fullband filter length of $N = 2048$ taps, an M value of 256 to 512 is suitable.

As shown in table I, for M greater than 32 the heaviest computational burden is in the fullband FIR filter. For every input sample, this filter calculates 2000-4000 multiplies and accumulations. For a sample rate of 16 kHz, this corresponds to 32 to 64 million multiplies and accumulates per second. Also, the taps in this filter are updated hundreds of times per second to keep track of changes in the room acoustics.

IV. Optimizations

A. Wordlengths

Wordlength affects performance in several ways. Small lengths consume less power and chip area, i.e. buses, computational units and memories should have minimum widths. On the other hand, the wordlengths also determine the performance in terms of resolution and dynamic range. Therefore, it is important to keep signals wide enough to keep overflow and rounding errors at a minimum. These optimizations are performed on each bus, arithmetic unit and memory, so that different hardware units can have different wordlengths. The effects of this truncation into fixed point are investigated using simulations.

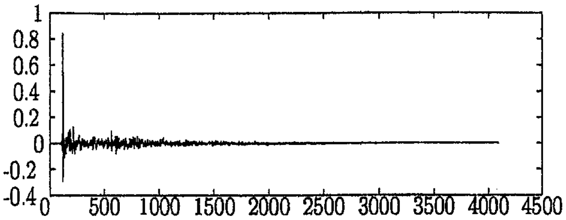


Fig. 4. A typical acoustic impulse response of a conference room. Note the different amplitude characteristics along the time axis.

B. Memory Management

Memories are often a bottle-neck in digital signal processors (DSPs), since a memory has a certain number of ports and a limited access frequency. Furthermore, every access has a cost in energy. The larger the memory, the larger energy consumption. In a typical ASDSP the power consumed by memory accesses is a substantial part of the total consumption. By memory management the number of accesses to large memories is reduced by rescheduling the arithmetic operations and utilizing memory hierarchies [5].

C. The Fullband FIR Filter

Since the taps in the fullband FIR filter are an estimate of a room acoustic impulse response, they have a certain shape. By exploring the properties of typical impulse responses, some optimizations of the fullband FIR filter can be made. Figure 4 shows a recorded impulse response. It begins with zeros for a time corresponding to the shortest distance from speaker to microphone, and then a set of peaks with high amplitude, created by the direct wave and the first main reflections. From there on, the amplitude is decreasing.

To cover the full dynamics of the impulse response, a multiply and accumulate unit with very large dynamic is required. If, on the other hand, the amplitude pattern of the impulse response is explored the filter can be split in parts, where each part has its own multiply and accumulate unit with certain dynamics. This gives a significant decrease in the number of bit transitions per sample, and a corresponding decrease in power consumption.

V. Hardware Mapping

One extreme implementation would be to map the algorithm directly onto hardware, i.e. each operation is directly mapped onto a dedicated hardware module. Due to the high complexity of the algorithm the number of computational units would be very large at the same time as the clock frequency would be extremely low, and therefore this is not an attractive solution.

One solution is to reuse hardware and have a few components that perform several different operations for each input sample. For example one adaptive filter circuit can be used to calculate the filter coefficients for all the subbands. The extreme here is a standard DSP with only a single multiplier and adder, performing all computations sequentially. The challenge is to find a trade-off between speed, power, area and flexibility in order to find a solution with good characteristics.

A modest clock rate should be used, since a low clock rate gives the opportunity to lower the supply voltage and thereby reduce power [6]. A sample rate of 16 kHz and a clocking frequency of the processor of 16 MHz gives 1000 clock cycles per sample, which is reasonable for the algorithm. Thus, the application specific architecture enabling the low clock frequency and parallel processing reduce power consumption.

When hardware is reused for different tasks, there is a need for a controller to schedule the operations to the correct computation unit at the correct moment. For this scheduling to be as effective as possible, careful investigation of the memory access pattern for various algorithmic transformations is done [5].

VI. Summary

For a hardware implementation of an acoustic echo canceller, an application specific digital signal processor solution is chosen. This solution gives a streamlined architecture in terms of hardware utilization and costly memory accesses, yielding realtime signal processing at an affordable overall power consumption. This is especially important in wireless multimedia systems.

Important aspects for the implementation are optimization of wordlength for buses and arithmetic units, determining of crucial parameters such as the number of subbands and construction of the control unit for steering the dataflow such that a good memory management is achieved.

References

[1] Dennis R. Morgan and James C. Thi. "Adaptive echo Cancellation for Speech Signals". In *IEEE Transaction on Signal Processing*, volume 43, No. 8, August 1995.

[2] M. Mohan Sondhi and Walter Kellerman. "Adaptive echo Cancellation for Speech Signals". In *Advances in Speech and Signal Processing*, edited by S. Furui and M. M. Sondhi, chapter 11. Marcel Dekker, New York, 1992.

[3] Jr. E. R. Ferrara. "Frequency-domain adaptive filtering". In *Adaptive filters*, edited by C. F. N. Cowan and R. M. Grant, chapter 6, pages 145–179. Prentice-Hall, 1985.

[4] A. Berkeman, V. Öwall, and M. Torkelson. "A Low Logic Depth Complex Multiplier". In *Proceedings of the 24th IEEE European Solid-State Circuits Conference*, 1998.

[5] Francky Catthoor, Sven Wuytack, Eddy De Greef, Florin Balasa, Lode Nachtergaele, and Arnout Vandecapelle. *"Custom Memory Management Methodology"*. Kluwer Academic Publishers, 1998.

[6] A. P. Chandrakasan and R. W. Brodersen. "Minimizing Power Consumption in Digital CMOS Circuits". *Proceedings of the IEEE*, 83(4), april 1995, pp. 498–523.